L02 Basic Preprocessing Techniques

Jiri Musil

Department of Science, Technology, Engineering & Math, Houston Community College

 Natural Language Processing (NLP) - ITAI 2373

Patricia McManus

June 8th, 2025

**Introduction**

Spending several hours in the pre-processing lab provided me with my first genuine insight into the stages that occur before an NLP model encounters a sentence. I came in only knowing that “some cleaning” happens. After finishing the assignment, I realized that these initial choices influence how much meaning survives the transition from raw text to actionable data.

**Reflections**

The main insight was that tidying up text is not a one-size-fits-all solution. It rather involves a series of trade-offs. Running three pipeline settings, such as light, moderate, and heavy, on the same laptop review resulted in dramatically different token counts and even varied moods. A simplified pass increased speed but removed intensifiers like “absolutely,” which carry emotional weight. That contrast revealed how each rule can silence part of the message.

Comparing NLTK and SpaCy side by side emphasized the importance of tool choice. NLTK was quick to set up and easy to adjust, making it excellent for testing ideas or teaching concepts. On the other hand, SpaCy provided additional information, including part-of-speech tags and lemmas, with minimal extra code. Those extras appear invaluable for tasks requiring linguistic nuance, such as entity recognition, but they require more memory. Now, selecting between them feels more like pairing a screwdriver with a screw than choosing a brand.

Two hurdles stood out. First, managing informal social posts proved more challenging than I anticipated. Emojis, hashtags, and creative spelling all convey attitude, yet a careless cleaning step removes them away. I learned that one pipeline won't fit every type of text, social data requires gentler treatment than, for example, academic prose. Second, stemming often felt too rough. Seeing “amazing” reduced to “amaz” looked strange, and identical words did not always share a common root. In contrast, lemmatization retained actual words, but was slower. That experience illustrated why basic search engines often emerge, while sentiment systems rely on lemmas.

The lab consistently pointed out that “little” words and symbols are not always expendable. Removing stop words certainly reduced the vocabulary, yet discarding "not” in a phrase like “not bad” would reverse the sentiment. Likewise, punctuation marks such as “?!” often indicate that a customer is frustrated. I now understand that every deletion must be justified in relation to the goal of the project. These lessons connect directly to my day job. When I analyze employee feedback for themes, conjunctions like “but” often indicate a shift in tone. Losing them could hide important details.

A few questions remain on my mind. Could a dynamic stop-word list refresh itself by checking which tokens never affect predictions? Is there an effective way to keep emojis as sentiment cues without overwhelming the feature space? When does stemming become faster than lemmatization even if it loses some meaning?

In the future, I plan to apply what I learned by treating preprocessing as a configurable layer in every NLP project. Creating adaptable pipelines and evaluating how each adjustment impacts accuracy, speed, and interpretability will help ensure that cleaning choices stay aligned with project goals instead of routine.

**Conclusion**

This exercise transformed pre-processing from a vague checklist into a crucial design stage, where every lowercase command or token split influences the narrative that the data can convey. Even with my beginner-level understanding, I now appreciate why experienced practitioners consider text cleaning as part of the model rather than an afterthought. The next time I start an NLP project, I will spend as much time planning the clean-up as I will selecting the algorithm.

**Resources**

1. GeeksforGeeks. (2024, October 3). *Text preprocessing in NLP*. GeeksforGeeks. <https://www.geeksforgeeks.org/text-preprocessing-for-nlp-tasks/>
2. GeeksforGeeks. (2025, June 4). *Tokenization in NLP*. GeeksforGeeks. <https://www.geeksforgeeks.org/nlp-how-tokenizing-text-sentence-words-works/>