Fisayo Jassey- Jabarr

Professor Patricia McManus

ITAI 2376: Deep Learning in Artificial Intelligence

March 07, 2024

**Technical or journal Report: Text Processing Techniques in Python (Module 2: Lab 02)**

Introduction

Like lab 01, lab02 exposed me to introduce text processing techniques using libraries like NLTK, spaCy, and WordCloud. Exposing me to the functionalities like text pre-processing, word cloud generation, part-of-speech tagging, stemming, lemmatization, and named entity recognition (NER).

My observations are as follows:

* Library Usage: The labs effectively utilize relevant libraries:
  + NLTK: for POS tagging and stemming/lemmatization.
  + spaCy: for named entity recognition (NER).
  + WordCloud: for text visualization.

Text Preprocessing: A *preProcessText* function cleans the text by removing special characters, numbers, and converting it to lowercase. This ensures consistency for subsequent processing steps.

Word Cloud Generation: The code demonstrates how to create a word cloud using the WordCloud library. It removes stop words before plotting, highlighting the most frequent words in the text.

Part-of-Speech (POS) Tagging: The code utilizes NLTK's POS tagger to identify the part of speech for each word. It includes a table explaining the tag meanings for better understanding.

Stemming vs. Lemmatization: The code showcases both stemming and lemmatization techniques. It effectively highlights the trade-off between speed (stemming) and accuracy (lemmatization).

Named Entity Recognition (NER): The code leverages spaCy to identify and categorize named entities like organizations and locations within the text.

My key Learning Points

* Importance of Text Preprocessing: Preprocessing text by removing noise and inconsistencies is essential for accurate Natural Language Processing (NLP) tasks.
* Word Cloud for Text Exploration: Word clouds offer a quick visual approach to identify frequently occurring words in text data.
* POS Tagging for Text Structure: POS tagging helps understand the grammatical structure of a sentence by identifying the role of each word.
* Stemming and Lemmatization for Efficiency: Stemming and lemmatization reduce words to their base forms, improving model performance and efficiency.
* NER for Information Extraction: NER allows for extracting valuable information like names and locations from textual data.

Conclusion

This exploration highlights the vast potential of Python libraries for text analysis and manipulation.