

文旅舆情监测词云模型

一、模型概述

本模型聚焦于对旅游行业相关舆情的监测分析，特别是围绕 X 市旅游情况，通过抓取各大网站上搜索、点击等行为所涉及的信息，提炼出相关关键词，并依据其被搜索、使用的频次来呈现词云效果，旨在为 X 市旅游管理部门、旅游企业以及相关研究人员等直观展示当下公众对 X 市旅游关注的热点内容、焦点话题，助力及时掌握舆情动态，以便制定针对性的应对策略和宣传推广方案。

二、数据来源与整理

搜索引擎数据：借助主流搜索引擎（如百度、谷歌等）提供的开放数据接口或者相关搜索指数工具，获取包含“X 市旅游”关键词的搜索记录，这些记录涵盖了不同用户在不同时间发起的搜索行为，能反映大众对 X 市旅游各方面的关注度，是挖掘热门关键词的重要数据源之一。

社交媒体平台数据：诸如微博、微信公众号、抖音、小红书等社交媒体平台上，用户发布的与 X 市旅游相关的话题、动态、评论以及点赞等互动信息，包含了大量关于 X 市旅游景点、美食、住宿、游玩体验等方面的关键词内容，通过网络爬虫技术（在遵循平台规则和合法合规的前提下）收集这些文本数据，能够从社交传播角度丰富关键词来源，了解大众的讨论热点。

旅游专业网站与论坛数据：像马蜂窝、携程旅行网、去哪儿网等专业旅游网站以及各类旅游论坛中，游客分享的游记、攻略、问答等板块包含众多涉及 X 市旅游的详细描述和关注点，从中提取相关关键词，可进一步补充完善反映 X 市旅游特色、游客需求等方面的词汇信息，使数据更具专业性和针对性。

数据整理流程：首先，将从上述多渠道收集到的海量文本数据汇总到统一的数据仓库中。接着，运用自然语言处理技术中的文本清洗工具，去除文本中的标点符号、停用词（如“的”“是”“在”等对语义分析无实质帮助的常用词）以及重复、无关的信息内容，对文本进行规范化处理。然后，通过词法分析、词性标

注等手段，将文本分解为一个个有意义的词语，并依据旅游行业相关的主题词库（事先构建好，包含景点名称、旅游活动、服务类型等各类旅游相关词汇）进行筛选，提取出真正与 X 市旅游相关的有效关键词，为后续的频次统计和词云生成做好准备。

三、核心算法与分析逻辑

关键词频次统计：运用数据分析软件（如 Python 的数据分析库 pandas 结合 collections 库中的 Counter 类等），对整理好的 X 市旅游相关关键词列表进行遍历统计，计算每个关键词在所有文本数据中出现的次数，形成“关键词 - 频次”的对应关系数据，以此量化各关键词受关注的程度，频次越高意味着该关键词所代表的内容越受大众关注，在舆情中越为热点话题。

权重计算与排序（可选，用于优化词云展示效果）：为了使词云更合理地展示关键词的重要性差异，可进一步计算关键词的权重。例如，可以采用 TF-IDF（词频 - 逆文档频率）算法，综合考虑关键词在单篇文本中的词频以及在整个语料库（所有收集的文本数据）中的普遍程度，赋予每个关键词相应的权重值，然后按照权重值大小对关键词进行排序，权重越高的关键词在后续词云展示中会越突出显眼。

四、模型输出与可视化

词云图展示：利用专业的可视化工具（如 Python 的 wordcloud 库或者在线词云生成平台等），根据关键词的频次（或权重）情况，生成以 X 市旅游为主题的词云图。在词云图中，将各个关键词以不同的字体大小、颜色等形式呈现，关键词被搜索、使用的次数越多（或权重越高），其在词云图中显示的字体越大、颜色越突出（可根据设计自行设定突出显示的方式），整体形成直观且形象的词云效果，让人一眼就能看出大众关注的重点话题和热门内容。

交互功能实现：借助前端可视化开发技术（如 JavaScript 结合 Echarts 等可视化库），为词云图添加交互功能。当鼠标指针悬停在某个关键词上时，在词云

图上方或旁边弹出信息框，清晰展示该关键词的具体内容以及被搜索、使用的次数（或权重数值等详细信息），方便用户进一步了解详情，深入洞察各关键词背后所反映的舆情情况。

五、模拟数据示例与可视化展示

假设通过收集和整理数据后，得到以下部分关于 X 市旅游的关键词及频次模拟数据（仅为示例示意）：

关键词	频次
X 市古城	500
X 市美食街	400
X 市景区门票	350
X 市特色民宿	300
X 市旅游攻略	250

对应的可视化展示如下：



（此处应插入一个简单的词云图示例，图中“X 市古城”字体最大、颜色最突出，“X 市美食街”次之，以此类推，各关键词按照频次情况在词云图中呈现不同大小和颜色，鼠标悬停在某个关键词上有信息框展示关键词及频次示例图）

X 市古城 X 市美食街 X 市景区门票 X 市特色民宿 X 市旅游攻略

以下是使用 Python 的 pandas、wordcloud 和 matplotlib 库来生成满足要求的

词云图的代码示例：

一、代码实现

隐藏过程

```
python
import pandas as pd
from wordcloud import WordCloud, ImageColorGenerator
import matplotlib.pyplot as plt
from colormap import rgb2hex

# 读取 Excel 文件
df = pd.read_excel('/mnt/26.xlsx')
# 假设数据中有一个列用于关键词和频次信息，例如'关键词'列和'频次'列
keywords = df['关键词'].tolist()
frequencies = df['频次'].tolist()
# 创建一个字典，将关键词和频次对应起来
keyword_dict = dict(zip(keywords, frequencies))
# 自定义颜色函数，根据频次生成不同颜色
def color_func(word, font_size, position, orientation, random_state=None, **kwargs):
    max_freq = max(keyword_dict.values())
    min_freq = min(keyword_dict.values())
    freq = keyword_dict[word]
    # 将频次映射到 0 - 1 之间
    normalized_freq = (freq - min_freq) / (max_freq - min_freq)
    # 这里简单地根据频次设置颜色从浅到深（例如蓝色调）
    r = int(0 + normalized_freq * 200)
    g = int(0 + normalized_freq * 200)
    b = int(255 - normalized_freq * 200)
    return rgb2hex(r, g, b)
# 生成词云
wordcloud = WordCloud(background_color='white').generate_from_frequencies(keyword_dict)
# 重新着色词云
wordcloud.recolor(color_func=color_func)
# 找出最大频次的关键词并设置其字体大小更大（例如两倍）
max_freq_keyword = max(keyword_dict, key=keyword_dict.get)
wordcloud_layout = wordcloud.layout_for(word, font_size, position, orientation, _)
for word, font_size, position, orientation, _ in wordcloud_layout:
    if word == max_freq_keyword:
        new_font_size = font_size * 2
        wordcloud_layout[(word, new_font_size, position, orientation, _)] = (word, new_font_size, position, orientation, _)
# 显示词云图
plt.figure(figsize=[10, 6])
plt.imshow(wordcloud, interpolation='bilinear')
```

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

二、代码解释

数据读取:

`pd.read_excel('/mnt/26.xlsx')` 读取上传的 Excel 文件, 假设文件中包含关键词和频次两列, 分别存储关键词和其对应的频次信息。

数据整理:

将关键词和频次信息整理成一个字典 `keyword_dict`, 以便 `wordcloud` 库使用。

颜色函数定义:

`color_func` 函数根据关键词的频次来生成不同的颜色。首先找到频次的最大值和最小值, 然后将每个关键词的频次映射到 0 – 1 之间, 根据这个归一化的频次来生成从浅到深的颜色 (这里是简单的蓝色调变化)。

词云生成:

`WordCloud(background_color='white').generate_from_frequencies(keyword_dict)` 创建一个 `WordCloud` 对象, 并根据关键词和频次生成词云, 背景颜色设置为白色。

词云重新着色:

`wordcloud.recolor(color_func=color_func)` 根据定义的颜色函数对词云进行重新着色。

设置最大频次关键词的字体大小:

找出频次最大的关键词 `max_freq_keyword`, 然后遍历词云的布局 `wordcloud.layout_`, 将最大频次关键词的字体大小设置为原来的两倍。

显示词云图:

使用 `matplotlib` 的 `imshow` 方法显示词云图, 设置图片大小并关闭坐标轴。

三、注意事项

数据列名:

确保代码中的关键词和频次列名与上传的 Excel 文件中的实际列名一致, 否则将无法正确读取数据生成词云。

颜色映射:

可以根据需求自定义颜色函数 `color_func` 中的颜色映射方案, 以更好地展示数据的差异。

字体大小调整:

对最大频次关键词字体大小的调整方式可以根据实际需求进行修改, 例如可以设置为其他倍数或者采用更复杂的字体大小调整逻辑。

六、应用场景与价值

舆情监测与危机管理: X 市旅游管理部门通过实时监测词云模型展示的舆情关键词及频次变化, 能够及时察觉公众对 X 市旅游存在的负面评价、热点争议等情况, 例如, 若某个与旅游服务质量相关的关键词频繁出现且负面情绪明显, 可迅

速介入调查，采取针对性措施加以解决，避免负面舆情的进一步扩散，维护 X 市旅游的良好形象。

宣传推广策略调整：旅游企业可以依据词云模型反映出的大众关注焦点，调整宣传推广策略。比如，发现游客对 X 市某特色景点关注度极高，可加大对该景点周边旅游产品（如配套住宿、餐饮、游玩项目等）的宣传力度，或者与景点合作推出联合营销活动；同时，针对游客搜索较多的旅游攻略类关键词，制作并分享更贴合需求的优质攻略内容，提高企业在市场中的曝光度和吸引力。

产品优化与服务提升：从词云图中了解到游客关注的高频关键词所涉及的内容，有助于 X 市旅游行业相关主体对自身产品和服务进行优化。例如，若“X 市特色民宿”搜索频次高，民宿经营者可据此研究游客需求，改进民宿设施、提升服务质量，打造更具特色的民宿产品；景区管理方看到与景区门票相关的高热度话题，可考虑优化门票价格体系、完善购票渠道等，以更好地满足游客需求，提升游客满意度。