Multiple Choice Questions(MCQs) Generation

Project-II (CS47006) report submitted to

Indian Institute of Technology Kharagpur

in partial fulfilment for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Gangaram Sudewad

(20CS30017)

Under the supervision of

Professor Sudeshna Sarkar

Department of Computer Science and Engineering

Indian Institute of Technology Kharagpur

Spring Semester, 2023-24

April 2, 2024

DECLARATION

I certify that

(a) The work contained in this report has been done by me under the guidance of my supervisor.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(d) Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Date: April 2, 2024 (Gangaram Sudewad)
Place: Kharagpur (20CS30017)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

KHARAGPUR - 721302, INDIA

CERTIFICATE

This is to certify that the project report entitled "**Multiple Choice Questions(MCQs) Generation**" submitted by **Gangaram Sudewad** (Roll No. 20CS30017) to Indian Institute of Technology Kharagpur towards partial fulfilment of requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering is a record of bona fide work carried out by him under my supervision and guidance during Spring Semester, 2023-24.

Date: April 2, 2024

Place: Kharagpur

Professor Sudeshna Sarkar
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
Kharagpur - 721302, India

Abstract

Name of the student: **Gangaram Sudewad** Roll No: **20CS30017**

Degree for which submitted: **Bachelor of Technology**

Department: **Department of Computer Science and Engineering**

Thesis title: **Multiple Choice Questions(MCQs) Generation**

Thesis supervisor: **Professor Sudeshna Sarkar**

Month and year of thesis submission: **April 2, 2024**

Automatic generation of multiple-choice questions (MCQs) from text has emerged as a prominent research field due to the widespread acceptance of MCQs for large-scale assessments across various domains. Despite their utility, manual MCQ generation is both costly and time-consuming. Consequently, since the late 1990s, researchers have been increasingly drawn to automatic MCQ generation. Numerous systems have been developed in this pursuit, prompting a systematic review in this paper. Our study presents the outcomes of this review, along with a structured workflow encompassing phases for automatic MCQ generation. In each phase, we explore and discuss the techniques found in the literature. Additionally, we examine evaluation techniques employed to gauge the quality of system-generated MCQs. Finally, we pinpoint areas for future research aimed at enhancing the existing literature.

Contents

Abbreviations

QG	**Q**uestion **G**eneration
QA	**Q**uestion **A**nswering
DG	**D**istractor **G**eneration
MCQ	**M**ulitple **C**hoice **Q**uestion
PAD	**Pad**ding
SEP	**Sep**erator
AQG	**A**utomatic **Q**uestion **G**eneration
NLP	**N**autral **L**anguage **P**rocessing
XML	**E**xtensible **M**arkup **L**anguage
LoRA	**L**ow **R**ank **A**daption
PEFT	**P**aramter **E**fficient **F**ine **T**uning
SQuAD	**S**tanford **Qu**estion **A**nswering **D**ataset
RACE	**R**e**A**ding **C**omprehension dataset from **E**xaminations

Chapter 1

Mulitple Choice Question Generation

1.1 Introduction

Automating multiple-choice question (MCQ) generation from text is pivotal for educational assessment, streamlining a laborious process. This report presents a unique approach utilizing advanced natural language processing (NLP) models for automatic MCQ generation. Our approach, unlike traditional methods, integrates flanT5, distilbert, and T5 models for MCQ generation. This multi-model synergy ensures accuracy and complexity. In contrast, other methods such as rule-based, template-based, machine learning, crowdsourcing, text summarization, and domain-specific heuristics are different from NLP models. This approach is built upon robust datasets. We utilized the SQuAD dataset for question generation and question answering components, leveraging its rich context-question-answer triples. For distractor generation, we employed the RACE [4] dataset, which provides diverse passages and associated multiple-choice questions, enabling the creation of plausible distractors.

1.2 Problem Definition

MCQ generation aims to generate high quality question to the growing demand for e-learning and remote education it has potential to be a valuable tool for a wide range of NLP task. Developing a methodology for automatic MCQ generation using NLP models. Evaluating the accuracy and effectiveness of the generated MCQs compared to manually created questions.
For the best results, sentences should be well-formed and use proper English grammar. Here are a few examples.

- *John* drove the car to work. → Who drove the car to work? *John*
- The pump is now operational. → Is the pump operational? Yes
- He waters the garden every day. → What does he do every day? waters the garden

Multiple Choice Questions:

1. **Context:**John drove the car to work.
   Question: Who drove the car to work?
   (i) John (ii) Mary (iii) David (iv) Sarah
2. **Context:**The pump is now operational.
   Question: What is the current status of the pump?
   (i) Out of orders (ii) Operational (iii) Under maintenance (iv) Unknown
3. **Context:**He waters the garden every day.
   Question: What is his daily activity?
   (i) Plants trees (ii) Mows the lawn (iii) Waters the garden (iv) Paints the fence

Chapter 2

Used and Related Methods

2.1 Related work for Multiple Choice Question Generation

Numerous efforts have been made to automate the interpretation of natural human languages, with most focusing on solving specific sub-problems.
Aldabe et al.[1] introduced ArikIturri, an Automatic Question Generation (AQG) system tailored for Basque language test questions. It utilizes linguistically analyzed real corpora encoded in XML markup language as its information source.

Distractor Generation in ArikIturri

ArikIturri employs several techniques to generate distractors for multiple-choice and error correction questions. For noun inflection forms, it utilizes replacement and duplication of declension cases, number variations, and inflection paradigms. When generating distractors for verb forms, ArikIturri changes subject and object persons, verb modes, tenses, aspects, and verb paradigms. These techniques are parameterized within the generator, allowing for flexibility and improvement based on research findings. By employing these techniques, ArikIturri ensures diverse and contextually relevant incorrect options, enhancing the effectiveness of language assessment.

Chidinma et al.[5] proposed a method for automatically generating distractors for multiple-choice English vocabulary questions. This approach introduces novel sources for collecting distractor candidates and leverages semantic similarity and collocation information to rank these candidates effectively.

Bidyut et al.[2] explored the application of NLP on narrative texts to identify discourse connectives for AQG. Discourse connectives are words or phrases indicating relationships between logical sentences or phrases, implying linked verbal expression. The system first extracts text from user-supplied materials using NLP text processing concepts to generate questions.

Yuni et al.[6] focused on creating an automatic factual open cloze question generation system capable of generating fill-in-the-blank questions without alternatives. The system initially extracts informative sentences from the input corpus based on Part-of-Speech tagging rules.

Folajimi et al.[3] developed a system capable of generating various logical questions from given text inputs. It employs a three-step strategy: selecting the best potential set of sentences for question generation, determining the subject and context of each sentence for core agenda identification (Gap Selection), and analyzing the optimal question form that can be derived from the sentence (Question Formation).

2.2 Method used for MCQ generation

The process of automatically generating multiple-choice questions (MCQs) involves several distinct steps. Initially, the flanT5 model takes charge of question generation. Trained specifically for this task, it analyzes a provided passage or context to craft pertinent questions tailored to its content. Subsequently, these questions are passed to the distilbert model, specializing in question answering, to pinpoint the most fitting answers within the context or passage at hand.

Once the correct answers are identified, the T5 model kicks in for distractor generation. Distractors, serving as the incorrect options in MCQs, are meticulously crafted by the T5 model to accompany the correct answers. These plausible distractors enrich the multiple-choice options, presenting meaningful challenges to test comprehension.

Finally, integration of all components culminates in the creation of complete MCQs. This pivotal phase merges the questions generated by the flanT5 model with their corresponding correct answers from the distilbert model and the distractors from the T5 model. The result is a comprehensive set of MCQs, ready for use in assessments or educational materials, effectively gauging knowledge and understanding.

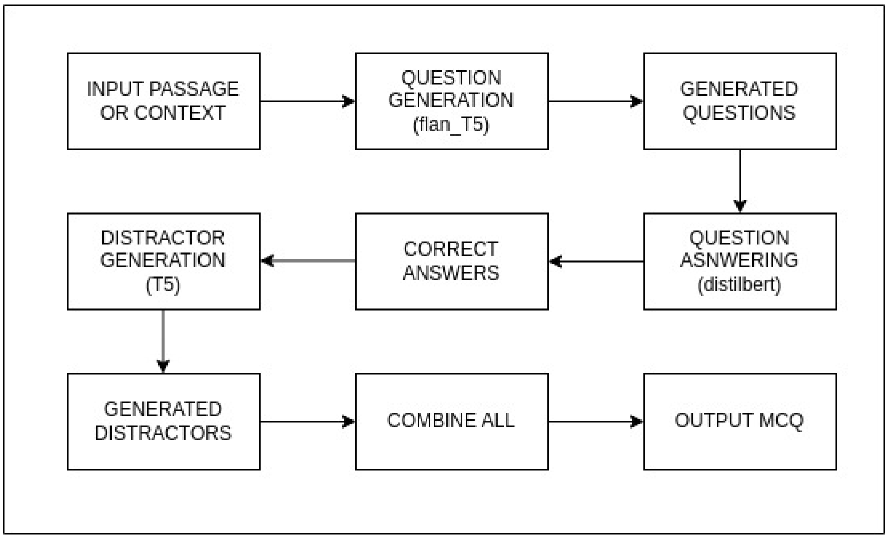

Figure 2.1: Flowchart of overall method

Chapter 3

Sequential Phases for Multiple Choice Question Generation

3.1 Question generation

3.1.1 Preprocessing input

Lowercasing: Convert all text to lowercase to ensure consistency and reduce the vocabulary size.

Tokenization: Break the text into individual words or subwords (tokens) using a tokenizer. This step is essential for converting the raw text into a format that the model can understand.

Removing Stopwords: Remove common words (e.g., "the", "is", "and") that carry little semantic meaning and may not contribute significantly to question generation.

Removing Punctuation: Remove punctuation marks from the text as they are often unnecessary for question generation and can potentially confuse the model.

Sentence Segmentation: Split the text into individual sentences if the context consists of multiple sentences. This can help the model focus on generating questions for each individual piece of information.

Pre-Processing Example

Text Cleaning:

The context is cleaned to remove any unwanted characters or formatting. For example, newline characters (\n) may be removed to ensure that the text is in a consistent format without unnecessary line breaks.

```
"World number one Novak Djokovic says he is hoping for a 'positive decision'..."
```

Original context

```
"World number one Novak Djokovic says he is hoping for a 'positive decision'..."
```

Cleaned context

Tokenization:

The cleaned context is tokenized, breaking it down into smaller units such as words or subwords. This step converts the text into numerical tokens that the model can process.

```
"World number one Novak Djokovic says he is hoping for a 'positive decision'..."
```

Original context

```
["World","number","one","Novak","Djokovic","says","he","is","hoping","a","'positive"]
```

Tokenized context

Encoding: The tokenized context is encoded, which means mapping each token to its corresponding numerical identifier (index) in the model's vocabulary. This step prepares the text for input into the model.

```
["World","number","one","Novak","Djokovic","says","he","is","hoping","a","'positive"]
```

Tokenized context

```
[124, 325, 56, 987, 234, 567, 89, 456, 123, 567, 890, 2345, 6789, ...]
```

Encoded context

3.1.2 Steps for Question Generation

Context Retrieval: The context is retrieved from the dataset using the specified index.

Training Transformer Model: Training involves parameter analysis, zero-shot inference, dataset tokenization, and advanced techniques like LORA configuration and PEFT fine-tuning. These steps enable the model to generate accurate questions from contexts through iterative training.

Prompt Construction: A prompt template is constructed by formatting the retrieved context into a template string. The prompt template typically includes some fixed text followed by the context and a placeholder for the question.
The start prompt and end prompt variables delineate the initiation and conclusion of the prompt, guiding the model to generate questions from the associated contexts in the dataset.

Prompt:
Generate Question from the following context. + {context} + Question

Tokenization: The prompt template is tokenized using the tokenizer. Tokenization converts the input text into numerical representations understandable by the model. The tokenized input is typically converted to PyTorch tensors for further processing.

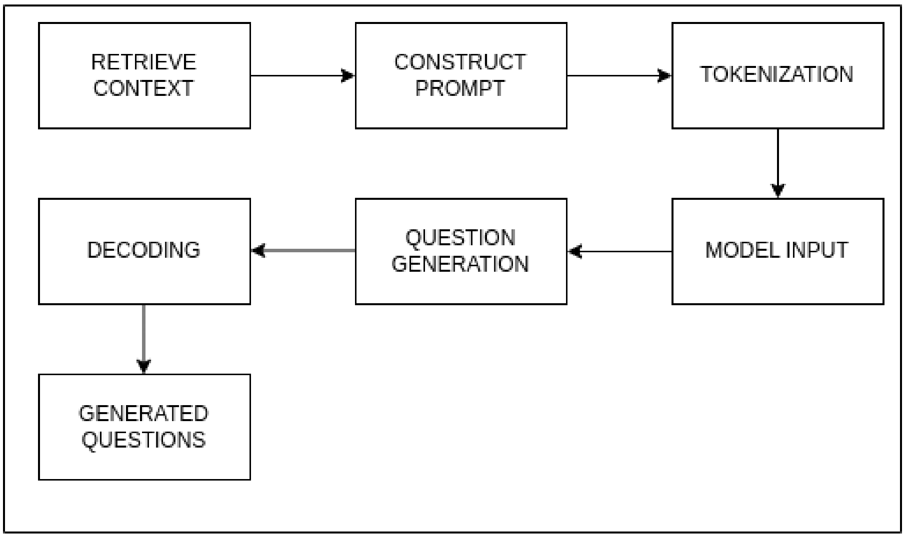

FIGURE 3.1: Flowchart of Question Generation

Model Input: The tokenized input is passed to the model(google/flan-t5-base) for question generation. The model generates the question based on the provided context. The model might be moved to the GPU for faster computation if available.

Question Generation: The model generates the question based on the tokenized input. Generation is performed using the generate() method of the model. The generated output is a sequence of token IDs representing the question.

Decoding: The generated token IDs are decoded using the tokenizer to obtain the final question text. Decoding involves converting token IDs back into human-readable text. Special tokens indicating the beginning and end of the sequence are typically skipped during decoding.

3.2 Question Answering

Training Model: The model adapts its parameters through iterative adjustments to minimize errors in predicting answers from context. The data is prepared, the model fine-tuned, and its performance evaluated periodically. Checkpoints are saved to capture progress, enabling training to be paused and resumed, or the model to be deployed for use.

Load Trained QA Model: This step involves loading a pre-trained question answering (QA) model. The model has been trained on a large dataset to understand natural language questions and provide accurate answers based on a given context.

Provide Data (Context, Question): In this step, the context and question are provided as input to the QA model. The context is the passage of text from which the model will generate an answer, and the question is the query posed to the model.

Preprocess Input Data (Tokenization, Padding, etc.): The input data (context and question) undergo preprocessing, which may include tokenization (splitting the text into individual tokens), padding (ensuring all inputs are of the same length), and other necessary transformations to prepare the data for input to the model.

Question: "What is the capital of France?" **Context**: "France is a country located in Europe. Its capital is Paris." **Answer**: "Paris" (starting at character position 45 in the context).

Tokenized input:

$$\begin{bmatrix} \text{[CLS]} & \text{what} & \text{is} & \text{the} & \text{capital} & \text{of} & \text{france} & \text{?} & \text{[SEP]} & \text{france} \\ \text{is} & \text{a} & \text{country} & \text{located} & \text{in} & \text{europe} & \text{.} & \text{its} & \text{capital} & \text{is} \\ \text{paris} & \text{.} & \text{[SEP]} & & & & & & & \end{bmatrix}$$

Offset mappings:

$$\begin{bmatrix} (0,0) & (0,4) & (5,7) & (8,11) & (12,19) & (20,22) & (23,29) & (29,30) & (30,31) \\ (32,38) & (39,41) & (42,43) & (44,51) & (52,59) & (60,62) & (63,69) & (69,70) & (71,74) \\ (75,82) & (83,85) & (86,88) & (89,94) & (94,95) & & & & \end{bmatrix}$$

Make Prediction: The preprocessed input data is fed into the trained QA model (distilbert-base-uncased), which generates a prediction for the answer to the question based on the provided context. The model utilizes its learned parameters and

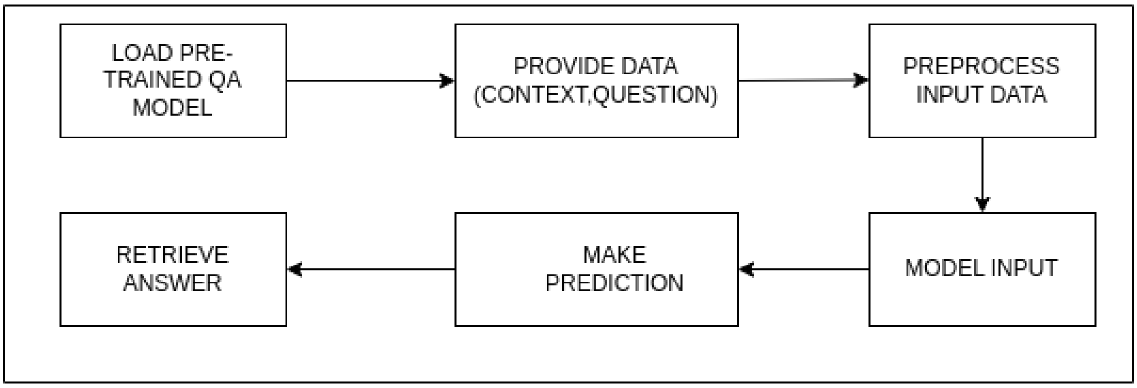

Figure 3.2: Flowchart of Question Generation

knowledge from pre-training to generate the answer.

Retrieve Answer from Model: Finally, the generated answer is retrieved from the model's output. This answer represents the model's prediction in response to the given question and context. The answer can then be used for further analysis or presented to the user.

3.3 Distractor Generation

Tokenize Text: Tokenization involves breaking down the text into smaller units, such as words or subwords, known as tokens. This process creates a structured representation of the text that the model can understand.

Encode Text: In the encoding step, each token is mapped to its corresponding numerical identifier in the model's vocabulary. This conversion transforms the text into numerical representations, facilitating processing by machine learning models.

```
input_text = question + ’ ’ + separator + ’ ’ + answer + ’ ’ + separator
+ ’ ’ + context
```

Prepare Input: The encoded tokens of the question, answer, and context are combined into a single input suitable for the model. This step organizes the input data into a format that the model can consume for further processing.

Train Model: During training, the RACE dataset is prepared for training and validation, and a T5 model is initialized and optimized with the Adam optimizer. The training loop iterates over epochs, computing loss for each batch, updating model parameters, and monitoring training loss. Validation is performed after each epoch to assess model performance.

Generate Distractors: Using a pre-trained model, such as T5, distractors are generated based on the input text. Distractors are alternative options provided in multiple-choice questions, contributing depth and complexity to the question.

Raw output generated by the model:

`<pad> Djokovic’s wish for a "positive decision"<sep> An Australian Open in Dubai<sep> Indian Wells and the Miami Open</s>`

After removing special tokens:

`Djokovic’s wish for a "positive decision"<sep> An Australian Open in Dubai<sep> Indian Wells and the Miami Open`

Final list of options:

`['Djokovicś wish for a "positive decision"', 'An Australian Open in Dubai', 'Indian Wells and the Miami Open']`

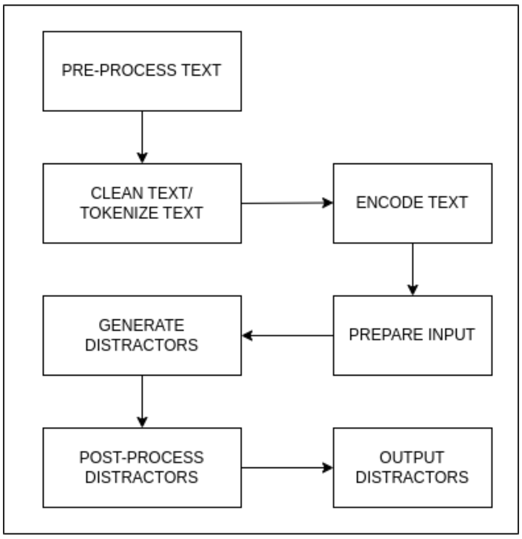

FIGURE 3.3: Flowchart of overall method

Post-process Distractors: Following distractor generation, the generated distractors undergo post-processing to clean up any special tokens or unwanted characters. This ensures that the distractors are coherent and suitable for use in the final multiple-choice questions.

Output Distractors: The final step involves outputting the generated distractors for further use or evaluation. These distractors can be seamlessly integrated into the final question generation pipeline or utilized independently for various purposes.

3.4 Result

In the analysis of the model-generated question, it's evident that there is a discrepancy between the intended question and the question generated by the model. This highlights a potential area for improvement in the model's understanding of context and its ability to generate relevant questions.
Some of the sample generated are provided below:

'id': '573382d24776f41900660c39', 'title': 'Warsaw',
'question': 'What was Warsaw ranked the 7th greatest of?', 'answers': 'text': ['emerging market', 'emerging market', 'emerging market'], 'answer$_{s}tart'$: $[470, 470, 470]$

Model Generated: In 2006, how many companies were registered in Warsaw?
In what year was the GDP per capita in Warsaw estimated at over 650 million euro?

What was fought between the colonies of British America and New France?
Most Similar Ground Truth Qns: Who fought in the French and Indian war?
Similarity Score: 0.648 (64.8%)

So from question generation we observed that average similarity score is 0.568 and the maximum and minimum value of similarity score is 0.676 and 0.144 respectively.

TABLE 3.1: Training Loss

Step	Training Loss
200	2.729700
400	0.093100
600	0.082500
800	0.078700
1000	0.076300
1200	0.074600
1400	0.074200
1600	0.070800
1800	0.073300
2000	0.071900

TABLE 3.2: Training loss of Question generation

Training the flan_5 model used for question generation where, Input: Tokenized context-question pairs for training. Output: Model predictions (tokenized questions) and loss during training. Training Data Samples: 87,599 samples from the SQuAD training split. Testing (Validation) Data Samples: 10,570 samples from the SQuAD validation split.

Now for the question answering part, training the distilbert model where,

Input during Training: Tokenized context-question pairs with answer positions. Output during Training: Predicted start and end positions of answers used for computing loss. Training Data Samples: Consist of approximately 80,000 tokenized context-question pairs from the SQuAD training split. Testing (Validation) Data Samples: Include around 10,000 tokenized context-question pairs from the SQuAD validation split, utilized for evaluating model performance

TABLE 3.3: Training and Validation Loss

Epoch	Training Loss	Validation Loss
1	4.7657	4.7041
2	4.4165	4.3591
3	4.2548	4.1965
4	4.1621	4.1062
5	4.1196	4.0519
6	4.0845	4.0193
7	4.0657	4.0018
8	4.0650	3.9961

TABLE 3.4: Training & validation loss of QA

For the question answering part the example outputs are given as follows:

Question: How many programming languages does BLOOM support?

Context: BLOOM has 176 billion parameters and can generate text in 46 natural languages and 13 programming languages.

Ground Truth: 46 natural languages and 13 programming languages **Generated**

Answer: 176 billion parameters and can generate text in 46 natural languages.
Similarity Score: 0.4999

Question: What is Warsaw's economy characterized by?

Context: " Warsaw's economy, by a wide variety of industries, is characterised by FMCG manufacturing, metal processing, steel and electronic manufacturing and food processing. "

Ground Truth: FMCG manufacturing, metal processing

Predicted Answer: metal processing

Similarity Score: 0.7071067811865475

Training the T5 model for distractor generation where, input consists of concatenated question-answer-context triples for training and generated distractors for testing (validation). There are 87,866 training samples. There are 4,887 testing (validation) samples.

```
batch_size: 8
num_workers: 0
num_epochs: 1
max_length: 512
Downloading readme: 100% 11.0k/11.0k [00:00<00:00, 24.4MB/s]
Downloading data: 100% 2.08M/2.08M [00:00<00:00, 3.99MB/s]
Downloading data: 100% 37.4M/37.4M [00:02<00:00, 13.1MB/s]
Downloading data: 100% 2.05M/2.05M [00:00<00:00, 6.73MB/s]
Generating test split: 100% 4934/4934 [00:00<00:00, 65372.45 examples/s]
Generating train split: 100% 87866/87866 [00:00<00:00, 114704.18 examples/s]
Generating validation split: 100% 4887/4887 [00:00<00:00, 142945.76 examples/s]
RaceQuestionAnswerGeneration Initialized
len_train_data: 87866
RaceQuestionAnswerGeneration Initialized
len_valid_data: 4887

Starting the training of T5-model from scratch

#parameters: 60506624
2024-03-27 09:42:57.701648, Epoch: 1/1  |  1/10984  |  loss = 6.41240120
Saved at /content/drive/MyDrive/Distractor_Finetune/save_dir/t5-small-Race-Distractor-Generation-version0-step0.pt
############
Valid Loss = 7.231214
Model improved
2024-03-27 09:43:57.476265, Epoch: 1/1  |  2/10984  |  loss = 6.50421000
2024-03-27 09:43:57.775835, Epoch: 1/1  |  3/10984  |  loss = 6.79857922
2024-03-27 09:43:58.050006, Epoch: 1/1  |  4/10984  |  loss = 11.10650539
2024-03-27 09:43:58.353711, Epoch: 1/1  |  5/10984  |  loss = 5.66474533
2024-03-27 09:43:58.642951, Epoch: 1/1  |  6/10984  |  loss = 7.72894144
2024-03-27 09:43:58.919181, Epoch: 1/1  |  7/10984  |  loss = 8.01847076
2024-03-27 09:43:59.182853, Epoch: 1/1  |  8/10984  |  loss = 6.53509426
2024-03-27 09:43:59.490442, Epoch: 1/1  |  9/10984  |  loss = 6.14823103
2024-03-27 09:43:59.744949, Epoch: 1/1  |  10/10984 |  loss = 6.68600225
```

FIGURE 3.4: Training for Distractor Generation

One example output of the generated distractors is shown below:

---

context: World number one Novak Djokovic says he is hoping for a "positive decision" to allow him to play at Indian Wells and the Miami Open next month. The United States has extended its requirement for international visitors to be vaccinated against Covid-19. Proof of vaccination will be required to enter the country until at least 10 April, but the Serbian has previously said he is unvaccinated. The 35-year-old has applied for special permission to enter the country. Indian Wells and the Miami Open - two of the most prestigious tournaments on the tennis calendar outside the Grand Slams - start on 6 and 20 March respectively. Djokovic says he will return to the ATP tour in Dubai next week after claiming a record-extending 10th Australian Open title and a record-equalling 22nd Grand Slam men's title last month.

Without Finetuning: What is the best title for the passage?

Djokovic's application for special permission to enter the United States

{Djokovic's preparation for the Miami Open, Djokovic's application for special permission to enter the United States, "Djokovic's preparation for the Miami Open", "Djokovic's preparation for the Miami Open"}

question: "What is the best title for the passage?"

answer: "Djokovic's application for special permission to enter the United States"

Generated Distractors:

Ex.1: **Q:** What is the best title for the passage?

A: New Rules for international visitors **B:** Djokovic's challenge

C: Djokovic's application for special permission to enter the United States

D: World number two Novak Djokovic's dream

Correct: Djokovic's application for special permission to enter the United States

Ex.2: **Q:** Why is Djokovic hoping for a positive decision?

A: To be unvaccinated **B:** To receive proof of vaccination

C: To be allowed to play at Indian Wells and the Miami Open

D: To be unvaccinated

Correct: To be allowed to play at Indian Wells and the Miami Open

Ex.3: Q: How many Grand Slam men's titles has Djokovic won?

A: 3 **B:** 16 **C:** 22 **D:** 10

Correct: 22

Ex.4: Q: Who had a goal ruled out for offside?

A: Ben Chilwell **B:** Joao Felix **C:** Kai Havertz **D:** Kai Havertz

Correct: Joao Felix

E5: Which of the following best describes the families of the astronauts on the ISS?

A: They are generous and brave. **B:** They are caring and thoughtful.

C: They are kind and brave. **D:** They are strong and hard-working.

Correct: They are caring and thoughtful.

Ex.6: Q: Where did the clerk find the necklace at last?

A: On the tree. **B:** On the trees.

C: In the river. **D:** In the lake.

Correct: On the tree.

Evaluating the similarity between the correct answer and each distractor using BERT embeddings based on the cosine similarity metric.

SIMILARITY SCORES

Ex. No.	Distractor1	Distractor2	Distractor3
1	0.63207	0.70482	0.74962
2	0.45337	0.53134	0.45337
3	0.83514	0.84683	0.83109
4	0.50160	0.50563	0.50563
5	0.92177	0.93989	0.80415
6	0.93919	0.83458	0.83636

FIGURE 3.5: Distractor similarity scores

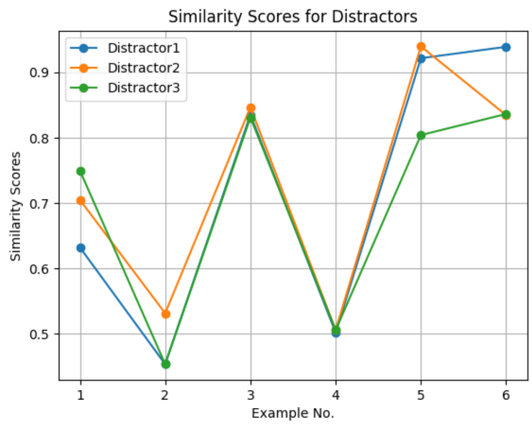

FIGURE 3.6: Similarity scores

3.5 Future Works and Upgrades for Multiple Question Generation

- **Ensemble Models:** Implementing ensemble models that combine the outputs of multiple models could potentially improve the overall quality and diversity of the generated questions and distractors.

- **Data Augmentation:** Augmenting the training data with additional examples and variations could help enhance the models' ability to generalize and generate more diverse and contextually relevant questions.

- **Incorporating Feedback Mechanisms:** Implementing feedback mechanisms where users can provide feedback on the generated questions and distractors could help refine the models over time and improve their performance.

- **Multi-Modal Input:** Integrating other modalities such as images, audio, or video along with text could enable the models to generate questions and distractors that are more comprehensive and contextually relevant.

- **Evaluation Metrics:** Developing robust evaluation metrics to quantitatively assess the quality and effectiveness of the generated questions and distractors could provide valuable insights for further model improvement.

- **Deployment and Integration:** Integrating the MCQ generation system with existing educational platforms or assessment tools could facilitate its real-world deployment and usage in educational settings.

Bibliography

[1] Itziar Aldabe, Maddalen Lopez De Lacalle, Montse Maritxalar, Edurne Martinez, and Larraitz Uria. Arikiturri: an automatic question generator based on corpora and nlp techniques. In *Intelligent Tutoring Systems: 8th International Conference, ITS 2006, Jhongli, Taiwan, June 26-30, 2006. Proceedings 8*, pages 584–594. Springer, 2006.

[2] Bidyut Das and Mukta Majumder. Factual open cloze question generation for assessment of learner's knowledge. *International Journal of Educational Technology in Higher Education*, 14:1–12, 2017.

[3] YO Folajimi and OE Omojola. Natural language processing techniques for automatic test questions generation using discourse connectives. *Journal of Computer Science and Its Application*, 20(2):60–76, 2013.

[4] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.

[5] Chidinma A Nwafor and Ikechukwu E Onyenwe. An automated multiple-choice question generation using natural language processing techniques. *arXiv preprint arXiv:2103.14757*, 2021.

[6] Yuni Susanti, Takenobu Tokunaga, Hitoshi Nishikawa, and Hiroyuki Obari. Automatic distractor generation for multiple-choice english vocabulary questions. *Research and practice in technology enhanced learning*, 13(1):15, 2018.