MINISTRY OF EDUCATION
IMAM ABDULRAHMAN
BIN FAISAL UNIVERSITY
COLLEGE OF COMPUTER
SCIENCE & INFORMATION
TECHNOLOGY

وزارة التعليم
جامعة الإمام
عبدالرحمن بن فيصل
كلية علوم الحاسب
وتقنية المعلومات

جامعة الإمام عبدالرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY

QUESTION GENERATOR USING NATURAL LANGUAGE PROCESSING
(FINAL REPORT)

ARTI 501
ACADEMIC YEAR 2023-2024_ FIRSTSEMESTER

#	Name	Student ID
1	Reham Alzahrani	2200001931
2	Reema Albrahim	2200001337
3	Haya Aldossary	2190002968
4	Wadha Alotaibi	2190002150
5	Wahbia Saleh	2200006267

Project Team Number: Team 5
Section Number: AI Group 1
Instructor Name: Dr. Nawaf Alharbi
Date of Submission: December 9, 2023

Table of Acronyms

N	Acronyms	Definition
1	AI	Artificial Intelligence
2	NLP	Natural Language Processing
3	QA	Question Answering
4	MCQ	Multiple Choice Question
5	BERT	Bidirectional Encoder Representation from Transformer
6	T5	Text-to-Text Transfer Transformer
7	KNN	K-Nearest Neighbor
8	Sense2Vec	Sense to Vector
9	NER	Named Entity Recognition
10	TREC	Text REtrieval Conference
11	CLEF	Cross-Language Evaluation Forum
12	LSTM	Long Short-Term Memory
13	NLTK	Natural Language Toolkit
14	POS	Part of Speech
15	PDF	Portable Document Format
16	TP	True Positive
17	TN	True Negative
18	FP	False Positive
19	FN	False Negative
20	WSD	Word Sense Disambiguation
21	URL	Uniform Resource Locator

Table 1: Table of Acronyms

Abstract

Extracting useful information from the growing amount of unstructured data in the digital environment is a complex task. Unstructured data is defined as data that lacks a predefined model or framework, making it difficult to evaluate and draw insights from. However, it is now possible to evaluate the data and find patterns, and undetectable patterns within the unorganized content due to the development of NLP. By analyzing and producing human language, the discipline of NLP, a branch of AI offers a powerful solution by interpreting and generating human language. In the proposed project, information will be extracted from written sentences using NLP algorithms to create MCQs that have both correct and erroneous answers. In addition, transformer-based architecture models like KeyBERT and T5, which have been successful in developing QA systems, are investigated in this project. Moreover, the performance evaluation of the study involves measures such as recall, precision, and F1-score. The findings indicated that ROUGE-1 and ROUGE-L performed the best, with a 0.85 precision rate, 0.92 F1-score, and 1 recall rate. This automated question generation process has the potential to revolutionize how educators and students construct inquiries, leading to more tailored assessments and improved learning outcomes.

Table of Contents

Table of Tables

Table of Figures

Introduction

In recent years, the growth of digital data and the constantly changing digital environment have made it increasingly difficult to extract valuable information from vast volumes of unstructured data. The amount of data generated each second is so large that it is becoming more and more impossible for humans to handle and analyze it manually [1]. In order to meet this problem, NLP emerged as a key tool and foundational technology to address this issue by enabling a more natural and intuitive human-machine comprehension and interaction. In addition, it focuses on creating algorithms and models that can interpret human language [2]. Creating question generation systems is very useful to comprehend text that is readable by humans and provide appropriate questions and answers. By employing the question-generating system in several sectors such as customer service, education, and healthcare, these sectors will use the system to offer an intelligent QA capability, enhance learning opportunities, and enable effective information retrieval. NLP plays a crucial role in question answering and generation tasks, offering significant benefits and advancement. It enhances user experience by enabling natural and intuitive interactions with computers, allowing users to ask questions in their own words. By understanding the intent and context of queries, NLP techniques ensure accurate and relevant responses through the extraction of key information and generation of precise answers. Natural language generation capabilities enable the production of human-like and contextually relevant responses, particularly valuable in chatbots and virtual assistants. NLP facilitates multilingual support, breaking down language barriers and catering to diverse user populations. It ensures scalability and efficiency by effectively processing large volumes of text, identifying relevant information, and generating concise responses. The main purpose of this project is to generate MCQs from written sentences using NLP methods, that include both the correct and closely related incorrect answers. Moreover, automating assessment and evaluation procedures seek to make it simpler to determine comprehension and knowledge levels. The project explores transformer-based architectures such as BERT and T5, which have demonstrated high efficiency in various language processing tasks, including question generation. In addition, it aims to contribute to the development of reliable question generation systems that can extract information, generate questions, and provide precise answers. To transform knowledge evaluation, information processing, and human-computer interaction in a variety of fields, cutting-edge NLP techniques such as BERT model that integrates key phrases and contextual information into the question generation process. The T5 model also offers a flexible structure for producing and modifying textual data [3]. The results of the project showed that the ROUGE-1 and ROUGE-L achieved the best accuracy, reaching a 0.85 precision rate, 0.92 F1-score, and 1 recall rate.

Contribution

In our proposed project, the input file can be of any type if it is converted into machine-readable text, the model can generate an MCQ based on its contents. By implementing the question generation task based on the given input only, the questions are extracted based on the relevancy of the sentences from provided file by using the KeyBert model and the RAKE algorithm which extract noun keywords and relevant keywords respectively. Moreover, the code provides the ability to adapt the MCQ into an interface that the user can access and test themselves.

Problem Statement

In today's fast-paced educational environment, there is a growing need for innovative tools that can streamline the process of generating questions from text. This is where a system that generates questions using NLP is useful. This is crucial in educational environments because teachers and students must come up with questions that evaluate comprehension and foster critical thinking. However, manually generating questions can be laborious and challenging, particularly for lengthy texts, and it can take a lot of effort and may not be as accurate as doing it using NLP tools. Large quantities of text can be quickly analyzed by NLP algorithms to extract the necessary data and create queries. Therefore, by automating the process and making it more accessible, effective, and faster, developing an MCQ generation tool utilizing NLP can offer a solution to this issue.

Objectives

Implementing an MCQ generation using NLP techniques addresses multiple key objectives with a focus on the improvement of the user's experience, the efficiency of knowledge extraction, and continuous learning. Below are the primary objectives of building an MCQ generation system using NLP algorithms:

1. Enhanced User Experience:
   - Implement a system that generates accurate output.
   - Implement a friendly model that can ensure user satisfaction and accessibility.
2. Efficient Knowledge Extraction:
   - Extract useful information from large amounts of unstructured written data.
   - Enable the user to retrieve a specific query quickly.
3. Continuous Learning and Improvement:
   - Receive feedback from the users and implement further development of the system.
   - Create a personalized system that caters to each system.
4. Context Understanding:
   - Build an MCQ generation system that can retrieve useful information from unstructured text and generate a question for it.
   - Prevent ambiguity by specifying the system's limits.

Limitation

When developing an NLP-based system that generates questions, there are several limitations to consider. The system is restricted to a certain domain based on Wordnet-specific knowledge; therefore, NLP algorithms may not be able to create queries accurately for subjects or domains on which they have not been taught. The model also can only create MCQs from a provided text file which is not sufficient for evaluating learning outcomes as users may only have files in unsupported format. Language barriers are another constraint that may prevent NLP algorithms from being useful in multilingual situations. These algorithms may not be able to process text accurately in languages other than the ones they were trained on in the WordNet English database due to this limitation.

Review Of Related Literatures

In a study conducted by Hoshino and Nakagawa [2], the research discussed using machine learning algorithms to build a system that helps in language testing by generating questions related to English grammar and vocabulary. To train such a model, a collection of fill-in-the-blank questions was used from a TOEIC book, and then two classifiers, Naive Bayes and KNN, were used to evaluate the quality of the generated questions. Moreover, it was found that the KNN classifier was more accurate and robust but had a limitation in that it produced fewer blanks. Also, the limited feature set was mentioned as a limitation, where the research suggested that a large feature set might result in more efficient and relevant questions.

A study by Aldabe et al. [3] The study aimed to generate different types of questions, including MCQs, fill-in-the-blank, word formation, and error correction question types. Additionally, the dataset employed came from the ArikIturri data bank, which is made up of phrases that have been morphologically and syntactically evaluated and whose input is represented by XML markup language. Their method was based on Corpora and NLP techniques. The result of their approach was positively evaluated using an auto-metric generator. Their approach was limited by the number of distractors generated.

In a study by Ali et al. [4], the research considered a Sentence-To-Question generation task where the system gets fed sentences from the user as input and then generates questions. The Question Answering Track dataset was used to develop the system. Furthermore, NLP tasks require data cleaning and processing; for this project, tokenization from the Oak system was used. Moreover, complex sentences will still reside in the dataset. Therefore, an elementary sentence construction technique was used to achieve more accurate questions. Finally, results were obtained using recall and precision, with the recall evaluation tested with 70 topics and the system only being able to generate questions for 20 topics. As for the precision, the evaluation was done with 5 topics from the 20 successful topics in the recall. Findings lead to the conclusion that the system does not perform well with grammatically wrong sentences.

In a study by Athira et al. [5], the goal of the study was to produce a concise and detailed response to a query that is posed in natural language on a particular topic. by matching a query against a vast number of documents to retrieve accurate data. Moreover, the study's data were gathered using the ontology-based domain-specific natural language question-answering system, all from a repository of documents. The study included four stages that enhance QA abilities for difficult queries. Query analysis and classification were done in the first stage, while document retrieval was done in the second. The third stage analyzed these papers, whereas the final stage used natural language processing to extract information and provide replies. Finally, the study achieved an accuracy rate of 94%.

A study by Chan and Fan [6] aimed to discover the usage of the BERT model for question-answer tasks with two sequential question generation models. The model is trained and evaluated on the SQuAD dataset, which contains 536 Wikipedia articles and 100,000 reading comprehension questions. Moreover, the model was trained using the PyTorch version of the BERT model. The data was split into a training set of 80%, a development set, and a test set of 10% each. The results were described as state-of-the-art results achieving a high matching rate, with the highest F1 score reaching 86.82 with the question average tokens of approximately 12

.

A study by Gumaste et al. [7], The purpose of the research is to use the semantic functions of words to produce more detailed questions. The model suggested a rule-based approach to question generation from sentences. The question-answering system used a wide range of text data. Syntactic and semantic analysis are included in pre-processing. Semantic analysis is performed with NER. The generated questions were compared with the inaccurate ones, considering questions created by English-speaking humans. The accuracy, precision, and recall were calculated using the confusion matrix's properties, which were all calculated in this manner and produced good results. The limitation of the research is that more accurate questions must be generated and reused from databases.

In a study by Nwafor and Onyenwe [8], the authors proposed a method to automate MCQ generation using NLP techniques. The dataset was collected and extracted from five different lecture materials: computer science, geology, philosophy, religion, and history, with varying sizes of sentences. Their approach was to convert the document into a text file, split the text into sentences, and then tokenize the sentences from which the corpus is built in TF-IDF and N-gram mode. The result was evaluated using precision, recall, and F-measure for each subject. The highest precision was 0.44, the highest F-measure was 0.58 for computer science, and the highest recall was 0.98 for philosophy.

In a study by Patil et al. [9], the system offered a platform for interactive reading where users can submit an e-book and receive a textual summary as well as questions like MCQs, fill-in-the blanks, and one-word puzzles. The dataset was taken from IEEEXPlore and Google Scholar. The main task was accomplished using the two methodologies, namely the pre-trained T5 model and the pre-trained DistilBART model. The two were judged on their performance using the evaluation benchmarks CNN/DM-ROUGE-1, CNN/DM-ROUGE-2, and CNN/DM-ROUGE-L. The accuracy for relevant question generation was 73.873%. In addition, the system's limitation was that it does not concentrate on developing difficult questions for a better learning process.

In a study conducted by Rathod et al. [10], the research aimed to use NLP techniques to generate educational questions from a given text. The training data is a collection of fill-in-the-blank questions from a TOEIC preparation book. The research started with question paraphrasing and found that it was not ideal. Therefore, a fine-tuned ProphetNet was used to generate two questions for a given context. Additionally, to be able to generate multiple questions, they used sampling repeatedly and considered sets of 2, 4, 6, and 8 questions. Moreover, ProphetNet's single-question output was paraphrased and used in the second question. By fine-tuning the output of two questions sequentially and sampling multiple times, a combination of PINC, a QA model, and SBERT were used to evaluate the generated questions. As a result, the highest outcome yielded for each approach was 98%.

A research paper by Thotad et al. [11] proposed a technique to help students and teachers prepare for competitive tests. This study set questions for the computer-based test. The dataset used in the study was from natural human language, which can be written with NLTK. Moreover, key word extraction used the TF-IDF technique, and a wiki was implemented to verify the validity of phrases. Additionally, the WordNet program was utilized to create triplets for the purpose of creating exam questions and to run input clarity checks. To create vector space representations of words, the Sense2vec neural network model makes use of enormous corpora of words. As a result, the program produced good performance and can be applied to the subject of education. However, the research was limited to generating only questions without paraphrasing them.

In a paper written by Jin et al. [12], it presented a QA system using the CQA dataset, which is a large-scale English database. The dataset used in the study was collected from Stack Exchange. Furthermore, the system consisted of three parts: first, a classifier using a BERT-based classification model. Secondly, an answer obtainer that uses a BERT-based semantic similarity algorithm to achieve the best match. Lastly, a summarizer that integrates a T5-based model for substance extraction while rectifying grammar mistakes. As for the result, the model was evaluated using CLEF and TREC, obtaining good findings.

In a study conducted by Chomphooyod et al. [13], it discussed the use of deep learning algorithms to automatically generate MCQs on English grammar. In the training process, a collection of grammatically correct sentences was used as a dataset, with a model trained using T5. The acceptance rate of the generated questions is 86%. However, the study contains a limitation related to the grammar topics, where it must depend on the POS token.

Summary Table

N	Author/s	Year	Title	Dataset	Technique/s	Results
[2]	A. Hoshino, H. Nakagawa.	2005	A real-time multiple-choice question generation for language testing	Fill-in-blank questions from TOEIC book	- Naive Bayes - KNN	KNN produced better results
[3]	Itziar Aldabe Maddalen Lopez de Lacalle Montse Maritxalar Edurne Martinez Larraitz Uria	2006	An Automatic Question Generator Based on Corpora and NLP TechniquesAnswering System	ArikIturri data bank	- Corpora - NLP	Positive result
[4]	Hussam Ali, Yllias Chali, Sadid A. Hasan	2010	Automatic question generation from sentences	Question Answering Track	Tokenization	5 out of 20 topics.
[5]	Athira P. M. Sreeja M. P. C. Reghuraj.	2013	Architecture of an Ontology-Based Domain-Specific Natural Language Question Answering System	Repository of Documents	Four QA stages queries. **Stage 1** - Query analysis - Query classification **Stage 2** - Document retrieval **Stage 3** - Paper analyzation **Stage 4** - Information extraction using NLP - Generating replies using NLP	Accuracy=0.94

Table 2 : Summary Table from paper

N	Author/s	Year	Title	Dataset	Technique/s	Results
[6]	Ying-Hong Chan, Yao-Chung Fan.	2019	A Recurrent BERT-based Model for Question Generation	SQuAD	BERT	F1 Score: 86.82%
[7]	Priti Gumaste, Shreya Joshi, Srushtee Khadpekar, Shubhangi Mali.	2019	Automated Question Generator system using NLP Libraries.	Wide range of text data.	- Semantic Analysis is / Named Entity Recognition (NER).	- Accuracy 80.00% - Precision 0.84 - Recall 0.87
[8]	Chidinma A. Nwafor Ikechukwu E. Onyenwe	2021	AN AUTOMATED MULTIPLE-CHOICE QUESTION GENERATION USING NATURAL LANGUAGE PROCESSING TECHNIQUES	Five different lecture materials: - Computer. - Science. - Geology. - Philosophy. - Religion. - History.	- TF-IDF - N-gram mode	**computer science** precision =0.44 F-measure= 0.58 **philosophy** recall=0.98
[9]	Spandan Patil, Lokshana Chavan, Janhvi Mukane, Dr. Deepali Vora, Prof. Vidya Chitre.	2022	State-of-the-Art Approach to e-Learning with Cutting Edge NLP Transformers: Implementing Text Summarization, Question and Distractor Generation, Question Answering	- IEEEXPlore - Google Scholar. - WordNet	- LSTMs - T5 - BERT - Sense2Vec	- Accuracy 73.873%
[10]	Manav Rathod, Tony Tu, and Katherine Stasaski	2022	Educational Multi-Question Generation for Reading Comprehension	- Quora Question (Chen et al., 2018) pairs dataset	- ProphetNet	- The highest outcome was 98%
[11]	Puneeth Thotad, Shanta Kallur, Sukanya Amminabhavi	2023	Automatic Question Generator Using Natural Language Processing	Natural human language dataset	- TF-IDF - Sense2vec	- Positive result

N	Author/s	Year	Title	Dataset	Technique/s	Results
[12]	Sol Jin, Xu Lian, Hanearl Jung, Jinsoo Park, Jihae Suh.	2023	Building a deep learning-based QA system from a CQA dataset	CQA	- BERT - T5	TREC - CLEF
[13]	P. Chomphooyod, A. Suchato, N. Tuaycharoen, P. Punyabukkana	2023	English grammar multiple-choice question generation using Text-to-Text Transfer Transformer	Collection of grammatically correct sentences	- T5 - POS tagging	Acceptance rate of 86%

Literature Survey Outcome

Based on a comprehensive reading of previous studies related to the use of NLP and other artificial intelligence techniques that are concerned with building MCQ-generating models, it can be observed that some algorithms appear to be used more frequently and result in a better outcome. The datasets used to train MCQ-generating models differ; some studies used WordNet, which is a large database of English synonym sets (synsets) that express a distinct concept, while other studies used English articles and books such as IEEEXPlore, Google Scholar, and the TOEIC preparation book to train the model. Moreover, techniques from machine learning and deep learning were used, such as in paper [2], where Naïve Bayes and KNN were used, TF-IDF in paper [8] [11], Sense2vec [11], and the BERT-based classification model that was mentioned in paper [6] [10] [12] as a pre-trained model. Also, the models implemented in the paper [9], [12], and [13] use T5-trained models, which resulted in an acceptance rate of 86%.

Description of the Proposed Techniques

KeyBERT (Key Bidirectional Encoder Representations from Transformers) model:

One of the most popular NLP models is BERT, a language representation technique developed by Google [16]. It utilizes two procedures, pre-training and fine-tuning, to create advanced models for various tasks, such as question answering and text generation [16]. An extended version of BERT is KeyBERT, implemented as a Python module that simplifies and enhances keyword extraction and embedding which can be done using "distilbert-base-nli-mean-tokens" [17]. This tool leverages BERT's capabilities to provide keyword representations and support a wide range of NLP applications. By considering the contextual data provided by BERT, KeyBERT can generate a list of the most relevant keywords based on their match with the input text, while also providing a list of the highest-ranked keywords.

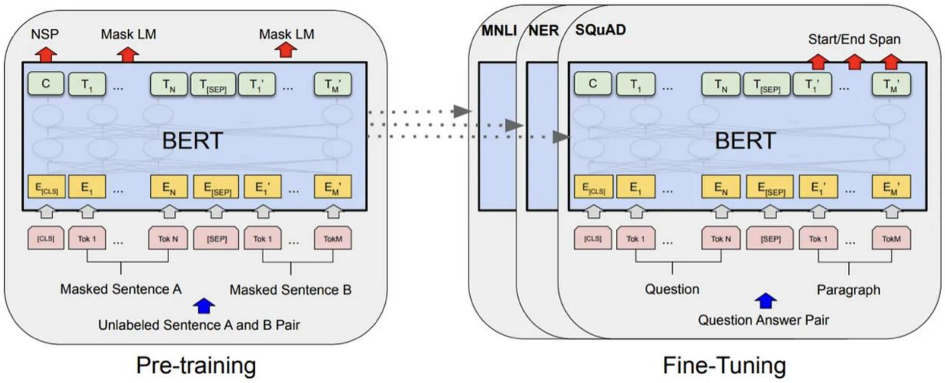


Figure 1:BERT model adapted from[18]

T5 (Text-to-Text Transfer Transformer) model:

Numerous NLP applications, such as text categorization, summarization, and question answering, heavily rely on the transformer-based T5 model. One of the model’s key advantages is its ability to generate and modify textual data by converting one text sequence into another. This makes it particularly suitable for question generation systems, as it can generate questions based on the provided text inputs. T5 is trained to predict masked tokens within a sentence during pretraining using the masked language modeling approach. During the fine-tuning process, T5 is trained on task-specific datasets and instructions to answer questions in two different ways. The project focuses on open-book question answering, where the model extracts the answer from the given context [19].

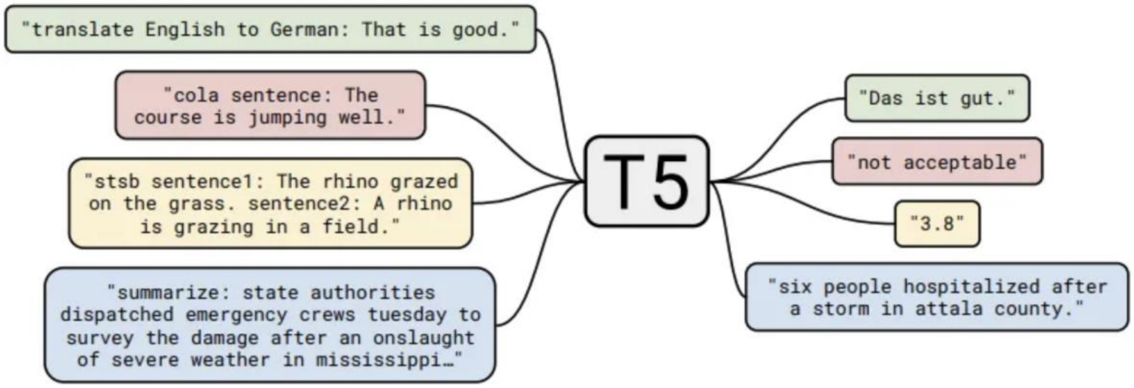

"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."

Figure 2:T5 framework adapted from[20]

Criteria for Performance Evaluation

To measure the performance of our model, the project used ROUGE score as an evaluation metric. It calculates the amount of overlap between the reference text and the text produced by the system. The higher the ROUGE score, the better the system's performance. The ROUGE score computes three performance measures: recall, precision, and F1-score. Precision, on the other hand, calculates the total number of true positive predictions in the positive class, whereas recall determines the total number of true positive predictions in all positive examples. Based on that, we can conclude that the F1 score is the average of precision and recall. The performance measures will be evaluated based on identifying TP, FP, TN, and FN [14].

$$\textbf{Precision} = \frac{TP}{TP+FP} \qquad \textbf{F1-Score} = \frac{2*(Recall*Precision)}{Recall+Precision} \qquad \textbf{Recall} = \frac{TP}{TP+FN}$$

Optimization Strategy

Several optimization techniques in NLP help to maximize the program's overall performance, processing speed, and memory use. One common approach is to preprocess the text data by tokenizing, converting it to lowercase, removing stopwords, and underscores. Another strategy is to use keyword extraction algorithms such as KeyBERT or RAKE, both can be used to extract keywords from text, which is helpful for a variety of NLP tasks, including information retrieval.

Additionally, techniques such as POS tagging, TextBlob, and using the T5 question answering pipeline can help to optimize the computational performance of NLP applications.

The program's optimization techniques:

- **Preprocess the text data:** Eliminating all unnecessary information from the data to prepare it for additional processing and increase the accuracy of the outcomes [15].

- **Using KeyBERT or RAKE:** enhances the precision of the outcomes derived from the data by focusing on the most crucial keywords and key phrases [16].

- **POS tagging and TextBlob:** By determining the most crucial grammatical components of a sentence, it can also aid in lowering the dimensionality of the data [17].

Data

Dataset Description

The dataset utilized in this project is a corpus titled "Importance of Software Engineering Principles in Developing Efficient and Reliable Software Products." It is an open-source dataset obtained from Kaggle and sourced from the extensive English WordNet database. The NLTK module was combined with the WordNet database to identify word definitions, synonyms, antonyms, and other relevant information. The WordNet database organizes synonym sets into hierarchies, making it easy to access and recognize hypernyms and hyponyms.

Note that this program includes a feature that allows users to upload their own data, which can be a text file, pdf, or URL.

Statical Analysis of The Dataset

The statistical analysis of the dataset captures the patterns within the data and provides a clear understanding of the characteristics such as quantity and variability in order to preprocess data and build accurate models. In this project, word frequency was calculated using a counter that calculated the total words of the dataset, and the number of unique words which are 523 and 254 respectively. In addition, the 10 most common words in the dataset, along with their frequencies were calculated and plotted as shown in the figure below.

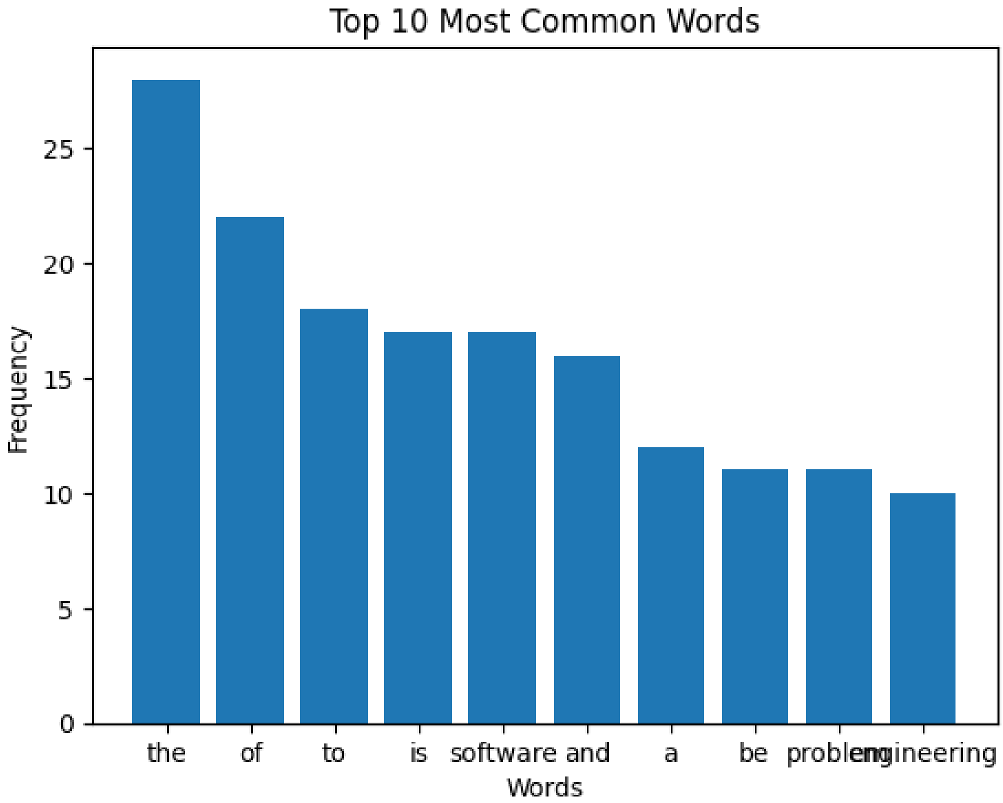


Figure 3 The top 10 most common words

Experimental Setup

The goal of the project is to conduct an MCQ generation task by using a variety of file sources, including text files, csv files, pdf files, and URL links. The task was conducted using the Python programming language, using the necessary libraries. Firstly, for a pdf file, text extraction is required to convert it into machine-readable format. Next, URL links are retrieved and stored locally for analysis. Afterwards, the data is preprocessed using several NLP techniques, such as tokenization, stopwords, and unnecessary character deletion. Afterwards, noun keywords are extracted using the KeyBert model, which can understand context and generate precise questions [16] based on the input file. As for the question-answering pipeline, a T5 model was used; this pipeline can find the correct choice for the MCQ generated. To find relevant keywords, a RAKE technique was used for such extraction. After the questions are generated, they are saved into a text file, which can be saved by the user. Moreover, the ability to take the MCQ test is embedded into the program, and an accuracy score based on the user's answers is displayed after the test is completed. The following figure shows the process of the experimental setup.

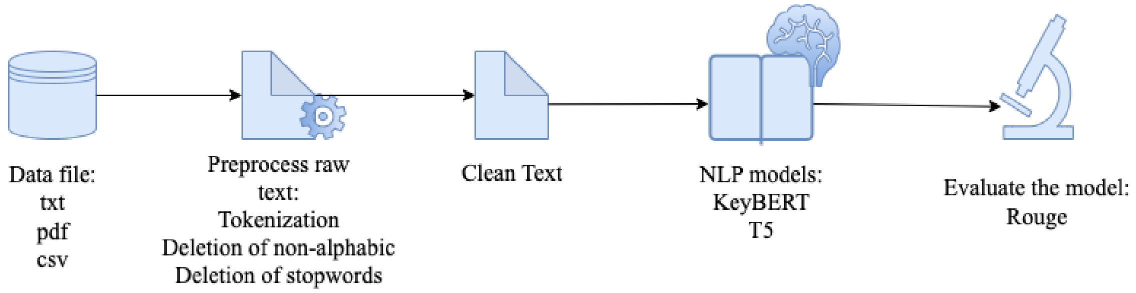


Figure 4: Experimental Setup Process

Tools

Our proposed model will be implemented using pre-trained T5 and KeyBERT models, along with a machine-learning algorithm and an advanced NLP algorithm for evaluation. The Python programming language will be utilized on the Collab platform to extract the dataset from Google Drive and facilitate collaboration. To ensure smooth implementation and testing of our development procedures, we will be using five laptops: Surface Pro 7 Core i7, Microsoft Surface Pro 7 Core i5, MacBook Air Core i5, MacBook Air Core M1, and MSI Core i7.

Result and Discussion

The model performance is analyzed using ROUGE scores. The rouge1 represents unigrams with a precision of 0.85, recall 1, and f-measure of 0.92. While the rouge2 evaluates based on bigram and it has a precision of 0.80, recall 0.94, and f-measure 0.87. Moreover, the last metric is rougeL that evaluates based on the longest common subsequence and it has precision, recall, and f-measure of 0.85, 1, and 0.92 respectively. The findings of the metrics indicate a good performance of the model, and it generates questions that capture most of the important information presented. The score for each metric is summarized in the table below.

Metrics	Precision	Recall	F-measure
rouge1	0.85	1	0.92
rouge2	0.80	0.94	0.87
rougeL	0.85	1	0.92

Table 3 ROUGE scores

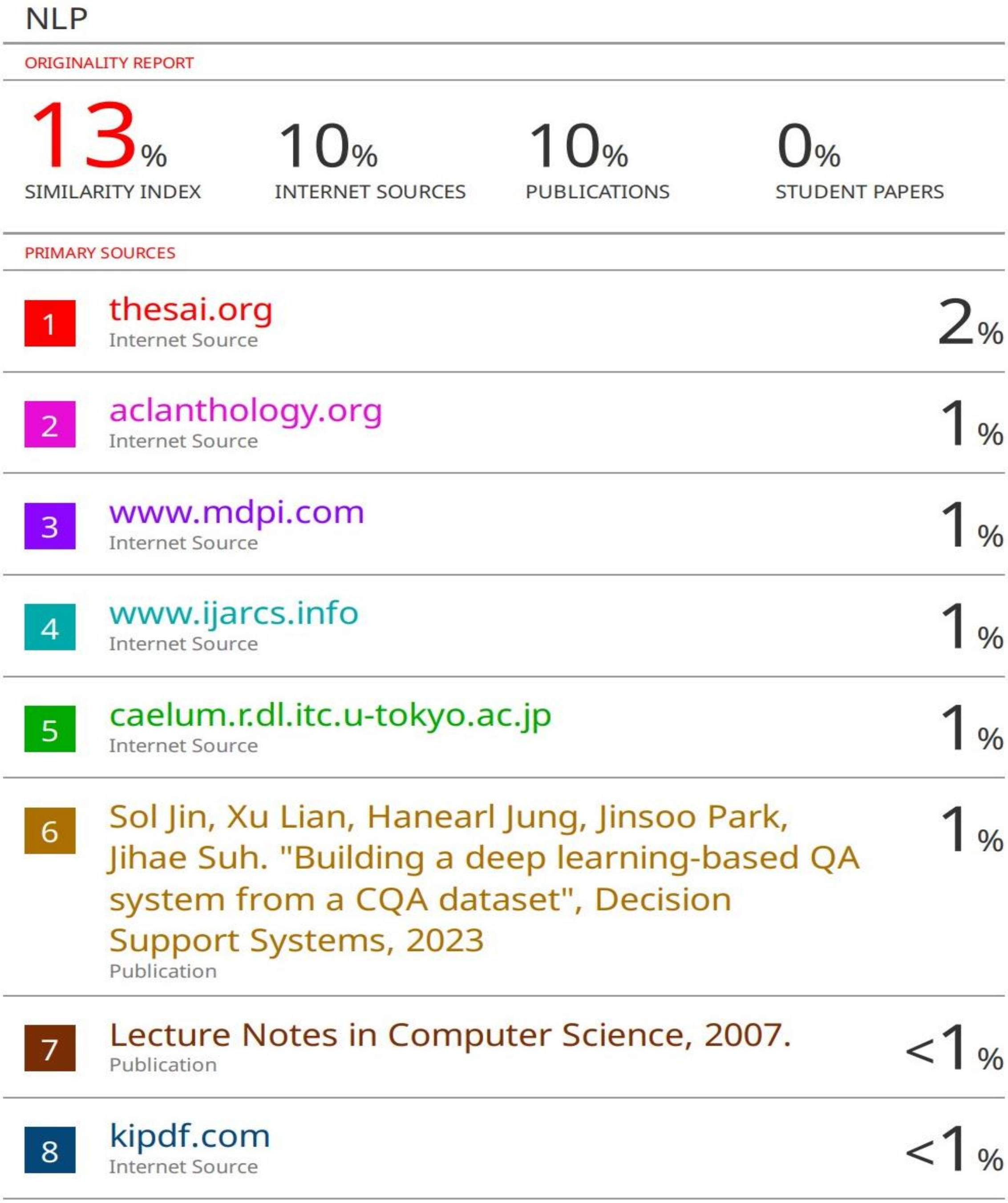
Conclusion and Recommendation

In this project, we investigated the use of two different models for creating MCQs from a variety of document sources, including text files, PDF files, CSV files, and URL links. This versatility makes the system more applicable in real-world applications. To improve the quality of the input data, the dataset was additionally preprocessed using tokenization and the removal of superfluous characters. This step refined the models' ability to comprehend and generate relevant MCQs. Furthermore, KeyBERT proved its ability to recognize crucial sentences and MCQs based on the primary content of the papers by utilizing keyword extraction. The T5 model demonstrated competence in comprehending context and generating questions with nuanced structure. By detecting important phrases, the unsupervised keyword extraction algorithm RAKE showcased its ease of use and effectiveness in producing MCQs. The Rouge score was used to evaluate the generated MCQs, offering a reliable way to measure the questions' quality and yielding a precision of 0.85, recall of 1, and an f-measure of 0.92. Moreover, insights into the usefulness and user satisfaction of the generated MCQs may be obtained by integrating user feedback. To ensure convenience, it would be recommended to include the feature that allows users to enter text directly into the software as a string.

Source Code

https://colab.research.google.com/drive/10XuCV8ItR5igOLnve2Pm41GjtHw-quGb?usp=sharin

Plagiarism Percentage



NLP

ORIGINALITY REPORT

13%	10%	10%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	thesai.org Internet Source	2%
2	aclanthology.org Internet Source	1%
3	www.mdpi.com Internet Source	1%
4	www.ijarcs.info Internet Source	1%
5	caelum.r.dl.itc.u-tokyo.ac.jp Internet Source	1%
6	Sol Jin, Xu Lian, Hanearl Jung, Jinsoo Park, Jihae Suh. "Building a deep learning-based QA system from a CQA dataset", Decision Support Systems, 2023 Publication	1%
7	Lecture Notes in Computer Science, 2007. Publication	<1%
8	kipdf.com Internet Source	<1%

As shown in the following picture, this is the plagiarism percentage using the Turnitin website for this report.

References

[1] M. Matson, “Unstructured Data Processing: Tech terms explained,” Blog, https://www.playerzero.ai/advanced/tech-terms-explained/unstructured-data-processing-tech-terms-explained (accessed Oct. 5, 2023).

[2] Devlin, J. et al. (2019) Bert: Pre-training of deep bidirectional Transformers for language understanding, ACL Anthology. Available at: https://aclanthology.org/N19-1423/

[3] Raffel, C. et al. (2023) Exploring the limits of transfer learning with a unified text-to-text transformer, arXiv.org. Available at: https://arxiv.org/abs/1910.10683

[2] A. Hoshino and H. Nakagawa, “A real-time multiple-choice question generation for language testing,” *Proceedings of the second workshop on Building Educational Applications Using NLP - EdAppsNLP 05*, 2005. doi:10.3115/1609829.1609832

[3] Aldabe, I., Lopez De Lacalle, M., Maritxalar, M., Martinez, E., & Uria, L. (2006). ArikIturri: an Automatic Question Generator Based on Corpora and NLP Techniques

[4] Ali, H., Chali, Y. and Hasan, S.A. (2010) Automatic question generation from sentences, ACL Anthology. Available at: https://aclanthology.org/2010.jeptalnrecital-court.36/(Accessed: 09 September 2023).

[5] P.M, A., M, S. and P.C, R. (2013) ‘Architecture of an ontology-based domain-specific natural language question answering system’, *International journal of Web &amp; Semantic Technology*, 4(4), pp. 31–39. doi:10.5121/ijwest.2013.4403.

[6] Chan, Y.-H., & Fan, Y.-C. (2019). *A Recurrent BERT-based Model for Question Generation*

[7] Gumaste, P. S., Joshi, S. S., Khadpekar, S. A., & Mali, S. R. (2019, December 31). AUTOMATED QUESTION GENERATOR SYSTEM: A REVIEW. *International Journal of Engineering Applied Sciences and Technology*, *04*(08), 171–176. https://doi.org/10.33564/ijeast.2019.v04i08.027

[8] A. Nwafor, C. and E. Onyenwe, I. (2021) ‘An automated multiple-choice question generation using natural language processing techniques’, International Journal on Natural Language Computing, 10(02), pp. 1–10. doi:10.5121/ijnlc.2021.10201.
Available at: https://aclanthology.org/2010.jeptalnrecital-court.36 /(Accessed: 09 September 2023)

[9] Patil, S., Chavan, L., Mukane, J., Vora, D., & Chitre, V. (2022). State-of-the-Art Approach to e-Learning with Cutting Edge NLP Transformers: Implementing Text Summarization, Question and Distractor Generation, Question Answering. *International Journal of Advanced Computer Science and Applications*, *13*(1). https://doi.org/10.14569/ijacsa.2022.0130155

[10] M. Rathod, T. Tu, and K. Stasaski, "Educational multi-question generation for reading comprehension," ACL Anthology, https://aclanthology.org/2022.bea-1.26/ (accessed Sep. 8, 2023)
[11] Thotad, Puneeth & Kallur, Shanta & Amminabhavi, Sukanya. (2023). Automatic Question Generator Using Natural Language Processing. Journal of Pharmaceutical Negative Results. 13. 2759-2764. 10.47750/pnr.2022.13. S10.330. https://www.researchgate.net/publication/369228453_Automatic_Question_Generator_Using_Natural_Language_Processin

[12] Jin, S., Lian, X., Jung, H., Park, J., & Suh, J. (2023). Building a deep learning-based QA system from a CQA dataset. *Decision Support Systems*, 114038. https://doi.org/10.1016/J.DSS.2023.114038

[13] P. Chomphooyod, A. Suchato, N. Tuaycharoen, and P. Punyabukkana, "English grammar multiple-choice question generation using text-to-text transfer transformer," *Computers and Education: Artificial Intelligence*, vol. 5, p. 100158, 2023. doi:10.1016/j.caeai.2023.100158

[14] Nighania, K. (2019) *Various ways to evaluate a machine learning model's performance*, *Medium*. Available at: https://towardsdatascience.com/various-ways-to-evaluate-a-machine-learning-models-performance-230449055f15 (Accessed: 02 October 2023).

[15] Agrawal, R. (2022, August 5). *Must Known Techniques for text preprocessing in NLP*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/06/must-known-techniques-for-text-preprocessing-in-nlp/

[16] Mansour, A. (2022, January 5). *Four of the easiest and most effective methods to Extract Keywords from a Single Text using Python*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2022/01/four-of-the-easiest-and-most-effective-methods-of-keyword-extraction-from-a-single-text-using-python/

[17] *TextBlob: Simplified Text Processing — TextBlob 0.16.0 documentation*. (n.d.). https://textblob.readthedocs.io/en/dev/