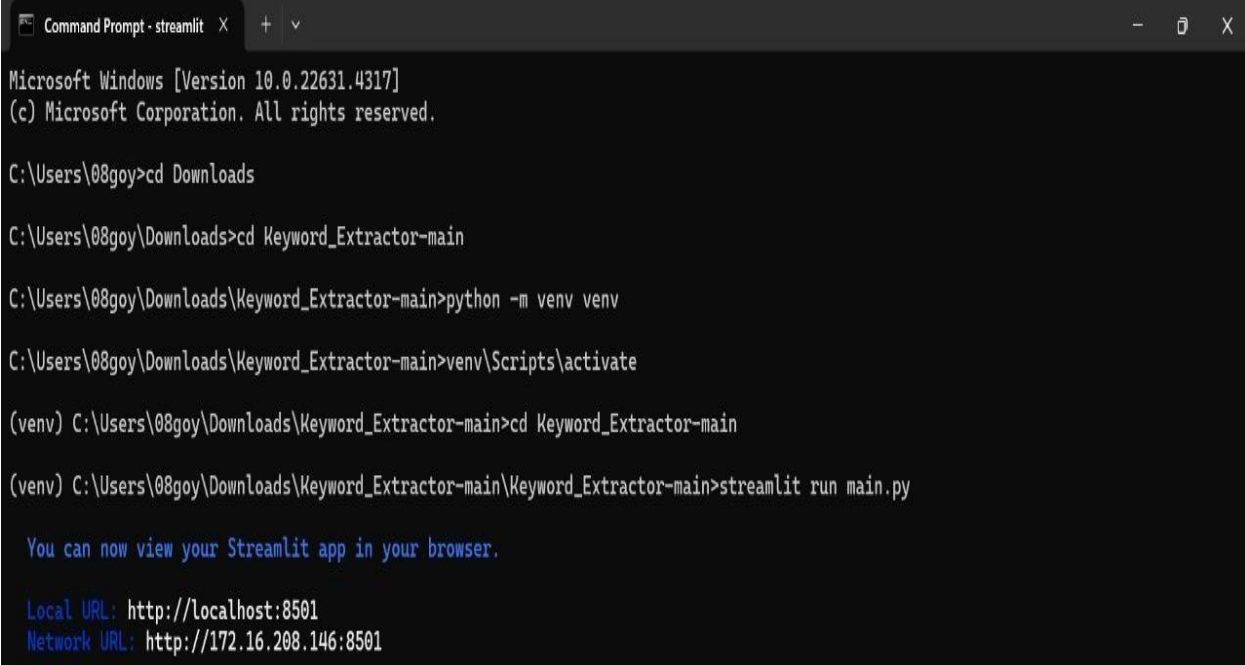


PROJECT DEMONSTRATION



```
Command Prompt - streamlit X + v
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\08goy>cd Downloads

C:\Users\08goy\Downloads>cd Keyword_Extractor-main

C:\Users\08goy\Downloads\Keyword_Extractor-main>python -m venv venv

C:\Users\08goy\Downloads\Keyword_Extractor-main>venv\Scripts\activate

(venv) C:\Users\08goy\Downloads\Keyword_Extractor-main>cd Keyword_Extractor-main

(venv) C:\Users\08goy\Downloads\Keyword_Extractor-main\Keyword_Extractor-main>streamlit run main.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://172.16.208.146:8501
```

The system consists of multiple components working in a structured manner to preprocess input text, extract keywords and key phrases, and generate reports in different formats (PDF or Word). Below is an explanation of its components:

User Interaction: The user starts the process by inputting raw text. The system initiates the workflow based on user-provided input and selected report preferences (e.g., PDF or Word).

Logger: The logger is an integral part of the system, tracking each step of the process. It records activities such as text preprocessing steps (e.g., lowercasing, removing stopwords), keyword/phrase extraction, and report generation. This ensures transparency and helps debug any issues during execution.

Preprocessing Function: The raw input text is passed to the preprocessing function. Here, several operations are performed sequentially, including:

- Converting text to lowercase.

- Removing unnecessary line breaks.

- Eliminating stopwords.

- Applying lemmatization to standardize words. The logger records each step for traceability, and the processed text is then forwarded to subsequent functions.

TextRank Function: This function is responsible for extracting keywords from the preprocessed text using the TextRank algorithm. It returns ranked keywords based on their relevance within the input text. The logger records the process of keyword extraction.

RAKE (Rapid Automatic Keyword Extraction) Function: Parallel to TextRank, the RAKE

function is used for extracting key phrases from the preprocessed text. This algorithm identifies top phrases, typically focusing on multi-word expressions. These key phrases are logged and returned to the system for inclusion in the final output.

Report Generation:

Based on the user's selection, either a PDF Generator or a Word Document Generator is invoked. For PDFs, the system creates the report, logs the PDF creation process, and saves the file. Similarly, for Word documents, the system generates the file, logs the process, and saves the output. The logging here ensures that the report creation is monitored, and any issues can be easily identified.

Completion: After generating the requested report, the system logs the completion of the process, signaling the end of the workflow.

These classes work together to perform text preprocessing, keyword extraction, logging, and exporting processed data. Below is an explanation of the components:

Core Classes:

TextPreprocessor: This class handles the preprocessing of raw text. It takes raw input text and processes it (e.g., lowercasing, removing stopwords, lemmatization). It also includes attributes to store the raw text, processed text, and the language of the text. This ensures flexibility for handling multi-language text processing.

StopWordsHandler: This class manages stopword removal. It stores a list of stopwords and generates filtered text after removing them from the input. It works closely with the TextPreprocessor to streamline the preprocessing pipeline.

Keyword Extraction:

TextRankExtractor: This class implements the TextRank algorithm to identify ranked keywords from the processed text. It stores attributes for the input text, the ID for tracking purposes, and the ranked keywords.

RAKEExtractor: This class performs keyword extraction using the RAKE (Rapid Automatic Keyword Extraction) algorithm. It identifies multi-word keywords and phrases, storing them as extractedKeywords. It operates independently, allowing for comparison or parallel processing with TextRank.

TFIDFExtractor: This class uses the TF-IDF method to score words based on their importance in

the text. It stores the input text, the computed TF-IDF scores, and an ID for identification. This ensures multiple keyword extraction approaches can be utilized for different use cases.

KeywordExtractor: This class acts as a composite, aggregating keywords from various extraction methods (e.g., TextRank, RAKE, TF-IDF). It stores the final list of keywords and text, providing a unified output for further use.

Logging and Exporting:

Logger: The logger is crucial for tracking system operations. It records log levels (e.g., debug, info, error), messages, and timestamps. This helps monitor the system's activities and identify any issues during the workflow.

FileExporter: This class handles the exporting of processed data or reports. It stores details about the file type (e.g., PDF, Word), file path, and export status, ensuring that the final output is saved and logged appropriately.

Steps for Implementation:

1. Set Up the Environment

Open the terminal or command prompt.

Navigate to the project directory using `cd Downloads` and then `cd Keyword_Extractor-main`.

Create a virtual environment for the project:

```
python -m venv venv
```

Activate the virtual environment:

```
venv\Scripts\activate
```

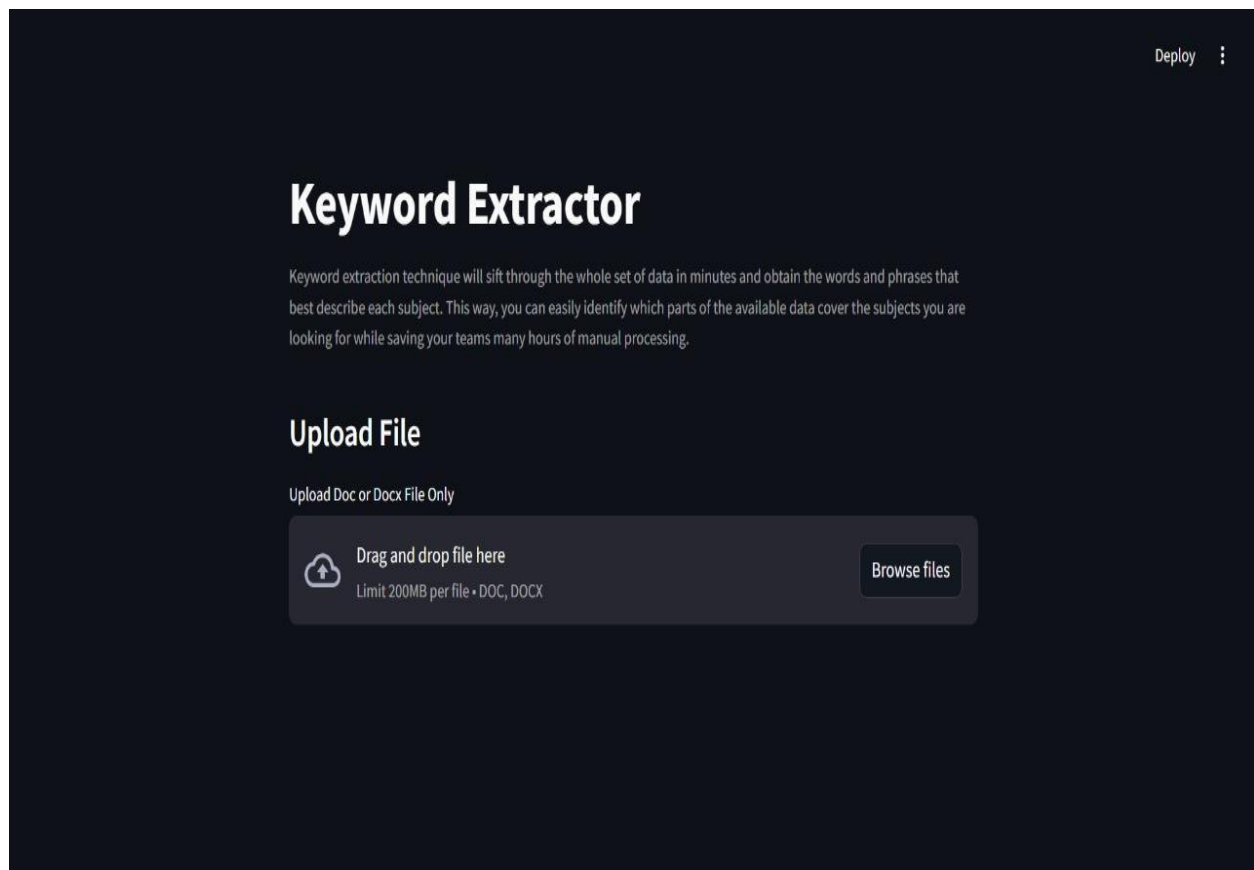
Ensure you are in the project folder before running any commands.

2. Run the Streamlit Application

Inside the active virtual environment, execute the following command to run the Streamlit app:

```
streamlit run main.py
```

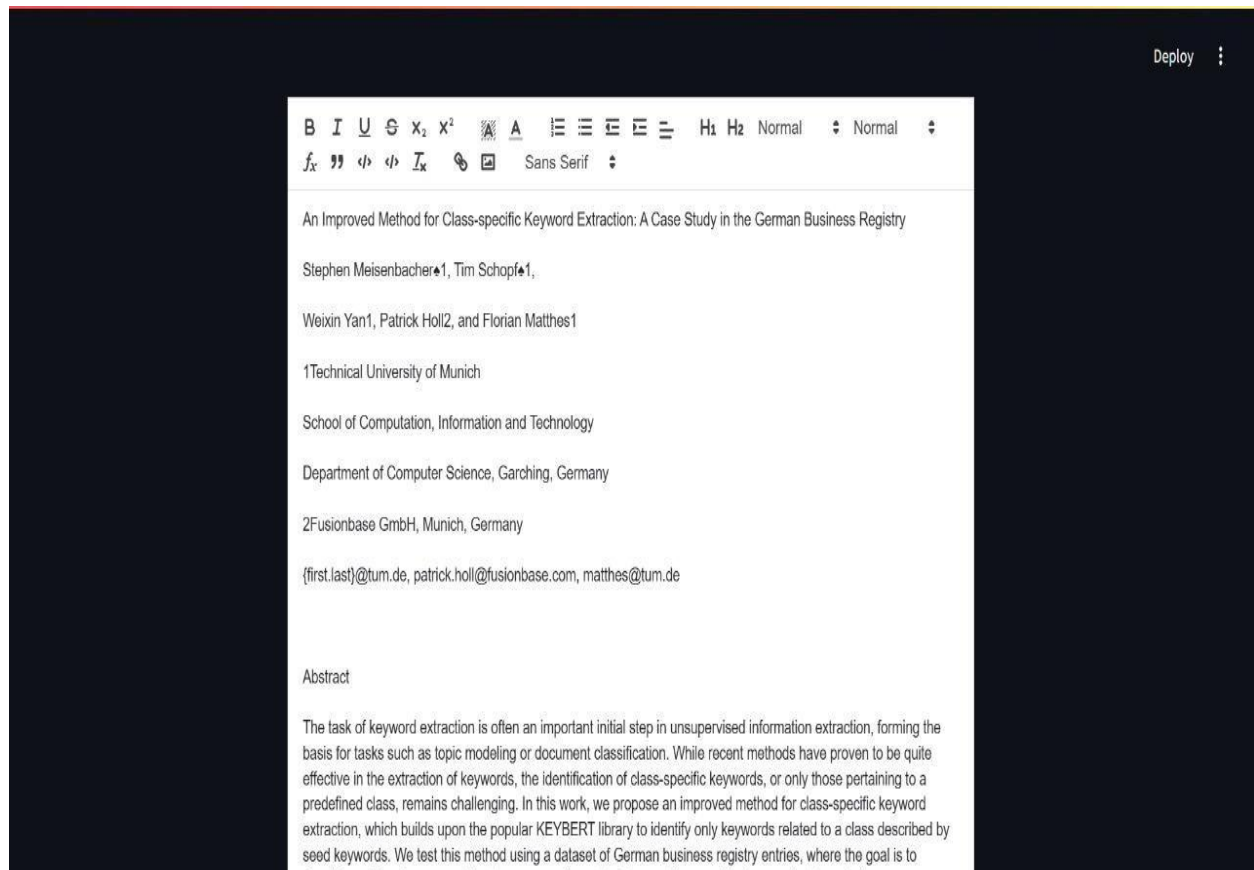
Streamlit will launch a local server. Use the URL provided in the output (`http://localhost:8501`) to access the application in your web browser.



Upload the File

Once the Streamlit application interface opens, you'll see a section to Upload File.

Use the Drag and Drop feature or Browse Files button to upload a .doc or .docx file. Ensure the file size does not exceed the 200MB limit.



Process the Uploaded File

After uploading, the content of the file will be displayed in a text editor format. The application will automatically preprocess the text using methods like:

Lowercasing the text.

Removing stopwords, special characters, and numbers.

Extracting keywords using TextRank, TF-IDF, and RAKE.

save & Extract

☒ Accept Terms & Condition

Download as

☒ PDF

☐ DOC

Export as PDF

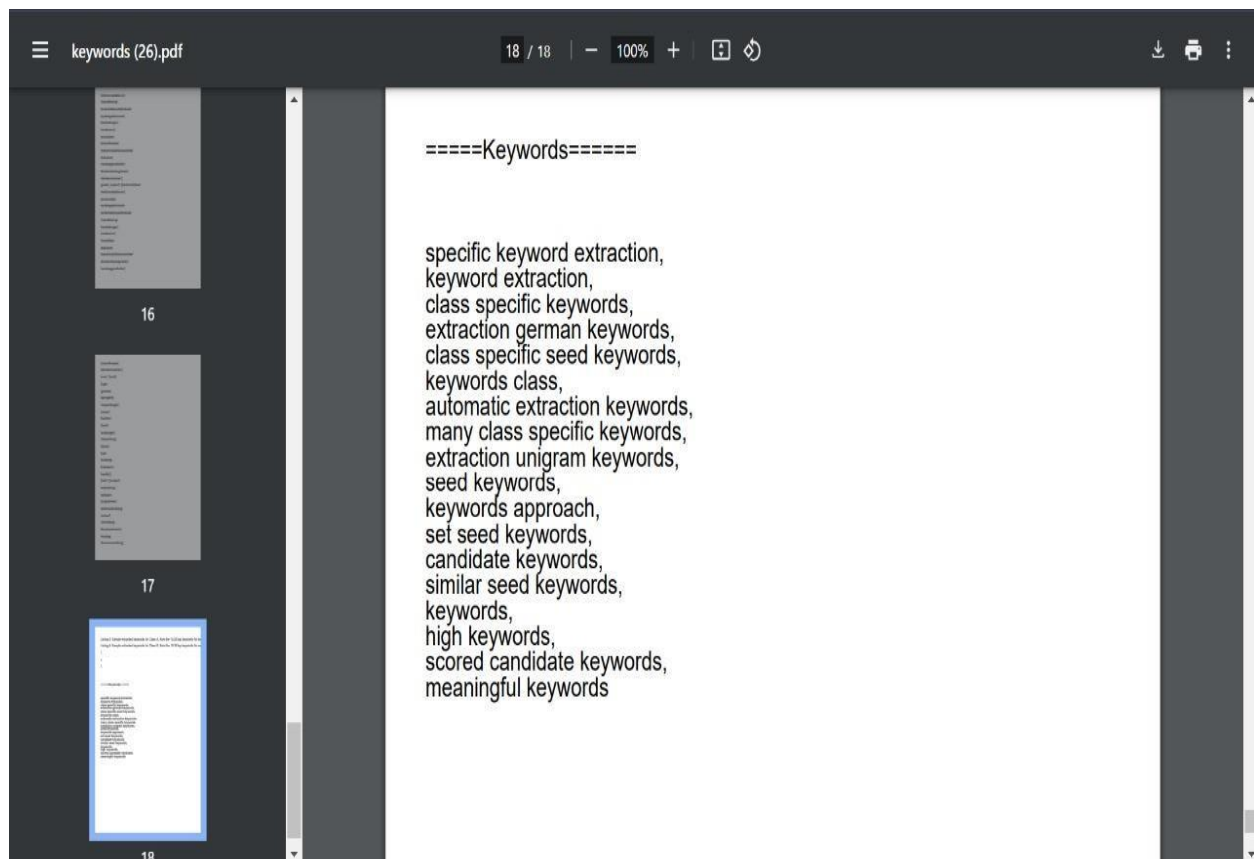
Select File Export Format

Below the text editor, choose the desired export format:

PDF

DOC

Accept the terms and conditions (checkbox) before proceeding.



Export Processed Data

Click on the Export as PDF or the respective button for .doc to download the file.
The generated file will contain the extracted keywords along with their rankings.

