```
import  jieba
import  jieba.analyse
import  requests
import  re


url  =  "https://raw.githubusercontent.com/cjwu/cjwu.github.io/master/courses/nlp/hw1-dataset.txt"
data  =  requests.get(url)
data  =  data.text

pattern  =  re.compile(r'[\s+\.\!\/_,$%^*(+\"\']+|[+——！，。？、~@#￥%……&*（）：]+')
data  =  re.sub(pattern,  '',  data)


words  =  jieba.cut(data)





word_count•=•{}
for•word•in•words:
••••if•len(word)•>•1:••
        word_count[word]  =  word_count.get(word,  0)  +  1

top_words_freq  =  sorted(word_count.items(),  key=lambda  x:  x[1],  reverse=True)[:100]
cloud_list  =  []
print("Top  100  words  by  frequency:")
for  word,  count  in  top_words_freq:
        cloud_list.append(word)
        print(f"{word}:  {count}")

tags  =  jieba.analyse.extract_tags(data,  topK=100,  withWeight=True,  allowPOS=())

print("\nTop  100  words  by  TF-IDF  weight:")
for  word,  weight  in  tags:
        print(f"{word}:  {weight}")
```

```
東西: 0.013103044028819884
這個: 0.013035855468815642
```

```
新聞: 0.007700848576481835
妹妹: 0.007592168417943672
鄉民: 0.007456807600466566
XD: 0.00736529223446084
一直: 0.00734561086605549
最強: 0.006846705160428393
ptt: 0.006799252748425424
機會: 0.006582327436411851
兩個: 0.006545043398409518
結婚: 0.006528096108408457
```

```
import matplotlib.pyplot as plt


top_words_tfidf = sorted(tags, key=lambda x: x[1], reverse=True)[:100]

word_indices = list(range(1, 101))
word_counts = [count for _, count in top_words_freq]
tfidf_weights = [weight for _, weight in top_words_tfidf]

fig1, ax1 = plt.subplots(figsize=(20, 10))
ax1.bar(word_indices, word_counts)
ax1.set_xticks(word_indices)
ax1.set_xticklabels([word for word, _ in top_words_freq], rotation=90, fontsize=12)
ax1.set_xlabel('Words', fontsize=14)
ax1.set_ylabel('Frequency', fontsize=14)
ax1.set_title('Top 100 words by frequency', fontsize=18)

fig2, ax2 = plt.subplots(figsize=(20, 10))
ax2.bar(word_indices, tfidf_weights)
ax2.set_xticks(word_indices)
ax2.set_xticklabels([word for word, _ in top_words_tfidf], rotation=90, fontsize=12)
ax2.set_xlabel('Words', fontsize=14)
ax2.set_ylabel('TF-IDF Weight', fontsize=14)
ax2.set_title('Top 100 words by TF-IDF weight', fontsize=18)

plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-51-e34e493a1ab0> in <module>
      4 top_words_tfidf = sorted(tags, key=lambda x: x[1], reverse=True)[:100]
      5 
----> 6 word_indices = list(range(1, 101))
      7 word_counts = [count for _, count in top_words_freq]
      8 tfidf_weights = [weight for _, weight in top_words_tfidf]

TypeError: 'list' object is not callable
```

SEARCH STACK OVERFLOW

```
from wordcloud import WordCloud

text = cloud_list[:33]
my_str = ''.join(text)
wc = WordCloud(font_path='font.ttf', background_color='white', width=800, height=600)
wc.generate(my_str)

plt.imshow(wc)
plt.axis('off')
plt.show()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-53-3d7f1ce48c58> in <module>
      4 my_str = ''.join(text)
      5 wc = WordCloud(font_path='font.ttf', background_color='white', width=800, height=600)
----> 6 wc.generate(my_str)
      7 
      8 plt.imshow(wc)

                              ⬍ 2 frames

/usr/local/lib/python3.9/dist-packages/wordcloud/wordcloud.py in generate_from_frequencies(self, frequencies,
max_font_size)
    408         frequencies = sorted(frequencies.items(), key=itemgetter(1), reverse=True)
    409         if len(frequencies) <= 0:
--> 410             raise ValueError("We need at least 1 word to plot a word cloud, "
    411                              "got %d." % len(frequencies))
    412         frequencies = frequencies[:self.max_words]

ValueError: We need at least 1 word to plot a word cloud, got 0.
```

SEARCH STACK OVERFLOW

新增區段

Colab 付費產品 - 按這裡取消合約

❗ 0 秒 完成時間：上午11:28 ● ✕