# **Semantic Search: TF-IDF Gloss Similarity Tool**

## **Overview**

The **TF-IDF Gloss Similarity Search Tool** is a text similarity and search system designed to work with a dictionary of linguistic glosses. The tool allows users to:

1. Search glosses containing a specific keyword.
2. Find glosses that are similar to a given word.
3. Retrieve glosses that are semantically close to a user-provided sentence.

This tool computes **TF-IDF (Term Frequency-Inverse Document Frequency)** vectors from gloss descriptions and uses **cosine similarity** to compare textual meanings across glosses. A command-line interface (CLI) enables interactive exploration of these capabilities.

## **Project Language**

The entire project is written in **Python 3**.

## **Core Libraries Used**

The following libraries are employed in the tool's development:

* **pandas**: For loading and processing tabular data from a CSV file (dictionary.csv).
* **math**: For performing mathematical operations such as logarithms and square roots.
* **re**: For regular expression-based text tokenization.
* **collections**: For utilizing defaultdict to efficiently count word occurrences.

## **Project Structure and Key Components**

### **1. Data Loading & Preparation**

The input to the system is a CSV file (**dictionary.csv**) containing the following columns:

* **letter**: Represents the linguistic category of the gloss.
* **pos**: Part of speech for the gloss.
* **gloss**: The definition or description.

If the **word** column is not already present, it is automatically generated as a unique identifier.

### **2. Tokenization**

The gloss descriptions are tokenized using a regular expression, which extracts lowercase words of at least two characters. This ensures that the system operates on clean, relevant terms.

### **3. TF-IDF Computation**

Each gloss is transformed into a sparse **TF-IDF vector**:

* **Term Frequency (TF)**: The frequency of a token within a specific gloss, normalized by the total number of tokens in the gloss.
* **Inverse Document Frequency (IDF)**: A measure that penalizes common terms by computing their frequency across all glosses in the dataset.

### **4. Cosine Similarity**

Cosine similarity is used to compute the similarity between two sparse TF-IDF vectors. This function compares glosses to determine their semantic similarity.

## **Features**

### **1. Keyword Search**

This feature allows users to search for a specific keyword within the glosses. The tool returns a list of glosses that contain the given keyword.

### **2. Word-to-Gloss Similarity**

Users can input a word, and the tool will identify the first gloss that contains this word. It will then rank all other glosses by their similarity to this reference gloss. This is useful for identifying semantic neighbors related to the provided word.

### **3. Sentence-to-Gloss Similarity**

Users can input a sentence, which will be transformed into a TF-IDF vector. The tool will then identify glosses that are semantically closest to the meaning of the sentence. This feature is useful for finding gloss entries that match a described concept or meaning.

### **4. Exit**

A clean exit command to terminate the interactive CLI session.

## **TF-IDF Workflow Summary**

1. Tokenize each gloss description.
2. Compute word frequencies for each gloss.
3. Calculate document frequencies (DF) across the entire dataset.
4. Derive IDF values based on document frequencies.
5. Construct sparse TF-IDF vectors for each gloss.
6. Perform similarity searches by calculating cosine similarity between vectors.

## **Potential Applications**

* Linguistic research tools
* Lexicographical data exploration
* NLP educational tools
* Glossary-based semantic search engines

## **How to Use**

1. Ensure that the **dictionary.csv** file is properly formatted, with columns for letter, part of speech, and gloss descriptions.
2. Run the script to start the CLI tool.
3. Use the menu to:
   * Search for glosses containing a specific keyword.
   * Find glosses similar to a given word.
   * Input a sentence and retrieve glosses with similar meanings.

## **Limitations and Considerations**

* The tool is case-insensitive and processes only alphabetic tokens of at least two characters in length.
* The first gloss containing a given word is used as the reference for word similarity.
* Gloss entries must not be empty; rows with missing gloss descriptions are excluded from processing.

## **Conclusion**

The **TF-IDF Gloss Similarity Search Tool** offers a lightweight, interpretable solution for semantic search using classical Natural Language Processing techniques (TF-IDF and cosine similarity). It is ideal for scenarios where vector embeddings or advanced language models are not feasible or necessary. This tool provides an efficient, transparent, and fast alternative for similarity-based search and data exploration in glossaries.