Lab 04: Text Representation

Jiri Musil

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

Natural Language Processing (NLP) - ITAI 2373

Patricia McManus

June 26th, 2025

Introduction

Before this assignment, I saw vectorizing text as a simple process that magically converted text into numbers. Spending four hours building pipelines for Bag-of-Words (BoW), TF-IDF, N-grams, and word embeddings from scratch showed me how many design choices are involved between a raw sentence and a machine learning model. Each method answered a different question about the same corpus, and this exercise helped me realize that choosing one is less about achieving the highest accuracy and more about fitting the story I want the data to tell.

Reflections

Counting tokens using a Bag-of-Words matrix felt intuitively clear because each column is labeled with a readable word. However, upon inspecting the entire document-term matrix, I noticed that more than three-quarters of its over 1,000 columns consisted entirely of zeros. This highlights the significant storage cost we incur for that straightforward readability. Applying TF-IDF changed the feature importance, causing common fillers like “movie” to decrease significantly while highlighting rarer terms such as “cinematography” and “must-watch,” which rose to prominence. Using cosine similarity on these weighted vectors then effectively matched the similarity scores with my intuition about which reviews genuinely belonged together. Introducing bigrams helped recover key sentiment flips such as "not bad” that unigrams misclassified. Although trigrams captured more expressive phrases like “edge of seat," it was necessary pruning infrequent n-grams using a min_df threshold to prevent the feature space from becoming unmanageably large.

Seeing a 50-dimensional word embedding model successfully execute the traditional arithmetic king – man + woman → queen was eye-opening. Vector space operations truly reflect semantic relationships. Swapping out a huge, mostly empty TF-IDF matrix with over a thousand columns for a smaller, dense embedding tensor of just fifty dimensions felt like trading a giant warehouse full of empty boxes for a compact suitcase that's packed with meaning more efficient and way richer in what it captures. This shift highlighted how dense representations can provide deeper insights while requiring less storage and computation.

Despite these benefits, I encountered practical challenges. Higher-order n-grams often increased feature counts beyond 5,000 in the full corpus, which required me to establish explicit upper limits and remove nearly duplicate rows to maintain manageable training times. Common adjectives like “great” and “good” appeared in both sentiment categories, so I improved my stop-word list by analyzing log-probability differences to quietly filter out these overused terms without dulling genuine enthusiasm. Finally, a quick bias audit showed that the vector for “doctor” was closer to “he” than “she,” a clear reminder that embeddings reflect societal biases, prompting me to conduct regular bias audits and pursue domain-specific fine-tuning whenever possible.

Conclusion

Vectorization is not just routine maintenance. It is a fundamental design decision as impactful as the choice of algorithm selection. Each representation emphasizes different aspects of the narrative. BoW helps us see who's responsible, TF-IDF points out unique features, N-grams keep the word order, and embeddings capture meaning in a compact shape. If I work on an NLP project, I will focus on making the representation layer adjustable, experiment with different approaches, verify their effectiveness, and ensure it aligns with the questions we're trying to answer with the data.

Resources

1. GeeksforGeeks. (2025, June 27). *Vectors and vectorization techniques in NLP*. GeeksforGeeks. https://www.geeksforgeeks.org/nlp/vectorization-techniques-in-nlp/
2. GeeksforGeeks. (2021, November 28). *Understanding Semantic Analysis NLP*. GeeksforGeeks. https://www.geeksforgeeks.org/understanding-semantic-analysis-nlp/