# InkSight User Manual

**Project Title:** ‘Tell me something I don’t know’: Generating self-insights for creative writers with NLP

**Application Name:** InkSight

**Team:** The Three Musketeers

**Team Members:**

- Micah Crossett (18122181)
- Montaka Khan (21472447)
- Quan Ngoc Minh Vuong (22495054)

# Table of Contents

# 1. Introduction

## 1.1 Overview

InkSight is a text analysis tool that helps you understand your writing. Upload a document and get instant insights about word patterns, distinctive vocabulary, and hidden themes in your text.

**How It Works:** Everything runs locally on your computer. Upload a file (.txt, .docx, or .md up to 5 MB), select what you want to analyze, and click **Analyze**.
InkSight processes your text using a Node.js backend and Python programs powered by the **NLTK** and **FastText** libraries.

**Privacy Features:** All analysis happens locally. No text leaves your computer, no cloud uploads, external servers, data storage, or logging. Once processing is complete, everything is cleared from memory.

**Main Analysis Features**

- **Word Frequency**: See which words appear most often and other general data.
- **Keyness Statistics**: Find words that make your writing unique compared to reference texts (news articles, literature, speeches, or general English).
- **Semantic Clustering**: Discover related words and themes using pretrained AI models.

**Save Your Results**

Download a complete HTML report that opens in any browser.
No account needed. No subscription required.

**Works Offline**

Because all processing happens locally, InkSight can run without an internet connection.

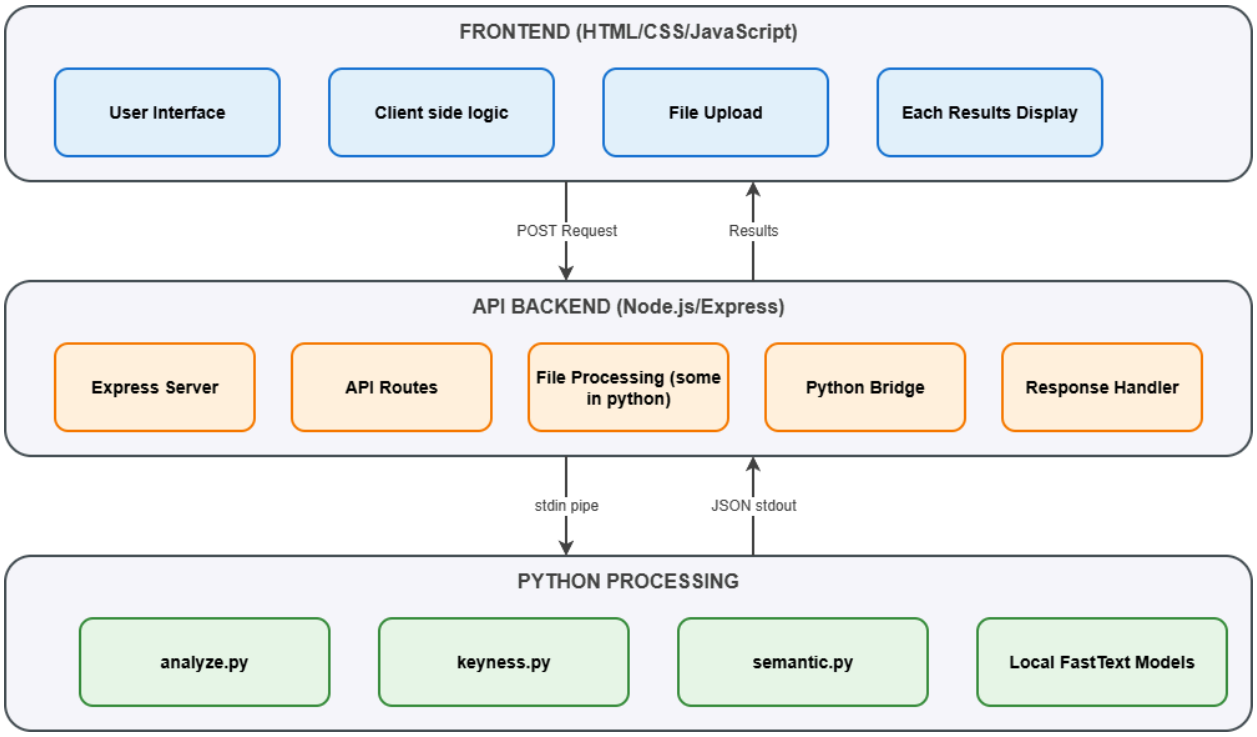
**NOTE - Chart.js Charts uses a CDN currently, so charts would not render offline. This could be fixed simply if needed- although at the cost of more installation overhead in the frontend (npm would need to be packaged there too).

**Unique Application Features:**

- **Responsive & Scalable** - Works on phone with responsive design, ready to be launched as a standalone application
- **Multiple File Format Compatibility** - Supports TXT, DOCX, MD files
- **Secure Processing Design** - In-memory file processing with no files stored on disk - All buffers and streams freed immediately after analysis
- **Word frequency analysis**, **keyness statistics** (with Chart.js visualization), and **semantic clustering**
- **Input Validation** - Robust file input validation/handling
- **Powerful Dashboard**
- **Fully Local Processing** - FastText and NLTK models loaded locally, no external API calls (could be an offline application because of this)
- **Privacy-Focused Design** - Input text never logged to files or console
- **Commercial-License-Free** - NLP processing runs locally in app using open-source models
- **Adaptive/Expandable App Framework** - NLP processing can be easily upgraded to have more features (Such as adding more corpora for keyness statistics)

# 1.2 System Architecture

InkSight architecture is designed for privacy and performance - featuring three layers with distinct responsibilities.




**Frontend:** The Frontend layer presents the user interface in your web browser, handling file uploads and displaying analysis results through client-side JavaScript logic.

**Backend:** When you submit a file for analysis, the API Backend layer (built with Node.js and Express) receives your request, extracts the text from your file in memory, and then uses python scripts to communicate with the python processing layer.

**Python Processing:** The Python Processing layer performs the actual text analysis using three dedicated scripts: analyze.py for word frequency analysis, keyness.py for statistical keyword identification, and semantic.py for semantic clustering using local FastText models.

➔ To summarize, your uploaded file is sent via POST request to the backend, which pipes the extracted text to Python scripts through stdin, and receives analysis results as JSON through stdout, which are then formatted in the backend JS controller, and returned to the front end/browser for display.

## 1.3 Target Audience

InkSight is for writers who want to get better at writing by gaining statistical insights about their own work.

- It gives data-based feedback on things like themes, word choice, and writing habits that people might not notice themselves. This helps them improve their writing and develop their style.
- Popular commercial LLMs provide risk of using your writing content to train or improve their models without your consent, and our tool provides a means to eliminate that concern entirely.

As such, InkSight would be useful for **researchers**, **students**, **professionals** and people who care about privacy and copyright law.

# 2. Quick Start Guide

## 2.1 Minimum Requirements

**Operating System:** Windows
Note: (most features are compatible with mac/linux - aside from some model installation commands, which you can adjust accordingly)

**Software Requirements:**

- Node.js v18 or higher
- npm v9 or higher
- Python v3.11.9
- pip (Latest version)

# 2.2 Installation Guide

**Option 1 (recommended): Automated Setup (Python or NPM Script)**

This option is ideal for users who want to install the application with the default settings and get it running quickly.

Assuming you already have the required dependencies installed, you can choose one of the following methods:

1. *Run the setup command:*
   Open a Command Prompt or PowerShell terminal and enter: "**npm run full-setup-with-fasttext**". Note: for installation of everything but semantic clusters and fastext try: "**npm run full-setup**".

2. *Run the Python setup script:*
   Navigate to api/utils/ and either:
   - Double-click the "**full-installation.py**" file, or
   - Run it manually using your Python interpreter (e.g: "**python "full-installation.py**")

If the installation fails at any point, please refer to Option 2.

**Option 2: Command Line Interface Setup (Fully Manual)**

Please see section 3.1 for detailed step by step instructions on using the CLI.

## 2.3 Running the Application

To run the Application, please follow these steps:

1. Open a terminal in the project directory.
2. Start the server by entering: “**npm run start-api**”.
3. Open your web browser and navigate to: **http://localhost:3000**
4. You should now see the Application homepage.
5. To stop the server: Press **Ctrl + C** in the terminal - or you can just close the terminal.

# 2.4 Using the Application and its Core Features

## Basic Usage

1. **Upload a file**

    - Click the upload area or drag and drop a file (.txt, .docx, or .md).
    - Maximum file size: 5 MB.

2. **Select analysis types**

    - Click the upload area or drag and drop a file (.txt, .docx, or .md).
    - **Word Analysis**: Displays word frequency and basic information
    - **Keyness Statistics**: Identifies distinctive words by comparing to an inbuilt corpus (requires selecting a reference corpus to proceed).
    - **Semantic Analysis**: Groups related words by meaning.

3. **Click the “Analyze Text” button.**

4. **View results**

    - Results appear in cards next to the upload area.

5. **Download your report**

    - Click “Download All” to export a complete HTML report.
    - Or download individual sections from each results card.

6. **Manually clear your data and results**

    - Click “Clear” to clear the results, or simply refresh the page or navigate away.

# 3. Getting Started

## 3.1 Setup

### Pre Installation Requirements

- Node.js v18 or higher
- npm v9 or higher
- Python v3.11.9 (For FastText)
- pip - Python package manager
- Windows (app can run on other operating systems, but not using this installation method)
- 2 GB free disk space

### Full Application Dependencies

*Node.js Packages*

- express: Web server framework
- multer: File upload handling middleware
- mammoth: DOCX file text extraction

*Python Packages*

- nltk: Tokenization and frequency distribution
- statsmodels: Cohen's h effect size calculation
- scipy: Chi-squared statistical testing
- fasttext-wheel: Word embeddings and vector representations
- scikit-learn: KMeans clustering algorithm
- numpy: Array operations and mathematical computations

*NLTK Data Packages*

- punkt_tab: Word tokenization
- brown: Brown Corpus (balanced American English)
- gutenberg: Gutenberg Corpus (classic literature)
- reuters: Reuters Corpus (news articles)
- inaugural: Inaugural Corpus (U.S. Presidential speeches)

*Pre-trained Models*

- FastText Model: cc.en.300.bin: 300-dimension English word embeddings

# 3.2 Installation

- **Please see 2.2 in the Quick Start Guide for Automatic Installation.**
- This section will provide comprehensive step by step guidance on installing the application and running it.

**Step 1: Navigate to the Project Directory**

Open your command line interface and move into the folder where you saved the project. For example:
**C:\Users\YourUsername\Documents\CSE3CAP_Team_InkSight_Project**
Make sure you're inside the project's main directory.

**Step 2: Install Node.js Dependencies**

Next, install all required Node.js packages by running the following commands one at a time:
**npm install**
**cd api**
**npm install**
**cd ..**

These commands install the main and API level dependencies packaged with npm.

**Step 3: Install Python Dependencies**

Install the Python libraries required for text analysis by running:
**pip install -r api/utils/requirements.txt**
This will automatically download and install all necessary Python packages listed in the requirements file.

**Step 4: Download NLTK Data**

Set up the NLTK data needed for tokenization and keyness analysis with this command:
**python api/utils/setup_nltk.py**
This script downloads all required corpora and tokenizers into the environment.

**Step 5: Download the FastText Model / Semantic Clustering Setup**

The semantic clustering feature requires the FastText library and a pre-trained English language model. Follow the steps below to install and configure it properly.

**1. Install FastText**

Clone and install the FastText library using these commands:
**git clone https://github.com/facebookresearch/fastText.git**
**cd fastText**
p**ip install .**

**2. Download the English Model**

Download the pre-trained English model by running:
**python download_model.py en**
This will download a file named **cc.en.300.bin** (967mb approx.).

**3. Move the Model to Your Project Folder**

After the download is complete, move **cc.en.300.bin** into the **models** folder in your project root (you will need to make the models folder for the first installation).

Installation complete.

**Common Issues**

- **Port 3000 in use:** Change the port number in app.js or close the conflicting application using the port.
- **FastText Fails to install:** Ensure Python 3.11.9 is installed and added to your PATH environment variable.
- **FastText download fails:** Manually download the model from https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.en.300.bin.gz and extract it and place it into the models folder in the directory root.

# 3.3 Basic Navigation

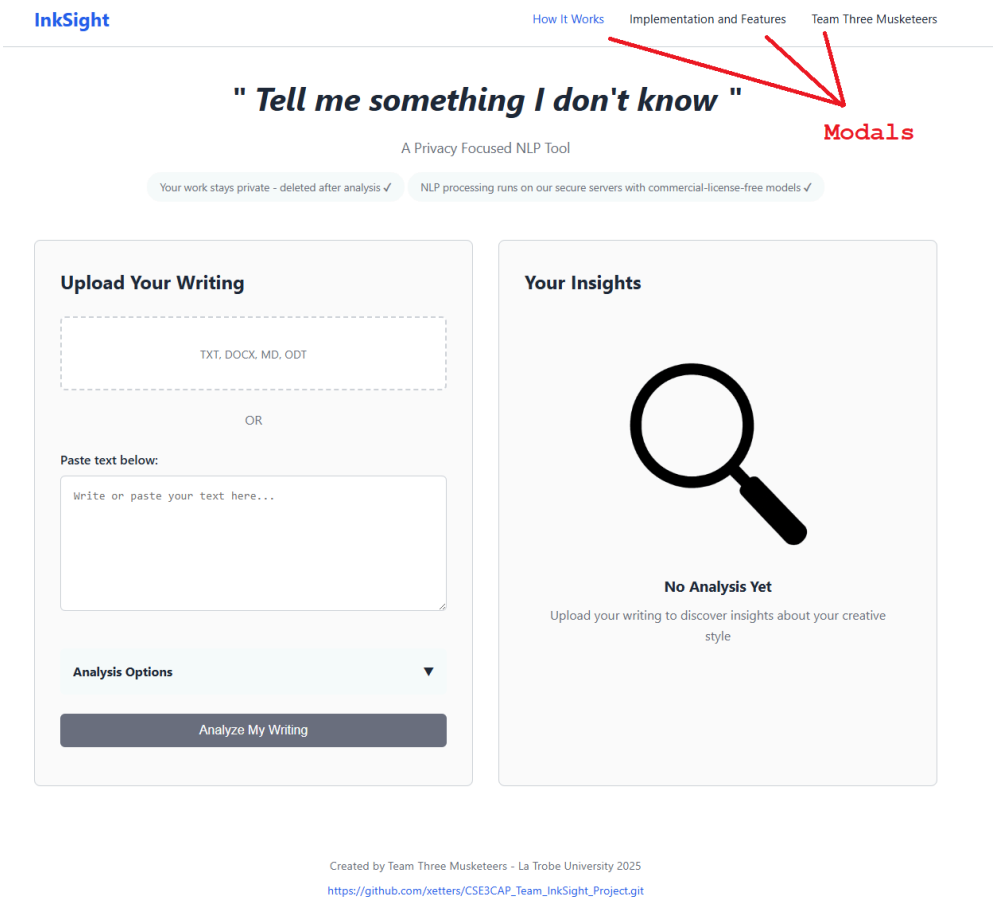
InkSight is a single page web application. All features are displayed on one page. Users can upload files, select analysis options, and view results without reloading or navigating to another page.

Each section of our page is detailed below:

## 3.31 Header Section

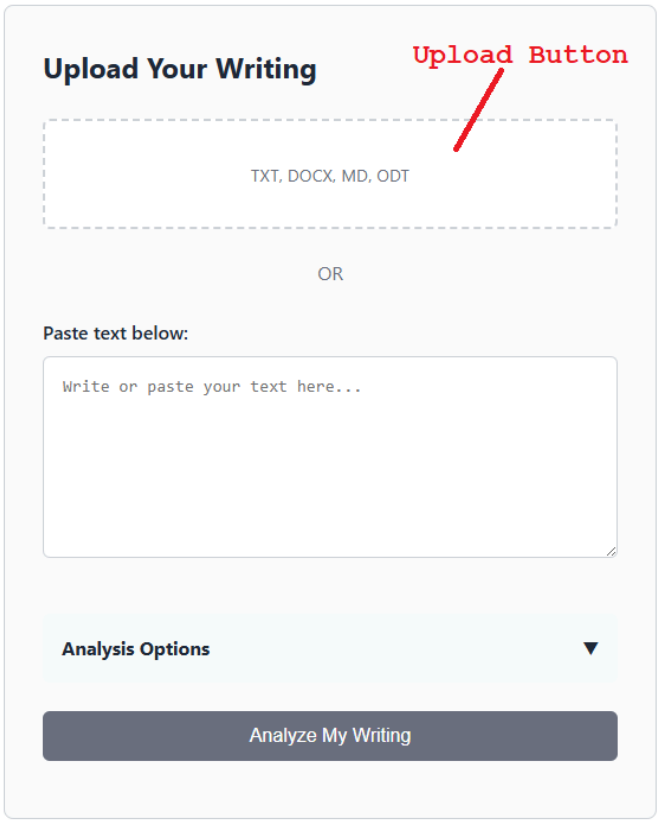
- The header displays the application title and information “links” at the top of the page. These “links” open popup modals, so a user can read while our tool processes.

- To open an informational modal, click the text link labeled **“How It Works”**, **“Implementation and Features”**, or **“Team Three Musketeers”** in the top navigation bar.



### 3.32 File Upload Area

- The file upload area is located at the left of the page (or top under header if on mobile).
- To upload a document, click anywhere inside the rectangular upload box displaying accepted file formats or drag and drop a file into this area.
- Supported file formats: .txt, .docx, .md.
- Maximum file size: 5 MB.
- When a file is uploaded, the filename will display in display highlighted yellow, indicated the browser has finished
- To proceed to analysis see the next section

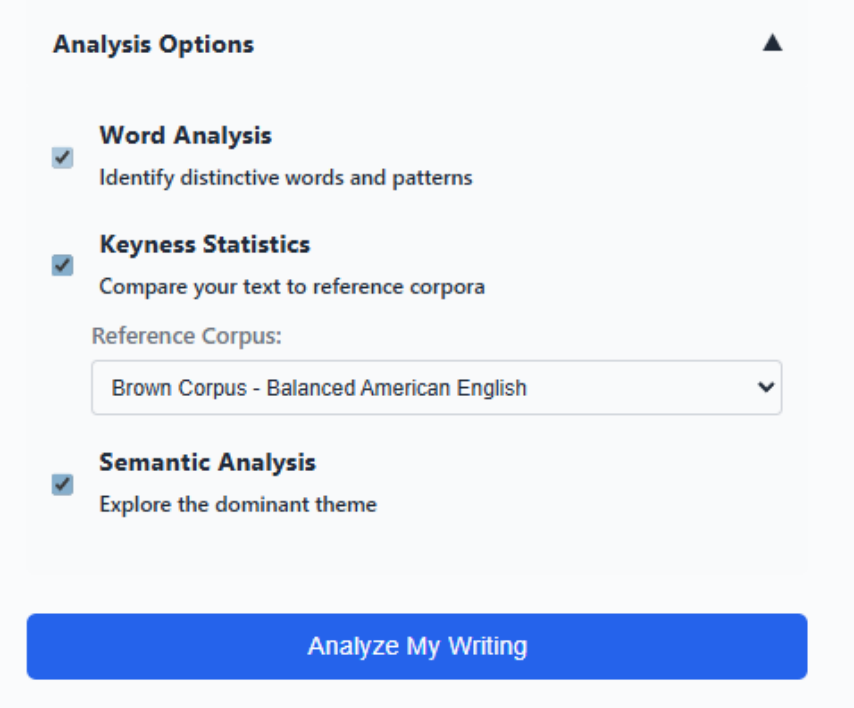


### 3.33 Analysis Options Panel

- The analysis options appear underneath the file upload area after a document is selected (or the text area has text entered in it).

- To select an analysis type: First expand the collapsable menu "**Analysis Options**". Then, click the checkbox next to **"Word Analysis"**, **"Keyness Statistics"**, or **"Semantic Analysis."** Multiple checkboxes can be selected at once.

- To start processing, click the button labeled **"Analyze My Writing"** at the bottom of the upload section.



### 3.34 Results Display Area

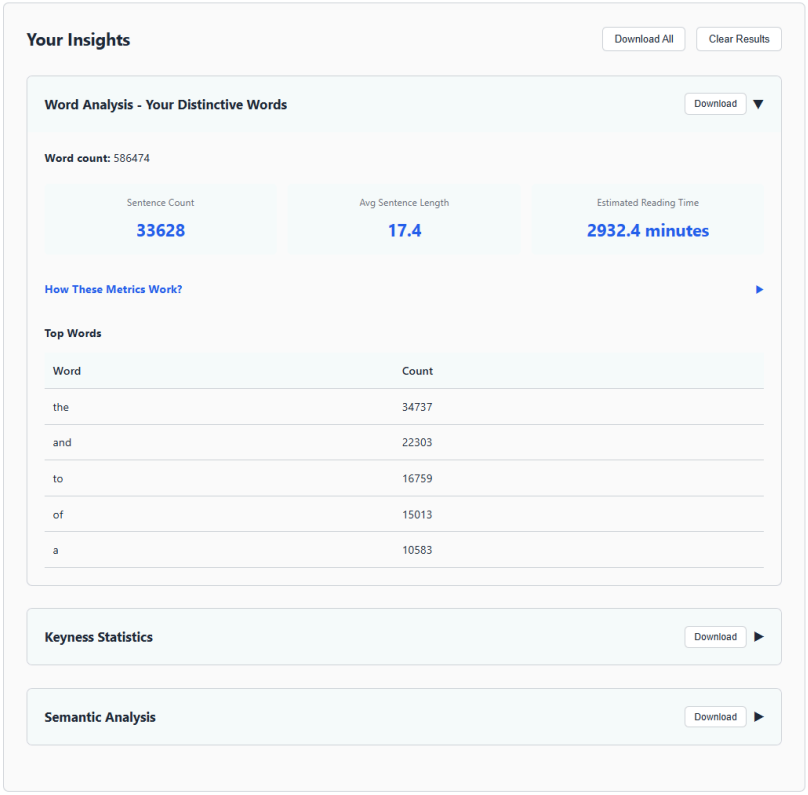
- After analysis completes, results appear in separate cards within the “Your Insights” section.

- Each analysis card corresponds to each one of your selected options:

    1. **Word Analysis:** Displays word counts and frequency data.
    2. **Keyness Statistics:** Shows distinctive words with statistical values.
    3. **Semantic Clusters:** Displays related words grouped by meaning.

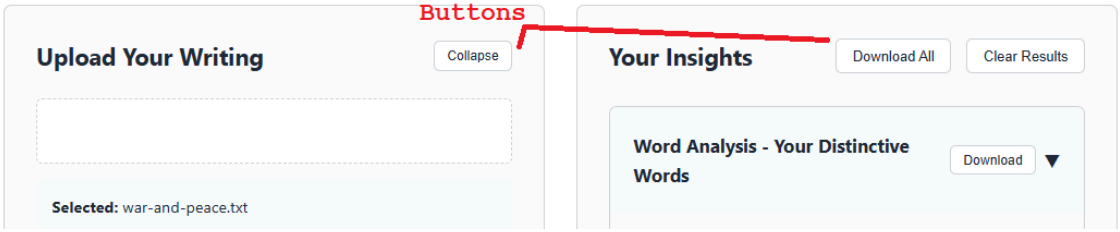
- Each card can be collapsed and expanded by clicking the button in the card

- To view just the results area fullscreen, the upload area can also be collapsed and expanded again - allowing the results area to expand.



### 3.35 Download / Clear Options (Visible above results card after analysis)

- To download one section, click the **“Download”** button at the bottom of the corresponding results card.
- To export all analyses, click **“Download All”** in the **Your Insights** header.
- Downloaded HTML reports can be opened in any web browser or printed as PDF files.




### 3.36 Navigation Notes

- The interface layout fits within a single page for desktop users (results may require scrolling).
- Mobile users can scroll vertically to access all sections.
- The **Collapse** and **Expand** buttons can be used to adjust viewing space between the upload and analysis sections.
- Results are cleared automatically when the page is refreshed.

# 4. User Guide

## 4.1 Core Features

### 4.11 Word Frequency Analysis

Analyzes uploaded text to count total words, most used words, and sentence count. It also calculates the average sentence length and the estimated reading time.

#### When to Use It

- To identify overused words in your writing
- To obtain basic text statistics such as word count and vocabulary size
- To analyze writing habits and repetitive language use
- To determine which terms dominate your writing style

#### How to Use It

- See section 2.4

#### How the Metrics are Calculated

- Sentence Count: Total number of sentences (split by . ! ?)
- Average Sentence Length: Total Words ÷ Sentence Count
- Estimated Reading Time: Total Words ÷ 200 (average reading speed: 200 words/minute)

## 4.12 Keyness Statistics

Compares the uploaded text against reference corpora (Brown, Gutenberg, Reuters, Inaugural) using chi-squared statistical tests. The feature identifies statistically distinctive words and calculates effect sizes to measure how different the vocabulary is from the selected corpus.

### Corpus Details

- Brown Corpus: Balanced general American English (1M+ words from various genres)
- Gutenberg Corpus: Literary works from classic literature (2M+ words)
- Reuters Corpus: News and journalism texts (1.3M+ words)
- Inaugural Corpus: U.S. Presidential inaugural addresses (formal political speech, smallest corpus)

### When to Use It

- To identify what makes your writing style unique
- To compare vocabulary against a specific genre (e.g., news, literature, formal speech)
- To detect characteristic words defining your text
- To analyze how your writing differs from standard English usage

### How to Use It

- See section 2.4

### What You Will See in Results

- Word: The distinctive term identified
- Effect Size: Cohen's h value showing strength of distinctiveness (higher = more distinctive)
- Significance: Statistical reliability indicators (*** p<0.001, ** p<0.01, * p<0.05)
- Visualization: Bar chart displaying effect sizes for top keywords

## 4.13 Semantic Clustering

Groups semantically related words using FastText word embeddings and the KMeans clustering algorithm. The feature reveals recurring patterns and hidden thematic patterns and topic clusters by analyzing contextual similarity among words.

### When to Use It

- To identify main themes and topics in your writing
- To explore relationships among different concepts
- To uncover semantic patterns that are not immediately visible
- To analyze the conceptual structure of your text

### How to Use It

- See section 2.4

### What You Will See in Results

- Total Words Analyzed: Number of words processed during clustering
- Total Clusters: Number of thematic groups identified (typically four)
- Cluster Groups: Lists of semantically related words for each cluster
- Cluster Size: Number of words within each thematic group
- Top Clusters: Largest and most significant thematic groups by size

## 4.14 HTML Export and Download

### What It Does

Generates HTML reports containing all analysis results with embedded CSS styling. Reports can be viewed in any web browser, printed to PDF, or shared without running the Application.
**Note: Charts are not included in the report, but all the data to construct one is included, or this feature could be added later.
**Note: Semantic analysis is also missing features in the downloads section.

### When to Use It

- To save analysis results permanently
- To maintain a portfolio of writing analyses over time
- To print or archive professional analysis reports

### How to Use It

1. See section 2.4 (or 3.2 for greater detail)

# 5. Known Issues / Tips

**Issue #1 (Intended):** Some features, such as keyness statistics, can not accurately produce data without having a good sample size to compare to the corpus (500+ words)

**Issue #2 (Not implemented yet):** Semantic analysis download feature not complete.

**Issue #3:** Express server address hardcoded into api.js (server file). This application is only a development model, but this would need to be addressed for production.

**Issue #4:** When viewing the application via Desktop, it is hard to see the results charts in full. To address this you can collapse the upload section (it will expand the results section to fill).

### Issue #5 - Future Scalability Concerns

Late in development we noticed that the current design creates a new Python process for each user request which could overwhelm server resources if many people use it at once.
Running this on a remote web server would need expensive auto-scaling infrastructure and conflict with the privacy design constraints (local server), while distributing it as a desktop app would keep processing local and avoid both cost and privacy concerns.

Potential solutions:

- rejecting privacy constraint
- adding a queue system for users (would be quite limiting)
- design app to be installed in full locally (would come with reasonable RAM requirements - a new phone or pc would be required)

### Issue #6 - Semantic Cluster Chart

Development for the chart visualization was not fully finished - but it is designed to be ready for easy updates and customization.

# 6. Glossary

## Analysis Terms

| Term | Definition |
|---|---|
| Effect Size | Indicates how much more (or less) frequently a word appears relative to the reference corpus. Higher values indicate greater distinctiveness. |
| Significance Level | Shows the reliability of keyness results |

## Technical Terms

| Term | Definition |
|---|---|
| Chi-squared Test | Statistical test measuring how different observed word frequencies are from those expected in a reference corpus. |
| FastText | A library for learning of word embeddings and text classification created by Facebook's AI Research lab |
| NLTK | Python library used for tokenization, word counting, and access to reference corpora. |
| Tokenization | The process of splitting text into individual tokens (words) for analysis. |

## File Formats

| Format | Details |
|---|---|
| TXT (Plain Text) | Unformatted text; fully supported without extraction. |
| DOCX (Microsoft Word) | Modern Word documents; fully supported with automatic text extraction. |
| MD (Markdown) | Lightweight markup; supported with formatting preserved. |
| ODT (OpenDocument Text) | Open-source format; partially supported — convert to TXT or DOCX for best results. |

# 7. Appendices

## Appendix A: Reference Corpora

| Corpus | Description |
|---|---|
| Brown Corpus | Balanced general American English (~1M words). |
| Gutenberg Corpus | Classic literature (~2M words). |
| Reuters Corpus | News articles (~1.3M words). Useful for journalistic or informational styles. |
| Inaugural Corpus | U.S. Presidential inaugural addresses. Appropriate for formal language. Also the shortest corpus |

# Appendix B: NLP Models and Config

## FastText Model

### Version and Source

Pre-trained Common Crawl model from https://fasttext.cc/docs/en/crawl-vectors.html.

### Model Specifications

- File: “cc.en.300.bin” (300-dimension reduced model)
- Location: models/ directory at project root
- Size: 6.73 GB (300-dim)
- Language: English (Common Crawl + Wikipedia)

### Configuration Options

- MODEL_PATH: “../models/cc.en.300.bin”
- NUM_CLUSTERS: 4
- PCA_COMPONENTS: 10

### Reducing Model Size

Run the following to reduce dimensions and improve performance:

“**cd fastText**”
“**python reduce_model.py cc.en.300.bin cc.en.100.bin 100**”

**NOTE**
Update MODEL_PATH and dimension settings in “semantic.py” after reduction.

## License Information

- NLTK: Apache 2.0 License (approved for commercial use)
- FastText: MIT License (approved for commercial use)
- Python Dependencies: Open source and commercial-friendly
- No restricted or proprietary models are included in the Application.

**End of User Manual**