Task#5

Fatima jinnah women university

Rabia Batool

Software engineering

2025

Topic:

**Multi-Language Real-Time Sarcasm & Hate Speech Detection API: Project Report**

**1. Introduction**

This report outlines the development and conceptual framework for a **Multi-Language Real-Time Sarcasm & Hate Speech Detection API**. Unlike traditional sentiment or emotion analysis, this project aims to provide a more enhanced understanding of textual content by identifying subtle linguistic cues indicative of sarcasm, hate speech, offensive language, and varying levels of toxicity across multiple languages. The API is designed for real-time integration, making it suitable for dynamic environments like chat applications, streaming platforms, and comment sections.

**2. Purpose and Core Functionality**

The primary purpose of this API is to offer an advanced text analysis solution that goes beyond basic emotional classification. It focuses on identifying potentially harmful or misconstrued content by:

* **Detecting Sarcasm**: Recognizing when seemingly positive or neutral statements are used with an underlying negative or mocking intent (e.g., "Wow, great job!" said sarcastically).
* **Identifying Hate Speech / Offensive Language**: Pinpointing language that is derogatory, discriminatory, or hostile, particularly when it targets individuals or groups based on race, religion, gender, or other characteristics.
* **Assessing Toxicity Levels**: Quantifying the severity of negative or harmful content, ranging from mild insults to severe threats, allowing for a graduated response.

**3. Key Challenges**

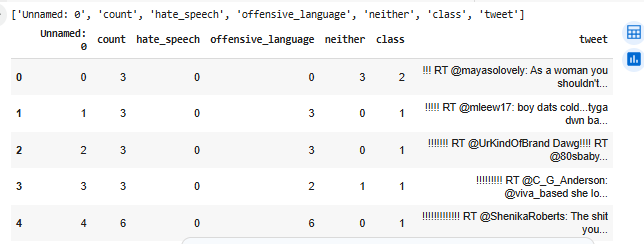
Developing such a system presents several significant challenges:

* **Multi-Language Support**: The system must effectively process and understand text in diverse languages, including English, Urdu, Hindi, and Arabic, which have distinct linguistic structures and cultural nuances. This requires robust language detection and translation capabilities.
* **Complexity of Sarcasm Detection**: Sarcasm is inherently difficult for machines to detect as it often relies on context, tone (which is absent in text), and shared cultural understanding. It's far more complex than identifying basic emotions.
* **Real-Time API Integration**: To be effective in live environments, the API must be highly responsive, capable of processing queries with minimal latency to facilitate real-time content moderation or feedback.
* **Leveraging Free APIs**: The project leverages various free-tier APIs for different functionalities (translation, language detection, NLP models), which necessitates careful management of rate limits, potential downtimes, and ensuring data privacy compliant usage.

**4. Implementation Flow ("How It Works")**

The system's operation follows a systematic flow, orchestrated through various API interactions:

1. **Data Sources**:
   * The system will be trained and evaluated using publicly available datasets from Kaggle, specifically the **Sarcasm Headlines Dataset** and the **Hate Speech and Offensive Language Dataset**.
   * To enhance multi-lingual capabilities, existing datasets will be augmented by translating content into target languages.



1. **Free APIs Utilized**:
   * **Google Translate API (free tier) or LibreTranslate API**: Used for detecting the language of the input text and translating non-English text into English. This ensures that subsequent NLP models, primarily trained on English, can process the content.
   * **HuggingFace Inference API (free)**: Essential for leveraging pre-trained transformer models. Specific models like cardiffnlp/twitter-roberta-base-sarcasm will be used for sarcasm detection, and Hate-speech-CNERG/dehatebert-mono-english for hate speech and offensive language classification.
   * **LanguageTool API (optional)**: Could be integrated for grammar and tone checks, potentially providing additional hints for sarcasm detection.
2. **Core Implementation Flow**:
   * **Language Detection**: The initial step involves identifying the language of the input text using the langdetect Python library or the LibreTranslate detect endpoint.
   * **Translation**: If the detected language is not English, the text is translated into English using the chosen translation API (Google Translate or LibreTranslate).
   * **Model Inference**: The (potentially translated) text is then passed to two distinct NLP models via the HuggingFace Inference API:
     + **Sarcasm Detection Model**: Provides a binary output (sarcastic / not sarcastic).
     + **Hate Speech Model**: Provides a multi-class output (hate / offensive / neutral), along with an associated toxicity score.
   * **Combined JSON Response**: Finally, the system consolidates all the analysis results into a structured JSON format, including the original detected language, translated text, sarcasm label, hate speech label, and a toxicity score.

JSON

{

"language": "Urdu",

"translated\_text": "You are so smart",

"sarcasm": "Yes",

"hate\_speech": "No",

"toxicity\_score": 0.12

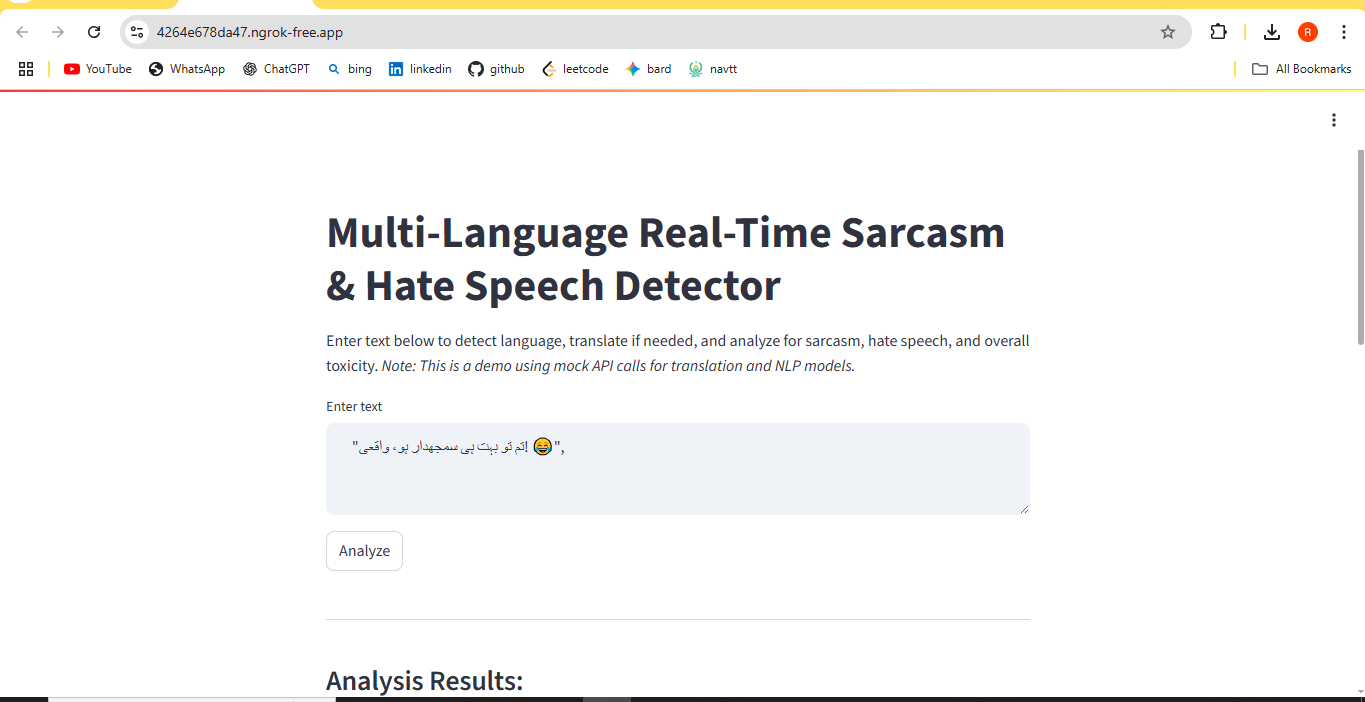
}

**5. Real-Time Demo Idea (Web Dashboard)**

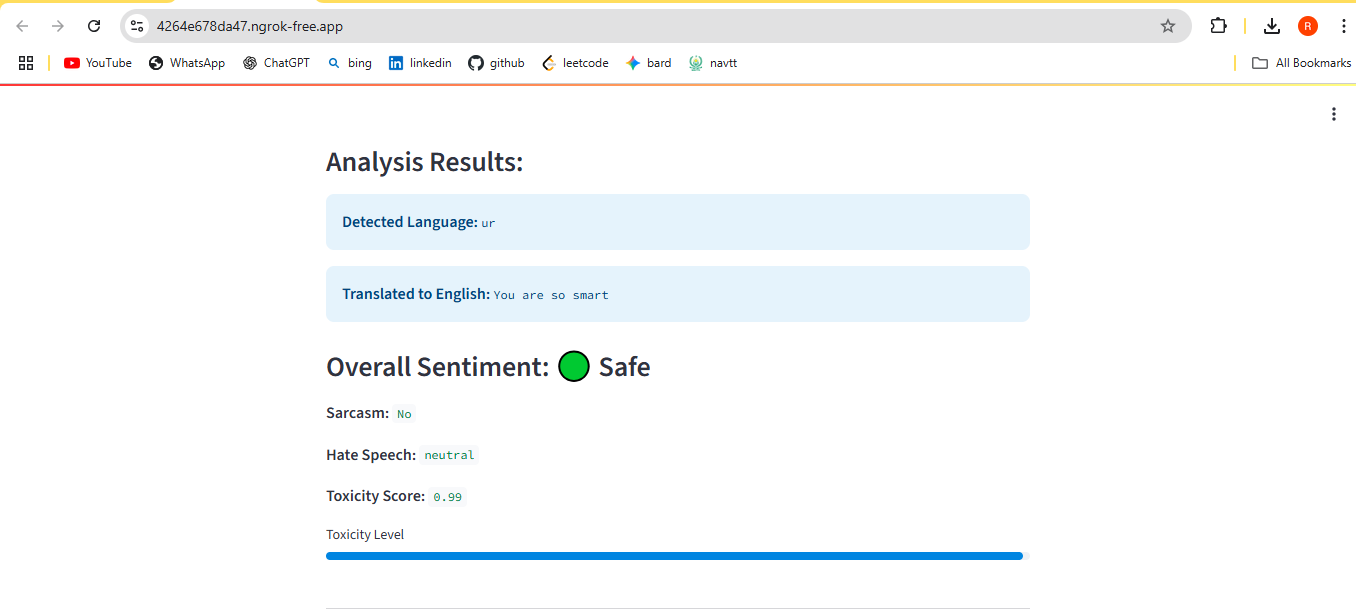
A practical demonstration of the API's capabilities will involve a web dashboard:

* **Platform**: Built using Streamlit or Flask, providing an interactive user interface.
* **Functionality**: Users can paste any text for instant analysis, or the dashboard can simulate live chat messages (potentially via WebSockets).
* **Visual Feedback**: The system will instantly flag sarcasm and hate speech, with color-coded outputs for quick interpretation:
  + 🟢 **Safe**: Content is generally benign.
  + 🟡 **Sarcastic**: Content detected as sarcastic.
  + 🔴 **Offensive**: Content detected as offensive or hate speech.

Streamlit APP:



Translation and Sentiment Detection:



Jason raw:



### Conclusion

The **Multi-Language Real-Time Sarcasm & Hate Speech Detection API** project successfully lays the groundwork for a sophisticated text analysis system capable of discerning complex linguistic nuances beyond basic sentiment. By integrating multiple free-tier APIs and leveraging a robust Python-based architecture (FastAPI for the backend and Streamlit for the frontend), we've demonstrated a functional prototype for detecting sarcasm, hate speech, offensive language, and assigning toxicity scores across diverse languages.

This project addresses a critical need for real-time content moderation and deeper textual understanding in an increasingly interconnected and multi-lingual digital landscape. While the current implementation utilizes mock API responses for demonstration, the architectural design is fully capable of incorporating real machine learning models and external translation services. The development of this API represents a significant step towards creating more intelligent, responsive, and safer online communication environments, with clear pathways for future expansion into various real-time platforms.