Fake News Detection Using Large Language Models with Retrieval Augmented Generation (RAG)

Rahul Kumar, Rajat Chaudhary, Pavankumar Kulkarni, Rahul Rai, Vishakha Kumari, Annam Mukunda Saiteja and Yelamarthi Manoj Kumar

Codebase is available at this GitHub repository
User interface website can be accessed here at link

Abstract. In today's digital era, where information is accessible instantly and news is disseminated across numerous platforms, distinguishing real news from fake has become critically important. The proliferation of misinformation can significantly influence public opinion, disrupt social harmony, and serve as a tool for personal or political gain. Consequently, developing systems to automatically detect and filter fake news has become a pressing technological and societal challenge. This project aims to build a prototype system for fake news detection using large language models (LLMs) in conjunction with a Retrieval-Augmented Generation (RAG) framework. The goal is to utilize advanced AI techniques to identify patterns, linguistic cues, and contextual inconsistencies that are indicative of false or misleading information.
Index Terms - Fake news detection, Large Language Models (LLMs), Retrieval-Augmented Generation (RAG), Prompt Engineering, Langchain

1 Introduction

In today's fast-paced world, misinformation spreads quickly, causing confusion, dangerous behaviors, and weakening trust in institutions. Reliable verification methods are critical to counter these effects. In response, we present a robust, multilingual news verification system that harnesses the power of large language models (LLMs) and Retrieval-Augmented Generation (RAG) techniques to classify user submitted claims— **textual or audio as REAL, FAKE or UNSURE.** Grounded in evidence from reliable sources, the system delivers contextual explanations that enhance transparency and trust. Our approach begins by defining the problem of misinformation and analyzing the unique requirements of multilingual and multimodal inputs. We then design and implement a modular architecture that combines multi-query generation, document retrieval, summarization and factual verification. Integrating APIs from Serper.dev, ChatGroq and SarvamAI. Performance is rigorously evaluated using precision, recall, classification accuracy, and latency metrics with system improvements guided by failure case analysis and user feedback.

Key features of the system include support for both text and audio inputs, making it flexible and user-friendly. It supports multiple Indian languages, ensuring accessibility to a diverse population. By leveraging real news sources, the system provides clear explanations about why a claim is classified as true or false, enhancing transparency. The decision making process harnesses advanced technologies, including large language models (LLMs) and Retrieval-Augmented Generation (RAG), to ensure accurate and contextually grounded verification.

2 Architecture

The proposed fact verification system is designed to evaluate the veracity of user-provided claims expressed in English or any supported Indian language, accepting both text and audio modalities. Given the predominance of trusted news sources and evidence repositories in English, all non-English claims undergo translation to English early in the processing pipeline to standardize downstream components. The main steps are as below:

- **Input (Text or Audio)** : When a claim is provided in text format, it is first translated into English using the Sarvam Translate API, which is powered by the Mayura-v1 model. For audio inputs, the system first transcribes the spoken claim into text using the Sarvam Speech-to-Text (STT) API, which utilizes the Sarika-v2.5 speech recognition model. The resulting transcription, which may be in any supported language, is then passed through the same translation API to obtain the English equivalent of the claim.
- **Claim Rephrasing** : To enhance evidence retrieval and ensure robust search coverage, the translated claim is processed through a rephrasing module that employs the Llama3-8b-8192 large language model. This component generates three alternative phrasings of the original claim, thereby implementing a multi-query strategy. The objective is to simulate the diversity of expressions that users might employ to convey the same underlying information, which in turn increases the likelihood of retrieving relevant documents.
- **Search for Evidence** : Each of these rephrased queries is then submitted to a targeted search process via the Serper.dev API. This component focuses on retrieving content from credible news sources, including News18, PTI, and PolitiFact. The resulting documents form the basis for evidence collection and are subsequently passed to a summarization module to distill their factual content.
- **Summarize** : Directly using raw evidence for claim verification often yields suboptimal results, particularly when the claim is false or when no highly relevant documents are retrieved. To address this, the system employs the Qwen3-32b language model to generate a concise summary of the collected evidence. This summarization step is designed not only to extract key information

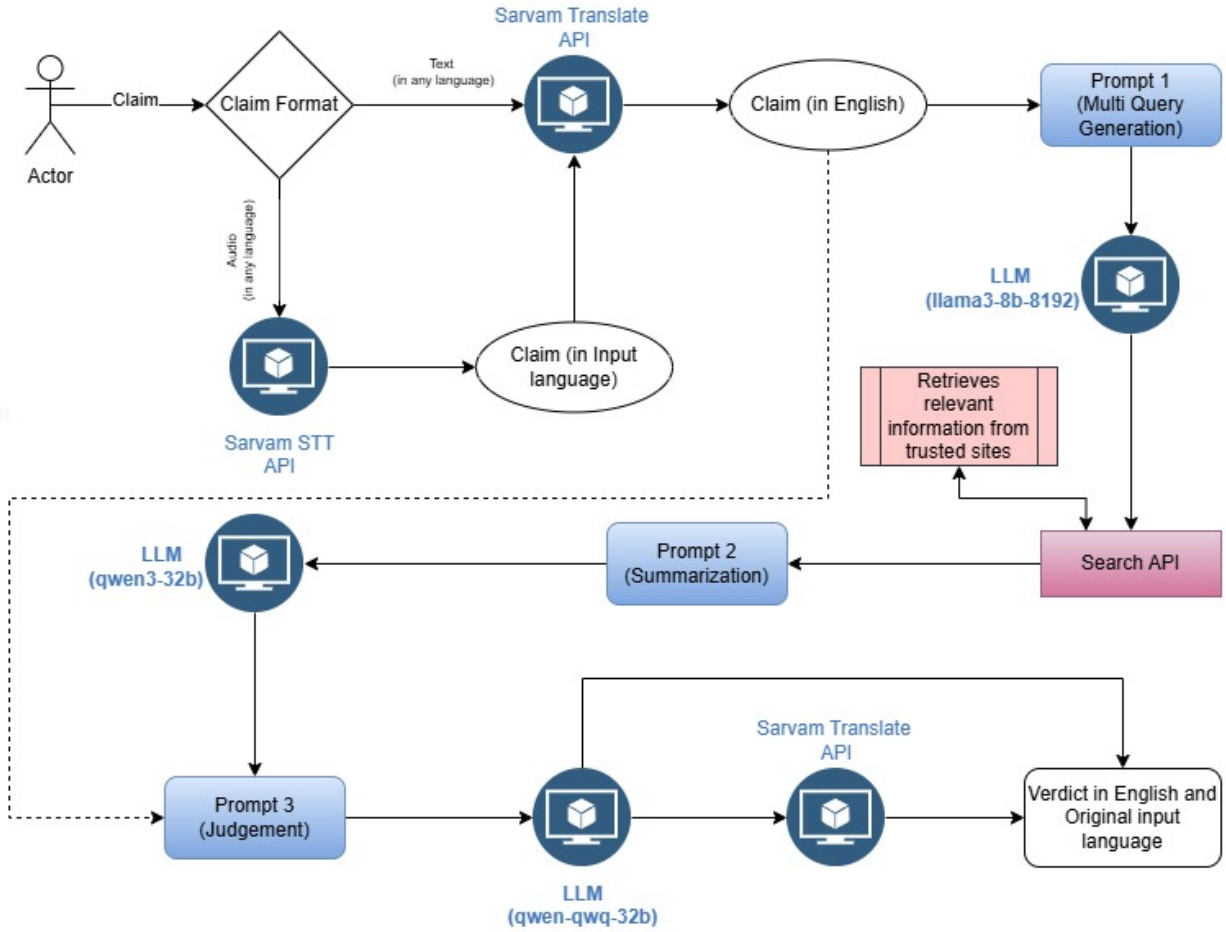

Figure 1. System architecture diagram for fake news detection.

from relevant sources but also to establish contextual grounding in cases where direct evidence is sparse or absent.

- **Verdict** : The final verdict is produced by the Qwen-QWQ-32b model, which receives both the original (translated) claim and the evidence summary as input. This model outputs a classification label—REAL, FAKE, or UNSURE—along with a brief explanatory rationale. The explanation is intended to enhance transparency and user trust in the system's decision-making process.
- **Output** : Finally, the system translates the verdict and its accompanying explanation back into the user's original input language using the same translation pipeline employed earlier. The final output comprises the verdict and explanation in both English and the original language of the claim.

  Supported languages: Bengali, English, Gujarati, Hindi, Kannada, Malayalam, Marathi, Odia, Punjabi, Tamil, Telugu

3 Model Evaluation

In light of the rapidly evolving ecosystem of large language models (LLMs) and task-specific transformers, selecting appropriate models for the individual components of our fact verification pipeline constituted a key design decision. Each model variant was tested under different prompting paradigms, and performance was assessed based on critical evaluation metrics and accuracy in final verdict classification. Quantitative evaluation emphasized overall classification coverage and F1-score, reflecting both model precision and robustness. However, due to constraints in computational resources and limited time availability, we selected the models for multi-query generation and evidence summarization based primarily on existing literature, online benchmarks, and empirical findings reported by the broader research community, rather than direct experimentation. Despite this constraint, the empirical evaluation conducted for verdict classification informed our final model choices across the pipeline:

Strategies

- **Strategy 1** : Multi-query generation from the input claim, followed by document retrieval and summarization. The resulting summary was then passed to the verdict-generation prompt (Judge Prompt).
- **Strategy 2** : Multi-query generation and document retrieval, with raw retrieved documents passed directly to the verdict-generation prompt, skipping summarization.
- **Strategy 3** : Direct retrieval using the original claim (no rephras-

Table 1. LLMs comparison (in English) across three evaluation strategies

Model	Strategy 1			Strategy 2			Strategy 3		
	Coverage	F1 (Real)	F1 (Fake)	Coverage	F1 (Real)	F1 (Fake)	Coverage	F1 (Real)	F1 (Fake)
llama3-8b-8192	58%	0.82	0.53	46%	0.93	0.38	64%	0.88	0.71
qwen/qwen3-32b	63%	0.89	0.60	33%	0.92	0.33	72%	0.90	0.74
mistral-saba-24b	63%	0.90	0.59	33%	0.91	0.33	67%	0.90	0.68
deepseek-r1-distill-llama-70b	65%	0.90	0.63	31%	0.93	0.32	69%	0.90	0.69
meta-llama/llama-4-scout-17b-16e-instruct	67%	0.90	0.65	30%	0.94	0.34	70%	0.91	0.71
meta-llama/llama-4-maverick-17b-128e-instruct	68%	0.90	0.66	32%	0.92	0.38	52%	0.95	0.65
qwen-qwq-32b	84%	0.90	0.88	32%	0.93	0.40	67%	0.91	0.73

Table 2. LLM in regional language (Hindi/Kannada) and strategy 1

Model	Coverage	F1 (Real)	F1 (Fake)
llama3-8b-8192	68%	0.92	0.69
qwen/qwen3-32b	75%	0.91	0.76
mistral-saba-24b	68%	0.90	0.60
deepseek-r1-distill-llama-70b	70%	0.88	0.58
meta-llama/llama-4-scout-17b-16e-instruct	70%	0.89	0.55
meta-llama/llama-4-maverick-17b-128e-instruct	66%	0.89	0.55
qwen-qwq-32b	66%	0.89	0.57

ing), followed by summarization and verdict generation.

Each combination of model and strategy was evaluated based on:

- **Total Coverage :** The proportion of claims for which the model returned a definitive classification (REAL or FAKE).
- **F1 Score (Real)**
- **F1 Score (Fake)** : The harmonic mean of precision and recall for correctly identifying real and fake claims, respectively.

4 Evaluation and Results

Based on the strategies and multiple model testing, the results are as follows:

- **LLM evaluation in english language (Table 1.)** : Compares the performance of LLMs in english across three different strategies for fake news detection. As strategy 1 was best performing, we carried subsequent tests using strategy 1 (due to compute resource constrains)
- **LLM evaluation in regional language Hindi/Kannada (Table 2.)**: LLMs performance for strategy-1 in regional language for fake news detection.
- **LLM model evaluation using audio (all languages) (Table 3.)**: LLMs evaluation using audio in all languages and strategy-1

Table 3. LLM model evaluation using audio (all languages) and strategy 1

Model	Coverage	F1 (Real)	F1 (Fake)
llama3-8b-8192	22%	0.57	0.28
mistral-saba-24b	22%	0.95	0.52
qwen-qwq-32b	51%	0.66	0.64

- **Sarvam Speech to Text**

Table 4. ASR Evaluation Metrics

Metric	Score
WER Score	0.288732
CER Score	0.088717

- **Sarvam Translation Score**

Table 5. Evaluation Metrics

Metric	Score
BLEU Score	0.2027
METEOR Score	0.4949
BERT Score	0.9149

5 Model Selection

Based on the empirical results observed in the evaluation section above, the final models selected are :

- Llama3-8b-8192 was adopted for multi claim generation as online benchmarks showed this model was best suited for the task.
- Qwen3-32b was adopted summarization as online benchmarks showed this model was best suited at summarization of large text corpus.
- Qwen-QWQ-32b was chosen for verdict generation, as it consistently exhibited strong alignment between the provided evidence summary and the final classification label, while also delivering concise, interpretable justifications.

Each of these models was paired with carefully crafted prompting strategies that were refined during the evaluation process to optimize task-specific performance. This hybrid model selection methodology—combining empirical validation with strategic model research—ensured that the system remained both computationally feasible and reliable in delivering transparent, multilingual fact verification across diverse input modalities.

6 Conclusion

The model demonstrates strong performance in distinguishing between real and fake news, achieving an F1 score close to 0.9 for both classes. To enhance coverage and further improve validation reliability, the list of trusted sources should be expanded. While the current implementation focuses on fake news detection, the underlying framework is designed to be adaptable and can be extended to a broad range of verification tasks. Trusted sources in such applications could include both online websites and locally managed data repositories.

7 Contributions

Rahul Kumar : rahulkumar18@iisc.ac.in
My top three contributions to the project were as follows: 1. Translation module 2. Model selection 3. Online hosting (via Huggingface spaces)

For the translation module, I evaluated multiple options and ultimately chose Sarvam AI due to its excellent support for a wide range of Indian languages and dialects. To ensure its effectiveness, I conducted comprehensive testing using publicly available datasets from Kaggle, which confirmed the model's strong performance and suitability for our project's multilingual requirements.

There are several models available today that offer the capabilities required for our project. I conducted extensive research online, including consulting reputable AI resources and chatbots, to identify the most recommended models. Our team then performed a comparative performance analysis on these shortlisted models using a common dataset. We evaluated them based on well-known metrics to determine the most suitable model for our fake news detection system.

I managed the online deployment of our fake news detection system using Hugging Face Spaces, a platform I had not used before. This required a significant learning curve, particularly in working with Gradio. The default Gradio template did not meet our project's requirements, so I extensively customized the UI to tailor it to our specific use case. This involved a great deal of experimentation with GradIO's components to design an intuitive and effective user interface that clearly presented our model's functionality and results.

In addition to my technical contributions, I was actively involved in team discussions throughout the project lifecycle. I regularly provided strategic inputs across different areas, to help refine and optimise the overall project performance.

Rajat Chaudhary : rajatc@iisc.ac.in
Contributed to developing the multilingual audio input pipeline for our fake news detection system. My key responsibilities included:

- Integrating Sarvam Speech-to-Text (STT) using the Sarika-v2.5 model to transcribe spoken claims into text across multiple Indian languages.
- Evaluating transcription accuracy using WER and CER metrics
- Implementing translation and formatting steps to align spoken inputs with LLM on transcribed claims.

I worked with models such as Qwen-QWQ-32b, LLaMA3 and Mistral to evaluate their ability to classify audio based claims as real or fake. This involved testing multilingual scenarios and analyzing model accuracy and failure patterns. I also contributed to audio claim cleaning and dataset preparation and validated both text and audio responses using the Hugging Face UI.
Experiment : I tried using phonemes to make transcriptions more consistent across different Indian languages and accents. I also tested some audio cleanup steps like removing background noise, trimming silence and automatically detecting the language being spoken. However, these methods sometimes made the results worse causing problems in translation
I also experimented with using models that directly convert spoken audio into English text, instead of following the usual two step process. First transcribing the speech in the original language, and then translating it to English. But unfortunately, this method often missed important details, especially when the original audio contained slang etc
Paper: Wrote this particular paper using LaTeX, led team discussions to collect everyone's input and managed the process of putting the final paper together for submission.
Learning : Through this project, I gained practical experience in building end to end audio processing pipelines applying evaluation metrics for transcription quality and aligning multimodal data for model inference. I developed skills in multilingual data handling, model selection and academic paper writing. Key challenges involved handling inconsistent transcriptions across diverse Indian languages, ensuring accurate translations. These efforts led to a robust pipeline for integrating audio inputs into our fake news detection framework.

Vishakha Kumari : vishakhak@iisc.ac.in

- Testing of the Speech Translation Module : I worked on testing the speech translation capabilities of the system with my team. On testing different alternatives, we chose Sarvam AI because of its improved Indian language and dialect support. For tool reliability, I performed extensive benchmarking with commonly available datasets of Kaggle. The result confirmed the high accuracy and contextual quality of translation of Sarvam AI and suggested it as a suitable choice for our multilingual fake news detection pipeline.
- Dataset Preparation : We collected raw data from Kaggle, Hugging Face, and other places, cleaned and processed it with Python scripts, removing noise, duplicates, and irrelevant records. The purified dataset enhanced model consistency and minimized overfitting in training and evaluation.
- Report Writing and Documentation I spearheaded the development of the project report with the ECAI LaTeX template on Overleaf. I wrote crucial sections, facilitated peer review coordination, and maintained clarity and technical precision. The content was refined using tools such as ChatGPT, Claude, and grammar checkers, and references were made from reputable academic sources. Furthermore, I was an active participant in team discussion forums, offering input across modules to inform strategic decision-making and overall project enhancement.

Yelamarthi Manoj Kumar : manojyk@iisc.ac.in
My key contributions to this project include:

- integrating Groq-hosted large language models (LLMs) into the fake news detection pipeline.
- designing and refining prompts for verdict classification and evidence-based explanation.
- evaluating model performance using a custom English dataset.
- experimenting with multiple Retrieval-Augmented Generation (RAG) strategies to enhance the system's accuracy.

In detail, I was responsible for integrating several Groq-hosted LLMs—including Qwen, LLaMA3, DeepSeek, and Meta-LLaMA—into the fake news detection workflow using LangChain. As a prompt engineer, I crafted and fine-tuned prompts that instructed the model to classify a user-submitted claim as REAL, FAKE, or UNSURE, and to provide a supporting explanation based on retrieved evidence. Initially, I experimented with Hugging Face models, which were able to produce verdicts but often failed to generate reliable or coherent explanations. Through several rounds of prompt alignment and testing, I achieved more consistent and explainable outputs using Groq-hosted models. To assess model performance, I developed

a synthetic dataset of English-language fake and real news claims. I conducted verdict-level evaluations using accuracy and F1 score as metrics. Due to variations in how models expressed reasoning and the lack of labeled explanation data, I did not perform formal evaluation on the explanation quality. Additionally, I implemented and compared two RAG strategies: one based on LangChain's multi-query approach and another using basic evidence concatenation. I repeated the evaluation process across multiple Groq-supported models for each strategy. While I also explored incorporating LangChain memory to retain dialogue context, I could not complete that part due to time limitations. Overall, my contributions significantly enhanced the accuracy and reliability of the English-language fake news verification system.

Annam Mukunda Saiteja : mukundaannam@iisc.ac.in

In this project, I was involved from the ideation phase, contributing to the definition of the problem scope and the overall system architecture. I designed and refined structured prompts for multiple LLMs—including OpenAI's ChatGPT and Google's Gemini models—to classify input claims as REAL, FAKE, or UNSURE, and to generate corresponding justifications grounded in evidence.

As part of the information retrieval pipeline, I evaluated multiple web scraping techniques to collect real-time contextual evidence. While exploring libraries like BeautifulSoup, Scrapy, and browser automation tools, I found that many approaches were either rate-limited, unreliable, or incurred additional infrastructure overhead. Several commercial APIs also posed limitations due to usage quotas or lack of a generous free tier. After evaluating these options, I selected Serper.dev, which offered a cost-effective and reliable solution for live web search. I then implemented a retrieval augmentation module using Serper's API to fetch news snippets and search results. These were embedded into a RAG (Retrieval-Augmented Generation) pipeline to improve the factual grounding of the model outputs. To ensure retrieval relevance, I experimented with various ranking and filtering strategies, such as deduplication, keyword relevance scoring, and date-based filtering, which helped reduce noise from unrelated or outdated sources. For improving prompt-to-response consistency, I further implemented prompt templating, multi-query reformulation, and fallback chaining using LangChain, increasing both evidence diversity and model robustness.

For model evaluation, I developed a benchmark dataset consisting of real-world and synthetic claims, and computed standard classification metrics including accuracy, precision, recall, and F1-score to quantify performance. Explanation quality was reviewed qualitatively, as no gold-standard reference existed for factual justification assessment.

Overall, my contributions spanned prompt engineering, retrieval pipeline design, evaluation methodology, and tooling trade-off analysis—resulting in a more accurate and context-aware fact-checking system capable of real-time web-grounded classification.

Rahul Rai : rahulrai@iisc.ac.in

Contributed in designing and implementing the Web Search Retrieval pipeline to fetch real-time evidence for user claims. I explored multiple APIs (including Bing and DuckDuckGo) and selected Serper.dev as the most suitable for our project due to its generous free tier (2,500 queries/month), easy JSON response format, and reliable search results. Integrated it with Newspaper3k to extract clean full-text content from articles, ensuring the LLM received relevant and trustworthy evidence from high-quality sources.

Contributed in building fallback and edge-case handling logic to guarantee evidence availability, including speculative language detection, language filtering, and basic date validation for time-sensitive claims.

Contributed in engineering and iteratively refining the fact-checking prompt template for the LLM: started with a basic prompt, then experimented with multiple variations—adjusting instructions, formatting, and tone—and observed that more precise, structured prompts consistently produced more accurate verdicts (True/False/Unverified) and clearer rationales with properly cited URLs.

Worked in collecting a balanced dataset of fake and real claims to benchmark and quantitatively evaluate the performance of our different prompt variants, measuring metrics (accuracy, F1, precision/recall) to identify which prompt formulations yielded the best fact-checking results.

Contributed in exploring deployment options for online hosting of our application. I was trying Docker-based hosting on Hugging Face Spaces but faced compatibility and runtime issues. Then we transitioned to use Gradio, which offered a more straightforward and reliable way to host our application with integrated web search, LLM verdicts, and classification, all accessible via a public Hugging Face URL.

Pavankumar Kulkarni : pavankumark@iisc.ac.in

I was involved from the ideation phase and proposed the integration of Sarvam APIs to enable translation capabilities within the project. My Contribution would be proposed end to end **design of the architecture**, and step-by-step data flow, beginning with the user-provided claim, followed by **web search, RAG** and finally, **LLM response generation**

Explored integrating the different components of our fake news detection pipeline using **LangChain**. This included chaining the modules for input parsing, retrieval (via web search), document embedding, and response generation through an LLM.

Actively participated in **evaluating the performance of different LLM configurations**. Through precision, recall, and F1-score calculations to quantitatively assess the accuracy of the predictions.

Also explored **Sarvam AI's APIs for speech-to-text and translation**. This investigation was aimed at extending our project to handle voice-based input in Indian languages. Also, tested our model with multiple languages to exercise the translation pipeline of the architecture.

8 References

- **Sarvam AI Docs :** https://docs.sarvam.ai/api-reference-docs/introduction
- **GradIO Doscs :** https://www.gradio.app/docs
- **Translation Data Sets :** https://www.kaggle.com/datasets/vaibhavkumar11/hindi-english-parallel-corpus https://www.kaggle.com/datasets/parvmodi/english-to-kannada-machine-translation-dataset
- **PTINews** https://www.ptinews.com/
- **Politifact** https://www.politifact.com/
- **News18** https://www.news18.com/
- **Hugging Face**: https://huggingface.co/docs/