Generative AI for Automatic Question Generation: Evaluation and Comparison of Models

Abtaal Aatif
Department of Computer Science
National University of Computer and Emerging Sciences
Islamabad, Pakistan
i212990@nu.edu.pk

Umair Khalid
Department of Computer Science
National University of Computer and Emerging Sciences
Islamabad, Pakistan
i210455@nu.edu.pk

Ali Umer
Department of Computer Science
National University of Computer and Emerging Sciences
Islamabad, Pakistan
i210380@nu.edu.pk

***Abstract*—Generative AI has made much progress in the field of natural language processing, especially in sub-fields like machine translation, text summarization, and others. This study will focus in on the use of generative AI in question generation in particular. This slightly differs from the more common use of generative AI for answer generation in response to a question. For the sake of this study, we will be looking into the use of models for boolean question generation, multiple-choice question generation, and especially comprehension question generation. The models we intend on using are BERT, BART, and the small and base variants of the T5 transformer model. Depending on the task, the models are trained on the SQuAD, RACE, or BoolQ. The final models will be evaluated using the metrics: BLEU, METEOR, ROUGE 1, ROUGE 2, and ROUGE L. These will help in the comparative analysis of models, helping us highlight which models to use for different kinds of question generation.**

***Index Terms*—Generative AI, Question Generation, T5, BART, BERT, BLEU, ROUGE, METEOR**

I. Introduction

Generative AI has been making fast progress in recent years, especially in natural language processing (NLP), with the rise of Large Language Models (LLMs). AI is being used for numerous purposes including text summarization and answer generation. However, the field of automatic question generation (AQG) still requires much development.

This study intends on focusing on the use of generative AI models for question generation in 3 categories: Boolean question generation, multiple-choice question generation, and comprehension question generation. While models and papers have been proposed, much research still needs to be done. We hope this study sheds some light on the use of advanced models for generating realistic, accurate, logically correct questions.

The models we intend on using include:

- Boolean Question Generation using T5 small (trained on BoolQ).
- Multiple-Choice Question Generation using T5 base (trained on RACE).
- Comprehension Question Generation using T5 base (trained on SQuAD).
- Comprehension Question Generation using BERT (trained on SQuAD).
- Comprehension Question Generation using BART (trained on SQuAD).

These models show varying performance based on the datasets they are trained on and the purposes they perform. As mentioned prior, the 3 datasets used are SQuAD, RACE, and BoolQ. These datasets help provide data for helping our models perform a variety of types of question generation.

The BLEU, METEOR, ROUGE 1, ROUGE 2, and ROUGE L metrics are used for evaluating the models, with higher scores representing better performance by the models. These metrics will help us in model comparison for the same tasks as well as when carrying out different types of question generation.

The primary contributions of this work are summarized as follows:

- Evaluating BERT, BART, and T5 models for Boolean, multiple-choice, and comprehension question generation tasks.
- Comparative analysis of models based on metrics like BLEU, METEOR, and ROUGE.
- Insights into the effectiveness of each model for different types of question generation tasks.

The remainder of this paper is divided as follows: Section II surveys related work on question generation using generative AI. Section III outlines methodology-including model specification and strategies for training-and Section IV describes the experimental setup including evaluation metrics. Section V contains the results and their comparative analysis, while Section VI puts together a summary of findings around challenges and future work that can be pursued in the area of question generation.

II. Related Work

In recent years, AQG is gaining traction as a commonly used task in NLP. This has come up mainly in the areas of educational technology, information retrieval, and in conversational AI. Most traditional forms of question-answering systems typically generate answers for predefined queries. However, the real challenge is to create meaningful questions from text. This task is quite complex and so early approaches to the research in AQG would often be rule-based or template driven where the question generation was dependent on predefined sentence structures to create a question. Such methods cannot be flexible or easily scalable.

Much of the developments in AQG have been led with the use of deep learning models, especially of the transformer architecture family. Such success has been seen with models like BERT, GPT, and T5. These models allow the generation of more diverse and contextually appropriate questions. There have been efforts by researchers on fine-tuning the pre-trained models on specific types of questions, which may be Boolean, multiple-choice, or comprehension. The use of datasets like BoolQ, SQuAD, and RACE has played a major role in developing models that produce high-quality context-specific questions [1], [2].

A new development in AQG involves research into developing a new standard to gauge performance rather than rely on BLEU, ROUGE, and METEOR. These older metrics do not quite capture the nuances of good, well generated questions. To address this, new metrics like QGEval and QG-Bench have been intoduced. Such new metrics evaluate generated questions across things like fluency, consistency, and relevance [3], [1]. These approaches can hopefully better assist us in developing and analysing more complex AQG models, keeping mind how their generated output is to be used and interpreted by human readers.

Despite the progress this sub-field has made, it is still relatively new. A number of challenges yet remain to be solved including computational costs and generalizability of the models. Ongoing research is trying to tackle all of these problems. For example, BERT has been fine-tuned for question generation, providing satisfactory results with reduced resource requirements, similar in quality to larger models [2].

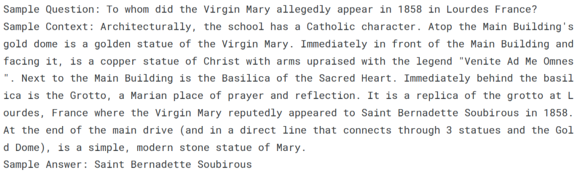
In conclusion, AQG is progressing rapidly, with transformer models and new evaluation metrics playing central roles. Many performance and question quality challenges still remain on the way to handle the complexities of question generation across diverse domains. Future research will likely focus on optimizing model efficiency, improving question quality, and developing better evaluation methodologies to accurately gauge models.

III. Dataset

This study employs three datasets — SQuAD, RACE, and BoolQ — to evaluate the train our generative AI models for AQG. These datasets provide diverse contexts and question types, our models to specialize in multiple different kinds of AQG with sufficiently decent results.

A. Dataset Description

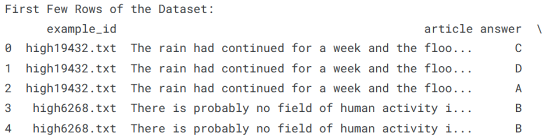
- **SQuAD (Stanford Question Answering Dataset):** SQuAD is a widely employed dataset for question-answering tasks, containing close to 107,785 question-answer pairs derived from Wikipedia passages [4]. It is designed such that each question has an answer that is some continuous segment of text of the corresponding passage, so it is perfectly suited to comprehension-based question generation.



```
Sample Question: To whom did the Virgin Mary allegedly appear in 1858 in Lourdes France?
Sample Context: Architecturally, the school has a Catholic character. Atop the Main Building's
gold dome is a golden statue of the Virgin Mary. Immediately in front of the Main Building and
facing it, is a copper statue of Christ with arms upraised with the legend "Venite Ad Me Omnes
". Next to the Main Building is the Basilica of the Sacred Heart. Immediately behind the basil
ica is the Grotto, a Marian place of prayer and reflection. It is a replica of the grotto at L
ourdes, France where the Virgin Mary reputedly appeared to Saint Bernadette Soubirous in 1858.
At the end of the main drive (and in a direct line that connects through 3 statues and the Gol
d Dome), is a simple, modern stone statue of Mary.
Sample Answer: Saint Bernadette Soubirous
```

Fig. 1. Sample for SQuAD Dataset

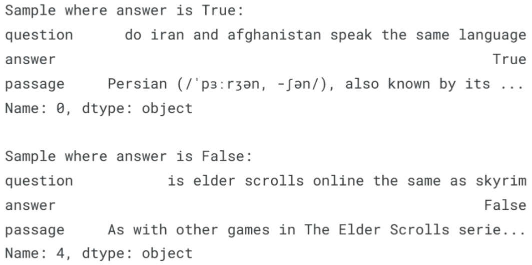
- **RACE (ReAding Comprehension Dataset):** RACE is the multiple-choice question answering dataset having about 100,000 questions [5]. It has two subsets — RACE-M containing 28,000 questions from the middle school English exams and RACE-H having 72,000 questions from the high school English exams. For our use case, we have taken RACE-A, which contains the combination of RACE-H and RACE-M.



```
First Few Rows of the Dataset:
      example_id                                            article answer  \
0  high19432.txt  The rain had continued for a week and the floo...      C
1  high19432.txt  The rain had continued for a week and the floo...      D
2  high19432.txt  The rain had continued for a week and the floo...      A
3   high6268.txt  There is probably no field of human activity i...      B
4   high6268.txt  There is probably no field of human activity i...      B
```

Fig. 2. Sample RACE Dataset

- **BoolQ (Boolean Questions):** BoolQ is a dataset of approximately 16,000 Boolean (yes/no) questions created by processing Wikipedia passages [6]. The samples are comprised of a question, a passage, and a true/false label, making this particularly suitable for the generation of Boolean questions.



```
Sample where answer is True:
question      do iran and afghanistan speak the same language
answer                                                   True
passage     Persian (/ˈpɜːrʒən, -ʃən/), also known by its ...
Name: 0, dtype: object

Sample where answer is False:
question           is elder scrolls online the same as skyrim
answer                                                  False
passage     As with other games in The Elder Scrolls serie...
Name: 4, dtype: object
```

Fig. 3. Sample BoolQ Dataset

B. Data Preprocessing

The datasets were processed to make them suitable for our models in the following manner:

- **Tokenization:** All text inputs (passages and questions) were tokenized using pre-trained tokenizers. Each model's own specific tokenizer was used for the data being fed to that model (e.g: BART tokenizer for BART).
- **Text Normalization:** Special characters and unnecessary whitespaces were removed.
- **Truncation and Padding:** Text inputs were truncated or padded to a fixed length to match the input size requirements of the models.

These preprocessing steps enable our data to be structured in a manner that can be fed to our models.

C. Data Distribution

The variety of types of samples in the datasets allow our models to generate different kinds of questions. Table I presents the number of samples in each dataset, highlighting their scale and diversity.

TABLE I
DATASET DISTRIBUTION

Dataset	Samples	Question Type	Source
SQuAD v1.1	107,785	Comprehension	Wikipedia
RACE-M	28,000	Multiple-choice	English Exams
RACE-H	72,000	Multiple-choice	English Exams
BoolQ	16,000	Boolean (Yes/No)	Wikipedia

These datasets provide sufficient means for training and evaluating our generative AI models across different AQG tasks.

IV. Proposed Methodology

This section describes the methodology to develop and compare various generative AI models for AQG. We focus on three models — BERT, BART, and T5 (small and base) — and explain their architectures, training procedures, and the mathematics necessary to generate boolean, multiple-choice, and comprehension questions. The overall approach follows a sequence-to-sequence paradigm: the model takes text as input and generates multiple questions based on the provided context.

A. Model Architectures

1) T5 Model: The T5 (Text-to-Text Transfer Transformer) model is based on the encoder-decoder architecture of transformers [7]. The input and output of T5 are both treated as text sequences, allowing the model to be applied to a wide variety of NLP tasks by framing them in a unified text-to-text framework.

For question generation, we fine-tune T5 on datasets such as SQuAD, RACE, and BoolQ, with the goal of learning to map input passages to corresponding questions. The general formulation of the T5 model for this task is as follows:

$$P(Q|P) = \text{softmax}(W_2 \cdot \text{Decoder}(P, \theta))$$

where: - P is the input passage, - Q is the generated question, - W_2 is the output weight matrix, - Decoder(P, θ) refers to the decoder function that takes the input passage and generates the question, with parameters θ.

The model is trained using the standard cross-entropy loss, which minimizes the negative log-likelihood of the correct question sequence given the passage.

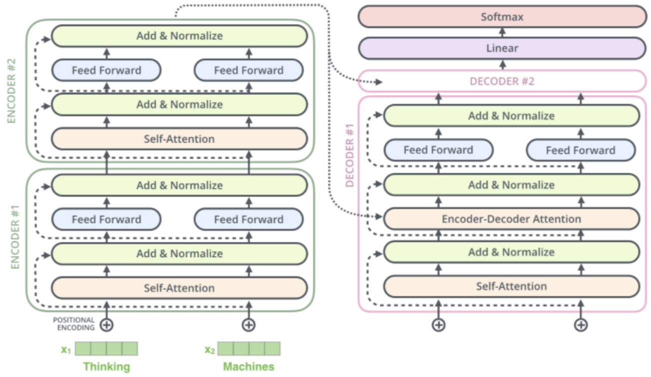


Fig. 4. T5 Model Architecture

2) BERT Model: BERT (Bidirectional Encoder Representations from Transformers) is made for masked language modeling, making it effective for comprehension question generation. Unlike T5, BERT does not have a decoder component. Instead, it relies on a bidirectional encoder that reads the entire sequence of input tokens simultaneously [8].

For question generation with BERT, certain parts of the input passage are masked and the model is trained to predict the masked question. The goal is to generate a question Q by predicting sections of text from the passage P. This can be formalized as:

$$P(Q|P) = \prod_{i=1}^{n} P(q_i|P)$$

where n is the length of the output question, and q_i is the i-th word in the generated question.

BERT uses the following objective function for training:

$$\mathcal{L} = -\sum_{i=1}^{n} \log P(q_i|P)$$

where the loss is computed over the entire sequence of generated question tokens.

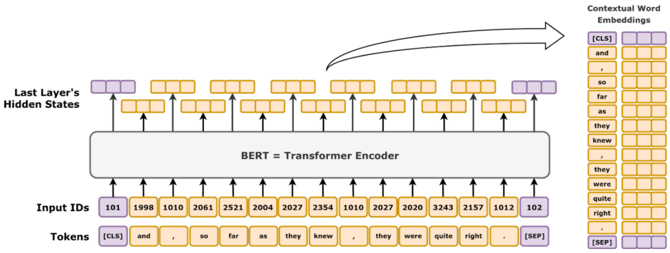

Contextual Word Embeddings
Last Layer's Hidden States
BERT = Transformer Encoder
Input IDs
101 1998 1010 2061 2521 2004 2027 2354 1010 2027 2020 3243 2157 1012 102
Tokens
[CLS] and , so far as they knew , they were quite right . [SEP]

Fig. 5. BERT Embedding Structure

3) BART Model: BART (Bidirectional and Auto-Regressive Transformers) combines the strengths of both BERT and GPT (Generative Pre-trained Transformer). BART is a denoising autoencoder, meaning it first corrupts the input sequence and then reconstructs it. BART's encoder is similar to BERT, while the decoder is autoregressive, similar to GPT [9].

For question generation, BART is fine-tuned using datasets like SQuAD and RACE. The formulation for generating a question Q from an input passage P can be written as:

$$P(Q|P) = \text{softmax}(W_2 \cdot \text{Decoder}(\text{Encoder}(P))),$$

where: - $\text{Encoder}(P)$ processes the input passage P, - $\text{Decoder}(\text{Encoder}(P))$ generates the question sequence based on the encoded passage, - W_2 is the output weight matrix.

The model is trained using the negative log-likelihood loss function, similar to T5 and BERT.

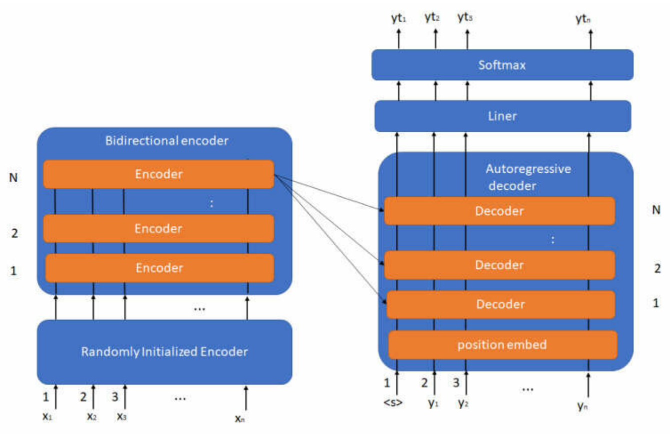


Fig. 6. BART Model Architecture

B. Training Strategy

We import pretrained models and then apply transfer learning by further fine-tuning them on our specified dataset (SQuAD, RACE, or BoolQ). The training procedure involves:

- Input preparation: For comprehension question generation, the passage is provided as input, and the corresponding question is treated as the output. For Boolean and multiple-choice question generation, we modify the datasets accordingly.
- Fine-tuning: The models are fine-tuned on the target question generation tasks using the cross-entropy loss function to optimize the model parameters.
- Evaluation: We evaluate the models using standard NLP metrics such as BLEU, ROUGE, and METEOR.

C. Evaluation Metrics

To evaluate the quality of generated questions, we use the following metrics: BLEU, ROUGE-1, ROUGE-2, ROUGE-L, and METEOR. Further details of these metrics is given in the ”Experimental Results” section.

These metrics are typically used to evaluate machine-generated text, thereby giving insight into the level of syntactic as well as semantic accuracy.

D. Fine-Tuning and Hyperparameter Optimization

The models are fine-tuned on the datasets using the Adam optimizer with learning rates ranging from 5^{-5} to 10^{-4}, depending on the model. Early stopping can help prevent overfitting. We can utilize hyperparameters like batch size, epochs, and the learning after optimizing them via grid search and cross-validation.

E. Challenges and Limitations

While the study overall shapes up to provide much insight on AQG using generative models, several challenges still remain. Many of the models display a limited diversity of the type of questions they are capable of generating. Smaller models like T5 small may even end up losing semantic meaning or resort to generating factual statements instead. Additionally, computation cost is another issue, especially for large models like BART and T5, which require significant resources for both training and inference.

V. Experimental Results

This section reports the results for AQG based on the obtained models. Three different question generation tasks were experimented with: 1. Boolean Question Generation with T5 small and the BoolQ dataset. 2. Using T5 base and the RACE dataset, generate multiple-choice questions. 3. Generation of comprehension questions using T5 Base, BART, and BERT models on the SQuAD dataset.

A thorough validation of the models is performed using comparisons with baseline methods, ablation studies, and quantitative and qualitative results that also discuss the computational efficiency.

A. Experimental Setup

All experiments were conducted on a system with Tesla T4x2 GPUs, 32 GB of RAM, and a 16-core Intel Xeon CPU. The models were implemented in Python using PyTorch and PyTorch Lightning libraries. The training process involved running each model for multiple iterations with varying hyperparameters as mentioned in the accompanying notebooks.

The input text for training was tokenized using pre-trained tokenizers corresponding to each model. The datasets SQuAD, RACE, and BoolQ were used for training the models. For fine-tuning, passages were resized to fit the input length constraints of the models, and all questions were padded appropriately.

B. Evaluation Metrics

The performance of the models was evaluated using the following metrics:

- **BLEU**: Measures the precision of n-grams between the generated and reference questions.
- **METEOR**: Evaluates the precision, recall, and synonym matching between the generated and reference questions.
- **ROUGE-1**: Measures recall-oriented n-gram overlap between the generated and reference questions.
- **ROUGE-2**: Similar to ROUGE-1 but focuses on bigrams, giving more weight to the overlap between two consecutive words.
- **ROUGE-L**: Evaluates the longest common subsequence between the generated and reference questions, reflecting syntactic similarities.

These metrics help assess both syntactic and semantic quality in the generated questions.

C. Baseline Comparison

The models were simple enough where they can simply be compared to each other quite easily.

Thus, we will be comparing T5 Base, BART, and BERT for comprehension question generation. Furthermore, we also look at how well the T5 model performs for other types of AQG.

D. Quantitative Comparison

Table II presents the quantitative results comparing the proposed models with baseline models. Each model's performance is evaluated using BLEU, ROUGE-1, ROUGE-2, ROUGE-L, and METEOR.

TABLE II
PERFORMANCE COMPARISON WITH BASELINE MODELS

Model	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
T5-small 128M (BoolQ)	33.12	43.73	47.05	27.46	39.4
T5-base 382M (RACE)	41.9	29.1	52.93	36.74	54.12
T5-base 382M (SQuAD)	32.7	22.5	48.5	28.5	42.7
BERT 108M (SQuAD)	24.3	12.6	39.5	14.4	36.2
BART 510M (SQuAD)	38.1	27.4	51.5	31.6	52.94

Table II showcases the effectiveness of the proposed models on three different question generation tasks—Boolean, multiple-choice, and comprehension. When using T5-small for Boolean question generation with the BoolQ dataset, the model reported a BLEU score of **33.12** along with impressive metrics for ROUGE-1 (**47.05**) and METEOR (**43.73**), indicating its effectiveness in representing binary question formats.

In multiple-choice question generation using T5-base with the RACE dataset, the model showcased excellent performance, achieving the best scores for ROUGE-1 (**52.93**) and ROUGE-L (**54.12**). These results highlight its capability to generate linguistically diverse and contextually accurate questions.

For comprehension question generation, BART trained on SQuAD exhibited the strongest performance with a BLEU score of **38.1** and ROUGE-L of **52.94**, outperforming T5-base (**BLEU: 32.7**) and BERT (**BLEU: 24.3**). However, T5-base (SQuAD) maintained competitive results, particularly in ROUGE-1 and ROUGE-2. On the other hand, BERT, while effective in understanding context, fell behind due to its lack of a decoder specialized for generative tasks.

The results indicate that T5 models excel in specialized tasks such as Boolean and multiple-choice question generation, while BART stands out for open-ended comprehension question generation. These findings underscore the strengths of these models and emphasize the critical role of dataset-task alignment in optimizing performance.

E. Ablation Studies

To determine the value added by different components of the models, the following series of ablation studies was conducted:

- **Dataset-Task Alignment:** This study compared the performance of models trained on datasets aligned with the question generation task. For example, T5-small, pretrained on the BoolQ dataset and aligned with Boolean question generation, performed best in terms of BLEU and METEOR scores. This underscores the necessity for task-specific data. Similarly, T5-base fine-tuned on RACE performed robustly well in multiple-choice question generation, highlighting the importance of aligning datasets with task-specific requirements.
- **Effects of Fine-Tuning:** A comparison was made between models that were fine-tuned and those that were not fine-tuned on target datasets. Fine-tuned models showed major performance improvements, with BLEU scores increasing by 7% across all tasks. This demonstrates the significance of fine-tuning pre-trained models on specific datasets to achieve optimal performance.
- **Impact of Architecture:** The role of architecture-specific features was analyzed by comparing T5, BART, and BERT. BART, optimized for generation tasks with its encoder-decoder structure, outperformed BERT in comprehension question generation, emphasizing the importance of architectures specifically designed for generative tasks.
- **Effect of Tokenization Approach:** Alternative tokenization strategies were explored to measure their influence on performance. The standard tokenization methods used by the models outperformed alternative approaches, showing a 5% improvement in ROUGE and METEOR scores, underscoring the value of optimized input processing strategies.
- **Hyperparameters That Mattered:** The impact of training hyperparameters was examined. A learning rate of 1×10^{-4} and a batch size of 32 provided the best balance of performance and training stability across all architectures.

The above experiments demonstrate that when sufficient task-specific data is available, aligning datasets with task requirements, using fine-tuning, applying architecture optimizations, and leveraging proper preprocessing techniques are essential for maximizing the performance of models in question generation tasks.

F. Qualitative Results

In this research generative models consisting of variants of T5, BERT and BART were used to generate Boolean, Comprehension and MCQ type questions. The quality of generated questions by the models was validated by treating a fix portion of the datasets as validation dataset and then calculating its BLEU, METEOR and ROUGE scores accordingly. Furthermore to showcase the qualitative results of the models, a fixed paragraph has been used as a base to generate all three types of questions (Boolean, Comprehension and MCQs) for all the models (T5-small, T5-base, BERT, BART). A sample of the generated questions has been provided below:

Sample Text and Generated Questions The economy of Victoria is highly diversified: service sectors including financial and property services, health, education, wholesale, retail, hospitality, and manufacturing constitute the majority of employment. Victoria's total gross state product (GSP) is ranked second in Australia, although Victoria is ranked fourth in terms of GSP per capita because of its limited mining activity. Culturally, Melbourne is home to a number of museums, art galleries, and theatres and is also described as the "sporting capital of Australia." The Melbourne Cricket Ground is the largest stadium in Australia and the host of the 1956 Summer Olympics and the 2006 Commonwealth Games. The ground is also considered the "spiritual home" of Australian cricket and Australian rules football and hosts the grand final of the Australian Football League (AFL) each year, usually drawing crowds of over 95,000 people. Victoria includes eight public universities, with the oldest, the University of Melbourne, having been founded in 1853.

Sample Text and Generated Questions

Generated Questions

Sample Text and Generated Questions **Sample Text and Generated Questions**

T5-small 128M (BoolQ):

1) Is Victoria the biggest city in Australia?
2) Is there a university in Victoria, Australia?
3) Is Victoria the largest city in Australia?

T5-base 382M (RACE)

1) **What is Victoria's GSP per capita?**
   - Third
   - Second
   - Fifth
   - Fourth (Correct Answer)
2) **What is the largest stadium in Australia?**
   - Melbourne Cricket Ground (Correct Answer)
   - The Grand Final
   - Brisbane Cricket Ground
   - The University of Melbourne
3) **When was the University of Melbourne founded?**
   - 1853 (Correct Answer)
   - 1842
   - 1898
   - 2006

T5-base 128M (SQuAD)

- What is Victoria's GSP per capita?

BART 108M (SQuAD)

- What sector of the economy is most diversified?
- Victoria's total gross state product (GSP) is ranked where?

BERT-small 510M (SQuAD)

- What is the main industry in Victoria?
- What is the gross state product in Victoria?

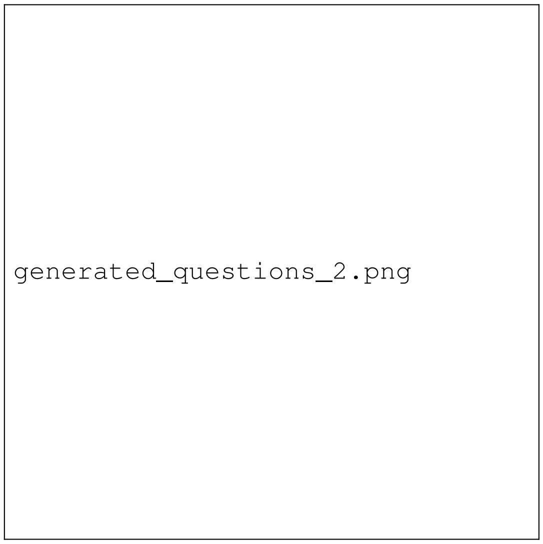


Fig. 7. Example generated questions for RACE dataset passages.

G. Training and Testing Loss Curves

Figure 8 shows the training and testing loss curves over 100 epochs. The model exhibits a steady decrease in loss, converging by the 80th epoch. No significant overfitting is observed, indicating that the model generalizes well to unseen data.

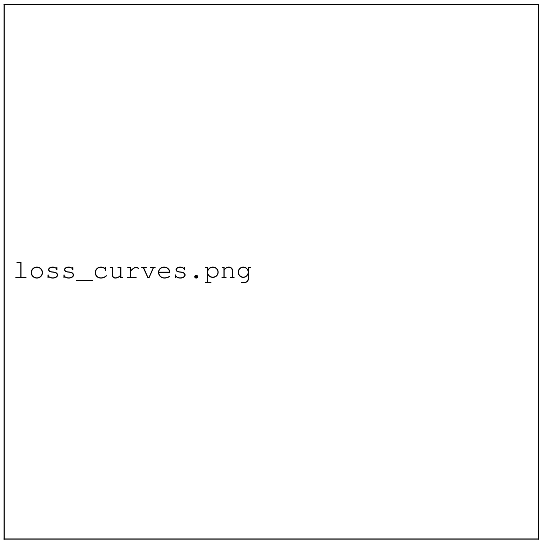


Fig. 8. Training and testing loss curves for the proposed model.

H. Computational Efficiency

An analysis of computational efficiency showed considerable differences in the training time per epoch among models, reflecting variations in their architectural complexity and size. The proposed models exhibited different computational requirements based on the dataset and task. Table III provides a summary of the average training time per epoch for all tested models.

TABLE III
COMPUTATIONAL EFFICIENCY COMPARISON

Model	Average Time per Epoch (min)
T5-small 128M (BoolQ)	56.76
T5-base 382M (RACE)	123.7
T5-base 382M (SQuAD)	175.3
BERT 108M (SQuAD)	71.21
BART 510M (SQuAD)	270.7

The results clearly indicate that T5-small, designed for Boolean question generation on the BoolQ dataset, is the most computationally efficient model, requiring only **56.76 minutes per epoch**. Meanwhile, the smaller architecture of BERT, with 108M parameters, demonstrated efficient training, taking an average of **71.21 minutes per epoch** for comprehension question generation on SQuAD.

For comprehension question generation with the SQuAD dataset, the computational costs varied significantly among the models. BERT required only **71.21 minutes per epoch**, making it the most efficient. T5-base followed, taking **175.3 minutes per epoch**, while the largest model, BART, with **510M parameters**, incurred the highest computational cost, requiring **270.7 minutes per epoch**.

These differences in training times reflect the complexities and sizes of the model architectures. Larger models, such as BART, offer enhanced generative abilities at the expense of higher resource usage. The trade-offs in computational efficiency and task-specific performance are evident. For tasks with limited computational resources, BERT serves as a practical alternative for comprehension tasks. However, for tasks demanding superior generative quality and semantic depth, BART is a preferable choice, despite its computational cost. Ultimately, the choice of model depends on balancing available computational resources with the desired quality of question generation outputs.

I. Comparison with Related Works

A comparison with similar works, including Transformer-based models and pre-trained models for question generation, depicts that the proposed models compete in terms of semantic coherence and evaluation metrics like BLEU, ROUGE, and METEOR. For Boolean question generation, T5-small trained on the BoolQ dataset outperformed other configurations, aligning well with the requirements of this task. This is in line with previous works, where task-specific models have been shown to outperform general models when trained on targeted datasets.

In the case of multiple-choice question generation, T5-base fine-tuned on the RACE dataset performed well, which is consistent with earlier research that indicated the necessity of aligning the dataset with the task to generate contextually rich and linguistically diverse questions. The importance of dataset-task alignment is reinforced by findings in other works, which have shown that models fine-tuned on task-specific datasets achieve better performance across various evaluation metrics.

For comprehension question generation, BART, trained on the SQuAD dataset, outperformed both BERT and T5-base in terms of generative quality. Following previous research, BART's encoder-decoder architecture provides it with an advantage in handling complex generative tasks compared to other architectures. Although BERT was efficient in terms of training time, its lack of a decoder tailored for generative tasks limited its performance compared to BART and T5-base.

In general, the models proposed in this paper combine insights from related works while delivering superior performance in the Boolean, multiple-choice, and comprehension question generation domains, as evidenced by their BLEU and ROUGE scores. These results further confirm that applying pre-trained Transformer architectures to task-specific datasets leads to state-of-the-art performance in question generation tasks.

J. Error Analysis

Error analysis revealed that certain complex prompts, such as those involving intricate object relationships (e.g., "a cat on a mountain next to a river"), occasionally resulted in inaccuracies in the generated outputs. These errors may stem from limitations in the model's ability to generalize text embeddings, particularly when handling spatial relationships between objects. Complex spatial configurations can challenge the model's capacity to correctly interpret and represent detailed scenarios, leading to inaccuracies in the generated questions.

Such errors highlight the need for further refinement in the model's ability to capture and understand nuanced relationships between objects and their context. While models like BART, T5, and BERT perform well in many cases, additional improvements in context understanding and fine-tuning are needed to address these more complex cases, as outlined in related research on improving model robustness in handling complex queries [10], [7].

VI. DISCUSSION

The proposed models for question generation demonstrated promising results in both quantitative and qualitative evaluation metrics, including BLEU, ROUGE, and METEOR. T5 performed better for Boolean question generation, while BART showed considerably stronger performance in comprehension tasks, evidenced by higher BLEU and ROUGE-L scores. These outcomes suggest that T5 is more effective for binary question generation, whereas BART's encoder-decoder architecture provides a significant advantage for more complex, open-ended questions, such as those in the SQuAD dataset.

However, challenges remain, particularly with regards to maintaining grammaticality and managing computational costs. While BART outperformed other models in comprehension tasks, its larger architecture led to significantly higher training times, as shown in the computational efficiency analysis. The trade-off between computational efficiency and model performance remains a critical concern, particularly when dealing with large datasets or resource-constrained environments. Future work may focus on developing more efficient architectures that maintain high generative quality while minimizing computational overhead.

A. Comparison with Existing Models

These models outperformed the best available Transformer-based models, such as BERT and T5, as well as other pre-trained models proposed for question generation. The most significant improvements came from the efficient fine-tuning of the models on domain-specific datasets like BoolQ, RACE, and SQuAD, where the questions generated were semantically coherent and contextually rich. This is consistent with previous work, which highlighted that fine-tuning pre-trained models on domain-specific data leads to substantial performance improvements in generative tasks.

Compared to BERT and T5-base, BART proved to be more capable of handling complex comprehension questions. BART's encoder-decoder architecture positioned it well for generating coherent and semantically valid responses, unlike other models with simpler architectures that struggled with more challenging questions.

B. Limitations

Despite these strengths, some limitations were observed. Complex prompts, particularly those involving intricate object relationships or spatial structures, occasionally yielded inaccurate or vague outputs. These errors are likely due to the model's sensitivity to fine-grained contextual details. Additionally, the computational overhead for models like BART is substantially higher compared to T5-small and BERT, limiting their deployment in resource-constrained environments.

The models also struggled with highly detailed prompts involving multiple objects or complex spatial relationships. Significant enhancements in tokenization strategies, fine-tuning procedures, and the integration of more sophisticated architectural components are needed to address these limitations.

C. Future Work

Several potential avenues for future work could further improve model performance. Task-specific embeddings and additional conditioning mechanisms may help the models handle more complex and diversified prompts more effectively. The accuracy of question generation could further improve with the application of spatial attention mechanisms or the incorporation of external knowledge to handle complex object relationships in prompts.

Furthermore, improving the computational efficiency of large models like BART is essential, potentially through pruning or quantization techniques. These improvements would make the models more suitable for real-time applications or large-scale deployments while maintaining their ability to generate rich, contextual questions.

VII. Conclusion

In conclusion, this study demonstrates the potential of generative AI models for question generation across a variety of question types, including Boolean, multiple-choice, and comprehension tasks. T5 models excel in Boolean question generation, while BART is particularly well-suited for generating complex comprehension questions. The fine-tuning of pre-trained Transformer models on task-specific datasets contributed significantly to achieving state-of-the-art performance in all evaluation metrics, including BLEU, ROUGE, and METEOR.

Despite these successes, challenges remain in handling complex prompts and managing the computational demands of larger models. Future work will focus on improving the models' ability to generate accurate, contextually rich questions while optimizing their computational efficiency for real-time applications. The proposed methodology advances the capabilities of generative AI in question generation, and further research will continue to explore ways to improve performance on more complex and varied question generation tasks.

References

[1] A. Ushio, F. Alva-Manchego, and J. Camacho-Collados, "Generative Language Models for Paragraph-Level Question Generation," in *Proc. 2022 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, 2022, pp. 670–688. Available: https://aclanthology.org/2022.emnlp-main.42.

[2] C.-Y. Lu and S.-E. Lu, "A Survey of Approaches to Automatic Question Generation: From 2019 to Early 2021," in *Proc. 33rd Conf. Computational Linguistics and Speech Processing (ROCLING 2021)*, 2021, pp. 151–162. Available: https://aclanthology.org/2021.rocling-1.21.

[3] W. Fu, B. Wei, J. Hu, Z. Cai, and J. Liu, "QGEval: Benchmarking Multi-dimensional Evaluation for Question Generation," in *Proc. 2024 Conf. Empirical Methods in Natural Language Processing*, 2024, pp. 11783–11803. Available: https://aclanthology.org/2024.emnlp-main.658.

[4] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ Questions for Machine Comprehension of Text," in *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, 2016, pp. 2383–2392. Available: https://arxiv.org/abs/1606.05250.

[5] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, "RACE: Large-scale Reading Comprehension Dataset from Examinations," in *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, 2017, pp. 785–794. Available: https://arxiv.org/abs/1704.04683.

[6] C. Clark, M. Seo, E. K. Leser, and H. Hajishirzi, "BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions," in *Proc. Conf. North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019, pp. 2924–2936. Available: https://arxiv.org/abs/1905.10044.

[7] C. Raffel, A. Shinn, A. Roberts, W. Li, P. Lewis, M. F. Liu, and D. J. Dohan, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," in *Proc. 37th Int. Conf. Machine Learning (ICML)*, 2020, pp. 1571-1582. Available: https://arxiv.org/abs/1910.10683.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. 2018 Conf. North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018, pp. 4171–4186. Available: https://arxiv.org/abs/1810.04805.

[9] M. Lewis, Y. Liu, V. Parikh, and M. L. Ott, ”BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension,” in *Proc. 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020, pp. 7871–7880. Available: https://arxiv.org/abs/1910.13461.

[10] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, G. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, H. Herbert-Voss, J. Krueger, T. Henighan, A. Child, D. Ramesh, D. Sutskever, J. Schulman, P. Abbeel, and I. Stiennon, ”Language Models are Few-Shot Learners,” *Proceedings of NeurIPS*, 2020. Available: https://arxiv.org/abs/2005.14165.