

function

```
In [53]: def load_dict(filepath: str) -> list:
    dict = []
    with open(filepath, "r", encoding="utf-8") as f:
        for line in f:
            word = line.strip()
            if word:
                dict.append(word)
    return dict

posemo = load_dict("dict_posemo.txt")
negemo = load_dict("dict_negemo.txt")

print(f"Loaded {len(posemo)} words")
print(posemo[:20])

print(f"Loaded {len(posemo)} positive words, {len(negemo)} negative words")
```

Loaded 620 words

['accept', 'accepta*', 'accepted', 'accepting', 'accepts', 'active', 'actively', 'admir*', 'ador*', 'advantag*', 'adventur*', 'affection*', 'agree', 'agreeable', 'agreeableness', 'agreeably', 'agreed', 'agreeing', 'agreement*', 'agrees']

Loaded 620 positive words, 744 negative words

```
In [54]: def analyze_text(text: str, posemo: list, negemo: list):
    import re
    words = re.findall(r'\b\w+\b', text.lower())
    total = len(words)
    pos_count = sum(1 for w in words if w in posemo)
    neg_count = sum(1 for w in words if w in negemo)
    pos_pct = (pos_count / total * 100) if total else 0
    neg_pct = (neg_count / total * 100) if total else 0
    print(f"Total words: {total}")
    print(f"Positive words: {pos_count} ({pos_pct:.2f}%)")
    print(f"Negative words: {neg_count} ({neg_pct:.2f}%)")

    # 範例輸入
    text = "I love you. I hate u"
    analyze_text(text, posemo, negemo)
```

Total words: 6

Positive words: 1 (16.67%)

Negative words: 1 (16.67%)

```
In [66]: # 擴充功能
iwords = ["i", "me", "my"]

# 以下維度字典檔案目前未提供，如需使用請自行準備檔案並取消註解
# social = load_dict("dict_social.txt")
# cogproc = load_dict("dict_cogproc.txt")
# allure = load_dict("dict_allure.txt")
# moral = load_dict("dict_moral.txt")

def clean_text(text: str) -> str:
    """
```

```

清理文字，處理特殊字符（如彎引號、連字號等）
"""

text = text.replace("'", "").replace("-", " ").replace("_", " ").rep
return text

def analyze_text_adv(text: str, posemo, negemo, iwords):
    import re
    text = clean_text(text) # 清理文字
    words = re.findall(r'\b[a-zA-Z]+\b', text.lower()) # 排除數字、URL、em
    total = len(words)
    def pct(count): return (count / total * 100) if total else 0

    def match_dict(word, dictionary):
        """
        支援字尾萬用字 (*) 匹配
        """
        return any(word.startswith(entry.rstrip('*')) for entry in dictio

stats = {
    "I-words (I, me, my)": sum(1 for w in words if w in iwords),
    "Positive Tone": sum(1 for w in words if match_dict(w, posemo)),
    "Negative Tone": sum(1 for w in words if match_dict(w, negemo)),
    # "Social Words": sum(1 for w in words if match_dict(w, social)),
    # "Cognitive Processes": sum(1 for w in words if match_dict(w, co
    # "Allure": sum(1 for w in words if match_dict(w, allure)),
    # "Moralization": sum(1 for w in words if match_dict(w, moral)),
}

print(f"{'Traditional LIWC Dimension':<25}{'Your Text (%)':<15}")
for k, v in stats.items():
    print(f"{k:<25}{pct(v):<15.2f}")

# 範例輸入
text = "I love you. I hate u"
analyze_text_adv(text, posemo, negemo, iwords)

```

Traditional LIWC Dimension	Your Text (%)
I-words (I, me, my)	33.33
Positive Tone	33.33
Negative Tone	16.67

RESULTS

Traditional LIWC Dimension	Your Text	Average for Other Language
I-words (I, me, my)	33.33	4.27
Positive Tone	16.67	3.50
Negative Tone	16.67	1.54
Social Words	33.33	8.16
Cognitive Processes	0.00	10.44
Allure	16.67	6.95
Moralization	0.00	0.26
Summary Variables		
Analytic	0.01	49.63
Authentic	99.71	49.95

Eaxmple

```
In [68]: from pathlib import Path

def analyze_file(filepath_str, posemo, negemo, iwords):
    """
    讀取指定檔案並分析文字內容
    :param filepath_str: 檔案路徑 (字串)
    :param posemo: 正向情緒字典
    :param negemo: 負向情緒字典
    :param iwords: I-words 字典
    """
    filepath = Path(filepath_str)
    if filepath.exists():
        with filepath.open("r", encoding="utf-8") as f:
            text = f.read()
            analyze_text_adv(text, posemo, negemo, iwords)
    else:
        print(f"檔案不存在: {filepath}")

# 使用範例
```

```
In [69]: analyze_file('test/test1.txt', posemo, negemo, iwords)
#跟網站的結果有差異，應該是因為2015年的字典 liwc 應該有更新字典
```

```
Traditional LIWC DimensionYour Text (%)
I-words (I, me, my)          0.00
Positive Tone                13.53
Negative Tone                 5.29
```

RESULTS

Traditional LIWC Dimension	Your Text	Average for Other Language
I-words (I, me, my)	0.00	4.27
Positive Tone	10.65	3.50
Negative Tone	4.14	1.54
Social Words	7.69	8.16
Cognitive Processes	13.02	10.44
Allure	1.78	6.95
Moralization	0.00	0.26
Summary Variables		
Analytic	94.68	49.63
Authentic	38.91	49.95

In [62]: `analyze_file('test/test2.txt', posemo, negemo, iwords)`

```
Traditional LIWC DimensionYour Text (%)
I-words (I, me, my)      0.00
Positive Tone            25.38
Negative Tone            5.58
```

RESULTS

Traditional LIWC Dimension	Your Text	Average for Other Language
I-words (I, me, my)	0.00	4.27
Positive Tone	21.83	3.50
Negative Tone	5.58	1.54
Social Words	6.09	8.16
Cognitive Processes	11.68	10.44
Allure	2.54	6.95
Moralization	0.51	0.26
Summary Variables		
Analytic	77.49	49.63
Authentic	16.22	49.95

誤差 (%) = |你的結果 - LIWC 網站結果| / LIWC 網站結果 * 100

1. Test 1:

- Positive Tone 誤差 (%) = $|13.53 - 10.65| / 10.65 * 100 = 2.88 / 10.65 \times 100 \approx 27.04\%$
- Negative Tone 誤差 (%) = $|5.29 - 4.14| / 4.14 * 100 = 1.15 / 4.14 * 100 \approx 27.78\%$

2. Test 2:

- Positive Tone 誤差 (%) = $|25.38 - 21.83| / 21.83 * 100 = 3.55 / 21.83 * 100 \approx 16.26\%$
- Negative Tone 誤差 (%) = $|5.58 - 5.58| / 5.58 * 100 = 0.00 / 5.58 * 100 = 0.00\%$

為什麼分析結果與 LIWC 官網不完全一致？

本專案的程式碼已盡量模擬 LIWC 的分析流程，但仍可能出現誤差，主要原因如下：

1. 字典版本差異

- 本專案使用的是 LIWC 2015 開放字典，官網可能使用更新版（如 LIWC 2022），詞彙內容有所不同，導致部分情緒詞無法命中。

2. 分詞規則細節

- 雖然已將彎引號、連字號、標點符號正規化，但 LIWC 官網可能有更進階的分詞演算法（如處理複合詞、特殊縮寫、Unicode 支援等），本程式僅用正則表達式，細節上仍有差異。

3. 萬用字匹配方式

- 本程式僅支援字尾萬用字 (*) 且用字首匹配，若字典設計或官網演算法有不同，可能造成部分詞彙未被正確辨識。

4. 比例分母定義

- 本程式分母為「可見英文詞 token 數」，若官網分母包含或排除其他類型詞（如數字、符號、複合詞），計算結果會有誤差。

5. 前處理流程差異

- 官網可能有額外的前處理步驟（如去除停用詞、特殊詞處理、語境判斷等），本程式僅做基本清理。