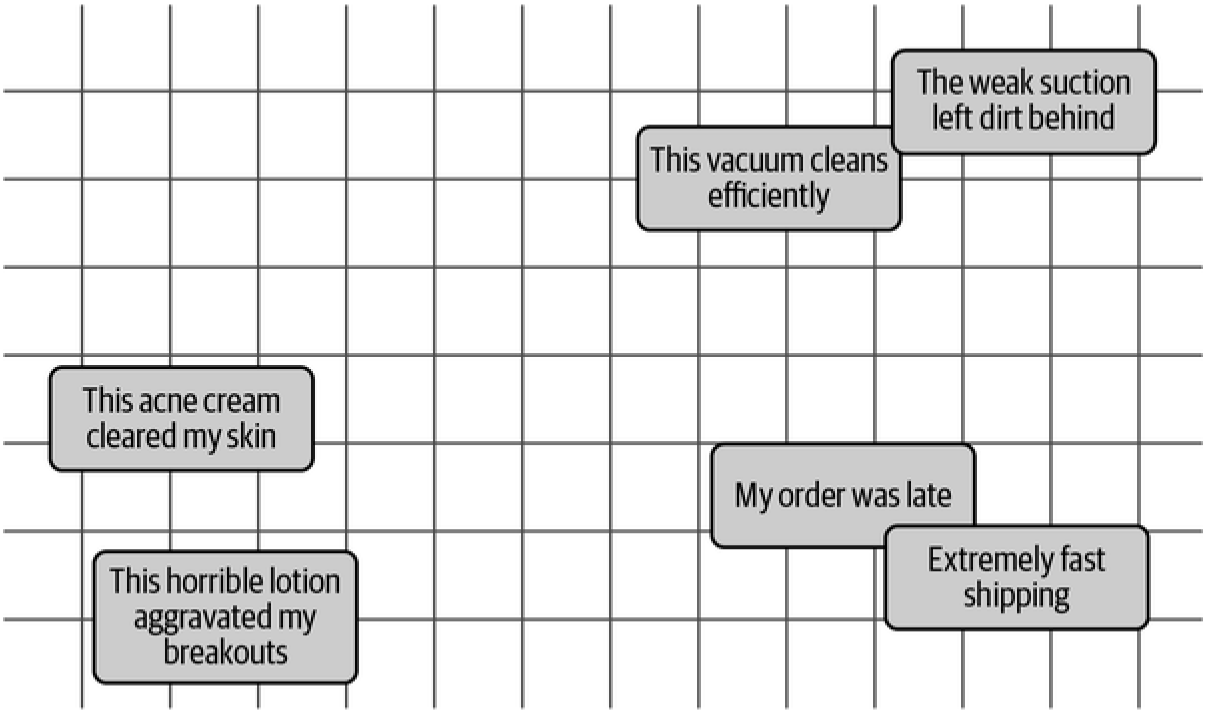
Chapter 10.Creating Text Embedding Models

This process of embedding the input is typically performed by an LLM, which we refer to as an embedding model. The main purpose of such a model is to be as accurate as possible in representing the textual data as an embedding.

However, what does it mean to be accurate in representation? Typically, we want to capture the semantic nature—the meaning—of documents. If we can capture the core of what the document communicates, we hope to have captured what the document is about. In practice, this means that we expect vectors of documents that are similar to one another to be similar, whereas the embeddings of documents that each discuss something entirely different should be dissimilar.



An embedding model, however, can be trained for a number of purposes. For example, when we are building a sentiment classifier, we are more interested in the sentiment of texts than their semantic similarity. As illustrated in Figure 10-3, we can fine-tune the model such that documents are closer in n-dimensional space based on their sentiment rather than their semantic nature.

There are many ways in which we can train, fine-tune, and guide embedding models, but one of the strongest and most widely used techniques is called contrastive learning.

CONTRASTIVE LEARNING :

Another way to look at contrastive learning is through the nature of explanations. A nice example of this is an anecdotal story of a reporter asking a robber “Why did you rob a bank?” to which he answers, “Because that is where the money is.”[1] Although a factually correct answer, the intent of the question was not why he robs banks specifically but why he robs at all. This is called *contrastive explanation* and refers to understanding a particular case, “Why P?” in contrast to alternatives, “Why P and not Q?”[2] In the example, the question could be interpreted in a number of ways and may be best modeled by providing an alternative: “Why did you rob a bank (P) instead of obeying the law (Q)?”

The importance of alternatives to the understanding of a question also applies to how an embedding learns through contrastive learning. By showing a model similar and dissimilar pairs of documents, it starts to learn what makes something similar/dissimilar and more importantly, why.

For example, you could teach a model to understand what a dog is by letting it find features such as “tail,” “nose,” “four legs,” etc. This learning process can be quite difficult since features are often not well-defined and can be interpreted in a number of ways. A being with a “tail,” “nose,” and “four legs” can also be a cat. To help the model steer toward what we are interested in, we essentially ask it, “Why is this a dog and not a cat?” By providing the contrast between two concepts, it starts to learn the features that define the concept but also the features that are not related. We get more information when we frame a question as a contrast.