Ruben Valenzuela  
Professor Patricia McManus

Natural Language Processing

10 May 2025

**Reflection Journal: Lab 02 - NLP Preprocessing Techniques**

Engaging with Lab 02 gave me a deeper appreciation for the foundational role that preprocessing plays in any Natural Language Processing (NLP) pipeline. One of the most valuable insights I gained is that raw text data, no matter how seemingly clean or short, cannot be effectively analyzed by machine learning models until it undergoes a series of structured transformations. Techniques like tokenization, stop word removal, stemming, and lemmatization are not just optional steps; they shape the quality and meaning of downstream results.

A challenge I encountered while working through the concepts was understanding the nuanced differences between stemming and lemmatization. Initially, they seemed similar, but as I examined their output side by side, I noticed how stemming can distort meaning by cutting words too aggressively (e.g., "batteri" instead of "battery"), while lemmatization preserved semantic clarity (e.g., "better" becoming "well"). This distinction made me realize that the choice of technique must align with the goals of the task, lemmatization being ideal for sentiment analysis, and stemming better suited for keyword-based searches.

Another significant realization was the contrast between NLTK and spaCy. While NLTK is more academic and transparent, spaCy impressed me with its robustness and performance, especially in handling messy, real-world data like social media posts. For instance, spaCy’s tokenization broke down contractions effectively ("It’s" into ["It", "'s"]) and offered detailed part-of-speech tagging, lemmas, and stop word indicators. This comparison raised important questions for me: When is it better to prioritize speed over linguistic depth? And how might these trade-offs affect large-scale applications like real-time chatbots or customer service platforms?

I also began connecting these techniques to real-world applications. For example, cleaning and standardizing social media content could make or break a sentiment analysis system designed to detect public opinion on a brand. In contrast, minimal preprocessing might be better for tasks like emotion detection or creative writing analysis, where punctuation and word flow matter.

As I experimented with different preprocessing pipelines, minimal, standard, and aggressive, I saw firsthand how each approach preserved or stripped information. Minimal processing retained full context but kept a lot of noise. Standard struck a balance, ideal for classification tasks. Aggressive processing removed the most data, which might be helpful for document retrieval but risks oversimplifying.

Looking ahead, I can see myself applying these insights to projects involving chatbots, recommendation systems, or even forensic linguistic analysis. Understanding how preprocessing affects data quality will help me design more effective NLP models that are not only technically sound but contextually aware.

In summary, Lab 02 challenged me to think critically about each preprocessing decision and its downstream consequences. I now recognize that there's no one-size-fits-all approach in NLP; the key is to match preprocessing strategies with the objectives and context of the task. This lab didn’t just teach me technical tools, it helped me start thinking like an NLP practitioner.