AUTOMATIC QUESTIONS GENERATION USING NATURAL LANGUAGE PROCESSING

BY

V.GAYATHRI

19PCA002

Project Report Submitted

In partial fulfillment of the requirements for the degree of

Master of Computer Applications

DEPARTMENT OF COMPUTER SCIENCE

AVINASHILINGAM INSTITUTE FOR HOME SCIENCE AND HIGHER EDUCATION FOR WOMEN

COIMBATORE – 641043

MAY - 2022

AUTOMATIC QUESTIONS GENERATION USING NATURAL LANGUAGE PROCESSING

BY

V.GAYATHRI

19PCA002

Project Report Submitted

In partial fulfillment of the requirements for the degree of

Master of Computer Applications

DEPARTMENT OF COMPUTER SCIENCE

AVINASHILINGAM INSTITUTE FOR HOME SCIENCE AND

HIGHER EDUCATION FOR WOMEN

COIMBATORE – 641043

MAY - 2022

Signature of the Head of the Department **Signature of Supervisor**

Viva-voce Examination Held on _____

Signature of Examiners

ACKNOWLEDGEMENT

ACKNOWLEDGMENT

I would like to express my sincere thanks to **God Almighty**, for his constant love and grace that he has shown upon me, which kept me in good health, and sound mind without which my project would not have reached a successful end.

I would like to express my deep sense of reverential gratitude and sincere thanks to **Professor, S. P. Thygarajan, Chancellor**, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore - 43, for the opportunity given to me for undertaking this study and for providing the needed facility during the course of my study.

I owe my great deal of gratitude to **Dr. Bharathi Harishankar Vice-Chancellor**, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for extending all resources that facilitated the conduct of the present study.

I express my gratitude to **Dr. S. Kowsalya, Registrar**, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for providing all facilities necessary for the study.

I would express my boundless thanks to **Dr. G. Padmavathi, M.Sc., M.Phil., Ph.D., Dean, School of Physical Sciences & Computational Sciences,** Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for granting the facility required.

I wish to place on record my deep sense of gratitude to **Dr. Vasantha Kalyani David, M.Sc., M.Phil(Maths)., M.Phil(CS)., Ph.D., Professor and Head**, **Department of Computer Science** for providing all the facilities to complete the project.

I express my heart full gratitude to my mentor **Dr. V. Srividhya, M.Sc., M.Phil., Ph.D., Professor of Computer Science,** for guidance and timely suggestions shown in aiding me to consummate this project work successfully.

I am grateful to the project coordinator **Dr. B. Kalpana M.Sc., M.Phil., Ph.D., Professor** who was instrumental in granting me the facilities required for doing the project.

Finally, yet importantly, I would like to thank my **family members and friends** for their support, encouragement, and prayers, which were instrumental in the successful completion of this project.

I have great pleasure in expressing my deep sense of gratitude to all other teaching and non-teaching staff members of the Department of Computer Science, who stood behind the screen for the completion of the project.

I would extend my hearty thanks to one and all that helped me directly or indirectly forthe successful completion of my project.

ABSTRACT

ABSTRACT

The Project named "**AUTOMATIC QUESTIONS GENERATION USING NATURAL LANGUAGE PROCESSING**". Students ask questions to satisfy their never ending questions for getting knowledge. To perform competently assessment of students on their major concepts they learned from the study material. Preparing a set of questions for assessment can be time-consuming for teachers while getting questions from external sources like assessment books or question banks might not be relevant to content studied by students.

Automatic Questions Generation (AQG) is the technique for generating a correct set of questions from a paragraph, which can be text. It is very important for many educational institutes but yet challenging problems in NLP. It is defined as the task of generating syntactically correct sound, semantically perfect and relevant questions from several input formats like text, paragraphs.

There are two models in this project. The first type of work is generating Multiple Choice Questions (MCQs), and the second model is generating Frequently Asked Questions (FAQs). There are six steps in MCQs. Load the raw text first. To construct MCQs, the BERT (Bidirectional Encoder Representations Transformer) approach is utilized to summarize the text. Then the keywords are taken from the summarized text and sentence mapping is performed. Distractor are created using WordNet to produce choices to the questions (A lexical database for English). Because the BERT model outperforms other legacy model and can analyze massive amount of data in less time, it is taken for summarization. There are six steps in the FAQ. To begin, load the raw text. The text is subsequently summarized using the abstractive T5 (Text-To-Text Transformer) approach. After that, extract the significant keywords from the summary text and load the pre-trained model for FAQs generation. Because it outperforms other legacy model and can analyze a large amount of data in less time, the T5 model has taken for generating FAQs from input text.

This project is implemented using python programming language.

CONTENTS

CONTENTS

INTRODUCTION

1. INTRODUCTION

1.1 NATURAL LANGUAGE PROCESSNG

Natural language processing (NLP) refers to the branch of computer science and more specifically, the branch of artificial intelligence or AI concerned with giving computers the ability to understand text and spoken words in much the same way human beings can. NLP combines computational linguistics rule-based modeling of human language with statistical, machine learning, and deep learning models. Together, these technologies enable computers to process human language in the form of text or voice data and to understand ‘it’s full meaning, complete with the speaker or writer‘s intent and sentiment.

NLP drives computer programs that translate text from one language to another, respond to spoken commands, and summarize large volumes of text rapidly even in real time. There‘s a good chance to interact with NLP in the form of voice-operated GPS systems, digital assistants, speech-to-text dictation software, customer service chatbots, and other consumer conveniences. But NLP also plays a growing role in enterprise solutions that help streamline business operations, increase employee productivity, and simplify mission-critical business processes.

1.1.1 NEED OF NLP

The biggest benefit of NLP for businesses is the ability of technology to detect, and process massive volumes of text data across the digital world including; social media platforms, online reviews, news reports, and others. Also, by collecting and analyzing business data, NLP is able to offer businesses valuable insights into brand performance. In addition, NLP models can detect any persisting issues and take necessary mitigation measures to improve performance.

Google speech to text is able to achieve all of this by training machines to understand human language in a faster, more accurate, and consistent way than human agents. The technology is able to consistently monitor and process data. This helps brands remain updated with their online presence, and not get riddled with inconsistencies.

1.1.2 BENEFITS OF NLP

- **Large volumes of textual data**

  Natural language processing helps computers communicate with humans in their own language and scales other language-related tasks. For example, NLP makes it possible for computers to read text, hear speech, interpret it, measure sentiment and determine which parts are important.

  Nowadays machines can analyze more language-based data than humans, without fatigue and in a consistent, unbiased way. Considering the staggering amount of unstructured data that‘s generated every day, from medical records to social media, automation will be critical to fully analyze text and speech data efficiently.

- **Structuring a highly unstructured data source**

  The complexity and diversity of human language is remarkable. Humans can communicate in a variety of ways, both verbally and in writing. There are hundreds of languages and dialects, and each has its own set of grammar and syntax rules, as well as words and slang. Humans misspell, truncate, and omit punctuation when writing. They have regional accents, mumble, stutter, and borrow terminology from other languages when speaking.

  While supervised and unsupervised learning, and specifically deep learning, are now widely used for modeling human language, there‘s also a need for syntactic and semantic understanding and domain expertise that are not necessarily present in these machine learning approaches. NLP is important because it helps resolve ambiguity in language and adds useful numeric structure to the data for many downstream applications, such as speech recognition or text analytics.

1.1.3 CLASSIFICATION OF NLP

The Following are two components of NLP.

1. Natural Language Understanding (NLU)

Natural Language Understanding (NLU) helps the machine to understand and analyses human language by extracting the metadata from content such as concepts, entities, keywords, emotion, relations, and semantic roles.

Both NLP and NLU aim to make sense of unstructured data, but there is a difference between the two. NLP is concerned with how computers are programmed to process language and facilitate natural back-and-forth communication between computers and humans. Natural language understanding, on the other hand, focuses on a machine's ability to understand the human language. NLU refers to how unstructured data is rearranged so that machines may "understand" and analyze it.

NLU mainly used in Business applications to understand the customer's problem in both spoken and written language. NLU involves the following tasks.

- It is used to map the given input into useful representation.
- It is used to analyze different aspects of the language.

2. Natural Language Generation (NLG)

Natural Language Generation (NLG) acts as a translator that converts the computerized data into natural language representation. It mainly involves Text planning, Sentence planning, and Text Realization.

NLG, a subfield of Artificial Intelligence (AI), is a software process that automatically transforms data into plain-English content. The technology can actually tell a story exactly like that of a human analyst by writing the sentences and paragraphs. NLG is one of the fastest growing technologies being adopted in the enterprise.

There many use-cases for NLG, but where it is seen to be most effective is when deployed to automate time-intensive data analysis and reporting activities. Fig 1.1 shows that the broad classification of NLP.

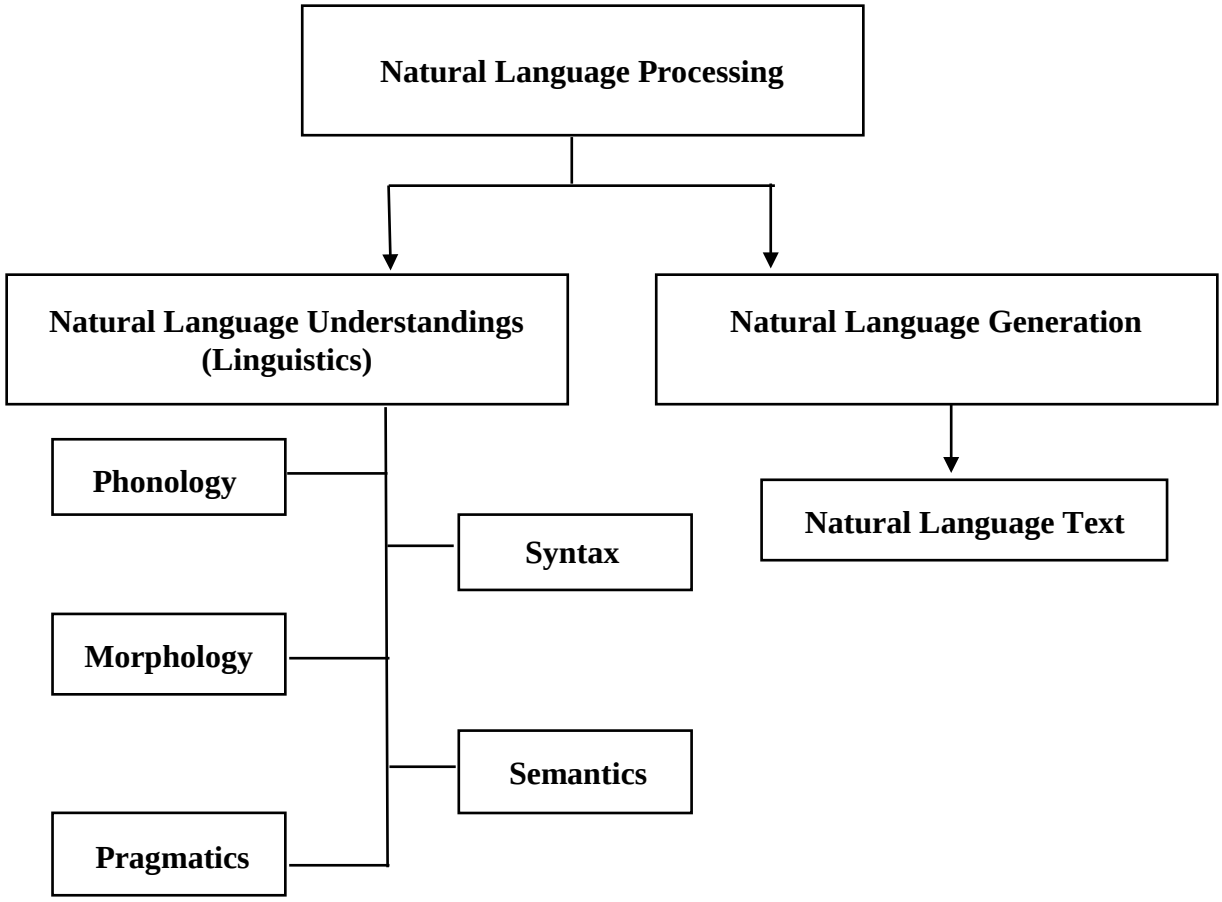

Fig 1.1: Classification of NLP

1.1.4 COMPONENTS OF NLP

The five major Component of Natural Language processing in AI are:

- Morphological and Lexical Analysis
- Syntactic Analysis

- Semantic Analysis
- Discourse Integration
- Pragmatic Analysis

I) MORPHOLOGICAL AND LEXICAL ANALYSIS

Lexical analysis is a vocabulary that includes its words and expressions. It depicts analyzing, identifying and description of the structure of words. It includes dividing a text into paragraphs, words and the sentences

Individual words are analyzed into their components, and nonword tokens such as punctuations are separated from the words.

II) SEMANTIC ANALYSIS

Semantic Analysis is a structure created by the syntactic analyzer which assigns meanings. This component transfers linear sequences of words into structures. It shows how the words are associated with each other.

Semantics focuses only on the literal meaning of words, phrases, and sentences. This only abstracts the dictionary meaning or the real meaning from the given context. The structures assigned by the syntactic analyzer always have assigned meaning

III) PRAGMATIC ANALYSIS

Pragmatic Analysis deals with the overall communicative and social content and its effect on interpretation. It means abstracting or deriving the meaningful use of language in situations. In this analysis, the main focus always on what was said in reinterpreted on what is meant.

Pragmatic analysis helps users to discover this intended effect by applying a set of rules that characterize cooperative dialogues.

IV) SYNTAX ANALYSIS

The words are commonly accepted as being the smallest units of syntax. The syntax refers to the principles and rules that govern the sentence structure of any individual languages.

Syntax focus about the proper ordering of words which can affect its meaning. This involves analysis of the words in a sentence by following the grammatical structure of the sentence. The words are transformed into the structure to show how the words are related to each other.

V) DISCOURSE INTEGRATION

It means a sense of the context. The meaning of any single sentence which depends upon that sentences.

1.1.5 APPLICATIONS OF NLP

Speech Recognition: Speech Recognition is a technology that enables the computer to convert voice input data to machine readable format. There are a lot of fields where speech recognition is used like, virtual assistants, adding speech-to-text, translating speech, sending emails etc. It is used in search engines where the user can voice out the name of their search requirements and get the desired result, making our work easier than typing out the entire command.

Voice Assistants and Chatbot: Voice assistant is software that uses NLP and speech recognition to understand voice commands of a user and perform accordingly. Similarly, Chatbots are programs that are designed to assist a user 24/7 and respond appropriately and answer any query that the user might have. Most Chatbots and Virtual Assistants have pre-programmed answering systems that follow specific rules and patterns while answering. Powerful AI has enabled some voice assistants to interact with the user and respond appropriately. With more usage, they even improve themselves. Assistants like Siri and Alexa can even have a conversation with the user like a normal human being.

Auto Correct and Auto prediction: There are many types of software available nowadays that check grammar and spelling of the text one type and save us from embarrassing spelling and grammatical mistakes in our emails, texts or other documents. NLP plays an important role in those softwares and functions. This is one of the most widely used applications of NLP. These softwares offer a lot of features like suggesting synonyms, correcting grammar and spellings, rephrasing sentences and giving clarity to the document and can even predict the tone of the

sentence that might be implied by the user. Auto prediction is also a feature developed through NLP where the computer suggests automatic prediction of the text one have started typing. This saves time of the user and makes the job easier for them.

Spam Filters: One of the most irritating things about email is spam. Gmail uses natural language processing (NLP) to discern which emails are legitimate and which are spam. These spam filters look at the text in all the emails receive and try to figure out what it means to see if it‘s spam or not.

Algorithmic Trading: Algorithmic trading is used for predicting stock market conditions. Using NLP, this technology examines news headlines about companies and stocks and attempts to comprehend their meaning in order to determine if one should buy, sell, or hold certain stocks.

Email Filtering: Most of the professional work is done through emails and it would be quite a hassle if all the emails we received were not segregated into different sections. Gmail classifies all the emails into primary, social and promotional sections. Even all the spam emails are sent to a different section so that they do not flood our inbox. This is done with the help of text classification, which is a technique of NLP. It has definitely helped to save time and not miss any important Email that might have gotten lost if all the useless emails started accumulating in the inbox.

Translation: Social Media has brought the entire world together but with unity comes challenges like language barrier. With different translating softwares that work individually or are integrated within other applications, this hurdle has been easily defeated. This is called Machine Translation, which uses Natural Language Processing, and has made a lot of improvement in the field due to availability of huge amounts of data and powerful machines, and advancement in the field of Machine learning and Neural networking. It has particularly helped businesses in grabbing foreign customers and social media users who are connected to people overseas. Platforms like Facebook and Instagram have their own translation software integrated within the main application.

Answering Questions: NLP can be seen in action by using Google Search or Siri Services. A major use of NLP is to make search engines understand the meaning of what humans are asking and generating natural language in return to give to the answers.

Summarizing Information: On the internet, there is a lot of information, and a lot of it comes in the form of long documents or articles. NLP is used to decipher the meaning of the data and then provides shorter summaries of the data so that humans can comprehend it more quickly. This application is used in Investigative Discovery to identify patterns in writing reports, Social Media Analytics to track awareness and identify influencers, and Subject-matter expertise to classify content into meaningful topics.

1.2 ROLE OF QUESTION GENERATION IN NLP

Automatic question generation is part of Natural Language Processing (NLP). It is an area of research where many researchers have presented their work and is still an area under research to achieve higher accuracy. Many researchers have worked in the area of automatic question generation through NLP, and numerous techniques and models have been developed to generate the different types of question automatically. Work has been done in many languages.

Nowadays, teachers/professors/tutors (academicians) spend a lot of time generating test papers and quizzes manually. Similarly, students spend a lot of time on self-analysis (self- calibration). Moreover, students are dependent on their mentors for the self-analysis. Hence, when one working on this NLP area, which has a huge scope of development at this moment. One wants to build a computer application system that can help in calibrating and remove any dependencies on mentors. Here, students can give the input text of whatever material they referred to, and on this basis they get a set of questions with answers from which they can do a self-analysis (self-calibration). A similar approach is used by mentors for creating test papers and quizzes. Moreover, online examinations have become very popular, including many major examinations, such as GATE, CAT, and NET.

Multiple Choice Questions (MCQ) is very easy for evaluations, and its evaluation is implemented through computerized application so that results can be declared within a few

hours, and the evaluation process is 100% pure. By making this computerized application, and also it can reduce the task of an educator. Much time and cost can be saved. Hence, this system generates a variety of logical questions based on the text input using Natural Language Processing.

1.3 MOTIVATION

All institutes, colleges, and schools have been switched to online learning. Assessment is an essential tool to test the knowledge of the students. The pattern of the assessment has changed from subjective based to objective based i.e. Multiple Choice Questions (MCQs). So the problem is, it is very difficult for the teachers to set the questions as well as for the students who are preparing for competitive exams. The current method involves the setting of questions manually which requires a lot of human intervention and time. So there is a growing need for a system that can create questions with ease and less amount of time and requires less human effort.

This becomes our main motivation to automate the generation of questions with the hope that the potential benefits from an automated QG system could assist humans in meeting their useful inquiry needs. Question Generation (QG) and Question Answering (QA) became the two major challenges for natural language understanding communities, recently QG has turned into an essential element of learning environments, systems, information seeking systems, etc Considerable interest from the Natural Language Processing (NLP), Natural Language Generation , Intelligent Tutoring System, and Information Retrieval (IR) communities have currently identified the Text-to-Question generation task as a promising candidate for the shared task In the Text-to Question generation task, a QG system is given a text (such as a word, a set of words, a single sentence, a text, a set of texts, a stretch of conversational discourse, an inadequate question, and so on), and its goal would be to make a set of questions for educational purpose.

The task of questions generation (QG) is defined as generating an interrogative sentence from a given paragraph or sentence (which is referred to as the context).

It is considered the inverse problem of questions generation (QG), the task of question generation is also considered an important sub-task for question answering systems because to extract appropriate answers from a given context, one must ask the right question.

1.4 PROBLEM STATEMENT

Automatic questions generator is designed to generate various types of questions automatically such as MCQ (Multiple Choice questions) and FAQ (Frequently Asked Questions) from the given text using NLP (Natural Language Processing).

1.5 OBJECTIVES

- The main objective of this project is to create a coherent and fluent summary of the given text.
- Extracting the most relevant words and expressions from text using keyword extraction algorithm.
- To generate relevant distractors using suitable algorithm.
- Automatically generate the structured questions using pre-trained model.

1.6 OVERVIEW OF THE PROJECT

AQG (Automatic Questions Generation) is the means of generating questions automatically using a computer. The systems usually work by supplying a text or a knowledge base to generate questions. AQG can be applied to an array of fields such as education, development of conversational agents, question answering machines or machine reading comprehension systems. The question type can vary between free response (FR), Multiple Choice Questions (MCQ), True and False questions (T/F), Gap-fill questions (GFQ) and what/when/who/why questions (wh-questions). The research on AQG has been rising in recent years. This is mainly a reaction to the growth of E-learning platforms MOOC‘s (Massive Open Online Courses) where people can enroll on different courses from home.

The goal of Question Generation is to generate a valid and fluent question according to a given passage using Natural Language Processing.

This project consists of two models. Generating MCQs Questions is the first model of work, while generating FAQs Questions is the second model of work. The six steps are included in the MCQ. The raw text must be loaded first. The extractive summarization phase is the next step. Keyword extraction is the third phase, which involves extracting the most relevant keywords from the summary text. The fourth step is sentence mapping. The keywords are used to map the sentences. MCQ creation is the work's fifth step. It generates MCQs with distractors. The last step of work is the comparison of algorithms to find the best approach for summarization. The five steps are listed in the FAQs. The raw text must be loaded first. Abstractive summarization is the next stage. It selects the most crucial sentences from the input material. Keyword extraction is the third step, and here, load the pre-trained model and FAQ production is the fourth and fifth step. The performance evaluation is the sixth step. Here it compares the two models, determining which one gives the best summary. Fig 1.2 shows that the overview of the project.

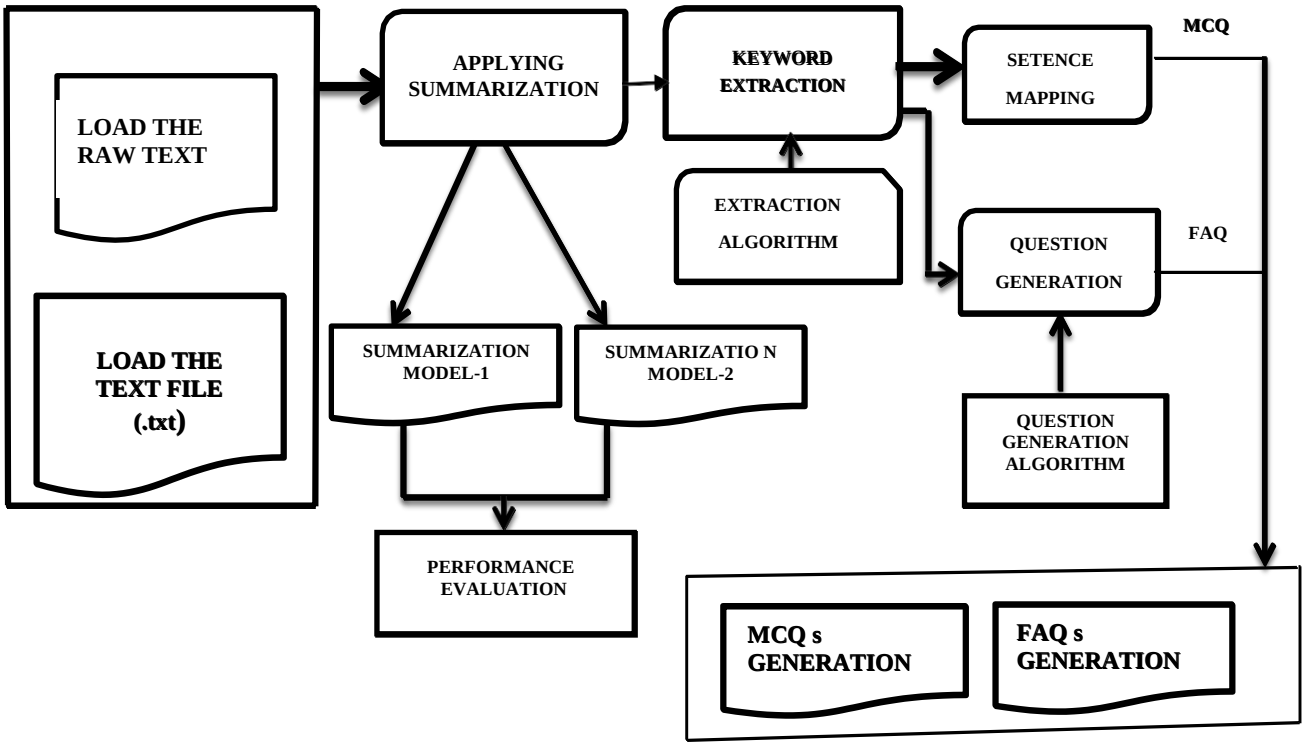

Fig 1.2: Overview Diagram

LITERATUREREVIEW

2. LITERATURE REVIEW

Literature review of various studies related to question generation approaches therefore, this section elucidates some of the research directions observed in this regard.

1. Santhanavijayan et al. They produce MCQs, using fireflies-based preference learning and an ontology-based approach in their suggested system. To make it possible to create questions, they employed a web corpus. Similarity measures like hypernyms and hyponyms are used to generate the distractors. The system also generates analogy questions to assess the verbal ability of the students,

2. Ayako Hoshino and Hiroshi Nakagawa, Machine learning is used to produce questions automatically. They use machine learning methods like Naive Bayes and K-Nearest Neighbors to generate grammar and vocabulary problems based on web news articles.

3. D. R. CH and S. K. Articles from the database are utilized to generate questions in this paper. For text summarizing and word frequency counting, an NLP-based summarizer is employed, and pattern matching techniques are used for key selection. They employed wordnet, pattern matching, domain ontology, and semantic analysis to generate distractors.

4. Deepshree S. Vibhandik et al. The Automatic Question Generation system generates particular trigger questions and multiple-choice questions. Using the Lingo algorithm. In addition, the system uses regular expression pattern matching algorithms to extract abbreviations from student review papers in order to generate multiple-choice questions.

5. Susanti, Y et.al. A technique has been presented for automatically creating distractors for Multiple-Choice English vocabulary questions. The proposed technique introduces additional sources for locating distractor candidates and rates them based on semantic similarity and collocation data.

6. Onur et al suggested a rule based methodology to automatized question generation. The methodology proposed in this paper focuses on analysis of both syntactic and semantic structure of a sentence. This work mainly aims to generate more comprehensive questions by exploiting the semantic roles of words.

7. The framework here proposes a rule based mode to generation of questions from sentence. Reliance based, Named Entity Recognition (NER) based, and semantic role (SRL) marking based layouts/rules are used. For deciding between who and what questions system has proposed a solution by using Chunking.

8. Amruta Umardand et al stated in the paper main idea of question paper generation system. It states various methods that can be approached for the generation of questions, being it a randomized or automated process. Such automation system can be of great help. Database can be added and an administrative level of security can be given to the system, wherein questions once generated can be stored and reused by another author who is willing to create questions based on same or different topics.

9. Aleena et al paper focuses on the implementation idea for a question generation system. The main idea lies in the natural language understanding of the system, only then can the machine handle and manipulate the data. Preprocessing of data, key phase extraction and NLP are main concerns of the system proposed. In this a way a fast, secure and randomized system can be developed which is beneficial in many aspects including education.

10. Kalpana et al aims to discover answers to questions as: 'What is POS tagging?', Working of POS tagging. POS Tagging is the process of assigning one of the parts of speech to the given word. Parts of speech include nouns, verbs, adverbs, adjectives, pronouns, conjunctions and their sub-categories. Taggers use few sort of data: word references, vocabularies, rules, etc. Dictionaries have category or classifications of a specific word. We may have many words belonging to the same category. For example, escape is both noun and verb. Taggers utilize probabilistic information to resolve this equivocalness. Following the same method POS tagging is done and tags are assigned.

11. Edward Loper et al have introduced a new methodology to a streamlined and adaptable method of sorting out the practical component of computational linguistics. The NLTK toolkit provides an extensible, simple framework for processing of natural language. It covers symbolic and statistical natural language processing. The toolkit is executed as a set of modules; each module defines an alternative data structure or a task.

12. Ankita, K. A. et al used intricacy of POS tagging lies in the number of computational levels required for determining POS tags for a sentence. This paper centers on the number of comparisons made by Hidden Markov Model for POS labeling and the complexity can be reduced. They have also stated an additional method using Bloom Filter for NER (Named Entity Recognition.) In spite of the fact that there are numerous POS taggers accessible, individuals are yet taking a shot to discover a way which sets aside less effort for execution and the with less number of complications as well. In the HMM bases tagger it doesn‘t find tag for an individual word, rather it finds tag for a sentence as a whole. It uses transition as well as emission probabilities. The algorithm proposed combines two words as a chunk and calculates the tag for them considering them as single unit.

13. Mohd Husain et al. have stated various approaches to the classification of queries. objective type, fill in the blanks, or "wh" type questions. All of them can be generated by extracting appropriate data from the text by integration and conversion. Descriptive and factual questions can be developed using the same.

14. Pranita Jadhav et al have stated various steps when developing a system which generates fill in the blank type of questions out of many classifications of automated generation of questions. The advantage of this AGM is to speed up the process of evaluation in the education domain. Using Euclidean distance method, POS and tokenization, they have proposed the system.

15. Bowen Xu et al formative study indicates that developers need some automated answer generation tools to extract a succinct and diverse summary of potential answers to their technical questions from the sheer amount of information in Q&A discussions. To meet this need, we propose a three stage framework for automated generation of answer summary. Our user studies demonstrate the relevance, usefulness and diversity of our automated generated answer summaries.

16. Surbhi et al stated system wherein questions generated were stored in database and using randomization, random questions could be picked as output.

17. Sheetal et al have stated various reviews of different papers which have proposed many different algorithms for questions generation using Semantic role labeler or NLP. Much more work can be done in the same field proposing more complex methodologies.

18. Susanti, Y., Tokunaga, T., Nishikawa, H., and Obari, Proposed a method for automatically generating distractors for multiple-choice English vocabulary questions. The proposed method introduces new sources for collecting distractor candidates and utilizes semantic similarity and collocation information when ranking the collected candidates.

19. Das, B., and Majumder, M., It is looked at the development of an automatic factual open cloze question generation system which can generate fill-in-the-blank questions without alternatives. The system first extracts a set of informative sentences from the given input corpus based on Part-of-Speech tagging based rules.

20. ArikIturri, Developed an AQG for Basque language test questions. The information source for this question generator consists of linguistically analyzed real corpora, represented in XML markup language.

21. Susanti, Y., Tokunaga, T., Nishikawa, H., and Obari, H., Proposed a method for automatically generating distractors for multiple-choice English vocabulary questions. The proposed method introduces new sources for collecting distractor candidates and utilizes semantic similarity and collocation information when ranking the collected candidates.

22. Ashok Immanuel and Tulasi. B, The system performs many tasks in fasted way without any complexity. The question paper generation system uses question banks and different mathematical models.

23. Surbhi Choudhary, Abdul Rais Abdul Waheed, Shrutika Gawandi and Kavita Joshi, The Question Paper Generator System has provided a ready to use built-in question bank. The paper aptly describes CQZ (Cloze Question Generation) putting more emphasis on the actual type of the questions.

24. Egonmwan et al. proposed to use sequence-to-sequence and transformer models to generate abstractive summaries. Proposed summarization model consists of two modules: an extractive model and an abstractive model.

25. Karen Mazidi. using Natural Language Understanding and Natural language Generator approach, presented questions from text passage. The Question Generation system presented by the author provides questions of high quality linguistically and semantically. The approach infuses the question generation process with natural language Understanding analysis.

METHODOLOGY

3. METHODOLOGY

In this proposed method, the methodology is constructed into two models. Each model has its own task, which is followed by the other. The first model is the generating MCQs, and the second model is the generating FAQs. In MCQs, it contains the six steps. The first step is loading the raw text. The second step is the extractive summarization phase. The third step is keyword extraction, and sentence mapping is the fourth step. The fifth step is the MCQs generation, and the sixth step is the performance evaluation. In the FAQs, it contains the six steps. The first step is loading the raw text. The second step is the abstractive summarization phase. The third step is the keyword extraction, and load the pre-trained model is the fourth step. The fifth and sixth step is the FAQs generation and performance evaluation. This chapter describes the above models and finds a way to experiment with generating the questions for MCQs and FAQs using different types of algorithms.

3.1 MODULE DESCRIPTION

Question Generation is a powerful NLP (Natural Language Processing) technique for categorize the text of document into an organized groups. The first model is generating MCQs and second model is generating FAQs.

3.1.1 MULTIPLE CHOICE QUESTIONS

I. LOAD THE RAW TEXT

The first step is to load the raw text i.e. input text of any domain for which the questions to be generated.

II. EXTRACTIVE SUMMARIZATION

Extractive summarization picks up sentences directly from the document based on a scoring function to form a coherent summary.

This method work by identifying important sections of the text cropping out and stitch together portions of the content to produce a condensed version.

Thus, they depend only on the extraction of sentences from the original text. Most of the summarization research today has focused on extractive summarization, once it is easier and yields naturally grammatical summaries requiring relatively little linguistic analysis. Moreover, extractive summaries contain the most important sentences of the input, which can be a single document or multiple documents. A typical flow of extractive summarization systems consists of:

1. Constructs an intermediate representation of the input text intending to find salient content. Typically, it works by computing TF (Term Frequency) metrics for each sentence in the given matrix.

2. Scores the sentences based on the representation, assigning a value to each sentence denoting the probability with which it will get picked up in the summary.

3. Produces a summary based on the top k most important sentences. Some studies have used Latent semantic analysis (LSA) to identify semantically important sentences.

Each sentence is not capable of generating questions. Only the sentences that contain a questionable fact can act as a candidate for creating MCQs. Therefore, sentence selection plays a crucial role in the automatic MCQ generation task. Hence for summarizing the text, BERT and GPT-2 models are used.

(a) BERT MODEL

BERT (Bidirectional Encoder Representations from Transformers) is a neural network-based technique for natural language processing.

It is a pre-trained open-sourced model from Google. It helps computers to understand the language a bit more as humans do. The input text is summarized using the BERT model, which is fine-tuned BERT for extractive summarization. The architecture of BERT is shown in the Fig- 3.1.

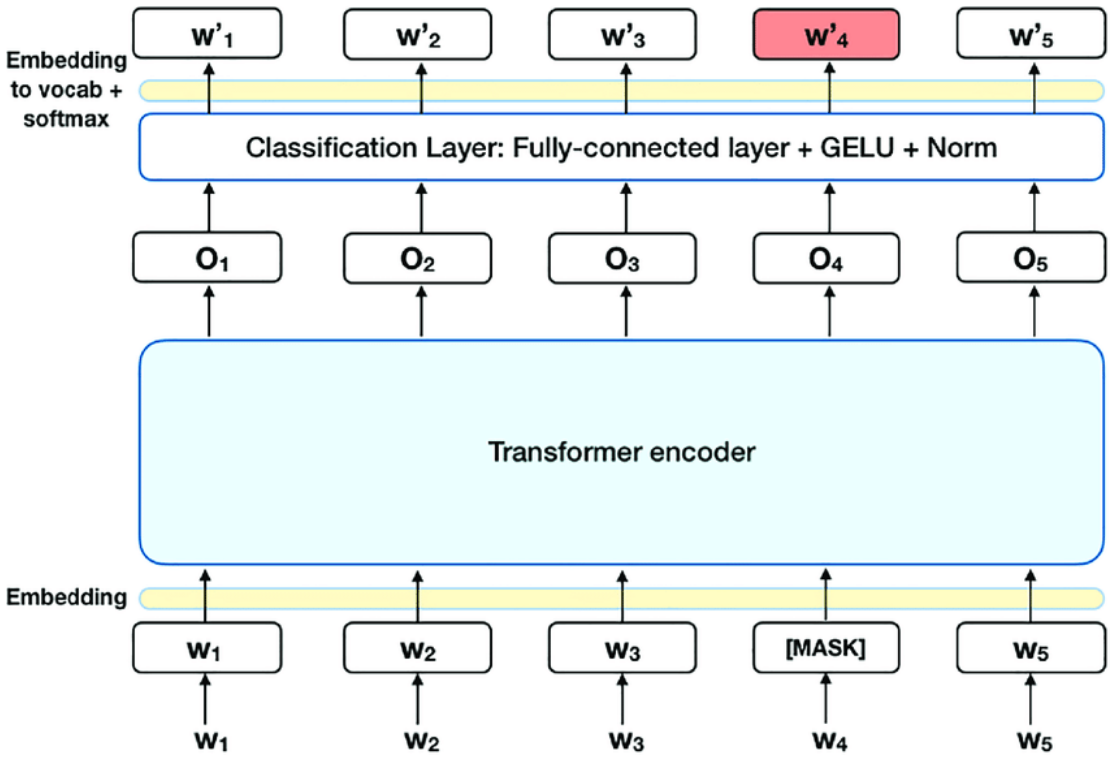

Fig. 3.1: Architecture of BERT model

In the BERTSUM model, at the start of each sentence, a [CLS] token is added, and between every two sentences, a [SEP] token is added to separate the sentences. Here, a [CLS] token is added to collect the preceding sentence context. Now each word of the sentence is tokenized using token embeddings. There is also a difference in segment embeddings. In the BERTSUM model, each sentence is assigned an embedding of Ea or Eb depending on whether the sentence is even or odd. If the sequence is [s1, s2, s3] then the segment embeddings are [Ea, Eb, Ea].

This way, all sentences are embedded and sent into further layers. BERTSUM assigns scores to each sentence that represents how much value that sentence adds to the overall document.

So, [s1, s2, s3] is assigned [score1, score2, score3]. The sentences with the highest scores are then collected and rearranged to summarize the input text. In the output, the precise and summarized text is generated.

(a) GPT-2 MODEL

The OpenAI GPT-2 exhibits impressive ability of writing coherent and passionate essays that exceeds what the anticipated cursrent language models are able to produce. Its architecture is very similar to the decoder-only transformer. The GPT-2 was, however, a very large, transformer-based language model trained on a massive dataset. The model takes a predefined starting token as input and outputs only one token at a time. After each new token is generated, the token is added after the previously generated token sequence, and this sequence will become the new input for the next step of the model. One needs to specify the length of the build or the end flag to complete the build task. Then the model ends the generation of the summary according to the predefined length or flag, and finally outputs a generation sequence. The architecture of GPT-2 is shown in the Fig 3.2.

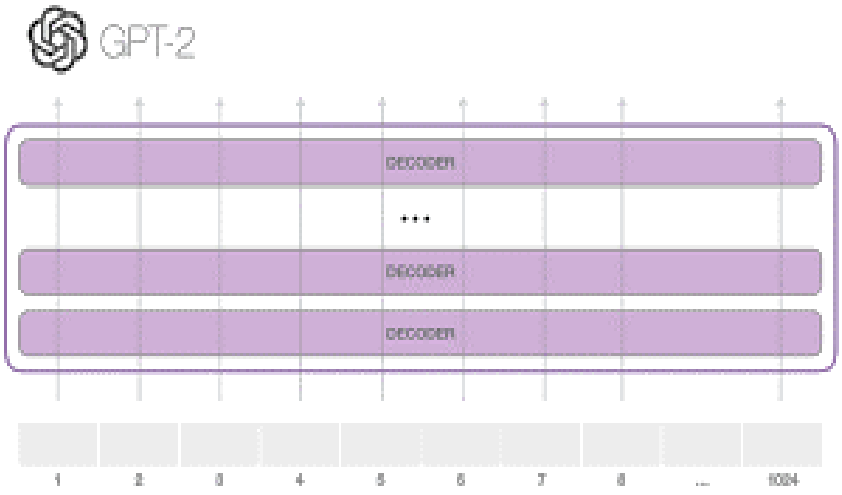

Fig 3.2: Architecture of GPT-2 model

III. KEYWORD EXTRACTON (PKE)

Keywords are chosen from the sentence after the text has been summarized. The answer to the question will be this keyword. Because all of the words in an informative sentence cannot be used as the key, keyword selection is necessary. MULTIPARTITERANK, a Python library, is used to extract keywords.

Multipartiterank is a keyword extraction model that is based on topics. It uses a multipartite graph structure to encode the topical information of a document. A complete directed multipartite graph is built, in which nodes are key phrase candidates that are connected only if they belong to different topics. This approach uses the mutually reinforcing link between candidate topics and candidate keywords to improve candidate ranking by representing the candidate keywords and subjects of a document in a single graph. This method consists of two steps for selecting candidate words as keywords: (i) graphing the entire document and (ii) assigning relevance score to each word. Then, the position information is captured between these two steps by adjusting edge weights. As a result, it outperforms various other key-extraction techniques the majority of the time.

IV. SENTENCE MAPPING

After the selection of a keyword, a sentence is mapped for each keyword, i.e. for each keyword, a sentence with the word from the summarized text is extracted.

V. MCQs GENERATION

The most important step in creating automated MCQs is generating distractors. The quality of distractors produced has a significant impact on the difficulty level of MCQs. A good distractor is the one that looks a lot like the key but isn't the key. As a result, the Wordnet approach is used to generate distractors.

Princeton created WordNet, a lexical database for the English language that is part of the NLTK corpus. The words in the WordNet network are linked by linguistic relations. The word (or sense) should be distinct from other synsets already in the WordNet. Hypernym, hyponym, meronym, holonym, and other linguistic relationships are examples of it.

Synsets (verbs, nouns, adjectives, and adverbs are grouped into sets of rational synonyms) are used by WordNet to store synonyms. Each word in a synset has the same meaning. Each synset is essentially a collection of synonyms. A definition is associated with each synset and the relationships between synsets are stored. This lesk algorithm takes the position that all words in a given "neighborhood" (section of text) have the same title. The dictionary definition of an ambiguous word is compared to the terms in its neighbourhood using a simplified version of the Lesk algorithm. For example, if the keyword "bat" appears in a sentence like "The bat flew into the jungle and landed on a tree," Here it denotes a mammalian bat with wings, not a cricket or baseball bat. Although the humans are adept at it, algorithms struggle to distinguish one from another. The process is known as Word Sense Disambiguation (WSD). In the wordnet, "bat" can refer to a cricket bat, a flying mammal, and other things. As a result, the function get word sense attempts to determine the correct meaning of the word in the sentence. Once the word sense has been determined, the get_distractors_wordnet function is used to retrieve the distractors.

This function uses the key's hypernyms and hyponyms to try to find the distractors. Hypernym - A hypernym is a word that refers to a broad category of words. Animal, for example, is a hypernym for dog.

A hyponym is a word that has a more specific meaning than a term that applies to it. For instance, car is a hyponym for vehicle.

All of the key's possible hypernyms and their corresponding hyponyms have now been discovered. These hyponyms have the potential to be distractors. The potential distractors are then ranked, and the final distractors are selected based on their position. The hypernym for the word 'bat' as the key is - an animal whose hyponyms include an eagle and other birds, which can serve as good distractors for the key. The potential distractors are ranked according to whether they appear in the text extracted from the summarization. Potential distractors with the same part of speech structure as the key have a higher rank than those with a different structure in the extracted text.

This is due to the fact that distractors with the same structure as the key tend to cause more confusion among test takers. As the final distractors, any three potential distractors with higher ranks are chosen at random.

VI. PERFORMANCE EVALUATION

This is last stage of MCQ Generation, in which the evaluation of text summaries is typically conducted experimentally, rather than analytically.

The experimental evaluation of summaries, rather than concentrating on issues of efficiency, usually tries to evaluate the effectiveness of summaries, i.e. its capability of taking the right categorization decisions.

(a) ROUGE SCORE

The evaluation of the quality of a generated summary is a key point in summarization research. Traditionally evaluation of summarization involves human judgments of different quality metrics, for example, coherence, conciseness, grammaticality, readability, and content. However, even simple manual evaluation of summaries on a large scale over a few linguistic quality questions and content coverage as in the Document Understanding Conference (DUC) would require over 3,000 hours of human efforts. This is very expensive and difficult to conduct in a frequent basis.

Therefore, how to evaluate summaries automatically has drawn a lot of attention in the summarization research community in recent years. The three content-based evaluation methods that measure similarity between summaries. These methods are: cosine similarity, unit overlap (i.e. unigram or bigram), and longest common subsequence. However, they did not show how the results of these automatic evaluation methods correlate to human judgments.

To overcome these shortcomings, here employed a package, ROUGE, for automatic evaluation of summaries. ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It includes measures to automatically determine the quality of a summary by comparing it to other (ideal) summaries created by humans.

Here the ROUGE-scores) using different settings for the ROUGE evaluation software and for the summarizer are presented. The agreements between the human written model summaries are also reported. Here, that the Rouge values, both the Rouge evaluation system and human judges gave a better score to human summaries than the automatically produced summaries. ROUGE is a set of metrics for evaluating automatic summarization of texts and machine translations.

There are several variants of the ROUGE metric; however, the basic idea behind them is to allocate a single numerical score to a summary that tells us how good it is compared to one or more reference summaries.

Here, the summaries generated by the BERT model and GPT-2 model are compared for its effectiveness. Depending on the above ROUGE metrics, BERT model is produced the best summary.

3.1.2 FREQUENTLY ASKED QUESTIONS

I. LOAD THE RAW TEXT

The first step is to load raw text i.e. input text of any domain for which the questions to be generated.

II. ABSTRACTIVE SUMMARIZATION

Abstractive summarization is an efficient form of summarization compared to extractive summarization as it retrieves information from multiple documents to create precise summary of information. This has gained its popularity due to the ability of developing new sentences to tell the important information from text documents. An abstractive summarizer displays the summarized information in a coherent form that is easily readable and grammatically correct. Readability or linguistic quality is an important catalyst for improving the quality of a summary.

Each sentence is not capable of generating questions. Only the sentences that contain a questionable fact can act as a candidate for creating FAQs. Therefore, sentence selection plays a crucial role in the automatic FAQs generation task. Hence for summarizing the text, T5 and BART models are used.

(a) T5 MODEL

T5 (Text-To-Text Transformer) is the abbreviation for "Text-to-Text Transfer Transformer". The idea behind the T5 model is transfer learning. The model was initially trained on a task containing large text in Transfer Learning before it was finely tuned on a downstream task so that the model learns general-purpose skills and information to be applied to tasks such as summarization

The model used for the summarization is built through a transfer learning process on a T5 model. The choice of T5 is justified by the fact it provides a better generalization. Thus, the T5 model is more suitable.

In addition, the model can further be used for other aims such as translation and answers to short questions. T5 is based on transformers. A transformer is composed of two types of layers: encoder and decoder layers. An encoder layer aims to encode long sequences input into a numerical form through the attention mechanism, while a decoder layer aims to output a summary by using the encoded information from the encoder layers.

Transformers try to solve the parallelization issue by using Convolutional Neural Networks (CNNs) together with attention models. Each encoder has two types of layers: multi-head self-attention layer and position-wise fully connected feed-forward network. In addition to both previous layers, a decoder also includes a masked multi-head self-attention layer. The architecture of Transformer is shown in the Fig- 3.3.

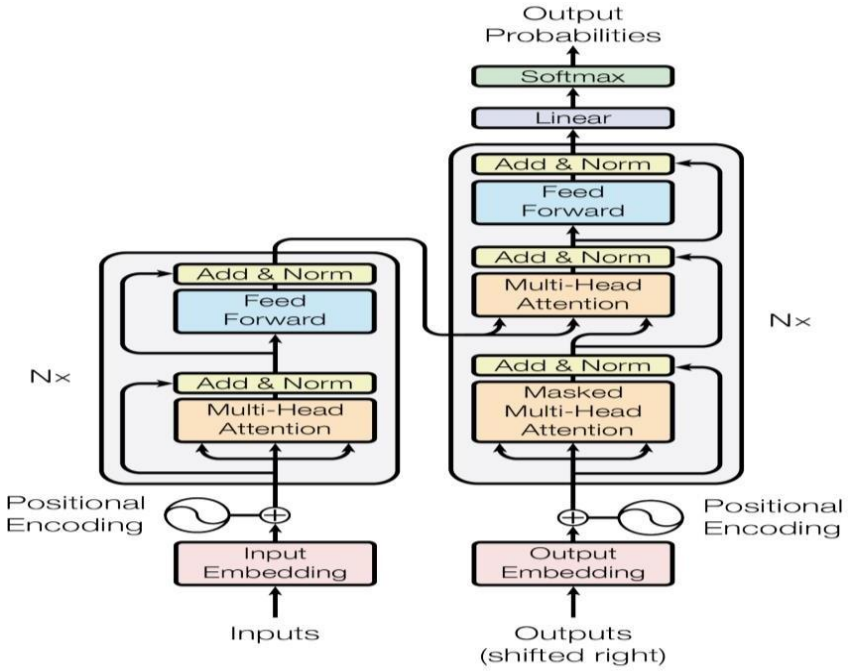

Fig 3.3: The Transformer Model Architecture

(b) BART MODEL

BART (Bidirectional and Auto-Regressive Transformer) is trained by corrupting documents and then optimizing a reconstruction loss the cross-entropy between the decoder‘s output and the original document. Unlike existing denoising autoencoders, which are tailored to specific noising schemes, BART allows to apply any type of document corruption. In the extreme case, where all information about the source is lost, BART is equivalent to a language model. Token Deletion Random tokens are deleted from the input. In contrast to token masking, the model must decide which positions are missing inputs.

Text Infilling A number of text spans are sampled, with span lengths drawn from a Poisson distribution ($\lambda = 3$). Each span is replaced with a single [MASK] token. 0- length spans correspond to the insertion of [MASK] tokens. Text infilling is inspired by SpanBERT, but SpanBERT samples span lengths from a different (clamped geometric) distribution, and replaces each span with a sequence of [MASK] tokens of exactly the same length. Text infilling teaches the model to predict how many tokens are missing from a span.

Sentence Permutation A document is divided into sentences based on full stops, and these sentences are shuffled in a random order. Document Rotation A token is chosen uniformly at random, and the document is rotated so that it begins with that token. This task trains the model to identify the start of the document. The architecture of BART is shown in the Fig- 3.4.

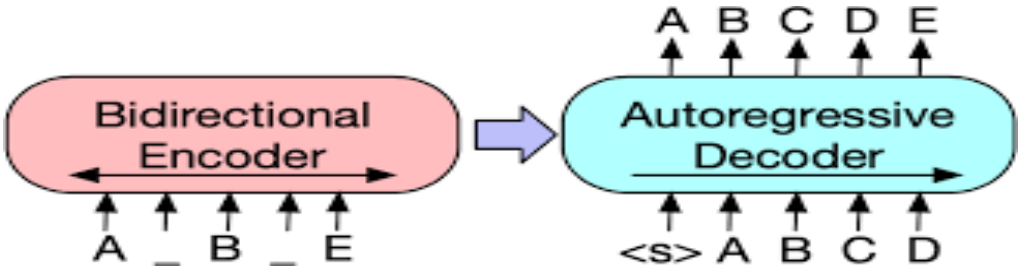

Fig 3.4: The BART model architecture

III. KEYWORD EXTRACTON (PKE)

Keywords are chosen from the sentence after the text has been summarized. The answer to the question will be this keyword. Because all of the words in an informative sentence cannot be used as the key, keyword selection is necessary. MULTIPARTITERANK, a Python library, is used to extract keywords.

Multipartiterank is a keyword extraction model that is based on topics. It uses a multipartite graph structure to encode the topical information of a document. A complete directed multipartite graph is built, in which nodes are key phrase candidates that are connected only if they belong to different topics.

This approach uses the mutually reinforcing link between candidate topics and candidate keywords to improve candidate ranking by representing the candidate keywords and subjects of a document in a single graph. This method consists of two steps for selecting candidate words as keywords: (i) graphing the entire document and (ii) assigning relevance score to each word.

Then, the position information is captured between these two steps by adjusting edge weights. As a result, it outperforms various other key-extraction techniques the majority of the time.

IV. LOAD THE PRE-TRAINED MODEL

The T5 reframing all NLP tasks into a unified text-to-text-format where the input and output are always text strings, in contrast to BERT-style models that can only output either a class label or a span of the input. T5 text-to-text framework allows us to use the same model, loss function, and hyper parameters on any NLP task, including machine translation, document summarization, question answering, and classification tasks (e.g., sentiment analysis). T5 also apply to regression tasks by training it to predict the string representation of a number instead of the number itself. The pre-trained model T5 is loaded for question generation in this step. The Fig 3.5 represents the T5 question generation framework.

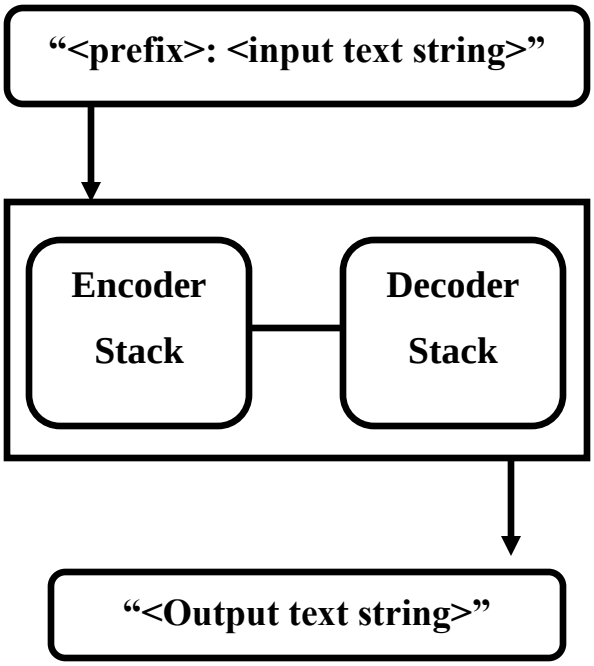

Fig 3.5: Question Generation Framework

V. FAQs GENERATION

In this step T5 transformer used for the question generation and easy to use and understand frequently asked questions generation algorithm using T5 Transformers. The system accepts a short passage of text and uses two fine-tuned T5 Transformer model to generate frequently asked questions pairs corresponding to the given text.

VI. PERFORMANCE EVALUATION

This is last stage of FAQs Generation, in which the evaluation of text summaries is typically conducted experimentally, rather than analytically. The experimental evaluation of summaries, rather than concentrating on issues of Efficiency, usually tries to evaluate the effectiveness of summaries, i.e. its capability of taking the right categorization decisions. Here, the summaries generated by the T5 model and BART model are compared for its effectiveness. Depending on the above ROUGE metrics, T5 model is produced the best summary.

EXPERIMENTAL RESULTS AND DISCUSSION

4. EXPERIMENTAL RESULTS AND DISCUSSION

This chapter explains experiment carried out using the designed methodology. The sample data taken for this work, the software used to implement this project and the evaluation metrics with observed results with charts are discussed in this chapter.

4.1 Data set

The data that are to be inserted plays an important role. In order to get the meaningful output, the input should be acceptable and understandable. To implement this project sample text data had taken and tested.

4.2 Experimental setup

The implementation of automatic question generation using python and a series of experiments were performed on a PC with Windows XP Operating system at 1.8 GHz dual core PC machine with 4 GB main memory running a 64-bit version of Windows 2010.

4.2.1 Python Environment

The project is developed in python 3.8 using appropriate package in anaconda environment.

(i) Anaconda

It is a free and open source distribution of the Python and R programming languages for data science and machine learning related applications (large-scale data processing, predictive analytics, scientific computing), that aims to simplify package management and deployment. Package versions are managed by the package management system conda. The Anaconda distribution is used by over 6 million users, and it includes more than 250 popular data science packages suitable for Windows, Linux, and macos.

(ii) Jupyter Notebook

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text.

Jupyter Notebook is maintained by the people at Project Jupyter. Jupyter notebooks basically provide an interactive computational environment for developing Python based Data Science applications. They are formerly known as ipython notebooks. The following are some of the features of Jupyter notebooks that makes it one of the best components of Python ML ecosystem –

- Jupyter notebooks can illustrate the analysis process step by step by arranging the stuff like code, images, text, output etc. in a step by step manner.
- It helps a data scientist to document the thought process while developing the analysis process.
- One can also capture the result as the part of the notebook.

4.2.2 Python Packages

There were a number of different Python packages and modules used for different areas of functionality across this project. Below are a list and a description of the packages.

- **GENSIM** is a Python library for topic modelling, document indexing and similarity retrieval with large corpora. it is designed to handle large text collections using data streaming and incremental online algorithms, which differentiates it from most other machine learning software packages that target only in-memory processing.

- **SCIKIT**-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy .

- **PKE** is an **open source** python-based **keyphrase extraction** toolkit. It provides an end-to-end keyphrase extraction pipeline in which each component can be easily modified or extended to develop new models. pke also allows for easy benchmarking of state-of-the-art keyphrase extraction models.
- **SPACY** is a free and open-source library for **Natural Language Processing** (NLP) in Python with a lot of in-built capabilities. It‘s becoming increasingly popular for processing and analyzing data in NLP. Unstructured textual data is produced at a large scale, and it‘s important to process and derive insights from unstructured data. To do that, one need to represent the data in a format that can be understood by computers.
- **NLTK** consists of the most common algorithms such as tokenizing, part-of-speech tagging, stemming, sentiment analysis, topic segmentation, and named entity recognition. NLTK helps the computer to analysis, preprocess, and understand the written text.
- **FLASHTEXT** is a Python library created specifically for the purpose of searching and **replacing** words in a document. Now, the way FlashText works is that it requires a word or a list of words and a string. The words which FlashText calls **keywords** are then searched or replaced in the string.
- **A TRANSFORMER** library is provides thousands of pre-trained models to perform tasks on different modalities such as text, vision, and audio.

4.3 Evaluation Metric

The experimental evaluation of summaries, rather than concentrating on issues of efficiency, usually tries to evaluate the effectiveness of summaries, i.e. its capability of taking the right categorization decisions.

4.3.1 ROUGE-1

It compares the uni-grams between the machine-generated summary and the human reference summary. ROUGE-1 metric has separate RECALL and PRECISION. ROUGE-1 recall is the ratio of the number of words that match (in machine generated summary and the human reference summary) and the number of words in the reference. In this case, ROUGE-1 recall will be 1. It has a perfect recall, but the summary is terrible.

Therefore, precision is also needed. ROUGE-1 precision is the ratio of the number of words that match and the number of words in the summary. Finally, the calculation of harmonic mean of the two is known as the F1 score.

F1-score = (2 x precision x recall) / (precision + recall)

4.3.2 ROUGE-2

It compares the bi-grams between the machine-generated summary and the human reference summary. ROUGE-2 metric has separate RECALL and PRECISION for the ROUGE-2 metric. ROUGE-2 recall is the ratio of the number of words that match (in machine generated summary and the human reference summary) and the number of words in the reference. In this case, ROUGE-2 recall will be 1. It has a perfect recall, but the summary is terrible. Therefore, precision is need precision as well. ROUGE-2 precision is the ratio of the number of words that match and the number of words in the summary. . Finally, the calculation of harmonic mean of the two is known as the F1 score.

F1-score = (2 x precision x recall) / (precision + recall)

4.2.3 ROUGE-L

ROUGE-L doesn‘t compare n-grams; instead treats each summary as a sequence of words and then looks for the longest common subsequence (LCS). The ROUGE-L precision is the ratio of the length of LCS and the number of words in the generated summary. The advantage of using ROUGE-L over ROUGE-1 and ROUGE-2 is that it doesn‘t depend on consecutive n-gram matches, and it tends to capture sentence structure much more accurately.

4.4 Observed Results

A. Multiple Choice Questions

The project's key component is the creation of a summary. MCQ questions were produced based on this summary. The results and accuracy of the proposed system are comparatively higher than other systems. Here the BERT model used for summarizing, which outperforms than GPT-2 model. As shown in Table- 1, all BERT-based models outperformed previous state-of-the-art models by a large margin.

BERT with Transformer achieved the best performance on all three metrics. The BERT model does not have an obvious influence on the summarization performance than the GPT-2 model. So among all the models, here the BERT model is chosen for the text summarization which has the best score compared to the GPT-2.

Table 1: Performance metrics of summarization models (MCQ)

MODELS	ROUGE-1	ROUGE-2	ROUGE-L
GPT-2	0.51852	0.5	0.51852
BERT	0.83636	0.79245	0.83636

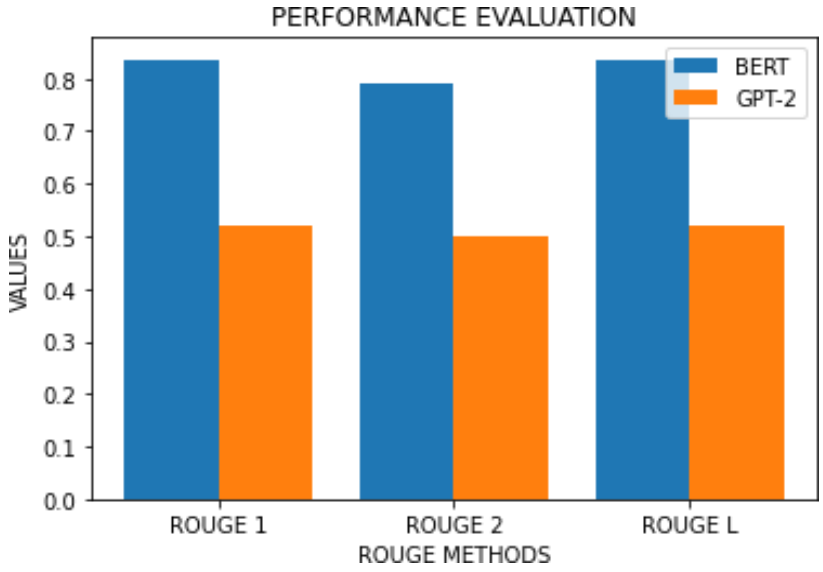

Fig 4.1: Comparison of summarization models

The above fig 4.1 shows that the ROUGE scores of BERT and GPT-2. This is applied on tested text data. Among these two pre-trained models BERT model has highest accuracy when compare to other model.

B. Frequently Asked Questions

The project's key component is the creation of a summary. FAQ questions were produced based on this summary. The results and accuracy of the proposed system are comparatively higher than other systems. Here the T5 model used for summarizing, which outperforms than BART model. As shown in Table- 2, all T5-based models outperformed previous state-of-the-art models by a large margin. T5 with Transformer achieved the best performance on all three metrics. The T5 model does not have an obvious influence on the summarization performance than the BART model. So among all the models, here the T5 model is chosen for the text summarization which has the best score compared to the BART.

Table 2: Performance metrics of summarization models (FAQ)

MODELS	ROUGE-1	ROUGE-2	ROUGE-L
BART	0.4	0.29167	0.36
T5	0.49057	0.39216	0.45283

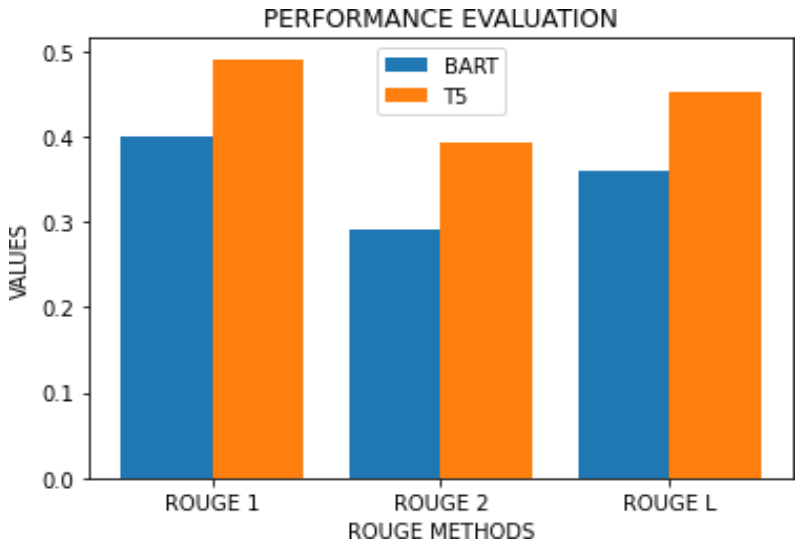

Fig 4.2: Comparison of summarization models

The Fig 4.2 shows that the ROUGE scores of summarization models. This is applied on tested text data. Among these two pre-trained models T5 model has highest accuracy when compare to other model.

CONCLUSION

5. CONCLUSION

Questions generation using Natural Language Processing (NLP) techniques is a system that helps teachers set multi-choice questions and Frequently Asked Questions from their documents in a text file and thereafter provides answers to the questions generated. Consequently, it can be used in every area of education sector to test the knowledge or level of the students.

Most of the algorithms mainly based on natural language processing. The work has basically been donated for the English language for the generation of questions which can be categories as Multiple Choice, Frequently Asked questions. What it focuses on is on application of NLP techniques to generate questions about facts explicitly based on keywords as stated in a text format of a given lesson material. From the results, it has shown that the system was capable of extracting keywords from lesson materials in setting examinable questions.

The creation of a summary is an important part of the project. This summary was used to create FAQ and MCQ questions. As a result, the ROUGE is used to do a summary evaluation of these circumstances in order to construct the question correctly. The proposed system solves the problem of manually creating questions. The suggested system uses natural language processing (NLP) to generate automated queries, reducing human participation and making it a cost- and time-effective method. This technique benefits both teachers and students who are preparing for competitive exams by assisting them with E-assessments. Students can put their problem-solving skills to the test as well as assess their knowledge of the ideas.

SCOPE FOR FUTURE ENHANCEMENT

6. SCOPE FOR FUTURE ENHANCEMENT

After successful implementation of this work, in future it will lead to Question Generation from entire documents as collections. After that the current system algorithm with minor improvement can be deployed. The ultimate system will be able to independently handle any pdf or word or any other type of text file, analyze it and find important sentences for QG. From those sentences a variety of questions could be formed with minimum inaccuracy. Also, the accuracy of generating possible multiple choice questions can be enhanced by adding similar meaning words from the context itself rather than the entire vocabulary. This will not only reduce the search space but also reduce the time elapse in producing desired results. One can even add functionalities to export the question set as some normalized output format such as csv or pdf that could be further used in printing or displaying the generated questions.

BIBLIOGRAPHY

BIBLIOGRAPHY

REFERENCES

[1] Santhanavijayan, A., Balasundaram, S.R., Hari Narayanan, S., Vinod Kumar, S., and Vignesh Prasad, V. (2017) _Automatic generation of multiple-choice questions for e-assessment', Int. J. Signal and Imaging Systems Engineering, Vol. 10, Nos. 1/2, pp.54–62.

[2] Ayako Hoshino and Hiroshi Nakagawa (2005) –A real-time multiple-choice question generation for language testing: A preliminary study‖, Ed Apps NLP 05: Proceedings of the second workshop on Building Educational Applications Using NLP.

[3] D. R. CH and S. K. Saha, –Automatic Multiple Choice Question Generation From Text: A Survey,‖ in IEEE Transactions on Learning Technologies, vol. 13, no. 1, pp. 14-25,1Jan.-March2020, doi: 10.1109/TLT.2018.2889100.

[4] Deepshree S. Vibhandik, Rucha C. Samant –Automatic / Smart Question Generation System for Academic Purpose‖, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 4, Issue 4, July - August 2015.

[5] Susanti, Y., Tokunaga, T., Nishikawa, H., and Obari, H. 2018. _'Automatic distractor generation for multiple Choice English vocabulary questions‖. Research and Practice in Technology Enhanced Learning, 13(1), 15.

[6] Onur KEKL –―Automatic Question Generation Using Natural Language Processing Techniques, July 2018. https://pdfs.semanticscholar.org/ec5e/fc74351f0339e34b91b965f99624aedf9200.pdf

[7] Amruta Umardand, Ashwini – –A survey on Automatic Question Paper Generation System‖, International Advanced Research Journal in Science, Engineering and Technology (IARJSET), Jan 2017.

[8] Aleena, Vidya – Implementation of Automatic Question Paper Generator System‖, International Research Journal of Engineering and Technology (IRJET), Feb 2019.

[9] Kalpana B. Khandale1, Ajitkumar Pundage, C. Namrata Mahender – –Similarities In Words Using Different Pos Taggers.‖, IOSR Journal of Computer Engineering (IOSRJCE),(PP 51-55).

[10] Edward Loper and Steven Bird – Nltk: The Natural Language Toolkit‖, July 2002.

[11] Ankita, K. A. Abdul Nazeer – Part-Of-Speech Tagging And Named Entity Recognition Using Improved Hidden Markov Model And Bloom Filter‖, International Conference on Computing, Power and Communication Technologies (GUCON), 2018.

[12] Mohd Shahid Husain – Automatic Question generation from Text‖, International Journal for Innovations in Engineering, Science and Management (IJIESM), April 2015.

[13] Pranita Jadhav, Manjushree Laddha – An Automatic Gap Filling Questions Generation using NLP‖, International Journal of Computer Science & Engineering Technology (IJCSET), Aug 2017.

[14] Bowen Xu, Zhenchang Xing, Xin Xia, Davi Lo - Answer Bot: Automated Generation of Answer Summary to Developers‘ Technical Questions‖.

[15] Priti G, Shreya J, Srushtee, Subhangi – Automated Question Generator System: A Review‖, International Journal of Engineering Applied Science and Technology (IJEAST), Dec 2019, Vol. 4, Issue 8,

[16] Surbhi, Abdul, Shrutika, Kavita – Question Paper Generator System‖, International Journal of Computer Science Trends and Technology (IJCST), Oct 2015.

[17] Sheetal, Dr. Y R Ghodasara –‒Literature Review of Automatic Question Generator Systems‖, International Journal of Scientific and Research Publications (IJSRP), Jan 2015.

[18] Susanti, Y., Iida, R., Tokunaga, T. (2015). Automatic generation of english vocabulary tests, In Proceedings of the 7th International Conference on Computer Supported Education (pp. 77–87). Setubal: INSTICC.

[19] Das, B., Majumder, M., Phadikar, S., Sekh, A.A. (2019). Automatic generation of fill-in-the-blank question with corpus-based distractors for e-assessment to enhance learning. Computer Applications in Engineering Education, 27(6), 1485–1495.

[20] ArikIturri et.al., An Automatic Question Generator Based on Corpora and NLP Techniques Conference: Intelligent Tutoring Systems, 8th International Conference, ITS 2006, Jhongli, Taiwan, June 26-30, 2006.

[21] Susanti, Y., Tokunaga, T., Nishikawa, H., Obari, H. (2017). Evaluation of automatically generated english vocabulary questions. Research and Practice in Technology Enhanced Learning, 12(1), 22.

[22] Ashok Immanuel and Tulasi. B, –Framework for Automatic Examination Paper Generation System, International Journal of Computer Science Trends and Technology, vol. 6, issue 1, Jan - March 2015.

[23] Surbhi Choudhary, Abdul Rais Abdul Waheed. Shrutika Gawandi and Kavita Joshi, –Question Paper Generator System,‖ International Journal of Computer Science Trends and Technology, vol. 3, issue 5, Sept –Oct 2015.

[24] Egonmwan et al. Transformer-based Model for Single Documents Neural Summarization‖. Proceedings of the 3rd Workshop on Neural Generation and Translation (WNGT 2019),

[25] Mazidi, Karen. "Automatic Question Generation From Passages." In International Conference on Computational Linguistics and Intelligent Text Processing, pp. 655-665. Springer, Cham, 2017.

APPENDIX

APPENDIX

A. WORK FLOW DIAGRAM

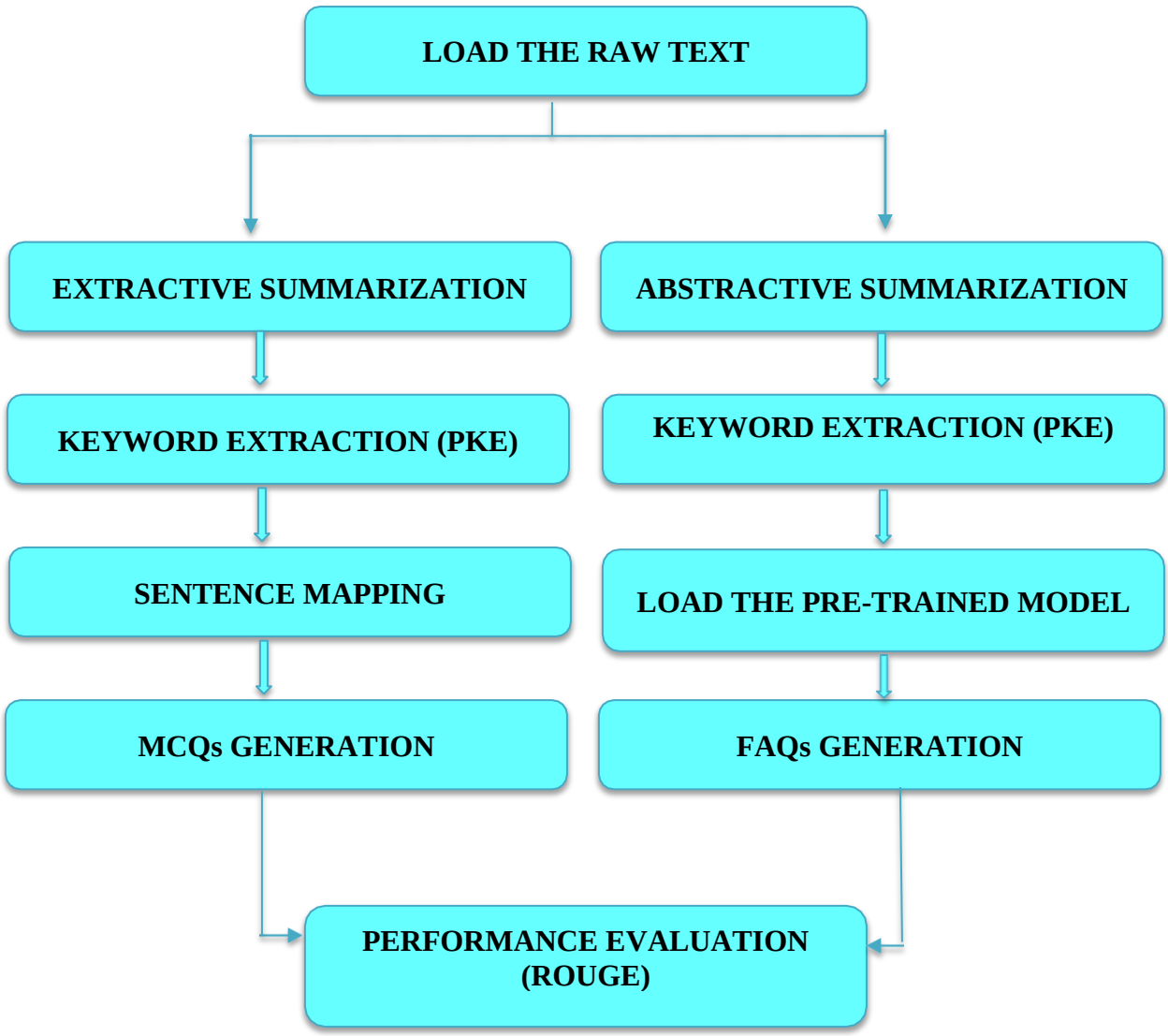

Fig 1: Overall Methodology Diagram

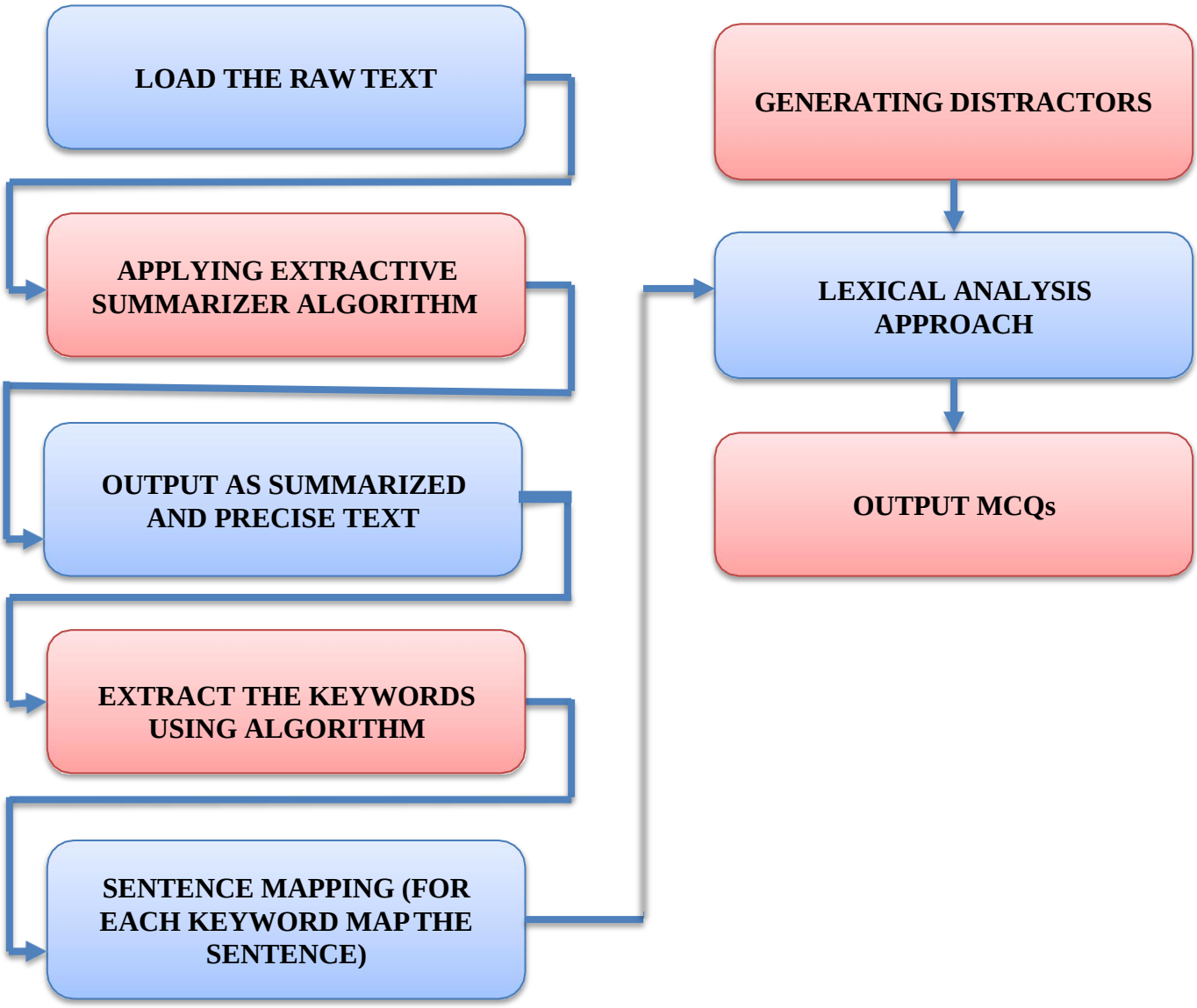

Fig 2: Architecture Diagram of MCQ

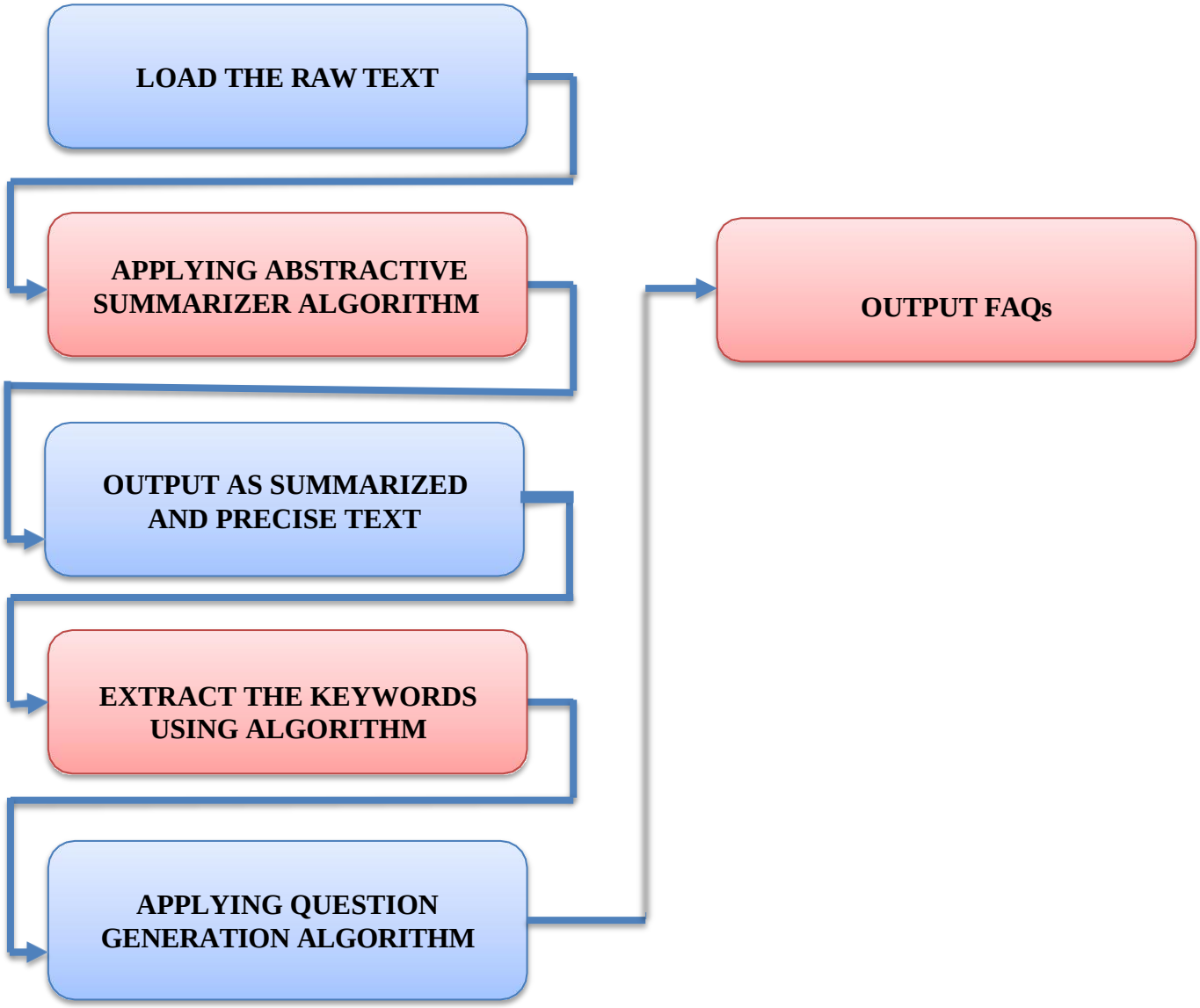

Fig 3: Architecture Diagram of FAQ

B.SAMPLE INPUT TEXT

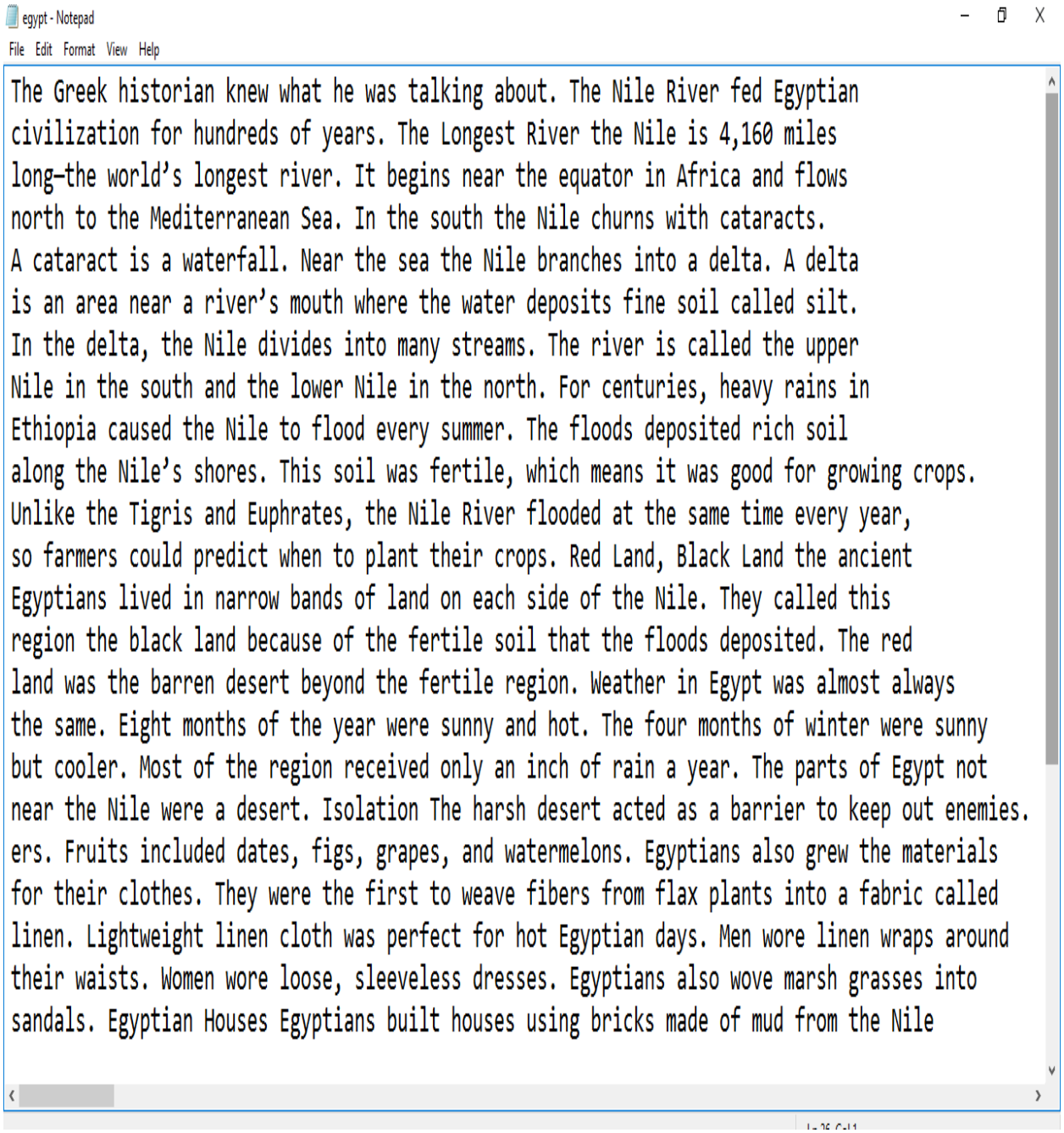

Fig 4: Load the raw text

C.SCREENSHOTS

MULTIPLE CHOICE QUESTIONS

```
[ ] from summarizer import Summarizer

    f = open("volcano.txt","r",encoding='unicode_escape')
    full_text = f.read()

    model = Summarizer()
    result = model(full_text, min_length=60, max_length = 1130 , ratio = 0.4)

    summarized_text = ''.join(result)
    print (summarized_text)
```

```
Downloading: 100% 571/571 [00:00<00:00, 13.0kB/s]
Downloading: 100% 1.25G/1.25G [00:35<00:00, 39.2MB/s]
Some weights of the model checkpoint at bert-large-uncased were not used when initializing BertModel: ['cls.predictions.decoder.weight'
- This IS expected if you are initializing BertModel from the checkpoint of a model trained on another task or with another architectur
- This IS NOT expected if you are initializing BertModel from the checkpoint of a model that you expect to be exactly identical (initia
Downloading: 100% 226k/226k [00:00<00:00, 4.79MB/s]
Downloading: 100% 28.0/28.0 [00:00<00:00, 230B/s]
The Nile River fed Egyptian civilization for hundreds of years. It begins near the equator in Africa and flows north to the Mediterrane
time: 1min 38s (started: 2022-04-07 06:27:39 +00:00)
```

Summarized text: The Nile River fed Egyptian civilization for hundreds of years. It begins near the equator in Africa and flows north to the Mediterranean Sea. This soil was fertile, which means it was good for growing crops. The red land was the barren desert beyond the fertile region. Isolation The harsh desert acted as a barrier to keep out enemies. When the birds arrived, the annual flood waters would soon follow. They were the first to grind wheat into flour and to mix the flour with yeast and water to make dough rise into bread. They grew vegetables such as lettuce, radishes, asparagus, and cucumbers. Egyptians often painted walls white to reflect the blazing heat. They were probably the first people in the world to mine turquoise. One ancient painting even shows a man ready to hit a catfish with a wooden hammer. A boomerang is a curved stick that returns to the person who threw it.

Fig 5: Summarized text of BERT (MCQ)

```
[ ] from summarizer import TransformerSummarizer
    f = open("egypt.txt","r",encoding='unicode_escape')
    full_text = f.read()
    GPT2_model = TransformerSummarizer(transformer_type="GPT2",transformer_model_key="gpt2-medium")
    summarized_text = ''.join(GPT2_model(full_text, min_length=60))
    print(summarized_text)
```

```
Downloading: 100% 718/718 [00:00<00:00, 14.4kB/s]
Downloading: 100% 1.42G/1.42G [00:36<00:00, 47.5MB/s]
Downloading: 100% 0.99M/0.99M [00:00<00:00, 10.6MB/s]
Downloading: 100% 446k/446k [00:00<00:00, 2.92kB/s]
The Nile River fed Egyptian civilization for hundreds of years. The red land was the barren desert beyond the fertile region. After the
```

Summarized Text: The Nile River fed Egyptian civilization for hundreds of years. The red land was the barren desert beyond the fertile region. After the waters drained away, farmers could plant seeds in the fertile soil. Then they used a tool called a shaduf to spread the water across the fields. They grew vegetables such as lettuce, radishes, asparagus, and cucumbers. However, the natural resources of the area allowed other economic activities to develop too. Egyptians looked for copper as early as 6000 B.C. Later they learned that iron was stronger, and they sought it as well. Nubia was the Egyptian name for the area of the upper Nile that had the richest gold mines in Africa. Some skilled artisans erected stone or brick houses and temples. Other artisans made pottery, incense, mats, furniture, linen clothing, sandals, or jewelry.

Fig 6: Summarized text of GPT-2 (MCQ)

```
        #    alpha controls the weight adjustment mechanism, see TopicRank for
        #    threshold/method parameters.
    extractor.candidate_weighting(alpha=1.1,
                                  threshold=0.75,
                                  method='average')
    keyphrases = extractor.get_n_best(n=20)
    for key in keyphrases:
          out.append(key[0])
    return out
keywords = get_nouns_multipartite(full_text)
print (keywords)
filtered_keys=[]
for keyword in keywords:
    if keyword.lower() in summarized_text.lower():
        filtered_keys.append(keyword)

print (filtered_keys)
```

```
['egyptians', 'nile river', 'egypt', 'nile', 'euphrates', 'tigris', 'old kingdom', 'crown', 'red land', 'lower egypt', 'longest river',
```

Keywords: ['egyptians', 'nile river', 'egypt', 'nile', 'euphrates', 'tigris', 'old kingdom', 'crown', 'red land', 'lower egypt', 'longest river', 'narmer united upper', 'africa', 'mediterranean sea', 'hyksos', 'new kingdom', 'black land', 'middle kingdom', 'ethiopia', 'papyrus']

Fig 7: Extracted Keywords Using PKE (MCQ)

```
def get_sentences_for_keyword(keywords, sentences):
    keyword_processor = KeywordProcessor()
    keyword_sentences = {}
    for word in keywords:
        keyword_sentences[word] = []
        keyword_processor.add_keyword(word)
    for sentence in sentences:
        keywords_found = keyword_processor.extract_keywords(sentence)
        for key in keywords_found:
            keyword_sentences[key].append(sentence)
    for key in keyword_sentences.keys():
        values = keyword_sentences[key]
        values = sorted(values, key=len, reverse=True)
        keyword_sentences[key] = values
    return keyword_sentences

sentences = tokenize_sentences(summarized_text)
keyword_sentence_mapping = get_sentences_for_keyword(filtered_keys, sentences)

print (keyword_sentence_mapping)
```

```
{'egyptians': ['Egyptians cut the stems into strips, pressed them, and dried them into sheets that could be rolled into scrolls.',
```

Sentence Mapping:{'egyptians': ['Egyptians cut the stems into strips, pressed them, and dried them into sheets that could be rolled into scrolls.', 'The Nile provided so well for Egyptians that sometimes they had surpluses, or more goods than they needed.', calendar.']

Fig 8: Sentence Mapping (MCQ)

RESULTS OF MCQS

1) The Nile provided so well for _____ that sometimes they had surpluses, or more goods than they needed.

a) Egyptians
b) Bantu
c) Algerian
d) Angolan

More options: ['Basotho', 'Beninese', 'Berber', 'Black African', 'Burundian', 'Cameroonian', 'Carthaginian', 'Chadian', 'Chewa', 'Congolese', 'Djibou tian', 'Egyptian', 'Ethiopian', 'Eurafrican', 'Ewe', 'Fulani']

2) As in many ancient societies, much of the knowledge of _____ came about as priests studied the world to find ways to please the gods.

a) Egypt
b) Malawi
c) East Africa
d) Somalia

More options: ['Togo', 'Zimbabwe', 'Gabon', 'Ghana', 'Lake Tanganyika', 'Ottoman Empire', 'Mozambique', 'Iran', 'Israel', 'Saudi Arabia', 'Lebanon', 'Turkey', 'Iraq', 'Levant', 'Syria', 'Jordan']

3) The _____ provided so well for Egyptians that sometimes they had surpluses, or more goods than they needed.

a) Port Sudan
b) Omdurman
c) Nile
d) Nyala

More options: ['Khartoum', 'Nubian Desert', 'Darfur', 'Libyan Desert', 'Kordofan', 'Gulu', 'Buganda', 'Entebbe', 'Jinja', 'Lake Edward', 'entebbe', 'gulu', 'kayunga', 'Upper Egypt', 'Suez Canal', 'Aswan High Dam']

4) It combined the red _____ of Lower Egypt with the white _____ of Upper Egypt.
a) Head
b) Capital
c) Masthead
d) Crown

More options: []

5) It combined the red Crown of Lower Egypt with the white Crown of _____ Egypt.
a) Upper Berth
b) Lower Berth
c) Upper

More options: []

6) It combined the red Crown of _____ with the white Crown of Upper Egypt.
a) Upper Egypt
b) Suez Canal
c) Lower egypt
d) Libyan Desert

More options: ['Aswan High Dam', 'Eastern Desert', 'Aswan', 'Lake Nasser', 'Suez', 'Thebes', 'Sinai']

7) It begins near the equator in _____ and flows north to the Mediterranean Sea.
a) Eurasia
b) Old World
c) Australia
d) Africa

More options: []

Fig 9: Results of MCQ

```
[ ]  # initialize setting of ROUGE to evaluate ROUGE-1, 2, W and SU4


     rouge = Pythonrouge(summary_file_exist=False,
                         summary=summary, reference=reference,
                         n_gram=2, ROUGE_SU4=True, ROUGE_L=True,ROUGE_W=True,ROUGE_W_Weight=1.2,
                         recall_only=False, stemming=True, stopwords=True,
                         word_level=True, length_limit=True, length=50,
                         use_cf=False, cf=95, scoring_formula='average',
                         resampling=True, samples=1000, favor=True, p=0.5)
     score = rouge.calc_score()
     print(score)
{'ROUGE-1-R': 0.85185, 'ROUGE-1-P': 0.82143, 'ROUGE-1-F': 0.83636, 'ROUGE-2-R': 0.80769, 'ROUGE-2-P': 0.77778, 'ROUGE-2-F': 0.79245, 'R
```

Fig 10: ROUGE Score of BERT (MCQ)

```
[ ]  # initialize setting of ROUGE to evaluate ROUGE-1, 2, W and SU4


     rouge = Pythonrouge(summary_file_exist=False,
                         summary=summary, reference=reference,
                         n_gram=2, ROUGE_SU4=True, ROUGE_L=True,ROUGE_W=True,ROUGE_W_Weight=1.2,
                         recall_only=False, stemming=True, stopwords=True,
                         word_level=True, length_limit=True, length=50,
                         use_cf=False, cf=95, scoring_formula='average',
                         resampling=True, samples=1000, favor=True, p=0.5)
     score = rouge.calc_score()
     print(score)
{'ROUGE-1-R': 0.5, 'ROUGE-1-P': 0.46667, 'ROUGE-1-F': 0.48276, 'ROUGE-2-R': 0.40741, 'ROUGE-2-P': 0.37931, 'ROUGE-2-F': 0.39286, 'ROUGE
```


Fig 11: ROUGE Score of GPT-2 (MCQ)

FREQUENTLY ASKED QUESTIONS

```
[ ] summarized_text = summarizer(text,summary_model,summary_tokenizer)



    #print ("\noriginal Text >>")
    #for wrp in wrap(text, 150):
      #print (wrp)
    #print ("\n")
    print ("Summarized Text >>")
    for wrp in wrap(summarized_text, 150):
      print (wrp)
    print ("\n")
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package brown to /root/nltk_data...
[nltk_data]   Unzipping corpora/brown.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Unzipping corpora/wordnet.zip.
/usr/local/lib/python3.7/dist-packages/transformers/generation_utils.py:1839: UserWarning: __floordiv__ is deprecated, and its behavior
  next_indices = next_tokens // vocab_size
Summarized Text >>
The longest river the nile is 4,160 miles long–the world's longest river. Heavy rains in ethiopia caused the river to flood every summe
rich soil along its shores. This soil was fertile, which meant it was good for growing crops. Unlike the tigris and euphrates, the floo
the same time every year, so farmers could predict when to plant.
```

Summarized Text: The longest river the nile is 4,160 miles long—the world's longest river. Heavy rains in ethiopia caused the river to flood every summer, deposited rich soil along its shores. This soil was fertile, which meant it was good for growing crops. Unlike the tigris and euphrates, the floods flooded at the same time every year, so farmers could predict when to plant.

Fig 12: Summarized text of T5(FAQ)

```
model=BartForConditionalGeneration.from_pretrained('facebook/bart-large-cnn')
inputs = tokenizer.batch_encode_plus([text],return_tensors='pt',truncation=True)
summary_ids = model.generate(inputs['input_ids'],max_length=150,min_length=20, length_penalty=2.0, num_beams=2, early_stopping=True)
summary_ids

bart_summary = tokenizer.decode(summary_ids[0], skip_special_tokens=True)
print(bart_summary)
```

```
The Nile River fed Egyptian civilization for hundreds of years. For centuries, heavy rains in Ethiopia caused the Nile to flood every s
```

Summarized Text: The Nile River fed Egyptian civilization for hundreds of years. For centuries, heavy rains in Ethiopia caused the Nile to flood every summer. The floods deposited rich soil along the Nile‘s shores. Unlike the Tigris and Euphrates, the Nile River flooded at the same time every year.

Fig 13: Summarized text of BART(FAQ)

```
[ ]   #print ("keywords unsummarized: ",keywords)
      keyword_processor = KeywordProcessor()
      for keyword in keywords:
        keyword_processor.add_keyword(keyword)

      keywords_found = keyword_processor.extract_keywords(summarytext)
      keywords_found = list(set(keywords_found))
      print ("keywords_found in summarized: ",keywords_found)

      important_keywords =[]
      for keyword in keywords:
        if keyword in keywords_found:
          important_keywords.append(keyword)

      return important_keywords[:4]


    imp_keywords = get_keywords(text,summarized_text)
    print (imp_keywords)
```

```
keywords_found in summarized:  ['river', 'farmers', 'world']
['river', 'world', 'farmers']
```

Keywords: ['river', 'farmers', 'world']['river', 'world', 'farmers']

Fig 14: Extracted Keywords using PKE (FAQ)

```
[ ]  question_model = T5ForConditionalGeneration.from_pretrained('ramsrigouthamg/t5_squad_v1')
     question_tokenizer = T5Tokenizer.from_pretrained('ramsrigouthamg/t5_squad_v1')
     question_model = question_model.to(device)
```

```
Downloading: 100% 1.21k/1.21k [00:00<00:00, 40.2kB/s]
Downloading: 100% 892M/892M [00:18<00:00, 63.6MB/s]
Downloading: 100% 792k/792k [00:00<00:00, 2.37MB/s]
Downloading: 100% 1.79k/1.79k [00:00<00:00, 54.8kB/s]
Downloading: 100% 1.86k/1.86k [00:00<00:00, 64.1kB/s]
```

Fig 15: Load the pre-trained model T5

```
[ ] for wrp in wrap(summarized_text, 150):
      print (wrp)
    print ("\n")

    for answer in imp_keywords:
      ques = get_question(summarized_text,question_model,question_tokenizer)
      print (ques)
      #print (answer.capitalize())
      print ("\n")
```

The longest river the nile is 4,160 miles long-the world's longest river. Heavy rains in ethiopia caused the river to flood every summe
rich soil along its shores. This soil was fertile, which meant it was good for growing crops. Unlike the tigris and euphrates, the floc
the same time every year, so farmers could predict when to plant.

What is the world's longest?

What is the longest river in the world?

Who could predict when to plant crops on the nile?

Fig 16: Results of FAQ Generation

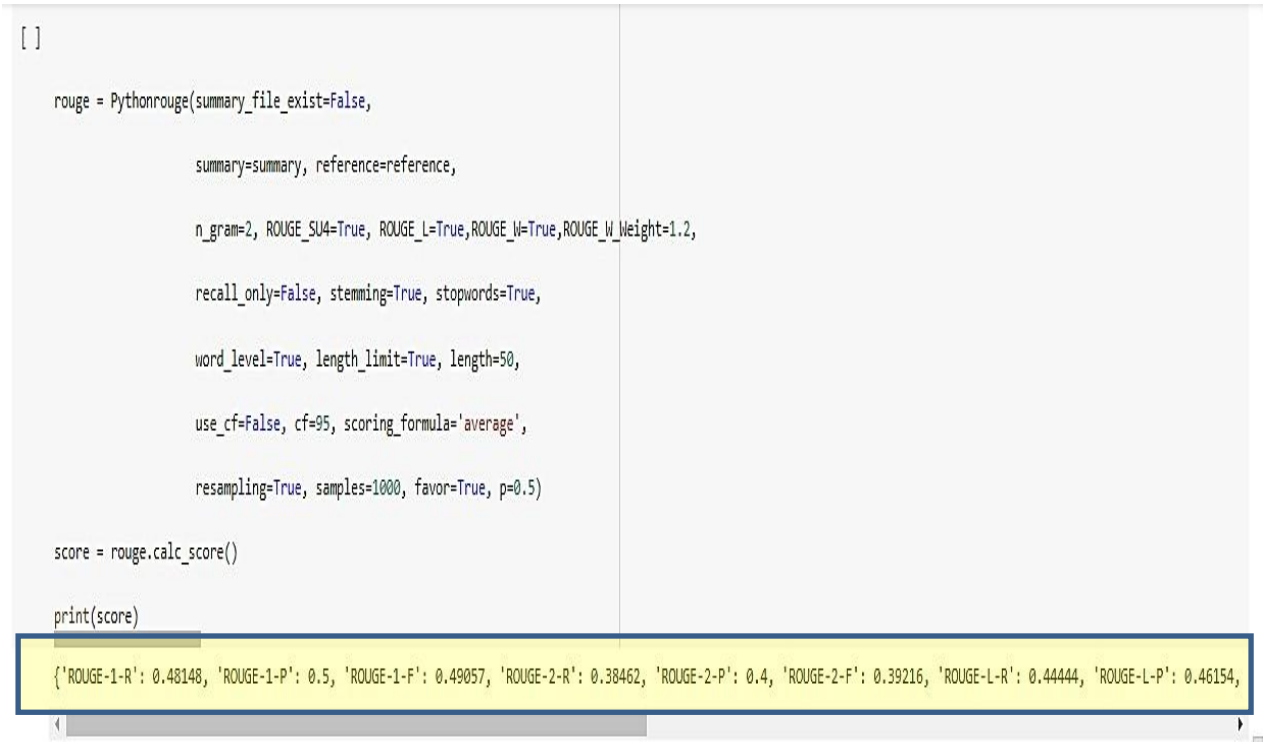

Fig 17: ROUGE Score of T5 (FAQ)

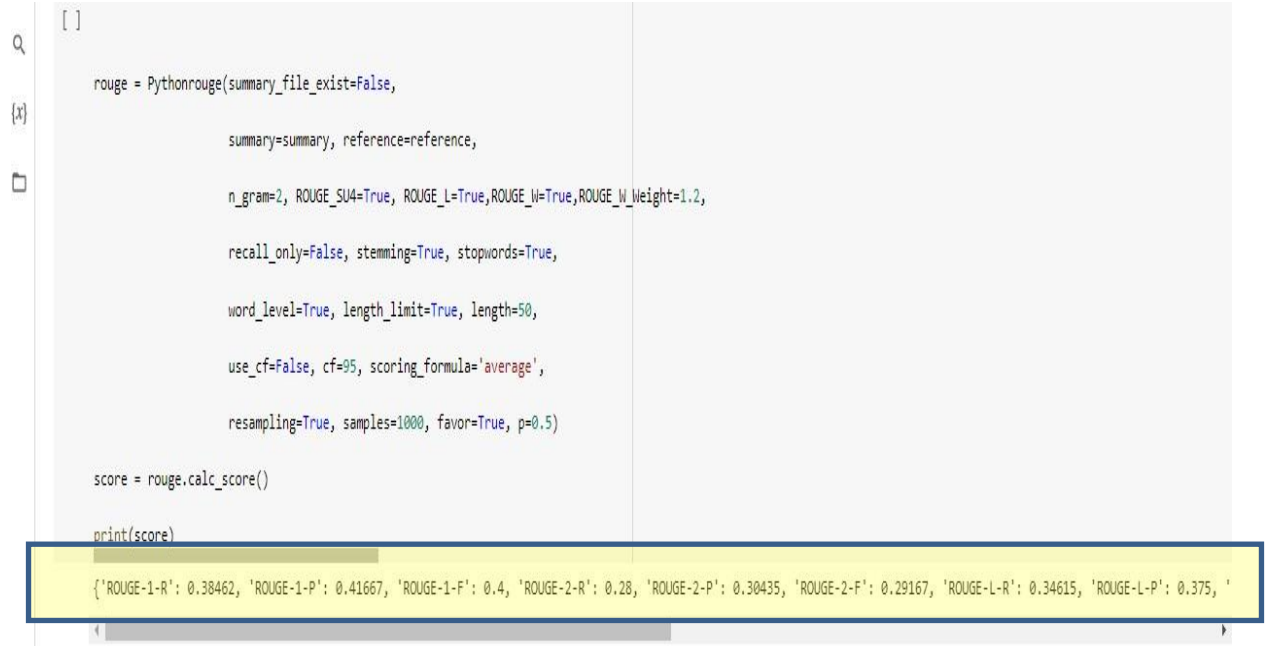

Fig 18: ROUGE score of BART (FAQ)