

## **AUTOMATED STRUCTURED, DIAGRAM QUESTION MARKING AND DOUBT CLARIFICATION SYSTEM**

<b>Name</b>	<b>ID Number</b>
R.R.A.M.P.Jayawardena	IT15037916
P.S.Suriyaarachchi	IT15035868
G.A.D.Thiwanthi	IT15040954
K.I.Withana	IT15057198

Bachelor of Science (Hons) in Information Technology

Department of Software Engineering

Sri Lanka Institute of Information Technology  
Sri Lanka

October, 2018

# **AUTOMATED STRUCTURED, DIAGRAM QUESTION MARKING AND DOUBT CLARIFICATION SYSTEM**

<b>Full Name</b>	<b>ID Number</b>
Rajapaksha Ranathunga Arachchige Mayantha Piuman Jayawardena	IT15037916
Pawani Saumya Suriyaarachchi	IT15035868
Gammanpila Arachchige Dinusha Thiwanthi	IT15040954
Kashmiera Iroshinie Withana	IT15057198

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

Department of Software Engineering

October, 2018

### **DECLARATION**

We declare that this is our 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, we hereby grant to Sri Lanka Institute of Information Technology the nonexclusive right to reproduce and distribute our dissertation, in whole or in part in print, electronic or other medium. We 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 B.Sc Dissertation under my supervision.

Signature of the supervisor:

Date

## **ABSTRACT**

Question Paper Marking has always been an exhaustive process which requires a lot of time and effort. We intend to address this issue by automating the marking of Structured and three Diagram-type questions which are Block, Logic Gate, and Flowchart diagrams via a web-based solution, providing marks and feedback to student answers at the end of the process. While automated marking of structured questions have been a tough area of research for many years, less research has been carried out in the area of diagram-type question marking. Time consumption is a problem faced by students when clarifying their subject-related doubts. Doubt Clarification feature will handle this issue by retrieving doubt-related information from the system through PDF references. We investigated the application of Natural Language Processing in Structured Question Marking and Doubt Clarification. Graph Matching using Depth-First Search (DFS) and Breadth-First Search (BFS) was employed in marking diagram-type questions while Logic Circuit Simulation and Flowchart conversion to a Python program was done to verify the logic of the corresponding diagrams. The system produced through this research to support these three major features will be immensely useful for teachers and students worldwide in carrying out online exams, and in clearing doubts.

**Keywords:** Structured Question Marking, Diagram Question Marking, Question Classification, Information Retrieval

## **ACKNOWLEDGEMENT**

We would like to thank Prof. Chandimal Jayawardena for guiding us continuously, and ensuring we took the right path by motivating us and providing us feedback, enlightening us on the areas we could improve on while sharing his knowledge and wisdom.

Dr. Dharshana Kasthurirathna, Mr. Isuru Kumarasiri, and Mr. Indika Baddegama are other lecturers we would like to thank for sharing their wisdom and helping us clarify the research-related doubts.

We would also like to show our gratitude to Mr. Ravi Deepali, Ms. Y. V. Wijesinghe, Mr. O. Rajitha Madhawa de Silva, and Ms. O. S. Samarasinghe for assisting us in testing the diagram marking module of our system by marking the student answers we collected.

## TABLE OF CONTENT

<b>DECLARATION</b>	i
<b>ABSTRACT</b>	ii
<b>ACKNOWLEDGEMENT</b>	iii
<b>TABLE OF CONTENT</b>	iv
<b>LIST OF TABLES</b>	v
<b>LIST OF FIGURES</b>	v
<b>LIST OF ABBREVIATIONS</b>	vii
<b>1.0 INTRODUCTION</b>	1
<b>1.1 Background and literature reviews</b>	1
<b>1.2 Research gap</b>	3
<b>1.3 Research Problem</b>	8
<b>1.4 Research objective</b>	9
<b>2.0 METHODOLOGY</b>	12
<b>2.1 Methodology</b>	12
<b>2.2 Commercialization aspects of the product</b>	19
<b>2.3 Testing and Implementation</b>	19
<b>2.3.1. Block diagram implementation testing</b>	21
<b>2.3.2. Logic circuit graph matching implementation testing</b>	23
<b>2.3.3. Logic circuit simulation implementation testing</b>	25
<b>2.3.4. Flowchart-to-Python-Program-Conversion and Program Execution implementation testing</b>	26
<b>2.3.5. Flowchart graph matching implementation testing</b>	28
<b>2.3.7 Question classification feature</b>	31
<b>2.3.8 Doubt clarification feature</b>	31
<b>3.0 RESULTS AND DISCUSSION</b>	34
<b>3.1 Result</b>	34
<b>3.2 Research Findings</b>	49
<b>3.3 Discussion</b>	50
<b>4.0 SUMMARY OF EACH STUDENT'S CONTRIBUTION</b>	52
• Commercialization aspects of the product	52
• Conclusion	52
<b>5.0 CONCLUSION</b>	54
<b>6.0 REFERENCES</b>	57

**LIST OF TABLES**

		Page
Table 1.2.1	Comparison of Existing Systems vs our System for Question Marking	6
Table 1.2.2	Comparison of Existing Systems vs our System for Doubt Clarification	7
Table 3.1.1	Comparison Of System-Allocated Marks Vs Actual Examiner-Allocated Marks For Block Diagram Marking	36
Table 3.1.2	Comparison Of System-Allocated Marks Vs Actual Examiner-Allocated Marks For Logic Circuit Marking	37
Table 3.1.3	Comparison Of System-Allocated Marks Vs Actual Examiner-Allocated Marks For Flowchart Marking	38
Table 3.1.4	Extracted headings and real topics comparison	46
Table 3.1.5	Result table for information retrieval feature	49

**LIST OF FIGURES**

Page

Figure 2.1.1	Semantic Similarity Score Calculation Equation	13
Figure 2.1.2	Cosine Similarity Equation	13
Figure 2.3.1.1	Block Answer Diagram of Teacher	21
Figure 2.3.1.2	Block Answer Diagram of first student	21
Figure 2.3.1.3	Output for block diagram answer of first student	22
Figure 2.3.1.4	Block Answer Diagram of second student	22
Figure 2.3.1.5	Output for block diagram answer of second student	22
Figure 2.3.2.1	Logic Circuit Answer of Teacher	23

Figure 2.3.2.2	Logic Circuit Answer of first student	24
Figure 2.3.2.3	Output for Logic Circuit Answer of first student	24
Figure 2.3.3.1	Logic Circuit Answer of Teacher	25
Figure 2.3.3.2	Logic Circuit Answer of first student	25
Figure 2.3.3.3	Output for Logic Circuit Answer of first student	26
Figure 2.3.4.1	Flowchart Answer of Student	26
Figure 2.3.4.2	Program generated for Flowchart Answer of Student	27
Figure 2.3.4.3	Output for Flowchart Program Execution of Student Answer	27
Figure 2.3.5.1	Flowchart Answer of Teacher	28
Figure 2.3.5.2	Flowchart Answer of first student	28
Figure 2.3.5.3	Output for Flowchart Answer of first student	29
Figure 3.1.1	Result set 1	34
Figure 3.1.2	Result set 2	35
Figure 3.1.3	Result set 3	35
Figure 3.1.4	Result for Example 1	39
Figure 3.1.5	Result for Example 2	39
Figure 3.1.6	Result for Example 3	40
Figure 3.1.7	Result for Example 4	41
Figure 3.1.8	Result for Example 1	41
Figure 3.1.9	Result for Example 2	42
Figure 3.1.10	Result for Example 1	42
Figure 3.1.11	Result for Example 2	43
Figure 3.1.12	Result for Example 3	43
Figure 3.1.13	Result for Example 4	44



Figure 3.1.14	Result for Example 1	44
Figure 3.1.15	Result for Example 2	45
Figure 3.1.16	Doubt clarification system identified headings	46
Figure 3.1.17	Identified entities	47
Figure 3.1.18	Extracted information from the system	48

## LIST OF ABBREVIATIONS

Abbreviation	Description
BFS	Breadth-First Search
DFS	Depth-First Search
JSON	JavaScript Object Notation
NLTK	Natural Language ToolKit
OBDD	Ordered Binary Decision Diagram
PDF	Portable Document Format
XML	Extensible Markup Language

## 1.0 INTRODUCTION

### 1.1 Background and literature reviews

Various techniques have been used for marking Structured-type questions. There are systems already existing to mark answers of simple questions which does not require answers in the form of sentences. Course Toolkit Learning Management System [1] is another system which has the ability to mark MCQs, and it also has the capability of grading simple questions like “Name two fruits”, provided that the answers are given to the system by the tutor. The existing systems are incapable of marking answers that compose of sentences, which the questions starts with “Describe”, “Explain”, “Justify”, ‘Make a statement’. Several researches have been carried out on this area before to figure out methods to solve this issue.

Following research done in 2003, UCLES has continued the automated short answer marking research [2] using three Machine Learning techniques named Inductive Logic Programming (ILP), Decision Tree Learning (DTL), and Bayesian Learning(BL) where they figured that ILP is capable of finding keywords while BL worked better than DTL when it came to the final results. They showed that they have achieved success at a considerable level even with the simple classifications they used which were similar to the “bag-of-words” approach which is not very precise when it comes to different ways of writing the same set of words, and that implies that although the sentence contains the required keywords, any alterations in the meaning of the sentence will not be recognized which leads to lower accuracy as the sentence meaning is not properly understood when allocating marks.

In the research done by Raheel Siddiqi[3] on the region-based approach to automated marking of short textual answers, an Open Source spell checker named Jortho has been used to perform the spell-checking, a Treebank-trained statistical parser named Stanford Parser has been used to parse the student’s answer text. Then, the part-of-speech tagged text which is one of the outputs of the Stanford Parser is chunked into noun phrases and verb groups by the Noun Phrase and Verb Group Chunker. The other output produced by the Stanford Parser is the typed dependency grammatical relation between individual words. After generating the tagged and chunked text, it is compared with the syntactic structures required as specified in the Question Answer Language (QAL). They have also used a clause negation detector to detect any negations in the student’s answer, as identifying this is important in order to mark

accurately. The answer to a question is divided into several regions in this research from which patterns are generated to compare with the student's answer. The results of the research done by Raheel Siddiqi where they have also used pattern-matching along with QAL, and providing the capability of specifying the definition of regions in an answer and the order in which the regions can appear seems to be significantly high too, although as in the previous researches it is not hundred percent accurate.

According to the study[4] Semantic similarity score for a pair of sentences is calculated using WordNet::Similarity. A method described in the above paper is adopted to the our system when calculating the semantic similarity between two sentences.

CourseMaster [5], [6] is one of the existing research systems that can mark Logic Circuits and Flowcharts, while we are not aware of any present system that marks block diagrams.

CourseMaster [5] has marked Flowcharts by converting the flowchart to BASIC programs, and testing the solution behaviour by executing it against test data while Logic circuits have been marked through simulation, and these research papers have not explained the approach taken in performing the above mentioned tasks.

There are several algorithms mentioned in literature which are used for graph matching. Ullmann's algorithm, VF<sub>2</sub> algorithm are some of the exact graph matching algorithms [7], [8]. These exact graph matching algorithms work by conserving the nodes and edges of the graphs, and they also take the neighbouring nodes into account when matching graphs. A-star algorithm [9], [10], [11], [12], [13] is one of the inexact error-tolerant graph matching algorithms mentioned in previously reported work, and the inexact algorithms work by taking the additional, deleted, and substituted nodes and edges of the graphs into account.

Binary Decision Diagrams, Ordered Binary Decision Diagrams, Simulation, Progressive Circuit Reduction, and implication-based methods have been used in former work [14], [15] related to logic verification of logic circuits.

Research work has been carried out in the context of Flowchart Recognition and Flowchart Plagiarism using Image Processing in the literature work. In the research where Flowchart Plagiarism is detected [16], they check the similarity of a flowchart against all the stored flowcharts in the database by comparing the flowchart under

analysis with the rest of the flowcharts in the database, but as before, the exact approach taken in this graph comparison is not mentioned in the paper.

Currently, there are some existing systems worldwide to give information to questions of the users, but most of the systems get answers through a web search. According to the Author's knowledge, there are no existing systems that can retrieve information according to user's questions using PDF books. One Question and Answering System is START Natural Language Question Answering System[18] which retrieves information/answers from the web, but the problem is most of the time, it gives "I don't know" as the output. In the research done by Alvaro Rodrigo, Joaquín P´erez-Iglesias on Question answering system based on Information Retrieval and Validation[19] they use Entity recognition to find the most correct answer, and they use an Acronym Checking method too. Dice algorithm has been used to compare the keywords and answers in the research done by Bolanle Ojokoh<sup>1</sup>, Peter Ayokunle<sup>2</sup> [20] on the Online Question Answering System. Then they pass the output to a modified version of the Levenshtein distance algorithm to compare characters of the strings since their system works with keywords.

## 1.2 Research gap

The existing systems do not mark short answer questions, and although several researches have been performed in this area previously, none of the systems developed through the researches are hundred percent reliable and accurate with the marking to completely replace a human, and to be used in actual marking in school term exams.

The results of the research done in 2003 by UCLES shows that they have achieved success to a considerable level even with the simple classifications they used which were similar to the "bag-of-words" approach which is not very precise when it comes to different ways of writing the same set of words, and that implies that although the sentence contains the required keywords, any alterations in the meaning of the sentence will not be recognized which leads to low accuracy as the sentence meaning is not properly understood when allocating marks. In the research which followed this research, they found that the results obtained by using ILP were not encouraging as learning rules like how a certain word or two certain words should be within a predefined distance seems to be hard, but they discovered that the identification of single keywords seems to be working well with ILP. The results of BL were more promising, and it achieved better results compared to both DTL and ILP while DTL had better

results in comparison to ILP. They also figured that using Machine Learning techniques can better help in providing comments to the students on the parts they have gone wrong in the answer, and that the usage of Decision Trees can be of more help as they are capable of suggesting more useful patterns.

The results of the research done by Raheel Siddiqi where they have also used pattern-matching along with QAL, and providing the capability of specifying the definition of regions in an answer and the order in which the regions can appear, seems to be significantly high too, although as in the previous researches it is not hundred percent accurate.

As it can be seen, most systems produced through research are not hundred percent accurate, and the existing systems do not support the grading of long-text answers. Although it may not be possible to make a hundred percent accurate system due to the difficulties and limitations, research can still be carried out to experiment various algorithms to find a solution which might help build a system with higher accuracy than that achieved so far to solve this automation issue.

This can be done by better addressing the issues like identifying the correct keywords and regions in the answer more frequently while the meaning of the answer must be checked properly to identify whether it is the exact answer required with no negation of clauses unless needed.

According to the author's best knowledge, surprisingly, there aren't many existing systems and research done to mark the three types of diagrams mentioned above and other diagrams, as also pointed out in [17]. Therefore, making an effort to research in the related areas, and building a system that supports the marking of Block diagrams, Logic Circuits, and Flowcharts is bound to be a task worthwhile.

Although there have been research work done in the area of flowcharts, there are no researches directed towards the marking of flowchart answers of students except CourseMaster research system. As described previously, the recent research work on flowchart plagiarism has the capability of identifying symbols at a considerably high level with the use of Image Processing, but they have not taken the logic of flowcharts into account when checking the similarity between two flowchart graphs, making them only able to identify structural changes in the graphs while detecting any plagiarism through orientation in addition to the direct structural differences.

When it comes to the CourseMaster system, one of the major concerns that can be seen is that the exercise developer has to make the marking scheme of the exercise as a Java class which implies that all the teachers using the system should have sufficient programming knowledge which may not be the case in certain scenarios, especially when it comes to schools. Therefore, having a much more user-friendly approach in defining the mark allocation details would attract more users to the system.

Apart from marking structured questions and diagram type question modules, this system has another module for clarifying student's doubts. There are some systems to clarify student's doubts. One of the systems is START Natural Language Question Answering System[18]. This system shows "I don't know the answer" for most of the questions. Answers of this system can be too light for student knowledge, since it gives some small answers from the web.

Stack overflow is a famous site to add questions and answers. But students get answers sometimes after some days. Some answers are not accurate enough considering the students age. Thus can be disadvantageous

Table 1.2.1: Comparison of Existing Systems vs our System for Question Marking

<b>Existi ng Syste ms</b>	<b>Mar k MC Q</b>	<b>Mark Structur ed type Questio ns.</b>	<b>Mark Block Diagra ms</b>	<b>Mark Logic Gate Diagra ms</b>	<b>Mark Flowch arts</b>	<b>Calculat e total number of correct answers and provide results</b>	<b>Give a detailed feedbac k to the student</b>
Our Syste m	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Auto Multip le Choice	Yes	No.	No	No	No	Yes	No
Course Toolki t Learni ng Manag ement Syste m	Yes	No. Only simple question s.	No	No	No	Yes	No
SA grader	No	Yes	No	No	No	Yes	Yes
Course Master	Yes	Yes	No	Yes	Yes	Yes	Yes

Table 1.2.2: Comparison of Existing Systems vs our System for Doubt Clarification

<b>Existing systems</b>	<b>Information is relevant to the question</b>	<b>Summarize the information as per student grade</b>	<b>Respond quickly</b>
Our system	Yes	Yes	Yes
START Natural Language Question Answering System	Yes. Most of the time	No	Yes
Stack overflow	Sometimes	No	Yes

There are so many existing methods, research papers and codebases for question classification, that use different machine learning techniques to achieve the targets.

In order to classify the question, certain attributes about the question that is provided by the student must be identified.

The attributes included in the classified question are,

1. Subject related to the question
2. Answer weight
3. Type of the question (Classified according to the Bloom's Taxonomy)
4. Main keywords
5. Synonyms for main keywords
6. Other special words or requirements

Doubt Clarification module needs the attributes mentioned above to obtain the accurate answer for the questions provided by the student.



### 1.3 Research Problem

Marking question papers have always remained a difficult task owing to the workload, time consumption and effort that has to be undertaken by examiners when the process is carried out manually. There are several concerns related to the marking of question papers, as all answers should be marked in a fair manner. Marking huge numbers of question papers within a short time period is a challenging and inconvenient task for teachers. Most of the term test papers in schools worldwide are currently marked by hand by the examiners as marking of papers always require human assistance due to the multiple ways in which an answer could be written to express the same meaning, and the subjectiveness of some answers.

There are times when teachers tend to mark papers quickly as they have very less time to spend on marking examination paper answers, as they might have other responsibilities. When we consider term tests in the Sri Lankan school system, no allowances are given to the teachers for marking term test answer scripts, so there are chances that the teachers may not have enough motivation to spend enough time on paper marking in order to mark the papers effectively. Another factor to be considered is that teachers get tired when they are engaging in the process of marking the papers for a longer period of time, and as a consequence, papers of some students will be marked relatively better than the papers of some other students. As teachers can make errors either intentionally or unintentionally during marking owing to the above stated reasons, the number of requests made by students for remarking of papers will also increase.

Another concern is that paper marking requires a lot of resources, and there should be a sufficient amount of teachers to mark the exam papers. Low staff/student ratios also often mean that teachers often have great difficulty in providing students with quality feedback on performances of students at the examinations.

There are subjects related to different disciplines such as Literature, Psychology, Law and Rights where the marks gained by students may heavily depend on the perception of the examiner. Mental and other external factors which affect an examiner while marking the answer scripts may have an impact on his perception and decisions taken at the time. As a result, some students may get treated unfairly.

When we consider national level examinations such as G.C.E Ordinary Level examination and G.C.E. Advanced Level examination, the government has to spend a lot of money for the process of marking the answer scripts related to each subject. As the paper marking process is carried out at schools, the schools have to be closed until the paper marking process is finished. It would be a great benefit if the above mentioned national expenditure can be minimized, thus, making a positive impact on the country's economy.

Another issue is that at times when examiners are carried out by foreign universities, there may be slight delay when getting the results, as papers should be sent safely to that particular foreign university for marking, and extra care and effort should also be taken for this process.

The main aspect of this question classification module is to generate the detailed question query.

While studying, a student can get doubts, and students tend to clarify their doubts by meeting and questioning teachers, but this process requires a considerable amount of time. Another way of clarifying doubts is by searching for information in the web, but the answers provided by the web are not summarized as per the age and knowledge of the student most of the time, while the answer provided may also be irrelevant at times.

#### **1.4 Research objective**

The main objective of our system is to automate the question paper marking process. Using the proposed system, it must be possible to mark a set of question papers within a shorter time period and give fair marks to all students with the minimum number of marking errors possible. Manual question paper marking process puts pressure on teachers, as papers need to be marked correctly, but humans can make errors while marking the papers. Marking bundle of question papers within a short time period is a hard and time consuming task. Paper marking also requires the continuous training of teachers, so that they allocate marks after proper consideration. Manual paper marking also arises the need to have enough resources. Marking question papers from any country using our web based system makes the process of marking question papers easy. Marking foreign universities papers is time consuming because we have to send all student's answer sheets safely to that foreign university, and this process requires some

more extra days to marks the papers. Apart from marking question papers, our system has another feature that provides the capability to retrieve information for any syllabus-related question. This feature is mainly for the learning purpose of the students. A student can enter any subject-related question to a system. The system will provide the question related information according to student's age, grade, and the knowledge. This feature helps students in practicing and improving their skills in answering questions properly before the exam, and it will also assist students in facing exams with more confidence.

### **Specific Objectives :**

In the Structured question marking module,

- The system should be able to check the meaning of the answer provided by the student, and the answer provided by the teacher.
- The system should compare the student answer with the teacher's answer.
- Marks must be allocated according to correct points, and the missing keywords and points must be given as feedback.

In the diagram marking module,

- The users should be able to enter their answer diagrams to the system.
- Teachers and students should only be able to add valid and system-accepted answers to the system.
- The answer diagram of each student should be compared with the teacher's to recognize the similarities and differences.
- Logic of the diagram should be analyzed if necessary (applies to Logic Circuits and Flowcharts).
- Each student answer must be allocated with marks.
- Each student answer must have feedback generated for it at the end of the process.

In the doubt clarification

- Students and teachers should be able to enter their questions to the provided interface after selecting related subject.
- System should have a function to filter keywords from the provided question.
- There should be a function to get other relevant key points.
- There should be a function to get synonyms for retrieved keywords.
- There should be a function to get antonyms for retrieved keywords.
- System should have a function to classify the question.

- Above generated data set should pass to the doubt clarification module.
- Teachers should be able to add reference books to the system by defining subject and the grade for reference book.
- Teacher could be able to view the added reference books and to be able to add important chapter details like chapter heading starting page and ending page to the system.
- System should have way to convert pdf documents in to XML and that process should be very fast.
- There should be a way to find out related references books.
- Then system should be able to find matching headings from the previously added references.
- There should be a way to compare extracted headings with the key points of the question.
- After retrieving information system should identify the entities of the extracted information, if it is a entity type question. Otherwise, it should display all information together.
- When the system finding related information system retrieve information from many reference books for same topic therefore information can be duplicated to avoid that system should use summarization methods.

## 2.0 METHODOLOGY

### 2.1 Methodology

PHP and Javascript are the languages we used in developing the Frontend of our application while Flask and Spring Boot were used in making the Python and Java Web Application Programming Interfaces(API). MySQL is the database we used in holding all the application-related data and records. Neo4j database was used for the construction and easy traversal of graphs in graph matching of the diagram marking module.

Py2neo and Pymysql are the Python libraries used to connect to Neo4j and MySQL databases respectively.

Structured question marking mainly focuses on the comparison between the teacher's answer and the student's answer by calculating the semantic sentence similarity among the answer sentences using semantic relation of word senses across different synsets using WordNet for different parts of speech in relation to the words. Firstly, the student's answer and the teacher's answer are tokenized. Stanford core NLP provides the ability to split the text into sentences, and separate the words into parts of speech using Stanford CoreNLP Parts-Of-Speech (POS) tagger. Nouns, Verbs, Adjectives, Adverbs, Subject, Object and the relationships between them can be identified using this POS tagger. The system does not consider all verbs which will be lematized into verb 'be' and verbs such as "have","do" when calculating Verb Similarity.

Stanford Named Entity Recognizer is used for identifying the named entities like Person, Organization, Location, Currency, Date and Time. Subjects and Objects can be identified using Stanford CoreNLP Dependency Parse Annotator. After tokenizing both student's answer and teacher's answer, synonyms sets are generated for all the nouns and verbs in the both answers. Then, the Semantic Similarity score is calculated by checking the words with each word in the related synonyms set separately until a maximum score is obtained. We used WordNet, which is a large English lexical database to obtain this relation between words. Semantic Similarity score returned at the end of the process is a value between 0 and 1. The score allocated for a particular answer changes depending on this returned value. The keywords must be provided by the teacher when creating a structured typed question.

The following equation is used to calculate the semantic similarity score of the teacher's answer and the student's answer.

$$score = Average \left( \sum_{i=1}^n NounScore + \sum_{i=1}^n VerbScore \right)$$

Figure 2.1.1: Semantic Similarity Score Calculation Equation

Cosine Similarity is used to check whether the students have included the keywords provided by the teacher in the question. In the process of checking cosine similarity, the system will check whether the exact word is there in the student's answer.

The following Cosine Similarity values are calculated for both the teacher's and the student's answer:

- Word Similarity
- Noun Similarity
- Verb Similarity
- Adjective Similarity

Cosine Similarity Equation:

$$CosineSimilarity = \frac{\sum_{i=1}^n FV_{S,i} * FV_{T,i}}{\sqrt{\sum_{i=1}^n (FV_{S,i})^2} + \sqrt{\sum_{i=1}^n (FV_{T,i})^2}} \quad (1)$$

Figure 2.1.2: Cosine Similarity Equation

This equation represents frequency vectors of the source sentence and the target sentence. (Target sentence is the teacher's answer and the Source sentence is the student's answer)

The number of unique words in the student's answer which overlaps with the unique words in teacher's answer is also evaluated as a ratio. The Length difference between the answers is also calculated to get the final mark for a particular answer. Finally, feedback is generated, providing the key points that are missing in the student's answer.

With the intention of enabling teachers and students to enter their diagram answers to the system, the proposed system provides a drag-and-drop canvas made using the GoJS diagramming pack under evaluation license.

MindFusion is another diagramming pack we checked when developing the canvas, and we chose GoJS over MindFusion due to the several reasons mentioned below.

- GoJS is more user-friendly than MindFusion when it comes to shifting around shapes.
- GoJS supplies more shapes by default than MindFusion where we must customize most of the Logic Circuit symbols.
- The JSON object produced by the MindFusion diagramming pack which contains all the necessary information is very large and contains a lot of information that we do not want, increasing the size of the object unnecessarily when compared to GoJS.

In order to ensure that all the diagram answers entered by the teacher and the student are valid and system-accepted to support the marking, they are validated in the client-side.

The resulting diagram is saved to the MySQL database in JSON string format, which is used later by the diagram marking API to read and construct the graph in Neo4j database.

The answer diagrams are marked by constructing the graphs in the Neo4j database, and performing graph matching. When marking student answers for a particular question, the answer graph of the teacher is created only once in Neo4j, while each student's answer is constructed and deleted after marking. While Depth-First Search (DFS) is used for the marking of Block diagrams and Flowcharts, Breadth-First Search (BFS) is used in marking Logic Circuits. We have taken a novel approach in graph matching using DFS and BFS.

The exact graph matching algorithms mentioned in literature are less suitable to handle the marking of student answers as they conserve the nodes and edges of the diagram when matching the graphs because the students can make mistakes at any position of the graph either by inserting, deleting, or substituting the nodes. The inexact graph matching algorithm A-star works by generating a search tree for all the possible combinations, while considering the graph edit operations like additions, deletions, and substitutions, and this is computationally expensive. Both these types of graph matching algorithms match graphs by matching a single node against all the unmatched nodes, and this also introduces a lot of complexity. As we know the exact location to expect a node in student answers, if the start node can be identified, it is possible to remove this node mapping complexity.

As mentioned before, DFS was used in graph traversal and matching when marking Block diagrams. As a system constraint, both the teacher and the student has to add a

‘start’ block and ‘end’ block to their answer, and this ‘start’ block is used by the system in identifying the point where each answer graph traversal must start, thus, reducing the computational complexity otherwise required in identifying the start point of the graph, by mapping one node of teacher’s graph against all the nodes in the student’s graph until a matching node is found. Once the start node is identified, as we know that a correct student answer should have all the blocks connected as it is in the teacher’s answer with the same number of blocks and connections, we can expect the matching blocks to be in the same position as the teacher’s, but since the students can always make mistakes, there is a possibility for the student’s answer to have additional, deleted, and substituted nodes. Therefore, after discovering the start nodes, in our approach, both the teacher and student graphs are traversed simultaneously using DFS by maintaining a stack where pushing and popping are done at the same time to both the stacks, and going down, analysing the depth.

After start nodes are added to the respective stacks, in a loop, the current teacher and student nodes are popped, which in the very first iteration is the start block, and child nodes of the current teacher node are analysed to find a match for each child node in teacher graph from the set of child nodes of the current student node. Node matching is done by checking text similarity using WordNet. It is done by first tokenizing the set of the words in the phrase, and part-of-speech tagging the words to identify Nouns, Verb, Adjectives, and Adverbs which are recognized and used by WordNet when retrieving synonym sets for each word. Afterwards, similarity value of the most similar word according to path similarity in WordNet hierarchy is taken for each word in the phrase to form the final score between zero and one after averaging the value. This score is used when matching the nodes. After studying several text matches, the threshold value to be taken for the identification of a correct match was decided to be kept at 0.55 and above. When a matching child node is found, they are added to the respective stacks to continue with the graph matching and further traversal of the graphs.

Semantic similarity was decided to be used over the three String similarity metrics: Jaccard, Levenshtein, and Sequence Matcher tried out as it performed better than them owing to its capability of analyzing the meaning which is not done by the string similarity metrics which take decisions based on the words and characters in a string phrase. Jaccard performed the lowest out of these three, while Sequence Matcher was the best along the string similarity metrics.

In an attempt to identify additional, deleted, and substituted child nodes of the current node, we make a comparison of the amount of child nodes of the current teacher node



against the amount of child nodes of the current student node during each iteration. If teacher's answer has more child nodes, it means the student has not drawn some of the required nodes and they are deleted. If the student's answer has more child nodes, it gives the indication that student has drawn additional nodes. The nodes that are found not to be matching while the amount of child nodes in both teacher and the student answers are the same are taken to be substituted. The nodes that are not found to be identical even when student has drawn additional nodes or deleted some nodes are also classified to be substituted. The additional, deleted, and substituted nodes are noted at each level, and once the whole graph has been matched, paths down these nodes are traversed to identify other incorrect nodes.

With the aim of enabling more graph matching without discarding the whole path when a substituted node is found, if there is only one substituted node, it is added anyway to the stack. If there are more than one substituted nodes, the immediate next level first child node of each teacher's child node is checked against student's immediate next level child nodes to recognize the path by identifying a matching node in the next immediate level. Only one child node of teacher's is checked against the child nodes of students because if the student has drawn the next level correctly checking one child node would be enough to find the correct path, and this was done to reduce the computational complexity involved which can also affect the performance of the system. While all match found nodes are added to the stack, any remaining unmatched node remaining after these operations have been applied are noted for later identification of other incorrect nodes as mentioned above. While traversing the graph, incorrect connections are also identified and noted by our system.

If two logic circuits respond to an arbitrary input pattern with the same output, they can be taken as functionally equal [14]. Hence, this makes it possible to use simulation to verify the logic of logic circuits.

Progressive Reduction of Circuits using implication-based methods can result in False Negatives as pointed out in [14]. Along with the circuit reduction, simulation is also used to check the functional equality. Implication-based methods uses boolean mathematics for identifying equivalent nodes and this can introduce complexities in the development of an application. Mostly these two methods: Progressive Reduction of Circuits and Implication-based methods have been used in literature when the circuit has a large number of inputs like 50 to analyse. Owing to the simplicity of logic circuit questions, these methods need not be used when we can guarantee a correct output with simulation.

A similar approach is taken to mark Logic Circuits using BFS and a queue by identifying the correct, additional, deleted, and substituted nodes. The inputs are identified and added to the teacher and student answer queues in the same order. A gate is taken as matched and added to the queue only if the number of times it has been visited is equal to the number of inputs to it. In order to ensure that the next gate connected to the current gate is exactly the gate as in the teacher's answer, other than the exact text match done to identify the symbol, we have enabled more tests like checking whether the number of parents of the current node with other children and without any other children are the same. Additionally, to improve performance by reducing the complexity and looping involved, all the childless parents of the current node are removed from the queue, so that they are not analysed again.

Instead of using Ordered Binary Decision Diagrams (OBDD), simulation results for each binary combination is stored in MySQL, as the usage of OBDD involved a huge amount of reads and writes to Neo4j.

Simulation of a Logic Circuit is also done by using BFS and a queue. A binary combination list is generated by using itertools library in Python according to the number of inputs to the circuit. All inputs are processed in the same order for each binary combination, and also when simulating both teacher and student answers. For each binary combination, simulation is run once to generate the output for that particular combination. When running the simulation, the current inputs to a gate are stored in the respective node in Neo4j. The gate is only added to the queue for further graph traversal, only when the number of stored inputs at the gate is equal to the number of input connections to the gate. When analysing a gate, the stored binary inputs are evaluated as per the gate to decide the output of the gate, and store it as an input in its child nodes.

Flowchart answers are marked by first converting the answer and generating a Python program file which is used to test the program logic by providing the inputs given by the teacher to observe whether the desired output is given by the program. Inputs are fed to the program through command-line arguments passed to the execution command. Output is taken by redirecting the standard output to a variable for the period of program execution. Marks are allocated for steps using DFS and a stack, and although a similar approach is taken in detection of correct, additional, deleted, and substituted nodes, unlike in Block diagrams, taking the flowchart rules into consideration, the symbols will not be expected to be in a fixed position similar to the teacher's answer. For instance, if teacher has taken an input before a decision, it is expected for the student to take an

input before a decision, but the location of the input need not be fixed as long as it is before the decision, as other operations like initialization and assignment can happen in between, in any position, if the teacher has them before the input. If the converted program does not return the expected output, marks will be given only for the correct input and output steps.

The type of the Decision node is identified by scrutinizing the graph structure. If there are no loops, it would be a conditional statement either If or If Else. If a loop is found and one of the child nodes have been visited, it is a Do-While loop, and otherwise, if one of the children are not visited, it is a While loop. Indenting whenever a Decision node is reached. Unindenting is done whenever the common node for both the paths are reached when it comes to If/If Else structures while unindenting is done in While and Do-While whenever the Decision node is reached for the second time.

In Doubt Clarification feature, we divided the component into four main parts. First one is Question Classification, which is also divided into four main subcategories. First module extracts keywords from the question provided by the student. This keyword extraction function is developed using Python: RAKE package. Afterwards, the keywords extracted are processed using the function that finds synonyms for each of the extracted keywords, and this particular function returns synonyms for a given word, and it is developed using Python and the WordNet lexical database. The third method is the Question Classification function, which is developed using a Convolutional Neural Network, and the results are categorized according to the Bloom's Taxonomy. The final module is responsible for sending the question-related subject, weight and the scope to the next part of the process with the generated question classification query attributes.

The second main component converts the PDF reference books into XML format. In order to convert PDF references into XML, we use an already existing site called pdfx v.19[21]. One of the problems we identified is that we can convert only 100 pages at a time using the available PDF Converters. Therefore, we decided to select the important chapters of the PDF and create a new PDF document using the python library: PyPDF2, and convert that newly created PDF to XML format later on. The next step in the process of related information retrieval is the finding the relevant information. We use the keywords extracted previously to gather information, and the topic related to the question. When gathering information, firstly, the heading of the XML document is extracted, and match those heading with extracted important keywords and the topics. To get a similarity of the text, we use a small mathematical calculation after tokenizing and removing stop words of the sentences. If A and B are the tokenized word sets, the

intercept of these word sets is taken and that value is divided from the union of A and B. It gives a number (P) between 0 and 1. The threshold was decided to be kept at 0.5, which means if P value is greater than 0.5, the two sentences are equal, or else the sentences are not equal.[1]

$$P = A \cap B / A \cup B$$

After identifying the topics we extract information under that topic, and classify the information according to question types. To show correct information to users, we use Entity Recognition using Python NLTK library if the answer is of the entity type. If the question is a descriptive type, then system shows an extracted paragraph. Finally, the information retrieved from the PDF documents is passed to the summarizing process since teachers can add many reference books on the same topic. There are two main approaches to summarization. For our proposed system, we used the Extractive method since it carries out summarization by removing the duplicate sentences without changing the sentences in the paragraph, and it shows the answer clearly to the user, and this Extractive Summarization method was developed using the Gensim Summarization module in Python.

## **2.2 Commercialization aspects of the product**

As Automated Structured, Diagram Question Marking and Doubt Clarification System is a web based solution, any user, both teachers and students from anywhere can access this system via the internet easily. This will be extremely beneficial for teachers and students as they will be able to carry out their online exams and get the marks awarded for the answers without much hassle.

As there are no existing systems that has the capability of marking the structured and the three diagram-type questions mentioned in this document, along with the doubt clarification feature to support students, while being immensely valuable, this system is bound to be easily commercializable.

It is also possible to earn a good revenue through this system as we can allow the users to make business accounts and charge users for each API call they make to the corresponding backends.

## **2.3 Testing and Implementation**

The main users of our web-based solution are teachers and students. The system is also accessible for developers who have to carry out maintenance of the product either by adding features or fixing bugs found in the system.

The industry standards and conventions have been maintained when coding, keeping the best practices like code reusability, code readability, and code modularity in mind, further supporting in the easier future maintenance of the system. Source controlling is also being done by maintaining all the source code changes in the respective GitHub repositories made for the frontend, and the backend Web APIs, enabling easy track of changes and history viewing possible to easily identify and fix issues by analysing the modifications done to the code through each commit.

In the diagram question marking module, as an attempt to handle the cases where marking of a question may stop abruptly due to a failure of the system or a bug in the code, we have ensured that the remarking picks up from the place where the marking had stopped, thereby, marking only the unmarked papers without marking the marked answers twice.

We have also made it certain that in the situations where a student may submit a flowchart answer that ends up in infinite loops, the looping is broken by maintaining a count of the loop, and this maximum count was decided to be taken as 25 after analysing the past paper questions which rarely had at least 20 loops. Although we considered leaving the looping for four minutes initially, in order to improve the performance of the system, we decided to go with the loop counting approach.

The diagram marking module was tested thoroughly by testing the system against all the possible scenarios in each type of diagram marking, and fixing the bugs that came up by debugging the application either through trace or print messages, or through Step In, Step Over, and Step Out. The testing was done by analysing the marks returned with the correct number of text, symbols, or steps identified as per the diagram types, along with feedback and the number of additional, deleted, and substituted nodes identified. Some of the tests carried out on the diagram marking module are outlined below.

### 2.3.1. Block diagram implementation testing

Teacher's Answer Diagram:

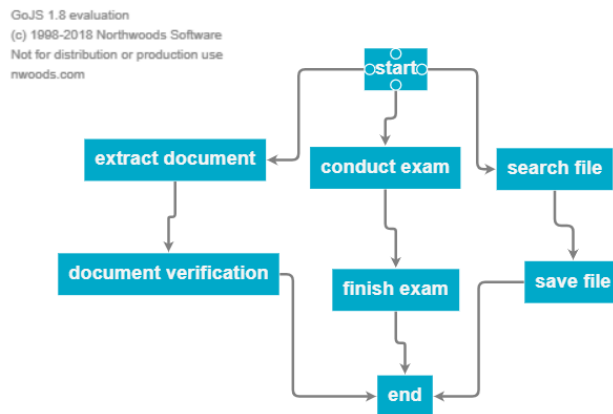


Figure 2.3.1.1: Block Answer Diagram of Teacher

Allocated Text Mark Per Block: 1 (Total will add up to 6)

Allocated Sequence Mark: 2

Scenario 01(Answer of first student):

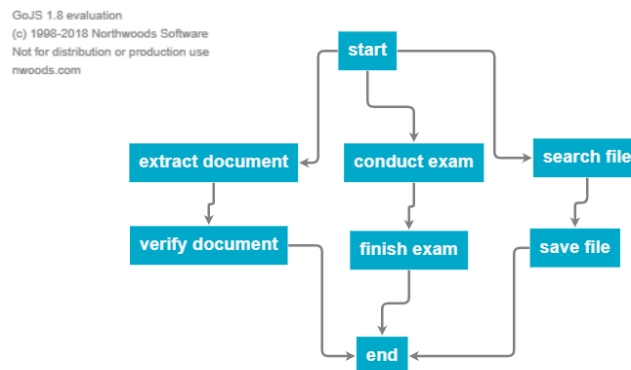


Figure 2.3.1.2: Block Answer Diagram of first student

Output for answer of first student:

```

Scored Text Mark: 6.0
Scored Sequence Mark: 2.0
Number of Additional Blocks Found: 0
Number of Deleted Blocks: 0
Number of other Incorrect Blocks and Connections Found: 0
Feedback: The block:search file connected to block:start is correct. The block:condu
    
```

Figure 2.3.1.3: Output for block diagram answer of first student

We can see that the system has awarded full marks and has not detected any type of incorrect node as expected, although the text ‘verify document’ has been written differently when compared with the teacher’s answer where the same piece of text with the same meaning is written as ‘document verification’.

Scenario 02(Answer of second student):

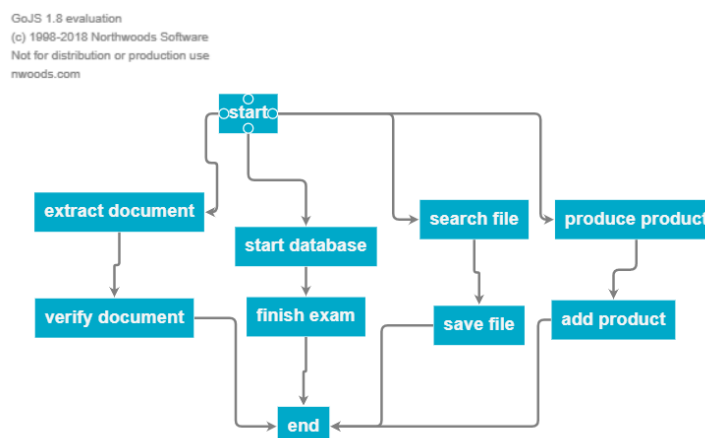


Figure 2.3.1.4: Block Answer Diagram of second student

Output for answer of second student:

```

Scored Text Mark: 5.0
Scored Sequence Mark: 1.44
Number of Additional Blocks Found: 2
Number of Deleted Blocks: 0
Number of other Incorrect Blocks and Connections Found: 0
Feedback: The block:search file connected to block:start is correct. The block:extract document connect
    
```

Figure 2.3.1.5: Output for block diagram answer of second student

We can notice that the two additional nodes in the Figure 2.3.1.4 has been identified correctly as additional by the system, and the sequence marks have been deducted in a suitable manner for this additional. It is also visible that the system has identified the block ‘start database’ which has substituted ‘conduct exam’ as a substituted node because a mark has been deducted from the scored text mark for this substitution.

### 2.3.2. Logic circuit graph matching implementation testing

Teacher’s Answer Diagram:

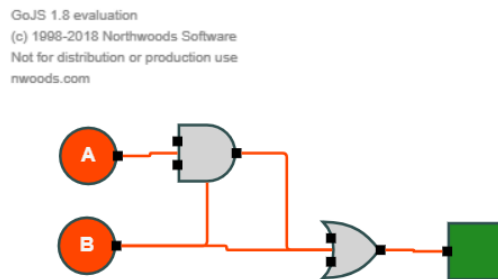


Figure 2.3.2.1: Logic Circuit Answer of Teacher

Allocated Symbol Mark Per Symbol: 1 (Total will add up to 5)

Allocated Sequence Mark: 2

Scenario 01(Answer of first student):



GoJS 1.8 evaluation  
(c) 1998-2018 Northwoods Software  
Not for distribution or production use  
nwoods.com

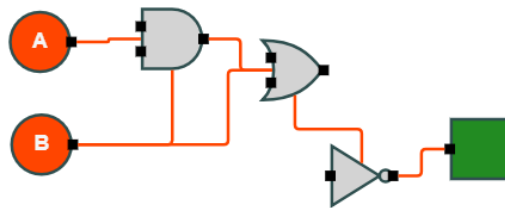


Figure 2.3.2.2: Logic Circuit Answer of first student

Output for answer of first student:

```

Scored Symbol Mark: 4.0
Scored Sequence Mark: 1.7333333333333334
Number of Additional Nodes Found: 0
Number of Deleted Nodes: 0
Number of other Incorrect Nodes: 0
Feedback: The gate: AND and its input connections are correct. The gate: OR an
    
```

Figure 2.3.2.3: Output for Logic Circuit Answer of first student

We can see from the output in Figure 2.3.2.3, the system has identified one substituted node which is the NOT gate in the location of the output, and therefore, one mark has been deducted from symbol marks, and marks have been reduced as required from the sequence marks as well. This node is detected as substituted rather than an extra because additional nodes are detected when located in a breadth-wise position rather than horizontal as an additional child of another node.

### 2.3.3. Logic circuit simulation implementation testing

Teacher's Answer Diagram:

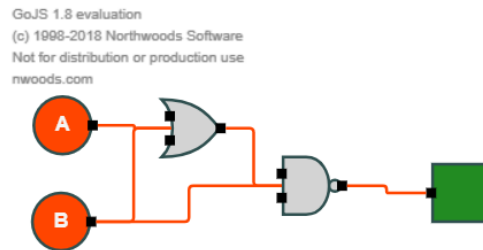


Figure 2.3.3.1: Logic Circuit Answer of Teacher

Allocated Symbol Mark Per Symbol: 1 (Total will add up to 5)

Allocated Sequence Mark: 2

Scenario 01(Answer of first student):

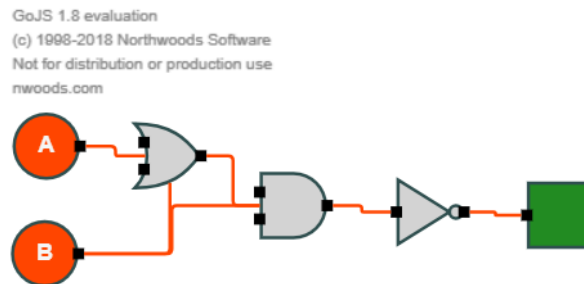


Figure 2.3.3.2: Logic Circuit Answer of first student

Output for answer of first student:

```

Scored Symbol Mark: 5.0
Scored Sequence Mark: 2.0
Number of Additional Nodes Found: 0
Number of Deleted Nodes: 0
Number of other Incorrect Nodes: 0
Feedback: The gate: OR and its input connections are corr
    
```

Figure 2.3.3.3: Output for Logic Circuit Answer of first student

It can be seen that the combination of an AND and NOT gate has been identified as an alternative for the NAND gate by the system, as the circuit returns the same set of outputs for all the binary combinations. Hence, full marks have been given for both the symbol and sequence marks.

#### 2.3.4. Flowchart-to-Python-Program-Conversion and Program Execution implementation testing

Student's Answer Diagram:

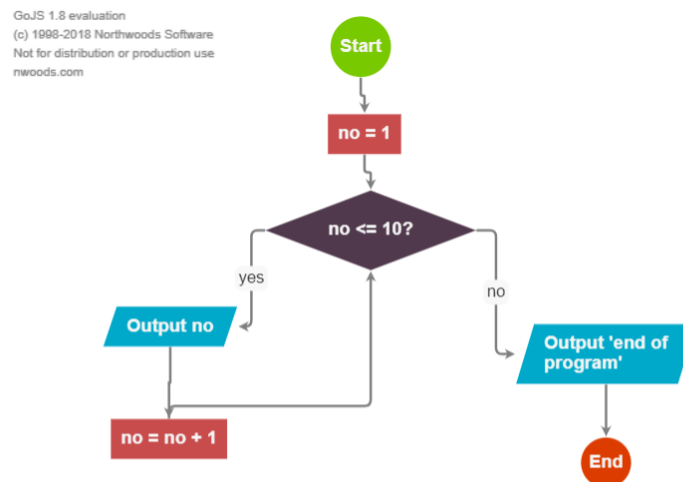
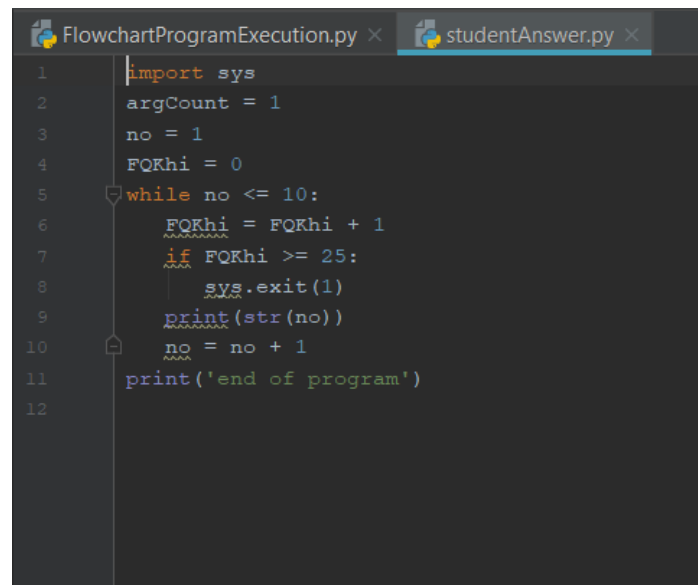


Figure 2.3.4.1: Flowchart Answer of Student

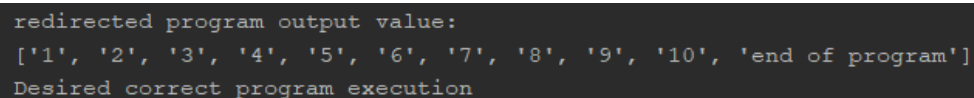
## Generated Python Program Code:



```
1 import sys
2 argCount = 1
3 no = 1
4 FQRhi = 0
5 while no <= 10:
6     FQRhi = FQRhi + 1
7     if FQRhi >= 25:
8         sys.exit(1)
9     print(str(no))
10    no = no + 1
11    print('end of program')
12
```

Figure 2.3.4.2: Program generated for Flowchart Answer of Student

## Program Execution Output:



```
redirected program output value:
['1', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'end of program']
Desired correct program execution
```

Figure 2.3.4.3: Output for Flowchart Program Execution of Student Answer

It is evident that the system has converted the flowchart answer to a Python program correctly while extra lines have been added by the system to handle imports, input to be fed to the program by maintaining the command-line argument number count at the moment, and to break infinite loops that can occur in incorrect student answers.

### 2.3.5. Flowchart graph matching implementation testing

Teacher's Answer Diagram:

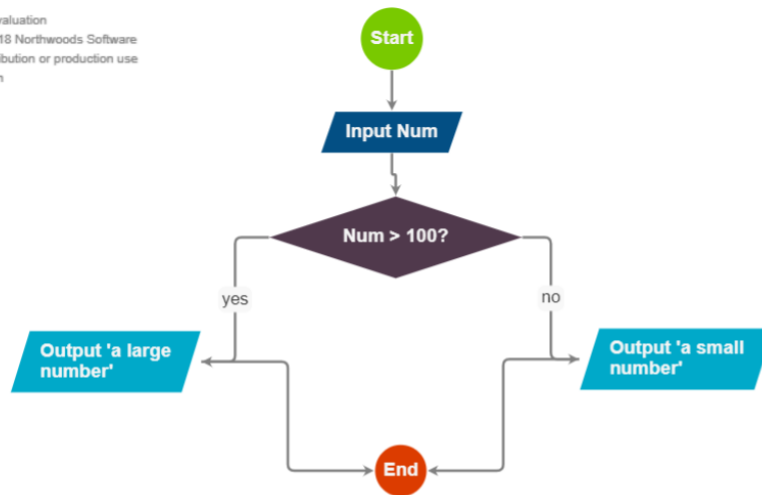


Figure 2.3.5.1: Flowchart Answer of Teacher

Allocated Step Mark: 4 (1 for each step except Start and End)

Allocated Sequence Mark: 1

Scenario 01 (Answer of first student):

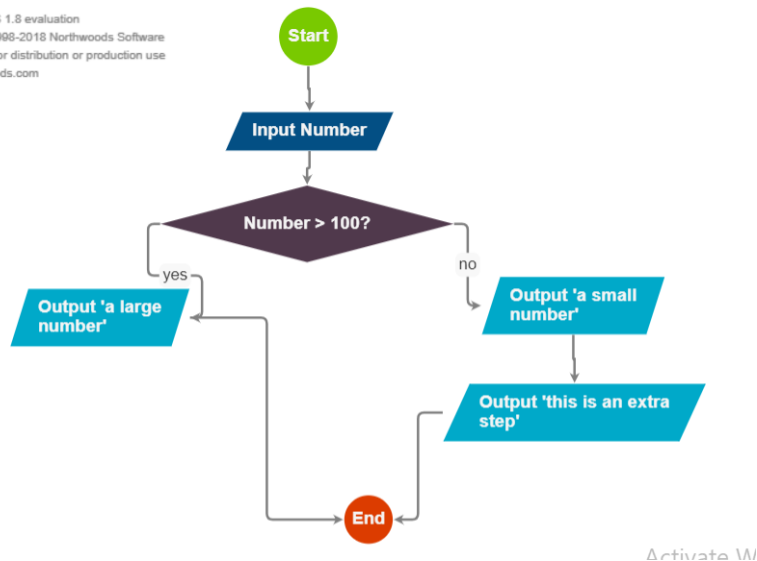


Figure 2.3.5.2: Flowchart Answer of first student

Output for answer of first student:

```
Scored Step Mark: 4.0
Scored Sequence Mark: 0.86
Number of Additional Nodes Found: 1
Number of Deleted Nodes: 0
Feedback: There are about 1 additional Output(s/es) before End node. Please ref
```

Figure 2.3.5.3: Output for Flowchart Answer of first student

It is apparent that the system has recognized the additional output node in the student answer if we view the above Figure 2.3.5.3 against the teacher and student answers, and it can also be seen that the system has deducted marks from the sequence mark appropriately.

### 2.3.6. Structured question marking implementation and testing

In Structured question marking ,the answers of the students and teacher's answers (answers in the marking scheme) are stored in a database. By checking the paper ID, question ID and student ID, the answer of the student with the corresponding answer in the marking scheme is compared. First, the two answers are compared using the cosine similarity.

- **Cosine Similarity**

In here, a list is maintained of the student's answer as well as for teacher's answer, which contains all the words in the two answers. The same word may be repeated in these lists. Then another list containing only unique words of a sentence is maintained for each answer. Also, within the system we are maintaining a list of words which are recognized as stop words.

First, the words in each answer (student's and teacher's) are extracted using a regex. Then, each word is checked to see whether the word is a stop word. If the word is not a stop word that word will be added to the list containing all the words of that answer. Then the word is added to the list of unique words only if that word was not already in the unique word list of that sentence.

After the word lists for each of the answers are finalized, the word frequency value which is required to evaluate cosine similarity as given in the above mentioned cosine similarity ratio.

- **Semantic Similarity**

Semantic Similarity was calculated based on the WordNet::Similarity. Two lists are maintained for each answer (student's answer and teacher's answer). One list contains all the Nouns in each answer, while the other list contains all the Verbs in the sentence. Then Wu Palmer semantic similarity score is calculated for each Noun of the Student's answer with each Noun of teacher's answer. For a given Noun, the highest semantic similarity value obtained by comparing with Nouns in the other answer is taken as the noun similarity score between the sentences. Similar method is followed regarding verbs too. Then the sum of highest semantic similarity value taken for each noun in the student's answer is calculated and the same method is followed for each verb in the student's answer. Then the two sums are added together and divided by the total number of nouns and verbs in student's answer to get the final answer. Stanford CoreNLP POS tagger was used to obtain nouns and verbs from the sentences in the answers.

- **Word Overlap Ratio**

In this approach, all the unique words in student's answer except stop words are put into a list. Similarly, all the unique words without stop words in teacher's answer are also put into a separate list. Then the number of common words in both lists are calculated. Then that number is divided by the number of words in the unique word list of teacher's answer (without stop words).

- **Length Difference Ratio**

In this approach, the number of words in student's answer and teacher's answer is calculated separately. Regex is used to split the answers into words. Then, the ratio is calculated using the following method, Length difference ratio =  $(\text{no. of words in student's answer} - \text{no. of words in teacher's answer}) / \text{no. of words in teacher's answer}$

A database containing student's answers and teacher's answers was maintained. Student answers were matched with the relevant marking scheme answer using question id and paper id. Then the cosine similarity value, semantic similarity value, word overlap ratio and length difference value between student's answer and the corresponding teacher's

answer was calculated. From those values the scores which are generated for correct answers and wrong answers were observed.

### **2.3.7 Question classification feature**

In this module student can enter their question to the system. The entered question is going through a few processors. Finally, a question related data query will pass to the Doubt Clarification module.

- **Keyword Extraction**

In question classification process, keyword extraction is the first step. When a student enters the question that question will pass to the keyword extraction micro service. This micro service was developed using python RAKE module and flask. Question will pass to the keyword extraction function and it will return the keywords and special points for the question.

- **Synonyms Finding**

After extracting the keywords, retrieved keywords will be forwarded to the synonyms finding micro service. Then it will check if there are relevant matches in the WordNet.

- **Antonyms Finding**

After extracting the synonyms, retrieve keywords from the keyword extraction micro service, it will be forwarded to the antonyms finding micro service. Then it will check if the relevant matches are there in WordNet.

- **Question Classification**

This is the final process of the main question classification module. By passing question entered this function will trigger and retrieve the question type accordingly.

### **2.3.8 Doubt clarification feature**

In the doubt clarification feature, information is retrieved from the PDF reference. Teachers can add reference books to the system by defining important chapter details. Then the details of the reference will be saved into the database and the PDF documents will be saved into the folder. Students and the teachers also have separate logins and we save details of those users into our MySQL database.



- **Read the reference books**

Firstly, we tried to read PDF references using font styles and sizes but when we tested it we recognized that authors don't follow the standard fonts. Therefore reading PDF using python PDF libraries is not successful. Therefore we implemented a new function to convert PDF documents into XML. To implement this we used an online site call pdfx v1.9. This online site converts PDF documents into XML. Reading content using XML tags easy than the reading PDF document. One of the problems is that converting a large number of pages at once takes a lot of time. Therefore we decided to convert important chapters of the PDF document into XML. Then the processing time gets reduced.

- **Matching headings with keywords using dice algorithm**

Read the heading of the XML document using XML tags. First, we implemented to read and compare the function for three heading tags in the xml document. Those heading tags are <h1>, <h2> and <h3>. But when we tested it we identified that the system missed most of the headings, since those headings are not under the h tags. Then we identified two other different tags called <outsider> and <region> which is mostly used in documents. Then the system compares retrieved headings with the extracted key points using dice algorithm. If the similarity is greater than 0.5 then the system takes that heading as a related topic.

- **Extract the information under the correct heading**

If the heading is matched with any of the keywords, the system saves the id of that heading XML tag. It retrieves the information from the next tag. It identifies the tags using heading tag id.

- **Extracted information summarization**

Information can be gathered from many reference books, therefore sometimes the same information can be duplicated again and again. To avoid that we used a summarization method in python, gensim. This summarization process happens only if the gathered information has more than 10 sentences otherwise it shows the retrieved information as it is.

## 3.0 RESULTS AND DISCUSSION

### 3.1 Result

In order to determine the effectiveness of Structured typed question marking, it is important to carry out evaluations using Teachers answer and students answer.

Now, Consider the following question and the answers.

*Eg:*

*1. What is a database?*

*Teachers Answer - A database is a collection of information that is organized so that it can be easily accessed, managed and updated.*

Teacher's answer will be the Target Sentence for all the below student's answers.

Comparing the Teachers Answer with the 1<sup>st</sup> students answer, target sentence is the teacher's answer.

*Target Sentence = " A database is a collection of information that is organized so that it can be easily accessed, managed and updated."*

*Source Sentence = "A database is a collection of information which can be easily accessed, managed and updated."*

These two sentences are connected to each other within the same flow of information. The system generated Semantic similarity score is close to 1 as it is 0.88. The answers that are this close will get higher marks.

Result set :

```
Semantic Similarity Score : 0.8854166666666666
Word overlap Scores      : [0.9230769230769231, 0.75]
Sentence Length Score    : 0.3611111111111111
Word Similarity Score     : 0.7687061147858074
Noun Similarity Score     : 0.8944271909999159
Verb Similarity Score     : 0.8660254037844387
Adjective Similarity Score : 0.0
```

Figure 3.1.1: Result set 1

Comparing the Teachers Answer with the 2nd students answer, target sentence is the teacher's answer.

*Target Sentence = “ A database is a collection of information that is organized so that it can be easily accessed, managed and updated.”*

*Source Sentence = "A database is a collection of information."*

The system generated Semantic similarity score is close to 0.5 as it is 0.55 and it is an average score so this answer is getting only half of marks out of total marks that assign for a particular question.

Result Set:

```
Semantic Similarity Score : 0.5520833333333334
Word overlap Scores      : [1.0, 0.3125]
Sentence Length Score   : 0.1388888888888889
Word Similarity Score    : 0.5720775535473553
Noun Similarity Score    : 0.89442719099999159
Verb Similarity Score    : 0.0
Adjective Similarity Score : 0.0
```

Figure 3.1.2: Result set 2

Comparing the Teachers Answer with the 2nd students answer, target sentence is the teacher's answer.

*Target Sentence = “ A database is a collection of information that is organized so that it can be easily accessed, managed and updated.”*

*Source Sentence = "A database is a collection of information."*

Result set :

```
Semantic Similarity Score : 0.383827215096565
Word overlap Scores      : [0.75, 0.1875]
Sentence Length Score   : 0.1111111111111111
Word Similarity Score    : 0.31980107453341566
Noun Similarity Score    : 0.31622776601683794
Verb Similarity Score    : 0.5
Adjective Similarity Score : 0.0
```

Figure 3.1.3: Result set 3

The system generated Semantic similarity score is below 0.5 as it is 0.38 and it is a low mark when compare to the above values. so this answer is not getting any mark because the key words have not been included well.

We tested the implementations of Block diagram, Logic Circuit, and Flowchart marking by hosting our web-based solution, and allowing a randomly selected sample of students

to answer a set of questions entered to the system. After the students had answered, we marked the answers using the system, and we obtained the assistance of two real lecturers and a teacher to get the answers marked by an actual person involved in question paper marking, and this helped us in analysing how well the system is performing with the mark allocation to student answers.

We were able to collect 25 answers to a set of Block Diagram Questions and 20 answers to the set of Flowchart Questions while we acquired 85 answers to the set of Logic Gate Questions where we could only get 66 of the Logic Gate answers marked by the time of writing this paper.

Table 3.1.1: Comparison Of System-Allocated Marks Vs Actual Examiner-Allocated Marks For Block Diagram Marking

Range of Difference (in marks)	Number of Answers
0 (same mark)	13
0.5	5
$1 < \text{mark} \leq 2$	3
$3 < \text{mark} \leq 5$	2
$5 < \text{mark} \leq 7$	2
Mark $> 7$	0

Table 3.1.2: Comparison Of System-Allocated Marks Vs Actual Examiner-Allocated Marks For Logic Circuit Marking

Range of Difference	Number of Answers
0 (same mark)	47
$0 < \text{mark} \leq 1$	3
$1 < \text{mark} \leq 2$	5
$2 < \text{mark} \leq 3$	3
$3 < \text{mark} \leq 5$	4
$5 < \text{mark} \leq 7$	4
Mark > 7	0

Table 3.1.3: Comparison Of System-Allocated Marks Vs Actual Examiner-Allocated Marks For Flowchart Marking

Range of Difference	Number of Answers
0 (same mark)	9
$0 < \text{mark} \leq 1$	3
$1 < \text{mark} \leq 2$	3
$3 < \text{mark} \leq 5$	2
$5 < \text{mark} \leq 7$	3
Mark $> 7$	0

In question classification module there are several micro-services to complete the question detail query.

## Keyword Extraction

Eg 1. Question : What is google cloud platform?

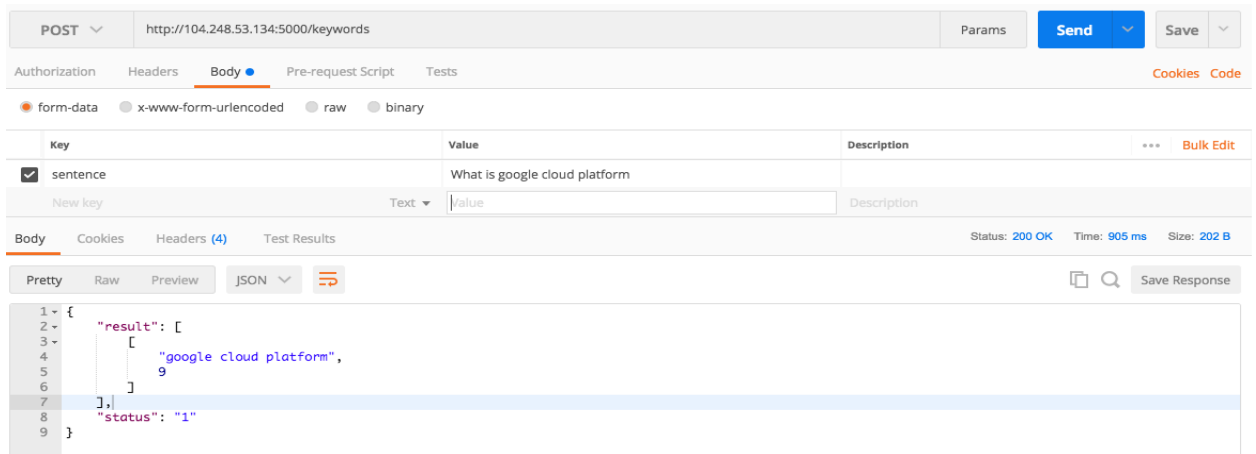


Figure 3.1.4: Result for Example 1

Eg 2. Question : What is NLP?

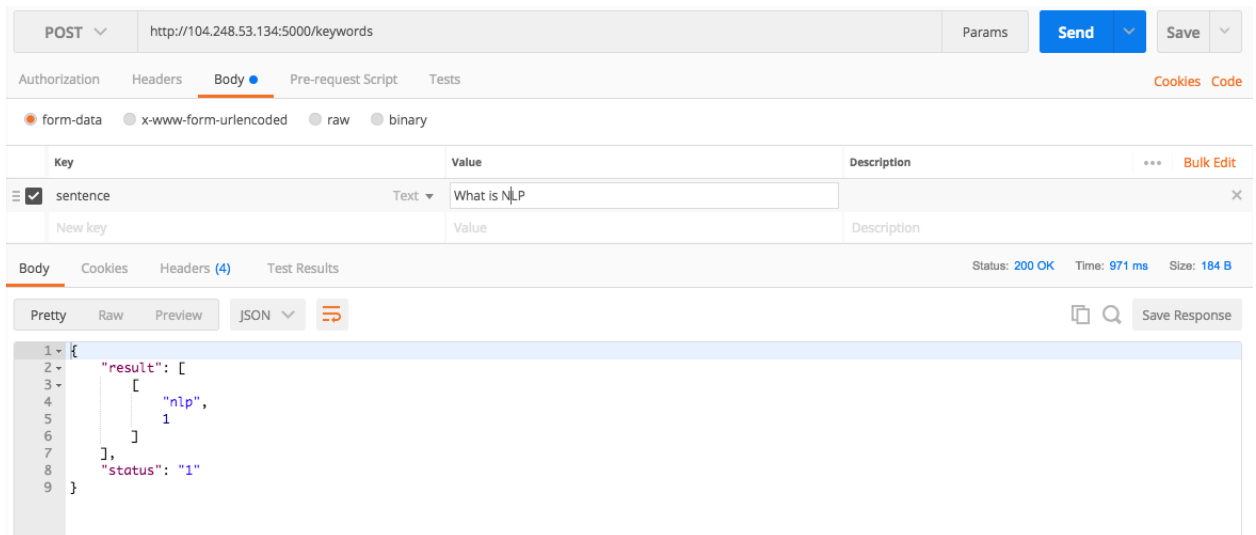


Figure 3.1.5: Result for Example 2



Eg 3. Question : Difference between Natural Language Processing and Image Processing?

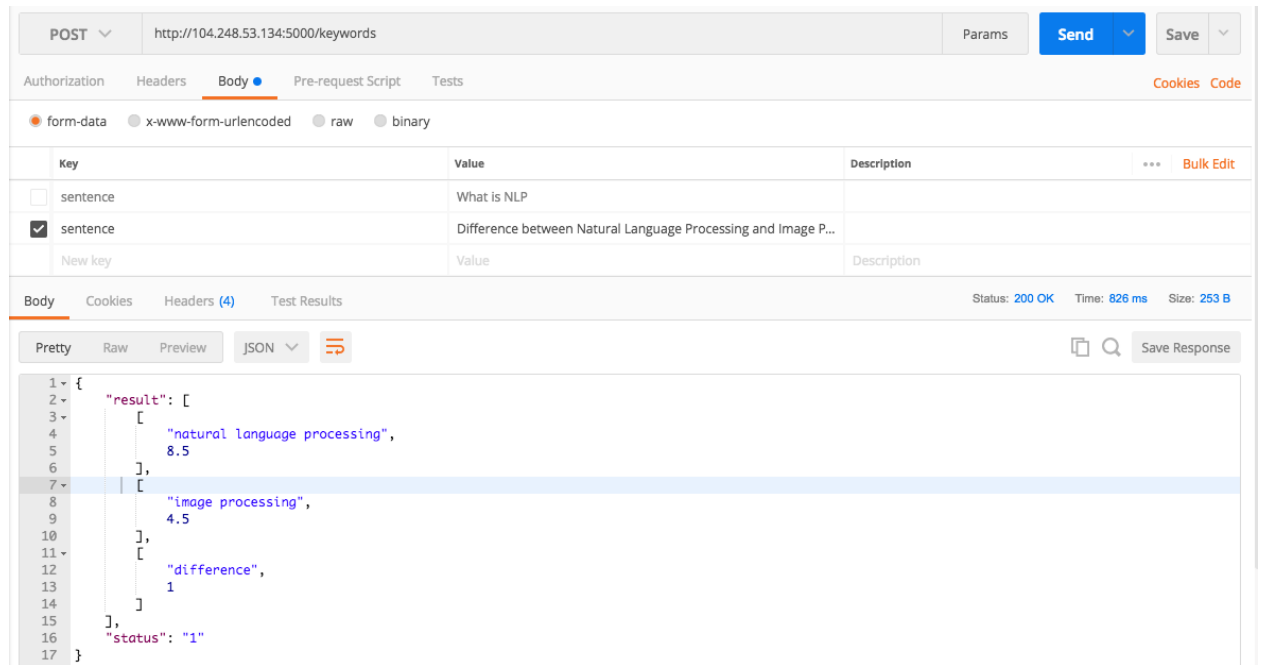


Figure 3.1.6: Result for Example 3

Eg 4. Question : Why do segmentation, CNNs have an encoder-decoder style / structure?

POST http://104.248.53.134:5000/keywords

Authorization Headers Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary

Key	Value	Description
✓ sentence	Why do segmentation, CNNs have an encoder-decoder style /...	
New key	Value	Description

Body Cookies Headers (4) Test Results Status: 200 OK Time: 853 ms Size: 258 B

Pretty Raw Preview JSON Save Response

```

1 {
2   "result": [
3     [
4       "decoder style / structure",
5       16
6     ],
7     [
8       "segmentation",
9       1
10    ],
11    [
12      "cnns",
13      1
14    ],
15    [
16      "encoder",
17      1
18    ]
19  ],
20  "status": "1"
21 }

```

Figure 3.1.7: Result for Example 4

In Eg 1. The python driven microservice was identified as the keyword “google cloud platform”.

In Eg 2. API was identified as “nlp”, the keyword for the given question.

In Eg 3. API was identified as “natural language processing”, “image processing” and “difference” as the keywords and special requirements for the given questions.

In Eg 4. API was identified as “decoder style / structure”, “segmentation”, “encoder” and “cnns” as the keywords and special requirements for the given question.

## Synonyms Finding

### Eg 1. Keyword : google cloud platform

POST http://104.248.53.134:5000/synonyms

Authorization Headers Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary

Key	Value	Description
✓ keyword	google cloud platform	
New key	Text Value	Description

Body Cookies Headers (4) Test Results Status: 200 OK Time: 807 ms Size: 173 B

Pretty Raw Preview JSON Save Response

```

1 {
2   "result": [],
3   "status": "1"
4 }

```

Figure 3.1.8: Result for Example 1

Eg 2. Keyword : nlp

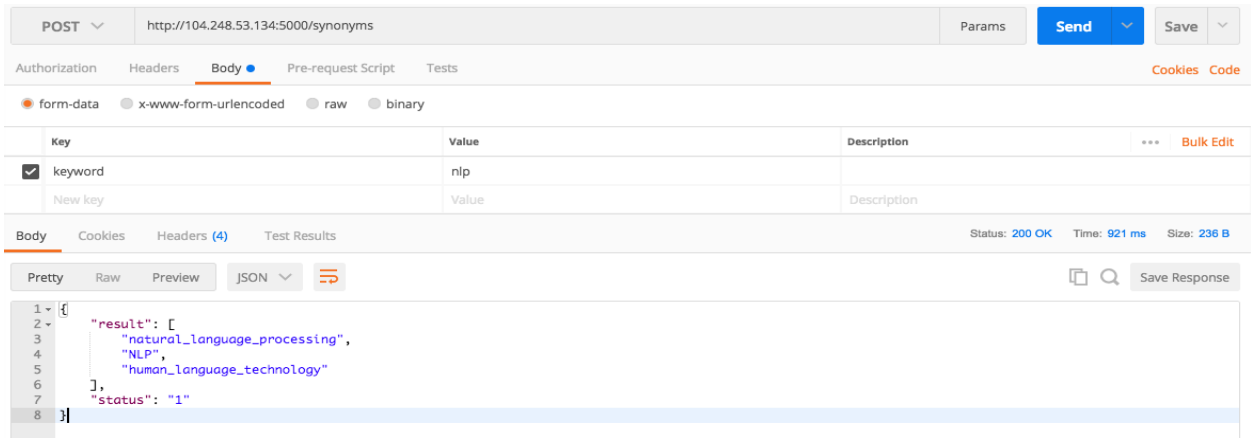


Figure 3.1.9: Result for Example 2

In Eg 1. There are no similar words for keyword “google cloud platform”.

In Eg 2. There are three similar words for keyword “nlp”. Those are ,

"natural\_language\_processing",

"NLP",

"Human\_language\_technology".

## Antonyms Finding

Eg 1. Keyword : google cloud platform

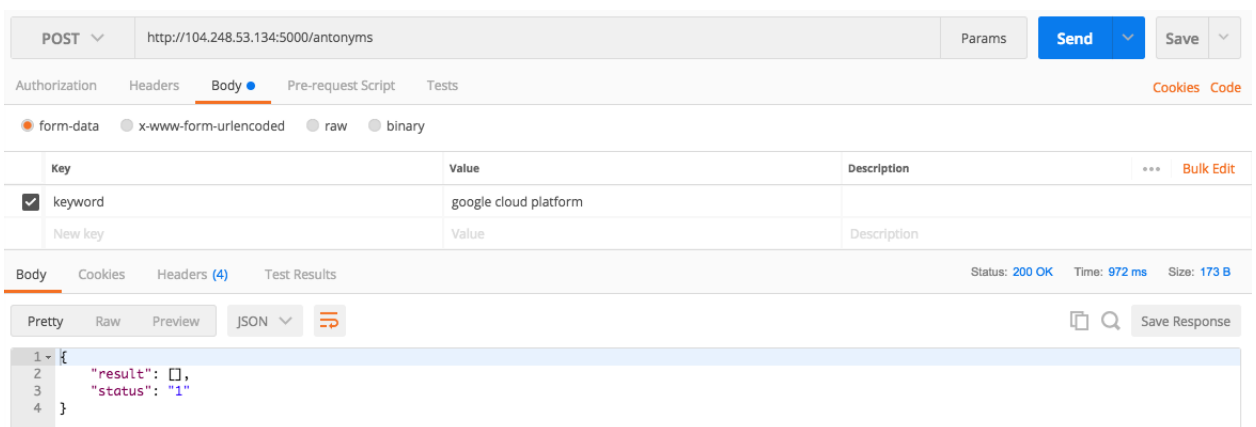


Figure 3.1.10: Result for Example 1

## Eg 2. Keyword : nlp

POST http://104.248.53.134:5000/antonyms

Authorization Headers Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary

Key	Value	Description
<input checked="" type="checkbox"/> keyword	nlp	
New key	Value	Description

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 878 ms Size: 173 B

Pretty Raw Preview JSON

```

1 {
2   "result": [],
3   "status": "1"
4 }
```

Figure 3.1.11: Result for Example 2

There are no antonyms for both Eg 1. And Eg 2.

## Eg 3. Keyword : Analysis

POST http://104.248.53.134:5000/antonyms

Authorization Headers Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary

Key	Value	Description
<input checked="" type="checkbox"/> keyword	Analysis	
New key	Value	Description

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 867 ms Size: 184 B

Pretty Raw Preview JSON

```

1 {
2   "result": [
3     "synthesis"
4   ],
5   "status": "1"
6 }
```

Figure 3.1.12: Result for Example 3

## Eg 4. Keyword : Present

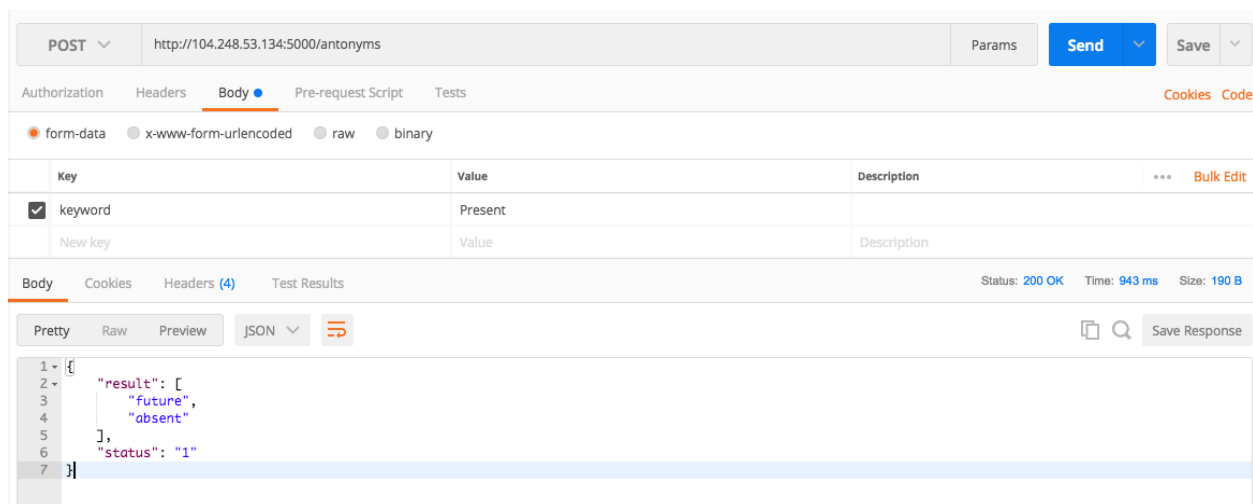


Figure 3.1.13: Result for Example 4

In Eg 3. The micro-service will identify the antonym for keyword “Analysis” as “synthesis”.

In Eg 4. The micro-service will identify the antonyms for keyword “Present” as “future”,and “absent”.

## Question Classification

Eg 1. Question: What is electronic mail?

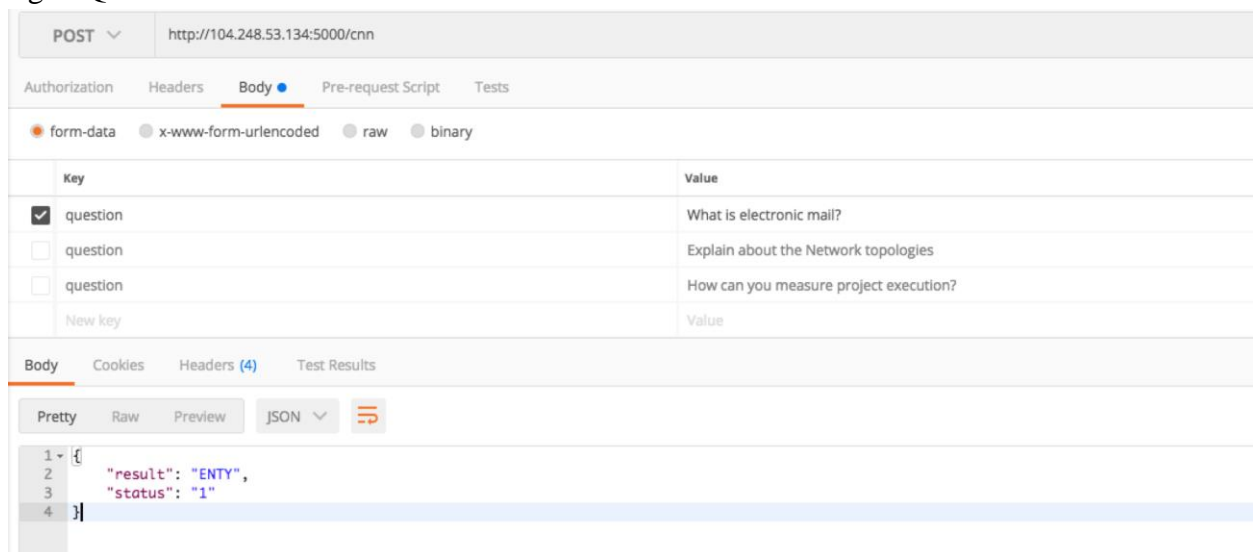


Figure 3.1.14: Result for Example 1

In this example question, the API retrieve this question as entity question.

Eg 2. Question: Explain about the Network topologies

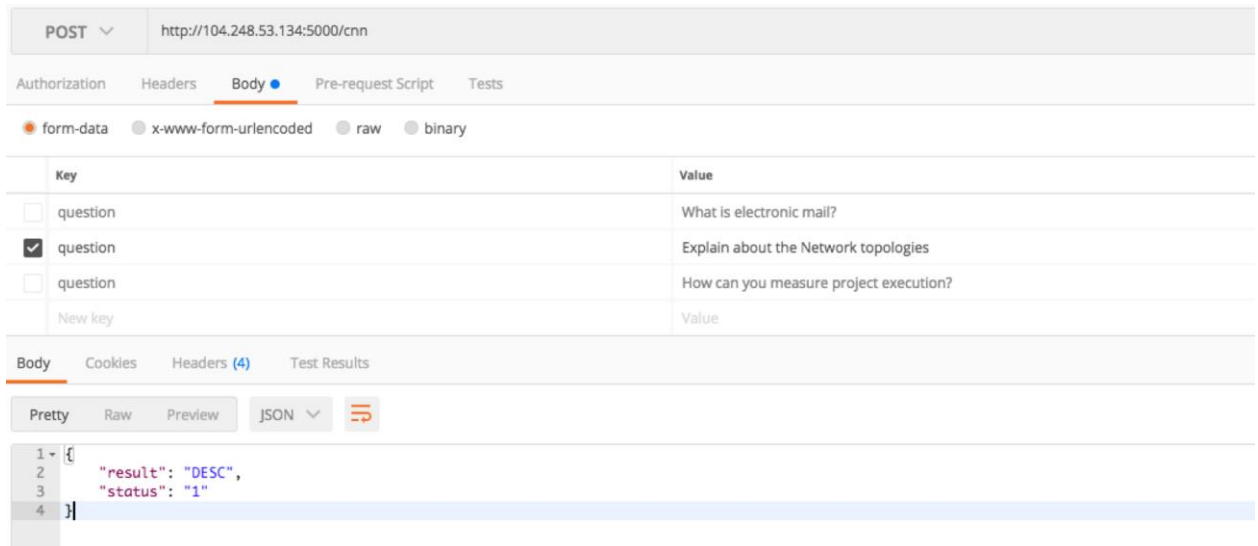


Figure 3.1.15: Result for Example 2

This example question, the API retrieve this question as description required question.

For doubt clarification module we used a PDF written by E. Alpaydin [24]. It is a PDF document about machine learning.

Real headings in the PDF book

- What is machine learning?
- Real world application
- How machine learning works
- Supervised Learning
- Using Supervised learning to predict heart attacks
- Unsupervised learning
- How do you decide which algorithm to use?
- When should you use machine learning?
- Real-world example
- Real-word example
- Real-word example
- Learn more

## Doubt clarification system identified headings

What is Machine Learning?  
 Real-World Applications With the rise in big data, machine learning has become particularly important for solving problems in areas like these:  
 How Machine Learning Works  
 Using Supervised Learning to Predict Heart Attacks  
 Supervised Learning  
 Suppose clinicians want to predict whether  
 Unsupervised Learning  
 Selecting an Algorithm  
 MACHINE LEARNING  
 Real-World Examples  
 Real-World Examples  
 Real-World Examples  
 Learn More

Figure 3.1.16 : Doubt clarification system identified headings

Table 3.1.4 :Extracted headings and real topics comparison

Reference book heading	Identified keywords	Value of Jaccard similarity algorithm
What is machine learning?	machine learning	0.75
How machine learning works	machine learning works	0.8
Supervised Learning	Supervised Learning	1.0
Unsupervised learning	Unsupervised learning	1.0

Table 6 shows the value got from the Jaccard similarity algorithm for keyword and heading comparison.

The system identifies entities of the information using entity recognition. Below figure shows outputs of the doubt clarification system.

```

My/PRP$
name/NN
is/VBZ
(PERSON Black/NNP)
./
I/PRP
live/VBP
in/IN
(GPE Sri/NNP Lanka/NNP)
./
I/PRP
do/VBP
n't/RB
like/VB
to/TO
work/VB
on/IN
Sunday.I/NNP
like/IN
to/TO
play/VB
cricket.I/JJ
work/NN
in/IN
google/NN
./
I/PRP
like/VBP
to/TO
visit/VB
(GPE Canada/NNP)

```

Figure 3.1.17: Identified entities

Students can ask the same question in different ways like,

What is teleconferencing ?

Meaning of teleconferencing

Explain teleconferencing

Briefly explain teleconferencing

Describe teleconferencing

All the above questions ask about teleconferencing in different ways. The doubt clarification feature supports all these types of questions since it identifies the key word of the question. For the above scenarios the identified keyword is teleconferencing. So the system finds related information about teleconferencing. Below figure 3.1.18 shows system extracted information.



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: cmd
d:\SLIIT\git\research_doubtClarification\research_doubtClarification>python findAnswerForQuestion.py
1
C:\Python27\lib\site-packages\gensim\utils.py:1197: UserWarning: detected Windows; aliasing chunkize to chunkize_serial
  warnings.warn("detected Windows; aliasing chunkize to chunkize_serial")
2018-10-06 23:19:55,648 : INFO : 'pattern' package not found; tag filters are not available for English
('xml Document', 'Data Communication Networking.xml')
Teleconferencing allows people in different physical locations to exchange ideas and information interactively using either the telephone or e-mail
. A more advanced form of teleconferencing is videoconferencing, which allows participants to see and hear one another. A videoconferencing system
includes rÇó video cameras, rÇó special microphones rÇó large television monitors rÇó a computer with a special device called a codec that can conv
ert analogue video images and sound waves into digital signals and compress them for transfer over digital telephone lines. (An ISDN rÇó Integrated
Services Digital Network line with a high bandwidth is required to transmit video.)
Numerous organisations offer many facilities to attract customers, benefits such as :ReadyConnect Make audio conference calls without advance reser
vations or operator assistance. ActionCall Conduct an effective, high-level audio conference call with full operator support. Passcode For a secure
d audio conference, access can be limited to select participants by issuing a passcode. Worldwide Participation Conference via a video monitor to p
articipants in worldwide offices. Audio & Video Presentation Stream your audio or video message over the Internet to participants for an effective
conference presentation. ActionData Share documentation or present slides over the web with your audience while conducting an audio conference.
('xml Document', 'SRSDocument-Learning tool.xml')

```

Figure 3.1.18: Extracted information from the system

Two reference books can contain information about teleconferencing. In that case the system takes both descriptions and summarize those and displays it to the user. Other than that system summarises the lengthy descriptions also.

Table 3.1.5: Result table for information retrieval feature

Question	Answer found/ not	Accuracy
What is internet?	Found	Not accurate
What is world wide web?	Found	Not accurate
What is intranet?	Found	Medium
What is electronic mail?	Not Found	
What is telephone?	Not Found	
Explain teleconferencing	Found	High
Describe asynchronous and synchronous transmission	Found	High
What is LAN?	Not Found	
What is WAN?	Found	High
What is MAN?	Not found	
Explain Network topologies	Found	High
Describe Star topologies	Not Found	
What is bus topologies?	Found	Low
What is ring topology?	Found	High

The table above shows the accuracy of the retrieved information. We used a reference book on Data Communication and Networking to select 14 questions from different topics, and test our system by feeding these selected questions. We were able to obtain information for 9 of these questions.

### 3.2 Research Findings

If we disregard the 0.5 differences or 0 – 1 differences in marks, we can see that around 72% of the marks allocated for block diagram answers by the system, 60% of the marks allocated for flowchart answers, and 75.8% marks allocated for logic circuit answers using the system are equivalent to the marks allocated by the actual lecturer or teacher.

It is also noticeable that while the most part marks fell under the ranges 1 - 2, 2 – 3, and 3 – 5, none of the marks generated by the system had a huge variance, and did not completely differ from the mark the actual marker allocated. The test results manifests a few scenarios where the system-allocated marks diverge from actual examiner allocated marks by considerable amounts as well.

Although the system is not 100% accurate in marking these diagram-type answers, it can be seen that a majority were marked accurately while rest of the marks have not deviated by very huge amounts.

Doubt Clarification feature identifies all the topics in the real document most of the time using XML tags. According to the Figure 3.1.16 it identifies all 12 topics. Sometimes the system identifies normal text also as headings when document is not well formatted. Doubt Clarification feature has given correct answers for five questions and one middle range answer for one question over 14 questions. This feature has failed to identify four questions and it has displayed the 'Not Found' message to the user.

It can be observed that the Doubt Clarification feature is not 100% accurate with all the available PDF documents.

### **3.3 Discussion**

When we analyze the block diagram marking results, we can see that as the mark ranges or gap increases, the number of marks that fell in that range has decreased which means the system marks more answers properly. Although there are fluctuations in the marks that fell in each gap range for Logic Circuit marking and Flowchart marking, we can see that the basic trend is decreasing which also suggests that although there are some scenarios which might have issues with the marking, a majority is marked well.

Two Logic Circuits can be said to be equivalent if both the circuits return the same output for all the binary combinations [14]. 13 answers out of the 47 answers marked correct were marked through simulation while the other 34 were marked through graph matching. This exhibits how simulation can be used to improve accuracy through its capability to recognize alternative answers correctly.

When it comes to the performance of the Logic Circuit answer marking, we monitored the time taken by the system to mark the answer using graph matching, and figured that it was much less compared to simulation. When graph matching took a time period around one second to mark an answer, simulation took around six seconds to mark the same answer. Therefore, when marking a Logic Circuit answer, we decided to allow the teacher to select whether the teacher wants an exact match done using graph matching, or whether alternative answers are also to be allowed which will be marked using simulation if an exact match could not be found, and this helped us improve the performance during marking. It is also possible for the teacher to decide in using simulation to guarantee correct mark allocation regardless of the alternative answers, as simulation promises the return of the correct result at all times while graph matching may tend to give slightly different marks at times.

The doubt clarification feature, allows only the direct questions like describe, what, when and who. This module takes sometime to convert pdf documents in to XML and it can maximally convert 100 pages per once. To increase the accuracy we get the important chapter from the reference books.

When student add their questions, they should add the subject of the question and the grade of the student to make it easier for the document classification. Identifying topics of the document is more accurate when the author of the book uses standard document format. This feature always tries to give related information within a short time period.

**4.0 SUMMARY OF EACH STUDENT'S CONTRIBUTION**

<b>ID Number</b>	<b>Name</b>	<b>Contribution</b>
IT1503791 6	R.R.A.M.P.Jayawardena	<ul style="list-style-type: none"> <li>● Commercialization aspects of the product</li> <li>● Classification Module <ul style="list-style-type: none"> <li>○ Background and literature reviews</li> <li>○ Specific Objectives</li> <li>○ Methodology</li> <li>○ Testing and Implementation</li> <li>○ Results and Discussion</li> </ul> </li> </ul>
IT1503586 8	P.S.Suriyaarachchi	<ul style="list-style-type: none"> <li>● Acknowledgement</li> <li>● Research Objectives</li> <li>● Information Retrieval Module <ul style="list-style-type: none"> <li>○ Background and literature reviews</li> <li>○ Specific Objectives</li> <li>○ Methodology</li> <li>○ Testing and Implementation</li> <li>○ Results and Discussion</li> </ul> </li> </ul>
IT1504095 4	G.A.D.Thiwanthi	<ul style="list-style-type: none"> <li>● Abstract</li> <li>● Research Gap</li> <li>● Conclusion</li> <li>● Diagram Marking Module <ul style="list-style-type: none"> <li>○ Background and Literature Reviews</li> <li>○ Specific Objectives</li> <li>○ Methodology</li> <li>○ Testing and Implementation</li> </ul> </li> </ul>

		<ul style="list-style-type: none"> <li>○ Results and Discussion</li> </ul>
IT1505719 8	K.I.Withana	<ul style="list-style-type: none"> <li>● Research Problem</li> <li>● Structured question marking module                             <ul style="list-style-type: none"> <li>○ Background and literature reviews</li> <li>○ Specific Objectives</li> <li>○ Methodology</li> <li>○ Testing and Implementation</li> <li>○ Results and Discussion</li> </ul> </li> </ul>

## 5.0 CONCLUSION

Question Paper Marking is a challenging task which requires a lot of resources like effort, time, and many examiners in abundance. Some of the major issues associated with the question paper marking by teachers are fatigue, lack of motivation, and the fact that teachers might have other responsibilities in both their private and professional, academic life. These can lead to either intentional or unintentional unfair marking where some students may undeservingly receive more marks when compared to the others, although the quality of the answer is similar. Therefore, it would be a great advantage if the question paper marking process could be automated, thereby, removing the time and effort consumption in handling and marking the papers manually, while also enabling teachers to focus more on their teaching schedules and materials, while they can also focus more on the important examination questions that the majority of students may have answered wrong. This would enable them to be more useful to students by spending time sharing their knowledge, guiding the students and providing feedback.

The process of doubt clarification is time consuming, when the time spent on doubt clarification could be used by students for studying or focussing on other important areas.

As there is room for improvement in the area of Structured Question Marking although some research has been done in the area and the existing systems are not accurate, it is worthwhile to carry out a research in this area. Owing to the fact there are no existing systems except for CourseMaster in the area of logic circuit and flowchart answer marking, and also since there is no present system that could mark block diagrams, it would be extremely useful to build a system that can support the marking of these set of diagrams. When it comes to doubt clarification, since the process consumes time, and the web retrieved answers may not be relevant and summarized as per the student's grade and knowledge, it is extremely valuable if a system with these advantageous abilities could be built.

For the Structured question marking we used the semantic sentence similarity approach for language processing using various word senses of WordNet. This method depends upon all the semantic relations of word senses across the different synsets using WordNet for different parts of speech of words. In addition to that this method can be used to identify the semantic similarity among positive and negative sentences too.

Although there are previous research work done in the context of graph matching using several exact and inexact graph matching algorithms like Ullmann's and A-star algorithm respectively, we have figured that there, the algorithms DFS and BFS would be more applicable to handle our scenario of question and answer marking. Therefore, in the diagram marking module, we have explored the application of DFS in graph marking to mark Block diagrams and Flowcharts, while BFS was used for graph matching and simulation of Logic Circuits. Nodes of Block diagrams were matched by checking semantic similarity using path similarity in WordNet hierarchy. Semantic similarity check based on WordNet path similarity was chosen over the String Similarity Metrics like Jaccard, Levenshtein, and Sequence Matcher, as it performed better when compared to the String Similarity metrics which took into account only the words and the characters in the string, without identifying the similarity of the meaning. Logic Gate Simulation was chosen over other available methods as it promised better results while removing the complexities in some cases and improving performance. As an attempt to verify the flowchart logic alongside the structural analysis of graphs, we converted the flowchart to a program to test its behaviour by providing inputs and observing the outputs to determine whether the output is as expected which would indicate that there is a high chance that the program is behaving as expected. It is evident that we have achieved accuracy above 70% for both block diagrams and logic circuits, and accuracy around 60% for flowcharts from the results shown above. The current system can be expanded to support other types of diagrams in the future like ER diagrams, Class diagrams, and Use Case diagrams as few systems exist in the diagram marking area. Additionally, we can strive in improving accuracy of Block diagrams and Logic circuit marking by investigating other scenarios that can be added to enable more graph matching. For instance, analysing the depth more in block diagrams if a match could not be found, although complications exist as students can make mistakes at any position where supporting one particular scenario can contradict other situations.

In doubt clarification feature, student's question is classified using the Convolutional Neural Network, and the keywords extraction function works with Python RAKE, while the synonym retrieving function runs on top of the WordNet in Python. System takes reference books in PDF format. Then the system converts the document into XML format using an online site called pdfx v1.9. Topic, and the answer are retrieved by using XML tags. To check the similarity of the retrieved information we use the dice algorithm. Summarization is done using an extractive approach. The algorithm used for the extractive approach is gensim.



It is evident that our web-based solution could help teachers and students all around the world with its capabilities, as there are no systems that have all these capabilities in one system. Our system would also help in solving most of the issues that are associated with the traditional approach of question paper marking.

## 6.0 REFERENCES

- [1] Coursetoolkit.com. (2018). Features: Automatic Test Marking - Course Toolkit LMS. [online] Available at: <http://www.coursetoolkit.com/features?feature=automated-test-marking> [Accessed 23 Mar. 2018].
- [2] S. G. Pulman and J. Z. Sukkarieh, “Automatic Short Answer Marking.”
- [3] Raheel Siddiqi, “A Region-based Approach to the Automated Marking of Short Textual Answers,” *SSU Res. J. Engg. Tech.*, vol. 1, no. 1, p. 7, 2011.
- [4] M. A. Tayal, M. Raghuwanshi, and L. Malik, “Word net based method for determining semantic sentence similarity through various word senses,” in *Proceedings of the 11th International Conference on Natural Language Processing*, 2014, pp. 139–145.
- [5] C. A. Higgins, P. Symeonidis, and A. Tsintsifas, “The marking system for CourseMaster,” *ACM SIGCSE Bull.*, vol. 34, no. 3, pp. 46–50, 2002.
- [6] C. Higgins, P. Symeonidis, and A. Tsintsifas, “Diagram-based CBA using DATsys and CourseMaster,” *Proc. - Int. Conf. Comput. Educ. ICCE 2002*, pp. 167–172, 2002.
- [7] C. Djeraba, M. Revenu, A. Baskurt, and C. Wolf, “T hèse de l ’ U niversité de L yon I nexact graph matching techniques : A pplication to object detection and human action recognition,” vol. 33, 2010.
- [8] V. Carletti, P. Foggia, and M. Vento, “Performance Comparison of Five Exact Graph Matching Algorithms on Biological Databases,” *New Trends Image Anal. Process. - ICIAP*, 2013.
- [9] H. Bunke and G. Allermann, “Inexact graph recognition matching for structural pattern,” *Pattern Recognit. Lett.*, vol. 1, no. May, pp. 245–253, 1983.
- [10] M. Neuhaus, K. Riesen, and H. Bunke, “Fast Suboptimal Algorithms for the Computation of Graph Edit Distance,” no. Im, pp. 163–172, 2006.

- [11] Z. Abu-Aisheh *et al.*, “An Exact Graph Edit Distance Algorithm for Solving Pattern Recognition Problems,” *4th Int. Conference Pattern Recognit. Appl. Methods 2015*, pp. 271–278, 2015.
- [12] K. Riesen, “Structural Pattern Recognition with Graph Edit Distance,” pp. 29–45, 2015.
- [13] R. Dijkman, M. Dumas, and L. García-Bañuelos, “Graph matching algorithms for business process model similarity search,” *7th Int. Conf. Bus. Process Manag.*, vol. Business P, pp. 48–63, 2009.
- [14] W. Kunz, “HANNIBAL: An Efficient Tool for Logic Verification Based on Recursive Learning”.
- [15] C.L. Berman, and L.H. Trevillyan, “Functional Comparison of Logic Designs for VLSI Circuits”.
- [16] J. S. Kuruvila, M. Lal, R. Roy, T. Baby, S. Jamal, and K. K. Sherly, “Flowchart Plagiarism Detection System: An Image Processing Approach,” *Procedia Comput. Sci.*, vol. 115, pp. 533–540, 2017.
- [17] R. Stone, F. Batmaz and C. Hinde, “Drawing and Marking Graph Diagrams,” 15 December 2015. [Online]. Available: <https://www.tandfonline.com/doi/full/10.11120/ital.2009.08020045>. [Accessed May 2018].
- [18] “START”. [Online]. Available: <http://start.csail.mit.edu/index.php>. [Accessed May 2018].
- [19] Á. Rodrigo, J. Perez-Iglesias, A. Peñas, G. Garrido, and L. Araujo, “A Question Answering System based on Information Retrieval and Validation,” *Noteb. Pap. CLEF 2010 LABs Work.*, pp. 22–23, 2010.
- [20] B. Ojokoh and P. Ayokunle, “Online Question Answering System,” *Int. J. Comput. Sci. Res. Appl. Int. J. Comput. Sci. Res. Appl. ISSN*, vol. 3, no. 3, pp. 2–9, 2013.

[21] “pdfx v1.9”. [Online]. Available: <http://pdfx.cs.man.ac.uk/>. [Accessed May 2018].

[22] Dell Zhang , Wee Sun Lee. “A web-based Question Answering system,” Available: <https://pdfs.semanticscholar.org/ad23/647c895d57668fc202259dccbf29edb9e683.pdf>

[23] N. Mittal Å, B. Agarwal Å, H. Mantri Å, R. Kumar Goyal Å, and M. Kumar Jain Å Å, “Extractive Text Summarization,” *Res. Artic. Int. J. Curr. Eng. Technol.*, vol. 870, no. 2, 2014.

[24] E. Alpaydin, “Machine Learning,” p. 168, 2016.