Misinformation Detection Using Large Language Model & APIs

Jeet Naik[1], Nishad Bankar[2], Upendra Naik[3], Lisha Khairnar[4], Shruti Parsewar[5], Mahesh Nigade[6]
[1,2]*Computer Engineering AISSMS Institute of Information Technology Pune, India*
[3]*Electrical Engineering AISSMS Institute of Information Technology Pune, India*
[4]*Information Technology AISSMS Institute of Information Technology Pune, India*
[5]*Artificial Intelligence and Data Science AISSMS Institute of Information Technology Pune, India*
[6]*Assistant Professor AISSMS Institute of Information Technology Pune, India*

***Abstract*—Fake information on the internet spreads quite fast [1], and it's getting tougher to handle. A message is endlessly shared, and long before people even bother to check it out, they already take it at face value. As a result, there is a need for tools that can help quickly and reliably check such claims. The website that we built for that purpose is discussed in this paper. It uses LLaMA, a large language model (LLM), Tavily API, and Jina API, which help in web search. Google's Fact Check Tools API is also used to check if the claim has already been verified or not. After the user enters a claim, a result is shown along with an explanation showing it is true, false, or uncertain. The system was tested on 120 different types of claims, and the results show us how accurately the system detects misinformation after integrating APIs, and not just relying on LLaMa. The LLaMA model alone achieves an accuracy rate of only 34.2%, as it is not trained on the latest data. However, using all APIs resulted in 100% accuracy.**

***Index Terms*—misinformation detection, large language model, LLaMA, fact-checking, Jina AI, Tavily, Google Fact Check API, natural language processing**

I. INTRODUCTION

Nowadays, it is very easy to share information. It takes just seconds for anyone to post or forward something. However, the misinformation gets spread just as easily. This can create serious problems, especially in areas like health, politics, or money. Many existing systems rely on humans or on fixed databases having verified claims. But these databases are not useful when the claim is new. This paper proposes a system that uses a large language model with APIs to provide an explanation of why a claim is true, false, or uncertain.

II. BACKGROUND AND RELATED WORK

Older systems used rule-based methods and keyword matching [2]. Now, machine learning models trained on datasets have become common.

Models like GPT and LLaMA can reason and provide explanations. But these models can hallucinate [4], meaning they can sometimes make up stuff that may or may not be wrong.

Retrieval Augmented Generation (RAG) is a method that combines LLMs with document retrieval to reduce hallucination [5]. Our system follows the same idea and uses publicly available APIs to gather and verify information.

III. SYSTEM DESIGN

A. Architecture Overview

It is a website where the user enters text or an URL as input. The backend processes the claim through three steps: evidence gathering, database search, and LLM reasoning. The final verdict and explanation are shown on the same page.

B. Evidence Gathering — Tavily API

When a claim is submitted, the system sends the text to Tavily. Tavily returns a set of relevant web pages along with short summaries of each page. These summaries are then given to the LLM as context.

C. Content Extraction — Jina API

Jina API extracts text content from the web pages retrieved through the Tavily API, instead of processing raw HTML. This extracted content is then given to the

LLM as context, allowing it to analyze the claim based on evidence.

D. Fact-Check Lookup — Google Fact Check Tools API
The Google Fact Check Tools API provides access to fact-checks published by actual journalists and organizations. The system queries this API with the claim and retrieves any matching results. These results include the verdict (for example, "False" or "Misleading") along with the name of the source organization.

E. LLM Reasoning — LLaMA
All gathered evidence and data are combined into a structured prompt and sent to LLaMA. The model is instructed to analyze the evidence, and produce a final verdict with a short explanation.

IV. IMPLEMENTATION

A. Technology Stack

TABLE I. TECHNOLOGY STACK

Component	Technology Used
Frontend	TypeScript, HTML, CSS
Backend & Database	Supabase
LLM	LLaMA (via API)
Web Search	Tavily API
Content Extraction	Jina AI API
Fact Check Database	Google Fact Check API

B. Workflow-
1) The user submits a claim.
2) Tavily API searches the web and returns relevant websites.
3) Jina API extracts clean and structured text from the retrieved websites.
4) Google Fact Check API returns any matching data from its database.
5) LLaMA receives all evidence and produces a final verdict with an explanation.
6) The result is displayed on the same page.

V. TESTING AND RESULTS

A. Dataset and Methodology
To conduct the evaluation, a test set consisting of 120 claims was created across six topics including: Science, Health, Technology, Finance, War/Geopolitics, and Viral/General (20 claims per topic). Each claim was labeled with its actual value of TRUE, FALSE, or MISLEADING based on verified sources.
Every claim was verified using two methods: (1) without external resources by running it through the LLaMA model alone; (2) with all external resources by employing the Tavily, Jina API, and Google Fact Check API.

B. Scoring System
A score between 0 to 100 is shown with the verdict. A score of 75 or above is considered as Likely Credible, scores around 50 are considered Uncertain, and scores of 15 or below are considered Likely Misinformation. Universal facts score 95.

C. Results — Overall Accuracy

Table II: Overall Accuracy Comparison

Metric	LLaMA Only	With APIs
Total Claims	120	120
Correct Verdicts	41	120
Accuracy	34.2%	100%

D. Results — Category-wise Accuracy

Table III: Category-wise Accuracy

Category	LLaMA Only	With APIs
Science	55%	100%
Health	45%	100%
Technology	25%	100%
Finance	20%	100%
War/Geopolitics	15%	100%
Viral/General	50%	100%

E. Key Observation
The initial stage of our project, which only had the LLaMA model, performed poorly on claims related to events from 2024 onwards, consistently returning an Uncertain verdict (score: 50/100, confidence: 50%) due to its training data cutoff. It correctly judged well known facts (e.g., scientific constants, historical events) and obvious lies.
Whereas the final stage of the project, which had the APIs integrated, solved all uncertain cases by retrieving evidence, resulting in correct verdicts.

VI. LIMITATIONS

The system depends on the availability of good web sources. For very recent events or claims, Tavily may not return helpful results. LLaMA can still give incorrect reasoning if the evidence is contradictory. The Google Fact Check API covers claims reviewed only by partner organizations. The system currently is not multi-modal.

VII. CONCLUSION

We developed a misinformation detection website that combines LLaMA with Tavily, Jina API, and the Google Fact Check Tools API. Evaluation on 120 claims across six categories showed improvement from 34.2% accuracy (LLaMA only) to 100% accuracy with the APIs. The key limitation of LLMs is the inability to handle recent or live information, which is solved by the API pipeline. Future work may include formal benchmark evaluation, multimedia and multilingual support.

REFERENCES

[1] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018.

[2] N. Hassan, G. Zhang, F. Adair, J. Hamilton, C. Li, M. Tremayne, J. Yang, and C. Yu, "ClaimBuster: The first-ever end-to-end fact-checking system," *Proceedings of the VLDB Endowment*, vol. 10, no. 12, 2017.

[3] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "FEVER: A large-scale dataset for fact extraction and verification," in *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.

[4] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, no. 12, 2023.

[5] P. Lewis, E. Perez, A. Piktus, *et al*., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2020.