**LLM Embeddings — Explained Simply**

[[Sandi Besen](https://medium.com/@sandibesen?source=post_page-----f7536d3d0e4b--------------------------------)](https://medium.com/@sandibesen?source=post_page-----f7536d3d0e4b--------------------------------)

[[AI Mind](https://pub.aimind.so/?source=post_page-----f7536d3d0e4b--------------------------------)](https://pub.aimind.so/?source=post_page-----f7536d3d0e4b--------------------------------)

[Sandi Besen](https://medium.com/@sandibesen?source=post_page-----f7536d3d0e4b--------------------------------)

·

[Follow](https://medium.com/m/signin?actionUrl=https%3A%2F%2Fmedium.com%2F_%2Fsubscribe%2Fuser%2F23260bfdf2e&operation=register&redirect=https%3A%2F%2Fpub.aimind.so%2Fllm-embeddings-explained-simply-f7536d3d0e4b&user=Sandi+Besen&userId=23260bfdf2e&source=post_page-23260bfdf2e----f7536d3d0e4b---------------------post_header-----------)

Published in

[AI Mind](https://pub.aimind.so/?source=post_page-----f7536d3d0e4b--------------------------------)

·

4 min read

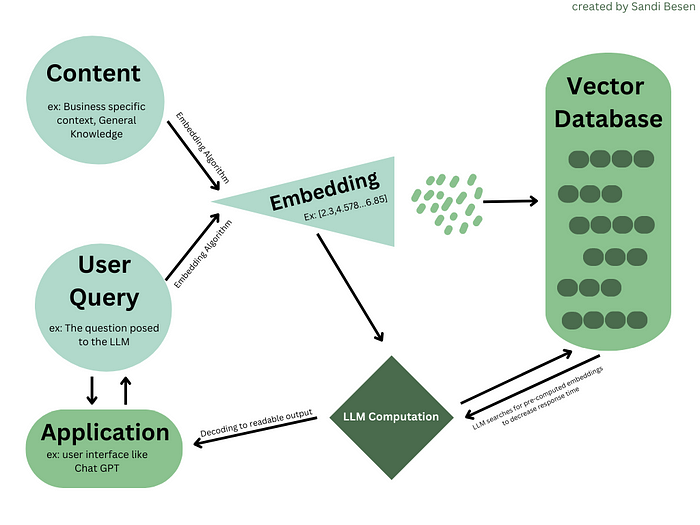
·

Jul 26

204

2

Embeddings are the fundamental reasons why large language models such as OpenAi’s GPT-4 and Anthropic’s Claude are able to contextualize information quickly and effectively. By the end of the article you’ll understand why and be able to follow the flow of data in the following image:



User perspective of flow of data used in Large Language Models

**What are Embeddings?**

In the context of large language models, *embeddings are vectors stored in an index within a vector database*. Before you freak out on me, let’s break the previous sentence into digestible chunks. I promise by the end you’ll be able to read that sentence and tell yourself “I totally get that”.

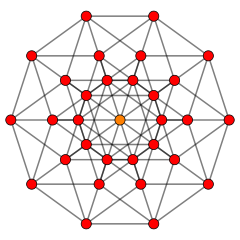
**Embeddings** *are vectors stored in an index within a vector database*:

Embeddings are a way to store data of all types (including images, audio files, text, documents, etc.) in number arrays called vectors.

For instance, the sentence “There are 7 stages in the sales cycle: Prospecting, Making contact, Qualify the prospect, Nurture the prospect, Present offer, Overcome objections, and Close the Sale.” could be represented as the embedding [ 0.00049438 0.11941205 0.00522949 ... 0.01687654 -0.02386115 0.00526433] with the “…” representing hundreds to thousands of other numbers. What precisely these numbers mean is known only to the transformer model that generated them, but we can be sure that they represent the words, their context (the relationship of the words to each other and other embeddings), and their meaning. The embedding stores all the information the LLM model needs to be able to search and generate relevant outputs based on the user’s query.

*Embeddings***are vectors***stored in an index within a vector database:*

Vectors are a way to organize embeddings with meaning. Although these embeddings are represented in 1D on your computer screen, they actually have many dimensions. Humans struggle to visualize more than a 3 dimensional space. One dimension is a line, 2 dimensions can be represented in a graph, 3 dimensions as a cube, but 4+ dimensions… exactly. In a vector, the number of dimensions depends on the length of your vector — “n”. So if an embedding has 150 numbers or n=150, the vector has 150 dimensions. Our human brains can’t fathom what that might look like, but here is a image of a 5D space for context.



source: <https://en.wikipedia.org/wiki/Five-dimensional_space>

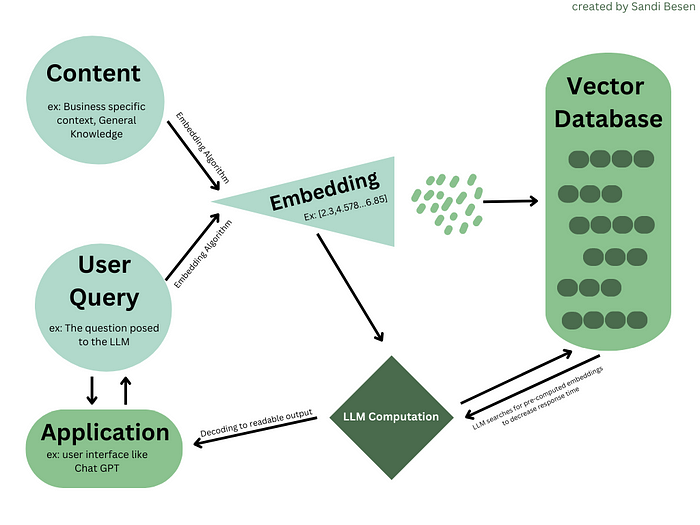
*Embeddings are vectors***stored in an index***within a vector database:*

An index is the order in which the vectors are stored. Why does the order matter? The origin (where the vector starts) , direction (which way the vector moves), and magnitude (length of the vector), determine the vector’s relationship and proximity with other vectors. This relationship is what is used for the LLM to quickly “understand” and generate relevant content. For example, the embedded vector representing the City of “Budapest” might be stored close to the embedded vector representing the city of “London” because they are both Country Capitals. Their distance to each other in the index demonstrates their relationship. So when you ask the LLM “what is the capital of Hungary and The United Kingdom” it can quickly query the answers Budapest and London based on therelationship of their vectors.

*Embeddings are vectors stored in an index***within a vector database***:*

The most efficient way to store a vectorized index is in a vector database, also sometimes referred to as vector store. Unlike traditional databases that store information in rows and columns, vector databases use an algorithm to index the vectors (like we talked about in the previous section). When information is extracted from a traditional database, the algorithm is looking for what row matched that specific search. In a vectorized database, function approximation is used to look for the most similar vector to the query — this process is called Retrieval Augmented Generation (RAG). The LLM then uses the context and pre-embedded vectors to aid in quickly computing the output to the user — resulting in an improved performance.

Now that you can breakdown the sentence “Embeddings are vectors stored in an index within a vector database” into digestible and understandable chunks, take another look at the flow of data process chart.



User perspective of flow of data used in Large Language Models

If you still have questions or think that something needs to be further clarified? Drop me a DM on [Linkedin](https://www.linkedin.com/in/sandibesen/" \t "_blank)! I‘m always eager to engage in food for thought and iterate on my work.