Report: Building a State-of-the-Art Question Answering System: A Case Study Using Quora Dataset

Author: Aishwarya Shrigiri

GitHub Repo: https://github.com/Aishuwu/Quora-Question-Answer-Case-Study

Notebook: Quora_QA_Case_Study.ipynb

Tools Used: Python, Google Colab, HuggingFace Transformers, Datasets, Matplotlib, Seaborn

1. Introduction

Answering questions in a way that feels natural and helpful is one of the most important goals in artificial intelligence today. In this case study, I've worked on building a question-answering system using the Quora Question Answer Dataset. The goal is to create a model that understands questions the way humans do and can respond with accurate and relevant answers

To do this, I explored powerful NLP models like T5 and GPT. Each of them has its strengths and limitations, and the idea was to test how well they perform on this task, using metrics like ROUGE, BLEU, and F1-score to compare them

This kind of system has real-world use cases, especially in industries like aviation where a fast and accurate response can make a big difference in user experience. Whether it's answering customer queries or assisting support teams, a good QA system can reduce workload and improve satisfaction. This report walks through my approach, findings, and what I'd recommend going forward

2. Literature Survey

Transformer-based models like T5 (Text-To-Text Transfer Transformer) and GPT-2 (Generative Pretrained Transformer 2) have achieved significant success in NLP tasks. T5 treats every NLP task as a text-to-text problem and is well-suited for supervised learning settings. GPT-2 on the other hand, is an autoregressive language model known for its fluency in generating text in an unsupervised way.

Both models have been successfully used in question answering tasks in various benchmarks like SQuAD, Natural Questions and QuAC. This study explores their effectiveness on Quora's open domain QA data

3. Methodology

To build and compare different question-answering models, I followed a structured pipeline that included data exploration, preprocessing, model training, and evaluation. Here's a breakdown of the steps I took and the reasoning behind each decision

3.1 Understanding and Preparing the Data

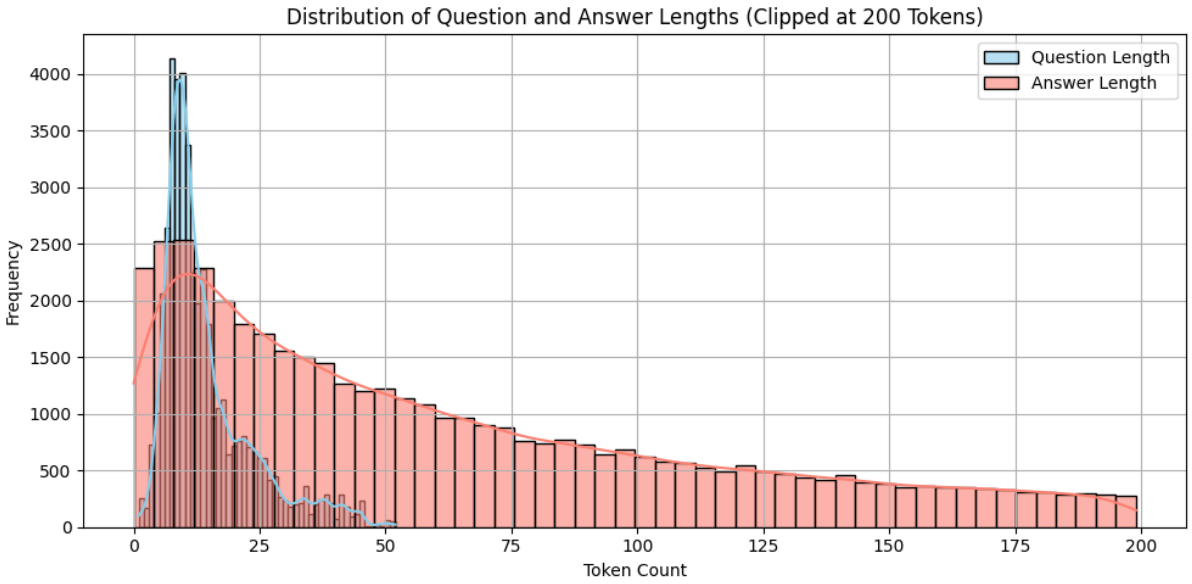
I worked with the Quora Question Answer Dataset which contains over 56,000 question-answer pairs. After loading and inspecting the dataset, I found that questions were generally concise (most had under 20 words), while answers varied widely in length, some going beyond 500 words

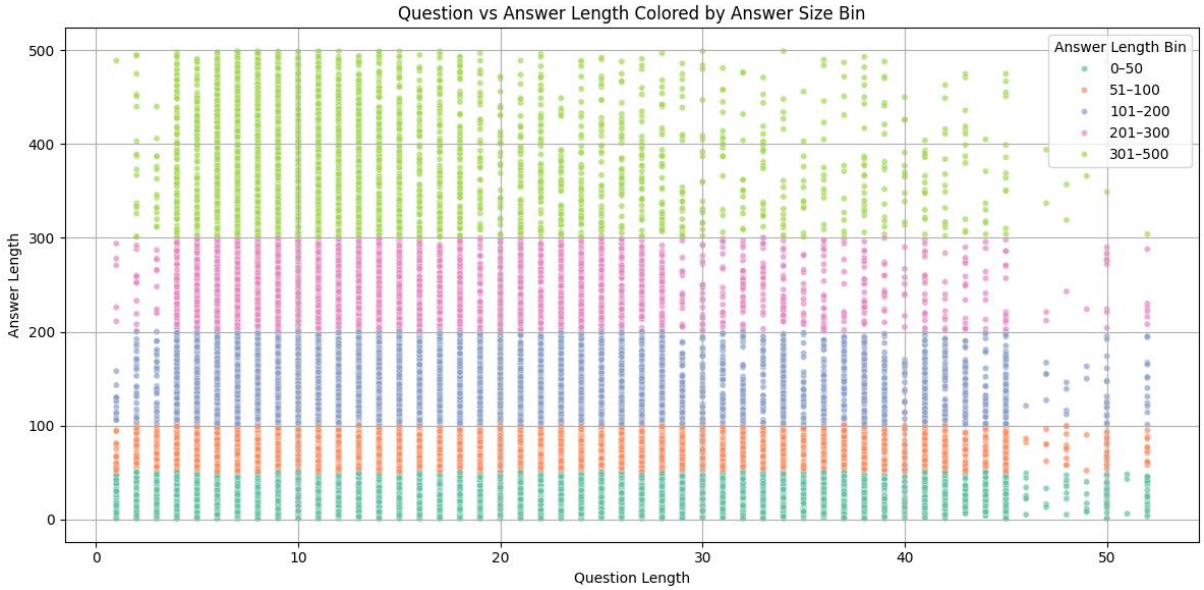
To make training manageable and avoid issues with token length limits in transformer models, I filtered out extremely long entries. This brought the final dataset to around 43,000 cleaned records

Visualizations confirmed the skewed distribution:

- Most questions were under 15 tokens
- Answers were longer, with a long tail of very detailed responses

These plots helped shape preprocessing strategies and guided how much content to feed into each model without losing quality





3.2 Data Cleaning and Preprocessing

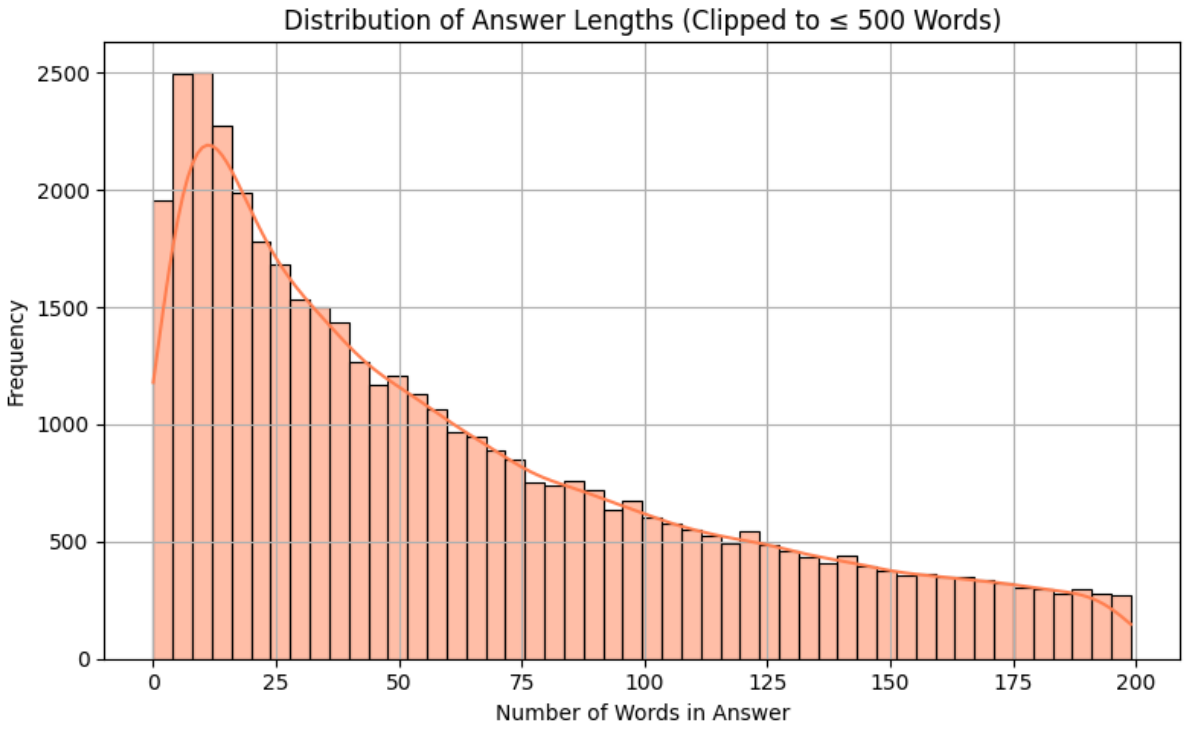
I removed noisy entries such as blank values, duplicate records, and very long texts. To ensure consistency:

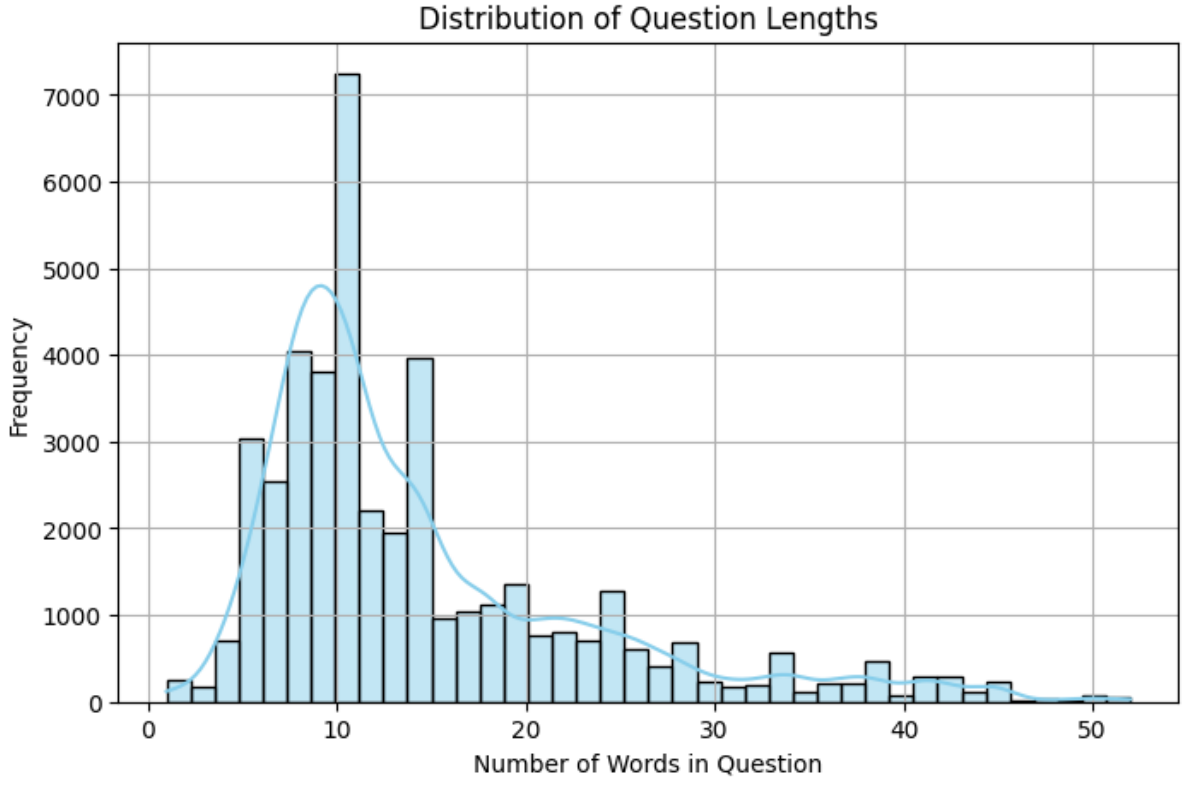
I cleaned the text by removing HTML tags, newline characters, and extra spaces

For T5, I reformatted each record into a plain format like question: <text> as input and the answer as target output

For GPT-2, I used a conversational format

I also prepared the datasets as Hugging Face Dataset objects and performed tokenization using the respective model's tokenizer. Inputs were padded and truncated to fit within the model's maximum token limit





3.3 Model Implementation and Training

I trained two models separately for comparison:

T5-Small

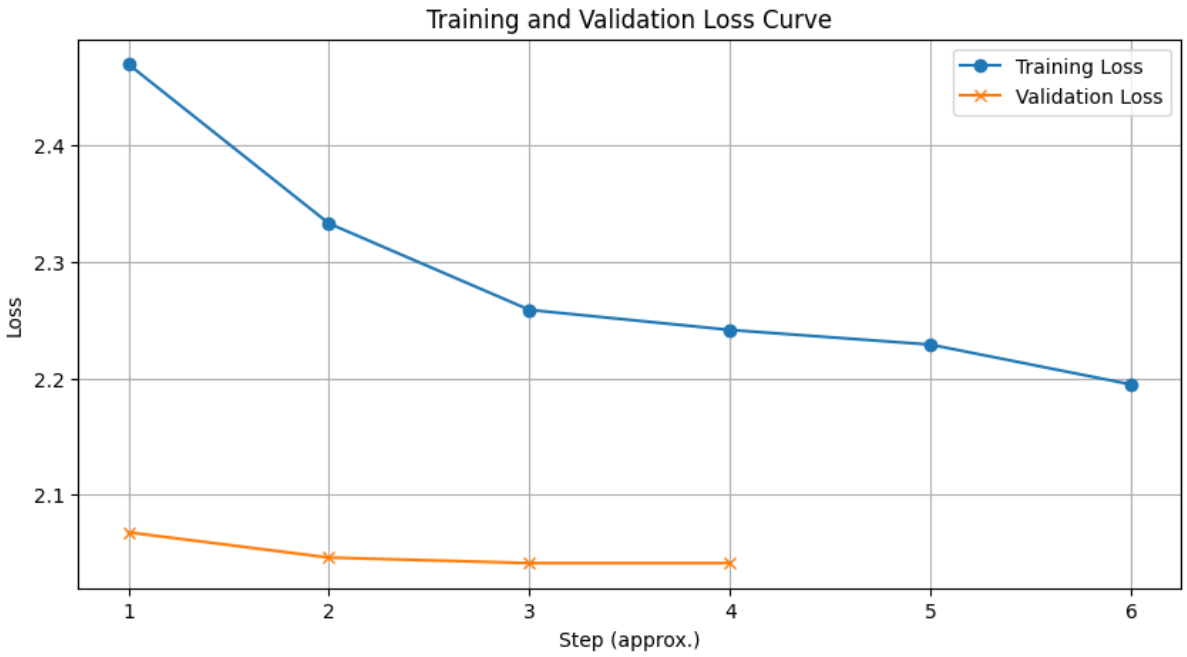
Used Hugging Face's Trainer API with t5-small

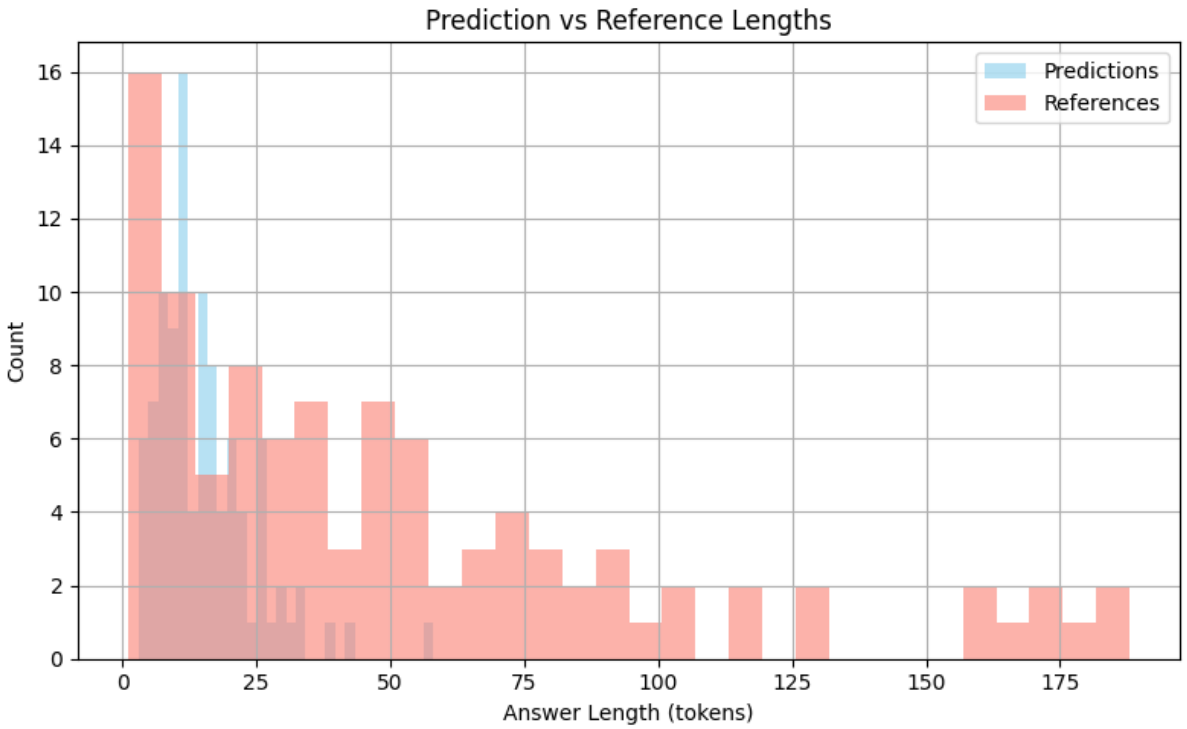
Trained on 10,000 samples with a batch size of 2, for 3 epoch

Used learning rate scheduling and gradient accumulation for better optimization

Both training and validation losses decreased steadily across epochs, which indicated stable learning

The model was saved and evaluated using standard generation metrics





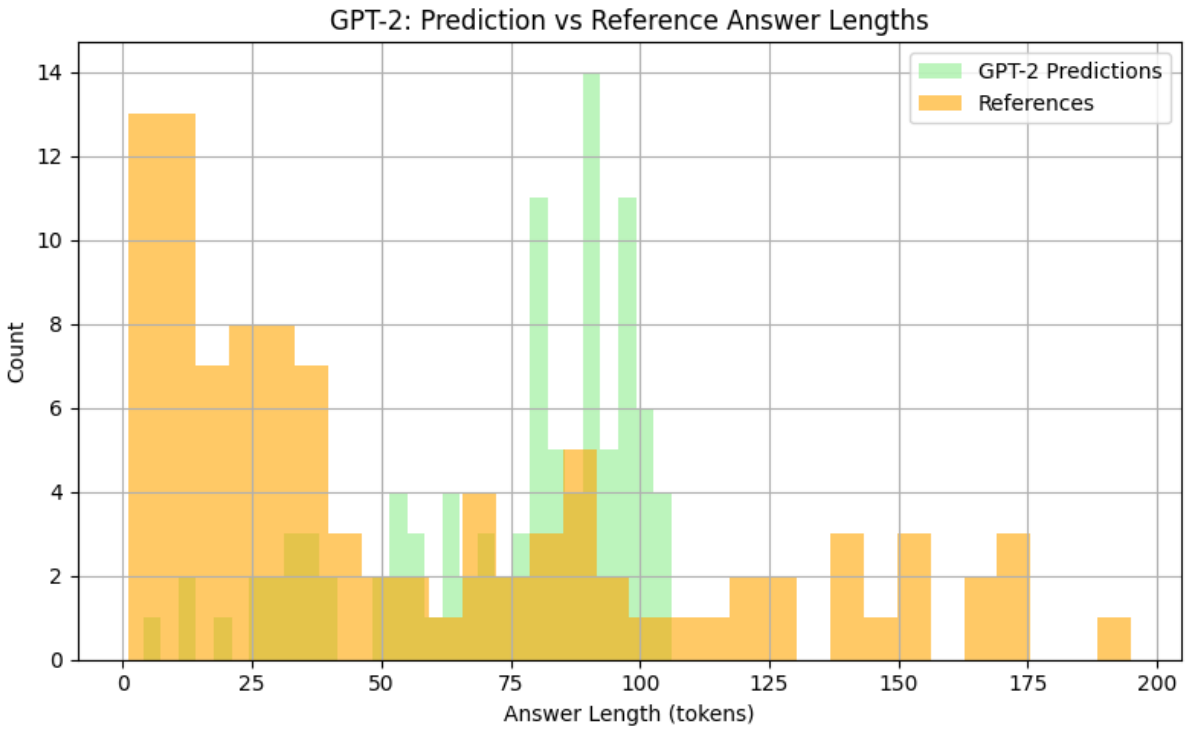
GPT-2

I fine-tuned the base gpt2 model using causal language modelling

Inputs were passed as a single concatenated text block, as GPT-2 is not trained to separate questions from answers explicitly

Padding and attention handling were configured properly to avoid generation issues

Like T5, GPT-2 was trained on 10,000 examples for 3 epochs



3.4 Evaluation Metrics

For both models, I used:

BLEU to measure n-gram overlap precision

ROUGE (ROUGE-1, 2, L, and Lsum) to measure recall-oriented similarity

Side by side answer comparisons for qualitative judgment

Evaluation was done on a small validation set (100 examples) for quick feedback

3.5 Visualization and Performance Comparison

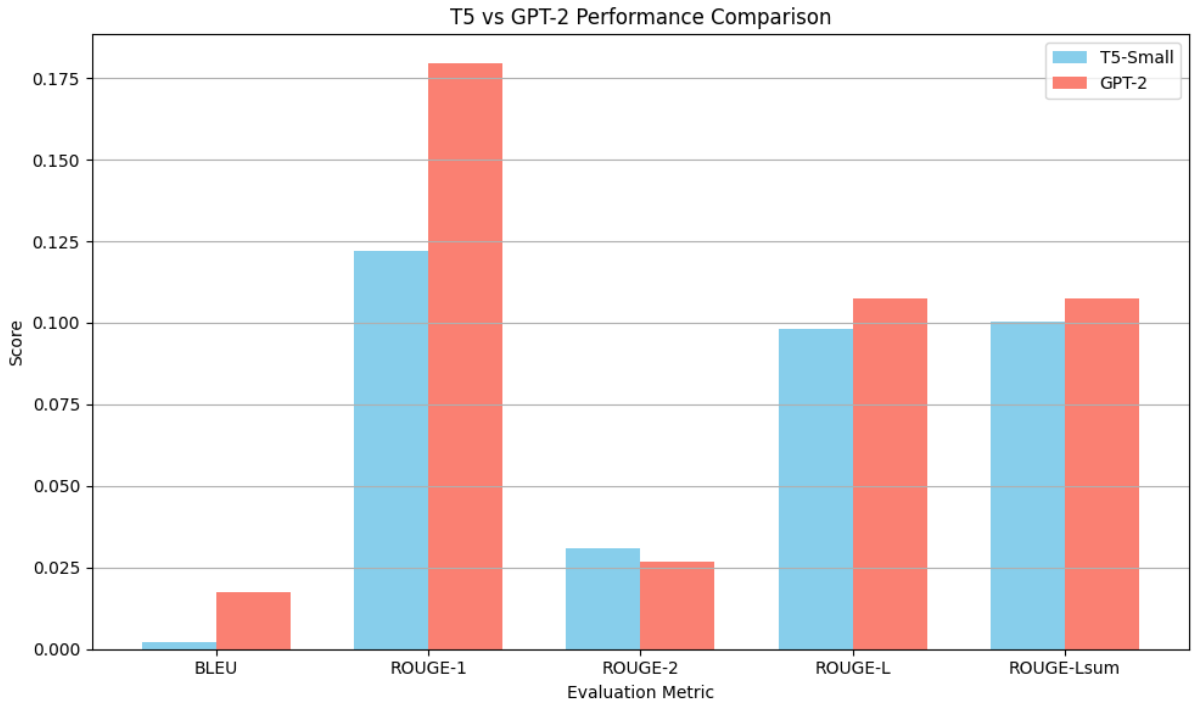
I visualized:

Loss curves for both models to understand convergence

Prediction length histograms versus reference lengths

Direct metric comparisons using grouped bar charts

These visuals made it easier to interpret performance beyond just numeric scores



4. Results

After training both T5-Small and GPT-2 models on the same 10,000 samples, I compared their performance across multiple metrics, visual patterns, and actual generated answers

4.1 Evaluation Metrics

Here's how the two models performed:

Metric	T5-Small	GPT-2
BLEU	0.0023	0.0174
ROUGE-1	0.122	0.1794
ROUGE-2	0.031	0.0268
ROUGE-L	0.098	0.1074
ROUGE-Lsum	0.1002	0.1074

- **GPT-2 outperformed T5** on most metrics, especially BLEU and ROUGE-1, which shows its fluency and surface-level overlap with reference answers
- T5 slightly led on **ROUGE-2**, indicating better handling of bigrams and slightly more coherence in some cases

4.2 Length Distributions

T5's answers were consistently shorter and often underwhelming in depth. This was confirmed by histogram plots showing that most of its predictions were clipped early

GPT-2 produced longer and more detailed responses, often more aligned with the expected length of real answers. However, it occasionally over-generated or included repetitive phrasing

4.3 Training Loss and Stability

T5 Loss Curve: Training and validation loss dropped gradually across all 3 epochs, indicating stable convergence

GPT-2 Loss Curve: Slightly noisier but still improving consistently, with a final validation loss of around 2.83 compared to T5's 2.04

This indicates that both models learned reasonably well, but GPT-2 required more care due to its generative flexibility

4.4 Qualitative Comparison

Here's a summary of how both models performed on example inputs:

Question: What is a proxy and how does it work?

- **T5-Small:** Returned a brief, generic explanation and cut off mid-sentence
- **GPT-2:** Gave a more complete, readable definition with better sentence flow

Question: How can I improve my credit score?

- **T5-Small:** Suggested online courses but sounded vague and generic
- **GPT-2:** Gave more realistic suggestions, such as financial tools, and even added a helpful call to action

Question: What are the symptoms of diabetes?

T5-Small: Mixed up symptoms with causes and felt incoherent at times

GPT-2: Explained it more naturally and medically, though with some redundancy

4.5 Model Behavior Differences

Aspect	T5-Small	GPT-2
Architecture Type	Encoder-decoder (Seq2Seq)	Decoder-only (Autoregressive)
Input Format	Structured (question: format)	Conversational (Q: ... A: format)
Answer Style	Direct and short	More natural, varied, sometimes verbose
Consistency	High due to strict decoding rules	Slightly unpredictable but creative
Ideal Use Case	Controlled QA pipelines or chatbots	Generative FAQs, support automation

Overall, **GPT-2 showed stronger real-world potential** in generating useful and well-structured answers, even if slightly verbose. T5, being lightweight and more predictable, might still be preferred for limited-resource or rule-bound environments

5 Insights and Recommendations

5.1 Key Insights

1. **Data Quality and Distribution Matter**
   During the cleaning phase, I noticed a significant range in answer length, from a few words to over 70,000 tokens. Trimming these extremes helped avoid memory issues and improved the reliability of the models during training
   Insight: Preprocessing decisions directly impact model performance and training stability.

2. **Model Architecture Affects Response Style**
   T5-Small being an encoder-decoder model, produced short, controlled responses but lacked depth in longer answers. GPT-2, on the other hand generated richer and more human-like responses thanks to its autoregressive design
   Insight: Architecture choice should align with the application's tone and flexibility requirements

3. **Evaluation Metrics Only Tell Part of the Story**
   While GPT-2 scored higher on most metrics, the real difference was clear in the generated examples. GPT-2 gave fuller and more contextual responses. However, it was also slightly more verbose and sometimes added filler content
   Insight: A hybrid approach of metrics + qualitative review provides a more complete evaluation

4. **Model Size vs Performance Trade-off**
   Both models used were small (T5-Small and base GPT-2), which allowed for faster experimentation. Still, there is clear headroom for improvement if larger variants are used, especially for GPT-2
   Insight: Model scaling could improve quality but requires resource planning

5.2 Recommendations

1. **Fine-tune Larger Models (eg, GPT2-Medium, T5-Base)**
   Upgrading to slightly larger versions can help improve fluency and coherence without a major increase in complexity. This is especially valuable for answers that require reasoning or nuance

2. **Add Contextual Support with Retrieval-Augmented Generation (RAG)**
   For real-world applications, answers could be improved by grounding them in external documents or a knowledge base. This would prevent hallucinations and improve accuracy

3. **Ensemble or Hybrid Systems**
   A potential next step is combining T5 and GPT-2. For example, using T5 for short factual queries and GPT-2 for more open-ended or conversational responses

4. **Integrate Confidence Scoring or Filters**
   Generated answers can be filtered based on a confidence score or answer length thresholds to improve quality in production environments

5. **Expand Evaluation Scope**
   While BLEU and ROUGE are useful, human evaluation or embedding-based metrics like BERTScore could provide a more nuanced view of model performance

6. **Aviation Use Case Fit**
   GPT-2's strength in natural language generation could be ideal for **passenger query bots**, **onboarding FAQs**, or **support ticket triaging**. T5 can be used in **internal tools** where predictability is more important than creativity

6. Conclusion

- In this case study, I set out to build and evaluate a state-of-the-art question-answering system using the Quora Question Answer Dataset. I explored two powerful transformer-based model, T5-Small and GPT-2, and compared their performance across multiple angles including metrics, output quality, and practical usability
- The results clearly showed that GPT-2 generated more natural, detailed and human-like answers, making it a stronger choice for applications that value conversational

quality. T5-Small, while more concise and consistent, was better suited for controlled, rule-based environments

- Beyond just training models, this project gave me a deeper understanding of the trade-offs involved in choosing architectures, the importance of data preprocessing, and how to balance evaluation metrics with human judgment
- This experience reaffirmed that great models aren't just about high scores, but about the ability to deliver useful, accurate and reliable outputs that can scale in real-world settings. With further tuning, retrieval integration, and scaling up, this QA pipeline has the potential to serve enterprise-grade use cases, especially in customer support, knowledge assistants, and digital help desks