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/142CPj6K-3HkyXN9q_r2_wkSQ7l7Z-aZs?usp=sharing

---

Student ID:B0928012

Name:王晟翰

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 07:42:53--  https://github.com/cjwu/cjwu.github.io/raw/master/courses/nlp2023/lab4-Dcard-Dataset.db
Resolving github.com (github.com)... 140.82.114.3
Connecting to github.com (github.com)|140.82.114.3|: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 07:42:53--  https://raw.githubusercontent.com/cjwu/cjwu.github.io/master/courses/nlp2023/lab4-Dcard-Dataset.db
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.109.133, 185.199.111.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|: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 0.03s

2023-04-24 07:42:54 (5.13 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	topics	forum_en	forum_zh
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			dressup	穿搭
	2022-03-		想找這套衣服□，但發現不知道該用什麼		衣服 | 鞋子 | 衣物		

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

因為文會有點長，先

```
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")
```

04T06:39:13.017Z 穿搭 | 男生穿搭

```
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，感覺大家好像比較喜歡這種風格，試試看新的風格，影片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'
```

... ...

```
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
```

	title	excerpt	forum_zh
355	開了新頻道	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，感覺大家好像比較喜歡...	YouTuber
359	一個隨性系YouTube頻道	哈哈哈哈，沒錯我就是親友團來介紹一個我覺得很北七的頻道，現在觀看真的低的可憐，也沒事啦，就多...	YouTuber
330	《庫洛魔法使》（迷你）服裝製作	又來跟大家分享新的作品了~，頻道常常分享 {縫紉} {服裝製作} 等相關教學，大家對服裝製...	YouTuber
342	自己沒搞清楚狀況就不要亂黑勾惡	勾惡幫主在自己頻道簡介跟每部影片的下方都已經說明了，要分會會長以上才能看全部影片，這個說明已...	YouTuber
338	廚師系YouTuber	友人傳了這篇文給我，我一看，十大廚師系YouTuber，就猜一定有MASA，果不其然，榜上有...	YouTuber
243	毀我童年的家人	小時候都很喜歡看真珠美人魚和守護甜心，但是！！，每次晚餐看電視的時候，只要有播映到這種場景....	有趣
349	喜歡看寵物頻道的有嗎？		YouTuber

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]

```
precision = 0
topk = 10

# YOUR CODE HERE!
# IMPLEMENTIG TRIE IN PYTHON


from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn import svm

# 建立TfidfVectorizer向量化器，用來轉換文本資料為向量表示
vectorizer = TfidfVectorizer()

# 將原始文本資料轉換為TF-IDF向量
X = vectorizer.fit_transform(texts).toarray()

# 建立SVM分類器模型
clf = svm.SVC()

# 訓練SVM分類器模型
clf.fit(X, df['forum_zh'])

# 設定查詢文本ID
docid = 355

# 使用SVM分類器模型對查詢文本進行預測
predicted_forum = clf.predict(X[docid].reshape(1, -1))

# 將預測結果與查詢文本的討論區名稱比較，計算預測結果中與查詢文本討論區相同的文件所佔的比例
cols_to_show = ['title', 'excerpt', 'forum_zh']
plist = df.loc[I.flatten(), cols_to_show]
precision = 0
for index, row in plist.iterrows():
        if predicted_forum[0] == row['forum_zh']:
                precision += 1

# 計算精度
# precision /= topk

# 計算查詢文本的討論區是否都有出現在預測結果中
query_forum = df.iloc[docid]['forum_zh']
predicted_forums = set(clf.predict(X[I.flatten()]))
all_forum_in_predicted = query_forum in predicted_forums









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

```
     precision =  0.8
```

改成用KNN做做看

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neighbors import KNeighborsClassifier

# 建立TfidfVectorizer向量化器，用來轉換文本資料為向量表示
vectorizer = TfidfVectorizer()

# 將原始文本資料轉換為TF-IDF向量
X = vectorizer.fit_transform(texts).toarray()

# 建立KNN分類器模型
clf = KNeighborsClassifier(n_neighbors=5)
```

```
# 訓練KNN分類器模型
clf.fit(X, df['forum_zh'])

# 設定查詢文本ID
docid = 355

# 使用KNN分類器模型對查詢文本進行預測
predicted_forum = clf.predict(X[docid].reshape(1, -1))

# 將預測結果與查詢文本的討論區名稱比較，計算預測結果中與查詢文本討論區相同的文件所佔的比例
cols_to_show = ['title', 'excerpt', 'forum_zh']
plist = df.loc[I.flatten(), cols_to_show]
precision = 0
for index, row in plist.iterrows():
    if predicted_forum[0] == row['forum_zh']:
        precision += 1

# 計算精度
precision /= topk

# 計算查詢文本的討論區是否都有出現在預測結果中
query_forum = df.iloc[docid]['forum_zh']
predicted_forums = set(clf.predict(X[I.flatten()]))
all_forum_in_predicted = query_forum in predicted_forums

# 顯示結果
print("precision = ", precision)
# print("All•query•forum•in•predicted•forums:",•all_forum_in_predicted)
```

```
precision =  0.8
```

✓ 0秒 完成時間：下午3:44