Fairy Tailer Language Model Project Documentation

Overview

This project builds and tests language models using texts from Hans Christian Andersen and the Brothers Grimm. It includes:

1. **Uniform Model**: All words have equal probability.

2. **Unigram Model**: Probabilities based on individual word frequencies.

3. **Bigram Model**: Probabilities based on pairs of words.

Installation

1. **Install Python**: Ensure Python 3.8 or higher is installed. Download Python

2. **Install Required Libraries**

Install the necessary libraries with:

{ pip install nltk }

File Structure

- language.py: Functions for text processing and model generation.
- language_tests.py: Unit tests for verifying functions.
- run_models.py: Script to process text and generate output.
- andersen.txt: Text corpus from Hans Christian Andersen.
- grimm.txt: Text corpus from the Brothers Grimm.

Usage

1. **Prepare Text Files**

Place andersen.txt and grimm.txt in the same directory as run_models.py.

2. **Run the Model Script**

Execute the following command to process the text and generate output:

{ python run_models.py }

Functions

- **tokenize(text)**: Converts text into a list of lowercase words.
- **build_vocabulary(words)**: Creates a set of unique words from the text.
- **count_unigrams(words)**: Counts occurrences of each word.
- **count_bigrams(words)**: Counts occurrences of word pairs.
- **uniform_probabilities(vocabulary)**: Computes uniform probabilities for all words.
- **unigram_probabilities(unigram_counts)**: Computes probabilities based on word frequencies.
- **bigram_probabilities(bigram_counts, unigram_counts)**: Computes probabilities for word pairs.
- **load_text(file_path)**: Reads text from a file.

Testing

Run the tests using:

{ python -m unittest language_tests }

Troubleshooting

- **FileNotFoundError**: Verify that andersen.txt and grimm.txt are in the correct directory.
- **SyntaxError**: Use raw string literals or forward slashes for file paths.