Software Requirements Specification (SRS)

Sentiment Analysis and Summarization of Online News Articles

Team Members:

Name	ID
Israt Tabassum Kochy	2252421055
Rifah Zakiah	2252421024
Raiyan Bin Sarwar	2252421096
Md. Rajaya Rabby Ador	2252421100
Faizah Mehnaz	2252421080
Ariyana Rubaiya Chowdhury	2252421116

1. Introduction

1.1 Purpose

The purpose of this document is to specify the requirements for a Python-based software application that accepts an online news/blog article URL, scrapes the article content, analyzes its sentiment, and generates a concise summary. This SRS is intended for developers, testers, and stakeholders involved in the project.

1.2 Scope

The project will deliver a command-line or simple GUI tool that:

- Accepts a URL of a news or blog article from the user.
- Extracts (scrapes) the main textual content of the article.
- Cleans and preprocesses the text.
- Performs sentiment analysis (positive, negative, neutral).
- Generates a summary of the article.
- Displays both results to the user.
  Optionally, the system may include result visualization and a simple user interface.

1.3 Definitions, Acronyms, and Abbreviations

- **SRS**: Software Requirements Specification
- **NLP**: Natural Language Processing
- **GUI**: Graphical User Interface
- **API**: Application Programming Interface

2. Overall Description

2.1 Product Perspective

This is a standalone software application or script, initiated and operated by an end-user on their local machine. It may optionally include a minimal web interface via frameworks like Streamlit or Flask.

2.2 Product Functions

- The user inputs a URL.
- The system downloads and parses the web page to extract the article text.
- The system preprocesses and cleans the article text.
- The system performs sentiment analysis using either simple or advanced models.
- The system generates a textual summary of the article.
- System outputs (displays) sentiment label/score and the summary.
- (Optional) System visualizes analysis results (e.g., sentiment distribution).

2.3 User Classes and Characteristics

- **End-users:** Individuals seeking quick insights into the sentiment and essence of news articles; no prior technical expertise required.

2.4 Operating Environment

- Local machine (Windows, macOS, or Linux)
- Requires Python 3.7+ and package dependencies (requests, beautifulsoup4, textblob, nltk, transformers, etc.)

2.5 Constraints

- The software depends on the availability and accessibility of the target URL.
- Website structures may vary and require periodic adjustments to the scraping logic.

- Internet access is required for downloading articles and accessing cloud-based inference (if using Hugging Face API/models).

2.6 Assumptions and Dependencies

- The user will have access to Python and the ability to install necessary packages.
- News/blog sites are not protected by paywalls, CAPTCHA, or dynamic content loading, requiring advanced scraping techniques.

3. Specific Requirements

3.1 Functional Requirements

ID	Requirement Description
FR-1	The system shall prompt the user to input a URL of a news or blog article.
FR-2	The system shall fetch the web page content via HTTP requests.
FR-3	The system shall parse the HTML to extract the main article text.
FR-4	The system shall clean and preprocess the extracted text (lowercase, remove stopwords).
FR-5	The system shall analyze the sentiment of the cleaned text (positive/negative/neutral).
FR-6	The system shall generate a textual summary of the article using NLP techniques.
FR-7	The system shall display the sentiment label/score and summary to the user.
FR-8	(Optional) The system shall visualize sentiment results.
FR-9	The system shall handle exceptions (invalid URLs, inaccessible pages, unexpected HTML).
FR-10	The system shall log or print errors and instructions clearly for users.

3.2 Non-Functional Requirements

ID	Requirement Description
NFR-1	The solution should be usable with basic computer and Python skills.
NFR-2	The system should respond within 10 seconds per analysis under normal conditions.
NFR-3	The code should be well-documented, with instructions for setup and usage in a README file.
NFR-4	The system must not distribute or share user data or scraped content externally.
NFR-5	The application should be modular, with distinct units for scraping, processing, and analysis.

3.3 External Interface Requirements

- **User Interface:**
    - CLI for basic implementation.
    - (Optional) Simple web UI via Streamlit/Flask.
- **Software Interfaces:**
    - Requests to external websites for scraping.
    - Hugging Face Transformers library/models for NLP tasks.
- **Hardware Interfaces:**
    - Runs on standard hardware (laptop/desktop).

4. System Features

4.1 User Input

- Asks the user for a URL via command-line prompt or simple form.

4.2 Text Extraction/Scraping

- Fetches and parses HTML to extract main article text using tag/class selectors.

4.3 Processing and Analysis

- Cleans text (lowercase, removes HTML tags, extraneous whitespace, punctuation, and stopwords).
- Analyzes sentiment.
- Generates a summary (extractive/abstractive).

4.4 Output

- Displays sentiment analysis (label and score/confidence).
- Prints or shows a generated summary.
- (Optional) Visualizes results with simple charts or graphs.

4.5 Error Handling

- Alerts user if URL is invalid, site not accessible, or no main article text is found.

5. Other Requirements

- **Documentation:** Usage instructions and troubleshooting guide shall be provided.
- **Extensibility:** The system should allow for easy replacement of underlying NLP models or extension to new sites.
- **Legal/Ethical:** Must respect robots.txt, website terms of service, and avoid misuse.