# Text Analyzer with Gemini 2.0 Flash Integration

## Project Summary

The Text Analyzer with Gemini 2.0 Flash Integration is a powerful yet user-friendly desktop application designed to provide insightful and efficient analysis of text. By combining traditional local processing with cutting-edge AI-powered capabilities using Google's Gemini 2.0 Flash API, this application empowers users to analyze, interpret, and extract valuable information from textual content in just a few clicks. Whether you're a writer, student, researcher, or educator, this tool offers a streamlined workflow for understanding language data with minimal setup.

## Project Overview

This application provides a dual-mode text analysis engine that enables users to:

1. Input Text: Users can either type or paste text directly into the interface or import content from a local `.txt` or `.docx` file.

2. Choose Analysis Mode: Select between a fast, offline Local Analysis or a more nuanced AI-Powered Analysis using the Gemini 2.0 Flash API.

3. Configure Analysis Settings: Toggle options such as stopword removal, keyword extraction, word/sentence statistics, and sentiment analysis.

4. Generate Results: Instantly view analytical insights such as word frequencies, readability scores, extracted keywords, tone/sentiment of the content, and more in a clean output panel.

## Core Features

### 🔁 Dual Analysis Modes

- Local Analysis: Fast and lightweight, ideal for basic tasks without the need for internet or API calls.

- AI-Powered Analysis: Leverages Google's Gemini 2.0 Flash API to provide advanced, context-aware insights, sentiment classification, and keyword extraction based on deep natural language understanding.

### 📁 Flexible Input Options

- Supports manual text input.

- Import files with support for `.txt`, `.csv`, and optionally `.docx` (expandable with `python-docx`).

- Handles large texts efficiently with no performance lag.

### ⚙️ Configurable Analysis Options

- Toggle between light and full analysis modes.

- Enable or disable:

- Stopword removal

- Word frequency computation

- Sentence breakdown and stats (average length, count)

- Sentiment scoring (positive/negative/neutral)

- Keyword detection using TF or AI

### 🌐 Gemini API Integration

- Custom prompt templates allow dynamic querying of the Gemini 2.0 Flash API.

- Users can input their own API key for secure and personalized usage.

- Parses and presents JSON responses in a readable format.

### 🧵 Threaded Execution

- All long-running operations (file loading, API communication, heavy processing) run in separate threads to prevent UI blocking or freezing.

### 🧠 Simple Local NLP Engine

- Includes a built-in `SimpleTextProcessor` class with:

- Regex-based tokenization

- Basic word normalization

- Predefined sentiment dictionary (expandable)

- Word and sentence statistics

### 🛠️ Robust Error Handling

- Detects invalid API keys, network errors, and file-related issues.

- Displays user-friendly error messages and guidance prompts.

## Architecture & Key Components

### GeminiClient

Responsible for communicating with Gemini 2.0 Flash API:

- Builds dynamic JSON prompts for various analysis modes (e.g., sentiment, keywords, summary).

- Handles secure HTTPS communication.

- Parses structured JSON responses into displayable output.

### SimpleTextProcessor

Executes offline analysis:

- Tokenizes text and removes stopwords.

- Computes word frequencies using `collections.Counter`.

- Performs rudimentary sentiment analysis using lexicon-based approach.

### TextAnalyzerApp (Main GUI)

- Implements the complete Tkinter-based graphical interface.

- Organizes widgets including input text boxes, dropdown menus, checkboxes, buttons, and results panel.

- Manages user interactions and controls application state.

- Initiates local or API analysis via threads and displays results.

## Potential Use Cases

### ✍️ For Writers

- Analyze tone, keyword density, and structure of their content.

- Improve SEO or tailor content tone based on sentiment feedback.

### 📚 For Students

- Use the tool to study text complexity, sentence variety, and vocabulary usage.

- Perform sentiment analysis of literature passages or essays.

### 🧪 For Researchers

- Conduct preliminary linguistic data analysis.

- Use AI-generated keyword extraction to assist in topic modeling.

### 📢 For Social Media Managers

- Evaluate user-generated content, comments, or posts to gauge sentiment.

- Plan responses or content strategy based on tone analysis.

### 🧑‍🏫 For Educators

- Demonstrate core NLP concepts (tokenization, frequency analysis, sentiment analysis).

- Use it as an educational tool in computer science or linguistics classrooms.

## Planned Enhancements (Future Scope)

- Multilingual Support: Add language detection and support for non-English input.

- Advanced Sentiment Models: Integrate ML-based sentiment scoring.

- Named Entity Recognition (NER): Highlight people, places, organizations.

- Export Options: Save results as PDF, CSV, or HTML report.

- Cloud Save and History: Store analysis logs or allow version comparison.

- Voice-to-Text Input: Incorporate speech input using tools like `speechrecognition`.

## Conclusion

The Text Analyzer with Gemini 2.0 Flash Integration is a versatile, extendable platform for anyone needing fast and accurate text analysis—ranging from basic statistics to intelligent AI insights. It’s built to be efficient, educational, and powerful without being overwhelming. The combination of local processing for speed and AI for intelligence makes it a standout tool in the growing field of natural language understanding applications.

Let me know if you'd like a README.md version, code boilerplate, or even an executable installer (.exe) generator guide using PyInstaller or Auto-py-to-exe.