Microservice Requirement Document: News Analysis Service

Version: 1.0
Date: 22-08-2025
Author: Aditya Pal

1. Introduction & Purpose

This document outlines the requirements for the **News Analysis Service**. This service is designed to continuously monitor and analyze online news articles from a wide range of national, regional, and hyper-local Indian media outlets.

The primary purpose of this microservice is to automatically identify and extract key entities—such as names of people, locations, organizations, and vehicle numbers—from news reports. It will correlate this information with subjects of interest in active investigations, providing alerts on relevant news developments. The service will be built using **Python**, the **FastAPI** framework, and will leverage Natural Language Processing (NLP) libraries.

2. Functional Requirements

2.1. Core Functionalities

- **News Aggregation:** The service must continuously scrape and aggregate news articles from a configurable list of online news sources (RSS feeds and direct website scraping).
- **Entity Recognition (NER):** The service must process the text of each article to identify and extract the following entities:
  - **Persons:** Names of individuals.
  - **Locations:** Cities, districts, states, specific landmarks.
  - **Organizations:** Company names, government bodies, political groups.
  - **Vehicle Numbers:** Indian license plate numbers.
  - **Phone Numbers:** 10-digit mobile numbers.
- **Keyword Monitoring:** The service will maintain a list of keywords (e.g., a subject's name, a vehicle number) for each active investigation.
- **Alerting:** When an article contains one or more keywords from an active investigation, the service must flag it as a "hit" and trigger an alert to the main application.

2.2. Required Output Data Points

When a relevant article is found, the service must return a structured JSON object.

- article_url: The direct URL to the source news article.
- source_name: The name of the news outlet (e.g., "Dainik Bhaskar").
- publication_date: The date the article was published.

- article_title: The headline of the article.
- matched_keywords: A list of the specific keywords from an investigation that were found in the article.
- extracted_entities: A JSON object containing lists of all entities found in the text.
  ```
  {
    "persons": ["Ramesh Kumar", "Sita Sharma"],
    "locations": ["Bhopal", "MP Nagar"],
    "organizations": ["Madhya Pradesh Police"],
    "vehicle_numbers": ["MP04AB1234"]
  }
  ```
- content_snippet: A 2-3 sentence snippet of the article text where the primary keyword was found.

3. API Endpoints Specification

3.1. POST /monitor/keywords (Manage Monitoring)

Adds, updates, or removes keywords associated with an investigation.

- **Method:** POST
- **Request Body:**
  ```
  {
    "investigation_id": "CASE-FILE-001",
    "action": "update", // "add", "update", "remove"
    "keywords": ["Ramesh Kumar", "MP04AB1234", "9876543210"]
  }
  ```
- **Success Response (200 - OK):**
  ```
  {
    "status": "success",
    "message": "Keyword list for investigation CASE-FILE-001 updated."
  }
  ```

3.2. GET /results/{investigation_id}

Retrieves all news "hits" for a specific investigation.

- **Method:** GET
- **Description:** Fetches a list of all articles that have matched the keywords for the given investigation ID.
- Success Response (200 - OK):
  A list of result objects, as defined in section 2.2.

4. Non-Functional Requirements

- **Security:** API must be secured with an API key (X-API-KEY header).
- **Performance:** The service should be able to process several hundred articles per minute. The NLP processing is resource-intensive and should be managed by a scalable worker pool.
- **Scalability:** The service must be containerized (Docker). The number of scraping and processing workers should be scalable based on the number of news sources and keywords being monitored.
- **Reliability:** The service must be resilient to changes in website layouts. A monitoring system should be in place to detect scraper failures.
- **Logging:** Log all scraping activities, NLP processing tasks, and keyword matches for auditing purposes.

5. Technology Stack

- **Language:** Python 3.9+
- **Framework:** FastAPI
- **Asynchronous Tasks:** Celery with Redis
- **Web Scraping:** BeautifulSoup4, Scrapy
- **News Aggregation:** feedparser for RSS feeds.
- **Natural Language Processing (NLP):** spaCy or NLTK for Named Entity Recognition (NER). Custom rules may be needed for Indian contexts (e.g., vehicle numbers).
- **Deployment:** Docker

6. Annexure 1: News Sources

National (English)

- The Times of India
- Hindustan Times
- The Indian Express
- The Hindu
- NDTV
- India Today

National (Hindi)

- Dainik Jagran
- Dainik Bhaskar
- Amar Ujala
- Navbharat Times

Madhya Pradesh (Bhopal Focus)

- **Dainik Bhaskar (MP):** Very strong local coverage.

- **Nai Dunia:** Another major Hindi daily in the region.
- **Patrika (Madhya Pradesh):** Widespread circulation.
- **The Hitavada:** An English daily with a strong presence in Central India.
- **MP Breaking News:** A local digital news portal.

Maharashtra (Mumbai/Pune Focus)

- **Lokmat (Marathi):** Leading Marathi newspaper.
- **Sakaal (Marathi):** Strong presence in Pune and Western Maharashtra.
- **Mumbai Mirror:** English tabloid with hyper-local city news.
- **Mid-Day:** English tabloid focused on Mumbai.

Delhi NCR

- **Navbharat Times (Delhi):** Strong local Hindi coverage.
- **Hindustan Times (Delhi edition):** Focus on city-specific news.
- **The Times of India (Delhi edition):** Detailed local reporting.

Uttar Pradesh

- **Amar Ujala (UP):** Extensive network of local editions.
- **Dainik Jagran (UP):** Dominant player in the state.
- **Rashtriya Sahara:** Hindi daily with a strong presence in UP.

South India (Representative)

- **Eenadu (Telugu - Andhra Pradesh/Telangana):** Largest Telugu daily.
- **Dinamalar (Tamil - Tamil Nadu):** Prominent Tamil newspaper.
- **Manorama Online (Malayalam - Kerala):** Online portal for Malayala Manorama.
- **Prajavani (Kannada - Karnataka):** Leading Kannada daily.