

Overview

The **Interactive Text Analysis and Content Improvement Tool** is an AI-powered Python application designed to analyze and improve English text. It provides grammar correction, readability scoring, sentiment analysis, and academic writing enhancement. The tool uses advanced NLP models and libraries to deliver accurate and efficient results.

Modular Design

Definition: Modular design organizes code into independent, reusable modules, each handling a specific task.

Purpose: Enhances maintainability, scalability, and collaboration by separating concerns.

Key Principles:

- **Single Responsibility:** Each module focuses on one function (e.g., text correction or analysis).
- **Reusability:** Modules can be used across the app or other projects.
- **Loose Coupling:** Modules interact via clear interfaces, minimizing dependencies.
- **Encapsulation:** Hides internal details, exposing only necessary functionality.

Implementation:

- Split code into utilities (e.g., constants, model management), processing (e.g., analysis, improvement), and API logic (Flask routes).
 - Use modular functions and classes to ensure each component is reusable and testable.
 - Ensure modules interact through well-defined interfaces to reduce coupling.
-

Modules and Functions

1. `text_analyses.py`

This module contains the core logic for text analysis, readability scoring, sentiment detection, grammar checking, and rule-based improvements.

Key Functions:

- `fix_text(text)`: Attempts to correct grammar and spelling using an AI model. Falls back to rule-based improvements if the AI model is unavailable.
- `rule_based_improve(text)`: Applies rule-based improvements:
 - Capitalizes sentences.
 - Removes weak words (e.g., "very", "really").
 - Replaces informal words with academic alternatives.
 - Eliminates redundant phrases.
- `TextAnalyzer` class:

- `analyze_readability(text)`: Calculates Flesch reading ease, grade level, and assigns a readability level (1–5).
 - `analyze_sentiment(text)`: Uses TextBlob to determine polarity, subjectivity, and overall tone; scores academic style.
 - `analyze_grammar(text)`: Detects repeated words and weak words.
 - `get_statistics(text)`: Returns word, sentence, and character counts, and average words per sentence.
 - `get_word_frequency(text)`: Lists the most common non-stopwords.
 - `comprehensive_analysis(text)`: Runs all analyses and returns a summary.
-

2. `text_improver.py`

This module is responsible for advanced text improvement, including grammar correction and paraphrasing.

Key Functions:

- `load_grammar_model()`: Lazily loads the grammar correction model (`pszemraj/flan-t5-large-grammar-synthesis`). Uses GPU if available, otherwise falls back to CPU.
 - `load_paraphraser_model()`: Would load the paraphrasing model (`Vamsi/T5_Paraphrase_Paws`).
-

How It Works

- **Text Analysis:**
 - The `TextAnalyzer` class processes the input text to calculate readability, sentiment, grammar issues, and statistics.
 - Readability is scored using the Flesch reading ease formula and mapped to a 5-level difficulty scale.
 - Sentiment is analyzed using TextBlob, and academic style is scored based on keywords.
 - **Text Improvement:**
 - Grammar correction is performed using an AI model (`flan-t5-large-grammar-synthesis`) or rule-based methods.
 - Rule-based improvements include removing weak words, replacing informal words, and eliminating redundant phrases.
 - **Lazy Model Loading:**
 - AI models are loaded only when needed to optimize performance and memory usage.
 - GPU acceleration is used if available; otherwise, the tool falls back to CPU.
-

Installation & Running

1. Install dependencies:

```
pip install -r requirements.txt
```

2. Run the server:

```
python start_app.py
```

—

Troubleshooting

- **Model Loading Errors:**

- Check CUDA/GPU setup or set models to run on CPU.
- Reduce text length for processing.

- **High Memory Usage:**

- Disable paraphrasing or use shorter texts.
- Restart the application to clear memory.

- **Slow Response:**

- Use a GPU or reduce text length.