分享連結：
https://colab.research.google.com/drive/1DxfUN3ah7XVvainYDDdVbW7DGcR1BZKd?usp=sharing

安裝相關套件與字體

```
pip install jieba
```

```
    Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/cola
    Requirement already satisfied: jieba in /usr/local/lib/python3.9/dist-packa
```

```
pip install zhon
```

```
    Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/cola
    Requirement already satisfied: zhon in /usr/local/lib/python3.9/dist-packag
```

```
## 下載中文字體
!wget -O TaipeiSansTCBeta-Regular.ttf https://drive.google.com/uc?id=1eGAsTN1HBp
```

```
    --2023-03-16 07:45:05--  https://drive.google.com/uc?id=1eGAsTN1HBpJAkeVM57
    Resolving drive.google.com (drive.google.com)... 172.253.63.100, 172.253.63
    Connecting to drive.google.com (drive.google.com)|172.253.63.100|:443... co
    HTTP request sent, awaiting response... 303 See Other
    Location: https://doc-0k-9o-docs.googleusercontent.com/docs/securesc/ha0ro9
    Warning: wildcards not supported in HTTP.
    --2023-03-16 07:45:06--  https://doc-0k-9o-docs.googleusercontent.com/docs/
    Resolving doc-0k-9o-docs.googleusercontent.com (doc-0k-9o-docs.googleuserco
    Connecting to doc-0k-9o-docs.googleusercontent.com (doc-0k-9o-docs.googleus
    HTTP request sent, awaiting response... 200 OK
    Length: 20659344 (20M) [application/x-font-ttf]
    Saving to: ‘TaipeiSansTCBeta-Regular.ttf’

    TaipeiSansTCBeta-Re 100%[===================>]  19.70M  --.-KB/s    in 0.1s

    2023-03-16 07:45:06 (149 MB/s) - ‘TaipeiSansTCBeta-Regular.ttf’ saved [2065
```

取得原始資料與資料預處理

```
## 取得原始文字黨 & 去除標點符號
import requests
import string
from zhon.hanzi import punctuation

url = 'https://raw.githubusercontent.com/cjwu/cjwu.github.io/master/courses/nlp/
response = requests.get(url)
data = response.text

data = data.replace(' ', '')
data = data.replace('\t', '')
for i in string.punctuation:
    data = data.replace(i, '')
for i in punctuation:
    data = data.replace(i, '')
spacial_punctuation = ['▂', '▬', '︴', '←', '┌', '︿', '─', '∷', '﹎', '︶', '☁',
for i in spacial_punctuation:
    data = data.replace(i, '')
```

```
## 切分文章 & 統計文章總數總數（一行視為一文章）
articles = data.split('\n')
lines = len(articles)
print("文章數：", lines)
```

```
    文章數： 418203
```

```
## 斷詞
import jieba

seg_articles = []

for article in articles:
    # seg_article => list([article], article_length)
    seg_articles.append((jieba.lcut(article), len(article)))
print(seg_articles[:4])
```

```
    [(['為', '什麼', '聖結', '石會', '被', '酸', '而', '這群', '人', '不會質', '感劇
```

計算IDF權重

```
## 計算idf
import math
from collections import Counter


iDFs = {}

for article in seg_articles:
    counter = Counter(article[0])
    for item in counter.items():
        exist_idf = iDFs.get(item[0])
        if exist_idf:
            iDFs.update({item[0]: exist_idf + item[1]})
        else:
            iDFs[item[0]] = item[1]

for iDF in iDFs.items():
    iDFs[iDF[0]] = math.log(lines/iDF[1], 10)

## 依照idf權重排列
lt_iDFs = sorted(iDFs.items(), key=lambda item:item[1], reverse=True)
lt_iDFs = lt_iDFs

dict_iDFs = {}
for lt in lt_iDFs:
    dict_iDFs[lt[0]] = lt[1]
```

計算TF

重複單詞（計算單詞詞頻時，不同文章同單詞有複數詞頻）

```
## 計算tf & 依照tf排列
N = 0
rtps = []
for article in seg_articles:
    counter = Counter(article[0])
    for item in counter.items():
        rtps.append((N , item[0], item[1] / article[1]))
        N += 1

rtps = sorted(rtps, key=lambda item:item[2], reverse=True)
print(rtps[:10])
```

```
    [(2843002, '咩', 1.0), (5148798, '人', 1.0), (5918547, '人', 1.0), (1814120,
## 計算tf-idfs
rtf_idfs = []
for item in rtps:
    rtf_idfs.append((item[0], item[1], item[2] * iDFs[item[1]]))

rtf_idfs = sorted(rtf_idfs, key=lambda item:item[2], reverse=True)
print(rtf_idfs[:10])
```

```
[(1467897, '鞥', 3.3651310367640437), (2865096, '噢', 3.0349711959673664), (
```

不重複單詞（計算單詞詞頻時，不同文章同單詞只保留最大的詞頻)

[ ] ↳ *2 個隱藏的儲藏格*

繪圖

```
import matplotlib as mpl
import matplotlib.pyplot as plt
from matplotlib.font_manager import fontManager

fontManager.addfont('TaipeiSansTCBeta-Regular.ttf')
mpl.rc('font', family='Taipei Sans TC Beta')
```

重複單詞

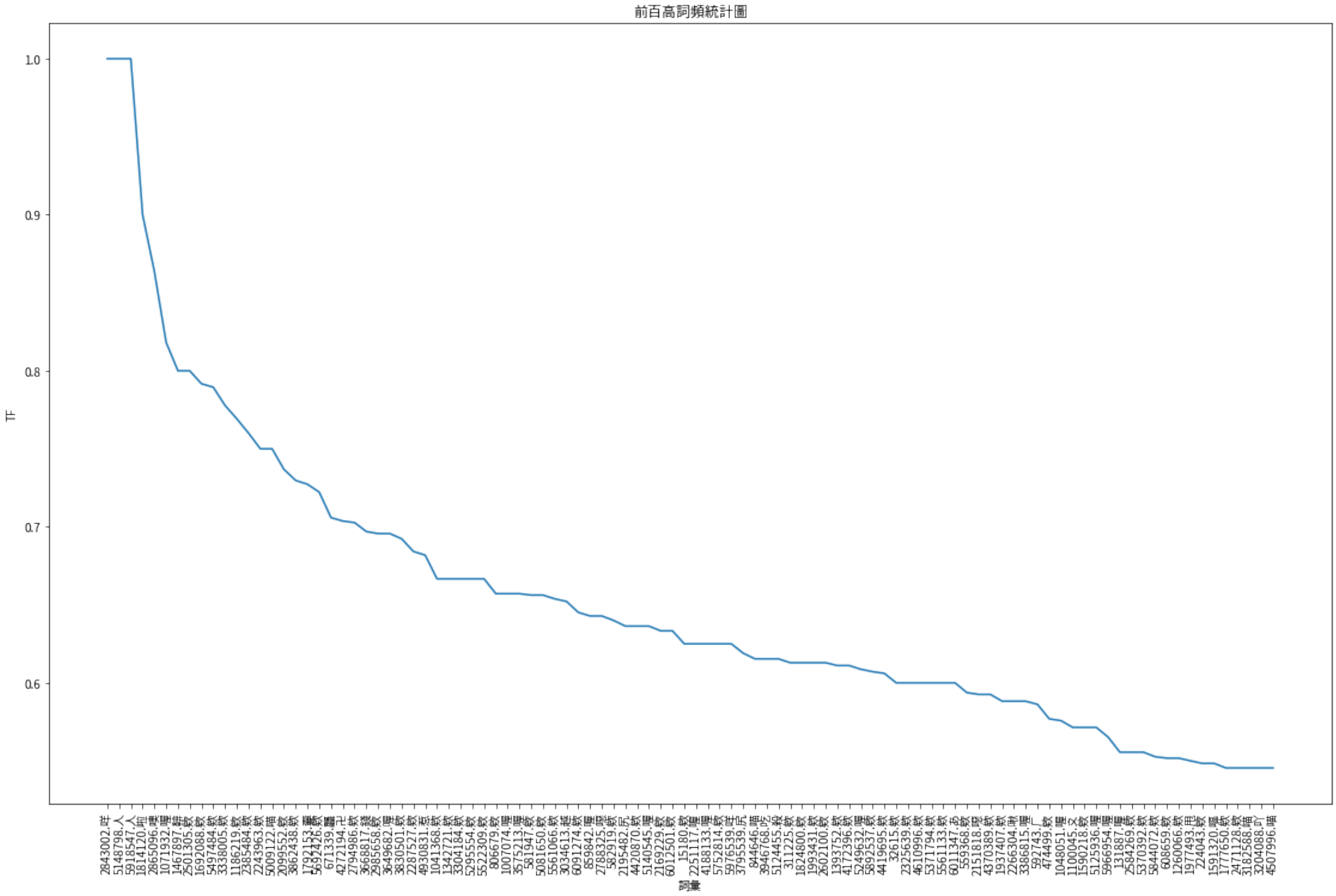
```
## 詞頻統計圖
x_axis = []
y_axis = []

for item in rtps[:100]:
    x_axis.append(str(item[0]) + '.' + item[1])
    y_axis.append(item[2])

print(">>> 前百高TF統計圖\n")

plt.figure(figsize = (19.2 , 12))
plt.plot(x_axis, y_axis)
plt.title("前百高詞頻統計圖")
plt.xlabel("詞彙")
plt.xticks(rotation = 90)
plt.ylabel("TF")
plt.show()

    >>> 前百高TF統計圖
```



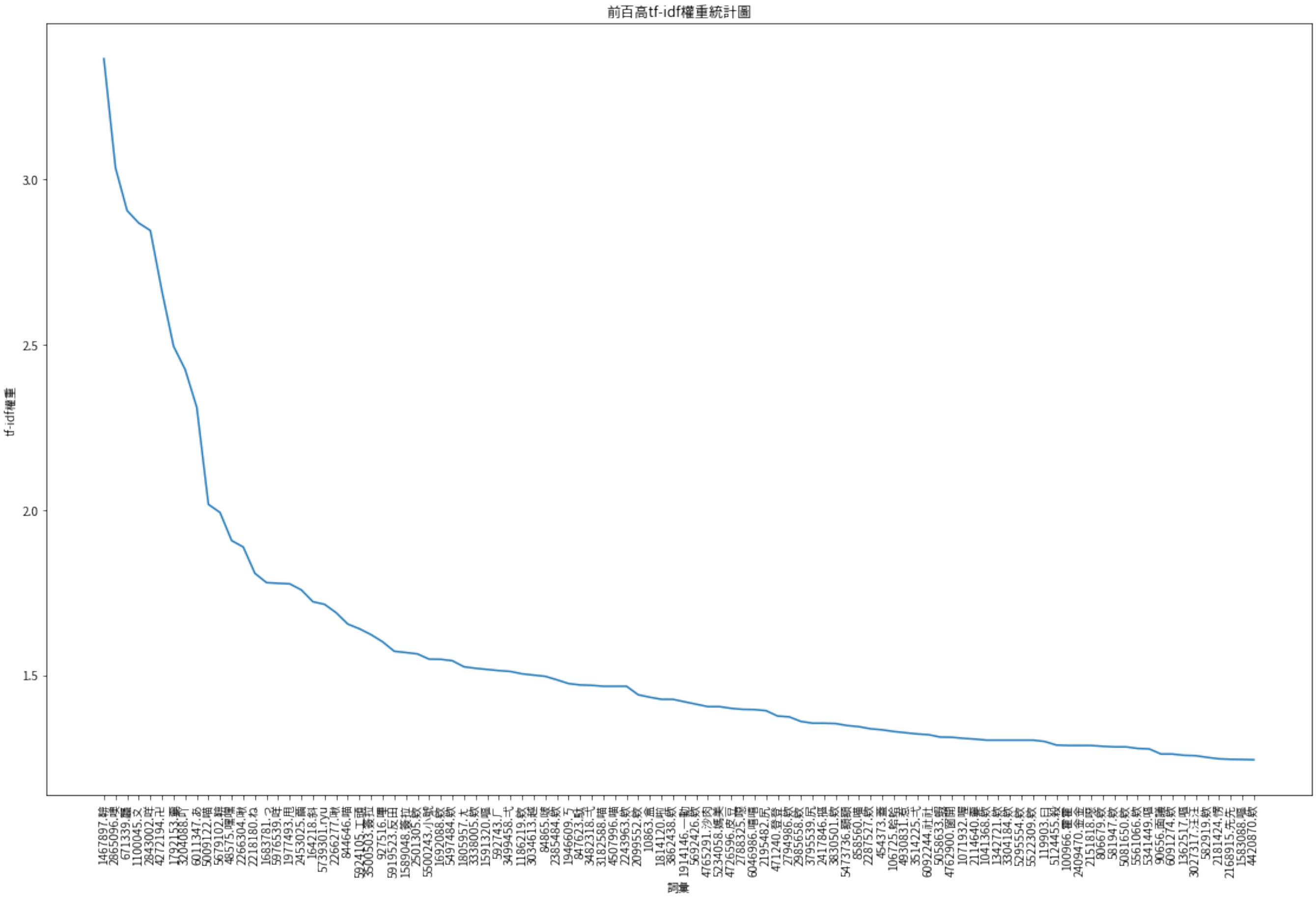
```
## tf-idf統計圖
x_axis = []
y_axis = []

for item in rtf_idfs[:100]:
    x_axis.append(str(item[0]) + '.' + item[1])
    y_axis.append(item[2])

print(">>> 前百高TF-IDF權重統計圖\n")

plt.figure(figsize = (19.2 , 12))
plt.plot(x_axis, y_axis)
plt.title("前百高tf-idf權重統計圖")
plt.xlabel("詞彙")
plt.xticks(rotation = 90)
plt.ylabel("tf-idf權重")
plt.show()
```

>>> 前百高TF-IDF權重統計圖

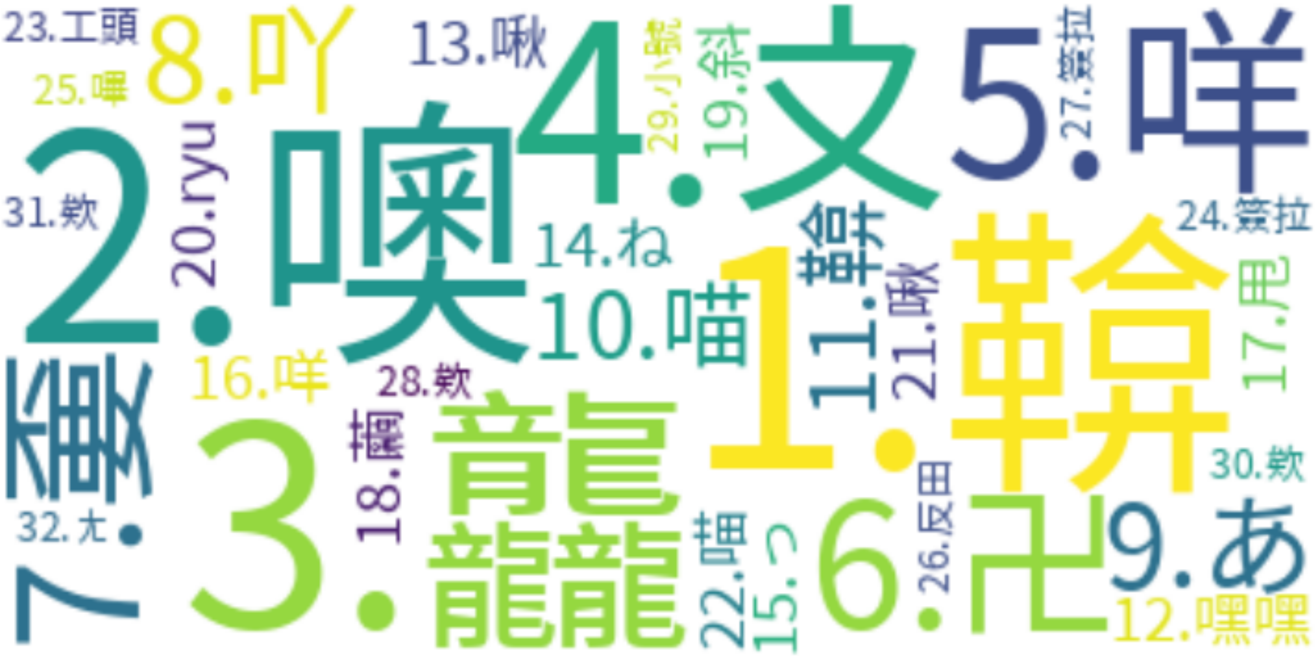


```
from wordcloud import WordCloud, STOPWORDS
X = 1
freq = {}
for l in rtf_idfs[:32]:
    freq[str(X) + '.' + l[1]] = l[2]
    X += 1

print(">>> 文字雲\n")

wordcloud = WordCloud(background_color="white", contour_width=3, contour_color='
plt.figure(figsize = (12.8 , 8))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```

```
>>> 文字雲
```



詞頻統計圖 (不重複單詞)

[ ] ↳ *3 個隱藏的儲藏格*

✓ 1 秒 完成時間：下午3:57 ● ×