TextRank Keyword Extraction Tool - User Guide

Objective: implement and develop the TextRank algorithm for keyword extraction from documents, generating co-occurrence graphs and optionally extracting keyphrases.

Introduction

TextRank is a powerful tool for extracting keywords from text using the TextRank algorithm, based on the paper "TextRank: Bringing Order into Texts" by Rada Mihalcea and Paul Tarau. This guide will walk you through using both the Windows application and the Python library.

The complete project is available on GitHub at:
https://github.com/alvaro-francisco-gil/text-rank

Installation

The source code is shared in the practice, but it can also be installed directly from GitHub by following these steps:

1. Make sure you have Python 3.6 or higher installed on your system
2. Install the text_rank library:

```
pip install git+https://github.com/alvaro-francisco-gil/text-rank.git
```

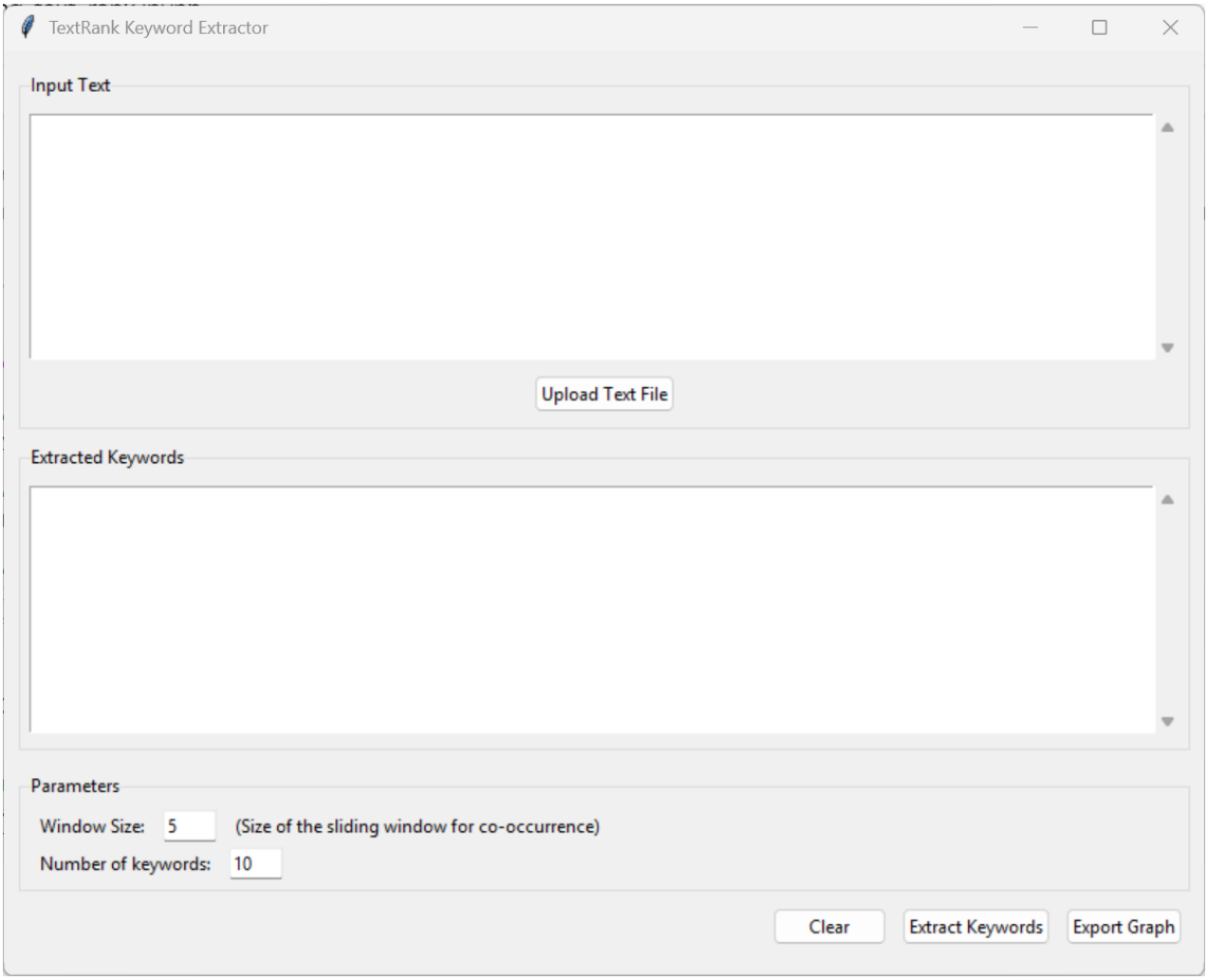
Part 1: Using the Windows Application

The Windows application provides a user-friendly interface for extracting keywords from text without writing any code.

Running the Application

To start the application, **run** the **TextRank_App.exe** file or use the launcher script:

```
python run_text_rank_app.py
```



Using the Application

1. Input Text:
   - Type or paste text directly into the input area, or
   - Click "Upload Text File" to select a text file from your computer
2. Configure Parameters:
   - Set the window size (default is 5) - This controls the size of the sliding window for co-occurrence graph construction
   - Set the number of keywords you want to extract (default is 10)
3. Extract Keywords:
   - Click the "Extract Keywords" button
   - The extracted keywords and their scores will appear in the output area
4. Export Graph:
   - Click the "Export Graph" button to save the co-occurrence graph in Pajek format
   - Choose a location to save the .net file
   - The graph can be visualized using network visualization tools like Pajek, Gephi, or Cytoscape
5. Clear:
   - Click the "Clear" button to reset both input and output areas

Part 2: Using the TextRank Library

For more advanced usage and integration into your own Python applications, you can use the TextRank library directly. This section explains the main components and functionality available in the library.

Main Components

TextRankKeywordExtractor

This is the core class that implements the TextRank algorithm for keyword extraction. It provides the following functionality:

1. Initialization: You can create an extractor with customizable parameters:
    - window_size: Controls the size of the sliding window for co-occurrence graph construction (default: 5)
    - pos_tags: Specifies which parts of speech to consider for keyword extraction (default: nouns and adjectives)
2. Keyword Extraction: The extract_keywords method analyzes text and returns a list of keywords with their scores:
    - Takes a text string as input
    - Optionally accepts a top_n parameter to limit the number of keywords returned
    - Returns a list of (word, score) tuples, sorted by score in descending order

3. Graph Construction: The build_cooccurrence_graph method creates a weighted graph representing word co-occurrences:
    - Takes a text string as input
    - Returns a NetworkX graph object where nodes are words and edges represent co-occurrence relationships
4. Graph Export: The export_pajek method exports the co-occurrence graph to Pajek format:
    - Takes a graph object and a filename as input
    - Creates a .net file that can be visualized with network analysis tools

Utility Functions

The library also provides several utility functions to simplify common tasks:

1. File Handling:
    - read_text_file: Reads text from a file with automatic encoding detection
    - analyze_text_file: Extracts keywords from a text file in one step
    - analyze_multiple_files: Processes multiple text files and returns keywords for each
2. Graph Export:
    - export_graph_to_pajek: Exports a single graph to Pajek format
    - export_multiple_graphs_to_pajek: Exports graphs for multiple files to a specified directory
3. Batch Processing:
    - Functions for processing multiple files in a directory
    - Functions for saving results to CSV or JSON files

Advanced Usage Scenarios

1. Customizing Keyword Extraction:
    - You can adjust the window size to control the granularity of keyword extraction
    - You can specify which parts of speech to consider (e.g., only nouns, or nouns and adjectives)
    - You can set a threshold for minimum word frequency
2. Working with Different Languages:
    - The library supports English by default
    - You can customize stopwords and POS tagging for other languages
3. Visualizing Results:
    - Export graphs to Pajek format for visualization
    - Use network visualization tools to explore word relationships
    - Identify clusters of related terms in your text