Project: Text Analytics on Song Lyrics

Authors: Serge Pellegatta, Selina Steiner

GitHub Repository: https://github.com/selina080701/Text-Analytics-Project

---

Project Description

This project is part of the *Text Analytics* module. Students work in teams on a practical project where learned methods are applied to solve an innovative, data-driven question. The task involves:

- Identifying a concrete challenge suitable for text analytics
- Preparing relevant data sources
- Selecting and evaluating appropriate models
- Implementing a functional Python prototype

This project is based on the [Genius Song Lyrics Dataset](https://huggingface.co/datasets/sebastiandizon/genius-song-lyrics) from Hugging Face.

Notebooks

1. `load-data-subset.ipynb`
- **Purpose:** Load a smaller subset of the full Genius Song Lyrics dataset from Hugging Face and save it locally.
- **Details:** Allows downloading a lightweight subset (e.g., 1%, 5%, 10%) to reduce memory and storage usage.
- **Output:** Saves the raw subset CSV files in `data/raw/`.

2. `data-cleaning.ipynb`
- **Purpose:** Clean and preprocess song lyrics for analysis.
- **Details:** Removes metadata tags (e.g., `[Intro]`, `[Verse]`), line breaks, and extra spaces. Renames the cleaned lyrics column to `lyrics`.
- **Output:** Saves the cleaned CSV files in `data/clean/`.

3. `tokenization.ipynb`
- **Purpose:** Perform tokenization and remove stopwords.
- **Details:** Splits the song lyrics into individual tokens, removes stopwords, and creates new columns `tokens`, `token_count`, `words` and `word_count`.
- **Output:** Saves the final CSV files as `data/clean/data.csv`.

4. `statistical-analysis.ipynb`
- **Purpose:** Explore patterns and distributions in the cleaned song lyrics across genres and artists.
- **Details:** Focuses on word frequencies, stylistic differences, and similarity structures.
- **Input:** Uses cleaned CSV `data/clean/data.csv`.

5. `word-embedding.ipynb`
- **Purpose:** Create and explore word embeddings.
- **Details:** Uses tokenized data to generate embeddings, visualize semantic relationships, and analyze similarity between words.
- **Input:** Uses tokenized CSV `data/clean/data.csv`.

Folder Structure

- `data/raw/` : Raw subsets of the dataset (e.g., 1%, 5%)
- `data/clean/` : Cleaned versions of the subsets
- `data/clean/data.csv` : Final dataset for analysis, embeddings, etc.
- `load-data-subset.ipynb` : Notebook to load and save raw subsets
- `data-cleaning.ipynb` : Notebook to clean the raw lyrics
- `tokenization.ipynb` : Notebook to tokenize lyrics and remove stopwords
- `statistical-analysis.ipynb` : Notebook to perform analysis on cleaned data
- `word-embedding.ipynb` : Notebook to generate and analyze word embeddings
- `requirements.txt` : Python dependencies for the project

Setup

1. **Create a virtual environment**
```bash
python3 -m venv .venv
```

2. **Activate the virtual environment**
* macOS / Linux:
```bash
source .venv/bin/activate
```
* Windows (PowerShell):
```bash
.venv\Scripts\Activate.ps1
```

3. **Install dependencies**
```bash
pip install -r requirements.txt
```

4. **Configure Git for Jupyter Notebooks (optional, avoids merge conflicts)**
```bash
# Install nbdime
```

```
pip install nbdime

# Enable Git integration globally
nbdime config-git --enable --global
```

5. **(Optional) Clear notebook outputs before committing**
```bash
jupyter nbconvert --ClearOutputPreprocessor.enabled=True --inplace <notebook>.ipynb
```