Proceedings of the International Conference on Sustainable Computing and Smart Systems (ICSCSS 2023)
DVD Part Number: CFP23DJ3-DVD; ISBN: 979-8-3503-3359-6

Automatic Question Answer Generation using T5 and NLP

Altaj Virani
Computer Engineering
Vidyavardhini's College of Engg and Technology, University of Mumbai, Vasai, India
altaj.s204513112@vcet.edu.in

Rakesh Yadav
Computer Engineering
Vidyavardhini's College of Engg and Technology, University of Mumbai, Vasai, India
rakesh.s204523101@vcet.edu.in

Prachi Sonawane
Computer Engineering
Vidyavardhini's College of Engg and Technology, University of Mumbai, Vasai, India
prachi.s204503202@vcet.edu.in

Smita Jawale
Associate Prof. Computer Engineering
Vidyavardhini's College of Engg and Technology, University of Mumbai, Vasai, India
Smita.jawale@vcet.edu.in

***Abstract*—Automatic Question Answer Generation (QAG) systems have received significant attention in recent years due to their potential to improve the efficiency of various natural language processing tasks. A QAG system is proposed that utilizes a state-of-the-art language model to generate high-quality question-answer pairs. The system supports various types of questions, including Wh-questions, fill-in-the-blank, full- sentence questions, multiple-choice questions, and true-false questions. System is developed and tested on a diverse range of text-based datasets, and the results show that it can generate accurate and relevant questions and answers for a given piece of text. The QAG system has significant potential for use in educational, industrial, and research settings where quick and efficient comprehension of text is crucial. The system's usability and effectiveness are demonstrated through experiments, and it is believed that it has the potential to be a valuable tool for a wide range of natural language processing tasks.**

Keywords— NLP (Natural Language Processing), QAG (mcq, fill ups, true or false, answer in one sentence) POS (part of speech), NER (Named Entity Recognizer)

I. INTRODUCTION

In today's world, there is an increasing demand for automatic question answer generation systems due to their potential to revolutionize the way information is learned and shared. With the exponential growth of digital content, such as textbooks, scientific papers, and educational videos, there is a need for efficient and effective ways to generate questions and answers from this vast amount of information. Additionally, the COVID-19 pandemic has accelerated the adoption of e-learning and remote education, further increasing the demand for automated question answer generation systems. The current trends in automatic question answer generation focus on leveraging the advancements in natural language processing (NLP) and machine learning (ML) techniques to develop more accurate and efficient systems. These systems utilize large datasets and pre-trained language models to generate high-quality questions and answers from various sources of information. Furthermore, the trend towards personalized education and adaptive learning has led to the development of systems that can generate customized questions based on the learner's knowledge level, learning style, and interests.

The present status of automatic question answer generation systems is promising. The recent advancements in NLP and ML techniques have led to the development of highly accurate and efficient systems that can generate questions andanswers from various sources of information. These systemshave the potential to reduce the workload of educators by automating the process of generating questions and answers for assessments and evaluations. Additionally, such systems can improve the learning outcomes of students by providing personalized and interactive learning experiences. The AQAG system is a sub-field of natural language processing (NLP) and information retrieval (IR). The system utilizes various techniques such as machine learning, statistical analysis, and knowledge representation to extract information and generate questions and answers. The system can be used to generate questions and answers for various domains, including education, customer service, and entertainment. The system is an advanced automatic question answer generation system that covers various types of questions suchas simple factoid, gap fill, MCQ based, true/false, and full sentence. It is designed to assist educators in creating assessments and evaluating student understanding of the material. The system uses natural language processing and machine learning algorithms to generate questions from various knowledge sources. One of the key features that set the system apart from other systems is its ability to generate questions of different types. Many existing systems are limited to generating only one or two types of questions, such as simple factoids or MCQs. The system, on the other hand, is versatile and can generate a wide range of questions, including fill-in-the-blank, true/false, and more complex sentence-based questions.

II. PROBLEM STATEMENT

The research study proposes an Automatic Question Answer Generation (QAG) system that aims to generate high-qualityquestions and answers from large digital content, catering to the growing demand for e-learning and remote education. Existing QAG systems are limited to generating

DVD Part Number: CFP23DJ3-DVD; ISBN: 979-8-3503-3359-6

specific types of questions, hence the need for a system that can generate wide range of question types accurately and efficiently. The proposed system utilizes state-of-the-art language models and machine learning techniques to generate accurate and relevant question-answer pairs for various types of questions. Its potential for use in educational, industrial, and research settings is demonstrated through experiments, and it has the potential to be a valuable tool for a wide range of natural language processing tasks.

III. RELATED WORK

A. REVIEW

Several approaches have been proposed to generate questions and answers automatically, depending on the type of question and the domain. However, most of these approaches have focused on a specific type of question, such as wh- questions, gap-fill questions, or multiple-choice questions (MCQs). For instance, some studies have proposed methods to generate wh-type questions, where the answer is a person, place, or thing. Other studies have focused on generating gap-fill questions, where students need to fill in the blank with the correct word or phrase. Still, others have proposed methods to generate MCQs, where students need to select the correct answer from the set of options.

However, in practice, teachers need to generate different types of questions to evaluate their students' knowledge, such as simple-factoid questions, true/false questions, and full-sentence questions. Thus, there is a need for an approach that can generate different types of questions and answers automatically. In this context, this study proposes a question answering generation system that can generate simple-factoid, gap-fill, true/false, full-sentence, and MCQ questions. The proposed system is based on a deep learning model that can generate questions and answers from textual input.

The reviews of recent studies published on Automatic question-answer generation have been put through in this section.

Min-Kyoung Kim et al. [1] introduced an Automated Question Generation System via Design of Question Answering. A rule-based system with Named Entity Recognition was devised to automatically generate questions and identify answers.

Sushmita Gangopadhyay et al. [2] presented a system for question and answer generation. Their approach included a focus generator module, utilizing the SQuAD dataset, and employing techniques such as the Neural Entity Selection Algorithm and Recurrent Neural Networks.

Mai Mokhtar [3] developed an Automatic Question Generation Model using a Deep Learning approach. The system utilized an encoder-decoder model, incorporating Named Entity Recognition and Seq-2-Seq techniques, trained on the SQuAD dataset.

Chidinma A. Nwafor et al. [4] proposed an Automated Multiple-Choice Question Generation system using Natural Language Processing Techniques. Their approach employed TF-IDF for generating MCQs.

Riken Shah et al. [5] developed an Automatic Question Generation system for Intelligent Tutoring Systems. Their system trained on a Wikipedia-based dataset and utilized Paradigmatic Relation discovery techniques for generating distractors. However, it has limitations in terms of dependence on a pre-existing knowledge base and a limited subject matter scope.

Pedro Álvarez et al. [6] aimed to develop Semantics and Service Technologies for Automatic Generation of Online MCQ tests. Their study focused on generating candidate distractors and establishing heuristics for their suitability using Semantic Trees.

Girish Kumar et al. [7] created an Automatic Fill-the-Blank Question Generator for Student Self-assessment. Semantic Similarity, Syntactic Similarity, and Context-fit techniques were employed with a high school biology textbook dataset, utilizing Word2Vec.

Chonlathorn Kwankajornkiet et al. [8] developed a method for automatically generating multiple-choice questions from Thai text. Their approach involved WordNet for retrieving potential distractors, translation using the Google Translate API, and ranking using linear regression models. A custom dictionary was used to improve accuracy.

Manish Agarwal and his team [9] created an Automatic Gap-fill Question Generation system from Key-list information in Biology textbooks. Their approach involved contextual and sentence similarity, utilizing Part of Speech Tagging and Term Frequency techniques.

Bidyut Das et al. [10] developed an E-Assessment tool for Automatic Generation of Fill-in-the-blank Questions with Corpus-based Distractors. Their approach used pattern search, a coarse-grained POS tag set, unigram and n-gram techniques for answer key identification, and multiword extraction.

Cheng Zhang et al. [11] proposed a method for Generating Adequate Distractors for Multiple-Choice Questions, tailored to different target types. Techniques used included POS tagging, NE tagging, semantic-role labeling, Fast-Text, and Word2Vec. Their dataset comprised US SAT Practice Reading Tests.

Akhil Killawala et al. [12] proposed a Computational Intelligence Framework for Automatic Quiz Question

DVD Part Number: CFP23DJ3-DVD; ISBN: 979-8-3503-3359-6

Generation. Their approach involved training the model on predefined questions, identifying potential gap candidates, and ranking based on semantic correctness. Techniques employed included Named-entity recognition, Super-sense tagging, and LSTM.

Bowei Zou et al. [13] developed Automatic True-False Question Generation for Educational Purposes. Their framework included an unsupervised domain-independent true/false question generation, template-based approach, and generative framework using a masking-and-infilling strategy. The dataset used was a passage about "Yellowstone National Park", and techniques employed included NLP and a TF-QG model.

Ruslan Mitkov et al. [14] proposed a method for generating multiple-choice test items from electronic documents. The method selects a clause, applies rules to generate a question with the subject as the answer. WordNet is used to retrieve distractors related to the answer through "hypernyms and coordinates" relation.

B. RESEARCH GAP

In the case of a scenario where an educator might need to generate questions for taking exams, similar to most human working processes, the system suffers due to bias. Due to having a preference for a particular topic, there could be some questions that might have been repeated in many question papers as the test developer has a personal inclination towards them. Hence, there is no guarantee that a paper could have purely randomly generated questions. The other challenges that the system may face are the unavailability of staff and resources, natural calamities, etc. Also, the security of the system could be compromised easily if leverage over the person who is responsible for generating question papers is obtained.

The limitations related to above research studies are:

- The need for users to provide keywords in order to generate a question: This could limit the usefulness of question generation tools, as users may not always know the best keywords to use.
- Limitations in the types of questions generated: If a tool can only generate simple factoid questions, it may not be useful for more complex or nuanced queries.
- Inability to generate questions of different categories: If a tool is limited to generating questions in a specific category, it may not be useful for users who need questions in other areas.
- Lack of answer generation: If a tool can only generate questions but not answers, users may still need to do additional research to find the information they need.
- Limitations of TFIDF: While TFIDF can be useful for determining the importance of keywords in a document, it does not take into account other factors such as position in text, semantics, and co-occurrences in different documents. This could limit the accuracy of question generation and answering tools that rely on TFID.
- Lack of question type variety: The system only allows for the creation of MCQ-based questions and does not provide options for other types of questions such as fill-in-the-blanks, true-false, and descriptive questions. This limitation could potentially impact the effectiveness and comprehensiveness of the assessments created using the system.
- Lengthy and difficult user training: users need to train the system, which is a lengthy and difficult task. This could potentially limit the usability and accessibility of the system for some users.
- Lack of algorithm details: only offers a high-level overview of the methods used, without delving into the specific algorithms employed. This could make it difficult for other researchers to replicate or build on the work done.
- Need for human judgments: The model requires human judgments through the Amazon Mechanical Turk platform to distinguish between good and bad gaps during the gap selection phase. This adds an additional layer of complexity to the system and could potentially impact its scalability and generalizability.
- No consideration of non-coinciding answer keys: It is unclear whether the approach takes into account situations where different versions of the same test may have different answer keys.

These gaps could potentially offer avenues for future research and development to improve the system and address its limitations.

IV.PROPOSED SYSTEM

The proposed Automatic Question Answer Generation (QAG) system utilizes various techniques to generate accurate and relevant question-answer pairs for various types of questions from large amounts of digital content. This section describes the working of the system. The QA generator employs the Hugging Face Transformers library to load a pre-trained T5 (Text-to-Text Transfer Transformer) model before generating question answers. The T5 model is a transformer-based model that has been pre-trained on a massive amount of text data, including diverse tasks such as language translation, summarization, and question answering. Once the T5 model is loaded, the QA generator receives the input text that needs to be processed. The input text can beany digital content, such as a textbook, scientific paper, or educational video. The QA generator then pre-processes the input text to prepare it for question and answer generation.

The pre-processing step includes several tasks such as tokenization, sentence splitting, and filtering out stop words.

1) *Tokenization*: Tokenization is the process of breaking the input text into individual words or tokens, which are then fed into the T5 model for processing.
2) *Sentence splitting*: Sentence splitting is defined as the

DVD Part Number: CFP23DJ3-DVD; ISBN: 979-8-3503-3359-6

process of breaking down the input text into individual sentences, which are then used as input for question generation.

3) *Stop word removal*: Stop words are commonly occurring words in the English language, such as "the," "is," and "and," which are filtered out because they do not add much meaning to the text and can lead to noisy question generation.

A. Factoid Wh-Type Question Answer Generation:

The server receives a request containing a passage of text and from the user. The passage of text can be a piece of a textbook, scientific paper, or educational video. The server then uses T5 model to generate the question and answer pairs.

T5 is a transformer-based language model that is trained on an enormous amount of text data for performing various natural language processing tasks, including text generation. The system uses a pre-trained T5 model called as *t5-base-question- generator* to generate questions and answers. The *t5-base- question-generator* model is a T5 model that has been fine- tuned on a dataset of question-answer pairs to improve its question generation abilities.

T5-Base-Question-Generator is a fine-tuned T5 model designed specifically for the task of generating questions from text. The model has been trained on a large dataset of text and question-answer pairs and uses a sequence-to-sequence architecture to generate questions. The model generates questions one token at a time using a decoder-only architecture and a sampling procedure from the model's probability distribution.

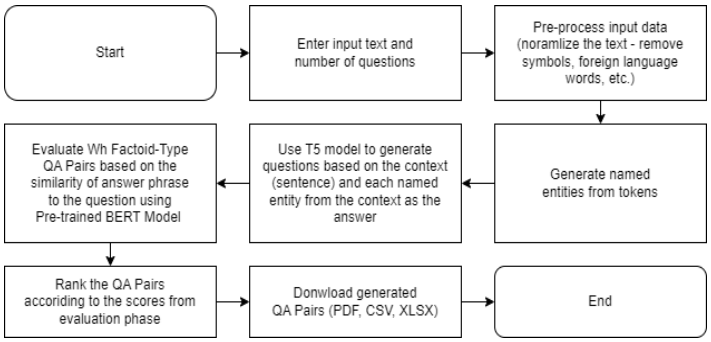

Figure 1: Wh-Type Question Answer Generation

When the server receives a request for a wh-question, it first pre-processes the passage of text by tokenizing it into a sequence of sub-words using the BERT tokenizer. Then, it formats the input for the T5 model by adding the special token "generate question:" at the beginning of the tokenized passage.

The wh-type question generation involves performing named entity recognition (NER) on the input sentence to identify the named entities. Then, the system iterates over the named entities in input sentence and converts them into the format required by the T5 model. The T5-based question generator model takes two inputs: answer and context. The answer is the specific information to generate a question about, while the context provides the necessary context for the answer. To create the formatted input, the answer is pre-appended with *"answer token*" and followed by the actual answer text, and then appended with *"context token"* and followed by the surrounding context text. For example, "The capital of France is Paris." would be formatted as *"answer token Paris context token the capital of France is Paris."*. This formatted input is fed into the T5 model to generate the desired output, which could be a question like "What is the capital of France?".

The input is then fed into the T5 model, which generates the question and answer pairs. The server post-processes the output by removing the special token and any trailing text, such as incomplete sentences or irrelevant information. Finally, it returns the generated wh-question and the corresponding answer to the user.

B. MCQ-Based Question Answer Generation:

In the MCQ-based Question Answer Generation module, the process is similar to that of generating a factoid wh-type question answer generation module. The difference lies in the distractor generation phase. After the answer is generated, it is sent to a separate function for generating distractors.

Distractor generation is an important part of creating multiple-choice questions (MCQs) as it helps to provide plausible but incorrect answer options that can challenge the knowledge of the test-taker. In the study you mentioned, the distractor generation process is performed using Named Entity Recognition (NER) from the same context, and when there are not enough words with the same NER, other techniques are used.

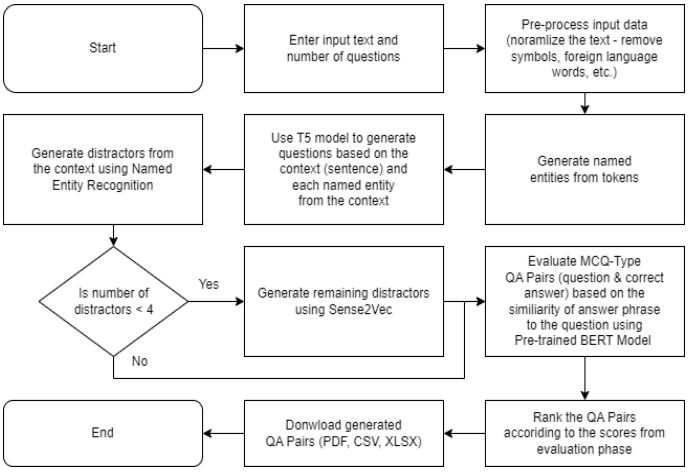

Figure 2: MCQ-Based Question Answer Generation

NER is a technique used in Natural Language Processing (NLP) to extract entities such as names, dates, and locations from unstructured text. In this case, the system uses NER to identify entities in the same context as the correct answer, and then uses those entities to generate distractors. For example, if the correct answer is "Canada", the NER tag for the answer would be “LOC”, the system might look for entities with the same NER tag “LOC”, i.e. names of other locations mentioned in the same text as potential distractors.

However, if there are not enough entities with the same NER in the context, the system also uses Sense2Vec to

DVD Part Number: CFP23DJ3-DVD; ISBN: 979-8-3503-3359-6

generate distractors. Sense2Vec is an extension of the popular Word2Vec algorithm that creates vector representations for not just words, but also their meanings or senses. In the context of distractor generation for MCQs, Sense2Vec can be used as a resource to generate plausible distractors when there are not enough similar Named Entity Recognition (NER) taggedwords available.

To use Sense2Vec for distractor generation, the system first identifies the correct answer and any related words or concepts in the context. If there are not enough similar NER tagged words available, the system uses Sense2Vec to identify related words or concepts that are similar to the correct answer.

To identify related words or concepts, the system calculates the cosine similarity between the vector representations of correct answer and other words or concepts in the Sense2Vec resource. The cosine similarity is used to measure the angle between two vectors and ranges from -1, which means completely dissimilar to 1, which means completely similar.

Once the related words or concepts have been identified, the system then filters out any candidates that are identical to the correct answer or that are not semantically relevant. This filtering is based on the appropriate similarity metric that can compare the similarity between the vector representations of the candidates and the correct answer.

After filtering, the system ranks the remaining candidates based on their similarity to the correct answer using the appropriate similarity metric. Finally, the system selects the three highest-ranked Sense2Vec distractors that are not identical to the correct answer to return as options for the MCQ question.

C. Full Sentence Question Answer Generation:

Full Sentence Question Answer Generation module is very similar compared to factoid wh-type module, the only difference being that full sentence question answer generation requires the answer token in the input data to be the entire context, which is the sentence instead of individual entities. Here, before the input data is given to the model, the data needs to be pre-processed and some checks need to be made, which are done in order to ensure that only a valid sequence is being fed to the model. These checks include making sure that a sequence has a maximum length of 490 tokens, and any special tokens in the sequence are removed. In the full sentence question answer generation phase of the Auto QAG repository, the T5 model is used to generate questions based on a given context. The context is provided as both the answer and the context, and is parsed into the expected format: answer token <answer text> context token <context text>. Similar to factoid, it returns the questions for each prepared input data instance. The T5 model uses a sequence-to-sequence architecture and generates questions by decoding the input sequence into a target sequence of text, which is the generated question. The decoding process is performed auto regressively, meaning thatthe model generates one token at a time and uses the previously generated tokens as context for the generation of the next token.

During training, the T5 model is fine-tuned on a large dataset of input-output pairs consisting of sentences and their corresponding questions, so that it learns to generate questions that are syntactically and semantically correct and that follow common question structures.

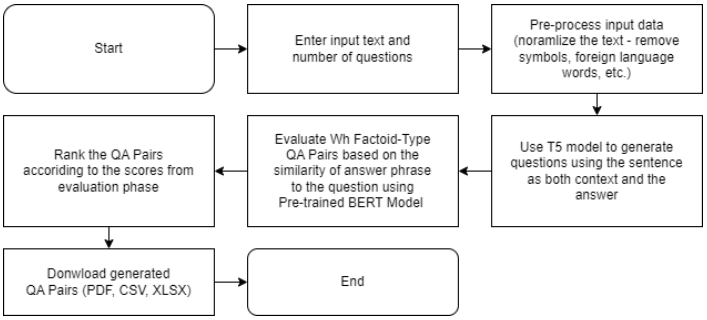

Figure 3: Full Sentence Question Answer Generation

D. Gap-fill Question Answer Generation:

The fill-in-the-blank (gap-fill) question answer generation module works by identifying a suitable sentence with a missing word that can be used to create a gap-fill question. The module first uses a part-of-speech (POS) tagger to identify the noun, adjective, or verb in the sentence that can be used to create a fill-in-the-blank question. The POS tagger identifies the word and its part of speech in the sentence.

Once the noun, adjective, or verb is identified, the module uses a list of rules to create the gap-fill question. The rules vary based on the part of speech of the identified word. For example:

If the identified word is a noun, the module creates a question using the following rules:

1) The module identifies the sentence with the missing word and replaces the missing word with a blank.
2) It then identifies the article or determiner before the missing word (if any) and uses it in the question. For example, if the original sentence is "The cat is sleeping on the mat" and the missing word is "mat", the module will use the article "The" in the question, such as "_cat is sleeping on the mat?"
3) The module then identifies the noun that comes after the missing word (if any) and uses it in the question. For example, if the original sentence is "The cat is sleeping on the mat" and the missing word is "mat", the module will use the noun "cat" in the question, such as "The_is sleeping on the mat?"

If the identified word is an adjective, the module uses a different set of rules to create the gap-fill question:

1) The module identifies the sentence with the missing word and replaces the missing word with a blank.
2) It then identifies the noun that comes before the missing word and uses it in the question. For example, if the original sentence is "The big cat is sleeping on the mat" and the missing word is "big", the module

DVD Part Number: CFP23DJ3-DVD; ISBN: 979-8-3503-3359-6

will use the noun "cat" in the question, such as "The ____ cat is sleeping on the mat?"

If the identified word is a verb, the module uses a different set of rules to create the gap-fill question:

1) The module identifies the sentence with the missing word and replaces the missing word with a blank.
2) It then identifies the subject of the sentence and uses it in the question. For example, if the original sentence is "The cat is sleeping on the mat" and the missing word is "sleeping", the module will use the subject "cat" in the question, such as "The cat is on the mat?"

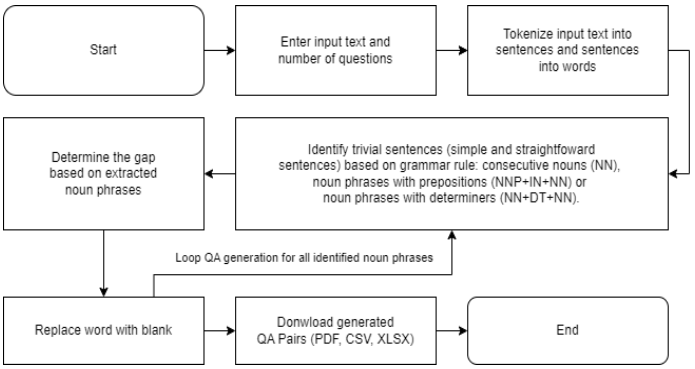

Figure 4: Gap-fill Question Answer Generation

E. Boolean Question Answer Generation:

The Boolean Question Answer Generation module uses the T5-Boolean-Questions model to generate a list of true/false questions with their corresponding yes or no answers. The model has been trained on the BoolQ dataset. The dataset contains a large number of questions and corresponding answers, with each answer being a simple "Yes" or "No". Due to its extensive training on this dataset, the T5-Boolean-Questions model is well-suited to a variety of natural language processing tasks. The Boolean Question Answer Generation module first tokenizes the input text using the T5 tokenizer and encodes it into a tensor of input IDs and attention masks. The input IDs are then fed into the T5-Boolean-Questions model, which generates a set of candidate true/false questions using beam search decoding. Beam search decoding is a method of generating multiple candidate outputs by maintaining a set of possible outputs, expanding them with each decoding step, and pruning them based on a scoring criterion.

Once the candidate questions are generated, each question is fed into the BoolQ model, which predicts a binary answer of "Yes" or "No". The BoolQ model is a binary question- answering model that is trained to predict the answer to a given question by classifying the input document as either supporting or refuting the answer. This is achieved by representing the input document and the question using a combination of attention and positional encodings, and then performing a binary classification task using a softmax layer.

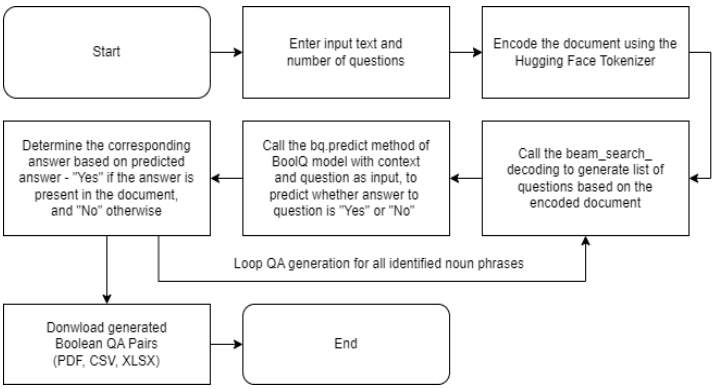

Figure 5: Boolean Question Answer Generation

The predicted answer is then added to a list of questions and their corresponding answers and returned to the user as the final output. This process enables the generation of a variety of true/false questions that can be used for assessing knowledge or testing comprehension, without the need for human intervention. The generated Boolean questions can be used in various applications such as quiz creation, exam preparation, and educational content creation.

V. RESULT

To assess the performance of the system, an evaluation was conducted by providing five different input texts, referred to as sentence1, sentence2, sentence3, sentence4, and sentence5. The user is required to input the text and select the question type and number of questions desired. The generated questions and corresponding answers are presented in a table format, with 'Q' representing the question and 'A' representing the answer. The accuracy of the generated questions and answers can be evaluated by comparing them with the original text and verifying the correctness of the answers manually. The system aims to provide a fast and efficient way of generating question-answer pairs, reducing the workload on human question setters, and ensuring fair and unbiased evaluations for students.

Table I: Result of Proposed System

Input Text	**Type of Question and Number of Questions**	**Generated Questions**	**Generated Answers**

DVD Part Number: CFP23DJ3-DVD; ISBN: 979-8-3503-3359-6

Sentence 1	MCQ Question, 2	1) How long did Indian Matchmaking last? 2) How many items of clothing does she have?	1) 1. 50s 2. nearly two weeks (correct) 3. 30s 4. years 2) 1. Hundreds 2. one 3. eight 4. dozens (correct)
Sentence 2	Gap-Filling Question, 2	1) _____ is an interpreted, high-level, general-purpose programming language. 2) _____ is often described as a “batteries included” language due to its comprehensive standard library.	1) Python 2) Python
Sentence 3	Boolean Question, 2	1) Python is created by Guido Van Rossum? 2) List is immutable in Python?	1) Yes 2) No
Sentence 4	Full Sentence Question, 2	1) What is the main reason why he has lectured businesses on the importance of mindfulness? 2) How does he rate the brain's concentration?	1) Mr. Campbell has lectured businesses on the significance of mindfulness and says the corporate world's view of mental health has changed a lot. 2) Dr. El-Imad thinks my neurons have behaved pretty well for a first outing, rating 30% concentration.
Sentence 5	Factoid Wh-Type Question, 2	1) Who is the mother of a child? 2) How much money is at stake in marriages in India?	1) Ms Taparia (correct) 2) millions of dollars (correct)

DVD Part Number: CFP23DJ3-DVD; ISBN: 979-8-3503-3359-6

VI. CONCLUSION

In this research study, the feasibility and effectiveness of using T5 model, Boolean questions, and NER and sense2vec techniques for automatic question generation is demonstrated. The approach has shown promising results in generating diverse and relevant questions for given texts, thereby assisting in various natural language processing tasks.

This approach is unique as a combination of different techniques has been utilized for question generation. The T5 model, which is a state-of-the-art language model, has been used as the base model for question generation. The Boolean question technique has been used to generate questions that require a yes/no answer, while NER and sense2vec techniques have been used to generate distractor options for multiple-choice questions. The approach can be applied to various domains, including education, research, and industry, to assist in automatic question generation. Future work could focus on improving the accuracy of the generated distractors and expanding the techniques used for question generation.

Despite the progress made in this field, there are still several challenges to overcome, such as the generation of more accurate and varied distractors, and the scalability of these systems to handle large datasets. Furthermore, there is a need for the development of more sophisticated evaluation metrics to assess the quality of generated questions.

VII. FUTURE WORK

The future work can focus on expanding the capabilities of the system to make it more versatile and useful for educational purposes.

1) ***Integration with test generating websites*:** One potential future work could be to integrate the system with some test generating website where teachers/professors can input the questions and answers generated by the system and use them to create tests for their students. This would make the system more useful and accessible for educational purposes.
2) ***Input modalities*:** Another potential future work could involve allowing the user to provide input in various modalities such as image, URL, etc. This would make the system more versatile and useful for generating questions from different types of inputs.
3) ***Bloom's Taxonomy Mapper*:** The development and integration of a Bloom's Taxonomy mapper module could be another potential future work. This module would map the generated questions according to appropriate Bloom's Taxonomy levels, such as knowledge, comprehension, application, analysis, synthesis, and evaluation. This would help in evaluating the complexity of the generated questions and would make the system more useful for educational purposes.

VIII. REFERENCES

[1] Min-Kyoung Kim, Han-Joon Kim, “Design of Question Answering System with Automated Question Generation”, 2008 IEEE 28th International Symposium on Computer-Based Medical Systems.

[2] Susmita Gangopadhyay, Ravikiran S.M, “Focused Questions and Answer Generation by Key Content Selection”, 2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM)

[3] Mai Mokhtar, “Automatic Question Generation Model Based On Deep Learning Approach”, International Journal of Intelligent Computing and Information Sciences, 2021

[4] Chidinma A.Nwafor, “An Automated Multiple- Choice Question Generation Using Natural Language Processing Techniques”, International Journal on Natural Language Computing (IJNLC), April 2021

[5] Riken Shah, Deesha Shah, Prof. Lakshmi Kurup, “Automatic Question Generation for Intelligent Tutoring Systems”, 2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA)

[6] Pedro Álvarez, Sandra Baldassarri, “Semantics and service technologies for the automatic generation of online MCQ tests”, 2018 IEEE Global Engineering Education Conference (EDUCON)

[7] Girish Kumar, Rafael E. Banchs, Luis Fernando D’Haro, “Automatic Fill-the-blank Question Generator for Student Self-assessment”, 2015 IEEE Frontiers in Education Conference (FIE)

[8] Chonlathorn Kwankajornkiet, Atiwong Suchato, Proadpran Punyabukkana, “Automatic Multiple- Choice Question Generation from Thai Text”, 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)

[9] Manish Agarwal, Prashanth Mannem, “Automatic Gap-fill Question Generation from Text Books”, IUNLPBEA '11: Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications

[10] Bidyut Das, Mukta Majumder, Santanu Phadikar, Sk. Arif Ahmed, “Automatic Generation of Fill-in- the- blank Question with Corpus-based Distractors for E- Assessment to Enhance Learning”

[11] Cheng Zhang, Yicheng Sun, Hejia Chen, Jie Wang, “Generating Adequate Distractors for Multiple- Choice Questions”, 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, 2020

[12] Akhil Killawala, Igor Khokhlov, Leon Reznik, “Computational Intelligence Framework for Automatic Quiz Question Generation”, 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)

DVD Part Number: CFP23DJ3-DVD; ISBN: 979-8-3503-3359-6

[13] Bowei Zou, Pengfei Li, Liangming Pan, Ai Ti Aw, “Automatic True/False Question Generation for Educational Purpose”, 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)

[14] Ruslan Mitkov, Le An Ha, Andrea Varga and Luz Rello, “Semantic similarity of distractors in multiple-choice tests: extrinsic evaluation”, EACL 2009 Workshop on GEMS: GEometical Models of Natural Language Semantics, Athens, Greece