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/1N7XrpKbGoHqi-BslnXy7g9ugTFnqBWMK?usp=sharing

Student ID: B0928001 **Name**: 賴霆瑞

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 06:53:01--  https://github.com/cjwu/cjwu.github.io/raw/master/courses/nlp2023/lab4-Dcard-Dataset.
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 [
--2023-04-24 06:53:02--  https://raw.githubusercontent.com/cjwu/cjwu.github.io/master/courses/nlp2023/lab4-Dca
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.1
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|: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 06:53:02 (9.31 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
```

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
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 | 男生穿搭 | 尋找	dressup
2	2022-03-04T07:24:25.147Z	#黑特 網購 50% FIFTY PERCENT 請三思	因為文會有點長，先說結論是，50%是目前網購過的平台退貨最麻煩的一家，甚至我認為根本是刻意刁...		黑特 | 網購 | 三思 | 退貨 | 售後服務	dressup
3	2022-03-04T06:39:13.017Z	尋衣服	來源：覺得呱吉這襯衫好好看~~，或有人知道有類似的嗎		衣服 | 尋找 | 日常穿搭 | 男生穿搭	dressup
4	2022-03-04T06:28:06.137Z	#詢問 想問	各位，因為這個證件夾臺灣買不到，是美國 outlet 的限量版貨，所以在以下的這間蝦皮上買，但...	詢問	穿搭 | 閒聊版 | 閒聊排解 | 假貨	dressup
...	...	...	...	...	...	...
			昨天卜了			

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

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.0/6.0 MB 15.1 MB/s eta
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 17.0/17.0 MB 73.6 MB/s e
```

```
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")
```

```
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

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]

小時候都很喜歡看真珠美人魚和守護甜心，但是！！，每

```
precision  =  0
topk  =  10

#  YOUR  CODE  HERE!
#  IMPLEMENTIG  TRIE  IN  PYTHON
```

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

```
precision = 0.8
```

```
!pip  install  scikit-learn

import  sqlite3
import  pandas  as  pd
import  numpy  as  np
import  tensorflow_hub  as  hub
from  sklearn.svm  import  LinearSVC
from  sklearn.pipeline  import  make_pipeline
from  sklearn.feature_extraction.text  import  TfidfVectorizer

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

#  Combine  text  fields
texts  =  "["  +  df['title']  +  ']  ['  +  df['topics']  +  ']  '  +  df['excerpt']

#  Create  label  and  target  arrays
labels  =  df['forum_zh'].values
targets  =  np.zeros(len(labels),  dtype=np.int8)
unique_labels  =  np.unique(labels)
for  i,  label  in  enumerate(labels):
        targets[i]  =  np.where(unique_labels  ==  label)[0][0]

#  Train  SVM  classifier
vectorizer  =  TfidfVectorizer()
svm_clf  =  make_pipeline(vectorizer,  LinearSVC())
svm_clf.fit(texts,  targets)
```

```
svm_clf.fit(texts,  targets)

#  Encode  input  text  with  Universal  Sentence  Encoder
embed_model  =  hub.load("https://tfhub.dev/google/universal-sentence-encoder-multilingual/3")
embeddings  =  embed_model(texts)
embed_arrays  =  np.array(embeddings)

#  Set  up  Faiss  index
index_arrays  =  df.index.values
embeddings  =  embed_arrays.astype("float32")
index  =  faiss.IndexFlatL2(embeddings.shape[1])
index  =  faiss.IndexIDMap(index)
index.add_with_ids(embeddings,  index_arrays)

#  Query  index  and  compare  results  with  SVM  classifier
docid  =  355
plabel  =  df.iloc[docid]['forum_zh']
query_text  =  texts[docid]

svm_pred  =  svm_clf.predict([query_text])[0]

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

precision  =  0
for  index,  row  in  df.loc[I.flatten()].iterrows():
        if  plabel  ==  row["forum_zh"]  and  svm_pred  ==  np.where(unique_labels  ==  row["forum_zh"])[0][0]:
                precision  +=  1

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

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.9/dist-packages (1.2.2)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.9/dist-packages (from scikit-learn) (1.1
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.9/dist-packages (from scikit-lea
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.9/dist-packages (from scikit-learn) (1.
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.9/dist-packages (from scikit-learn) (1.
precision =  0.8
```