Project Report on

GENERATING DIVERSE QUESTION AND ANSWER PAIRS FROM A GIVEN PROBLEM CONTEXT

Submitted in partial fulfillment of the requirements

of the degree of Bachelor in Engineering

by

Pratiksha Rajesh Rao	BE3 - 44
Tanay Navneet Jhawar	BE4 - 22
Yash Avinash Kachave	BE4 - 23

Under the guidance of

Prof. Vaishali Hirlekar

DEPARTMENT OF COMPUTER ENGINEERING

SHAH AND ANCHOR KUTCHHI ENGINEERING COLLEGE

CHEMBUR, MUMBAI – 400088.

2021 – 2022

www.shahandanchor.com Tel: 022 2558 0854

Mahavir Education Trust's

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE

Mahavir Education Trust Chowk, W.T. Patil Marg, Chembur, Mumbai 400 088

Affiliated to University of Mumbai, Approved by D.T.E. & A.I.C.T.E.

ISO 9001:2008 Certified

UG Programs Computer Engineering & Information Technology accredited by NBA for 3 years w.e.f. 1st July 2019

Certificate

This is to certify that the report of the project entitled

Generating Diverse and Consistent Question and Answer pairs from Problem Contexts

is a bonafide work of

Name of Student	Class	Roll No.
Pratiksha Rajesh Rao	BE3	44
Tanay Navneet Jhawar	BE4	22
Yash Avinash Kachave	BE4	23

submitted to the

UNIVERSITY OF MUMBAI

during semester VIII in partial fulfilment of the requirement for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER ENGINEERING.

(Ms. Vaishali Hirlekar)

Guide

Signature valid
Digitally signed by UDAY KRISHNA BHAVE
Date: 2022.05.02 18:[illegible]:05:30
Reason: Approved
Location: Mumbai

(Prof. Uday Bhave)

I/c Head of Department

Signature valid
Digitally signed by BHAVESH VALJI PATEL
Date: 2022.05.[illegible]2:05:08 +05:30

(Dr. Bhavesh Patel)

Principal

Approval for Project Report for B. E. Semester VIII

This project report entitled *"*GENERATING DIVERSE QUESTION AND ANSWER PAIRS FROM A GIVEN PROBLEM CONTEXT" by Pratiksha Rajesh Rao, Tanay Navneet Jhawar and Yash Avinash Kachave is approved for Semester VIII in partial fulfilment of the requirement for the award of the degree of Bachelor of Engineering.

Examiners

1._____

2._____

Guide

1._____

Date:

Place:

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Name of the Student	Class	Roll No.	Signature
Pratiksha Rajesh Rao	BE3	44	
Tanay Navneet Jhawar	BE4	22	
Yash Avinash Kachave	BE4	23	

Date:

Place:

Attendance Certificate (from college)

Date: 04/05/2022

To,
The Principal
Shah and Anchor Kutchhi Engineering College,
Chembur, Mumbai-88

Subject: Confirmation of Attendance

Respected Sir,

This is to certify that Final year (BE) students Pratiksha Rajesh Rao, Tanay Navneet Jhawar, and Yash Avinash Kachave have duly attended the sessions on the day allotted to them during the period from 02/02/2022 to 28/04/2022 for performing the Project titled "GENERATING DIVERSE QUESTION AND ANSWER PAIRS FROM A GIVEN PROBLEM CONTEXT". They were punctual and regular in their attendance. Following is the detailed record of the student's attendance.

Attendance Record:

Date	**Pratiksha Rajesh Rao**	**Tanay Navneet Jhawar**	**Yash Avinash Kachave**
02/02/2022	Present	Present	Present
09/02/2022	Present	Present	Present
16/02/2022	Present	Present	Present
02/03/2022	Present	Present	Present
09/03/2022	Present	Present	Present
16/03/2022	Present	Present	Present
23/03/2022	Present	Present	Present
06/04/2022	Present	Present	Present
13/04/2022	Present	Present	Present
20/04/2022	Present	Present	Present
28/04/2022	Present	Present	Present

Prof. Vaishali Hirlekar

Signature and Name of Internal Guide

ABSTRACT

In the education industry answering questions is used as a common parameter to judge one's understanding of a topic. Taking quizzes on a regular basis helps an individual feel confident and it also helps the professor assess the student's understanding on a particular topic. Generating question and answer pairs is a time-consuming task. To solve this problem, this paper discusses methods to generate automatic Natural Language Processing models which create diverse types of question-answer pairs. The model takes an input in the form of text in the English language and produces output as Complex Questions, Multiple Choice Questions with relevant distractors, and Fill in the Blanks type of questions. To generate Complex Questions a Rule-Based Algorithm is used. To generate Multiple Choice Questions and Fill in the Blanks type questions, a Vector Algorithm from the GLoVe Model is used along with Rule-Based Algorithms. This report also includes a detailed explanation of the analysis of the pattern and rules that are observed in the question-making process. The SQuAD dataset is used for this analysis and used the same dataset to train the model. The implementation process of this model focused on generating diverse questions with higher syntactic correctness than the existing models. The approach mentioned in this report can be used in the fields of education, entertainment, generation of quizzes, virtual learning assistance and to get a deeper insight into any topic.

Keywords- *Automatic question generation, analytical model, complex question, discourse tokens, education, electronic learning, fill in the blanks, multiple choice question, natural language processing, quiz, rule-based, text analysis*

TABLE OF CONTENTS

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 Background

Answering questions about a given topic is an efficient way to assess one's knowledge of any subject. It is challenging and time-consuming to come up with a new set of questions on a regular basis. A question-answer system can be used in many forms in the education industry. It can be used by teachers to conduct quizzes, make flashcards, judge how much a student has understood a particular topic, have a fun interaction with the class, etc. Due to the pandemic, teachers find it difficult to understand if the student has truly grabbed the information that has been taught. Hence taking daily quizzes makes it easier to access and builds the student-teacher relationship on a deeper level. Generating a different set of questions is a very difficult and time-consuming task. We focus on solving this problem.

Our model uses Natural Language Processing (NLP), that is, it uses the technique of teaching a machine to understand human language and generate questions in human language with correct grammar. There have been a few attempts in making a question-answer model which is mentioned in this report but the major drawback of each model was the accuracy of the questions which were generated. The accuracy depends on the quality of the questions generated, how relevant the question is and how much is the question grammatical correct. Keeping these metrics on our mind, we have built the question answering model.

1.2 Objective

Automating the process of question generation will save time and energy. The main objective of our model is to help the user generate diverse type of questions by providing a

text as the input. The text can contain any type of article from a web page, a paragraph from a book, Wikipedia content, etc. This methodology is also beneficial for users to create quality questions without having information about the input text in just one click. This helps reduce cost of having to hire subject expert. Our approach will assist in the generation of variety of question and answer pairs that may be utilized in the fields of education, entertainment, generation of quizzes, virtual learning assistance and to get a deeper insight of any topic. Our objective is to correctly generate three types of questions that is, complex questions, multiple choice questions, and fill in the blanks. This work intends to use Rule Based algorithms to solve Question Answering Model Problem.

1.3 Organization of the report

The main body of the report is preceded by detailed contents including list of figures and tables. Chapter 1 gives an introduction to the project. It debriefs upon the importance of the topic, the purpose of pursuing this project. Chapter 2 discusses the current implementation and research done to tackle the problems faced. Chapter 3 discusses the purpose and product scope of the software. It also elaborates about the working environment, design and implementation constraints and user documentation. Chapter 4 shows the timeline of the project. Chapter 5 explains about the data analysis and the algorithm used for the model. Chapter 6 describes the methodology of each module and snapshots of the implementation. Chapter 7 tabulates the black box testing of the system. Chapter 8 deals with the analysis of the results obtained from our model. Chapter 9 concludes the report with describing the future scopes of the system. The main report is followed by references which have been used for considering certain inputs in the making of the model. A copy of the IEEE paper which was published is attached in the end along with acceptance letter and the presentation certificate provided by ICEARS conference committee.

CHAPTER 2

LITERATURE SURVEY

2.1 Survey Existing System

There have been many approaches that have been taken in the past to solve the issues faced by question-answer generating models. Below are some of the work in this field that has been summarized.

2.1.1 Focused Questions and Answer Generation by Key Content Selection [1]

The model mentioned in this paper uses a module called focus generator. This module guides the decoder in an existing "encoder-decoder" model to generate questions based on selected focus contents [1]. The input text is divided into three focus sequences based on the Neural Entity Selection algorithm. The module focuses on these sequences to generate questions. It generates a range of complexities in the question by combining focus sequences according to the need. This model was highly inspired by the baseline model by Cho et al [6]. The extracted training, development and test sets contain 86,635, 8,965 and 8,964 triples respectively [7], on the SQuAD dataset. It automatically generates Answer tags without manual intervention using the NES algorithm. The drawback of this model was its inability to produce diverse questions and this can be overcome using modern libraries and architectures of Natural Language Processing.

2.1.2 Questionator -Automated Question Generation using Deep Learning [2]

The model mentioned in this paper is based on Natural Language Processing techniques and Image Captioning techniques. The model used ILSVRC-2012-CLS Image classification dataset [9]. The image captioning module converts a given input image into

a natural language description [2]. Later this image was converted to a question. They have used Convolutional Neural Network and Long Short-Term Memory (LSTM) as encoders and decoders algorithms. The drawback of this model was that it was incompetent to create questions if the database did not have a similar image saved. Since every image had to be captioned it is difficult to get or create a diverse database. Since similar images can lead to misleading questions, the model needs to be trained with an advanced dataset.

2.1.3 Computational Intelligence Framework for Automatic Quiz Question Generation[3]

This framework is based on information. In this paper, for "WH"-questions generation, pattern matching is used to a certain extent [3]. They used the proposed methodology from Agarwal et al. [11] to generate MCQ questions. They made use of the LSTM neural network layer to train their model. Even though shallow LSTM networks already perform better than shallow feedforward networks in language modeling, LSTMs still show more consistent improvements when adding additional hidden layers than their feedforward counterparts [10]. They made use of Named Entity Recognition (NER) and Super-sense tagging (SST) to create WH-questions. The major drawback of this model was to get adequate pronouns in the formation of the questions and answers, this limitation was due to the use of the LSTM approach.

2.1.4 Thematic Question Generation over Knowledge Bases [4]

This model follows a template-based approach for the generation of Multiple Choice Questions with distractors. The model also follows the approach of regenerating question answering templates automatically by importing the used templates, hence more than 2000 templates can be imported at a time. They have referred to the Wikipedia structure to identify and establish the limits of the topic. They have used and updated existing question answering templates by [12]. To choose the distractors correctly, they made use of the Page Rank score. Learning to rank is a relatively new research area, which received increasing attention in both the Information Retrieval and Machine Learning research communities, during the past decade [15]. The distractors with a higher score are selected and this method helps to filter ineffective distractors. Their approach was experimented on a dataset with 1430 questions and was able to generate 1394 questions successfully. The drawback of the model was that it was not able to include literal in the question generation process.

2.1.5 Automatic Question Generation from Children's Stories for Companion Chatbot[5]

The previous work in Chinese question generation relied solely on rule based approach. It generated an unsatisfactory performance due to the complexity in language and limitation of parser. The proposed model generates automatic questions in the Chinese language for children story book. In Chinese, generating questions can usually replace target answer phrases with interrogatives [5]. This method ensures that the quality of the question is maintained. They also use a ranking model to remove lower-quality questions. After evaluation, it was concluded that 50% of the questions generated were correct. Who-type question was not generated correctly due to lack of semantic understanding.

2.2 Limitation of existing System

The major drawbacks of the existing models were that the models were not able to get a higher accuracy at generating question-answer pairs and could not handle complex syntactic questions. If the Question Answer generating model's parser creates flawed results, the questions would be grammatically incorrect. Some distractors were deficient and of substandard quality. The distractors dataset was not diverse hence could not create diversity to choose.

2.3 Problem Statement and Objective

2.3.1 Problem Statement

To create a model that can generate question and answer pairs from a text that is both diverse and effective.

2.3.2 Objectives

- To allow the user to save time in making questions.
- To develope a wide range of questions that are both understandable and grammatically correct.
- To save time of the user to create questions without any domain expertise.
- To create a friendly user interface.

2.4 Scope

The accuracy and correctness of the questions can be increased by introducing some new rule-based algorithm. New datasets can be added to increase the quality of distractors. Gamification modules can be added to make the application more interactive.

CHAPTER 3

SOFTWARE REQUIREMENT SPECIFICATION (SRS)

3.1 Introduction of SRS

3.1.1 Purpose

Generating diverse questions on a regular basis is a time-consuming task. This application can be used by an individual to generate variety of complex type of questions, fill in the blanks and multiple-choice questions given a text is provided. The user need not have all the knowledge about the input text to make consistent quality questions effectively.

3.1.2 Document Conventions

Every requirement statement has its own priority.

3.1.3 Intended Audience and Reading Suggestions

The SRS for this application can be used by variety of audience like developers and team managers. It can also be used by users who would like to know about the application in detail. The document is not required to be read in a sequential manner; user can jump to any section that they find relevant.

3.1.4 Product Scope

This product aims to make effective question and answer pair from a text without wasting any time and it takes less effort than the traditional way of making questions. It also aims to create diverse questions which are easy to understand and are grammatically correct. It can be used for educational purposes for example to practice a specific chapter by giving quizzes. Students tend to be more interested when there are external activities and game-based teaching involved, hence this application will be very useful in the educational industry.

3.1.5 References

https://spacy.io

https://www.nltk.org/

3.2 Overall Description

3.2.1 Product Perspective

This is a new self-contained product. It can be used on its own to create an assortment of question answer pairs.

3.2.2 Product Functions

The functions of the product occur in the order that is mentioned:

- Product takes input text from the user
- User selects the type of questions to be printed
- User selects the number of the questions to be printed
- If Complex question is chosen, product creates an output with complex question and answer pair from a text
- If Multiple Choice Questions is chosen, product creates an output with multiple choice questions with effective distractors.
- If Fill in the blanks is chosen, product creates an output with fill in the blank questions

3.2.3 User Classes and Characteristics

The target users of this product are teachers, students, practitioners, researchers, quiz makers, and more.

General User: This user will be able to input a text and generate an assortment of questions.

3.2.4 Operating Environment

The components of the software must operate within the following:

- Windows operating system which is of version 7 or newer.
- MAC OS X version 10.7 or higher
- Linux operating system

3.2.5 Design and Implementation Constraints

The variety and accuracy of distractors used for multiple-choice questions depend on the dataset used in the model. The quality of the questions is proportional to the set of words acquired from sentences after stemming and lemmatization of the tokenized words.

3.2.6 User Documentation

A user manual in the form of a PDF must be provided with the software. It will contain the details to run the application along with the answers to frequently asked questions.

3.2.7 Assumptions and Dependencies

There is no specific assumption or dependencies which are considered at this moment.

3.3 External Interface Requirements

3.3.1 User Interfaces

The user interface includes a text box for the user to intake the input data. The screen will consist of three buttons. The first button is to generate complex questions, the next button generates multiple choice questions and the last button is used to generate fill in the blank's questions.

3.3.2 Hardware Interfaces

The application runs on the internet; the hardware interfaces shall be compatible to connect to any modem, WAN-LAN, ethernet cross-cable, etc. type of connection.

3.3.3 Software Interfaces

The software takes an input and processes it through Python libraries like spaCy, NLTK, and gensim to generate the outputs.

3.4 System Features

3.4.1 Generation of complex questions

3.4.1.1 Description and Priority

Generation of complex sentences consists of generating questions like list the examples of xyz or questions which start from is, are, why, when, who, how, etc. The priority of this feature is same as the priority of the other mentioned features.

3.4.2 Generation of Multiple-Choice Questions

3.4.2.1 Description and Priority

Generation of multiple-choice questions consists of generating questions with maximum of 5 distractors. The priority of this feature is same as the priority of the other mentioned features.

3.4.2.2 Stimulus/Response Sequences

After we take the input text and the user selects the button of Multiple-Choice Questions, a dialog box pops up asking for the number of questions to be generated. Once the user enters the required number, multiple choice questions are generated for the user.

3.4.3 Generation of Fill in the Blanks type of Questions

3.4.3.1 Description and Priority

Generation of fill in the blanks type questions consists of generating sentences with appropriate blanks. The priority of this feature is same as the priority of the other mentioned features.

3.4.3.2 Stimulus/Response Sequences

After we take the input text and the user selects the button of Fill in the blanks, a dialog box pops up asking for the number of questions to be generated. Once the user enters the required number, fill in the blanks type questions are generated for the user.

3.5 Other Nonfunctional Requirements

3.5.1 Performance Requirements

1. The product shall provide questions/output within 10 seconds.
2. The number of quality questions generated will depend on the keywords extracted from the text.

3.5.2 Safety Requirements

Since the product is not storing the text which is used for input, there is no safety requirements at this moment.

3.5.3 Security Requirements

Since the product is not storing the text which is used for input, there is no security requirements at this moment.

3.5.4 Software Quality Attributes

The product is highly adaptive. It can be upgraded by adding new datasets. It creates grammatically correct questions; it has an accuracy of 75%. It is easy to maintain the product. All these factors make the product reliable.

CHAPTER 4

PROJECT SCHEDULING AND PLANNING

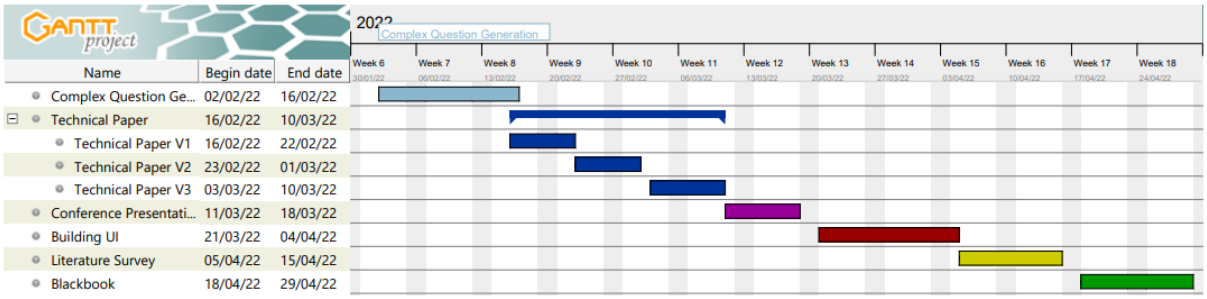


Figure 4.1 Project Scheduling and Planning

CHAPTER 5

PROPOSED SYSTEM

5.1 Dataset Analysis

SQuAD [13] is a very vast dataset that contains 10000+ Wikipedia passages and has questions and answers based on those passages. The dataset is partitioned into 80% Training data, 10% Development Data and 10% Test Data. The questions and answers on the passages are created by crowd workers selected from the USA and Canada. When given a question, humans usually defer in the way they answer the question, hence to match this phenomenon, this dataset has additional answers for each question in the test and development phase. The approach to building this model was to understand how questions and answers are picked from the text given, hence the questions and answers are analyzed separately.

5.1.1 Questions

The dataset contains 98169 questions. This analysis is based on how many words from the question were a part of a sentence and how many of them were a part of the rest of the paragraph. Table 5.1 depicts our analysis, where pickle is used to find out the mean, minimum, maximum containment found in sentences and paragraphs.

Table 5.1 Percentage of similar words found in question to sentences and paragraphs.

	Sentence	**Paragraphs**
Count	98169	98169
Mean	46.3937	58.2157
Std	19.0377	15.9055
Min	00.0000	00.0000
25%	33.3333	33.3333
50%	46.1538	60.0000
75%	60.0000	60.0000
Max	100.0000	00.0000

5.1.2 Answers

The word length of a regular answer is analyzed in the dataset. The words which were not in the question are considered. Table 5.2 depicts the results.

Table 5.2 Word length of the answer

	Answer
Count	98169
Mean	3.355061
Std	3.731794
Min	1.000000
25%	1.000000
50%	2.000000
75%	4.000000
Max	46.000000

The above table shows that the mean length of the answer is 3 and the mode of the length of answers found in the dataset is 1. This depicts that the word count of answers in the SQuAD dataset is very less and most answers are one-word or two-word answers.

Further analysis on the answers provided us with data that the small word counts usually contained only nouns like India, Notre Dame, The Manchester Guardian, etc. From this analysis, it is concluded that for our Multiple-Choice Questions the dataset needs to find distractors for nouns effectively. The text is further analyzed to understand how the dataset created questions by performing the count of Named Entity Recognition tokens and stop words in the passages and questions.

Table 5.3 shows the results of Named Entity Recognition, where PERSON and CARDINAL gives the highest results.

Table 5.3 Analysis of named entity recognition tagging

TYPE	**COUNT**
PERSON	1058
CARDINAL	991
DATE	930
ORG	464
GPE	297
PERCENT	151
MONEY	89

5.2 Algorithms

5.2.1 Complex Questions

Fig. 5.1 describes the algorithm followed by the model to generate complex questions. The input goes through sentence tokenization. Here the input text gets divided into sentences. Each sentence is then searched for discourse tokens to be categorized into Discourse Based, Non-Discourse Based or NER tokens.

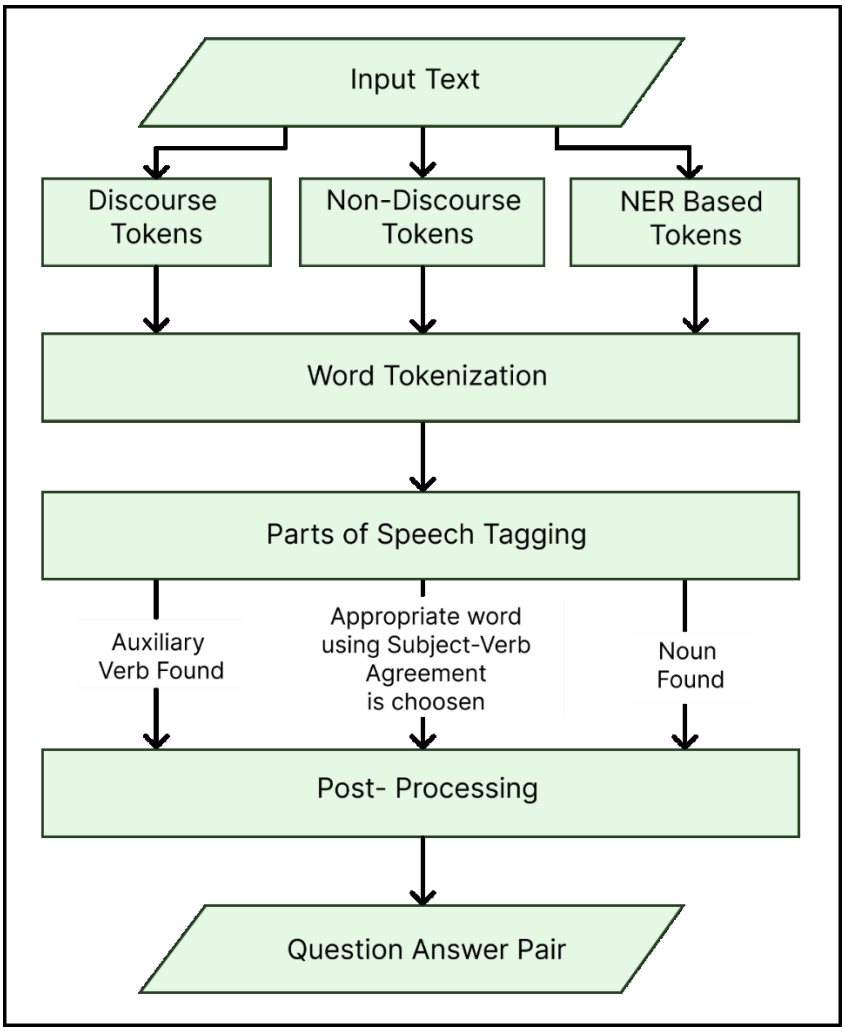


Figure 5.1 Algorithm to generate Complex Questions

5.2.2 Fill in the Blanks and Multiple-Choice Questions

Fig. 5.2 describes the algorithm followed by the model to generate multiple choice questions and fill in the blank type of questions. This algorithm follows similar steps as that of generation of Complex Questions.

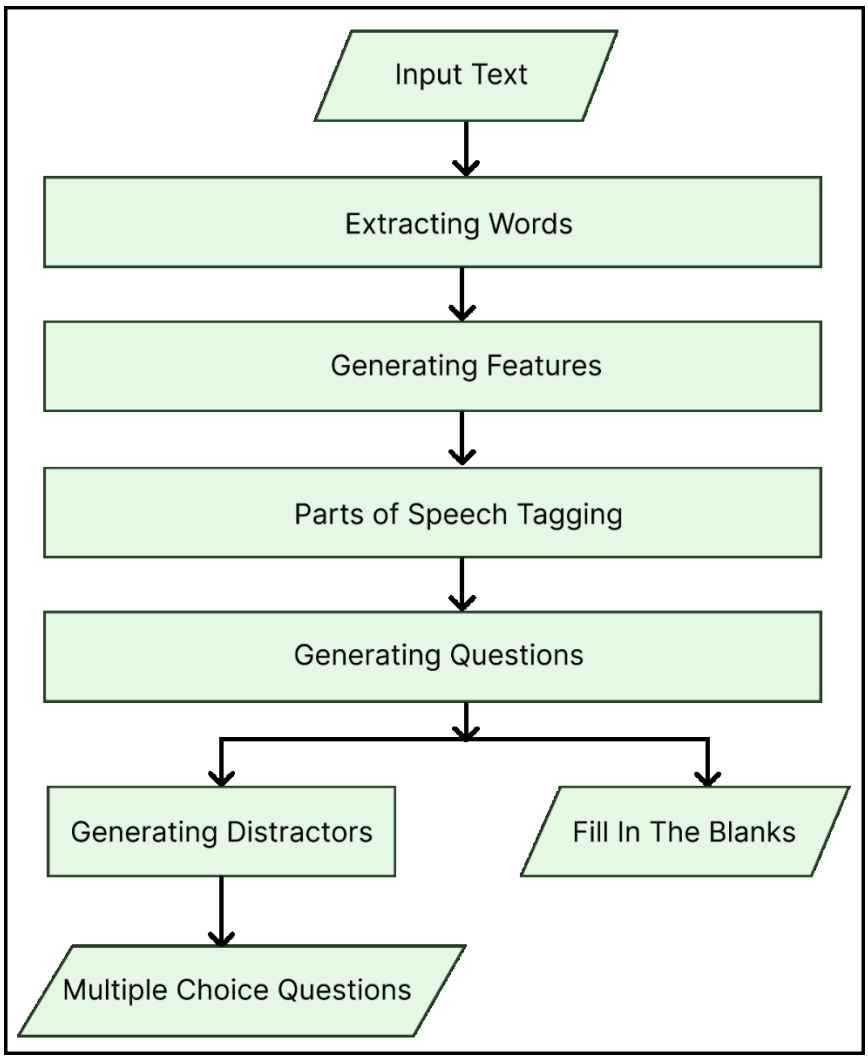


Figure 5.2 Algorithm to generate Multiple Choice Questions and Fill in the blanks questions.

5.3 Details of Hardware & Software

5.3.1 Software Used

5.3.1.1 Jupyter Notebook [14]

The Jupyter Notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience. Jupyter supports over 40 programming languages, including Python, R, Julia, and Scala. Notebooks can be shared with others using email, Dropbox, GitHub and the Jupyter Notebook Viewer. Jupyter Notebooks are an open document format based on JSON. They contain a complete record of the user's sessions and include code, narrative text, equations, and rich output. The Notebook communicates with computational Kernels using the Interactive Computing Protocol, an open network protocol based on JSON data over ZMQ, and WebSockets. Kernels are processes that run interactive code in a particular programming language and return output to the user. Kernels also respond to tab completion and introspection requests.

Figure 5.3 Logo of Jupyter Notebook

5.3.1.2 Python [15]

Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.

Python is dynamically-typed and garbage-collected. Python is meant to be an easily readable language. Its formatting is visually uncluttered, and often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are allowed but rarely used. It has fewer syntactic exceptions and special cases than C or Pascal.

Natural language processing (NLP) is a field that focuses on making natural human language usable by computer programs. Many Python packages make it easier to implement NLP.

Figure 5.4 Logo of Python Language

5.3.1.3 SpaCy [16]

spaCy is an open-source software library for advanced natural language processing, written in the programming languages Python and Cython. The library is published under the MIT license and its main developers are Matthew Honnibal and Ines Montani, the founders of the software company Explosion. paCy v3.0 introduces a comprehensive and extensible system for configuring your training runs. Your configuration file will describe every detail of your training run, with no hidden defaults, making it easy to rerun your experiments and track changes.

Figure 5.5 Logo of spaCy library

5.3.1.4 nltk [17]

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

Thanks to a hands-on guide introducing programming fundamentals alongside topics in computational linguistics, plus comprehensive API documentation, NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, community-driven project. NLTK has been called “a wonderful tool for teaching, and working in, computational linguistics using Python,” and “an amazing library to play with natural language.”

5.3.2 Hardware Requirements:

- Processor: Minimum 1GHz; Recommended 2GHz or more
- Ethernet connection (LAN) OR wireless adapter (Wi-Fi)
- Hard Drive: Minimum 32 GB; Recommended 64 GB or more
- Memory (RAM): Minimum 1 GB; Recommended 4 GB or above.

5.4 Design Details

The user interface consisted of 3 buttons to carry out the main functionalities of our model i.e generation of Complex Questions, Multiple Choice Questions and Fill in the Blanks type questions. The user enters the text in the blank white region and selects any one button for the desired output.

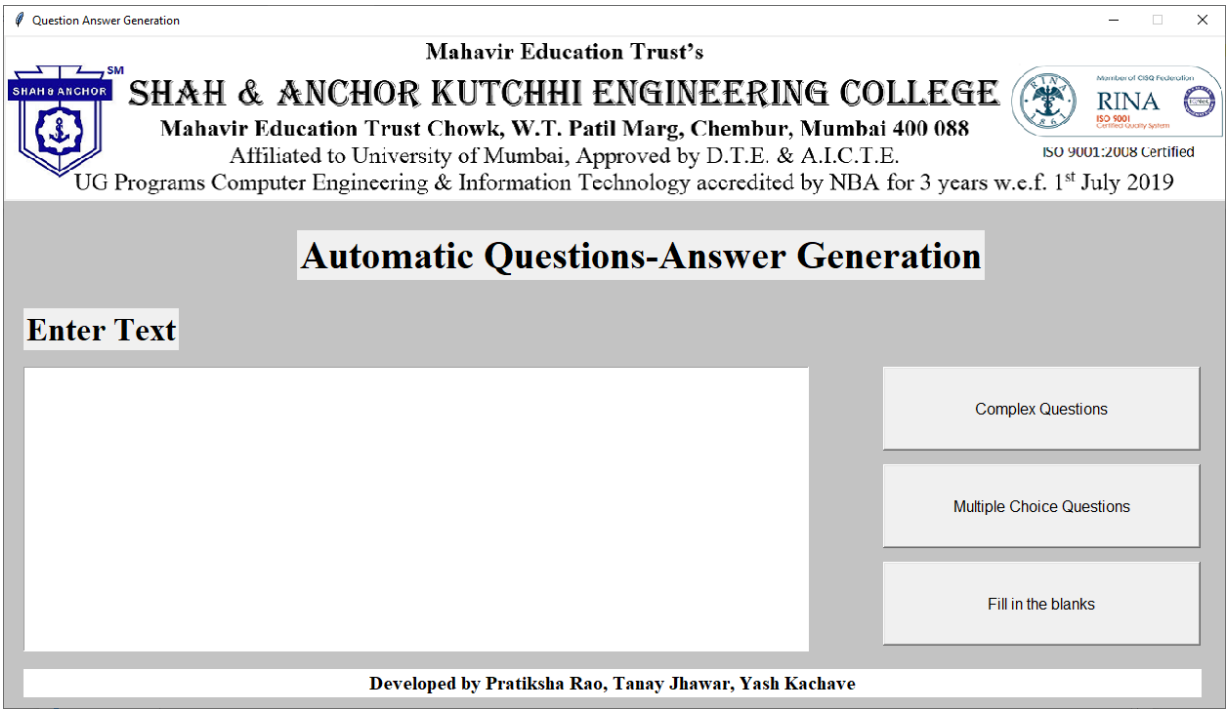


Figure 5.6 User Interface for Taking Input

CHAPTER 6

IMPLEMENTATION DETAILS

6.1 Module and Description

The main goal of our model was to solve the drawbacks faced by the existing models which were mentioned in the 'Related Works' section. Our model solves the complexity, diversity, and accuracy issues faced by the existing models. This model follows a rule-based methodology. It is a pipeline of algorithms of Complex questions, Multiple Choice Questions and Fill in the blank questions.

6.1.1 Complex Questions

Complex Question generation is divided into 3 types:

6.1.1.1 Discourse Based & Non-Discourse Based:

The sentences containing the discourse tokens, such as 'although', 'as a result, 'because', 'for example' and 'since' are called Discourse Based Tokens. The sentences that do not contain the discourse tokens are known as Non- Discourse Based. These sentences are taken through a rule- based algorithm:

a) Pre-Processing: In the word tokenization stage, the 'nltk' or 'spaCy' [14] library can be used to obtain direct and simple results. The sentences are first tokenized into different sentences following the rule of separation by a full stop. These sentences are then further tokenized into words following the rule of separation caused by space. The tokenized words are then taken through parts of speech tagging to identify the roles of each word in the sentence and the auxiliary verb is found. In Non-Discourse based sentences, the first word is always chosen as the appropriate auxiliary verb. If an auxiliary verb is not found in discourse-based

sentences, then an appropriate word according to the subject-verb agreement is chosen.

b) Post-Processing: The sentence up to the auxiliary verb or the chosen word is selected. Sentences are rearranged according to the tags given by parts of speech tagging and a question is formed. The model is unique as it has a pre-defined dictionary that maps the auxiliary word to 'wh'-word to generate the semantically correct questions.

6.1.1.2 NER Based Tokens

The sentences which are declarative and assertive are categorized into this group. The sentence goes through word tokenization and then parts of speech tagging occur to find the category of each word in the sentence. The wh-word associated with the noun, pronoun, adverb, or the determiner is chosen. The wh-word replaces the suitable parts of speech to form the sentence. If the proper replacement is not found, then the steps followed by the non-disclosure sentences take place. That is pre-processing to find an appropriate auxiliary verb and post-processing to form the question.

6.1.1.3 Snapshots of Complex Questions

The figure below shows all three types of output.

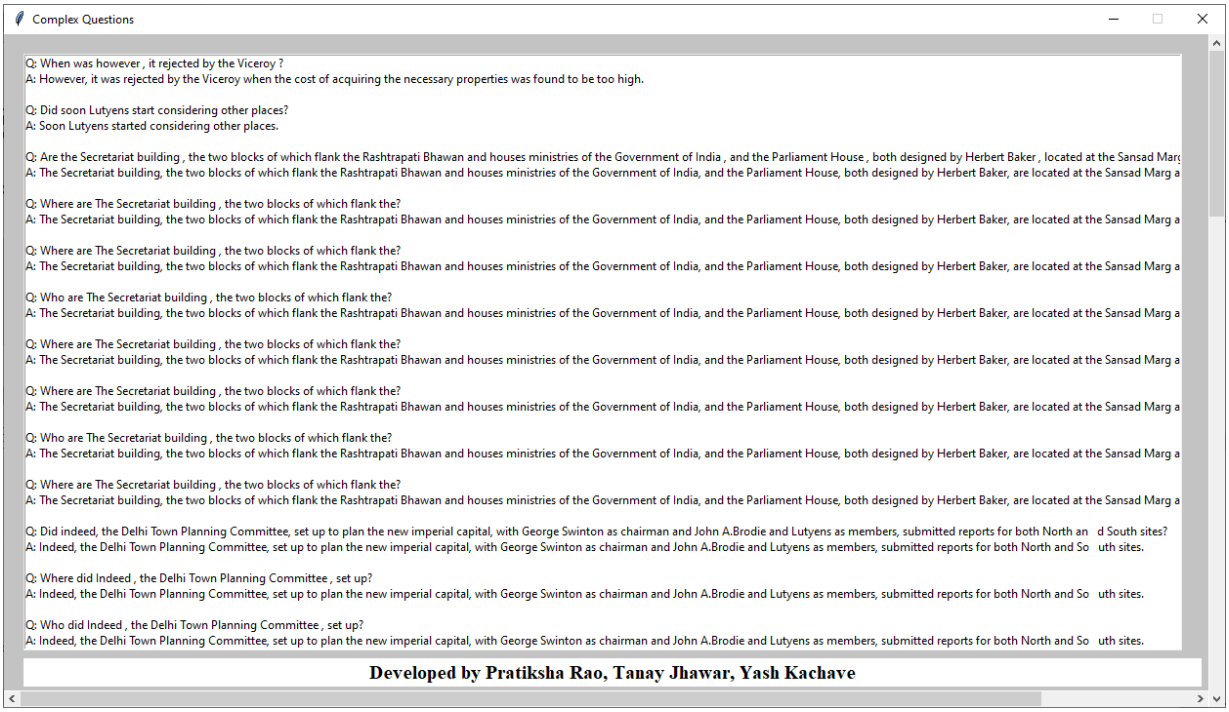


Figure 6.1 Output page for Complex Questions

6.1.2 Fill in The Blanks and Multiple-Choice Questions

Fig 6.2 depicts the algorithm followed by the model to generate Multiple Choice Questions and Fill in the blank type of questions. It follows an approach similar to generating complex questions.

a) Extracting Words: The input is taken from the user, and then the process of word tokenization is applied to the text.

b) Generating Features: Each word is passed through parts of speech tagging to generate features. Every word gets a rating depending on its importance and uses in the sentence.

c) Extracting Keywords: After features are generated, all the ratings of the features are ranked in descending order. The highest-ranked words are known as keywords. These words are passed through a question generation function to generate the outputs

d) Generating Questions: Questions are generated by placing a blank on the keyword. For Multiple Choice Questions, the keywords are passed to generate distractors. The list of questions is directly passed on for Fill In The Blank Questions with keywords passed on as the answers.

e) Generating Distractors: The vector model is used to generate the distractors using the GLoVe dataset. The model works by giving every word a vector, with words having similar meanings or words which come under the same category are clustered into being neighbors. The neighbors are picked accordingly to form the distractors for the Multiple-Choice type questions.

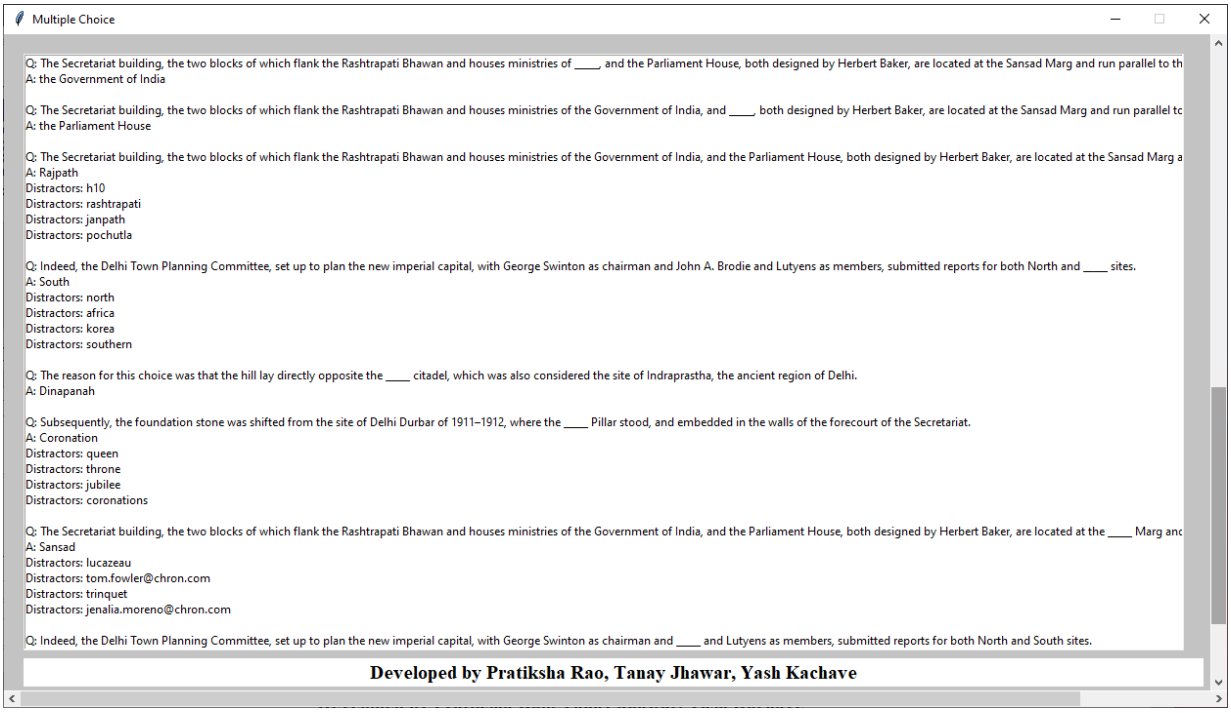


Figure 6.2 Output page for Multiple Choice Questions

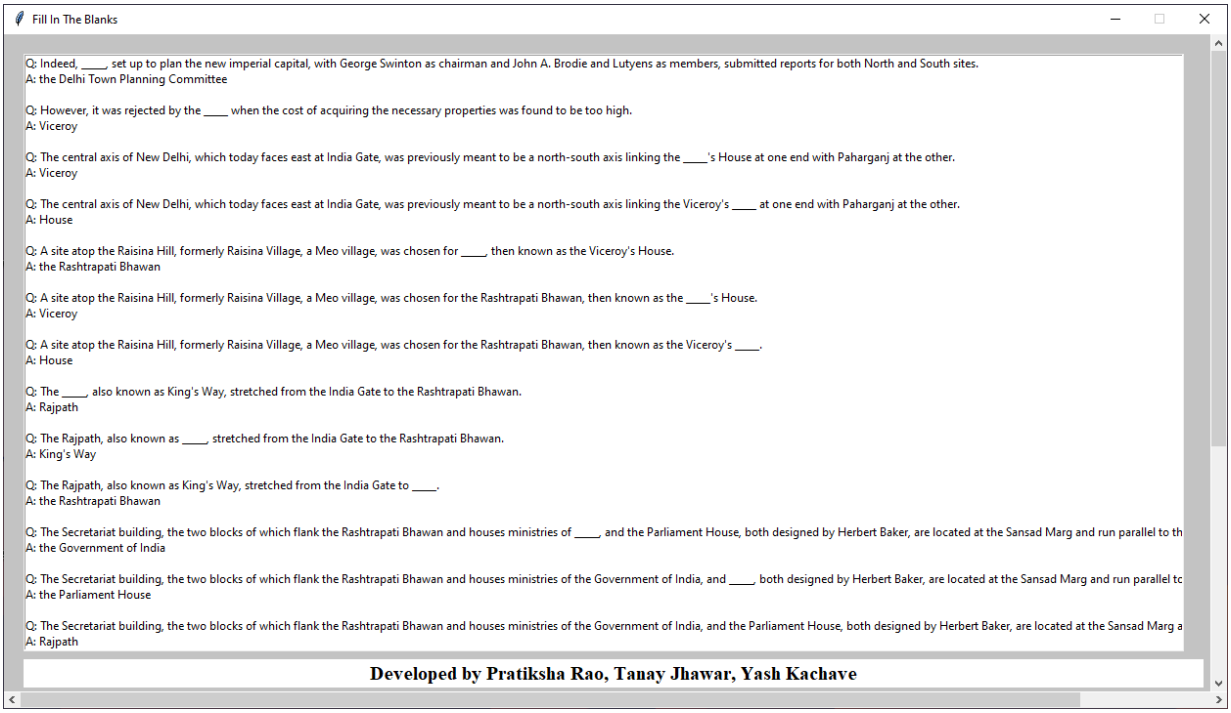


Figure 6.3 Output page for Fill in the Blanks Questions

CHAPTER 7

TESTING

Table 7.1 Test Cases

Test Case ID	Objective	Steps/ Descriptions	Input	Expected Output	Actual Output	Result	Remark
1	Taking Input	Write in text box	Manchester United is a football club. They play in PL.	Manchester United is a football club. They play in PL.	Manchester United is a football club. They play in PL.	Input text accepted by the system.	Working
2	Complex Question button	Click on Complex Question button	Click	System asks for enter number of questions	System asks for enter number of questions	Button function working	Working
3	Multiple Choice Question button	Click on Multiple Choice Question button	Click	System asks for enter number of questions	System asks for enter number of questions	Button function working	Working
4	Fill In The Blanks button	Click on Fill In The Blanks button	Click	System asks for enter number of questions	System asks for enter number of questions	Button function working	Working

5	Enter Number of Questions	Entering a number	2	What is Manchester United? Where do they play?	What is Manchester United? Where do they play?	System generates asked number of questions	Working
6	Error at entering a negative number of questions	Enter a negative number	-2	Error	Error	System shows error for entering negative number of questions	Working
7	Execute Complex Questions	Enter text and click on Complex Question button	Manchester United is a football club. They play in PL.	What is Manchester United? Where do they play?	What is Manchester United? Where do they play?	Complex Questions generated	Working
8	Execute Multiple Choice Questions	Enter text and click on Multiple Choice Questions	Manchester United is a football club. They play in PL.	Manchester United is a _____ -Club -Football -City -Soccer	Manchester United is a _____ -Club -Football -City -Soccer	Multiple Choice Questions generated	Working
9	Execute Fill In The Blanks	Enter text and click on fill in the blanks	Manchester United is a football club. They play in PL.	Manchester United is a _____	Manchester United is a _____	Fill In The Blanks generated	Working

CHAPTER 8

RESULT AND ANALYSIS

8.1 Results

The model generates 3 types of questions based on the above-mentioned algorithms as Complex Questions, Multiple Choice Questions and Fill in the blanks.

Fig. 8.1 is an example of a discourse token-type question. Here the model has identified 'For example' Token and generated the question accordingly. Similarly, the model generates questions for Discourse tokens that have been defined.

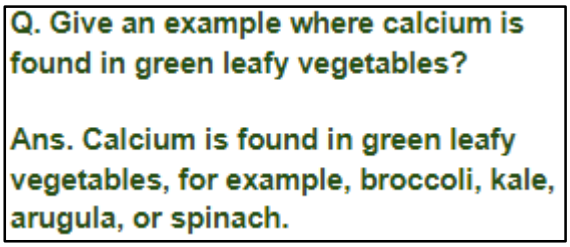


Figure 8.1 Example of discourse token type question

Fig. 8.2 is an example of Non-Discourse based questions. Sentences that are not categorized in Discourse Tokens or Named Entity recognition and generally labelled as Non-Discourse Token type questions.

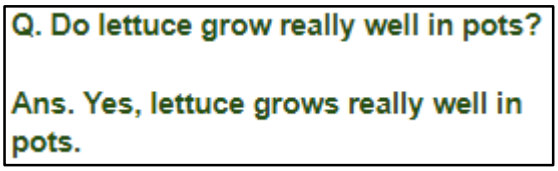


Figure 8.2 Example of discourse token type questions

Named Entity recognition-based questions are declarative or assertive types of questions. Questions generated from this type of token are categorized as WH questions such as who, when, where, what, why, which, whose, and how. In fig. 8.3 the model has identified a declarative sentence and used 'what' to seek information from the text. Similarly, other Wh-type questions can be generated using this model.

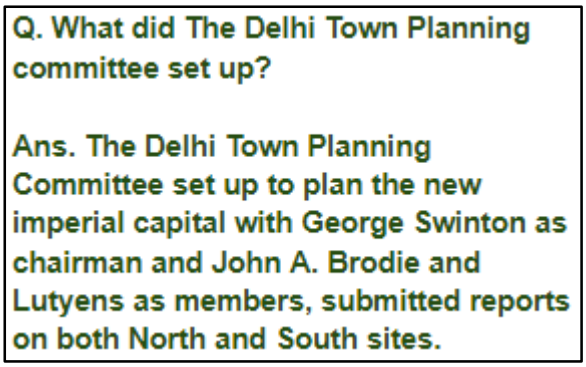


Figure 8.3 Example of a Named Entity Based type of Question

In Fig. 8.4, the Model finds South to be a keyword, and distractors are generated for the keyword using the GLoVe model [8].

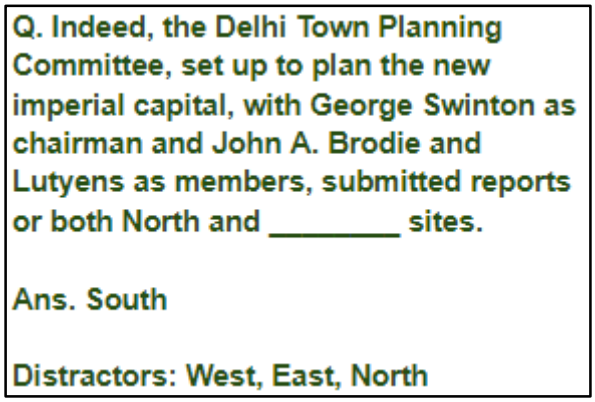


Figure 8.4 Example of Multiple-Choice Question

In Fig. 8.5, The Model finds Rajpath to be the keyword in the sentence and creates a Fill In The Blank Question by replacing the keyword with a blank.

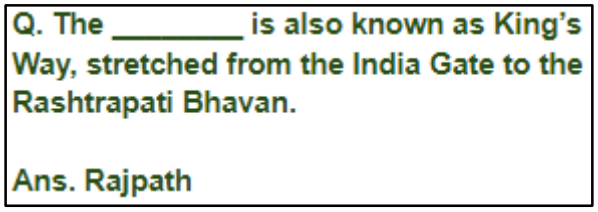


Figure 8.5 Example of Fill in the Blanks type question.

8.2 Analysis

Fig. 8.6 depicts the syntactical correctness of the model. It has been evaluated manually by providing it with a variety of blocks of texts from Wikipedia. Out of the evaluated question and answer pairs, the model produced 79.2% of syntactic correctness.

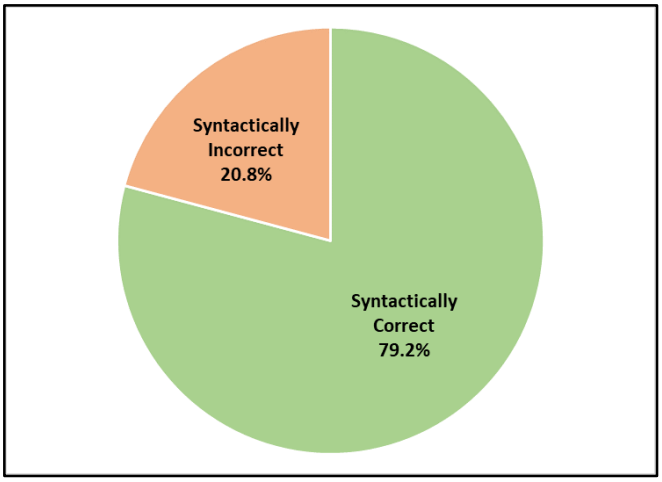


Figure 8.6 Syntactic Evaluation of the model

Table 8.1 depicts the efficiency of our GLoVe model [8] in the generation of quality distractors. There are 6 examples in the table and the model can have an overall efficacy of 85%, by successfully creating 34 out of the 40 options.

Table 8.1 Distractor evaluation

Keyword	***Distractors***	***Efficiency***
South	North, West, East, South-East	100%
University	Professor, College, School, Pre-School	75%
Gandhi	Nehru, Bose, Patel, Dandi	75%
Car	Train, Airplane, Mercedes, Jaguar	50%
Stack	Push, Pop, List, Books	75%
Oxygen	Helium, Hydrogen, Carbon, Neon	100%
India	Pakistan, China, America, Rupees	75%
One	Two, Three, Four, Five	100%

CHAPTER 9

CONCLUSION AND FUTURE SCOPE

We were successful in generation of assortment of questions of the type complex question, multiple-choice question and fill in the blanks. By recognizing the named entities in the supplied input text and making use of the algorithms mentioned in the report, the model was quite effective at producing quality questions. The model has showed remarkable promise and maybe improved significantly by focusing on future scopes and creating a pleasant user interface. This is the first model to divide the questions in three different types of tokens. This overcame the drawback of only getting simple questions and provided the output with more diversification.

The syntactic correctness of the Questions can be improved by working with the post-processing algorithms. The efficiency of getting quality distractors can be enhanced by increasing the variety of tuples in the dataset. The model can be further pipelined with gamification applications to create interactive quizzes.

REFERENCES

[1] S. Gangopadhyay and S. M. Ravikiran, "Focused Questions and Answer Generation by Key Content Selection," 2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM), 2020, pp. 45-53, doi: 10.1109/BigMM50055.2020.00017.

[2] A. Srivastava, S. Shinde, N. Patel, S. Despande, A. Dalvi and S. Tripathi, "Questionator-Automated Question Generation using Deep Learning," 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), 2020, pp. 1-5, doi: 10.1109/ic-ETITE47903.2020.212

[3] A. Killawala, I. Khokhlov and L. Reznik, "Computational Intelligence Framework for Automatic Quiz Question Generation," 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2018, pp. 1-8, doi: 10.1109/FUZZ-IEEE.2018.8491624.

[4] T. Raynaud, J. Subercaze and F. Laforest, "Thematic Question Generation over Knowledge Bases," 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), 2018, pp. 1-8, doi: 10.1109/WI.2018.0-114.

[5] C. -H. Lee, T. -Y. Chen, L. -P. Chen, P. -C. Yang and R. T. -H. Tsai, "Automatic Question Generation from Children's Stories for Companion Chatbot," 2018 IEEE International Conference on Information Reuse and Integration (IRI), 2018, pp. 491-494, doi: 10.1109/IRI.2018.00078.

[6] J. Cho, M. Seo, and H. Hajishirzi, "Mixture content selection for diverse sequence generation," arXiv preprint arXiv:1909.01953, 2019.

[7] Q. Zhou, N. Yang, F. Wei, C. Tan, H. Bao, and M. Zhou, "Neural question generation from text: A preliminary study," in National CCF Conference on Natural Language Processing and Chinese Computing. Springer, 2017, pp. 662–671.

[8] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical*

[9] *Methods in Natural Language Processing (EMNLP)*, pages 1532-1543, Doha, Qata. Association for computational Linguitics.

[10] *Olga* Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 115(3):211–252, 2015.

[11] M. Sundermeyer, H. Ney and R. Schlüter, "From Feedforward to Recurrent LSTM Neural Networks for Language Modeling," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 517-529, March 2015, doi: 10.1109/TASLP.2015.2400218

[12] M. Agarwal, R. Shah, and P. Mannem, "Automatic question generation using discourse cues," in Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications. Association for Computational Linguistics, 2011, pp. 1–9.

[13] A. Abujabal, M. Yahya, M. Riedewald, and G. Weikum, "Automated template generation for question answering over knowledge graphs," in Proc. 26th int. conf. on world wide web, 2017, pp. 1191–1200.

[14] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

[15] https://jupyter.org/

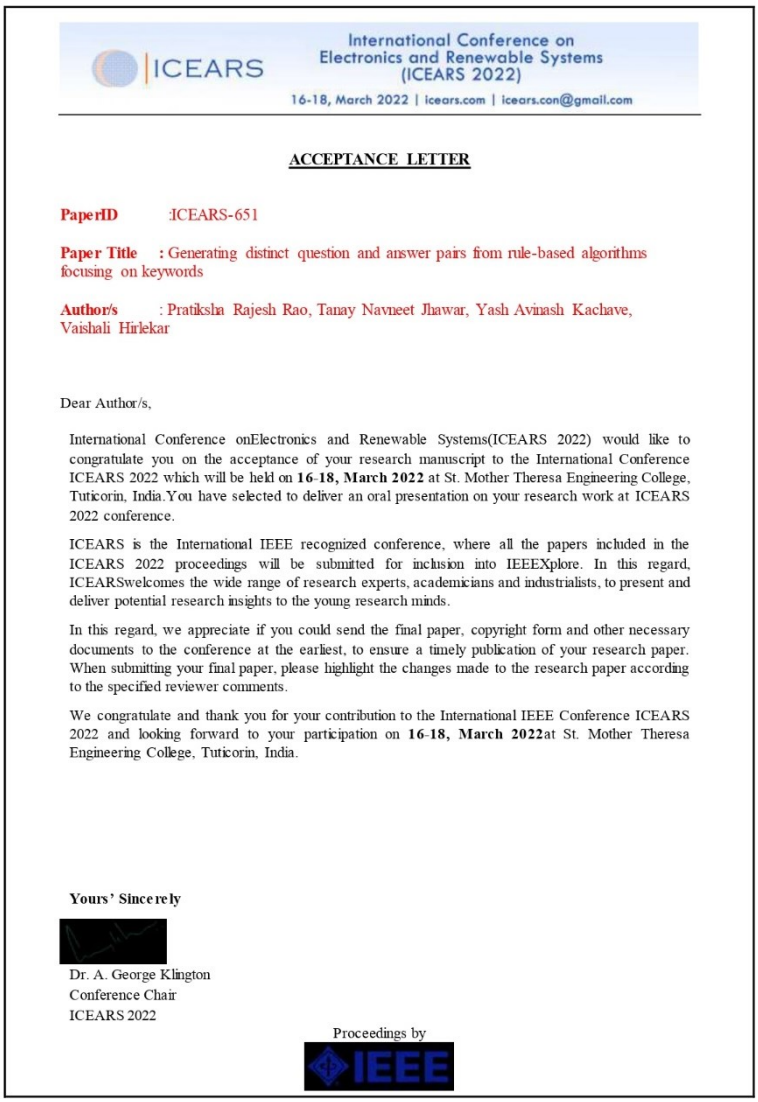
[16] https://en.wikipedia.org/wiki/Python_(programming_language)

[17] https://en.wikipedia/wiki/SpaCy

APPENDIX

International Conference on Electronics and Renewable Systems (ICEARS 2022)

1. ACCEPTANCE LETTER



ICEARS

International Conference on Electronics and Renewable Systems (ICEARS 2022)

16-18, March 2022 | icears.com | icears.con@gmail.com

ACCEPTANCE LETTER

PaperID :ICEARS-651

Paper Title : Generating distinct question and answer pairs from rule-based algorithms focusing on keywords

Author/s : Pratiksha Rajesh Rao, Tanay Navneet Jhawar, Yash Avinash Kachave, Vaishali Hirlekar

Dear Author/s,

International Conference onElectronics and Renewable Systems(ICEARS 2022) would like to congratulate you on the acceptance of your research manuscript to the International Conference ICEARS 2022 which will be held on **16-18, March 2022** at St. Mother Theresa Engineering College, Tuticorin, India.You have selected to deliver an oral presentation on your research work at ICEARS 2022 conference.

ICEARS is the International IEEE recognized conference, where all the papers included in the ICEARS 2022 proceedings will be submitted for inclusion into IEEEXplore. In this regard, ICEARSwelcomes the wide range of research experts, academicians and industrialists, to present and deliver potential research insights to the young research minds.

In this regard, we appreciate if you could send the final paper, copyright form and other necessary documents to the conference at the earliest, to ensure a timely publication of your research paper. When submitting your final paper, please highlight the changes made to the research paper according to the specified reviewer comments.

We congratulate and thank you for your contribution to the International IEEE Conference ICEARS 2022 and looking forward to your participation on **16-18, March 2022**at St. Mother Theresa Engineering College, Tuticorin, India.

Yours' Sincerely

Dr. A. George Klington
Conference Chair
ICEARS 2022

Proceedings by IEEE

2. PAPER

Proceedings of the International Conference on Electronics and Renewable Systems (ICEARS 2022)
IEEE Xplore Part Number: CFP22AV8-ART; ISBN: 978-1-6654-8425-1

Generating QA from Rule-based Algorithms

Pratiksha Rajesh Rao
Dept. of Computer Engineering
Shah and Anchor Kutchhi Engineering College
Mumbai, India
pratiksha.rao@sakec.ac.in

Tanay Navneet Jhawar
Dept. of Computer Engineering
Shah and Anchor Kutchhi Engineering College
Mumbai, India
tanay.jhawar@sakec.ac.in

Yash Avinash Kachave
Dept. of Computer Engineering
Shah and Anchor Kutchhi Engineering College
Mumbai, India
yash.kachave@sakec.ac.in

Vaishali Hirlekar
Dept. of Computer Engineering
Shah and Anchor Kutchhi Engineering College
Mumbai, India
vaishali.hirlekar@sakec.ac.in

2022 International Conference on Electronics and Renewable Systems (ICEARS) | 978-1-6654-8425-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICEARS53579.2022.9751723

***Abstract*— In the education industry answering questions is used as a common parameter to judge one's understanding of a topic. Taking quizzes on a regular basis helps an individual feel confident and it also helps the professor assess the student's understanding on a particular topic. Generating question and answer pairs is a time-consuming task. To solve this problem, this paper discusses methods to generate automatic Natural Language Processing models which creates diverse types of question-answer pairs. The model takes an input in the form of text in the English language and produces output as Complex Questions, Multiple Choice Questions with relevant distractors, and Fill in the Blanks type of questions. To generate Complex Questions a Rule-Based Algorithm is used. To generate Multiple Choice Questions and Fill in the Blanks type questions, a Vector Algorithm from the GLoVe Model is used along with Rule-Based Algorithms. This paper also includes a detailed explanation of the analysis of the pattern and rules that are observed in the question-making process. SQuAD dataset is used for this analysis and used the same dataset to train the model. The implementation process of this model focused on generating diverse questions with higher syntactic correctness than the existing models. The approach mentioned in this paper can be used in the fields of education, entertainment, generation of quizzes, virtual learning assistance and to get a deeper insight into any topic.**

Keywords—Automatic question generation, complex question, discourse tokens, education, fill in the blanks, multiple-choice question, natural language processing, quiz, rule-based, text analysis

I. INTRODUCTION

Answering questions about a given topic is an efficient way to assess one's knowledge. It is challenging and time-consuming to come up with new sets of questions on a regular basis. A question-answer system can be used in many forms in the education industry. It can be used by teachers to conduct quizzes, make flashcards, judge how much a student has understood a particular topic, or take interactive sessions in a lecture. Due to the pandemic, teachers find it difficult to understand if the student has truly grabbed the information that has been taught. Hence taking daily quizzes makes it easier to assess the students and to build a student-teacher relationship on a deeper level. Generating different sets of questions is a difficult and time-consuming task. This paper focuses on solving the problem of generating a variety of questions when a knowledge base is provided by the user. Automating the process of question-answer generation will save time and energy for the person trying to form adequate questions. This model will help the user generate diverse types of questions by providing any type of text as the input. The input can contain text blocks of articles taken from sources such as any web page or a book. This methodology is also beneficial for users to create quality questions without having information about the input text in just one click. This helps reduce the cost of having to hire subject experts. The approach mentioned will assist in the generation of a variety of question and answer pairs that may be utilized in the fields of education, to take interactive sessions, to revise or learn a topic and more.

The model mentioned in this paper uses Natural Language Processing, that is, it uses the technique of teaching the machine to understand the human language and generate questions in human language with correct grammar. There have been a few attempts in making a question-answer model which is mentioned in this paper but the major drawback of each model was the accuracy of the questions which were generated and the inefficacy in the generation of complex questions. The accuracy depends on the quality of the questions generated, how relevant the question is, and how grammatically correct it is. This paper proposes a model keeping the above metrics in mind. It also explains the approach which has been taken to analyse questions made in Stanford Question Answering Dataset (SQuAD) [13]. The model consists of three types of questions: Complex Questions, Multiple Choice Questions, and Fill In The Blank. This work intends to use a Rule-Based Algorithm to solve the Question Answering Model Problem.

II. RELATED WORKS

There have been many approaches that have been taken in the past to solve the issues faced by question-answer generating models. Below are some of the work in this field that has been summarized.

A. Focused Questions and Answer Generation by Key Content Selection [1]

The model mentioned in this paper uses a module called focus generator. This module guides the decoder in an existing "encoder-decoder" model to generate questions based on selected focus contents [1]. The input text is divided into three focus sequences based on the Neural Entity Selection (NES) algorithm. The module focuses on these sequences to generate questions. It generates a range of complexities in the question by combining focus sequences according to the need. This model was highly inspired by the baseline model by Cho et al [6]. The extracted training, development and test sets contain 86,635, 8,965 and 8,964 triples respectively [7], on the SQuAD dataset. It automatically generates Answer tags without manual intervention using the NES algorithm. The drawback of this

model was its inability to produce diverse questions and this can be overcome using modern libraries and architectures of Natural Language Processing.

B. *Questionator -Automated Question Generation using Deep Learning [2]*

The model mentioned in this paper is based on Natural Language Processing techniques and Image Captioning techniques. The model used ILSVRC-2012-CLS Image classification dataset [9]. The image captioning module converts a given input image into a natural language description [2]. Later this image was converted to a question. They have used Convolutional Neural Network and Long Short-Term Memory (LSTM) as encoders and decoders algorithms. The drawback of this model was that it was incompetent to create questions if the database did not have a similar image saved. Since every image had to be captioned it is difficult to get or create a diverse database. Since similar images can lead to a misleading question, the model needs to be trained with an advanced dataset.

C. *Computational Intelligence Framework for Automatic Quiz Question Generation [3]*

This framework is based on information. In this paper, for "WH"-questions generation, pattern matching is used to a certain extent [3]. They used the proposed methodology from Agarwal et al. [11] to generate MCQ questions. They made use of the LSTM neural network layer to train their model. Even though shallow LSTM networks already perform better than shallow feedforward networks in language modeling, LSTMs still show more consistent improvements when adding additional hidden layers than their feedforward counterparts [10]. They made use of Named Entity Recognition (NER) and Super-sense tagging (SST) to create WH-questions. The major drawback of this model was to get adequate pronouns in the formation of the questions and answers, this limitation was due to the use of the LSTM approach.

D. *Thematic Question Generation over Knowledge Bases [4]*

This model follows a template-based approach for the generation of Multiple Choice Questions with distractors. The model also follows the approach of regenerating question answering templates automatically by importing the used templates, hence more than 2000 templates can be imported at a time. They have referred to the Wikipedia structure to identify and establish the limits of the topic. They have used and updated existing question answering templates by [12]. To choose the distractors correctly, they made use of the Page Rank score. Learning to rank is a relatively new research area, which received increasing attention in both the Information Retrieval and Machine Learning research communities, during the past decade [15]. The distractors with a higher score are selected and this method helps to filter ineffective distractors. Their approach was experimented on a dataset with 1430 questions and was able to generate 1394 questions successfully. The drawback of the model was that it was not able to include literal in the question generation process.

E. *Automatic Question Generation from Children's Stories for Companion Chatbot [5]*

The previous work in Chinese question generation relied solely on rule based approach. It generated an unsatisfactory performance due to the complexity in language and limitation of parser. The proposed model generates automatic questions in the Chinese language for children story book. In Chinese, generating questions can usually replace target answer phrases with interrogatives [5]. This method ensures that the quality of the question is maintained. They also use a ranking model to remove lower-quality questions. After evaluation, it was concluded that 50% of the questions generated were correct. Who-type question was not generated correctly due to lack of semantic understanding.

The major drawbacks of the existing models were that the models were not able to get a higher accuracy at generating question-answer pairs and could not handle complex syntactic questions. If the Question Answer generating models parser creates flawed results, the questions would be grammatically incorrect. Some distractors were deficient and of substandard quality. The distractors dataset was not diverse hence could not create diversity to choose.

III. Dataset Analysis

SQuAD [13] is a very vast dataset that contains 10000+ Wikipedia passages and has questions and answers based on those passages. The dataset is partitioned into 80% Training data, 10% Development Data and 10% Test Data. The questions and answers on the passages are created by crowd workers selected from the USA and Canada. When given a question, humans usually defer in the way they answer the question, hence to match this phenomenon, this dataset has additional answers for each question in the test and development phase. The approach to building this model was to understand how questions and answers are picked from the text given, hence the questions and answers are analyzed separately.

1) *Questions*

The dataset contains 98169 questions. This analysis is based on how many words from the question were a part of a sentence and how many of them were a part of the rest of the paragraph. Table 1 depicts our analysis, where pickle is used to find out the mean, minimum, maximum containment found in sentences and paragraphs.

TABLE 1. PERCENTAGE OF SIMILAR WORDS FOUND IN QUESTIONS WITH RESPECT TO SENTENCES AND PARAGRAPHS.

	Sentence	Paragraphs
Count	98169	98169
Mean	46.3937	58.2157
Std	19.0377	15.9055
Min	00.0000	00.0000
25%	33.3333	33.3333
50%	46.1538	60.0000

	Sentence	Paragraphs
75%	60.0000	60.0000
Max	100.0000	00.0000

It can be concluded that the majority of the questions contain similar words from the paragraphs than from a single sentence.

A further analysis is run on these questions and concluded that some questions used synonyms instead of the words mentioned in the sentence. Some sentences changed the tense when changing into question.

2) Answers

The word length of a regular answer is analyzed in the dataset. The words which were not in the question are considered. Table 2 depicts the results.

TABLE 2. WORD LENGTH OF THE ANSWER

	Answer
Count	98169
Mean	3.355061
Std	3.731794
min	1.000000
25%	1.000000
50%	2.000000
75%	4.000000
Max	46.000000

The above table shows that the mean length of the answer is 3 and the mode of the length of answers found in the dataset is 1. This depicts that the word count of answers in the SQuAD dataset is very less and most answers are one-word or two-word answers.

Further analysis on the answers provided us with data that the small word counts usually contained only nouns like India, Notre Dame, The Manchester Guardian, etc. From this analysis, it is concludeded that for our Multiple-Choice Questions the dataset needs to find distractors for nouns effectively.

The text is further analyzed to understand how the dataset created questions by performing the count of NER tokens and stop words in the passages and questions.
Table 3 shows the results of NER, where PERSON and CARDINAL gives the highest results.

TABLE 3. ANALYSIS OF NAMED ENTITY RECOGNITION TAGGING

TYPE	COUNT
PERSON	1058
CARDINAL	991
DATE	930
ORG	464
GPE	297
PERCENT	151
MONEY	89
NORP	68
ORDINAL	37
FAC	32
QUANTITY	32
LOC	30
EVENT	9
TIME	7
LAW	5
	93969

Similarly, Parts of Speech (POS) Tagging of the answers is analyzed to understand more about which words to select as the keyword.

TABLE 4. ANALYSIS OF PARTS OF SPEECH (POS) TAGGING

TYPE	COUNT
PROPN	2689
NUM	1705
NOUN	622
ADJ	193
DET	123
SYM	72
VERB	64
ADP	58
X	45
ADV	42
AUX	9
PUNCT	3
INTJ	3
PRON	2
PART	1
	5631

Table 4 concludes that the PROPN, i.e. Proper Noun is the most common answer and this fits our previous analysis of the dataset.

IV. PROPOSED METHODOLOGY

The main goal of our model was to solve the drawbacks faced by the existing models which were mentioned in the 'Related Works' section. Our model solves the complexity, diversity, and accuracy issues faced by the existing models. This model follows a rule-based methodology. It is a pipeline of algorithms of Complex questions, Multiple Choice Questions and Fill in the blank questions.

A. *Complex Questions*

Fig. 1 describes the algorithm followed by the model to generate complex questions. The input goes through sentence tokenization. Here the input text gets divided into sentences. Each sentence is then searched for discourse tokens to be categorized into Discourse Based, Non-Discourse Based or NER tokens.

The model in Fig. 1 is the first model to divide the questions in three different types of tokens. This overcame the drawback of only getting simple questions and provided the output with more diversification.

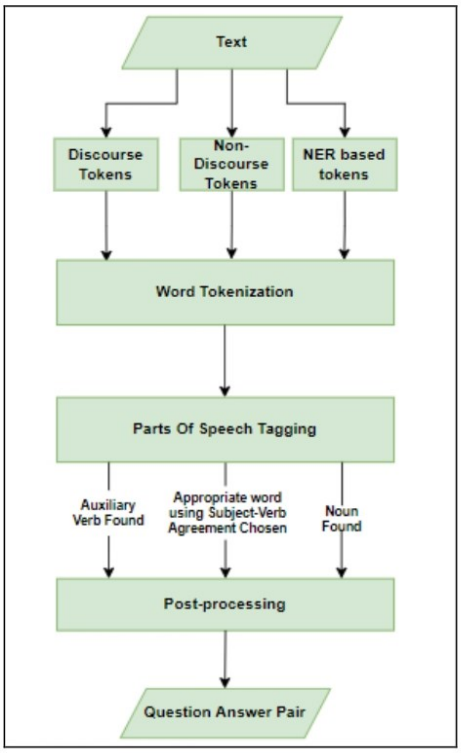


Fig. 1. Algorithm for complex question generation.

1) Discourse Based & Non-Discourse Based :

The sentences containing the discourse tokens, such as 'although', 'as a result, 'because', 'for example' and 'since' are called Discourse Based Tokens. The sentences that do not contain the discourse tokens are known as Non-Discourse Based. These sentences are taken through a rule-based algorithm:

a) Pre-Processing: In the word tokenization stage, the 'nltk' or 'spaCy' [14] library can be used to obtain direct and simple results. The sentences are first tokenized into different sentences following the rule of separation by a full stop. These sentences are then further tokenized into words following the rule of separation caused by space. The tokenized words are then taken through parts of speech tagging to identify the roles of each word in the sentence and the auxiliary verb is found. In Non-Discourse based sentences, the first word is always chosen as the appropriate auxiliary verb. If an auxiliary verb is not found in discourse-based sentences, then an appropriate word according to the subject-verb agreement is chosen.

b) Post-Processing: The sentence up to the auxiliary verb or the chosen word is selected. Sentences are rearranged according to the tags given by parts of speech tagging and a question is formed. The model is unique as it has a pre-defined dictionary that maps the auxiliary word to 'wh'-word to generate the semantically correct questions.

2) NER Based Tokens

The sentences which are declarative and assertive are categorized into this group. The sentence goes through word tokenization and then parts of speech tagging occur to find the category of each word in the sentence. The wh-word associated with the noun, pronoun, adverb, or the determiner is chosen. The wh-word replaces the suitable parts of speech to form the sentence. If the proper replacement is not found, then the steps followed by the non-disclosure sentences take place. That is pre-processing to find an appropriate auxiliary verb and post-processing to form the question.

B. *Fill In The Blanks and Multiple Choice Questions*

Fig. 2 depicts the algorithm followed by the model to generate Multiple Choice Questions and Fill in the blank type of questions. It follows an approach similar to generating complex questions.

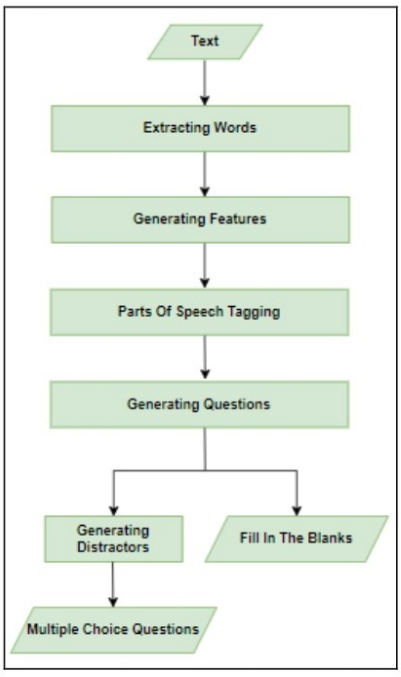


Fig. 2. Algorithm for Multiple Choice Question and Fill in the Blanks Questions.

1) Extracting Words: The input is taken from the user, and then the process of word tokenization is applied to the text.

2) Generating Features: Each word is passed through parts of speech tagging to generate features. Every word gets a rating depending on its importance and uses in the sentence.

3) Extracting Keywords: After features are generated, all the ratings of the features are ranked in descending order. The highest-ranked words are known as keywords. These words are passed through a question generation function to generate the outputs

4) Generating Questions: Questions are generated by placing a blank on the keyword. For Multiple Choice Questions, the keywords are passed to generate distractors. The list of questions is directly passed on for Fill In The Blank Questions with keywords passed on as the answers.

5) Generating Distractors: The vector model is used to generate the distractors using the GLoVe dataset. The model works by giving every word a vector, with words having similar meanings or words which come under the same category are clustered into being neighbors. The neighbors are picked accordingly to form the distractors for the Multiple Choice type questions.

V. RESULTS

The model generates 3 types of questions based on the above-mentioned algorithms as Complex Questions, Multiple Choice Questions and Fill in the blanks.

Fig. 3 is an example of a discourse token-type question. Here the model has identified 'For example' Token and generated the question accordingly. Similarly, the model generates questions for Discourse tokens that have been defined.

Q. Give an example where calcium is found in green leafy vegetables?

Ans. Calcium is found in green leafy vegetables, for example, broccoli, kale, arugula, or spinach.

Fig. 3. Example of discourse token type questions.

Fig. 4 is an example of Non-Discourse based questions. Sentences that are not categorized in Discourse Tokens or Named Entity recognition and generally labeled as Non-Discourse Token type questions.

Q. Do lettuce grow really well in pots?

Ans. Yes, lettuce grows really well in pots.

Fig. 4. Example of discourse token type questions.

Named Entity recognition-based questions are declarative or assertive types of questions. Questions generated from this type of token are categorized as WH questions such as who, when, where, what, why, which, whose, and how. In fig. 5 the model has identified a declarative sentence and used 'what' to seek information from the text. Similarly, other Wh-type questions can be generated using this model.

Q. What did The Delhi Town Planning committee set up?

Ans. The Delhi Town Planning Committee set up to plan the new imperial capital with George Swinton as chairman and John A. Brodie and Lutyens as members, submitted reports on both North and South sites.

Fig. 5. Example of a Named Entity Based type of Question.

In Fig. 6, the Model finds South to be a keyword, and distractors are generated for the keyword using the GLoVe model [8].

Q. Indeed, the Delhi Town Planning Committee, set up to plan the new imperial capital, with George Swinton as chairman and John A. Brodie and Lutyens as members, submitted reports or both North and _____ sites.

Ans. South

Distractors: West, East, North

Fig. 6. Example of Multiple Choice Question.

In Fig. 7, The Model finds Rajpath to be the keyword in the sentence and creates a Fill In The Blank Question by replacing the keyword with a blank.

Q. The _____ is also known as King's Way, stretched from the India Gate to the Rashtrapati Bhavan.

Ans. Rajpath

Fig. 7. Example of Fill in the Blanks type question. caption.

Fig. 8 depicts the syntactical correctness of the model. It has been evaluated manually by providing it with a variety of blocks of texts from Wikipedia. Out of the evaluated question and answer pairs, the model produced 79.2% of syntactic correctness.

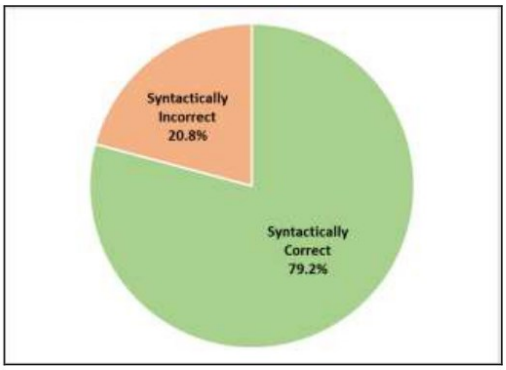


Fig. 8. Syntactic Evaluation of the model

Table 5 depicts the efficiency of our GLoVe model [8] in the generation of quality distractors. There are 6 examples in the table and the model can have an overall efficacy of 85%, by successfully creating 34 out of the 40 options.

TABLE 5. DISTRACTOR EVALUATION

Keyword	*Distractors*	*Efficiency*
South	North, West, East, South-East	100%
University	Professor, College, School, Pre-School	75%
Gandhi	Nehru, Bose, Patel, Dandi	75%
Car	Train, Airplane, Mercedes, Jaguar	50%
Stack	Push, Pop, List, Books	75%
Apple	Mango, Grapes, Chickoo, Peru	100%
Oxygen	Helium, Hydrogen, Carbon, Neon	100%
India	Pakistan, China, America, Rupees	75%
One	Two, Three, Four, Five	100%
Monday	Tuesday, Sunday, Saturday, Wednesday	100%

VI. APPLICATION

The question-answer system has a huge market in the education industry. It can be integrated with a management system to conduct e-learning. It can also be used in traditional classrooms to have interactive sessions with the students. The professors can use the question answering module to generate questions in bulk which can further be used to make different sets of question papers. This model makes the process of answer correction hassle-free for professors as anybody with the answer sheet provided by the model can check the paper. Since the model takes the input as text and creates questions, anybody without the domain expertise of the input text can use the system and create questions on the topic within seconds.

VII. CONCLUSION

The model mentioned in this paper makes the effective question and answer pairs from a text without wasting any time and it takes less effort than the traditional way of making questions. It also creates diverse questions which are easy to understand and are grammatically correct. The model has been inspired by the SQuAD analysis performed. Our model can be used for educational purposes for example to practice a specific chapter by giving quizzes. Students tend to be more interested when there are external activities and game-based teaching involved, hence this application will be very useful in the educational industry. Our model was successful in the generation of an assortment of questions of the type Complex Questions, Multiple Choice Questions, and Fill in The Blanks type of Questions. By recognizing the named entities in the supplied input text and making use of the algorithms mentioned in the paper, the model was quite effective at producing quality questions. The model showed remarkable promise and maybe improved significantly by focusing on future scopes.

VIII. FUTURE SCOPE

The syntactic correctness of the Questions can be improved by working with the post-processing algorithms. The efficiency of getting quality distractors can be enhanced by increasing the variety of tuples in the dataset. The model can be further pipelined with gamification applications to create interactive quizzes.

REFERENCES

[1] S. Gangopadhyay and S. M. Ravikiran, "Focused Questions and Answer Generation by Key Content Selection," 2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM), 2020, pp. 45-53, doi: 10.1109/BigMM50055.2020.00017.

[2] A. Srivastava, S. Shinde, N. Patel, S. Despande, A. Dalvi and S. Tripathi, "Questionator-Automated Question Generation using Deep Learning," 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), 2020, pp. 1-5, doi: 10.1109/ic-ETITE47903.2020.212

[3] A. Killawala, I. Khokhlov and L. Reznik, "Computational Intelligence Framework for Automatic Quiz Question Generation," 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2018, pp. 1-8, doi: 10.1109/FUZZ-IEEE.2018.8491624.

[4] T. Raynaud, J. Subercaze and F. Laforest, "Thematic Question Generation over Knowledge Bases," 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), 2018, pp. 1-8, doi: 10.1109/WI.2018.0-114.

[5] C. -H. Lee, T. -Y. Chen, L. -P. Chen, P. -C. Yang and R. T. -H. Tsai, "Automatic Question Generation from Children's Stories for Companion Chatbot," 2018 IEEE International Conference on Information Reuse and Integration (IRI), 2018, pp. 491-494, doi: 10.1109/IRI.2018.00078.

[6] J. Cho, M. Seo, and H. Hajishirzi, "Mixture content selection for diverse sequence generation," arXiv preprint arXiv:1909.01953, 2019.

[7] Q. Zhou, N. Yang, F. Wei, C. Tan, H. Bao, and M. Zhou, "Neural question generation from text: A preliminary study," in National CCF Conference on Natural Language Processing and Chinese Computing. Springer, 2017, pp. 662–671.

[8] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

[9] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet

Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 115(3):211–252, 2015.

[10] M. Sundermeyer, H. Ney and R. Schlüter, "From Feedforward to Recurrent LSTM Neural Networks for Language Modeling," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 517-529, March 2015, doi: 10.1109/TASLP.2015.2400218

[11] M. Agarwal, R. Shah, and P. Mannem, "Automatic question generation using discourse cues," in Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications. Association for Computational Linguistics, 2011, pp. 1–9.

[12] A. Abujabal, M. Yahya, M. Riedewald, and G. Weikum, "Automated template generation for question answering over knowledge graphs," in Proc. 26th int. conf. on world wide web, 2017, pp. 1191–1200.

[13] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

[14] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," 2017, to appear.

[15] M. Liu, V. Rus and L. Liu, "Automatic Chinese Factual Question Generation," in IEEE Transactions on Learning Technologies, vol. 10, no. 2, pp. 194-204, 1 April-June 2017, doi: 10.1109/TLT.2016.2565477.

3. PRESENTATION CERTIFICATE

IEEE ICEARS St. MOTHER THERESA ENGINEERING COLLEGE
Approved by AICTE & Affiliated to Anna University | ISO Certified Institution
Counselling Code : 4933 Vagaikulam, Thoothukudi

International Conference on
Electronics and Renewable Systems
(ICEARS 2022)

16-18, March 2022 | Tuticorin, India

Certificate of Presentation

This is to certify that

Vaishali Hirlekar

has successfully presented the paper entitled

Generating QA from rule-based algorithms

at the
International Conference on Electronics and Renewable Systems (ICEARS 2022)
organized by St. Mother Theresa Engineering College, Tuticorin, Tamil Nadu, India
held on 16-18, March 2022.

Session Chair | Organizing Secretary Dr. K. Jeyakumar | Conference Chair Dr. A. George Klington

IEEE ICEARS St. MOTHER THERESA ENGINEERING COLLEGE
Approved by AICTE & Affiliated to Anna University | ISO Certified Institution
Counselling Code : 4933 Vagaikulam, Thoothukudi

International Conference on
Electronics and Renewable Systems
(ICEARS 2022)

16-18, March 2022 | Tuticorin, India

Certificate of Presentation

This is to certify that

Pratiksha Rajesh Rao

has successfully presented the paper entitled

Generating QA from rule-based algorithms

at the
International Conference on Electronics and Renewable Systems (ICEARS 2022)
organized by St. Mother Theresa Engineering College, Tuticorin, Tamil Nadu, India
held on 16-18, March 2022.

Session Chair | Organizing Secretary Dr. K. Jeyakumar | Conference Chair Dr. A. George Klington

IEEE ICEARS St. MOTHER THERESA ENGINEERING COLLEGE
Approved by AICTE & Affiliated to Anna University | ISO Certified Institution
Counselling Code : 4933 Vagaikulam, Thoothukudi

International Conference on
Electronics and Renewable Systems
(ICEARS 2022)

16-18, March 2022 | Tuticorin, India

Certificate of Presentation

This is to certify that

Tanay Navneet Jhawar

has successfully presented the paper entitled

Generating QA from rule-based algorithms

at the
International Conference on Electronics and Renewable Systems (ICEARS 2022)
organized by St. Mother Theresa Engineering College, Tuticorin, Tamil Nadu, India
held on 16-18, March 2022.

Session Chair | Organizing Secretary Dr. K. Jeyakumar | Conference Chair Dr. A. George Klington

IEEE ICEARS St. MOTHER THERESA ENGINEERING COLLEGE
Approved by AICTE & Affiliated to Anna University | ISO Certified Institution
Counselling Code : 4933 Vagaikulam, Thoothukudi

International Conference on
Electronics and Renewable Systems
(ICEARS 2022)

16-18, March 2022 | Tuticorin, India

Certificate of Presentation

This is to certify that

Yash Avinash Kachave

has successfully presented the paper entitled

Generating QA from rule-based algorithms

at the
International Conference on Electronics and Renewable Systems (ICEARS 2022)
organized by St. Mother Theresa Engineering College, Tuticorin, Tamil Nadu, India
held on 16-18, March 2022.

Session Chair | Organizing Secretary Dr. K. Jeyakumar | Conference Chair Dr. A. George Klington

4. COPYRIGHT

FORM XIV
APPLICATION FOR REGISTERATION OF COPYRIGHT
[SEE RULE 70]

Diary Number: 9615/2022-CO/L

To

The Registrar of Copyrights,
Copyright Office,
Department of Industrial Policy & Promotion,
Ministry of Commerce and Industry,
Boudhik Sampada Bhawan,
Plot No. 32, Sector 14, Dwarka,
New Delhi-110075
Email Address: copyright@nic.in
Telephone No.: (Office) 011-28032496, 08929474194

Sir,

In Accordance with Section 45 of the Copright Act, 1957 (14 of 1957), I hereby apply for registration of Copyright and request that enteries may be made in the Register of Copyrights as in the enclosed Statement of Particulars.

1. I also send herewith duly completed the Statement of further Particulars relating to the work. (for Literary/Dramatic, Musical, Atristic works only) **Literary/ Dramatic works**

2. In accordance with rule 16 of the Copyright Rules, 1958, I have sent by prepaid registered post copies of this letter and of the Statement of Particulars and Statement of Further Particulars to other parties concerned as shown below:

Name of Party	Address of Party	Date of Dispatch
SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE	MAHAVIR EDUCATION TRUST, CHOWK, W.T.PATIL MARG, CHEMBUR, NEXT TO DUKE'S COMPANY, MUMBAI, MAHARASHTRA-400088	07/05/2022

[See columns 7,11,12, and 13 of the Statement of Particulars and party referred in col.2 (e) of the Statement of Further Particulars.]

3. The prescribed fee has been paid, as per details below: **500/-**

Payment ID	Payment Date	Amount	Bank Name	Payment Mode
279436	07/05/2022	500		

4. Communications on this subject may be addressed to:

VAISHALI V. HIRLEKAR
FLAT NO 204, OM SHREE VINAYAK CHS., PLOT NO.56, SECTOR- 50 (E), SEAWOODS,
NEW MUMBAI-400706
8454844993

5. I hereby declare that to the best of my knowledge and belief, no person, other than to whom a notice has been sent as per paragraph 2 above any claim or interest or dispute to my copyright of this work or its use by me.

6. I hereby verify that the particulars given in this Form and the Statement of Particulars and Statment of Further Particulars are true to the best of my knowledge, belief and information and nothing has been concealed there from.

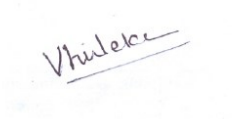
List of Enclosures:

1. 2 Copies of Work
2. DD/IPO of Rs.500 Per Work
3. Authorization from author/publisher
4. If the application is being filed through attorney , a specific Power of Attorney in original duly signed by the applicant and accepted by the attorney

Place:

Date: **07/05/2022**

For : PRATIKSHA RAJESH RAO



Proprietor

STATEMENT OF PARTICULARS

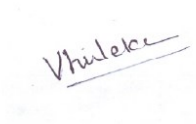
Diary Number: 9615/2022-CO/L

1.	Registration Number	
2.	Name, Address and Nationality of the Applicant	NAME: PRATIKSHA RAJESH RAO, ADDRESS: 303 TRIPOLI, SKYLINE OASIS, PREMIER ROAD, VIDYAVIHAR, GHATKOPAR, MUMBAI-400086, Indian NAME: TANAY NAVNEET JHAWAR, ADDRESS: 510 AJITNATH, NEELKANTH ENCLAVE, OPP. HELIX-3, L.B.S. MARG, GHATKOPAR WEST, MUMBAI-400086, Indian NAME: YASH AVINASH KACHAVE, ADDRESS: 46, NEHRU HOUSING SOCIETY, NEAR NAVRANG WATER TANK, DEOPUR, DHULE-424002, Indian NAME: SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE, ADDRESS: MAHAVIR EDUCATION TRUST, CHOWK, W.T.PATIL MARG, CHEMBUR, NEXT TO DUKE'S COMPANY, MUMBAI--400088, Indian NAME: VAISHALI HIRLEKAR, ADDRESS: FLAT NO 204, OM SHREE VINAYAK CHS., PLOT NO 56, SECTOR 50 E, SEAWOODS, NEW MUMBAI--400706, Indian
3.	Nature of the Applicant's interest in the Copyright of the work	Owner
4.	Class and description of the work	Literary/ Dramatic Work
5.	Title of the work	Generating QA from rule based algorithms
6.	Language of the work	English
7.	Name, Address and Nationality of the Author and if the Author is deceased, the date of decease.	NAME: PRATIKSHA RAJESH RAO, ADDRESS: 303 TRIPOLI, SKYLINE OASIS, PREMIER ROAD, VIDYAVIHAR, GHATKOPAR, MUMBAI-400086, Indian, NAME: TANAY NAVNEET JHAWAR, ADDRESS: 510 AJITNATH, NEELKANTH ENCLAVE, OPP. HELIX-3, L.B.S. MARG, GHATKOPAR WEST, MUMBAI-400086, Indian, NAME: YASH AVINASH KACHAVE, ADDRESS: 46, NEHRU HOUSING SOCIETY, NEAR NAVRANG WATER TANK, DEOPUR, DHULE-424002, Indian, NAME: SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE, ADDRESS: MAHAVIR EDUCATION TRUST, CHOWK, W.T.PATIL MARG, CHEMBUR, NEXT TO DUKE'S COMPANY, MUMBAI--400088, Indian, NAME: VAISHALI HIRLEKAR, ADDRESS: FLAT NO 204, OM SHREE VINAYAK CHS., PLOT NO 56, SECTOR 50 E, SEAWOODS, NEW MUMBAI--400706, Indian,
8.	Whether the work is Published or Unpublished	Unpublished
9.	Year and Country of first publication, and Name, Address and Nationality of the publisher	N/A
10.	Year and Countries of subsequent publications, if any, and Name, Address and Nationality of the publisher	N/A
11.	Name, Address and Nationality of the Owners of the various rights comprising the copyright in the work and extent of rights held by each, together with particulars of assignments and licence. If any	NAME: PRATIKSHA RAJESH RAO, ADDRESS: 303 TRIPOLI, SKYLINE OASIS, PREMIER ROAD, VIDYAVIHAR, GHATKOPAR, MUMBAI-400086, Indian NAME: TANAY NAVNEET JHAWAR, ADDRESS: 510 AJITNATH, NEELKANTH ENCLAVE, OPP. HELIX-3, L.B.S. MARG, GHATKOPAR WEST, MUMBAI-400086, Indian NAME: YASH AVINASH KACHAVE, ADDRESS: 46, NEHRU

		HOUSING SOCIETY, NEAR NAVRANG WATER TANK, DEOPUR, DHULE-424002, Indian NAME: SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE, ADDRESS: MAHAVIR EDUCATION TRUST, CHOWK, W.T.PATIL MARG, CHEMBUR, NEXT TO DUKE'S COMPANY, MUMBAI--400088, Indian NAME: VAISHALI HIRLEKAR, ADDRESS: FLAT NO 204, OM SHREE VINAYAK CHS., PLOT NO 56, SECTOR 50 E, SEAWOODS, NEW MUMBAI--400706, Indian
12.	Name and address and nationality of other persons, if any authorized to assign or licence the rights comprising the copyright	NAME: PRATIKSHA RAJESH RAO, ADDRESS: 303 TRIPOLI, SKYLINE OASIS, PREMIER ROAD, VIDYAVIHAR, GHATKOPAR, MUMBAI-400086, Indian NAME: TANAY NAVNEET JHAWAR, ADDRESS: 510 AJITNATH, NEELKANTH ENCLAVE, OPP. HELIX-3, L.B.S. MARG, GHATKOPAR WEST, MUMBAI-400086, Indian NAME: YASH AVINASH KACHAVE, ADDRESS: 46, NEHRU HOUSING SOCIETY, NEAR NAVRANG WATER TANK, DEOPUR, DHULE-424002, Indian NAME: SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE, ADDRESS: MAHAVIR EDUCATION TRUST, CHOWK, W.T.PATIL MARG, CHEMBUR, NEXT TO DUKE'S COMPANY, MUMBAI--400088, Indian NAME: VAISHALI HIRLEKAR, ADDRESS: FLAT NO 204, OM SHREE VINAYAK CHS., PLOT NO 56, SECTOR 50 E, SEAWOODS, NEW MUMBAI--400706, Indian
13.	If the work is an 'Artistic work', the location of the original work, including name, address and nationality of the person in possession of the work, (In the case of an architectural work, the year of completion of the work should also be shown)	N/A
14.	If the work is an 'Artistic work' which is used or capable of being used in relation to any goods or services, the application should include a certification from the Registrar of Trade Marks in terms of the provision to Sub-Section (i) of Section 45 of the Copyright Act, 1957	N/A
15.	If the work is an 'Artistic work' whether it is registered under the Desings Act 2000 if yes give details.	N/A
16.	If the work is an 'Artistic work' capable of being registrar as a design under the Designs Act 2000, whether is has been applied to an article though an industrial process and,if yes ,then number of times it is reproduced	N/A
17.	Remarks, if any	

Place:
Date: **07/05/2022**

For : PRATIKSHA RAJESH RAO



Proprietor

STATEMENT OF FURTHER PARTICULARS

(For Literary/Dramatic, Musical and Artistic works only)

Diary Number: 9615/2022-CO/L

1. Is the work to be registered

(a) an orginal work? : Yes

(b) a translation of a work in the public domain? : N.A.

(c) a translation of a work in which Copyright subsists? : N.A.

(d) an adaptation of a work in the public domain? : N.A.

(e) an adaptation of a work in which Copyright subsists? : N.A.

2. If the work is a translation or adaptation of a work in which copyright subsists

(a) Title of the original work : N.A.

(b) Language of the original work : N.A.

(c) Name, address, and nationality of the author of the original work and if the author is deceased, the date of decease : N.A.

(d) Name, address, and nationality of the publisher, if any, of the original work : N.A.

(e) Name, address, and nationality of the publisher, or adaptation including the name, address and nationality of party authorizing : N.A.

3. Remarks, if any

Place:
Date: **07/05/2022**

For : PRATIKSHA RAJESH RAO

Proprietor

5. PLAGIARISM REPORT

(a) Black book Report

Curiginal

Document Information

Analyzed document	Major Project_Group 19.pdf (D135670433)
Submitted	2022-05-06T15:11:00.0000000
Submitted by	
Submitter email	tanay.jhawar@sakec.ac.in
Similarity	5%
Analysis address	vaishali.hirlekar.sakec@analysis.urkund.com

Sources included in the report

W	URL: https://www.semanticscholar.org/paper/Automatic-factual-question-generation-from-text-Smith-Heilman/ebd9458f8c7e14a04f3fb711891a4f1bcd065297 Fetched: 2021-11-30T10:05:26.4870000	3
SA	**Team 12.pdf** Document Team 12.pdf (D106324790)	1
SA	**150102014,150104014 - Moushree Dey.pdf** Document 150102014,150104014 - Moushree Dey.pdf (D53889342)	2
SA	**final report of monika.docx** Document final report of monika.docx (D110906264)	1
W	URL: https://www.frontiersin.org/articles/10.3389/fpsyg.2019.00825/full Fetched: 2020-05-01T10:44:30.9570000	1
W	URL: https://aclanthology.org/W18-0533 Fetched: 2022-05-06T15:20:03.9930000	1
SA	**121722015_Rohan_Tipare.pdf** Document 121722015_Rohan_Tipare.pdf (D53840145)	1

(b) Research Paper Report

Curiginal

Document Information

Analyzed document	IEEEPaper_AutomaticQuestionGeneration.pdf (D126245010)
Submitted	2022-01-27T05:59:00.0000000
Submitted by	Vaishali Vaibhav Hirlekar
Submitter email	vaishali.hirlekar@sakec.ac.in
Similarity	3%
Analysis address	vaishali.hirlekar.sakec@analysis.urkund.com

Sources included in the report

W	URL: https://link.springer.com/article/10.1007/s40593-019-00186-y Fetched: 2019-11-26T06:15:09.9730000	3
W	URL: https://ieeexplore.ieee.org/document/9232485 Fetched: 2021-05-13T09:05:04.0370000	1

ACKNOWLEDGMENT

Our efforts into this project would not have born fruits without the kind support and help of our dearest faculty members. We wish to express our profound gratitude to our principal Dr. Bhavesh Patel for allowing us to go ahead with this project and giving us the opportunity to explore this domain. We would also like to convey our thanks to our Head of department Prof. Uday Bhave for his constant encouragement and kind co-operation towards achieving our goals.

We are immensely thankful to the Review Committee for their valuable suggestions and feedback without which our work would have been extremely tedious.

We take this opportunity to express our profound gratitude and deep regards to our guide Prof. Vaishali Hirlekar for her cordial support, exemplary guidance, constant supervision and encouragement throughout the course of project. Her lessons, help and blessings shall carry us a long way in the journey of life on which we are about to embark. Further, we would like to take this opportunity to appreciate our classmates who willingly imparted knowledge and helped us out whenever we were stuck at any point. Finally, we would like to take this opportunity and convey our gratitude and thanks to all those without whom this project wouldn't have been implemented successfully.

Pratiksha Rajesh Rao

Tanay Navneet Jhawar

Yash Avinash Kachave