Framework for Questions Answering

Mouhamadou Lamine Bara THIAW

May 2025

1 Introduction:

This report presents a modular framework for building and evaluating Question Answering systems, particularly focused on short, security-related questions. The system explores multiple families of models: encoder-decoder transformers (e.g., T5, BART), decoder-only models (e.g., LLaMA 3 via Ollama), and retrieval-augmented generation (RAG) pipelines. It includes preprocessing, supervised finetuning, few-shot in-context learning (ICL), and advanced retrieval augmented generation (RAG) techniques.

The full codebase is publicly available here.

2 Dataset desciption:

The dataset contains question-answer pairs, primarily centered around security policies. Each question is a short prompt, and the corresponding answer is a short declarative sentence.

Here are some visualizations.

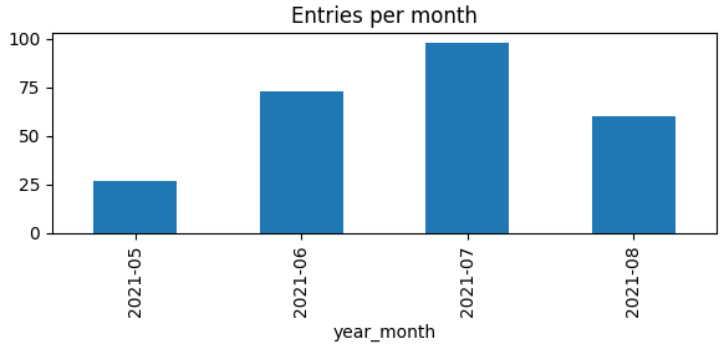

Figure 1: Entries per month

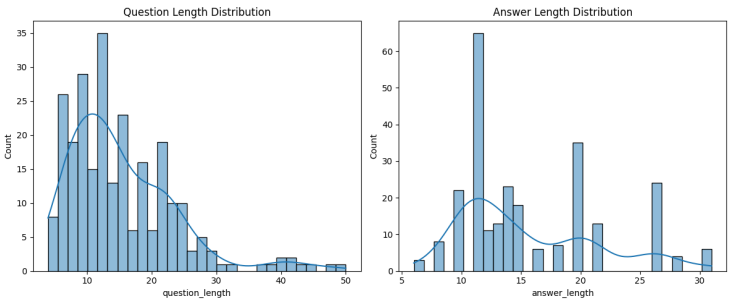

Figure 2: Length distribution

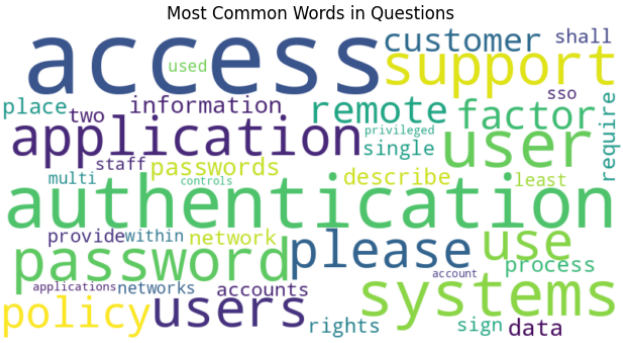

Figure 3: Most common words

2.1 Processing:

The training data was split into train (70%), validation(15%), and test (15%) sets with a fixed seed for reproducibility. A separate blind test set is used for final submission. We computed and visualized token length distributions to set appropriate generation limits for some LLMs. We extracted also the top 100 most common words for context enrichment with advanced RAG techniques.

3 Models and Evaluation Metrics:

3.1 Models

We evaluate the QA system on:

- **Supervised Question Answering:** Finetune seq2seq models on the labeled train data.
- **In-Context Learning (ICL):** Provide few training examples as prompt context.
- **RAG:** Use context retrieved from Wikipedia with our most 100 commons words and training data to augment answers.

3.2 Metrics

We report standard generation metrics:

- **BLEU-1 to BLEU-4:** Measures n-gram overlap between generated and reference answers.
- **ROUGE-L:** Captures the longest common subsequence.

4 Implementation details

We implemented our question-answering framework using Python and Hugging Face's `transformers` library. For the sequence-to-sequence models, we fine-tuned `FLAN-T5` and `BART` on a labeled dataset of security-related questions and answers. The models were trained with a batch size of 4, a learning rate of 5×10^{-4}, and up to 10 epochs maximum.

To ensure reproducibility, we fixed all random seeds and used consistent train/validation/test splits. We also implemented a Retrieval-Augmented Generation (RAG) pipeline using `LangChain` and `Ollama`, where documents retrieved from Wikipedia and the training set were used to enrich the generation context. For text similarity and retrieval, we employed the `nomic-embed-text` model for dense embedding generation.

All experiments were conducted on colab.

5 Results and discussions:

We evaluated all our different strategies with the metrics discussed previously. Here are the results.

5.1 Results

Table 1: Evaluation Metrics on the Split Test Dataset

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L
BART	0.0074	0.0041	0.0028	0.0018	0.0105
T5	0.6610	0.6451	0.6399	0.6360	0.6864
Simple RAG	0.6807	0.6617	0.6483	0.6357	0.7248
Advanced RAG	0.6820	0.6592	0.6447	0.6317	0.7367

5.2 Discussions:

Our experiments demonstrate that fine-tuned encoder-decoder models such as `FLAN-T5` achieve strong performance in the context of domain-specific question answering, particularly when trained on a carefully curated dataset. Compared to `BART`, `FLAN-T5` showed better generalization, as reflected in BLEU and ROUGE-L scores on the test set.

Incorporating In-Context Learning with demonstrations from the training set slightly improved the results, while Retrieval-Augmented Generation (RAG) proved to be a more effective strategy. By enriching the input context with external knowledge (e.g., Wikipedia entries of domain-specific keywords), RAG helped the model provide more accurate and informative answers.

One key challenge we observed lies in balancing answer length and relevance. Generative models tend to produce verbose outputs, which can hurt exact match metrics even if the content is semantically correct. We mitigated this by restricting the number of generated tokens and prompting the model to produce concise responses.

6 Future extensions:

Future improvements could involve reinforcement learning with human feedback (RLHF) or supervised reward modeling to better align outputs with user expectations. Additionally, using more advanced retrieval systems can be studied.