Abdulkadar Kanchwala
Roll No 31031524012
Class Msc Cs Part 1

News Summarization and Sentiment Analysis Documentation

1. Introduction & Objective

Objective

The **News Summarization and Sentiment Analysis** project is an NLP-based application designed to automate the process of extracting essential insights from news articles. In the digital era, individuals are overwhelmed with excessive information from multiple news sources. Manually reading and summarizing articles to extract key points can be time-consuming and inefficient.

This project addresses this challenge by leveraging **Natural Language Processing (NLP)** techniques to **summarize news articles** and **analyze their sentiment**. By extracting relevant content, the system provides users with concise summaries and sentiment insights, allowing them to grasp the core message of an article quickly.

The solution offers:

- **Automated news summarization** using text processing techniques.
- **Sentiment analysis** to classify news as Positive, Negative, or Neutral.
- **A simple and interactive GUI** using Tkinter for user-friendly interaction.
- **Excel logging functionality** to store and retrieve summarized news data for future reference.

2. Problem Statement

Challenges in the Current News Consumption Market:

Information Overload:

- Readers are bombarded with countless news articles daily, making it difficult to process and absorb key information.
- The need for quick decision-making is hindered by the vast amount of text that users must read and comprehend.

Lack of Efficient Summarization Tools:

- Existing solutions require manual effort or rely on external websites.
- Many available tools do not offer direct sentiment analysis, requiring additional tools for deeper insights.

Subjectivity in News Reporting:

- News articles can be influenced by biased reporting, leading to skewed interpretations.
- Users need an unbiased sentiment analysis tool to evaluate the tone of articles.

Time Constraints:

- Professionals and researchers often have limited time to read lengthy articles.
- A quick summarization and sentiment classification system can help in making informed decisions efficiently.

3. Dataset Overview

The dataset is dynamically generated as users provide URLs to news articles. It is stored in an Excel file (`summaries.xlsx`) and consists of the following columns:

- **Timestamp**: Date and time when the summary was generated.
- **Title**: Extracted title of the news article.
- **Summary**: The auto-generated summary of the article.
- **Polarity**: Sentiment polarity score from sentiment analysis.
- **Sentiment**: Classification of sentiment (Positive, Negative, Neutral).
- **URL**: Original source of the news article.

4. System Overview

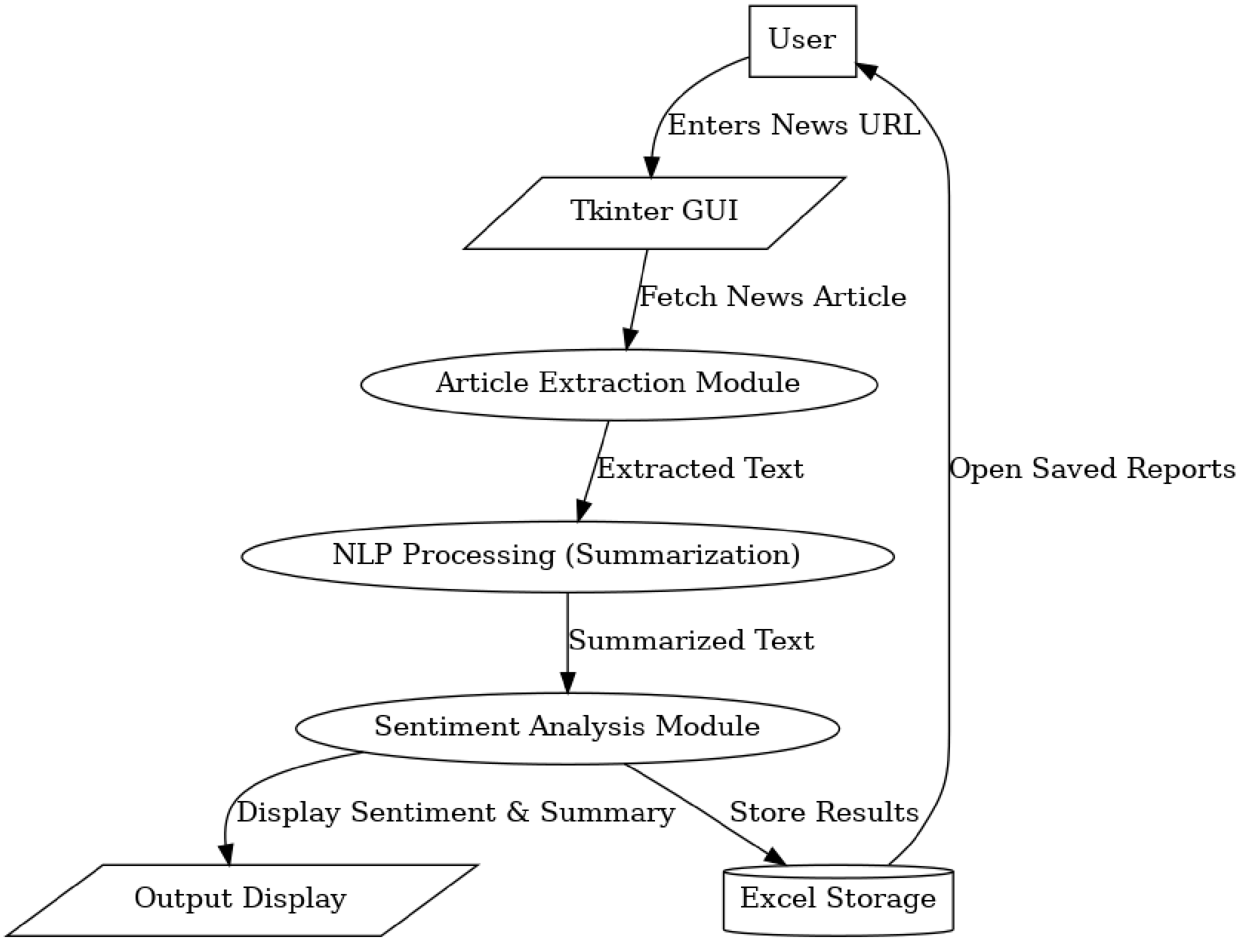
The system consists of three main components:

1. **User Interface (Tkinter GUI):** Allows users to input article URLs and view results.

2. **NLP Processing Layer:** Handles text extraction, summarization, and sentiment analysis.

3. **Data Storage Layer:** Saves summaries and analysis results in an Excel file.

4.1 Architecture Diagram



The architecture diagram illustrates the interactions between components, ensuring a structured and efficient flow of data.

5. Technology Stack

The project leverages multiple tools and frameworks:

- **Programming Language:** Python
- **GUI Framework:** Tkinter
- **NLP Libraries:**
    - `nltk` (Natural Language Toolkit) for text processing
    - `newspaper3k` for web article extraction
    - `vaderSentiment` for sentiment analysis
- **Data Storage:** Pandas (Excel file for logging summaries)
- **Other Libraries:** OS (file operations), Datetime (timestamp management)

6. Detailed Workflow & Implementation

6.1 User Input & Data Preprocessing

- The user enters a news article URL into the interface.
- The URL is validated before processing.
- The article is fetched, parsed, and cleaned using `newspaper3k`.

6.2 NLP Model - Summarization & Sentiment Analysis

- The `newspaper3k` library applies **natural language processing (NLP) techniques** to extract the article text and generate a **summary**.
- The `vaderSentiment` tool performs **sentiment analysis** on the extracted content.
    - **Positive sentiment**: Score ≥ 0.05
    - **Negative sentiment**: Score ≤ -0.05
    - **Neutral sentiment**: Score between -0.05 and 0.05

6.3 Output & Storage

- The summarized text and sentiment analysis results are displayed in the Tkinter interface.
- Data is **logged into an Excel file** (`summaries.xlsx`) for future reference.

7. Code Snippets & Explanations

7.1 Importing Libraries & Reading Data

```
import tkinter as tk
import nltk
import os
import pandas as pd
from datetime import datetime
from newspaper import Article
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

nltk.download('punkt')
```

Explanation:

- The required libraries are imported.
- `nltk.download('punkt')` ensures the Punkt tokenizer is available for text processing.

7.2 News Article Processing

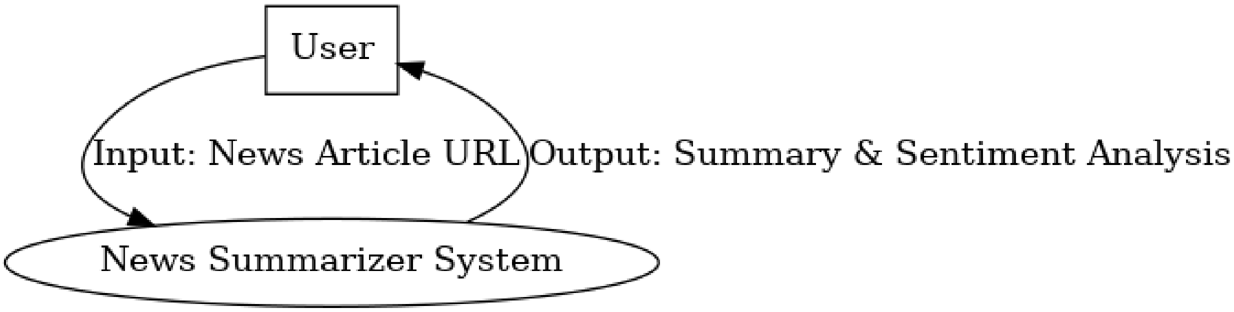
```
def summarize():
    url = utext.get('1.0', "end-1c").strip()
    article = Article(url)

    try:
        article.download()
        article.parse()
        article.nlp()
    except Exception as e:
        summary.insert('1.0', f"Error: {e}")
        return

    title_text = article.title
    summary_text = article.summary

    analyzer = SentimentIntensityAnalyzer()
    sentiment_scores = analyzer.polarity_scores(article.text)
    compound_score = sentiment_scores['compound']
    sentiment_result = "Positive" if compound_score >= 0.05 else "Negative" if compound_score <= -0.05 else "Neutral"

    log_summary_to_excel(title_text, summary_text, compound_score, sentiment_result, url)
```

Explanation:

- The `summarize()` function extracts and processes a news article from a URL.
- `article.download()`, `article.parse()`, and `article.nlp()` fetch and analyze the article using `newspaper3k`.
- Sentiment analysis is performed using `SentimentIntensityAnalyzer()`.
- The extracted title, summary, and sentiment classification are stored in an Excel file for future reference.

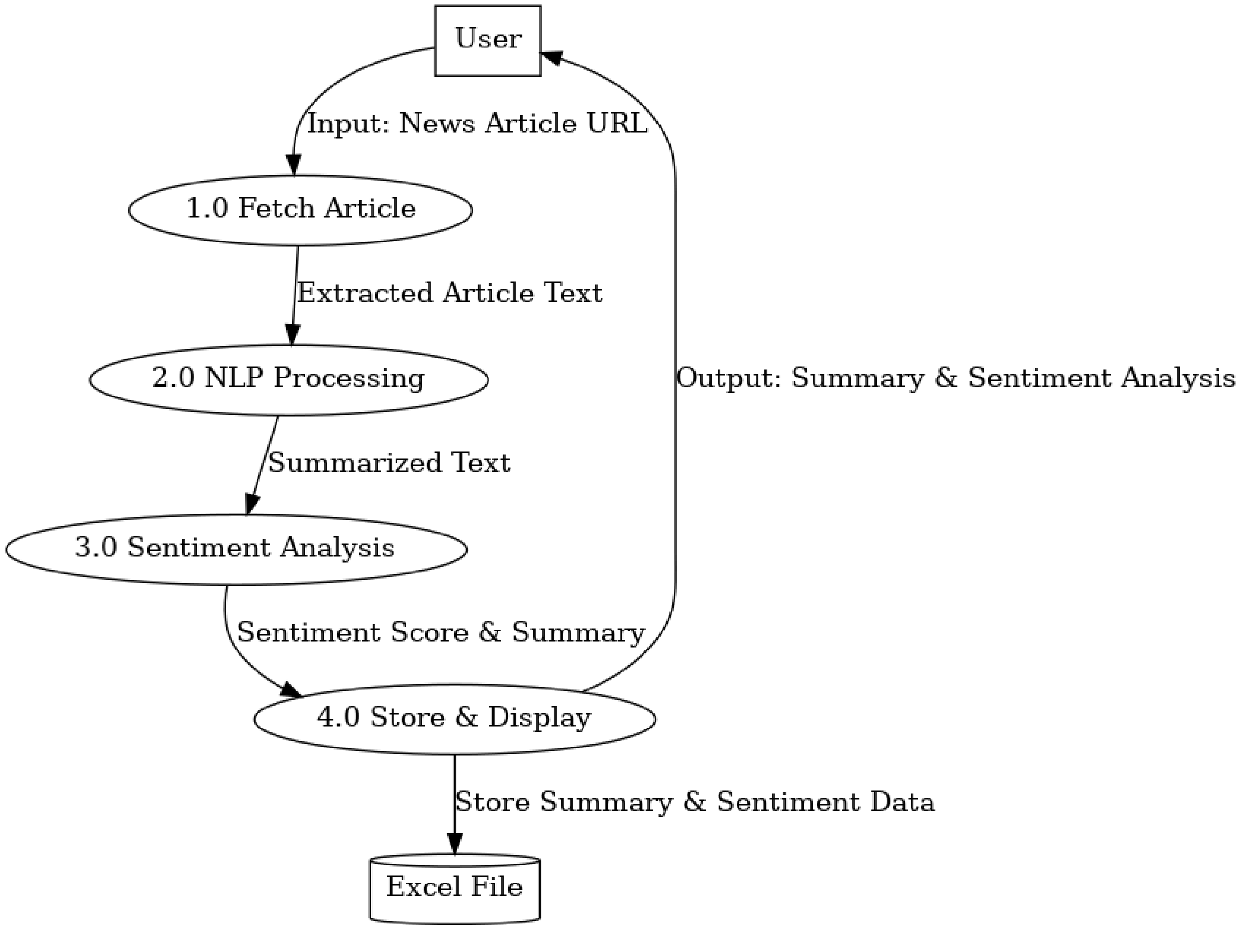
8. Data Flow Diagram (DFD)

8.1 DFD Level 0



This represents the system as a whole and its interactions with users and external systems.

8.2 DFD Level 1 - Detailed Explanation



Explanation of Level 1 DFD:

1. **User Input:**
   - The user enters a URL for a news article.
2. **Preprocessing:**
   - The system validates and fetches the article.
3. **Summarization Module:**
   - Extracts key sentences using NLP techniques.
4. **Sentiment Analysis Module:**

   - Analyzes sentiment using VADER.

5. **Data Storage:**
   - Stores extracted information in an Excel file.
6. **Output Display:**
   - Presents the summary and sentiment classification in the GUI.

9. Future Enhancements

- **Multi-Language Support** for analyzing news in different languages.
- **Integration with APIs** to fetch and summarize news headlines from various sources.
- **Advanced ML Models** such as Transformer-based text summarization for improved accuracy.
- **Graphical Reports** for sentiment trend analysis over time.

10. Conclusion

The **News Summarization and Sentiment Analysis** project demonstrates the power of **NLP** in automating news processing. By providing **quick summaries** and **sentiment insights**, the system helps users stay informed without information overload.