Trevon Woods
Lab 02 - Journal
NLP - ITAI - 2373
06/10/2025

Throughout this lab I had the opportunity to gain a lot of valuable insights in regards to language preprocessing. Firstly, my understanding of how to code and use Stemming and Lemmatization has increased. I already understood the underlying premise behind these techniques but haven't seen them in action. I now have a deeper understanding of the differences between these two techniques along with how and where to apply them. Through this journal I will convey the key insights along with challenges, connections, questions, and comparisons I've obtained from this lab's content if I had any. I will also touch on how I may use these techniques in future projects.

The first major insight that I gained was during the tokenization section. NLTK had a more basic approach to how it tokenized the text and spaCy was a lot more verbose. SpaCy gave detailed token information such as POS which stands for "part-of-speech", this gives insight into whether it is a noun, proper noun, etc. Their tokenizer also shows whether the token is alphanumeric or if it is a stop word. This added information is very valuable if I want to get a broad understanding of what tokens are in the text. Now one prop I will give to NLTK is that it has an option to tokenize sentences. There was no such capability like this for spaCy in this lab but I'm sure through a little research I can find an implementation.

Next major insight I gained was from the stop words section. There is not much to say about this section except that spaCy has a lot of weird stop words. My guess is that it is trying to catch all contractions in the text. Even with this being said spaCy seems to be the best for this task. NLTK just split the text but kept things like contractions in the data while spaCy caught these things and removed them from the data.

The final insight that I gained pertained to the stemming and lemmatization sections. The interesting thing here is that we used NLTK for stemming and spaCy for lemmatization. I initially thought that each library would have their own packages for both stemming and lemmatization but it was cool to see that they each have their own role to play. Other than that NLTK seems to have a weird stemmer for a word like “flying” it outputs “fli” which is wrong in my opinion. This makes me think I’ll probably have to code my own or find a better library if I want proper stemming. SpaCy’s lemmatization was spot on as usual, it correctly found the base words of each token and even gave additional information for analysis.

Overall, I understand that text preprocessing is very important for NLP tasks mainly to reduce noise and inconsistencies in the data while also dumbing it down so our models can find representations better. In terms of future applications, I'm sure I will have to use these techniques if I want to create any kind of chat bot or LLM. In order to have my model train on corpuses of data I will need to remove the noise from the data and augment it to a base representation. I also did not have many challenges or questions throughout this lab as I was already familiar with these techniques.