文章编号：2096-1472(2023)-06-55-04
DOI:10.19644/j.cnki.issn2096-1472.2023.006.012

基于大数据的微博舆情分析系统的设计与实现

霍 英[1]，丘志敏[2]，李小帆[1]，李彦廷[1]

(1.韶关学院信息工程学院，广东 韶关 512005；
2.韶关学院智能工程学院，广东 韶关 512005)
✉ huoying@sgu.edu.cn; 250437325@qq.com; 14929099@qq.com; kidi@qq.com

摘 要：为了对社会突发事件引发的舆情及时开展分析、引导及治理，在大数据环境下，采用SSM(Spring+SpringMVC+MyBatis)架构设计并实现了一个集微博舆情信息采集、去重、分析、处理及可视化的综合平台，同时对微博舆情分析的数据处理进行了介绍，并对文本情感分析、计算文本相似度给出了具体算法。系统通过数据采集层、数据处理层和数据展示层3个子层实现对微博舆情信息的趋势性预警及有效监管。目前，该平台完成了相关微博舆情热点话题的追踪及趋势预警，在实际应用中，观察到在单机日采集量约100万条时，对舆情数据分析的有效率可以达到90%以上。

关键词：网络舆情；大数据；微博
中图分类号：TP319 **文献标识码**：A

Design and Implementation of Weibo Public Opinion Analysis System Based on Big Data

HUO Ying[1], QIU Zhimin[2], LI Xiaofan[1], LI Yanting[1]

(1.*School of Information Engineering, Shaoguan University, Shaoguan* 512005, *China*;
2.*School of Intelligent Engineering, Shaoguan University, Shaoguan* 512005, *China*)
✉ huoying@sgu.edu.cn; 250437325@qq.com; 14929099@qq.com; kidi@qq.com

Abstract: In order to timely analyze, guide, and manage public opinion triggered by social emergencies, this paper proposes to adopt SSM (Spring +SpringMVC +MyBatis) architecture to design and implement a comprehensive platform for collecting, de-repeating, analyzing, processing and visualization of Weibo public opinion information under the big data environment. At the same time, the data processing of public opinion analysis is introduced, and the specific algorithms of text sentiment analysis and text similarity calculation are also introduced. The system realizes trend warning and effective supervision of Weibo public opinion information through three sub-layers: data acquisition layer, data processing layer and data display layer. At present, the tracking and trend warning of relevant hot topics of public opinion on Weibo have been completed on the platform. In practical application, it is observed that the efficiency of public opinion data analysis can reach more than 90% when the daily collection volume of a single machine is about 1 million.

Keywords: Internet public opinion; big data; Weibo

1 引言(Introduction)

网络舆情分析包括对网络信息进行采集、去重、抽取、分词、索引、存储和分析等一系列过程，是进行网络舆论感知、监控与引导的重要依据。如何规范大数据环境下网络舆情信息的及时感知、收集、分析与监管，并及时有效地引导舆情向良好的方向发展，如何实现“传播力决定影响力，话

基金项目：广东省自然科学基金项目(2021A1515011803)；广东省哲学社会科学规划学科共建项目(GD18XXW07).

语权决定主导权，时效性决定有效性，透明度决定公信度”的使命，是我国各级政府相关部门面临的现实问题，也是亟待解决的重要课题[1-2]。基于此，在大数据环境下，本文给出了一种微博舆情分析系统的设计与实现方案，并对文本情感分析、计算文本相似度给出了具体实现算法，以实现对舆情信息的趋势性预警及有效监管。

2 系统功能及流程(System functions and processes)

2.1 系统功能

本系统主要是通过采集微博用户指定话题的相关数据，并对数据进行初步加工后，通过业务系统展示出来，能清晰直观地体现某热点话题的传播速度，以及公众对该舆情事件的态度。

本系统主要分成三个子层：数据采集层、数据处理层及数据展示层(业务子系统展示层)。其中，数据采集层主要通过官方提供的数据应用获取接口(API)及使用自定义的网络爬虫[3-5]获取网络数据；数据处理层主要完成中文的分词、文本的清洗、标准化、特征提取、情感分析等工作；业务子系统展示层主要是面向平台用户，将数据处理层处理后的数据以图表、文字的形式呈现给用户，并完成与用户的交互等功能。三个子层都可以当成一个独立的系统使用，拥有良好的扩展性。系统整体运行流程如图1所示，系统主要功能如图2所示。

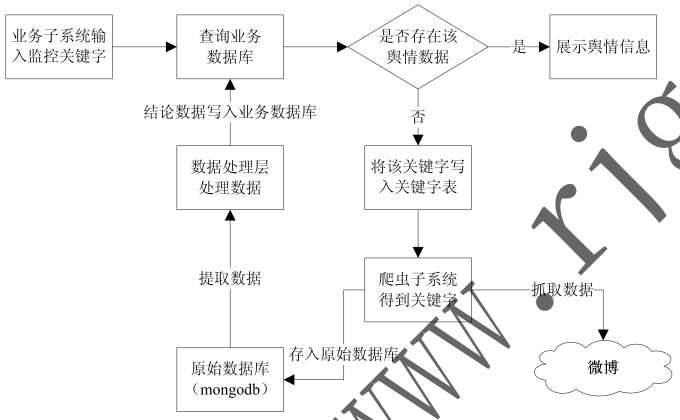

图1 系统运行图

Fig. 1 System run diagram

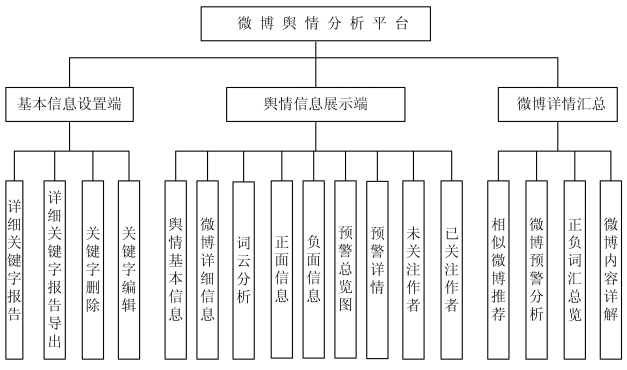

图2 微博舆情分析平台功能图

Fig. 2 Functional map of microblog public opinion analysis platform

2.2 系统流程

系统采用SSM(Spring+SpringMVC+MyBatis)架构，共划分为四个层次：View(表现)层、Controller(控制)层、Service(业务逻辑)层、DAO(数据持久)层，如图3所示，其流程主要分为以下几个步骤。

(1)客户端(用户)发送请求到分发器。

(2)分发器查询 Hander Mapping(处理器映射)，并找到处理请求的控制器。

(3)控制器调用业务逻辑层后，处理返回 Model and View(模型和视图对象)。

(4)分发器负责查询视图解析器，并找到 Model and View 指定的视图。

(5)视图把结果展示到客户端。

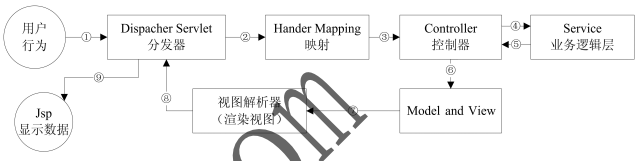

图3 系统处理流程

Fig. 3 System processing flow

3 数据处理技术(Data processing technology)

微博舆情分析中的数据处理主要采用以下技术：对获取的原始文本进行中文分词、文本清洗、情感分析等。

3.1 中文分词

由于语言的差异，因此中英文在分词的思路上有较大区别。多数情况下，英文通常使用空格就可以自动完成分词任务，但中文的语法复杂，通常会通过第三方库进行分词操作。本系统采用第三方库 Jieba 分词器进行中文分词工作[6]。Jieba 分词器工作的主要原理是基于统计词典，首先建立一个前缀词典，然后利用前缀词典对句子展开切分，并根据所有切分的可能及切分位置构造一个有向无环图，通过相关动态规划算法计算出最大概率路径，从而获得最终切分形式。

3.2 文本清洗

经过分词之后获取的文本中通常含有类似标点符号、停用词等无用信息，因此需要对文本开展分步清洗工作。本系统使用正则表达式结合百度的停用词库进行文本清洗。

正则表达式的基本原理如下[7]：定义文法 $G=\{Vn,Vt,S,P\}$，其中 Vn 是一个非空有限的符号集合，它的元素称为非终结符号；Vt 也是一个非空有限符号集合，它的元素称为终结符号，并且 $Vt\cap Vn=\varnothing$；S 是文法 G 的起始符号，并且 $S\in Vn$；P 是一个非空有限集合，它的元素称为产生式。所谓产生式，其形式为 $\alpha\rightarrow\beta$，它的元素为形如 $\alpha\rightarrow\beta$ 的产生式，其中 α、β 是由终结符和非终结符组成的符号串，并且 α、$\beta\in(Vt\cup Vn)^*$，$\alpha\neq\varepsilon$。开始符 S 必须至少在某一产生式的 α 中出现一次。正则表达式是一种3型文法，它以整个字母表作为 Vt 集合。假设一个文法的产生式为$\{S\rightarrow Sa;\ S\rightarrow b;\}$，那么对应的正则表达式为 $\mathbf{ba}^*$。例如，<h1>hello，world</h1>等包含 html 标签的文本，可使用“<\/?.+?\/?>”表达式去除 html 标签。

去除停用词的原理如下：对文本进行中文分词操作后，将产生一个包含所有词语的列表，通过逐个取出列表中词语与停用词库里的词语进行匹配，若相同则在列表中删除该词，从而实现去除停用词。

3.3 文本情感分析及评分生成规则

文本情感分析主要是指对文本开展检测、分析以及挖掘的过程，文本中可能包含有用户的观点、喜好、情感等主观因素，通过情感分析可以归纳、推理文本和文本中各个人物所要表达的情感色彩。SnowNLP[8]是 python 编写的一个类库，它提供了情感倾向分析等功能，可以方便地处理中文文本内容，本系统主要借助它实现文本情感分析和文本评分功能。通过人工预先把部份微博数据分为消极的和积极的两种类别，以此为基础，通过贝叶斯公式推断接下来提取的微博数据属于积极类别的概率。当某条微博数据有 60%以上的概率属于积极类，则认为该条微博属于正面微博，并将该微博属于积极类别的概率值减去 0.6 后的取值，作为其文本评分。

3.4 敏感词及敏感分数生成规则

通过用户提供的敏感词典(*.txt 文件)，从每条微博中提取出敏感词汇，存储在业务数据库中。通过自定义的词库种类，还可以提取出正负词汇和关键词等。敏感分数则由该微博中出现敏感词的个数决定，出现一个敏感词加一分，并存储到业务数据库中。

4 核心算法(Core algorithm)

4.1 SnowNLP 文本情感分析

SnowNLP 是 python 编写的一个类库，它提供了情感倾向分析等功能，但因其使用的模型是通过影评训练生成的，并不完全契合本系统的需求，因此需要重新训练生成新的模型，并替换 SnowNLP 中原有模型。

贝叶斯模型[9]是用于情感分类的基本模型。对于两个类别的分类问题，其特征表示为 $w_1,w_2,\cdots,w_n$，并且各个特征之间相互独立，对于其中一个类别的贝叶斯模型可以表示如下：

$$P(c_1|w_1,\cdots,w_n)=\frac{P(w_1,\cdots,w_n|c_1)P(c_1)}{P(w_1,\cdots,w_n)}$$

例如，有一句话 text1："我想出去玩。"

text1 的分词结果为["我","想","出去","玩"]，A_1，A_2，A_3，A_4 分别表示 4 个词，C_1 表示积极类，C_2 表示消极类。

计算每个单词出现的频率：则 $P(A_1)=A_1$ 的词频/训练集的单词总数，同理 $P(A_2)=A_2$ 的词频/训练集的单词总数。

计算 C_1 情绪中单词出现的概率：$P(A_1|C_1)=C_1$ 中 A_1 的词频/训练集中属于 C_1 的单词总数。$P(A_2|C_1)=C_1$ 中 A_2 的词频/训练集中属于 C_1 的单词总数。

计算训练集中 C_1 类别出现的概率：$P(C_1)=C_1$ 的句子总数/训练集的句子总数。

则 text1 属于 C_1 类的概率：

$$P(C_1|A_1,\cdots,A_4)=\frac{P(C_1)P(A_1,A_2,A_3,A_4|C_1)}{P(A_1,A_2,A_3,A_4)}$$

$$=\frac{P(C_1)\prod_{i=1}^{4}P(A_i|C_1)}{P(A_1,A_2,A_3,A_4)}$$

为了得到新的模型，需要重新对贝叶斯模型进行训练，生成新的模型并替换原有模型，而训练的实质就是计算每个特征出现的频次，当统计好训练样本中的 total 和每一个特征 key 的 d[key]后，训练过程就构建完成。这个过程的核心代码如下：

```
def train(self, data):
        # data 中既包含正样本,也包含负样本
    For d in data: # data 中是 list
        # d[1]:正/负样本的标记
            C=d[1]
        If c not in self.d:
            self.d[c]=AddOneProb() # 类的初始化
        For word in d[0]: # 分词结果中的每一个词
            self.d[c].add(word, 1)
    self.total=sum(map(lambda x: self.d[x].getsum(), self.d.keys())) # 取得所有 d 中的和 sum
Class AddOneProb(BaseProb):
    Def __init__(self):
        self.d={}
        self.total=0.0
        self.none=1 # 默认所有的 none 为 1
    # 这里如果 value 也等于 1,则当 key 不存在时,累加的是 2
    Def add(self, key, value):
        self.total +=value
        # 不存在该 key 时,需新建 key
        If not self.exists(key):
          self.d[key]=1
          self.total +=1
        self.d[key] +=value
Def classify(self, x):
    tmp={}
    For k in self.d: # 正类和负类
        tmp[k]=log(self.d[k].getsum())-log(self.total) # 正/负类的所有之和的 log 函数
        For word in x:
            tmp[k] +=log(self.d[k].freq(word)) # 词频,不存在就为 0
    ret, prob=0, 0
    For k in self.d:
        Now=0
        try:
            For otherk in self.d:
                Now +=exp(tmp[otherk]-tmp[k])
                Now=1/now
        Except OverflowError:
            Now  0
        If now > prob:
            ret, prob=k, now
    Return (ret, prob)
```

4.2 计算文本相似度算法

该算法主要集成在系统的推荐功能上，以初始文本作为质心，计算与其距离最近的微博，即文本相似度最高的微博

推送即可。由于计算机不能直接识别中文，因此需要先把文本转化为向量的形式，其主要步骤可分为去除停用词、计算 *TF-IDF*（*TF* 和 *IDF* 的乘积）的值，以此作为该文本向量的表现形式。其中，*TF* 定义为某个词或短语在一个文本中出现的频率，当该词频率很高时，可认为其是该文本较为明显的特征，有较好的区分效果；其计算公式可表示为 *TF*＝某词出现的次数/该文档中所有的词的总数。

IDF 主要用于评价一个词语的普遍性和重要性，它是以某文件数目除以包含该词语的文件的数目，再以 10 为底，取对数计算得到。当包含该词的文件数越少，便可认为该词在文档中的重要性越高，即 *IDF* 的值越大[10]；其计算公式可表示如下：

$$IDF_i=\lg\frac{|D|}{1+|\{j:t_i\in d_j\}|}$$

其中，*D* 为总文档数，分母包含该词语的文档总数。

本文所提算法的具体实现方案如下。

(1)对每个文档进行分词，并去除停用词。详细流程如图 4 所示：

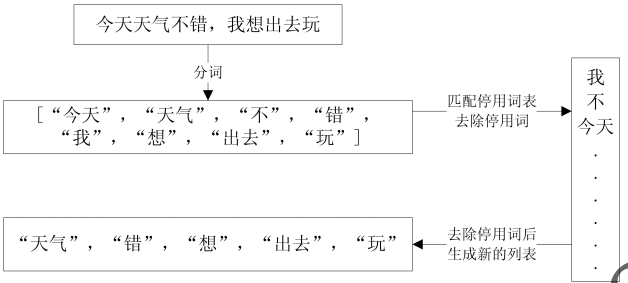

图 4 去除停用词流程

Fig. 4 Process of removing stop words

(2)计算每个文档的 *TF* 即词频，假设 *IDF* 值均为 2。通过图 4 可得出文档的词频为{“天气”＝0.2，“错”＝0.2，“想”＝0.2，“出去”＝0.2，“玩”＝0.2}。因此，*TF-IDF* 的可表示为{“天气”＝0.4，“错”＝0.4，“想”＝0.4，“出去”＝0.4，“玩”＝0.4}。

(3)以 *TF-IDF* 作为该文本向量的表现形式，即[0.4，0.4，0.4，0.4，0.4]为该文本向量化后的结果。

(4)重复上述步骤，计算从数据库中随机查找出来的微博内容的向量值，并以字典的形式保存，如{微博 ID＝[0.4，0.4，0.4，0.4，0.4]}。最终将形成一个 *i* 行 *j* 列的矩阵：

$$\begin{bmatrix}X_{11} & \cdots & X_{1j}\\ \vdots & & \vdots\\ X_{i1} & \cdots & X_{ij}\end{bmatrix}$$

(5)通过欧式距离计算每条微博到用户当前查看的微博的距离，选取距离最近的前 *N* 条展示出来。欧式距离计算公式如下：

$$d_{ik}=\sqrt{\sum_{j=1}^{n}(X_{ij}-X_{kj})^2}$$

其中，*n* 为向量维数，*i* 和 *k* 为行号。

5 系统实现效果(System implementation effect)

目前，已利用该平台对“西安奔驰女车主维权事件”的整个微博舆情事件发展全过程进行了追踪与分析，对 2020 年 3 月至 2020 年 6 月新冠疫情发生期间的微博舆情热点话题进行了趋势预警，对 2020 年 12 月至 2021 年 5 月韶关地区各旅游景点的微博舆情进行了采集及预警处理。系统部分运行效果如图 5 至图 7 所示。图 5 为舆情基本信息总览，图 6 为舆情数据采集来源及占比分析，图 7 为各类事件随时间发展的趋势变化展示。在实际应用中，系统整体运行稳定性较高，在单机日采集量约 100 万条时，对数据分析的有效率可以达到 90%以上，当并发采集的数据量过大时，数据分析的效率有所下降，研究人员后期将在降数据规模算法方面进行进一步的优化。

图 5 舆情基本信息

Fig. 5 Basic public opinion information

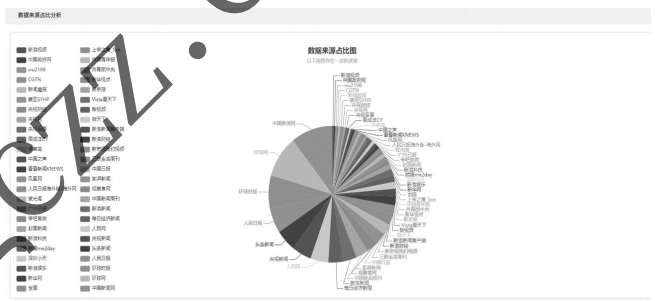

图 6 舆情数据来源展示

Fig. 6 Display of public opinion data sources

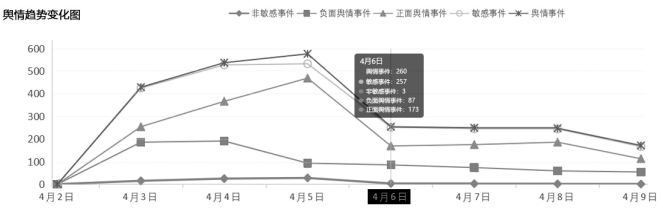

图 7 舆情趋势变化展示

Fig. 7 Display of public opinion trend change

6 结论(Conclusion)

设计并实现了一个基于大数据的集舆情信息采集、去重、分析、处理及可视化的综合平台。该平台主要对采集到的用户指定话题的微博数据进行加工处理后，通过业务子系统展示出来，清晰直观地体现微博某热点话题的传播速度，以及公众对该舆情事件的态度。通过在相关舆情事件开展追踪、分析、预警及处理的实际应用过程中，系统稳定性较高，整体表现良好。

（下转第 54 页）