RAG for Fact Checking using FEVER Dataset
with Explainable-AI and Reflection

Apple Analytics : Saurav, Lahiri, Savithri, Taniksha

1. Introduction

Automated fact-checking has become a critical tool to combat misinformation. In this project we explore:

a) Retrieval-Augmented Generation (**RAG**) combined with
b) **Explainable AI** (XAI) : token level attribution for claim & retrieved evidence passages.

to build an interpretable fact-checking system. Using the FEVER dataset, a hybrid pipeline that retrieves relevant evidence from Wikipedia-like sources indexed in a Vector DB, and uses a local large language model (LLM) to evaluate the claims. The project emphasizes transparency, providing visual and token-level explanations for how conclusions are derived.

We also aim to explore the following advanced techniques

- **Fine Tunin**g LLM with Fever Dataset using **PEFT** (**LoRA** / **QLoRA).**
- **Reflection**: RAG pipeline enhanced with limited Self-Reflection capabilities (Retrieval, document relevance etc)

2. Dataset: FEVER

The FEVER dataset contains over 250K human-generated claims labelled as *SUPPORTED, REFUTED, or NOT ENOUGH INFO*, with evidence retrieved from Wikipedia.

- *Embed* (e.g. *sentence-transformers/all-mpnet-base-v2*) "evidence sentences" from Wikipedia *and Index* them in a vector store (*Chroma, FAISS etc*). Claim is used as input query
- *Chunking* is not required as we are indexing sentences and not full articles.
- We would use consolidated dataset from Huggingface : copenlu/fever_gold_evidence

*Alternately : After completing with FEVER dataset, we will explore implementation on Muptihop **HOVER** Dataset*

3. Methodology

3.1 RAG Architecture

- **Retriever**:
  - *MultiQueryRetriever* from langchain. It has better recall as it can generate multiple variations of the query
  - Sentence embeddings generated (using e.g.*BAAI/bge-m3*) and indexed in *Vector DB (Chroma or FAISS*).
- **Generator**: LLM (e.g. *Mistral-7B-Instruct Or llama2-7B*) via *HuggingFace* transformers to generate fact checking response.
- **Pipeline**: Leverage *LangChain RetrievalQA* retriever integrated with LLM.
- **Prompting**: Reasoning-focused prompt template to elicit step-by-step explanations.

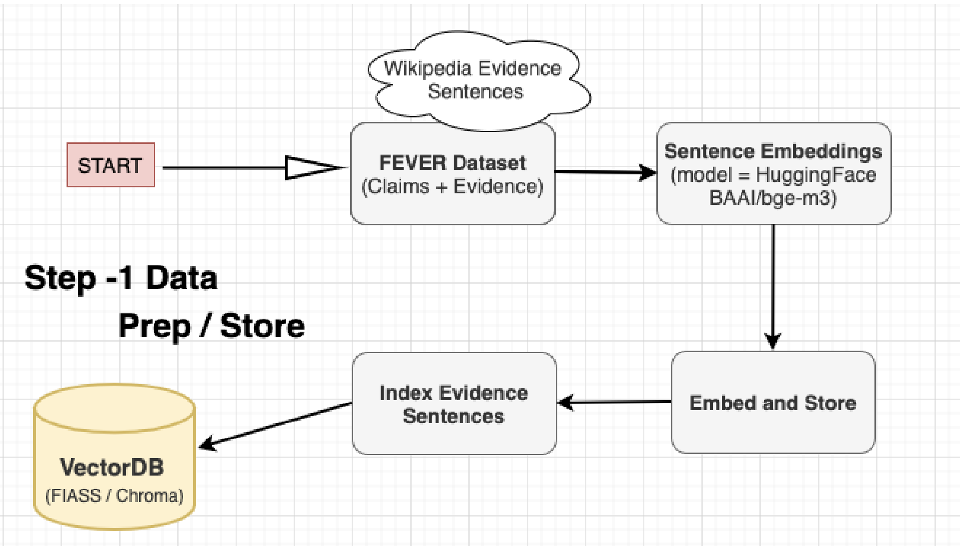

Figure 1: Data Preparation into Vector DB

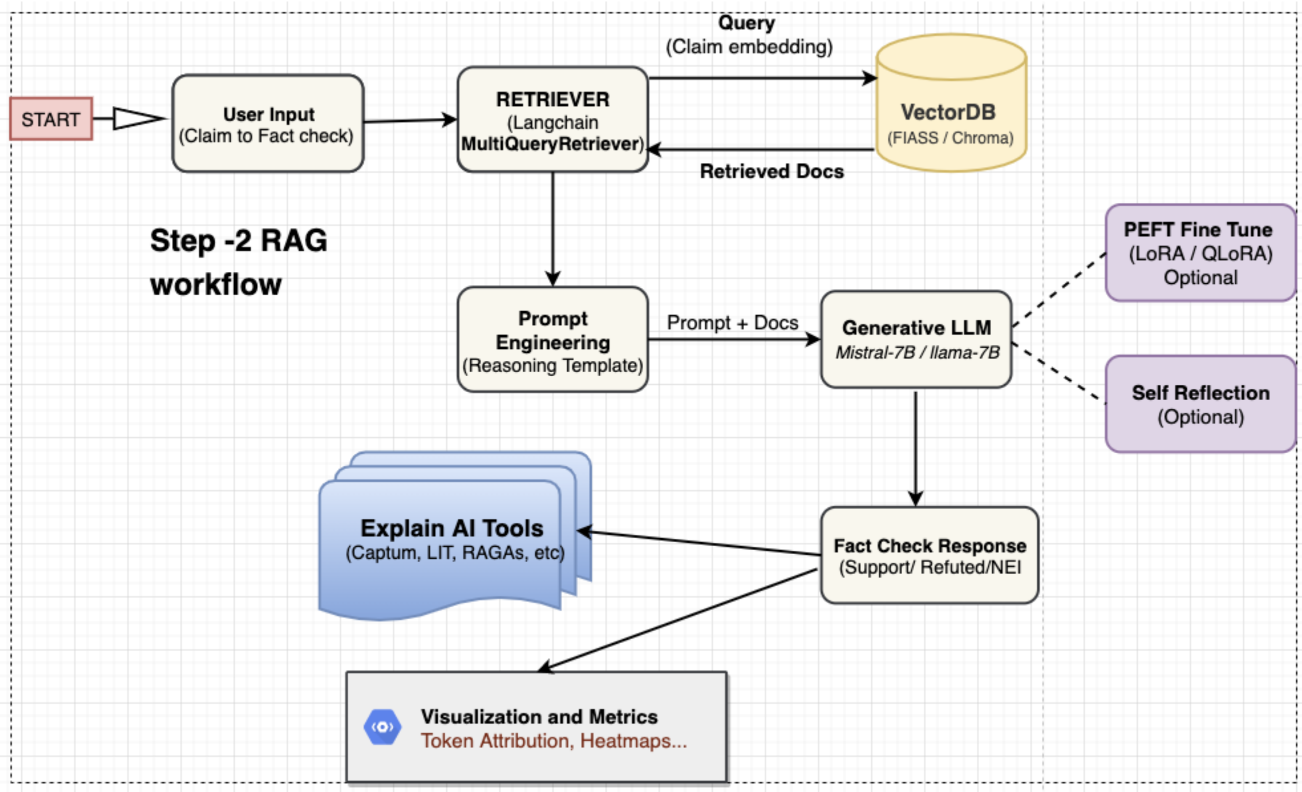

Figure 2: RAG pipeline and flow

3.2 (XAI) Explainability Integration

Some XAI tools listed below which could be integrated for XAI

- **Captum**: For token level attribution to explain LLM decisions.
    - Claim tokens & Evidence Tokens retrieved from DB
- Explore **"Langchain callbacks**" to log full chain, prompt LLM outputs etc
- **LIT** (Language Interpretability Tool): To analyze LLM predictions, attention mechanisms, embeddings, and counterfactual
- **Attention visualization** : Using tools like ***transformers-interpret*** from Huggingface

3.3 User Interface (Optional)

Streamlit web interface for Input claim, view evidence, decisions, and explanations.

4. Evaluation and Results

4.1 Comparative Metrics Analysis

Compare Accuracy and Confusion Matrix (Precision, Recall, F1 scores) on

Baseline RAG	Baseline RAG Multi-Model	With PEFT Tuned LLM	Reflective RAG
• Accuracy • Confusion • RAGAs metrics	• Accuracy • Confusion • RAGAs metrics	• Accuracy • Confusion • RAGAs metrics	• Accuracy • Confusion • RAGAs metrics

4.2 Faithfulness, Answer relevancy using RAGAS

Ragas a framework to evaluate RAG pipelines for additional metrics like *faithfulness, Answer correctness, relevancy, factual correctness.*

4.3 Visualizations

Visualizations that would be used in reporting (not limited to).

- **Heatmaps** showing evidence relevance scores
- **Bar plots** of token attributions via Captum
- **Pie charts** for prediction label distributions
- **Similarity Heatmaps**: Visualize Embedding/Cosine similarity b/w claim and retrieved evidence.

4.4 Multi-Model Analysis (optional)

For comparing alternative LLMs (e.g., LLaMA-2, GPT-4, etc.)

5. Steps

- Step 1: Build Basic RAG pipeline with base LM
- Step 2: Explanable AI integration in the pipeline for decisions
- Step 3: Relf-RAG based improvements (Optional)
- Step 4: PeFT Fine-tune/Adaptation of base LM on given Dataset (Optional)
- Step 5: Future work on RAGs and reflect if RAG is required when Latest models like Gemini 2.0 have 1M large context window

6. Conclusion

This project demonstrates the feasibility of using a local, interpretable RAG pipeline for fact-checking. By integrating XAI techniques, we offer insights not just into what the model predicts, but *why*. This transparency is vital for high-stakes applications such as misinformation detection. We also study the effects of integration of advanced techniques like Self-Reflection abilities and PEFT Fine tuning on performance.

7. Future Work

- Full Fine-tuning the LLM on FEVER-style prompts.
- Explore other Multi-hop fact datasets like HOVER
- Deploying the pipeline via a web app for public or research use.

8. References

[1] Patrick Lewis et el, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks". 34th International Conference on Neural Information Processing Systems (NIPS '20). Curran Associates Inc., Red Hook, NY, USA, Article 793), 2020. https://arxiv.org/abs/2005.11401v4

[2] M. Abdul Khaliq et el, "RAGAR, Your Falsehood Radar: RAG-Augmented Reasoning for Political Fact-Checking using Multimodal Large Language Models". Seventh Fact Extraction and VERification Workshop (FEVER), pages 280–296, USA. Association for Computational Linguistics. https://arxiv.org/abs/2404.12065

[3] Russo, Daniel & Menini, Stefano & Staiano, Jacopo & Guerini, Marco. (2024). Face the Facts! Evaluating RAG-based Fact-checking Pipelines in Realistic Settings. https://arxiv.org/abs/2412.15189

[4] FEVER Dataset : https://huggingface.co/datasets/copenlu/fever_gold_evidence

[5] *RAGAS for additional RAG performance metrics:* https://github.com/explodinggradients/ragas

[6] https://fever.ai/

[7] FAISS : https://github.com/facebookresearch/faiss

[8] Chroma: https://github.com/chroma-core/chroma

[9] https://www.rdworldonline.com/recursive-fact-checking-tool-addresses-gaps-in-genai-fact-checking/

[10] RAG Driven Generative AI by Denis Rothman, packt, 2024. https://www.amazon.com/RAG-Driven-Generative-retrieval-generation-LlamaIndex/dp/1836200919

[11] HOVER dataset : https://hover-nlp.github.io/