Lab#4, NLP@CGU Spring 2023

This is due on 2023/04/20 16:00, commit to your github as a PDF (lab4.pdf) (File>Print>Save as PDF).

IMPORTANT: After copying this notebook to your Google Drive, please paste a link to it below. To get a publicly-accessible link, hit the *Share* button at the top right, then click "Get shareable link" and copy over the result. If you fail to do this, you will receive no credit for this lab!

LINK: paste your link here

https://colab.research.google.com/drive/1rqn1x6NgtvDSMQBAkDYGMWlHZXkWxtId?usp=share_link

---

Student ID: B0928026

Name: 洪詩晴

Word Embeddings for text classification

請訓練一個 kNN或是SVM 分類器來和 Google's Universal Sentence Encoder (a fixed-length 512-dimension embedding) 的分類結果比較

```
!wget -O Dcard.db https://github.com/cjwu/cjwu.github.io/raw/master/courses/nlp2023/lab4-Dcard-Dataset.db
```

```
--2023-04-24 05:31:47--  https://github.com/cjwu/cjwu.github.io/raw/master/courses/nlp2023/lab4-Dcard-Dataset.db
Resolving github.com (github.com)... 140.82.112.3
Connecting to github.com (github.com)|140.82.112.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/cjwu/cjwu.github.io/master/courses/nlp2023/lab4-Dcard-Dataset.db [following]
--2023-04-24 05:31:48--  https://raw.githubusercontent.com/cjwu/cjwu.github.io/master/courses/nlp2023/lab4-Dcard-Dataset.
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.109.133, 185.199.108.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 151552 (148K) [application/octet-stream]
Saving to: ‘Dcard.db’

Dcard.db            100%[===================>] 148.00K  --.-KB/s    in 0.008s

2023-04-24 05:31:48 (17.4 MB/s) - ‘Dcard.db’ saved [151552/151552]
```

```
import sqlite3
import pandas as pd

conn = sqlite3.connect("Dcard.db")
df = pd.read_sql("SELECT * FROM Posts;", conn)
df
```

	createdAt	title	excerpt	categories	
0	2022-03-04T07:54:19.886Z	專題需要數據🥺 🥺 幫填～	希望各位能花個20秒幫我填一下		
1	2022-03-04T07:42:59.512Z	#詢問 找衣服😢	想找這套衣服😢 ，但發現不知道該用什麼關鍵字找， （圖是草屯囝仔的校園演唱會截圖）	詢問	衣服 | 鞋子 | 衣物 |
2	2022-03-	#黑特 網購50% FIFTY	因為文會有點長，先說結論是，50%是目前網購過的平台退		黑特 | 網購 | 三思 |

```
!pip3 install -q tensorflow_text
!pip3 install -q faiss-cpu

     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.0/6.0 MB 37.5 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 17.0/17.0 MB 71.7 MB/s eta 0:00:00
```

```
import tensorflow_hub as hub
import numpy as np
import tensorflow_text
import faiss

embed_model = hub.load("https://tfhub.dev/google/universal-sentence-encoder-multilingual/3")
```

```
                    2022-03-   #問 阿神和放火發生過什麼    想問有沒有人知道阿神和放火是認識還是有結過什麼仇之類
```

```
docid = 355
texts = "[" + df['title'] + '] [' + df['topics'] + '] ' + df['excerpt']
texts[docid]
```

```
'[開了新頻道] [Youtuber | 頻道 | 有趣 | 日常 | 搞笑] 昨天上了第一支影片，之前有發過沒有線條的動畫影片，新的頻道改成有線條的，感覺大家好像比
內容主要是分享自己遇到的小故事，不知道這樣的頻道大家是否會想要看呢？喜歡的話也'

360 rows × 7 columns
```

```
embeddings = embed_model(texts)
embed_arrays = np.array(embeddings)
index_arrays = df.index.values
topk = 10
# Step 1: Change data type
embeddings = embed_arrays.astype("float32")

# Step 2: Instantiate the index using a type of distance, which is L2 here
index = faiss.IndexFlatL2(embeddings.shape[1])

# Step 3: Pass the index to IndexIDMap
index = faiss.IndexIDMap(index)

# Step 4: Add vectors and their IDs
index.add_with_ids(embeddings, index_arrays)

D, I = index.search(np.array([embeddings[docid]]), topk)

plabel = df.iloc[docid]['forum_zh']

cols_to_show = ['title', 'excerpt', 'forum_zh']
plist = df.loc[I.flatten(), cols_to_show]

precision = 0
for index, row in plist.iterrows():
  if plabel == row["forum_zh"]:
    precision += 1

print("precision = ", precision/topk)
precision = 0

df.loc[I.flatten(), cols_to_show]
```

```
precision =  0.8
```

Implemement Your kNN or SVM classifier Here!

請比較分類結果中選出 topk 相近的筆數，並計算 forum_zh 是否都有在 query text 的 forum_zh 中

> [開了新頻道] [Youtuber | 頻道 | 有趣 | 日常 | 搞笑]

```
docid = 355
texts = "[" + df['title'] + '] [' + df['topics'] + '] '
texts[docid]
```

```
'[開了新頻道] [Youtuber | 頻道 | 有趣 | 日常 | 搞笑] '
```

```
precision = 0
topk = 10

# YOUR CODE HERE!
# IMPLEMENTIG TRIE IN PYTHON









# # DO NOT MODIFY THE BELOW LINE!
print("precision = ", precision/topk)
```

```
precision =  0.0
```

```
class TrieNode:
    def __init__(self):
        self.children = {}
        self.is_end_of_word = False

class Trie:
    def __init__(self):
        self.root = TrieNode()

    def insert(self, word):
        current = self.root
        for char in word:
            if char not in current.children:
                current.children[char] = TrieNode()
            current = current.children[char]
        current.is_end_of_word = True

    def search(self, word):
        current = self.root
        for char in word:
            if char not in current.children:
                return False
            current = current.children[char]
        return current.is_end_of_word



trie = Trie()
for _, row in df.iterrows():
    trie.insert(row['forum_zh'])

precision = 0
for index, row in plist.iterrows():
    if trie.search(row['forum_zh']):
        precision += 1

print("precision = ", precision/topk)
```

```
precision =  1.0
```

```
class TrieNode:
    def __init__(self):
        self.children = {}
        self.is_end_of_word = False
```

```
class Trie:
    def __init__(self):
        self.root = TrieNode()

    def insert(self, word):
        current = self.root
        for char in word:
            if char not in current.children:
                current.children[char] = TrieNode()
            current = current.children[char]
        current.is_end_of_word = True

    def search(self, word):
        current = self.root
        for char in word:
            if char not in current.children:
                return False
            current = current.children[char]
        return current.is_end_of_word



trie = Trie()
for _, row in df.iterrows():
    trie.insert(row['forum_zh'])

precision = 0
for index, row in plist.iterrows():
    if trie.search(row['forum_zh']):
        precision += 1

print("precision = ", precision/topk)
```

```
    precision =  1.0
```

```
import json
import random
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report

movie_list = hub.load("https://tfhub.dev/google/universal-sentence-encoder-multilingual/3")

train_data = []
train_labels = []
test_data = []
test_labels = []
for i, movie in enumerate(movie_list):
    if movie['forum_zh'] is not None and movie['excerpt'] is not None:
        if i < 200: # train
            train_data.append(movie['excerpt'])
            train_labels.append(movie['forum_zh'])
        elif i > 200 and i < 300: # test
            test_data.append(movie['intro'])
            test_labels.append(movie['forum_zh'])

combined = list(zip(train_data, train_labels)) # 打亂順序
random.shuffle(combined)
train_data[:], train_labels[:] = zip(*combined)

vectorizer = CountVectorizer() # 特徵向量 TD-IDF
tfidf_transformer = TfidfTransformer()
X_train = tfidf_transformer.fit_transform(vectorizer.fit_transform(train_data))
y_train = train_labels

clf = SVC(kernel='linear')
clf.fit(X_train, y_train)

X_test = tfidf_transformer.transform(vectorizer.transform(test_data))
y_pred = clf.predict(X_test)

accuracy = accuracy_score(test_labels, y_pred)
print(f"Prediction Precision: {accuracy:.2%}")
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-21-809520af90f8> in <cell line: 13>()
     11 test_data = []
     12 test_labels = []
```

```
import json
import random
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report
import tensorflow_hub as hub

movie_list = hub.load("https://tfhub.dev/google/universal-sentence-encoder-multilingual/3")

train_data = []
train_labels = []
test_data = []
test_labels = []
for i, movie in enumerate(movie_list):
    if movie['forum_zh'] is not None and movie['excerpt'] is not None:
        if i < 200: # train
            train_data.append(movie['excerpt'])
            train_labels.append(movie['forum_zh'])
        elif 200 <= i < 300: # test
            test_data.append(movie['excerpt'])
            test_labels.append(movie['forum_zh'])

combined = list(zip(train_data, train_labels)) # 打亂順序
random.shuffle(combined)
train_data[:], train_labels[:] = zip(*combined)

vectorizer = CountVectorizer() # 特徵向量 TD-IDF
tfidf_transformer = TfidfTransformer()
X_train = tfidf_transformer.fit_transform(vectorizer.fit_transform(train_data))
y_train = train_labels

clf = SVC(kernel='linear')
clf.fit(X_train, y_train)

X_test = tfidf_transformer.transform(vectorizer.transform(test_data))
y_pred = clf.predict(X_test)

accuracy = accuracy_score(test_labels, y_pred)
print(f"Prediction Precision: {accuracy:.2%}")
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-22-507fe8756834> in <cell line: 14>()
     12 test_data = []
     13 test_labels = []
---> 14 for i, movie in enumerate(movie_list):
     15     if movie['forum_zh'] is not None and movie['excerpt'] is not None:
     16         if i < 200: # train

TypeError: '_UserObject' object is not iterable
```

SEARCH STACK OVERFLOW

5 秒 完成時間：下午3:16