计算机 22A4 2022015232 孙春辉
2025 年 4 月 10 日 星期四 第九次作业

Report 14

Part 1: Course exercises

1.1: 词形还原

1.2: 词性标注

1.3: 停用词

1.4: NLTK

1.5: 精确

2.1: F

2.2: T

2.3: F

2.4: F

2.5: F

3.1: D

3.2: ABCD

3.3: D

3.4 A

3.5: B

4.1: 文本预处理的基本流程包括分词、词形统一化、删除停用词。

4.2:

1、 对文本进行分词操作，从中找出正向情感词、负向情感词、否定词以及程度副词。

2、判断每个情感词之前是否有否定词及程度副词，将它之前的否定词和程度副词划分为一组。如果存在否定词，则将情感词的情感权值乘以-1；如果有程度副词，就乘以程度副词的程度值。

3、将所有组的得分加起来，得分大于 0 的归于正向，小于 0 的归于负向。

Part 2:

Re-implement the codes from the product evaluation analysis Demo in the text book page 249-253 summarize the results and screenshot the codes in the report.

```
import jieba
import pandas as pd
from nltk import FreqDist
from matplotlib import pyplot as plt
from wordcloud import WordCloud

file_path = open('./商品评价信息.csv')
file_data = pd.read_csv(file_path)
# print(file_data)
file_data.drop(file_data[file_data['评价信息'] == '此用户没有填写评价。
'].index,inplace=True)
# print(file_data)

cut_words = jieba.lcut(str(file_data['评价信息'].values),cut_all =
False)
print(cut_words)

stop_path = open('./停用词表.txt',encoding='utf-8')
```

计算机 22A4 2022015232 孙春辉
2025 年 4 月 10 日 星期四 第九次作业

```
stop_words =stop_path.read()

new_data = []
for word in cut_words :
    if word not in stop_words :
        new_data.append(word)

print(new_data)

freq_list = FreqDist(new_data)

most_common_words = freq_list.most_common()
print(most_common_words)

font = r'C:\Windows\Fonts\STXINGKA.TTF'
wc = WordCloud(font_path = font, background_color = 'white',width =
1000,height = 800).generate(" ".join(new_data))
plt.imshow(wc)
plt.axis('off')
plt.show()

Prefix dict has been built successfully.
['['  "'", '穿', '上', '挺舒服', '的', ',', '是', '我', '有史以来', '在', '网上'
, '有史以来', '网上', '买', '漂亮', '最舒服', '衣服', '面料', '挺不错', '
[('衣服', 3), ('买', 2), ('喜欢', 2), ('裤子', 2), ('男朋友', 2), ('比较', 2),

Process finished with exit code 0
```

计算机 22A4 2022015232 孙春辉
2025 年 4 月 10 日 星期四 第九次作业

Part 3:

从 Moodle 下载‘Report9-习近平在北京大学师生座谈会上的讲话.txt’文件并生成词云，具体方法参考本章节最后的实例以及网络资源，要求词云中不同频率的重点词语用不同颜色和字体大小进行显示，词云的形状要求贴合讲话内容，制作出的词云要求美观、大气、生动，*如果是动态词云可加分*.

```
import jieba
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from collections import Counter
from matplotlib.animation import FuncAnimation


# 1. 文本预处理
def process_text(file_path):
    with open(file_path, 'r', encoding='utf-8') as f:
        text = f.read()

    # 清洗文本 - 移除换行符和多余空格
    text = text.replace('\n', ' ').replace('\r', ' ').strip()

    # 分词处理
    words = jieba.lcut(text)

    # 过滤单字
    words = [word for word in words if len(word) > 1]

    # 词频统计
    word_freq = Counter(words)
    return word_freq

# 2. 动态词云生成
def generate_dynamic_wordcloud(word_freq, output_path):
    # 颜色函数
    def random_color_func(word=None, font_size=None, position=None,
                          orientation=None, font_path=None,
random_state=None):
        h = random_state.randint(0, 360)
```

```
        s = random_state.randint(70, 100)
        l = random_state.randint(40, 70)
        return f"hsl({h}, {s}%, {l}%)"

    # 创建词云对象
    wc = WordCloud(
        font_path='simhei.ttf',  # 使用黑体
        background_color='white',
        width=1000,  # 设置画布宽度
        height=700,  # 设置画布高度
        max_words=200,
        max_font_size=150,
        min_font_size=10,
        color_func=random_color_func
    )

    # 生成词云
    wc.generate_from_frequencies(word_freq)

    # 创建图形
    fig, ax = plt.subplots(figsize=(12, 8))
    ax.axis('off')
    img = ax.imshow(wc, interpolation='bilinear')

    # 动画更新函数
    def update(frame):
        # 每帧改变颜色和旋转角度
        current_wc = WordCloud(
            font_path='simhei.ttf',
            background_color='white',
            width=1000,
            height=700,
            max_words=200,
            max_font_size=150,
            min_font_size=10,
            color_func=random_color_func,
            prefer_horizontal=0.9 - frame * 0.01  # 动态调整词语方向
        )
        current_wc.generate_from_frequencies(word_freq)
        img.set_array(current_wc.to_array())
        return [img]

    # 创建动画
    ani = FuncAnimation(fig, update, frames=30, interval=200,
```

```
blit=True)

    # 保存动态图
    ani.save(output_path, writer='pillow', fps=5, dpi=300)
    plt.close()


# 3. 主程序
if __name__ == "__main__":
    # 参数设置
    text_file = 'mao.txt'
    output_gif = 'mao_speech_wordcloud.gif'

    # 处理文本
    word_freq = process_text(text_file)

    # 生成动态词云
    generate_dynamic_wordcloud(word_freq, output_gif)

    print(f"动态词云已生成并保存为 {output_gif}")
```