Reflection Journal: Basic Preprocessing Techniques

Key Insights

I gained several insights while performing the listed tasks on this project from understanding the key concepts on natural language programming to different jargons used while preprocessing NLP model such as tokenization, lemmatization, stemming. The main insight from this lab is that pre-processing is key to having a quality output and that the process is not one-size-fits-all task. For instance, labeling a word as a noise in one context might be different in another context. In the lab, the importance of sentiment analysis, information extraction and noise removal are evaluated. In addition, I gained understanding of the Spacy library as a powerful engine for processing language that helps in solving problems on tokenization, POS tagging, named entity recognition, and lemmatization in a single code run efficiently making it a powerful and robust tool in comparing with the single modular, academic approach of NLTK.

Challenges

Key challenges encountered during the lab included common way to preprocess text. My natural instinct was finding the universal technique that can be applied for pre-processing text. For example, I was surprised by how much meaning was lost when stemming words such as flying->"fli" or amazing ->"amaz" as outputs. This prompted me to have a critical look on this approach as accepting these words will change the nuances in the text. In addition, the results generated when handling contractions and emojis by Spacy differently from NLTK made me more curious about the tools and libraries that are being used in performing preprocessing of text in NLP model.

Connections to Real-World Applications

The connection between these techniques and real-world applications became very clear during this lab.

- Search Engines: The need for speed and high recall in a search engine makes stemming a logical, but imperfect choice while searching for particular texts.
- Chatbots: Lemmatization in a chatbot shows a vivid representation that is applicable. A chatbot that can't understand that "better" is a comparative form of "good" may not be beneficial as intended.
- Social Media Analysis: Analyzing customer feedback, performing sentiment analysis on market data and identifying trends on social media contents require careful handling of emojis and hashtags rather than discarding them abruptly.

Questions That Arose

- How to handle out-of-vocabulary words or any slang/jargon that are not present in a model's vocabulary?
- Are there more advanced techniques for handling sarcasm, where the literal meaning of the words is the opposite of the intended sentiment?
- How do you decide when to update your stop word list?
- When is the right time to use aggressive preprocessing without changing the context completely?

Comparisons

- NLTK vs. spaCy: NLTK is a great tool for research, education and experimenting purpose that is string-based, while spaCy is a more powerful, industrial- strength library, object-oriented and efficient tool for building production-ready systems. SpaCy's integrated pipeline performs tokenization, POS tagging, Lemmatization, while NLTK performs tokenization only.
- Stemming vs. Lemmatization: Stemming does not have good grip on comparative form of words as shown in the “better->better”, “good->good” results because stemming is a rule-based process that primarily works by chopping off common prefixes and suffixes with no understanding of the semantics while lemmatization is more precise and linguistically-informed tool. Lemmatization provides deep understanding of semantics.

Future Applications

There are several applications that I think the pre-processing concepts learned in this lab will be used in later dates:

- Creating sentiment analysis model for movie reviews, market trends, socio-economic surveys where lemmatization would be highly beneficial for standard preprocessing.
- Analyzing a large corpus of text such as research or scientific articles to identify key themes and topics.
- Creating a simple chatbot for a specific purpose where preprocessing pipeline that will handle slangs/ jargons adequately without distorting query response.

Overall, this lab has provided me more nuanced and practical understanding of the challenges/opportunities in NLP preprocessing.