CDS Language Analytics Assignment #1: Linguistic Feature Extraction

This script aims to extract linguistic features from a text corpus (USEcorpus), and arrange these feature statistics into organized `.csv` files.

This code uses CodeCarbon to monitor the environmental effects of running this code. The effects of which can be found in the `out/emissions` folder

Setup

1. Make sure to have python and Git Bash installed!
2. Open a Git Bash terminal and use Git to download the repository:

```
git clone https://github.com/missingusername/cds-lang-git.git
```

3. Navigate to the project folder for this assignment:

```
cd assignment1
```

4. Before running the program make sure you have Spacy, Pandas, and TQDM installed. This can be done by simply running the `setup.sh` script from inside the `assignment1` folder, again using Git Bash:

```
bash unix_setup.sh
```

or

```
bash win_setup.sh
```

5. Before we can run the script, we first need to get the actual data. You can click here to download the dataset directly.
   The `USEcorpus.zip` can also be downloaded manually from the Oxford Text Archive.
   When downloaded, unzip the folder and place "USEcorpus" in the `in` folder of the assignment1

directory.

This should leave you with a file structure like this:

```
assignment1/
        ├── in/
        │   └── USEcorpus/
        │       ├── a1/
        │       ├── a2/
        │       ├── a3/
        │       ├── ...
        │       └── c1/
        │
        ├── out/
        │   ├── a1 table.csv
        │   ├── a2 table.csv
        │   ├── a3 table.csv
        │   ├── ...
        │   └── c1 table.csv
        │
        └── src/
        └── main.py
```

1. To finally execute the file, simply run the OS-appropriate bash script in the same Git Bash terminal:

```
bash unix_run.sh
```

or

```
bash win_run.sh
```

Takeaways from output

To examine the output of the code, we can use one of the tables as an example:

c1 table.csv

Filename	RelFreq NOUN	RelFreq VERB	RelFreq ADJ	RelFreq ADV	Unique PER	Unique LOC	Unique ORG
0140.c1.txt	1777.16	1054.33	534.07	455.8	38	0	5
0165.c1.txt	2084.55	976.68	694.85	340.14	27	0	3
0200.c1.txt	1393.69	1209.05	768.31	601.55	17	0	8
0219.c1.txt	1587.79	1122.14	648.85	557.25	26	0	6
0238.c1.txt	1252.24	1332.74	456.17	330.95	18	0	3
0501.c1.txt	1422.89	1184.42	532.59	492.85	14	0	5
0502.c1.txt	1500.0	1384.06	492.75	463.77	15	0	5

- RelFreq NOUN (Relative Frequency of Nouns):

Indicates the density of nouns in the text. Higher values suggest a text that may be more focused on objects, entities, or subjects.

- RelFreq VERB (Relative Frequency of Verbs):

Reflects the density of verbs in the text. Higher verb frequencies suggest a text with more actions or events, possibly indicating a narrative.

- RelFreq ADJ (Relative Frequency of Adjectives):

Shows the density of adjectives. A higher frequency of adjectives indicates more descriptive language, which may suggest a more vivid or expressive text.

- RelFreq ADV (Relative Frequency of Adverbs):

Indicates the density of adverbs. Texts with higher adverb frequencies may provide more detail about actions, describing how actions are performed. This could indicate a more nuanced or detailed narrative.

- Unique PER (Unique Person Entities):

The number of unique named person entities in the text. Higher numbers suggest texts that mention many different individuals, such as a story or a text that covers multiple people, e.g. an article.

- Unique LOC (Unique Location Entities):

The number of unique named location entities in the text. A high count suggests the text references multiple places, which might be seen in travel writing, reports involving different regions, or any text where geography is important.

- Unique ORG (Unique Organization Entities):

The number of unique named organizations in the text.

The Code

The script utilizes the SpaCy library to tokenize and analyze the text, extracting part-of-speech tags and named entities. It iterates over each text file in each subfolder of the input directory, reads it using latin1 / ISO-8859-1 encoding, cleans the text by removing metadata, and then calculates the relative frequency of nouns, verbs, adjectives, and adverbs per 10,000 words (I use spaCy to not count punctuation as words, which would impact the relative frequency of the POS tags).

Additionally, it counts the total number of unique entities (persons, locations, and organizations) mentioned in each text. The extracted information is handled using pandas dataframes and is saved in CSV format, with each sub-folder corresponding to a separate CSV file containing a table with filenames and their extracted linguistic and entity information. The code is modularized for readability, with functions to process individual files and entire folders.

TQDM is used for showing the scripts progress in its processing of each folder, helps the user track how far along the script is in its processing of the data.

Limitations and Improvements

While the end results fulfills the requirements set forth in the assignment, the tables generated are still hard to really understand and gain any real insight from without some sort of visualization to accompany them.