Training Day 4 Report

Website Summarization:

This script extracts and summarizes text from a website using NLP (Natural Language Processing) techniques**.**

Key Functionality:

1. **Text Extraction (extract_website_text):**
    - Uses requests to fetch a webpage.
    - Parses HTML with BeautifulSoup.
    - Removes <script> and <style> tags.
    - Extracts and cleans visible text.
2. **Text Summarization (summarize_text):**
    - Uses the sumy library's LSA (Latent Semantic Analysis) summarizer.
    - Converts text to sentences using nltk tokenizer.
    - Returns a summary with a default of 5 sentences**.**
3. **Saving Output (save_summary_to_file):**
    - Saves the generated summary to summary.txt.
4. **User Input & Flow (__main__):**
    - Prompts user for a website URL.
    - Extracts and summarizes the site's content.
    - Prints and saves the summary.

🛠️ **Libraries Used:**

- requests – for HTTP requests
- BeautifulSoup – for HTML parsing
- sumy – for text summarization

- nltk – for sentence tokenization

How it works (Internally):

1. **Tokenization**: The input text is split into sentences using nltk's Punkt tokenizer.
2. **Latent Semantic Analysis**: Sumy's LSA algorithm analyzes the semantic meaning of the sentences and selects the most relevant ones.
3. **Text Extraction**: HTML tags are stripped out, leaving only human-readable text.
4. **Summarization**: Only the most important ideas are kept, helping reduce information overload.

Possible Enhancements:

- Add support for keyword extraction or title detection.
- Use other summarization algorithms (e.g., LexRank, TextRank).
- Build a GUI with Tkinter or a web app using Flask.
- Allow user to choose summarization method.
- Export summary to PDF or DOCX.

Website Text Summarizer and Link Extractor:

1. Purpose of the Program

The program is designed to:

- Extract readable text content from a website.
- Summarize the main ideas of that content.
- Extract links (internal and external) from the webpage.
- Save the results in a report file.

This combines the concepts of web scraping, natural language processing (NLP), and URL analysis.

---

2. Concepts Involved:

A. Web Scraping:

- Definition: Web scraping is the process of automatically extracting data from websites.
- Tools Used:
    - requests: To fetch the webpage HTML.
    - BeautifulSoup: To parse and navigate the HTML structure.

Steps in this code:

- The requests.get() function retrieves the HTML content.
- BeautifulSoup parses the HTML, and unnecessary tags like <script> and <style> are removed.
- soup.get_text() is used to extract only visible and meaningful text from the page.

B. Link Extraction

- Every website contains hyperlinks (<a href="..."">) which can be either:
    - Internal: Pointing to the same domain.
    - External: Pointing to a different domain.

Tools & Logic:

- urllib.parse helps:
    - Convert relative links to full absolute URLs using urljoin().
    - Analyze domain names using urlparse().

Process:

- All anchor (<a>) tags are looped.
- Each href is checked:
    - If it shares the same domain → Internal.
    - If not → External.

C. Text Summarization (NLP)

- Definition: Summarization is the process of shortening a set of data computationally to create a summary that retains the essential information.
- Library Used: sumy with the LSA (Latent Semantic Analysis) algorithm.

LSA Summary Process:

- Text is tokenized into sentences using nltk (Natural Language Toolkit).
- Sentences are represented in a high-dimensional vector space.
- LSA identifies relationships between terms and sentences using matrix factorization.
- Top sentences that represent the main content are selected.

D. File Handling

- The save_report() function writes:
    - Internal links
    - External links
    - Summary
      into a text file named website_report.txt.

Concepts Used:

- File I/O (open(), write(), etc.)
- Character encoding (utf-8) to support all types of characters.

---

3. Flow of Execution

1. User enters a website URL.
2. extract_links() function extracts all internal and external links.
3. extract_website_text() retrieves and cleans the visible text.
4. summarize_text() generates a short summary using LSA.

5. All data is displayed on-screen and saved in a report file.

---

4. Key Benefits of This Program

- Automates content and link analysis of any webpage.
- Can be used for:
    - Academic research
    - SEO audits
    - News/article summarization
    - Competitive analysis
- Provides both text insights and structural (link) information.