Programming Assignment 1: Tokenize and Count Words in Alice in Wonderland

CS 584-WN: Natural Language Processing

Due: 11:59pm, October 16th, 2024

Assignment Overview

The goal of this assignment is to thoroughly tokenize the text (.txt file) *Alice in Wonderland*, which can be found in the 'Programming Assignment 1' module, and create two frequency dictionaries:

1. A **Token Frequency Dictionary**: Case-sensitive tokens based on specific tokenization rules.
2. A **Full Word Frequency Dictionary**: Complete, correctly spelled words.

These will be python dictionaries. You must thoroughly comment on your code, line-by-line, explaining new functions and what they do to process text. You do NOT need to comment on well-known, trivial functions like the print statement. However, you MUST explain how each regex or tokenization function is changing the text.

Words in the 'Full Word Frequency Dictionary' must be spelled correctly. You will be graded at least partially on how correct and comprehensive this dictionary is.

Output Requirements

1. Token Frequency Dictionary

- **Data Structure**: Python `dictionary`.
- **Keys**: Case-sensitive tokens (e.g., `"Alice"`, `"ALICE"`, `"said"`).
- **Values**: Frequency of each token (e.g., the number of occurrences).
- **Punctuation**: Include punctuation as standalone tokens.
- **Example**:

```
"Alice": 5, "said": 10, "ca": 3, "n't": 3, ",": 15
```

2. Full Word Frequency Dictionary

- **Data Structure**: Python `dictionary`.
- **Keys**: Complete words.
- **Values**: Frequency of each complete word.
- **Example**:

```
"Alice": 5, "can't": 3, "believe": 2
```

Instructions

1. **Tokenization**:
   - Tokenize the text of *Alice in Wonderland* using industry-standard rules:
     - Split punctuation if it's not part of the word (e.g., commas, periods).
     - Keep contractions as separate tokens (e.g., `"ca"`, `"n't"`).
     - Maintain case sensitivity (e.g., `"Alice"` is distinct from `"ALICE"`).
     - Split hyphenated words unless they are common expressions or proper nouns.
2. **Dictionary Creation**:
   - Create a **Token Dictionary** that tracks the frequency of each token.
   - Create a **Full Word Dictionary** that tracks the frequency of each full word in the text (e.g., `"can't"` remains one word in this dictionary).

Submission Instructions

- Submit your assignment as a `.py` or `.ipynb` file.
- Upload your file to the **Programming Assignment 1** section on the course platform.
- I should be able to run your file by simply downloading it, and running it in Google Colab.
- Include comments and explanations in your code of all unique regex, tokenization, and edit distance functions
- If you are using external libraries, include the installation commands at the top of your file (e.g., `!pip install <package>`).

Due Date: October 16th, 2024

Good Luck!