7.3 Lesson: Sentence and Document Embeddings

Sentence and Document Embeddings

Word embeddings provide a powerful way to represent individual words in a low-dimensional vector space, but many tasks in machine learning require representations of entire sentences, paragraphs, or documents. This presents a new challenge: How can we construct a meaningful single vector from a sequence of word vectors?

One simple and surprisingly effective approach is to take the **average** of the word embeddings in a sentence. For example, the embedding for the sentence “The cat sat on the mat” could be computed by averaging the vectors for “the,” “cat,” “sat,” “on,” “the,” and “mat”. This method is fast, effective for many tasks, and easy to implement. However, like BOW, it still ignores word order and assigns equal importance to all words, regardless of their contribution to the sentence’s meaning.

More advanced methods use models that are explicitly trained to produce embeddings for entire sentences or documents. These include:

- **Doc2Vec**, which extends Word2Vec by learning an additional vector for each document that is used, along with word vectors during training. This document vector captures information about the broader context that cannot be derived from word co-occurrence alone.
- **Universal Sentence Encoder** and **Sentence-BERT**, which apply deep neural networks to encode full sentences into dense vectors. These models are typically trained on tasks like next-sentence prediction, paraphrase detection, or semantic similarity. As a result, they produce embeddings that reflect more than just the content of individual words—they capture higher-level meaning and some syntactic structure as well.

Sentence and document embeddings are especially useful in applications such as semantic search, document classification, clustering, and similarity detection. By converting text into a fixed-length vector, these models make it possible to compare, group, and analyze texts using standard machine learning tools.

Visualizing Document Embeddings

Document embeddings can be used in visualization tasks, just as we saw with BOW.

Figure 1 shows an analysis of all of Shakespeare’s plays, where each play is the average of all the word embeddings, reduced to 2D using PCA, and color coded by genre - red for tragedies, blue for histories, and green for comedies. Histories cluster together in the upper right, comedies gather in the lower left, while tragedies such as *Hamlet* and *Macbeth* sit farther apart, illustrating that the model groups documents by genre and thematic similarity.

Clustering by genre is clear, as well as many interesting details, such as the grouping of *Othello* with the comedies, and the uniqueness of *Hamlet*, off by itself!

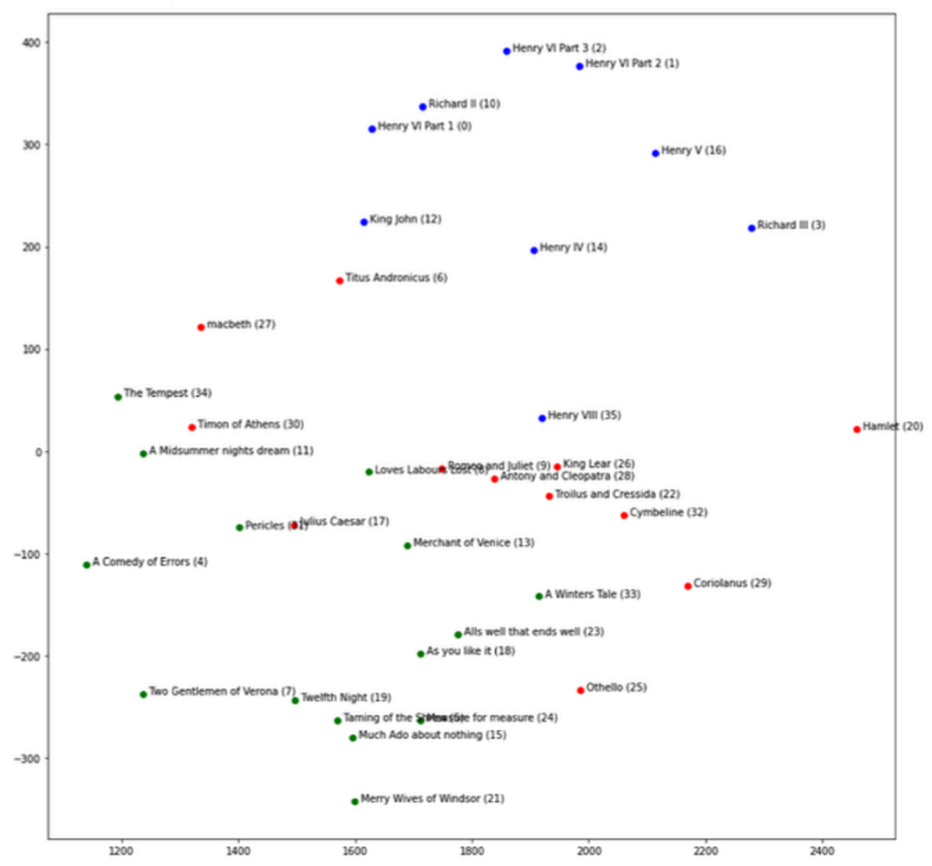

Figure 1: Visualizing Document Embeddings of Shakespeare

Conclusion

Vector models transform language into geometry, allowing us to compare, cluster, and analyze texts using mathematical tools. From sparse one-hot vectors to rich contextual embeddings, these representations provide increasingly powerful ways to capture the meaning and usage of words. But there are limits. Even the best word or sentence embedding models lose information about **word order**, **syntax**, and **discourse structure**. To handle these, we need models that treat text as sequences, not just bags of features.

In the next module, we'll take the next step by introducing **recurrent neural networks (RNNs)—**a family of models designed specifically to handle sequences over time. These models allow us to process language and other time series, capturing patterns that embeddings alone cannot.