Untitled Notebook

Anonymous

2025 年 6 月 20 日

数据库 schema：
数据库 sentence.db schema 如下：

sentence(sid,sentence1, sentence2, similar_score,sen1_vector,sen2_vector,vecSim_score)

分别是句子 id，句子 1，句子 2，句子相似度得分，句子 1 嵌入向量，句子 2 嵌入向量，向量相似度得分）

任务要求：
调用嵌入模型：可以使用 sqlite-vss 或 sqlite-vec 或调用本地中文嵌入模型（huggingface 等拉取），对每个数据库中每个元组的两个句子分别进行向量嵌入，并计算向量相似度得分。更新数据库 sentence.db 的后三列。

输出要求：
按照这样的方法：

```
df = pd.read_sql_query("""
    SELECT sid, sentence1, sentence2, similar_score, sen1_vector,
sen2_vector,vecSim_score FROM sentence
""", conn)
print(df.head(20))
```

输出数据库的前 20 个元组。但请注意，助教评分时可能会运行你的代码，抽查其他元组结果。

[1]:
```
import sqlite3
import os
import pandas as pd
```

[2]:
```
# 文件路径修改成自己的文件路径
DATA_FILE = 'sts-b-train.txt'
DB_FILE = 'sentence.db'

if os.path.exists(DB_FILE):
    os.remove(DB_FILE)

conn = sqlite3.connect(DB_FILE)
cursor = conn.cursor()
```

```
cursor.execute('''
CREATE TABLE IF NOT EXISTS sentence (
    sid INTEGER PRIMARY KEY AUTOINCREMENT,
    sentence1 TEXT NOT NULL,
    sentence2 TEXT NOT NULL,
    similar_score REAL NOT NULL,
    sen1_vector BLOB,
    sen2_vector BLOB,
    vecSim_score REAL
)
```

```
''')
```

[3]:
```
conn.commit()
```

```
with open(DATA_FILE, 'r', encoding='utf-8') as f:
    for line in f:
        parts = line.strip().split('\t')
        if len(parts) != 3:
            continue
        sentence1, sentence2, score = parts
        cursor.execute('''
            INSERT INTO sentence (sentence1, sentence2, similar_score)
            VALUES (?, ?, ?)
        ''', (sentence1, sentence2, float(score)))
```

[4]:
```
conn.commit()
```

```
df = pd.read_sql_query("""
      SELECT sid, sentence1, sentence2, similar_score, sen1_vector, sen2_vector,
vecSim_score FROM sentence
""", conn)
```

[ ]:
```
print(df.head(10))
```

```
   sid          sentence1                 sentence2  similar_score  \
0    1         一架飞机要起飞了。               一架飞机正在起飞。            5.0
1    2       一个男人在吹一支大笛子。                 一个人在吹长笛。            3.0
2    3  一个人正把切碎的奶酪撒在比萨饼上。  一个男人正在把切碎的奶酪撒在一块未煮好
的比萨饼上。            3.0
3    4           三个人在下棋。                 两个人在下棋。            2.0
4    5         一个人在拉大提琴。           一个坐着的人正在拉大提琴。            4.0
5    6           有些人在战斗。                 两个人在打架。            4.0
6    7          一个男人在抽烟。                一个男人在滑冰。            0.0
7    8           那人在弹钢琴。                 那人在弹吉他。            1.0
8    9        一个男人在弹吉他和唱歌。       一位女士正在弹着一把原声吉他，唱着
歌。            2.0
9   10     一个人正把一只猫扔到天花板上。            一个人把一只猫扔在天花板
上。            5.0

  sen1_vector sen2_vector vecSim_score
0        None        None         None
1        None        None         None
2        None        None         None
3        None        None         None
4        None        None         None
5        None        None         None
6        None        None         None
7        None        None         None
8        None        None         None
9        None        None         None
```

```
from sentence_transformers import SentenceTransformer

# 加载模型，需要先下载到本地文件夹下
```

[6]:
```
model = SentenceTransformer('./text2vec-base-chinese')
```

```
c:\Users\vectorpikachu\AppData\Local\Programs\Python\Python312\Lib\site-
packages\tqdm\auto.py:21: TqdmWarning: IProgress not found. Please update jupyter
and ipywidgets. See https://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
```

[7]:
```
cursor = conn.cursor()

cursor.execute("SELECT sid, sentence1, sentence2 FROM sentence")
rows = cursor.fetchall()

# 提取句子和对应的 sid
sid_list = []
sentences1 = []
sentences2 = []

for row in rows:
    sid, s1, s2 = row
    sid_list.append(sid)
    sentences1.append(s1)
    sentences2.append(s2)

# 生成嵌入向量
embeddings1 = model.encode(sentences1, batch_size=64, show_progress_bar=True)
embeddings2 = model.encode(sentences2, batch_size=64, show_progress_bar=True)
```

```
Batches: 100%|██████████| 82/82 [00:44<00:00,  1.85it/s]
Batches: 100%|██████████| 82/82 [00:44<00:00,  1.85it/s]
```

[8]:
```
import numpy as np
def cosine_similarity(vec1, vec2):
    return float(np.dot(vec1, vec2) / (np.linalg.norm(vec1) * np.linalg.norm(vec2)))
```

[9]:
```
for sid, vec1, vec2 in zip(sid_list, embeddings1, embeddings2):
    # 计算余弦相似度
    sim_score = cosine_similarity(vec1, vec2)

    # 将向量转换为二进制格式
    vec1_blob = vec1.tobytes()
    vec2_blob = vec2.tobytes()

    # 更新数据库记录
    cursor.execute("""
        UPDATE sentence
        SET sen1_vector = ?, sen2_vector = ?, vecSim_score = ?
        WHERE sid = ?
    """, (vec1_blob, vec2_blob, sim_score, sid))

# 提交更改
conn.commit()
```

```
df = pd.read_sql_query("""
      SELECT sid, sentence1, sentence2, similar_score, sen1_vector, sen2_vector,
vecSim_score FROM sentence
""", conn)
```

[ ]:
```
print(df.head(20))
```

```
    sid           sentence1                  sentence2  similar_score  \
0     1          一架飞机要起飞了。                一架飞机正在起飞。            5.0
1     2       一个男人在吹一支大笛子。                 一个人在吹长笛。            3.0
2     3   一个人正把切碎的奶酪撒在比萨饼上。   一个男人正在把切碎的奶酪撒在一块未煮
好的比萨饼上。            3.0
3     4            三个人在下棋。                 两个人在下棋。            2.0
4     5         一个人在拉大提琴。           一个坐着的人正在拉大提琴。            4.0
5     6            有些人在战斗。                 两个人在打架。            4.0
6     7          一个男人在抽烟。                一个男人在滑冰。            0.0
7     8            那人在弹钢琴。                 那人在弹吉他。            1.0
8     9        一个男人在弹吉他和唱歌。          一位女士正在弹着一把原声吉他，唱着
歌。             2.0
9    10     一个人正把一只猫扔到天花板上。              一个人把一只猫扔在天花板
上。             5.0
10   11        那人用棍子打了另一个人。             那人用棍子打了另一个人一巴
掌。             4.0
11   12          一个人在吹长笛。                一个男人在吹竹笛。            3.0
12   13          一个人在叠一张纸。                有人在叠一张纸。            4.0
13   14  一条狗正试图把培根从他的背上弄下来。              一只狗正试图吃它背上的培
根。             3.0
14   15        北极熊在雪地上滑行。           一只北极熊在雪地上滑行。            5.0
15   16          一个女人在写作。                一个女人在游泳。            0.0
16   17       一只猫在婴儿的脸上摩擦。                一只猫在蹭婴儿。            3.0
17   18            那人在骑马。                一个人骑着马。            5.0
18   19          一个人往锅里倒油。              一个人把酒倒进锅里。            3.0
19   20          一个男人在弹吉他。                一个女孩在弹吉他。            2.0

                                        sen1_vector  \
0   b'tw\xbb\xbe\xbc=#>\xfc\x86\xac=lG→6\x8aw>wzU...
1   b'\t\x03\xa5\xbc\xf0\xdd\x92\xbe\xc1&\x86\xbf\...
2   b'\xac\xc1\n?\x0b\xbb\x0b?\x8dz\x1b\xbe\x83\xd...
3   b'\xd5C\xfd>T\xfa#\xbf\xac\xcd\x04?\x18\x1e\x9...
4   b'Q\x92\xde;;q\x04?\xb0\x1d\x83\xbe\xd3\x08\x9...
5   b'\x8c\xcf\xd5\xbe\xc0M\`\xbe\xe4d\xa9\xbeu\x19...
6   b'W|I\xbe\`\xfaz?Z\x1dq?zP\xab?\xb0@\x12\xbdH\x...
7   b's\xb8V?\xcbj\xf0=\x97\xeeh\xbe\x84|5?$h\xe7=...
8   b'\xce\x80\xe4>%\xaf\xf6>\x83\xbe7?\xc8H"?h\xa...
9   b'U1\xc5\xbd}p\x98=\x17\xdc\xef=jWR\xbe\xc7\xc...
10  b'&\x88\xc4={\x19\t\xbe\x19\xb9\x99\xbeI\xcd ?...
11  b'\x92c\x90\xbe\xd9\xb8\xb1\xbe\xdc/\x00\xbf>\`...
12  b'\x17B\xb9\xbe\xae:\x06>\x9c\x9c\x03?\xd6\xd5...
13  b'\xc8O\xc6>\x0e\xeaS\xbf\xb7=D>2\xba\xaa\xbev...
14  b'K\xb2P\xbd+\xbb\x08\xbf};\x88\xbe\xfc\xab\xc...
15  b'Q\xb1&?\xee\x0b\xef\xbd@Q(?\x85wb?\xc2U\xab\...
16  b'\x03M\xb4\xbd7\x9c\x9f\xbdn\xe0\x14>>\x81k\x...
17  b'\x13\xa3\xe0<\xb8\x0bX\xbfUG\x01?\xff\x15\x9...
18  b'\xb2\x8a\x19?]Q\xae?\xf9\x16-\xbd\xc7\x82^>$...
19  b'\x97\x85-?\xe2w&?\'\xa6\xfc>\x1c\xf62>\xbd\x...

                                        sen2_vector  vecSim_score
0   b'_\x96\x1e\xbf\x11_y>\xec\x88\x03?\xdc(\xf1=\...      0.956823
1   b'\x92c\x90\xbe\xd9\xb8\xb1\xbe\xdc/\x00\xbf>\`...      0.884109
```

```
2   b'\x16\x0c\x0b?\xc5\xbd\x0e?\x9aU:\xbe\xf4\xb8...      0.974107
3   b'\xf2&\xe4>\'{;\xbf\xd5]J?\xb7)S?\x079X?h\x84...      0.747162
4   b'P\xfe\x8f=\x11\x06?)\xa0\xb1\xbe\xf9\xb5\x8...      0.961783
5   b'\x12h\xe9\xbc\x93o\x9c\xbe)&_\xbe"\xd8\x17?\...      0.846043
6   b'\xba\xdb\xc1?\xa6P\xa9=\x9a\xcd\xc4\xbd\xf0m...      0.231173
7   b'IP\x07?B\xbb\xc5>\xc0;\xef>\xf5\x93\x87>i\xf...      0.484320
8   b'\xdbi>;\xdb\xa3<\xbb\xcb6\x85\xbd{\xdf\x8b?\...      0.734604
9   b'\xb8\xb8\x14\xbd\xca\x81\xc1\xbd2\x86\xc9=\x...      0.983240
10  b'@\xcd\x88>\xa2\xc2\x8c>\xd0\x18,\xbe\x9e\xcc...      0.909791
11  b']\xd0\xb6>\x89\x89\x1a=\xc3lq\xbf\x81\xf6\xc...      0.836796
12  b'\x1f\xd2\xaa\xbe\xf8\xa3j> \xb9\xfa>\x88\x9b...      0.979772
13  b'r\x93\xa8>\xfa\x1e\x95\xbew\xd9\xfa\xbe\x89\...      0.798483
14  b'\xae\xa9\xb7\xbc\x10\xb2\xda\xbe\x0e\x9d\x83...      0.985583
15  b'H\xf8\x95?Nm\xd8>\xa2L\x0e\xbf\xc3\x95\x8e>@...      0.425817
16  b'A\xac\xa7\xbea\xba=\xbf*A\x80\xbf\x9b=\xfd\x...      0.798582
17  b'=t\t=DIS\xbf\xb7E*>\x9b\xabw?\x8d\x1e)\xbe\x...      0.944069
18  b'8=E?D\x82\xdc?\xd5$\r\xbc\xad\x1f<>\xc9q\xe5...      0.687595
19  b'd\xaeA?\xf6X\xeb<PL\x94>\xcd\xa3\x9e>\xe9\xc...      0.706035
```

[11]:
```
conn.close()
```

提示：一些可能的嵌入向量的方式：

1.安装 sqlite-vss（基于 FAISS）

```
!pip install faiss-cpu
!pip install sqlite-vss

import sqlite3
import sqlite_vss
conn = sqlite3.connect("sentence.db")
sqlite_vss.load(conn)
```

2.安装 sqlite-vec(建议 python 版本>=3.9)

```
!pip install sqlite-vec


import sqlite3
import sqlite_vec
conn = sqlite3.connect("sentence.db")
sqlite_vec.load(conn)
```

3. 使用 hugging_face 等拉取嵌入模型

```
!pip install sentence-transformers
!pip install torch

model = HuggingFaceEmbeddings('YOU MODEL PATH')
```