Vector Embeddings

Vector embeddings convert real-world content, like documents and images, into 1-D numerical representations (arrays).

These arrays have N values, representing N dimensions. They are called vectors and can be compared with each other efficiently.

These vectors aren't random blobs of numbers. They live in a **semantic multi-dimensional space**, and their position encodes real meaning.

Placing Documents in Space

Let's say you're plotting every document in your system onto a 2D graph:

- **X-axis**: Length of the document (short → long)
- **Y-axis**: Realism (fictional → real)

Let's place some documents in this space.

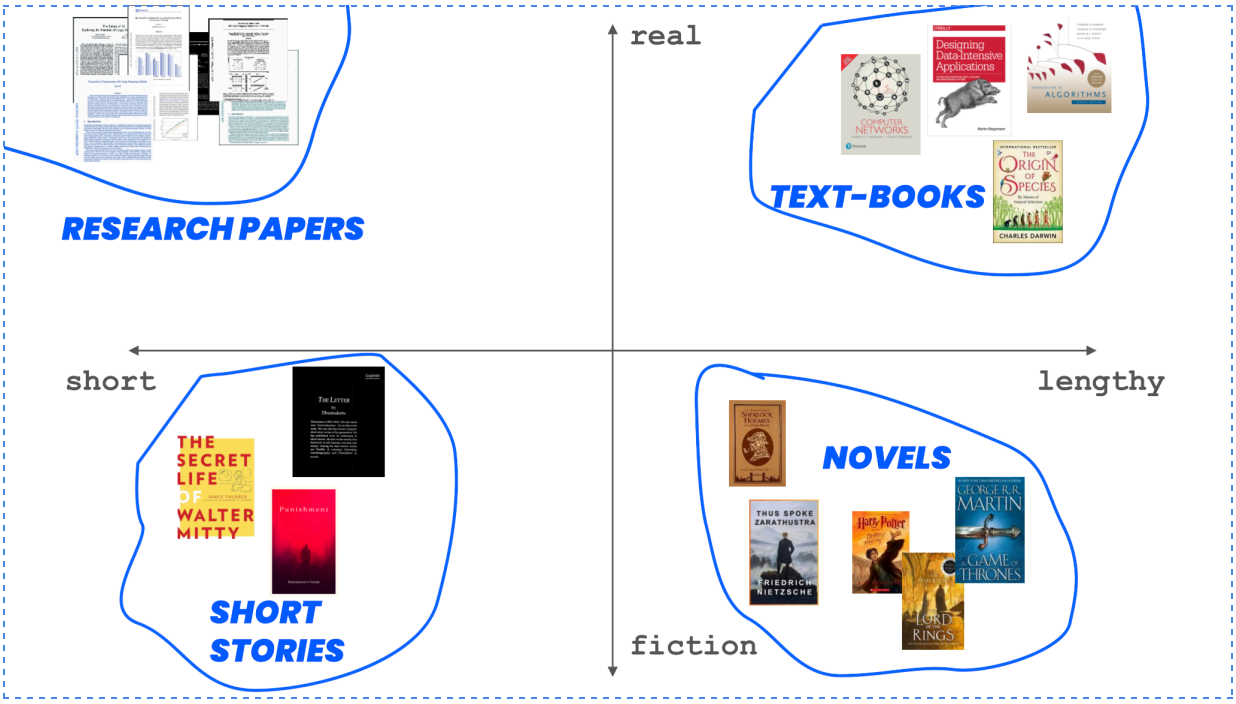

Figure 1: Vectors represent the type of document. Similar documents cluster together.

Emergent topological meaning

Looking at a document, you can immediately classify it as one of four types.

- *Bottom-right*: Long, fictional books.
- *Top-right*: Long, real books (textbooks, reports).
- *Top-left*: Short, real texts (scientific papers, news).
- *Bottom-left*: Short, fictional content (flash fiction, jokes).

Without ever assigning a label, **meaning starts to emerge from position**.

But two dimensions aren't enough. If we want to account for popularity, creation time, and author country, we need more dimensions.

So we move to a D-dimensional space, where every axis represents a property of the input.

In short, a vector embedding encapsulates the input into a position in high-dimensional space.

Figure 2: Objects are converted to positions in the multidimensional space.
Every object's position is encapsulated by a vector with D values. (In LLaMa 3.0, D = 4096).

Key Takeaways

- Vector embeddings = position in a multi-dimensional space.
- Each axis can be thought of as representing a property: realism, length, time, and popularity.
- Similar vectors = semantically similar content.
- Clusters = emergent structure from data, not hard-coded.