Assignment: Process Description

Topic: String Embedding

Course Designation: EGR-3350:05

Course Title: Technical Communications for Engineers and Computer Scientists

Instructor: Ms. Alysoun Taylor-Hall

Author: David Castro

Major: Computer Science

Date: 25 September 2025

Quote of the Day:

> *"What can be said at all can be said clearly, and what we cannot talk about we must pass over in silence. "*
>
> — Ludwig Wittgenstein

Introduction

In natural language processing (NLP), *string embedding* is the process of converting text, called a string in computer science, into a vector that encodes semantic information. This process can be used in search engines. Particularly, in business settings where search algorithms based on click-counts/popularity do not make practical sense. String embedding allows search queries to return results based on meaning rather than by simply comparing letters directly. It is also related to how AI technologies like ChatGPT process language. In NLP parlance, “embedding” is used as both a noun and verb and can refer to the output of the embedding process or it can refer to the process itself. For clarity, we will refer to the output as “vector” or “embedded string vector” and the process as “embedding”. It is worth noting that embedded vectors can be created from a variety of input such as audio and images or from even more abstract forms of data such as knowledge graphs. However, these processes differ in their exact descriptions. Mathematically, a vector is analogous to an arrow. It has some position, direction, and length. A vector can be notated identically to a point on a graph, using parentheses such as (x, y), up to any number of dimensions (x, y, …, n) but the arrow-like nature of a vector implies access to mathematical operations using its direction and length. Embedded string vectors are created through the following steps: tokenization, mapping, and vectorization.

Discussion

See Figure 1 for examples of embedded string vectors on a 2-dimensional graph.

Tokenization

Tokenization is the process of splitting words into subcomponents called tokens. While different strategies exist for tokenization, in the most basic sense, words are split into parts representing grammatical function. For example, the word “eating” might be split into “eat” and “ing”. This splitting is computed using dictionaries of tokens. This makes it easier for the computer to store relatively small dictionaries of tokens that combine in ways to form any word in a language and for the computer to better capture semantic information. For example, the suffix “ing” appears in a variety of words and contains the semantic meaning of an ongoing process. After the input string is tokenized, it is passed to the mapping step.

Mapping

Mapping refers to the process of assigning numerical identifiers to tokens called token IDs. These numbers are arbitrary; they do not contain any meaning at all. However, they are necessary to allow the computer to perform mathematical operations. These IDs exist in a predefined dictionary. IDs are then passed to the vectorization step.

Vectorization

Vectorization refers to the process of converting a token ID into a vector using a special type of function called a pre-trained model. A common term used to describe functions that are not specifically programmed in a pre-defined way is “black box”. This computation is a black box since the model itself is not specifically programmed but is instead designed to learn on its own (training). For string embedding, we assume this training step has already been completed such that a pre-trained model exists as a component of the overall embedding process and is able to map any token ID to some particular vector. Token IDs are passed to this model, and a vector is created for each token in the input string. The vectors are then combined using calculations involving averages and outputted as a single vector.

Conclusion

String embedding is the process of converting a string into a vector that encodes semantic information. Embedded string vectors thereby allow for things like more relevant search engine results or more recently helps enable AI to “understand” human language. String embedding makes use of tokenization, mapping, and vectorization with pre-trained models to output these vectors.

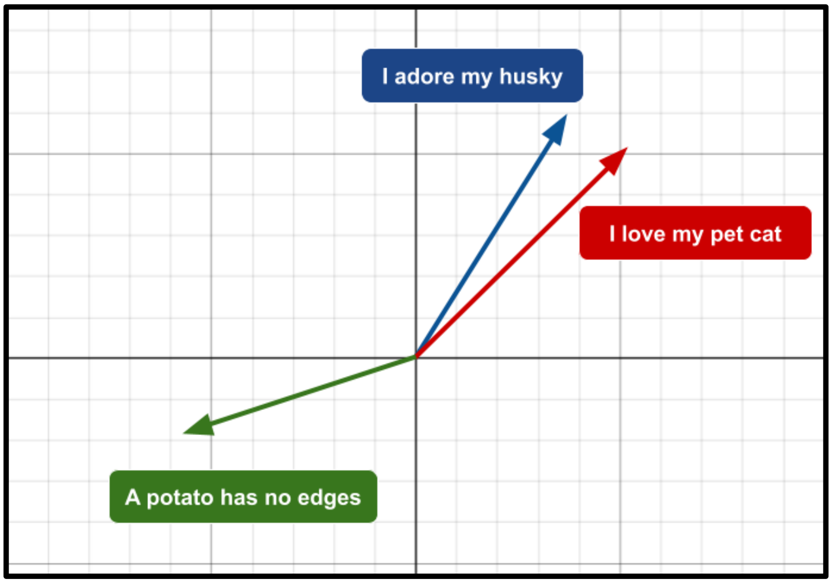

Figure 1: A 2D representation of embedded string vectors of 3 sentences. Note how “I adore my husky” and “I love my pet cat” are very close together showing their semantic similarity. Whereas “a potato has no edges” is far.

Source: Author