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/xxxxxxxxx

```
# 此內容會顯示為程式碼
```

Student ID:B0928006

Name: 廖采葳

Word Embeddings for text classification

訓練使一個 LSTM 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 07:32:07--  https://github.com/cjwu/cjwu.github.io/raw/master/courses/nlp2023/lab4-Dcard-Dataset.db
Resolving github.com (github.com)... 140.82.112.4
Connecting to github.com (github.com)|140.82.112.4|: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 07:32:07--  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, 185.199.111.133, 185.199.109.133, ...
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 0.02s

2023-04-24 07:32:07 (6.67 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_z
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 想找這套衣服□，			dressup	穿搭

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

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.0/6.0 MB 48.3 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 17.0/17.0 MB 70.8 MB/s eta 0:00:00
```

校園演唱

```
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-	50% FIFTY					

```
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 樣的頻道大家是否會想要看呢？喜歡的話也，

來源：嚮

儲存成功!

```
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]

```
330 [illegible] YouTuber
```

```
precision = 0
topk = 10

# kNN
from sklearn.neighbors import KNeighborsClassifier

#train
X_train = embeddings
y_train = df["forum_zh"]


knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)

predicted_labels = knn.predict(embeddings[I.flatten()])

for i, label in enumerate(predicted_labels):
    if label == plabel:
        precision += 1
```

儲存成功!

```
()]["forum_zh"].values.tolist()
# if plabel in searched_labels:
#     print(f"{plabel} is in the searched labels.")
# else:
#     print(f"{plabel} is not in the searched labels.")



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