本論文爬蟲的程式碼參考：

https://cloud.tencent.com/developer/article/1339809

https://kknews.cc/code/mbq4lpp.html

https://mgleon08.github.io/blog/2018/11/19/instagram-crawler-with-api/

本階段程式分為三個python檔案

getUsersData.py 取得網紅的資料，與圖片、影片的標籤

getWord2VecScore.py 利用訓練好的Word2Vec model計算網紅權重

getWordNetScore.py 利用WordNet計算網紅權重

getUsersData.py

1. 先至Google Cloud Platform 申請使用Google Cloud Vison API、Google Cloud Video Intelligence API，並下載使用API的驗證檔案（.json）。

2. 依照Google的API說明文件，安裝所需要的python套件

3. 開啟getUsersData.py修改參數

# 需要爬的網紅 變數資料型態必須為List

usersName = ["user1", "user2"]

# 需要爬取的貼文時間

newestTime = 1584028799 # 本論文設定在2020/03/11 23:59:59

# 最新貼文數量

maxPost = 320

# 取得Label的API最低分數

APIMinScore = 0.7

# 所有資料將儲存的資料夾

storeFolder = 'D:\\instagram'

# 儲存網紅貼文資料的json檔案 包含貼文中的標籤

usersDataFile = "D:\\instagram\\usersData.json"

# Django網頁平台讀取100張圖片的url

usersUrlsFile = "D:\\instagram\\usersUrls20200312.json"

# API的金鑰憑證json檔的路徑

credential\_path = 'D:\\APIKey.json'

4. 執行getUsersData.py

產出usersData.json

包括使用者的當前追蹤人數，每則貼文的按讚、留言、影像標籤、

getWord2VecScore.py

1. 設定參數

# 需要被計算的分類

myTypes = ['animal', 'vehicle',  'food', 'fashion', 'dog', 'cat', 'car', 'motorcycle']

# 計算完網紅權重存放的位置

scorePath = "D:\\instagram\\score"

# getUsersData.py儲存網紅貼文資料的json檔案，拿來計算分數

usersDataFile = "D:\\instagram\\usersData.json"

with open(usersDataFile, 'r') as load\_f:

    usersData = json.load(load\_f)

# 訓練好的Word2Vec model的存放位置，該位置只能放Word2Vec Model，否則會拋出例外

modelPath = "D:\\word2vec\\model\\"

2. 執行getWord2VecScore.py

產出每個Word2Vec Model計算出的分數

每個json檔包含網紅當前的追蹤人數，還有在每50倍數則貼文在每個分類的權重、平均按讚、留言。

getWordNetScore.py

1. 設定參數

# 需要被計算的分類

myTypes = ['animal', 'vehicle',  'food', 'fashion', 'dog', 'cat', 'car', 'motorcycle']

# 計算完網紅權重存放的位置

scorePath = "D:\\instagram\\score"

# getUsersData.py儲存網紅貼文資料的json檔案，拿來計算分數

usersDataFile = "D:\\instagram\\usersData.json"

with open(usersDataFile, 'r') as load\_f:

    usersData = json.load(load\_f)

2. 執行getWordNetScore.py

產出WordNet計算出的分數

每個json檔包含網紅當前的追蹤人數，還有在每50倍數則貼文在每個分類的權重、平均按讚、留言。

其他檔案是畫圖用的檔案

檔案參數自行調整即可使用

先執行getUserScore100.py

之後看需求執行

pt.py

ptAll.py

ptSingle.py

ptType.py