Scaling the automated retrieval of FAQs

Right now I am using a TF-IDF model, which is a base model and does not scale very well. The main limitations of this model is that it heavily relies on a counting mechanism, rather than the context of the data.

In order to scale this application, I would use an already trained neural embedding model (likely from HuggingFace). A well fed neural network is capable of carrying optimal weights, given a certain amount of data. Next, I would use a library such as LangChain. If we have access to the complete document of FAQs, using LangChain we can split up this document into chunks and each chunk would be turned into a vector by the neural embedding model. This time the vectors do not represent counting frequencies and the context of the words is intact.

As to where to store these vectors, I would choose a vector database, such as Pinecone. We can easily fetch vectors based on cosine similarity.

This time for generating an answer I would use a pretrained LLM, like: Mistral, DeepSeek and so on.

So now finally, when the user asks a question, his question gets embedded with the same model as we have used for our FAQs document. We take this query and match against all the vectors that we have in our vector database and we retrieve the 3 vectors that are the most similar to the user's query. We retrieve the text that these 3 vectors are carrying and from it we build the prompt that we will use in our LLM. Now the LLM, together with the user's query and the context retrieved from the vector database, can generate an answer to the user's question.