

How to use a **local embedding model** from Hugging Face Transformers with **LangChain** and **FAISS** for Reading a text file with question and answering:-

🔗 Example: Using a local embedding model with LangChain + FAISS

Here's a complete working example using `sentence-transformers/all-MiniLM-L6-v2`, a small, fast, and good general-purpose embedding model.

First, install the required packages:

```
pip install sentence-transformers langchain faiss-cpu
```

```
pip install -U langchain-huggingface
```

🔗 Required Python Packages

🔗 1. langchain-core

Handles document and core abstractions (like `Document`).

```
bash
CopyEdit
pip install langchain-core
```

🔗 2. langchain-community

Includes `FAISS` and other integrations like file loaders.

```
bash
CopyEdit
pip install langchain-community
```

🔗 3. langchain-huggingface

This provides the **new, updated** `HuggingFaceEmbeddings`.

```
pip install langchain-huggingface
```

❓ 4. sentence-transformers

Required internally by `HuggingFaceEmbeddings` for loading the model.

```
pip install sentence-transformers
```

❓ 5. faiss-cpu

FAISS vector index library (CPU version).

```
pip install faiss-cpu
```

Python Code Example

```
from langchain_community.vectorstores import FAISS
from langchain_huggingface import HuggingFaceEmbeddings # ❓ Updated import
from langchain_core.documents import Document

# 1. Load the local embedding model
embedding = HuggingFaceEmbeddings(model_name="sentence-transformers/all-MiniLM-L6-v2")

# 2. Load text from a .txt file
file_path = "myfile.txt" # ❓ Change to your file path

with open(file_path, "r", encoding="utf-8") as f:
    text = f.read()

# Optional: Split into chunks (basic)
documents = [Document(page_content=chunk) for chunk in text.split("\n\n") if
chunk.strip()]

# 3. Create the FAISS vector store
vectorstore = FAISS.from_documents(documents, embedding)

# 4. Do a similarity search
query = "Tell me about machine learning"
results = vectorstore.similarity_search(query, k=2)

# 5. Print results
for i, doc in enumerate(results, 1):
```

```
print(f"Result {i}: {doc.page_content}")
```

output:-

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\python demo> & "C:/Users/Big Data/AppData/Local/Programs/Python/Python313/python.exe" "d:/python demo/langchain-reading.py"
Result 1: Machine learning is a field of artificial intelligence that uses statistical techniques to give computer systems the ability to learn from data.
Result 2: AI is transforming industries such as healthcare, finance, and transportation through automation and data-driven decisions.
PS D:\python demo>
```

myfile.txt file :-

Machine learning is a field of artificial intelligence that uses statistical techniques to give computer systems the ability to learn from data.

Natural language processing enables computers to understand, interpret, and respond to human language.

Data science combines domain expertise, programming skills, and knowledge of mathematics and statistics to extract meaningful insights from data.

Neural networks are computing systems inspired by the biological neural networks that constitute animal brains.

AI is transforming industries such as healthcare, finance, and transportation through automation and data-driven decisions.