Question Generation Dataset: EDA & Preprocessing Report

Prepared by: Purushottam Kumar
Date: November 14, 2025

Abstract

This report documents exploratory data analysis (EDA) preprocessing performed on an education-oriented question–answer dataset used for training a T5-based Question Generation (QG) model. The documented steps include dataset inspection, cleaning, token-length analysis using a T5 tokenizer, control token addition, deduplication, token-aware truncation strategy, and creation of train/validation/test splits for three downstream tasks: `context_question`, `context_answer`, and `contex_joint_ques_ans`.

Contents

1 Source Code Link

https://github.com/aathanush/CS787-Project-Question-Generator

2 Introduction

In the Educational technology (EdTech) domain, the ability to automatically generate assessment materials is a crucial task. Educators often spend significant time manually crafting questions and answers to evaluate student understanding. This project addresses this problem by developing an automated pipeline capable of generating context-aware questions and corresponding answers based on educational text, while also providing students with a tool that can generate questions and answers for them to practice.

Using a dataset derived from NCERT textbooks, specifically for science and math-based subjects from grades 10 to 12, we aim to train generative models that take a specific context (explanation), grade level, difficulty, and subject as input to generate relevant questions. Furthermore, we implement a separate system to generate accurate answers for those questions.

We trained several models on the dataset including BERT-SQG, T5-small, T5-base, and BART-base. Our results show that T5-base outperforms BERT-SQG in terms of BLEU score and BERTScore F1.

The remainder of this report is organised as follows: Section provides a review of relevant literature. The link to the GitHub repository of the code is provided in Section 3. Section 4 outlines the details of the dataset. Section 5 provides details regarding the Exploratory Data Analysis carried out in the dataset. Section 6 presents the experimental results and analysis, and Section 7 concludes the report with directions for future work.

3 Literature Review

4 Dataset description

The raw CSV file has the following columns:

```
Topic, Explanation, Question, Answer, Difficulty, StudentLevel,
QuestionType, QuestionComplexity, Prerequisites, EstimatedTime, subject, grade
```

Total rows loaded: **30,706**.

5 Exploratory Data Analysis (EDA)

5.1 Missing values

All columns in the dataset have complete entries (no nulls in the original CSV reading step):

```
Topic                 0
Explanation           0
Question              0
Answer                0
Difficulty            0
StudentLevel          0
QuestionType          0
QuestionComplexity    0
Prerequisites         0
EstimatedTime         0
```

```
subject               0
grade                 0
```

5.2 Text length statistics (words)

Word-length statistics for the main text fields (computed as word counts):

	count	mean	std	min	50%	max
len_Explanation	30706	77.93	23.01	1	74	308
len_Question	30706	14.08	5.21	1	14	81
len_Answer	30706	32.18	19.56	1	32	177

5.3 Duplicates

Initial duplicate counts discovered during EDA:

- Duplicate **questions**: 2,429
- Duplicate **answers**: 1,677
- Duplicate **question+answer pairs**: 1,008

5.4 Categorical distributions

Subject counts (raw):

- Physics: 10,505
- Chemistry: 9,911
- Biology: 6,894
- Science: 3,396

Grade distribution (raw):

- Grade 11: 14,533
- Grade 12: 12,777
- Grade 10: 3,396

Difficulty / StudentLevel :

- Difficulty: Medium (10,267), Hard (10,242), Easy (10,193), plus a few malformed rows.
- StudentLevel: Intermediate (10,267), Advanced (10,242), Beginner (10,193), plus a few malformed rows.

5.5 Token-length analysis

We tokenized **Explanation** texts with `t5-base` tokenizer (fast tokenizer) to get realistic token lengths for transformer inputs. Summary of `expl_tok_len` (token counts for Explanation):

```
count    30706.000000
mean       117.108904
std         40.394809
min          4.000000
25%         90.000000
50%        109.000000
75%        134.000000
90%        167.000000
95%        194.000000
98%        232.000000
99%        258.950000
max        502.000000
```

Coverage at candidate truncation cutoffs:

- 256 tokens covers **98.95%** of Explanations
- 320 tokens covers **99.79%**
- 384 tokens covers **99.97%**
- 512 tokens covers **100%**

From these numbers we selected a conservative policy (see below).

6 Analysis and Results

This section details the quantitative results of our experiments. We evaluated three model architectures (T5, BART, and BERT-SQG) on our two primary tasks: Question Generation (QG) and Answer Generation (AG). Furthermore, we analyze the impact of iterative prompt engineering on model performance.

6.1 Prompt Engineering

The design of the input prompt is critical for guiding a model's generative process. We experimented with two distinct prompt structures.

6.1.1 Prompt 1 (P1)

Our initial prompt was highly detailed, explicitly including all features from the dataset, such as `QuestionComplexity`.

- **QG Prompt 1:** `Generate a {Difficulty} question with complexity {Score} for a grade {Grade} {Subject} student. Explanation: {Context}`

6.1.2 Prompt 2 (P2)

Based on initial observations, we hypothesized that the `QuestionComplexity` score was redundant and potentially confusing to the model. We revised the prompt to be more direct and natural.

- **QG Prompt 2:** `Generate a {Difficulty} question for a grade {Grade} {Subject} student using this context: {Context}`
- **AG Prompt:** The Answer Generation prompt remained consistent for both experiments: `Answer the following {Difficulty} grade {Grade} {Subject} question. Explanation: {Context} Question: {Question}`

6.2 Experimental Results

We evaluated all generated outputs against their reference counterparts using two standard metrics: BLEU-4 and BERTScore F1.

NOTE: BERT-SQG baseline model was trained on a significantly reduced dataset of **5,000 samples**. This was due to the model's architectural inefficiency, which resulted in a ∼15x data explosion during preprocessing, making training on the full 30,000-sample dataset computationally infeasible within our time constraints. All T5 and BART models were trained on the full 30,000-sample dataset.

6.2.1 Performance with Prompt 1

The initial results from our first prompt set the baseline for our models.

Table 2: Question Generation (QG) Performance using Prompt 1

Model	BLEU-4 Score	BERTScore F1
T5-Small	19.01	0.9082
T5-Base	25.75	0.9201
BART-Base	23.26	0.9126
BERT-SQG	16.92	0.8750

Table 3: Answer Generation (AG) Performance using Prompt 1

Model	BLEU-4 Score	BERTScore F1
T5-Small	9.88	0.8737
T5-Base	13.35	0.8776
BART-Base	11.79	0.8743

6.2.2 Performance with Prompt 2 (Improved)

After modifying the QG prompt, we re-trained and evaluated the T5-Base and BART-Base models.

Table 4: Question Generation (QG) Performance using Prompt 2

Model	BLEU-4 Score	BERTScore F1
T5-Small	22.68	0.9135
T5-Base	**28.06**	0.9202
BART-Base	25.79	**0.9193**
BERT-SQG	-	-

6.3 Interpretation of Results

Our results lead to the following conclusions:

1. **Effectiveness of Prompt Engineering:** Comparing Table 2 and Table 4, the impact of our prompt refinement is clear. The T5-Base model's BLEU-4 score for QG jumped from 25.75 to 28.06, and BART's score increased from 23.26 to 25.79. This confirms our

Table 5: Answer Generation (AG) Performance using Prompt 2

Model	BLEU-4 Score	BERTScore F1
T5-Small	13.35	0.8776
T5-Base	**15.88**	**0.8800**
BART-Base	12.71	0.8796

hypothesis that the simpler, more natural language prompt (P2) allowed the models to generate more accurate questions.

2. **T5-Base provided optimal performance:** Across all experiments, the **T5-Base** model provided better performance. For Question Generation, it achieved the highest BLEU score (28.06), and for Answer Generation, it also achieved the highest BLEU (15.88) and BERTScore (0.8800). This suggests its pre-training objective is exceptionally well-suited for text-to-text generation tasks.

3. **Semantic vs. Lexical Scores:** For QG (Table 4), the BERTScores for T5-Base (0.9201, P1) and BART (0.9193, P2) are extremely high and very close. This indicates that both models are generating questions that are semantically **identical** to the ground truth, even if they use different wording. The T5-Base's higher BLEU score suggests it is slightly better at matching the exact phrasing of the reference questions.

4. **Lower Scores for Answer Generation:** The BLEU scores for Answer Generation (Tables 3 and 5) are consistently lower than their QG counterparts. This is expected, as an answer can be phrased correctly in many ways, making an exact lexical match (which BLEU measures) difficult. The high BERTScores (all ∼0.87-0.88) are more informative here, through which we can infer that the models are generating semantically correct answers.

5. **BERT-SQG Baseline:** The BERT-SQG model was used as a baseline, showing the lowest performance (BLEU 16.92). This is attributable to both its inefficient architecture for generation and its limited training dataset (5,000 samples), which validates our decision to focus on T5 and BART as the primary models for this project.

7 Concluding remarks