Semantic-Search with SQLite Database

Goal: given a user input, find the desired entry from a SQLite database using semantic search

Using Datasette which supports full-text search and semantic search, details can be found at <https://docs.datasette.io/en/stable/full_text_search.html> and this tutorial here: <https://medium.com/@joseph.r.martinez/implement-semantic-search-with-datasette-b2fc20a6f1d9>

Approach:

1. Given the initial database here, calculate embeddings for DeviceID, Type, StartBandwidth, and StopBandwidth. For now, we’ll assume that the user will only ask for a device that operates at a certain wavelength, and we want to make sure that wavelength is within the (StartBandwidth ~ StopBandwidth) range图形用户界面, 应用程序

   描述已自动生成
2. Create metadata file to enable advanced functionality, details could be found here: <https://docs.datasette.io/en/stable/metadata.html>, the metadata file I used is here:

| {  "title": "Device Designs",  "description": "Full text and semantic search on all previous designs.",  "databases": {  "semantic": {  "queries": {  "semantic-search": {  "hide\_sql": false,  "sql": "select value from json\_each(faiss\_search\_with\_scores('semantic', 'device\_embeddings', (select openai\_embedding(:query, :openai\_api\_key)), 3)) where length(coalesce(:query, '')) > 0"  }  }  }  },    "plugins": {  "datasette-faiss": {  "tables": [  ["semantic", "device\_embeddings"]  ]  }  }    } |
| --- |

1. We use a datasette-faiss plugin to do similarity search calculations.

Test 1:

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

Test 2:

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

Test 3:

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

We can ask different questions and Datasette will return the output in the format of DeviceID (entry number) + confidence score.

From the example tests, we see that entries 3,2,8 are returned, and from the SQLite database, we can tell those 3 entries each represent:

| Entry Number | Device Name | Operating Wavelength Range |
| --- | --- | --- |
| 2 | 1x2 MMI | 1530nm-1565nm |
| 3 | 2x2 MMI | 1530nm-1565nm |
| 8 | 1x2 MMI | 1260nm-1360nm |

Observation:

1. In test 1, entry number 2 should be fetched if we asked for a mmi 1x2 that operates at a wavelength of 1550nm, however, entry number 8 has the highest confidence score, which has a different operating wavelength range
2. For test 3, even if we specify the wavelength range to be 1530nm-1565nm in the user query, the model still made mistakes.

Next steps:

1. Do more investigation on Datasette and see if we can customize models of better performance.
2. Try semantic search with other formats of data such as .csv or .json instead of sqlite