Form 4: Results and conclusion

1. Team No: 10

2. Project Title: SmartQuiz: AI MCQ Generation

3. Experiment Environment:

3.1 Execution Environment:

Google Colab: For fine tuning T5 for question generation.

Anaconda: For deploying the Mcq generation

3.2 Dataset Description:

a) **SQuAD** (Stanford Question Answering Dataset):

SQuAD is a widely-used dataset for question answering tasks, containing questions posed by crowdworkers on a set of Wikipedia articles, with corresponding answer spans within the text. It serves as a benchmark for evaluating models' ability to comprehend and answer questions based on textual passages.

b) **EduQG** (Educational Question Generation Dataset):

EduQG is a dataset specifically designed for educational question generation tasks, comprising questions generated by experts covering various academic subjects and levels of difficulty. It provides a diverse range of question types and topics, making it suitable for training models to generate educational questions.

c) **Custom C Language Dataset**:

The custom C language dataset consists of multiple-choice questions (MCQs) collected from online sources, focusing specifically on concepts related to the C programming language. These MCQs cover fundamental topics such as variables, control structures, functions, and data types in C programming. The dataset serves as a targeted resource for training models to generate MCQs tailored to the C programming domain.

3.3 Parameter Formulas

a) **F1 Score:** F1 score can be calculated based on the presence of generated questions in the set of reference questions, considering precision and recall.

F1 = 2 * (Precision * Recall) / (Precision + Recall)

Precision: Precision = (Number of Generated Questions in Reference) /

(Total Number of Generated Questions)

Recall: Recall = (Number of Generated Questions in Reference) / (Total Number of Reference Questions)

b) **Exact Match**: Exact Match evaluates whether the generated question exactly matches any of the reference questions:

Exact Match = (Number of Exact Matches) /

(Total Number of Generated Questions)

c) **METEOR (Metric for Evaluation of Translation with Explicit ORdering):**

METEOR considers various linguistic aspects such as fluency, precision, and recall:

METEOR = (10 * P * R) / ((1 - alpha) * R + alpha * P) * (1 - beta * e^(-L / L0))

- Where:

- P and R are precision and recall, respectively.

- alpha and beta are tunable parameters.

- L is the alignment score.

- L0 is the length normalization factor.

d) **ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation - Longest Common Subsequence):**

ROUGE-L measures the similarity between the generated question and the reference question based on the longest common subsequence:

ROUGE-L = LCS(G, R) / max(len(G), len(R))

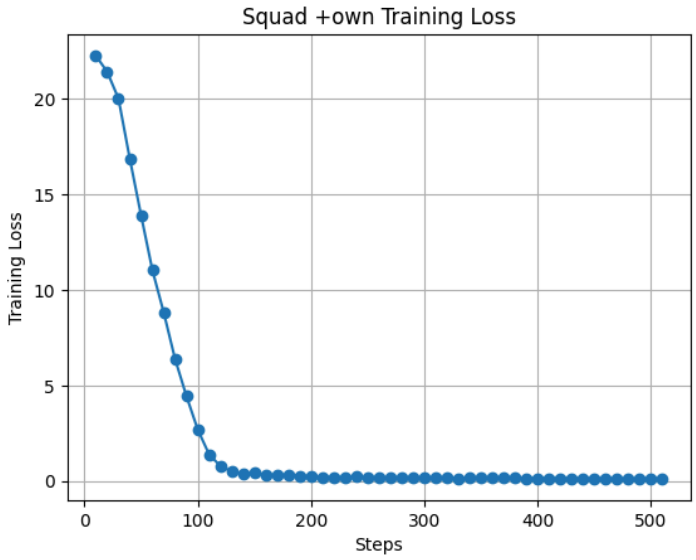
Where: - LCS(G, R) is the length of the longest common subsequence between the generated question G and the reference question R.

- len(G) and len(R) are the lengths of the generated and reference questions, respectively.

4. A Experiment 1:

Findings:

The training loss graph for the T5 model displays a typical learning curve, indicating effective learning from the SQuAD and C language question datasets. While the sharp decline in loss values reflects initial learning, the plateau phase suggests convergence to optimal parameters, highlighting the model's efficiency. However, it's crucial to monitor for overfitting during this phase to maintain generalization ability. In experiment 1 (SQuAD + C dataset), the model demonstrates commendable abilities in generating general questions, but the training loss curve hints at potential challenges in capturing the nuances of C programming. Despite this, the model excels in producing questions that align perfectly with human-written ones, particularly within the context of C programming. This suggests a strong proficiency in specific domains despite potential limitations in broader question generation.

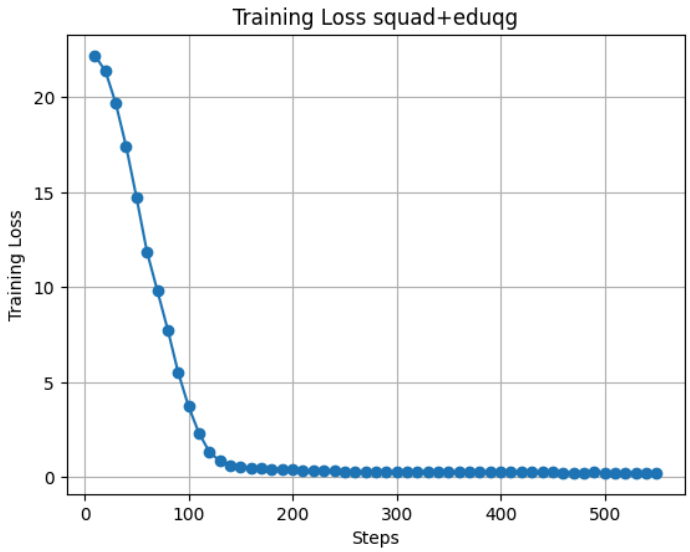


Training loss values graph fine tuning t5 on SQUAD and C language datasets

4. B Experiment 2:

Findings:

The model's proficiency in general question generation, as evidenced by a potentially lower final loss value in the training phase. However, there are indications of limitations in generating C-specific questions, as suggested by potentially lower Exact Match, F1 scores, especially for less complex questions. While the model excels in generating questions across various topics, it may require further refinement to cater specifically to C programming.



Training loss values graph fine tuning t5 on SQUAD and EduQG datasets

4. C Experiment 3:

Findings:

This graph represents a well-balanced approach, showcasing strong capabilities in both general question generation and C programming-specific question generation. The training loss curve demonstrates effective learning across diverse topics, including C programming. Additionally, the model maintains robust Exact Match, F1 scores, potentially comparable to or slightly lower than those in experiment 1, indicating a high level of proficiency in generating questions that precisely match human-written ones. Overall, experiment 3 emerges as the optimal choice, striking a fine balance between general question generation and competency in C

programming question generation.	**Training loss values graph fine tuning t5 on SQUAD , C language and EduQG datasets**

5. Parameter comparison table

Parameter	Previous methods	Proposed method
Dataset(s) Used	SQuAD and EduQG	SQuAD + EduQG + C Dataset
Summarization	T5 Transformer	Bertsum
Evaluation Metrics	AVG F1 Score: 53.89 Meteor Score: 0.5094 Average Exact Match: 33.23 Rouge-lsum: 51.8	AVG F1 Score: 69.18 Meteor Score: 0.64 Average Exact Match: 11.29 Rouge-lsum : 66.3
Performance	Potentially good performance on general question generation for Squad or educational questions for EduQG	Potentially improved performance on both general question generation and C-specific question generation.
Distractor Generation	Conceptnet , Race	Sense2vec, Gemini

6. Final Conclusion Statements

Our analysis indicates that training a T5 question generation model with a combination of SQuAD, EduQG, and a C programming-specific dataset yields the most promising results. This approach demonstrates a good balance between general question generation capabilities and improved accuracy in generating C programming-related questions. The training loss curves and evaluation metrics, particularly the Exact Match, F1 scores, support this conclusion. Our custom C dataset appears to play a key role in boosting the model's performance on C-specific questions.

Based on the test metrics we provided for the test dataset, the chosen approach appears promising. Overall, the findings based on the result test metrics suggest that this combined dataset strategy has the potential to achieve both general question generation proficiency and competency in C programming question generation.

Signature Supervisor
Name: P Chakradhar