Stylometric Analysis Tool for Authorship Attribution

Dinh, Tien
pdinh@my.harrisburgu.edu

Fernandes, Kenneth
kfernandes@my.harrisburgu.edu

May 29, 2025

1 Group Members and Assigned Duties

This project was completed by a team of two members: Tien Dinh and Kenneth Fernandes. The components of the stylometric analysis tool were collaboratively designed, implemented, and tested, with responsibilities distributed as follows:

1.1 Tien Dinh

- **System Architecture Design:** Created modular project structure with separate directories for source code, tests, and data
- **Dataset Integration:** Added Gutenberg dataset for comprehensive author sample collection and testing
- **CI/CD Pipeline:** Set up GitHub Actions continuous integration pipeline to automatically run tests on code commits
- **Distance Metric Development:** Implemented weighted distance comparison system for author attribution

1.2 Kenneth Fernandes

- **Enhanced Feature Implementation:** Developed algorithms for two additional stylometric measures (punctuation density and stopword ratio) beyond the five textbook features
- **User Interface Development:** Built both interactive and command-line interfaces for text analysis
- **Code Integration:** Ensured seamless integration of all components and system reliability

1.3 Shared Contributions

- **Testing Framework:** Both team members contributed to creating comprehensive unit tests, integration tests, and error handling validation
- **Documentation:** Collaborative effort in preparing usage instructions, system documentation, and code comments
- **Core Implementation:** Joint implementation of the five fundamental stylometric features from textbook methodologies (average word length, vocabulary diversity, hapax legomena ratio, average sentence length, sentence complexity)

The collaborative approach allowed for efficient division of labor while maintaining consistency in design decisions and implementation standards across all system components.

2 Design Process of Improvements to the Book Code

The stylometric analysis tool represents significant enhancements over basic textbook implementations of authorship attribution systems. Key improvements include:

2.1 Enhanced Feature Set

While basic implementations often focus on simple metrics like word frequency, this system incorporates seven sophisticated stylometric features that capture different aspects of writing style:

- **Vocabulary Diversity:** Uses type-token ratios to measure lexical richness beyond simple word counts
- **Hapax Legomena Analysis:** Identifies words appearing exactly once, providing insights into vocabulary creativity
- **Sentence Complexity:** Analyzes syntactic structure rather than just sentence length
- **Punctuation Density:** Measures punctuation usage patterns, revealing author preferences for specific punctuation styles
- **Stopword Ratio:** Analyzes function word usage, which is often unconscious and highly distinctive for individual authors

2.2 Robust Text Processing

Improvements over basic textbook approaches include:

- UTF-8 encoding support for international texts
- Sophisticated tokenization handling contractions and hyphenated words
- Robust sentence boundary detection with abbreviation handling
- Normalization procedures ensuring consistent feature scaling

2.3 Weighted Distance Metrics

Rather than simple Euclidean distance, the system implements weighted distance calculations allowing different stylometric features to contribute proportionally to their discriminative power.

2.4 Comprehensive Testing Infrastructure

The design includes extensive testing capabilities rarely found in textbook examples:

- Isolated unit testing for individual functions
- Integration testing for complete workflows
- Temporary file handling for safe I/O testing
- Coverage analysis integration
- Automated continuous integration through GitHub Actions pipeline

2.5 Dataset Integration

The system incorporates the Project Gutenberg dataset, providing access to a vast collection of literary works from established authors. This enhancement significantly expands the system's capability for:

- Training with diverse writing styles across different time periods
- Testing attribution accuracy on well-documented authorship cases
- Benchmarking performance against known literary works
- Supporting academic research with standardized datasets

2.6 Modular Architecture

The code organization follows software engineering best practices with clear separation of concerns, making the system maintainable and extensible compared to monolithic textbook implementations.

3 Overview of Code Functionality

3.1 Project Structure

The system follows a modular architecture:

```
authorship_attribution/
    src/                        # Source code
        kenneth/                # Kenneth's implementation
        tien/                   # Tien's implementation
        enhanced/               # Enhanced group implementation
    tests/                      # Test suite
        kenneth/                # Tests for Kenneth's
           implementation
        tien/                   # Tests for Tien's implementation
        enhanced/               # Tests for enhanced
           implementation
    data/
        known_authors/          # Known author samples
        mystery_texts/          # Mystery texts
    run.py                      # Entry point
    run_tests.py                # Test runner
    README.md
```

The system analyzes text samples using seven key stylometric measures:

1. **Average Word Length:** Computes mean character count per word, excluding punctuation
2. **Vocabulary Diversity:** Measures lexical richness using type-token ratios
3. **Hapax Legomena Ratio:** Calculates proportion of words appearing exactly once
4. **Average Sentence Length:** Determines mean word count per sentence
5. **Sentence Complexity:** Analyzes syntactic complexity through structural features
6. **Punctuation Density:** Measures the frequency of punctuation marks relative to total text length
7. **Stopword Ratio:** Calculates the proportion of common function words (stopwords) in the text

Each author's writing style is captured in a normalized signature vector combining all seven stylometric features. These signatures serve as fingerprints for comparison operations.

Unknown texts are compared against known author signatures using weighted distance metrics. The system identifies the closest match and provides confidence indicators based on distance thresholds.

3.2 User Interfaces

Users can launch interactive analysis sessions:

```
python run.py
```

3.3 Testing and Validation

The system includes multiple testing categories:

```
# Complete test suite
python run_tests.py

# Specific test categories
python run_tests.py TestTextAnalysis
python run_tests.py TestSignature
python run_tests.py TestIntegration

# Coverage analysis
coverage run run_tests.py
coverage report
```

4 Challenges and Future Work

4.1 Current Challenges

The system faces several key limitations. Text length dependency requires sufficient content (over 100 words) for reliable feature extraction, limiting use with short messages. Genre and topic bias can affect accuracy when authors write across different domains (technical vs. creative writing). The system also doesn't account for temporal style variation as authors evolve their writing over time, and currently only supports English text analysis.

4.2 Future Work

Future enhancements could include machine learning integration using neural networks or support vector machines for improved accuracy. Additional stylometric features such as part-of-speech distributions and punctuation patterns could enhance discrimination. System scalability improvements through parallel processing and database integration would support larger author corpora. Finally, domain-specific adaptations for social media text, code authorship attribution, and historical document analysis would expand practical applications.