RAG Using TF-IDF and BART/FLAN-T5

1. Problem Statement

This project implements a **Retrieval-Augmented Generation (RAG)** system that combines traditional document retrieval (TF-IDF) with generative language models (BART and FLAN-T5) for question answering.

The system aims to:

- Retrieve relevant document sections using TF-IDF.
- Generate accurate answers using base and fine-tuned versions of BART and FLAN-T5 (on SQuAD v2).
- Evaluate the comparative performance of each model.

2. System Overview

2.1 Text Corpus and Questions

- Documents from five technology domains:
    - Machine Learning
    - Artificial Intelligence
    - Cybersecurity
    - Web Technology
    - Social Networking
- Each topic includes 8–9 curated questions to evaluate retrieval and generation quality.

2.2 Data Preprocessing

- Text is cleaned (lowercased, whitespace removed).
- Tokenization: Split into chunks of **M=3 sentences** with an **overlap of N=1 sentence**.
- Corpus and metadata are stored for efficient indexing and retrieval.

2.3 Document Retrieval using TF-IDF

- TF-IDF vectorizer encodes both documents and queries.

- Cosine similarity is used to retrieve the **top-k** most relevant chunks.
- Each "chunk" (token) may contain multiple sentences to preserve context.

2.4 Answer Generation

- Retrieved context is combined with the user question to form a complete prompt.
- Four transformer models are used for generation:
    - `facebook/bart-base`
    - `a-ware/bart-squadv2`
    - `google/flan-t5-base`
    - `sjrhuschlee/flan-t5-large-squadv2`
- Fine-tuned models (on SQuAD2) improve contextual understanding and accuracy.

3. Evaluation

Evaluation is performed using:

- **Exact Match (EM):** Measures if the predicted answer exactly matches the reference.
- **F1 Score:** Measures overlap between predicted and actual answers.

3.1 Average Performance Metrics

Model	Exact Match	F1 Score
FLAN-T5	0.36	0.61
FLAN-T5 (SQuAD2)	0.59	0.83
BART	0.00	0.05
BART (SQuAD2)	0.11	0.57

4. Challenges

4.1 Model Limitations

- Base models (e.g., BART) are not optimized for QA tasks.
- Fine-tuned models (e.g., BART-SQuAD2, FLAN-T5-SQuAD2) showed marked improvement.

4.2 Computational Efficiency

- TF-IDF becomes less efficient on large datasets due to its sparse nature.
- Retrieval and answer generation are computationally expensive for real-time use.

5. Recommendations for Improvement

- **Dense Retrieval**: Replace TF-IDF with dense embeddings (e.g., Sentence-BERT) for better scalability and relevance.
- **Model Fine-Tuning**: Further fine-tune models on domain-specific QA datasets for higher precision.
- **Use of Frameworks**: Adopt RAG frameworks like **Langchain** for model orchestration, experimentation, and evaluation.