Erwin Cheng

L04_Reflection Journal

ITAI 2373 Natural Language Processing

Anna Rachapudi

When I started this lab, I thought text representation would just be about “turning words into numbers” so that a computer can use them. But as I went through the different methods, I realized it’s much more than that—the way you represent text can completely change what the model actually learns.

Bag of Words was my first stop, and it felt simple enough—just counting words. But pretty quickly I noticed the problem: common words like “the” or “movie” took up a lot of space without really telling me much about meaning. That’s when TF-IDF felt like a breakthrough. It basically said, “Hey, not every word is equally important,” and that shift made the results look so much smarter. Suddenly, words like “predictable” or “amazing” carried more weight, which made the text comparisons feel more meaningful.

N-grams added another layer of understanding for me. I liked how bigrams and trigrams could capture little phrases like “not good,” which unigrams completely miss. That showed me how context really matters in language. At the same time, I also noticed the downside—adding higher-order n-grams made the feature space grow really fast, which would definitely be a challenge with bigger datasets. It gave me a new appreciation for why we can’t just keep adding complexity without thinking about efficiency.

Embeddings were probably the coolest part of the lab. The idea that words could be placed in a vector space where relationships like *king – man + woman ≈ queen* actually work still amazes me. It feels like the math is capturing meaning in a way that's almost human-like. But I also couldn't ignore the ethical side—if the training data is biased, then those biases end up inside the embeddings too. That made me think about how important it is to question not just how well a model works, but also what values and assumptions are baked into it.

By the end of everything, I realized text representation isn't just a preprocessing step—it's really the foundation for how machines "understand" language. Sparse methods like TF-IDF are still super useful when you want something simple and easy to interpret, while embeddings open up a more nuanced, semantic view of language. The most surprising part for me was how much insight could come from what seemed like small mathematical tricks.

If I keep exploring this area, I'd like to learn more about contextual embeddings like BERT, which don't just give every word a single fixed vector but adjust meaning based on context. That feels like the next big step beyond what we covered here. Overall, this lab made me appreciate how much thought goes into something as "basic" as representing words, and how those choices ripple into everything else that happens in NLP.