***University***

**Project Report:**

### ***SumNews***

**Course:**

***Natural Language Processing***

**Group Members:**

1. ***Maaz***
2. ***M.Sibtain***
3. ***Abdullah***

### **SumNews**

## **Abstract**

The system aims to automatically collect recent news articles from specified news sites through an interface that utilizes NLP to extract summaries which will be presented in an easy-to-use format. The system runs using Python paired with Flask for backend management and BeautifulSoup for web scraping alongside a TF-IDF-based summarization algorithm. Through the UI interface users can access news content via click-friendly cards showing condensed versions before reading entire articles. Users can achieve more efficient information consumption through this system's ability to transform lengthy articles into brief summaries.

## **Introduction**

Digital news production creates enormous quantities of content instantly every passing minute. Users must spend excessive time browsing through multiple articles to discover relevant and easy-to-understand content. The project creates an automated system which extracts news content from various sources then produces summaries before displaying them using user-friendly layouts. Time efficiency and better user engagement result from presenting essential information immediately to users.

## **Objective**

The primary objectives of this project are:

* The project automates the extraction of news content from multiple Internet news websites.
* The project uses web scraping tools to extract important content from available articles.
* The system produces correct and condensed summaries through implementation of a TF-IDF-based summarization method.
* Through an interactive interface the project will display news summaries alongside complete articles.

## **Methodology**

The system follows a modular approach consisting of the following steps:

1. **Web Scraping:**

* The Requests module retrieves website information which Beautifulsoup extracts into usable data.
* The system extracts important content including headlines followed by article bodies and subsequent links.

1. **Preprocessing:**

* Through the use of nltk the cleaning process removes stop words and punctuation while eliminating unnecessary data.
* The text processing system breaks down sentences before beginning the summarization operations.

1. **Summarization:**

* A TF-IDF algorithm measures sentence importance for subsequent ranking purposes.
* The system uses cosine similarity to prevent redundant content from appearing.
* Generation of meaningful summary depends on the selection of top-ranked sentences from the original text.

1. **Backend API:**

* ·Implemented using Flask.
* The system presents content from scraping operations through JSON formatted endpoints.
* The flask\_cors library permits frontend cross-origin requests from the application.

1. **Frontend Display:**

* Users can view news content organized in card-display formats throughout the app's user interface.
* Users can view the article summary and complete text by clicking on any card.

## **Tools and Technologies**

* **Backend Framework:** Flask (Python)
* **Web Scraping:** BeautifulSoup, requests
* **Summarization:** TF-IDF, NLTK, Scikit-learn
* **Frontend:** HTML/CSS/JavaScript
* **API Integration:** Flask APIs
* **Other Libraries:** numpy, sklearn, nltk, flask\_cors