22377316 匡昕 python week2

1-5 代码放在一块

import jieba

from collections import Counter

from wordcloud import WordCloud

import matplotlib.pyplot as plt

plt.rcParams['font.sans-serif'] = ['SimHei']

plt.rcParams['axes.unicode\_minus'] = False

def read\_file(file\_path):

"""

读取文件，一行一句为单位，返回包含每一行文本的列表

"""

lines = []

try:

with open(file\_path, 'r', encoding='utf-8') as file:

for line in file:

lines.append(line.strip())

except FileNotFoundError:

print("文件未找到！")

return lines

def load\_stopwords(stopwords\_file):

"""

加载停用词列表

"""

stopwords = set()

try:

with open(stopwords\_file, 'r', encoding='utf-8') as file:

for line in file:

stopwords.add(line.strip())

except FileNotFoundError:

print("停用词文件未找到！")

return stopwords

def tokenize\_and\_count(lines, stopwords):

"""

使用jieba对文档进行分词，并统计词频，同时过滤停用词

"""

# 分词并将结果存储在列表中

tokens = []

for line in lines:

words = jieba.cut(line) # 使用jieba进行分词

for word in words:

if word not in stopwords: # 过滤停用词

tokens.append(word)

# 统计词频

word\_counts = Counter(tokens)

return word\_counts

def generate\_wordcloud(word\_counts, title):

"""

生成词云图

"""

# 生成词云图

wordcloud = WordCloud(font\_path='simhei.ttf', width=800, height=400, background\_color='white').generate\_from\_frequencies(word\_counts)

# 显示词云图

plt.imshow(wordcloud, interpolation='bilinear')

plt.title(title)

plt.axis('off')

plt.show()

# 读取文件内容

file\_path = "D:\\python homework\\week2.txt"

lines = read\_file(file\_path)

# 加载停用词列表

stopwords\_file = "D:\\python homework\\stopword.txt"

stopwords = load\_stopwords(stopwords\_file)

# 对文档进行分词并统计词频

word\_counts = tokenize\_and\_count(lines, stopwords)

# 生成高频词的词云图

generate\_wordcloud(word\_counts, "高频词词云图")



根据结果显示，数据应该是直播间带货，主要推广的是女性的穿搭。

1. 我选择的是动词进行可视化

import jieba

import jieba.posseg as pseg

from collections import Counter

from wordcloud import WordCloud

import matplotlib.pyplot as plt

plt.rcParams['font.sans-serif'] = ['SimHei']

plt.rcParams['axes.unicode\_minus'] = False

def read\_file(file\_path):

"""

读取文件，一行一句为单位，返回包含每一行文本的列表

"""

lines = []

try:

with open(file\_path, 'r', encoding='utf-8') as file:

for line in file:

lines.append(line.strip())

except FileNotFoundError:

print("文件未找到！")

return lines

def load\_stopwords(stopwords\_file):

"""

加载停用词列表

"""

stopwords = set()

try:

with open(stopwords\_file, 'r', encoding='utf-8') as file:

for line in file:

stopwords.add(line.strip())

except FileNotFoundError:

print("停用词文件未找到！")

return stopwords

def tokenize\_and\_count(lines, stopwords):

"""

使用jieba对文档进行分词，并统计词频，同时过滤停用词

"""

# 分词并将结果存储在列表中

tokens = []

for line in lines:

words = pseg.cut(line) # 使用jieba进行分词和词性标注

for word, flag in words:

if flag.startswith('v') and word not in stopwords: # 选择动词

tokens.append(word)

# 统计词频

word\_counts = Counter(tokens)

return word\_counts

def generate\_wordcloud(word\_counts, title):

"""

生成词云图

"""

# 生成词云图

wordcloud = WordCloud(font\_path='simhei.ttf', width=800, height=400, background\_color='white').generate\_from\_frequencies(word\_counts)

# 显示词云图

plt.imshow(wordcloud, interpolation='bilinear')

plt.title(title)

plt.axis('off') # 不显示坐标轴

plt.show()

# 读取文件内容

file\_path = "D:\\python homework\\week2.txt"

lines = read\_file(file\_path)

# 加载停用词列表

stopwords\_file = "D:\\python homework\\stopword.txt"

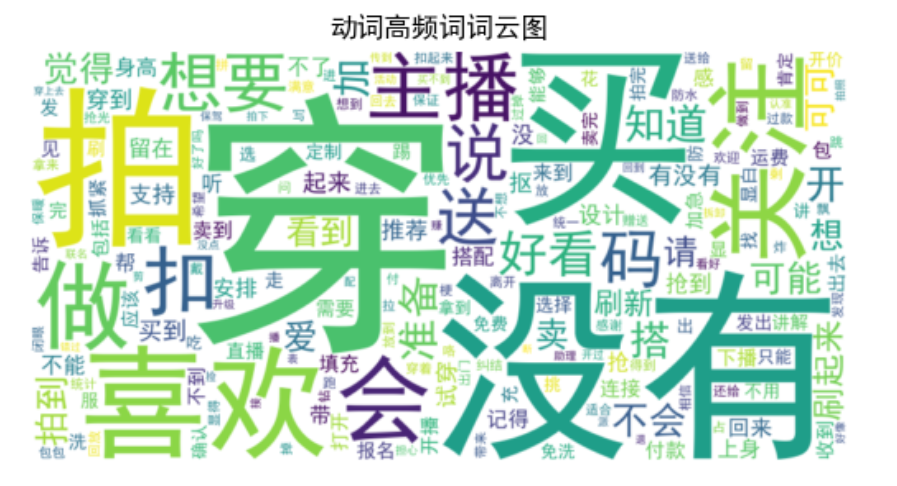
stopwords = load\_stopwords(stopwords\_file)

# 对文档进行分词并统计动词频率

verb\_word\_counts = tokenize\_and\_count(lines, stopwords)

# 生成动词高频词的词云图

generate\_wordcloud(verb\_word\_counts, "动词高频词词云图")



1. 尝试给chatgpt，但是无法接收太大的数据

8.

import jieba

from collections import Counter

from wordcloud import WordCloud

import matplotlib.pyplot as plt

def read\_file(file\_path):

"""

读取文件，一行一句为单位，返回包含每一行文本的列表

"""

lines = []

try:

with open(file\_path, 'r', encoding='utf-8') as file:

for line in file:

lines.append(line.strip())

except FileNotFoundError:

print("文件未找到！")

return lines

def tokenize\_and\_count(lines):

"""

使用jieba对文档进行分词，并统计bigram词频

"""

# 分词并将结果存储在列表中

tokens = []

for line in lines:

words = jieba.lcut(line) # 使用jieba进行分词

for i in range(len(words)-1):

tokens.append((words[i], words[i+1]))

# 统计bigram词频

bigram\_counts = Counter(tokens)

# 对词频进行排序

sorted\_bigram\_counts = sorted(bigram\_counts.items(), key=lambda x: x[1], reverse=True)

# 输出前20个高频bigram

print("前20个高频bigram：")

for bigram, count in sorted\_bigram\_counts[:20]:

print(f"{bigram}: {count}")

return sorted\_bigram\_counts

def generate\_wordcloud(word\_counts, title):

"""

生成词云图

"""

# 提取高频的bigram和对应的频率

bigrams = [item[0] for item in word\_counts]

frequencies = [item[1] for item in word\_counts]

# 将bigrams转换为字符串

bigram\_strings = [' '.join(bigram) for bigram in bigrams]

# 将bigrams和frequencies组合成字典

bigram\_freq\_dict = dict(zip(bigram\_strings, frequencies))

# 生成词云图

wordcloud = WordCloud(font\_path='simhei.ttf', width=800, height=400, background\_color='white').generate\_from\_frequencies(bigram\_freq\_dict)

# 显示词云图

plt.imshow(wordcloud, interpolation='bilinear')

plt.title(title)

plt.axis('off') # 不显示坐标轴

plt.show()

# 读取文件内容

file\_path = "D:\\python homework\\week2.txt"

lines = read\_file(file\_path)

# 对文档进行分词并统计bigram词频

sorted\_bigram\_counts = tokenize\_and\_count(lines)

# 生成高频bigram的词云图

generate\_wordcloud(sorted\_bigram\_counts, "高频Bigram词云图")



9.

可以利用词频来进行特征词的筛选；

词袋模型：将文本表示为一个向量，向量的每个维度对应一个特征词，向量的值表示对应特征词在文本中出现的次数或者权重。

可以使用余弦相似度等方法来计算不同句子之间的相似性。余弦相似度是通过计算两个向量之间的夹角余弦值来衡量它们的相似程度的方法，值越接近1表示相似度越高，越接近-1表示相似度越低。