

## 第八届“泰迪杯”全国数据挖掘挑战赛论文报告

所选题目： C 题

“智能政务”中的文本挖掘应用

综合评定成绩： \_\_\_\_\_

评委评语：

评委签名：

## “智能政务”中的文本挖掘应用

### 摘要

随着科技的快速发展，微信、微博、市长信箱、阳光热线等网络问政平台逐步成为政府了解民意、汇聚民智、凝聚民心的重要渠道，导致各类社情民意相关的文本数据量不断攀升，给传统的主要依靠人工来进行留言划分和热点整理的相关部门的工作带来了巨大的挑战。随着大数据、云计算和人工智能的迅速发展，如何建立基于自然语言处理技术的智能政务使之成为国家治理的新趋势并且能够提升政务工作效率和管理水平也是一门重要的研究对象。

根据给出的问题，我们对所给数据做了挖掘分析和筛选，过滤掉不符合的数据，便于解决给出的问题。

在第一题中，我们首先使用 jieba 分词过滤掉意见文本中无用的符号或词语，得到稀疏数组，通过 TF-IDF 模型训练，再使用 jieba 分组来处理三级标签，随后将两者进行相似度比较，从而将意见分配到对应的一级标签，同时判断与人工处理结果是否一致然后我们在原来 excel 的基础上创建名字为自动分配是否为之匹配的新列来进行标记（0 为不匹配，1 为匹配），对各类总数，查准数，查全数进行统计并放入新的 excel 表格，最后我们通过给出的 F-Score 来进行模型评价。

在第二题中，我们先将数据导入到数据库，然后第一步是对标题和详情进行分词、去停用词；第二步是文本特征选择和文本表示，将分词的结果转换成向量，再将此向量转换成键值对，降低相似度比较的时间复杂度；第三步就是通过之前得到的键值对进行文本相似度的比较，将相似度大于 75% 的列为相同类型的问题；第四步是根据自定义好的热度公式：“热度 = (相同类型问题的最高热度) + 20 + 点赞数 \* 2 - 反对数”，即是每条问题增加 20 点热度，点赞增加 2 点热度，反对减少 1 点热度；最后一步是根据热度来对问题进行排序，选出热度前五的问题类型，生成最终结果。

在第三题中，我们的思路是：对于所有答复，先通过层次分析模型判断相关性、完整性、可解释性的重要程度，再通过对答复平均时间，回复文字长度与留言长度的平均比例，回复内容占五个热点问题的比例的计算，对答复内容进行综合评价。

**关键词：**相似度算法 jieba 分词 层次分析模型 TF-IDF 模型 机器学习

## Abstract

With the rapid development of technology, online questioning platforms such as WeChat, Weibo, Mayor's Mailbox, and Sunshine Hotline have gradually become an important channel for the government to understand public opinion, gather people's wisdom, and gather people's hearts, leading to the continuous increase in the amount of text data related to various social conditions and opinions, Which has brought huge challenges to the work of traditional departments that mainly rely on humans for message division and hotspot sorting. With the rapid development of big data, cloud computing and artificial intelligence, the establishment of smart government affairs based on natural language processing technology has become a new trend of national governance. The formation of smart government affairs has greatly promoted the efficiency and management level of government affairs.

According to the given problems, we did mining analysis and screening of the given data to filter out the non-conforming data, so as to solve the given problems.

In the first question, we first use jieba word segmentation to filter out useless symbols or words in the opinion text, get a sparse array, train through the TF-IDF model, and then use jieba grouping to process three-level labels, and then compare the two similarities. Compare, so that the opinions are assigned to the corresponding first-level labels, and at the same time judge whether they are consistent with the results of manual processing. Then we create a new column based on the original excel to automatically match the matching to mark (0 is not matching, 1 is a match), make statistics on all kinds of total numbers, accurate numbers, and complete numbers and put them into a new excel table. Finally, we evaluate the model through the given F-Score.

In the second question, we first import the data into the database, then the first step is to segment the title and details, and to stop the words; the second step is text feature selection and text representation, converting the results of the word segmentation into a vector, Then convert this vector into key-value pairs to reduce the time complexity of the similarity comparison; the third step is to compare the text similarity through the previously obtained key-value pairs, and classify the similarity greater than 75% as the same type of problem; The fourth step is based on the customized heat formula: "Hotness = (maximum heat of the same type of question) + 20 + number of likes \* 2-antilog", that is, each question is increased by 20 points, and the likes are increased Two points of heat, opposed to reducing one point of heat; the last step is to sort the questions according to the heat, select the top five types of questions, and generate the final result.

In the third question, our idea is: for all responses, first determine the importance of relevance, completeness, and interpretability through the analytic hierarchy process model, and then by the average time of the response, the average ratio of the length of the reply text to the length of the message, Calculate the proportion of the reply content to the five hot questions, and make a comprehensive evaluation of the reply content.

**Keywords:** Similarity algorithm   Jieba word segmentation AHP model   TF-IDF model

# 目录

1. 挖掘目标.....	5
1.1 问题背景.....	5
1.2 目标任务.....	5
2. 分析方法与过程.....	6
2.1 问题分析.....	6
2.2 总体流程图.....	7
3. 数据预处理.....	7
3.1 数据筛选.....	7
3.2 数据统计.....	8
4. 建立模型及问题求解.....	8
4.1 问题一（代码见附录一，二，三）.....	8
4.2 问题二（代码见附录四，五，六，七，八）.....	10
4.3 问题三(代码见附录九).....	12
5. 结果分析.....	14
6. 结论.....	16
7. 算法的改进和推广.....	16
7.1 算法的改进.....	16
7.2 算法的推广.....	16
8. 参考文献.....	16
附录一.....	17
附录二.....	18
附录三.....	21
附录四.....	21
附录五.....	23
附录六.....	25
附录七.....	26
附录八.....	28
附录九.....	29

# 1. 挖掘目标

## 1.1 问题背景

随着网络问政平台的普及，使得各类社情民意相关的文本数据量不断的攀升，给传统的人工来划分留言类型和热点管理带来了巨大的挑战。如何运用互联网+、大数据、云计算、人工智能，建立起一套基于自然语言处理技术的智慧政务已是一个热门话题，也是当今社会治理创新发展的新趋势。如何将智能政务系统中的数据进行加工处理也是一门很有普遍意义的课题。

## 1.2 目标任务

题目收集了来自互联网公开来源的群众问政留言记录，及相关部门对部分群众留言的答复意见，此数据具有很高的可信度和不规则性。我们将利用这些数据来解决所给出的问题。

**问题一：**对群众留言进行分类，我们需要按照一定的划分体系（三级标签体系）来对留言进行分类并进行相应的计数。最后，我们需要建立关于留言内容的一级标签分类模型，来提高留言分类的准确性。

**问题二：**对热点问题的挖掘。所谓热点问题—某一时段内群众集中反映的某一时间。对于热点问题，如果及时发现，有助于相关部门进行有针对性地处理，提升办事效率。我们需要将根据附件 3，首先对某一时段内反映特定地点或特定人群的留言进行分类并定义合理的热度评价指标，得出结果。随后按照题目规定格式得出热点问

题存放相应文件中，并将对应的热点问题对应的留言存放另一相应文件中。

**问题三：**对于给出的相关部门对留言的答复意见，我们需要从答复的相关性、完整性、可解释性等，从多角度对答复意见的质量进行评价，并给出一套相对优越的评价方案加以实现

## 2. 分析方法与过程

### 2.1 问题分析

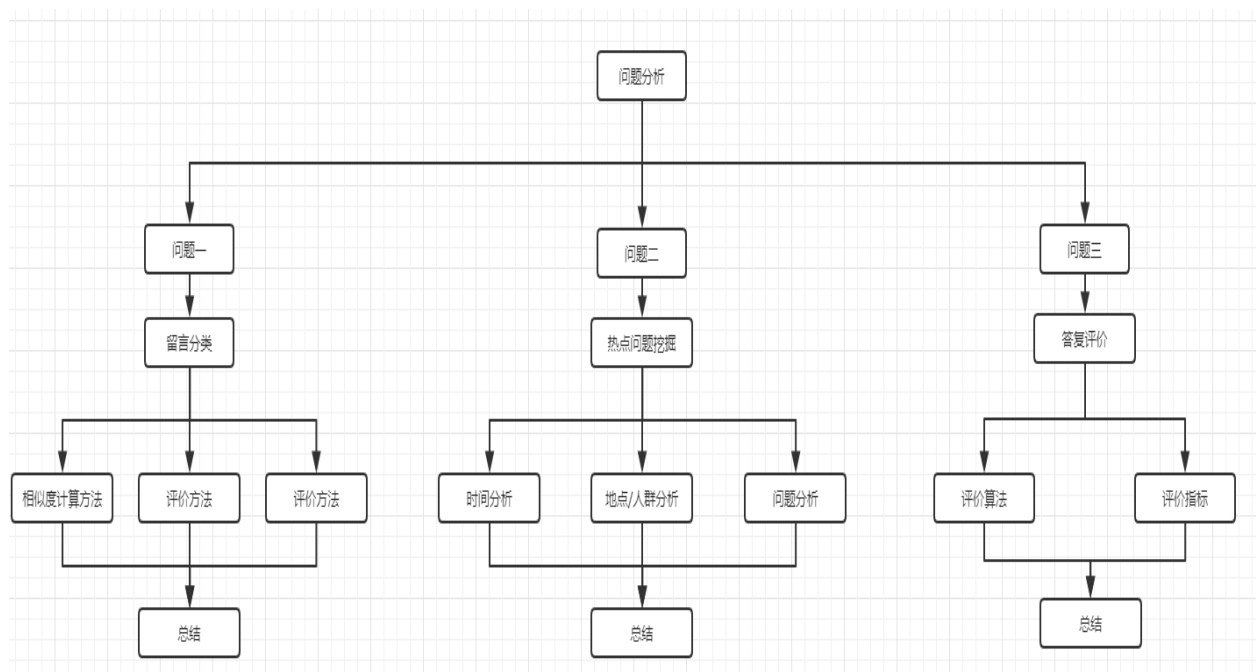


图 2.1 问题分析图

## 2.2 总体流程图

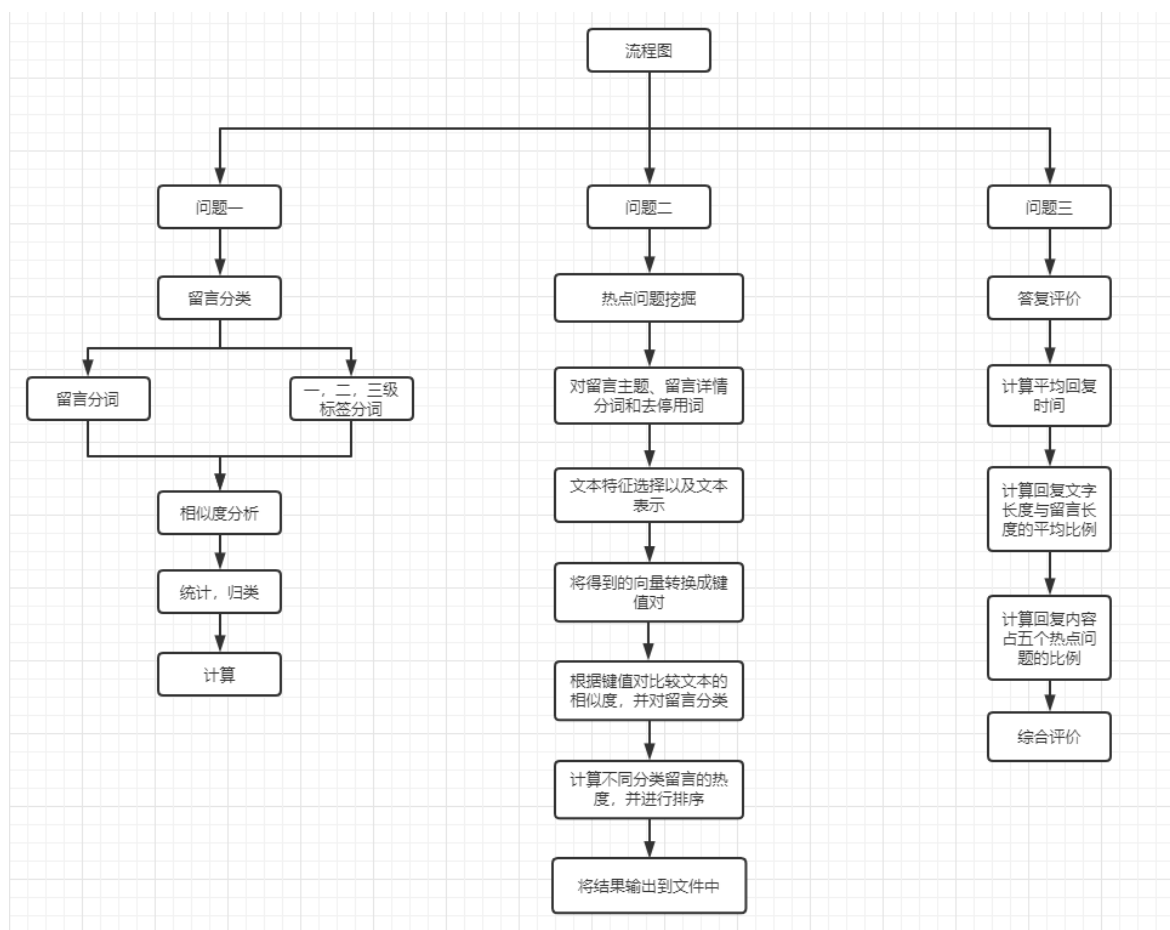


图 2.2 总体流程图

## 3. 数据预处理

### 3.1 数据筛选

①根据题目所给数据，我们将特殊字符运用正则表达式来进行处理，例如数据中存在“前不久去人事部门办理人事续聘手续，可每年的档案托管费要 240 元，感觉太贵。0.0”，我们会将那些不必要的字符，表情，标点等特殊字符去掉。

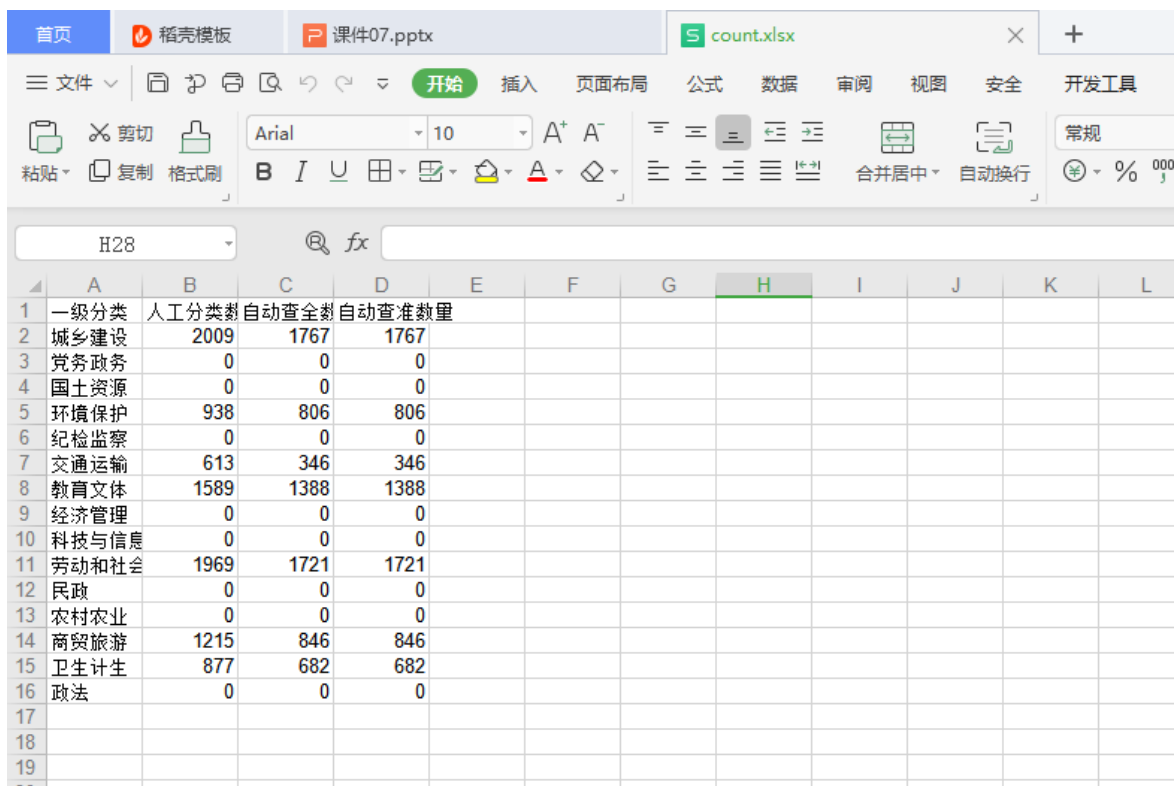
②对数据进行分词，运用 jieba 分词，将一段连续话分成一个个具有不同词性的词，便于后续工作进行分类统计。

③过滤数据中无意义的词，例如：的、了、啊、哈、啦等，保留数据中比较重要的词，并且对数据进行文本特征选择，用数学的方法选取最具分类信息特征的词

④最后我们运用文本表示，将所有重要的词放在一列，随后就可以通过这些被处理的数据用于后续的处理

## 3.2 数据统计

①首先对整体的数据进行计数统计，找到各个数据联系并对数据进行优化。通过相似度算法，我们将原有的数据进行统计得出归类的数据放入到 count.xlsx 文件中。



	A	B	C	D	E	F	G	H	I	J	K	L
1	一级分类	人工分类数	自动查全数	自动查准数								
2	城乡建设	2009	1767	1767								
3	党务政务	0	0	0								
4	国土资源	0	0	0								
5	环境保护	938	806	806								
6	纪检监察	0	0	0								
7	交通运输	613	346	346								
8	教育文体	1589	1388	1388								
9	经济管理	0	0	0								
10	科技与信息	0	0	0								
11	劳动和社会	1969	1721	1721								
12	民政	0	0	0								
13	农村农业	0	0	0								
14	商贸旅游	1215	846	846								
15	卫生计生	877	682	682								
16	政法	0	0	0								
17												
18												
19												

## 4. 建立模型及问题求解

### 4.1 问题一（代码见附录一，二，三）

(1) 先将主题和留言详情生成分词列表

例：A市西湖建筑集团占道施工有安全隐患->['A', '市', '西湖', '建筑',



‘集团’，‘占’，‘道’，‘施工’，‘有’，‘安全隐患’]

(2) 基于主题和留言建立字典，并获取字典特征数

例：A市西湖建筑集团占道施工有安全隐患->(Dictionary(60 unique tokens: ['A', '占', '安全隐患', '市', '建筑']...))

(3) 基于词典将分词列表集转化为稀疏向量集

例：A市西湖建筑集团占道施工有安全隐患->[[ (0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1), (7, 1), (8, 1), (9, 1)]]

(4) 创建 TF-IDF 模型，传入稀疏向量集进行训练

(5) 将一级标签二级标签三级标签拼接成的字符串也转化为稀疏向量

(6) 用训练好的结果集处理 (5) 产生的稀疏向量

(7) 相似度计算，并在 excel 创建新列用于存放是否与人工监测结果相同(代码见附录一)

	A	B	C	D	E	F	G
1	留言编号	留言用户	留言主题	留言时间	留言详情	一级标签	自动分类是否与之匹配
2	24	A0007401	A市西湖建	2020/1/6	1A3区大道	城乡建设	1
3	37	U0008473	A市在水一	2020/1/4	1位于书院路	城乡建设	1
4	83	A0006399	投诉A市A	2019/12/30	尊敬的领导	城乡建设	1
5	303	U0007137	A1区蔡锷	2019/12/6	A1区A2区	城乡建设	1
6	319	U0007137	A1区A2区	2019/12/5	A1区A2区	城乡建设	1
7	379	A0001677	投诉A市盛	2019/11/28	我在2015年	城乡建设	1
8	382	U0005806	咨询A市楼	2019/11/27	由于西地省	城乡建设	1
9	445	A0001920	A3区桐梓	2019/11/15	尊敬的胡干	城乡建设	1
10	476	U0003167	反映C4市	2019/11/15	我们是梅家	城乡建设	1
11	530	U0008488	A3区魏家	2019/11/10	尊敬的A市	城乡建设	1
12	532	U0008488	A市魏家坡	2019/11/10	尊敬的A市	城乡建设	1
13	673	A0008064	A2区秦华	2019/10/24	请求依法出	城乡建设	1

(8) 通过一级标签和(7)创建的新列进行分析，创建新的 excel 放入对人工分类数，自动查全查准数进行统计(代码见附录二)

	A	B	C	D	E
1	一级分类	人工分类数量	自动查全数量	自动查准数量	
2	城乡建设	2009	1767	1767	
3	党务政务	0	0	0	
4	国土资源	0	0	0	
5	环境保护	938	806	806	
6	纪检监察	0	0	0	
7	交通运输	613	346	346	
8	教育文体	1589	1388	1388	
9	经济管理	0	0	0	
10	科技与信息	0	0	0	
11	劳动和社会	1969	1721	1721	
12	民政	0	0	0	
13	农村农业	0	0	0	
14	商贸旅游	1215	846	846	
15	卫生计生	877	682	682	
16	政法	0	0	0	
17					

(9) 通过给出的 F-Score 来进行模型评价(代码见附录三)

得到结果为：0.78925 (结果保留五位小数)

## 4.2 问题二（代码见附录四，五，六，七，八）

（1）先将数据进行分词和去停用词，得到如下结果（部分示例）

msgNo	msgTitle	withoutStop	detailWithoutStop
188006	A3区一米阳光摄影艺术摄影是否合法纳税?	A3 区 一米阳光 摄影 艺术摄影 是否 合法 纳税	座落在 A 市 A3 区联丰路米兰春天 G2 栋 320 一家 名叫一米阳光 摄影 艺术摄影 摄影 摄影 年单 一个
188007	南海A6区道路命名规划初步成果公示和城乡门牌问题	南海 A6 区 道路 命名 规划 初步 成果 公示 城乡 门牌 问题	A 市 A6 区 道路 命名 规划 已经 初步 转化 成为 正式 成果 希望 加快 完成 路名 规范 道路
188031	反映A7县春华镇金鼎村水坝路、自来水到户的问题	反映 A7 县 春华 镇 金鼎 村 水坝 路 自来水 到户 问题	春华 镇 金鼎 村 七组 村民 不知 是否 相关 水坝 路 到户 政策 自来水 到户 政策 政府 主导 投资 村民
188039	A2区黄兴路步行街古道巷卫生间设施外排	A2 区 黄兴 路 步行街 古道 巷 卫生间 设施 外排	靠近 黄兴 路 步行街 城南路 街道 小巷 第一步 再 建 厕所 卫生间 设施 旁边 外 第一 单元 住户 卫生间
188059	A市A3区中海国际社区三期与四期中间空地夜间施工噪音扰民	A 市 A3 区 中海 国际 社区 三期 四期 中间 空地 夜间 施工 噪音 扰民	A 市 A3 区 中海 国际 社区 三期 四期 中间 空地 夜间 施工 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188073	A3区麓泉社区单方面改变麓谷明珠小区6栋架空层使用性质	A3 区 麓 泉 社区 单方面 改变 麓 谷 明珠 小区 栋 架空 层 使用 性质	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188074	A3区富康新村房产的性质是什么?	A2 区 富 康 新村 房产 性质	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188119	A市地铁违规用工问题的函疑	A 市 地铁 违规 用工 问题 函疑	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188170	A市6路公交车随意变道通行	A 市 路 公交车 随意 变道 通行	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188249	A3区保利麓谷林语梧桐梓城路与麓松路交汇处地铁流量点施工扰民	A3 区 保利 麓 谷 林语 梧桐 梓城 路 与 麓 松 路 交汇 处 地铁 流量 点 施工 扰民	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188251	A7县特立路与东四路口晚高峰太堵，建议调整信号灯配时	A7 县 特立 路 东四 路口 晚 高峰 太堵 建议 调整 信号 灯 配时	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188260	A3区青青家园小区水果摊零食杂货占道摆放空调扰民	A3 区 青青 家园 小区 水果 摊 零食 杂货 占道 摆放 空调 扰民	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188396	关于拆除麓高龙墓在西地曾南学院宿舍旁安装变压器请求	拆除 麓 高 龙 墓 在西 地 曾 南 学院 宿舍 旁 安装 变压器 请求	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188399	A市利保壹号公馆项目夜间噪声扰民	A 市 利保 壹 号 公馆 项目 夜间 噪声 扰民	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188409	A市地铁3号线星沙大道站地铁出入口设置极不合理!	A 市 地铁 3 号 线 星 沙 大道 站 地铁 出入口 设置 极 不合理	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188414	A4区北辰小区非法占道问题何时能解决?	A4 区 北辰 小区 非法 占道 问题 何时 能 解决	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188416	湘湖K3县乡村医生卫生室执业许可证	湘 湖 K3 县 乡 村 医生 卫生 室 执业 许可 证	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188451	A7县春华镇石塘铺村党员家开麻将馆	A7 县 春华 镇 石塘 铺 村 党员 家 开 麻将 馆	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188455	西院异地办理出国签证的问题	西 院 异地 办理 出国 签证 问题	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188467	投诉A市波斯顿英语培训学校拖延退费	投诉 A 市 波斯 顿 英语 培训 学校 拖延 退费	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188475	A6区松源国际广场停车场违建乱建现象严重	A6 区 松源 国际 广场 停车场 违建 乱建 现象 严重	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188535	A7县时代星城4号楼有非法经营的家政服务	A7 县 时代 星城 4 楼 有 非法 经营 的 家政 服务	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188546	A2区佳兆业水新小区垃圾无人处理	A2 区 佳兆 业 水新 小区 垃圾 无人 处理	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188553	A市沙坪老街上有无证理疗馆骗取老人钱财	A 市 沙坪 老 街 上 有 无 证 理 疗 馆 骗 取 老 人 财 钱	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188560	A市港湾餐饮店拖欠工资，员工维权难	A 市 港湾 餐 饮 店 欠 拖 工 资 员 工 维 权 难	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188592	A市长云时代小区三期后面要建垃圾站	A 市 长 云 时 代 小 区 三 期 后 面 要 建 垃圾 站	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188665	A市松雅湖南方韵筑2楼有传销窝点	A 市 松雅 湖南 方 韵 筑 2 楼 有 传 销 窝 点	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188676	希望A市政府出台解决落实退休教师待遇各项补贴的长效办法	希望 A 市 政府 出 台 解 决 落 实 退 休 教 师 待遇 各 项 补 贴 长 效 办 法	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188691	举报A市杜弘毅教育培训机构涉嫌欺诈	举报 A 市 杜 弘 毅 教 育 培 训 机 构 涉 嫌 欺 诈	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业
188774	A2区政府东门万美路路段经常有改装车飙车，真的很扰民	A2 区 政府 东 门 万 美 路 段 经 常 有 改 装 车 飙 车 真 的 很 扰 民	麓 泉 社区 麓 谷 明珠 小区 栋 居民 最近 感觉 噪音 扰民 旁边 空地 一直 处于 三不 管 状态 物业

（2）运用文本特征选择和文本表示，将分词的结果转成向量，并将向量转成键值对的形式。得到结果如下（部分示例）

msgNo	arrIndex	arrValue
188006	144,286,2254,2920,4580,6138,6427	1,1,1,1,1,1,1
188007	147,1234,1448,2338,2345,2574,3953,6656,7157,7378,7389	1,1,1,1,1,1,1,1,1,1,1
188031	148,1464,2156,4572,5105,6384,7347,7389	1,1,1,1,1,1,1,1
188039	143,949,1995,2204,2692,5016,6091,7726	1,1,1,1,1,1,1,1
188059	144,336,586,601,2414,2425,2486,2732,4047,4467,5903,5989	1,1,1,1,1,1,1,1,1,1,1,1
188073	144,983,1849,1943,3090,3898,4294,4549,4831,5903	1,1,1,1,1,1,1,1,1,1
188074	143,1664,3898,3978,4433	1,1,1,1,1,1
188119	2534,5651,6803,7064,7389	1,1,1,1,1,1
188170	1190,2194,7142,7455	1,1,1,1
188249	144,452,808,1011,1351,2534,2554,4047,4467,4805,4887,6777	1,1,1,1,1,1,1,1,1,1,1,1
188251	148,486,1029,2823,3733,5546,6766,6891,7200,7675	1,1,1,1,1,1,1,1,1,1
188260	144,672,1194,3027,3090,4047,4249,5469,5995,7146,7506,7534	1,1,1,1,1,1,1,1,1,1,1,1
188396	2189,2378,2967,3051,4110,6343,6623,6750	1,1,1,1,1,1,1,1
188399	1246,2412,2657,2732,3310,4047,7563	1,1,1,1,1,1,1,1
188409	410,1379,2232,2534,2779,4566,6708	1,1,1,2,1,1,1
188414	145,970,1573,3090,4295,6667,7389,7541	1,1,1,1,1,1,1,1
188416	188,691,1859,1992,4033,6698	1,1,1,1,1,1
188451	148,1141,3038,4572,5853,7316,7724	1,1,1,1,1,1,1,1
188455	1387,1510,2345,3777,6061,7389	1,1,1,1,1,1,1,1
188467	2605,2941,4075,4124,5385,6459,7100	1,1,1,1,1,1,1,1
188475	147,542,702,1094,1600,2486,3649,5603,7057	1,1,1,1,1,1,1,1,1,1
188535	148,3037,4529,4558,7543	1,1,1,1,1,1
188546	143,1611,2556,2665,3090,4486,5098	1,1,1,1,1,1,1,1
188553	4652,5180,5620,6281,6571,7292,7656	1,1,1,1,1,1,1,1
188560	2325,3381,4127,6195,7616	1,1,1,1,1,1
188592	336,2302,2562,3090,3565,3977,4529,6646	1,1,1,1,1,1,1,1,1,1
188665	493,933,3425,4955,6008,6414	1,1,1,1,1,1,1,1
188679	1385,1509,2239,3401,3603,4338,6515,6595,6667,7084,7361	1,1,1,1,1,1,1,1,1,1,1,1

(3) 将得到的键值对数据进行文本相似度比较，随后将相似度大于 75% 的列归为相同类型的问题。即计算得到的相似度 > 75% (a to b 或 b to a 的相似度 > 75%) 的两条问题。

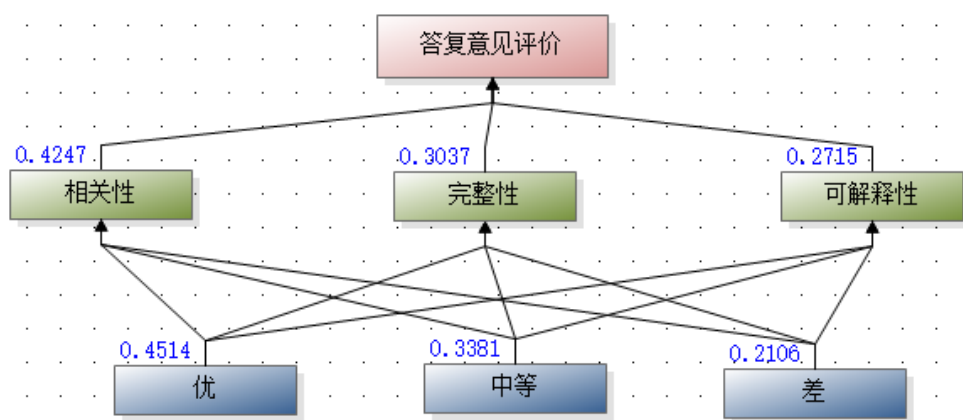
	noa	nob	btoa	atob
▶	188007	198975	0.36363636363636365	0.8
	188059	191327	0.6666666666666666	0.8888888888888888
	188059	215834	1	1
	188260	261103	0.5	0.8571428571428571
	188414	215105	0.625	0.8333333333333334
	188416	323149	1	1
	188467	254068	0.8571428571428571	0.8571428571428571
	188467	282638	0.5714285714285714	0.8
	188876	246915	0.4444444444444444	0.8
	189093	197669	0.875	1
	189093	214447	0.625	0.8333333333333334
	189093	214709	0.875	1
	189093	240551	0.75	1
	189180	270015	0.5714285714285714	1
	189381	287386	1	1
	189381	360104	0.875	1
	189739	214447	0.8571428571428571	1
	189739	240551	0.7142857142857143	0.8333333333333334
	189739	254063	0.8571428571428571	1
	189739	266213	0.8571428571428571	0.75
	189950	208285	0.625	0.8333333333333334
	189950	231136	1	1
	189950	253040	1	1
	189950	255008	0.625	0.8333333333333334
	189950	261072	0.625	0.8333333333333334
	189950	266665	0.625	0.8333333333333334
	190077	274826	1	1
	190108	213464	0.8	0.4444444444444444
	190108	216824	0.8	0.36363636363636365
	190108	217700	0.8	0.5714285714285714
	190108	235362	0.8	0.4

(4) 根据自定义好的热度公式：“热度=(相同类型问题的最高热度) + 20 + 点赞数 \* 2 - 反对数”，即是每条问题增加 20 点热度，点赞增加 2 点热度，反对减少 1 点热度，最后得出结果

msgNo	msgTitle	stype	heat
268251	西地省58车贷立案近半年毫无进展，单位回复让人心寒	308	5049
272413	西地省A市58车贷恶性退出，A4区立案已近半年毫无进展	308	5049
272858	A市58车贷恶性退出案件为什么不发布案情进展通报？	308	5049
194343	承办A市58车贷案警官应跟进关注留言	308	5049
214238	请问A4区公安派出所对58车贷一案办案的进度如何了	308	5049
217032	严惩A市58车贷特大集资诈骗案保护伞	308	5049
218132	再次请求过问A市58车贷案件进展情况	308	5049
220711	请书记关注A市A4区58车贷案	308	5049
223787	西地省58车贷案件创造全国典型诈骗案，立案至今无公告	308	5049
226265	恳请A市经侦公正办理58车贷案件，还我们受害人一个公道	308	5049
234320	不要让A市因为58车贷案件而臭名远扬	308	5049
240554	A市58车贷老板跑路美国，经侦拖延办案	308	5049
254532	A市58车贷恶性退出立案近半年没有发过一次案情通报	308	5049
264119	58车贷立案五个月过去，A4区公安分局未公布过任何案情	308	5049
208069	A5区五矿万境K9县的开发商与施工方建房存在质量问题	309	4384
208636	A市A5区汇金路五矿万境K9县存在一系列问题	309	4384
215507	A市五矿万境K9县存在严重的消防安全隐患	309	4384
234086	A市五矿万境K9县房子的墙壁又开裂了	309	4384
239227	恳请A7县依法拘留星沙K9县郡b栋302业主，并拆除他违章搭建的书房等	309	4384
252650	A市五矿万境K9县交房后仍存在诸多问题	309	4384
262599	A市五矿万境K9县房屋出现质量问题	309	4384
275491	A市五矿万境K9县负一楼面积缩水	309	4384
223297	反映A市金毛湾配套入学的问题	(Null)	3539
191951	A4区绿地海外滩小区距渝长厦高铁太近了	311	1522
202575	咨询A市绿地海外滩二期与长赣高铁问题	311	1522
216316	A4区绿地海外滩二期业主被噪音扰得快烦死了	311	1522
243551	A市至赣州高铁对绿地海外滩二期小区影响太大了	311	1522
246974	渝长厦高铁的长赣高铁征地路线对A6区周边小区影响巨大	311	1522
263672	A4区绿地海外滩小区距长赣高铁最近只有30米不到，合理吗？	311	1522
266931	按照当前的高铁规划，A市绿地海外滩小区会饱受噪音困扰	311	1522
268299	惊！！A市伊景园滨河苑商品房竟然捆绑销售车位	14	632

### 4.3 问题三(代码见附录九)

(1) 首先用层次分析模型判断相关性，完整性和可解释性对于问题的重要性



#### 最终结果

备选方案	权重
优	0.4514
中等	0.3381
差	0.2106

#### 1. 答复意见评价 一致性比例: 0.0486; 对"答复意见评价"的权重: 1.0000; $\lambda_{\max}$ : 3.0505

答复意见评价	相关性	完整性	可解释性	$W_i$
相关性	1.0000	1.7500	1.2500	0.4247
完整性	0.5714	1.0000	1.4000	0.3037
可解释性	0.8000	0.7143	1.0000	0.2715

#### 2. 相关性 一致性比例: 0.0000; 对"答复意见评价"的权重: 0.4247; $\lambda_{\max}$ : 3.0000

相关性	优	中等	差	$W_i$
优	1.0000	1.0000	2.0000	0.4000
中等	1.0000	1.0000	2.0000	0.4000
差	0.5000	0.5000	1.0000	0.2000

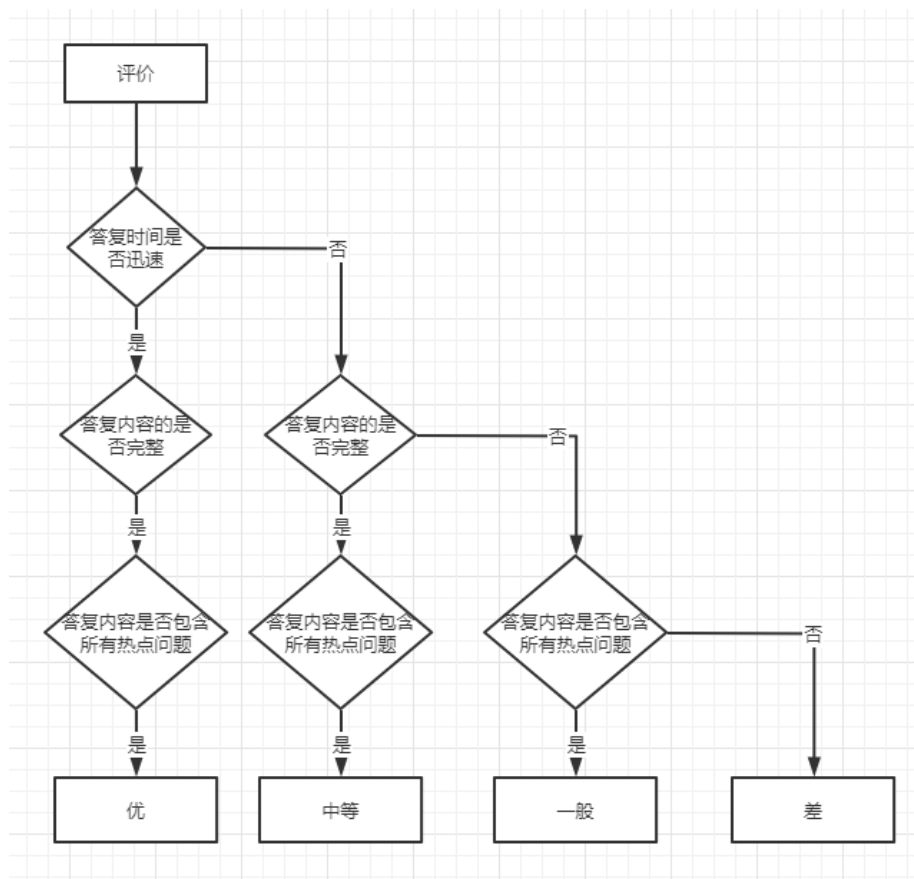
#### 3. 完整性 一致性比例: 0.0115; 对"答复意见评价"的权重: 0.3037; $\lambda_{\max}$ : 3.0120

完整性	优	中等	差	$W_i$
优	1.0000	1.5000	2.5000	0.4893
中等	0.6667	1.0000	1.2000	0.2924
差	0.4000	0.8333	1.0000	0.2184

#### 4. 可解释性 一致性比例: 0.0115; 对"答复意见评价"的权重: 0.2715; $\lambda_{\max}$ : 3.0120

可解释性	优	中等	差	$W_i$
优	1.0000	1.5000	2.5000	0.4893
中等	0.6667	1.0000	1.2000	0.2924
差	0.4000	0.8333	1.0000	0.2184

- (2) 通过(1)的判断选出三个用于评价的指标①答复平均时间②答复内容是否完整③答复内容是否包含问题二中查找出的所有热点问题
- (3) 做出流程图



## 5. 结果分析

**第一题：**数据通过分词，字典特征数和稀疏向量的处理过后，所获得的结果能够符合题目的要求，通过 F-Score 来评价，所获得的结果表明这类方法的实现准确性较高。

通过计算最终结果为 0.78925 (结果保留五位小数)，由此可见算法的准确度在接受范围内。

**第二题：**数据通过分词，去停用词处理后，接着将分词的结果转化成向量，再由向量转化成键值对的形式，此方法可以降低相似度比较的时间复杂度，能够提高运算的效率，最后将结果通过相似度比较，按照自己规定的公式，可以得出热点问题。

从结果分析得出，若字段 heat 数值越大，则热度就越高



msgNo	msgTitle	stype	heat
268251	西地省58车贷立案近半年毫无进展，单位回复让人心寒	308	5049
272413	西地省A市58车贷恶性退出，A4区立案已近半年毫无进展	308	5049
272858	A市58车贷恶性退出案件为什么不发布案情进展通报？	308	5049
194343	承办A市58车贷案警官应跟进关注留言	308	5049
214238	请问A4区公安派出所对58车贷一案办案的进度如何了	308	5049
217032	严惩A市58车贷特大集资诈骗案保护伞	308	5049
218132	再次请求过问A市58车贷案件进展情况	308	5049
220711	请书记关注A市A4区58车贷案	308	5049
223787	西地省58车贷案件创造全国典型诈骗案，立案至今无公告	308	5049
226265	恳请A市经侦公正办理58车贷案件，还我们受害人一个公道	308	5049
234320	不要让A市因为58车贷案件而臭名远扬	308	5049
240554	A市58车贷老板跑路美国，经侦拖延办案	308	5049
254532	A市58车贷恶性退出立案近半年没有发过一次案情通报	308	5049
264119	58车贷立案五个月过去，A4区公安分局未公布过任何案情	308	5049
208069	A5区五矿万境K9县的开发商与施工方建房存在质量问题	309	4384
208636	A市A5区汇金路五矿万境K9县存在一系列问题	309	4384
215507	A市五矿万境K9县存在严重的消防安全隐患	309	4384
234086	A市五矿万境K9县房子的墙壁又开裂了	309	4384
239227	恳请A7县依法拘留星沙K9县郡b栋302业主，并拆除他违章搭建的书房等	309	4384
252650	A市五矿万境K9县交房后仍存在诸多问题	309	4384
262599	A市五矿万境K9县房屋出现质量问题	309	4384
275491	A市五矿万境K9县负一楼面积缩水	309	4384
223297	反映A市金毛湾配套入学的问题	(Null)	3539
191951	A4区绿地海外滩小区距渝长厦高铁太近了	311	1522
202575	咨询A市绿地海外滩二期与长赣高铁问题	311	1522
216316	A4区绿地海外滩二期业主被噪音扰得快烦死了	311	1522
243551	A市至赣州高铁对绿地海外滩二期小区影响太大了	311	1522
246974	渝长厦高铁的长赣高铁征地路线对A6区周边小区影响巨大	311	1522
263672	A4区绿地海外滩小区距长赣高铁最近只有30米不到，合理吗？	311	1522
266931	按照当前的高铁规划，A市绿地海外滩小区会饱受噪音困扰	311	1522
268299	惊！！A市伊景园滨河苑商品房竟然捆绑销售车位	14	632

### 第三题：

- (1) 总体平均回复时间为 26.35443 天(结果保留五位小数)，证明回复效率中等。
  - (2) 总体回复内容和意见内容的比例为 1.58973(结果保留五位小数)，证明回复内容基本长于提问内容，回复可解释性，完整性较高。
  - (3) 回复列表中对前五热点问题均有涉及，证明回复相关性较高。
- 综上所述：回复总体结果为中等。

## 6. 结论

在遇到数据量多，数据量不规则需要进行分类的时候，我们可以采用 jieba 分词先将数据解析，随后通过文本特征，稀疏向量等操作后，建立 TF-IDF 模型将向量集进行处理，最后通过相似度计算来统计所获得的结果

在实现热点问题查找，通过分词，去停用词这些处理数据的方法先前过滤掉没有用处的词语，随后通过文本特征，稀疏向量，然后通过相似度计算，再加上自定义的方法，在此题中，我们主要运用自定义热度公式：“热度=(相同类型问题的最高热度) + 20 + 点赞数 \* 2 - 反对数”，即是每条问题增加 20 点热度，点赞增加 2 点热度，反对减少 1 点热度

## 7. 算法的改进和推广

### 7.1 算法的改进

- (1) 算法在遇到长文本时，时间复杂度较高，可以在文本过长的时候，先清除无用的称呼，感谢，自我介绍等等
- (2) 可以更多次的训练模型，使数据更加精确
- (3) 在处理文本表示的向量时，将二维数组转换成键值对能有效地降低比较文本相似度时的时间复杂度。

### 7.2 算法的推广

- (1) 算法不仅仅限于留言分类，聚集，在其他需要文本分类的地方通常也会有较高的准确性。
- (2) 在热度计算公式的基础上增加热度减少公式，便可以应用在话题的热搜。

## 8. 参考文献

- [1] 施利萍，基于 LDA 模型的微信留言文本发现研究，《科教导刊-电子版》，24 期，



2017 年。

[2] 吴柳 程恺 胡琪, 基于文本挖掘的论坛热点问题时变分析, Computer Engineering & Software, 04 期, 2017 年。

[3] 王春柳, 杨永辉, 邓霏, 赖辉源. 文本相似度计算方法研究综述[J]. 情报科学, 2019, 37(03):158-168.

[4] 江大鹏. 基于词向量的短文本分类方法研究[D]. 浙江大学, 2015.

[5] 谭静. 基于向量空间模型的文本相似度算法研究[D]. 西南石油大学, 2015.

[6] 孙润志. 基于语义理解的文本相似度计算研究与实现[D]. 中国科学院研究生院(沈阳计算技术研究所), 2015.

## 附录一

```
from jieba import lcut
from gensim.similarities import SparseMatrixSimilarity
from gensim.corpora import Dictionary
from gensim.models import TfidfModel
import xlrd
import xlwt
first = xlrd.open_workbook('附件 1.xlsx')
second = xlrd.open_workbook('附件 2.xlsx')
sheet1 = first.sheet_by_name('Sheet1')
sheet2 = second.sheet_by_name('Sheet1')
#输出 excel
f = xlwt.Workbook()
sheet3 = f.add_sheet('Sheet1', cell_overwrite_ok=True)
a = []
k = 0
for j in range(sheet2.nrows):
    a.append(0)
    p = 1
    q = 0
    texts = [sheet2.row_values(j)[2], sheet2.row_values(j)[4]]
    texts = [lcut(text) for text in texts]
    dictionary = Dictionary(texts)
    num_features = len(dictionary.token2id)
```

```

corpus = [dictionary.doc2bow(text) for text in texts]
tfidf = TfidfModel(corpus)
tf_texts = tfidf[corpus]
for i in range(sheet1.nrows):
    if a[k] == 1:
        break
    if(sheet2.row_values(j)[5] == sheet1.row_values(i)[0]):
        p = 2
        kw_vector =
dictionary.doc2bow(lcut(sheet1.row_values(i)[0]+sheet1.row_values(i)[1]+
sheet1.row_values(i)[2]))
        tf_kw = tfidf[kw_vector]
        # 6、相似度计算
        sparse_matrix = SparseMatrixSimilarity(tf_texts, num_features)
        similarities = sparse_matrix.get_similarities(tf_kw)
        if similarities[0]>0 or similarities[1]>0:
            a[k] = 1
    elif p == 2:
        break
for q in range(0,7):
    if q == 6 and k == 0:
        sheet3.write(k,q,"自动分类是否与之匹配")
    elif q == 6 :
        sheet3.write(k,q,a[k])
    else:
        sheet3.write(k,q,sheet2.row_values(j)[q])
    k += 1
f.save('auto.xlsx')

```

## 附录二

```

import xlrd
import xlwt
excel = xlrd.open_workbook('auto.xlsx')
sheet1 = excel.sheet_by_name('Sheet1')
#输出 excel

```

```

f = xlwt.Workbook()
a = []
b = []
for i in range(0, 15):
    a.append(0)
    b.append(0)
sheet3 = f.add_sheet('Sheet1', cell_overwrite_ok=True)
sheet3.write(0, 0, "一级分类")
sheet3.write(0, 1, "人工分类数量")
sheet3.write(0, 2, "自动查全数量")
sheet3.write(0, 3, "自动查准数量")
sheet3.write(1, 0, "城乡建设")
sheet3.write(2, 0, "党务政务")
sheet3.write(3, 0, "国土资源")
sheet3.write(4, 0, "环境保护")
sheet3.write(5, 0, "纪检监察")
sheet3.write(6, 0, "交通运输")
sheet3.write(7, 0, "教育文体")
sheet3.write(8, 0, "经济管理")
sheet3.write(9, 0, "科技与信息产业")
sheet3.write(10, 0, "劳动和社会保障")
sheet3.write(11, 0, "民政")
sheet3.write(12, 0, "农村农业")
sheet3.write(13, 0, "商贸旅游")
sheet3.write(14, 0, "卫生计生")
sheet3.write(15, 0, "政法")
for j in range(sheet1.nrows):
    if sheet1.row_values(j)[5] == "城乡建设":
        if sheet1.row_values(j)[6] == 1:
            b[0] += 1
            a[0] += 1
    elif sheet1.row_values(j)[5] == "党务政务":
        if sheet1.row_values(j)[6] == 1:
            b[1] += 1
            a[1] += 1
    elif sheet1.row_values(j)[5] == "国土资源":
        if sheet1.row_values(j)[6] == 1:
            b[2] += 1
            a[2] += 1

```

```

elif sheet1.row_values(j)[5] == "环境保护" :
    if sheet1.row_values(j)[6] == 1:
        b[3] += 1
    a[3] += 1
elif sheet1.row_values(j)[5] == "纪检监察" :
    if sheet1.row_values(j)[6] == 1:
        b[4] += 1
    a[4] += 1
elif sheet1.row_values(j)[5] == "交通运输" :
    if sheet1.row_values(j)[6] == 1:
        b[5] += 1
    a[5] += 1
elif sheet1.row_values(j)[5] == "教育文体" :
    if sheet1.row_values(j)[6] == 1:
        b[6] += 1
    a[6] += 1
elif sheet1.row_values(j)[5] == "经济管理" :
    if sheet1.row_values(j)[6] == 1:
        b[7] += 1
    a[7] += 1
elif sheet1.row_values(j)[5] == "科技与信息产业" :
    if sheet1.row_values(j)[6] == 1:
        b[8] += 1
    a[8] += 1
elif sheet1.row_values(j)[5] == "劳动和社会保障" :
    if sheet1.row_values(j)[6] == 1:
        b[9] += 1
    a[9] += 1
elif sheet1.row_values(j)[5] == "民政" :
    if sheet1.row_values(j)[6] == 1:
        b[10] += 1
    a[10] += 1
elif sheet1.row_values(j)[5] == "农村农业" :
    if sheet1.row_values(j)[6] == 1:
        b[11] += 1
    a[11] += 1
elif sheet1.row_values(j)[5] == "商贸旅游" :
    if sheet1.row_values(j)[6] == 1:
        b[12] += 1

```

```

        a[12] += 1
    elif sheet1.row_values(j)[5] == "卫生计生" :
        if sheet1.row_values(j)[6] == 1:
            b[13] += 1
            a[13] += 1
        elif sheet1.row_values(j)[5] == "政法" :
            if sheet1.row_values(j)[6] == 1:
                b[14] += 1
                a[14] += 1
for i in range(0,15):
    sheet3.write(i+1,1,a[i])
    sheet3.write(i+1,2,b[i])
    sheet3.write(i+1,3,b[i])

f.save('count.xlsx')
```

### 附录三

```

import xlrd
excel = xlrd.open_workbook('count.xlsx')
sheet1 = excel.sheet_by_name('Sheet1')
s = 0
k = 0
for j in range(1,sheet1.nrows):
    if sheet1.row_values(j)[1] != 0:
        k += 1
        b = sheet1.row_values(j)[2]/sheet1.row_values(j)[1]
        c = sheet1.row_values(j)[3]/sheet1.row_values(j)[1]
        s += 2*b*c/(b+c)
print("%.5f" % (s/k))
```

### 附录四

```

import jieba.analyse
import time
import pymysql.cursors
# 第一步 去停用词
```

```

# 加载本地的停用词文件
def stopwordslist(filepath):
    stopwords = [line.strip() for line in open(filepath, 'r',
encoding='utf-8').readlines()]
    return stopwords

# 去停用词
def seg_sentence(sentence):
    sentence_segged = jieba.cut(sentence.strip())
    stopwords = stopwordslist('../data/cn_stopwords.txt')
    outstr = ''
    for word in sentence_segged:
        if word not in stopwords:
            if word != '\t':
                outstr += word
            outstr += " "
    return outstr

t_start = time.time()
connection = pymysql.connect(host='localhost', port=3306, user='root',
password='981217', db='teddy', charset='utf8',
cursorclass=pymysql.cursors.DictCursor)

cursor = None
try:
    cursor = connection.cursor()

    sql = "SELECT msgNo, msgTitle, msgDetail from t_forum"
    cursor.execute(sql)
    res = cursor.fetchall()

    for i in res:
        no = i['msgNo']
        title = i['msgTitle']
        text = i['msgDetail']

        titleSeg = seg_sentence(title)
        textSeg = seg_sentence(text)

```

```

        subSql = "UPDATE t_forum set withoutStop = '" + titleSeg + "',
detailWithoutStop = '" + textSeg + \
                "' WHERE msgNo = '" + str(no)
        cursor.execute(subSql)
        print("执行完成...")

        connection.commit()
except Exception as e:
    connection.rollback()
    connection.commit()
    print("执行失败")
    print(e.__str__())
finally:
    cursor.close()

```

## 附录五

```

from sklearn.feature_extraction.text import CountVectorizer
import pymysql.cursors

# 第二步 进行文本特征选择 并且将得到的二维数组变成键值对存入数据库

connection = pymysql.connect(host='localhost', port=3306, user='test',
password='test', db='teddy', charset='utf8',
                             cursorclass=pymysql.cursors.DictCursor)

cursor = None
text = []
i, j = 0, 0
try:
    cursor = connection.cursor()

    sql = "SELECT msgNo, withoutStop FROM t_forum"
    cursor.execute(sql)
    res = cursor.fetchall()

    for i in res:
        msgNo = i['msgNo']
        ws = i['withoutStop']
        # 句子
        text.append(ws)

```

```

cv = CountVectorizer()
data = cv.fit_transform(text)
# 特征名
print(cv.get_feature_names())
# 特征 vec
arr = data.toarray()
# 暴力法 预计花费 17 小时
# for i in range(len(res)):
#     temp = arr[i]
#     ci = len(res[i])
#     for j in range(len(res)):
#         tj = arr[j]
#         eq, cj = 0, len(res[j])
#         for k in range(len(tj)):
#             if tj[k] > 0 and temp[k] > 0:
#                 eq += abs(tj[k] - temp[k])
#             si = (eq * 1.0 / ci)
#             sj = (eq * 1.0 / cj)
#             if si > 0.85 or sj > 0.85:
#                 tempSql = "INSERT INTO t_similar VALUES (" +
str(res[i]['msgNo']) + ", " + str(res[j]['msgNo']) + \
#                 ", " + str(si) + ", " + str(sj) + ")"
#                 cursor.execute(tempSql)
#         print(i)
#         if i % 100 == 0:
#             connection.commit()

# 将 vec 转成键值对
for i in range(len(res)):
    temp = arr[i]
    tempSql = "INSERT INTO t_arr VALUE "
    ts = []
    for j in range(len(temp)):
        if temp[j] > 0:
            ts.append("(" + str(res[i]['msgNo']) + ", " + str(j) + ",
" + str(temp[j]) + ")")
    if len(ts) > 0:
        print(tempSql + ",".join(ts))
        cursor.execute(tempSql + ",".join(ts))

```



```

        if i % 100 == 0:
            print(i)
            connection.commit()
        connection.commit()
except Exception as e:
    print(i, j)
    connection.rollback()
    connection.commit()
    print("执行失败")
    print(e.__str__())
finally:
    cursor.close()

```

## 附录六

```

import pymysql.cursors
# 第三步 计算各数据间的相似度

connection = pymysql.connect(host='localhost', port=3306, user='test',
                             password='test', db='teddy', charset='utf8',
                             cursorclass=pymysql.cursors.DictCursor)

cursor = None
try:
    cursor = connection.cursor()
    # 列转行 减少数据冗余度
    sql = "SELECT msgNo ,group_concat(arrIndex Separator ',') AS arrIndex,
group_concat(arrValue Separator ',') " + \
        "AS arrValue FROM t_arr GROUP BY msgNo"
    cursor.execute(sql)
    res = cursor.fetchall()
    for i in range(len(res)):
        ii = str(res[i]['arrIndex']).split(",")
        iv = str(res[i]['arrValue']).split(",")
        ci, ti = len(ii), 0
        for k in range(ci):
            ti += int(iv[k])
    for j in range(len(res)):
        if i == j:
            continue

```

```

        ji = str(res[j]['arrIndex']).split(",")
        jv = str(res[j]['arrValue']).split(",")
        cj, eq, tj = len(ji), 0, 0
        for t in range(cj):
            tj += int(jv[t])
        for k in range(ci):
            for t in range(cj):
                if ii[k] == ji[t]:
                    eq += min(int(iv[k]), int(jv[t]))
        si = (eq * 1.0 / ti)
        sj = (eq * 1.0 / tj)
        if si > 0.75 or sj > 0.75:
            tempSql = "INSERT INTO t_similar VALUES (" +
str(res[i]['msgNo']) + ", " + str(res[j]['msgNo']) + \
            ", " + str(si) + ", " + str(sj) + ")"
            cursor.execute(tempSql)
        if i % 100 == 0:
            print(i)
            connection.commit()
        connection.commit()
except Exception as e:
    connection.rollback()
    connection.commit()
    print("执行失败")
    print(e.__str__())
finally:
    cursor.close()

```

## 附录七

```

import pymysql.cursors
import queue
# 第四步 问题分类

connection = pymysql.connect(host='localhost', port=3306, user='test',
password='test', db='teddy', charset='utf8',
                                cursorclass=pymysql.cursors.DictCursor)

cursor = None
try:

```

```

    cursor = connection.cursor()
    q = queue.Queue()
    sql = "SELECT DISTINCT noa FROM t_similar"
    cursor.execute(sql)
    res = cursor.fetchall()
    tempDict = {}
    for i in res:
        tempDict[i['noa']] = 0
    count = 1
    for i in res:
        if tempDict.get(i['noa']) == 1:
            continue
        q.put(i['noa'])
        tempDict[i['noa']] = 1
        while not q.empty():
            noa = q.get()
            # 更新 type
            updateSql = "UPDATE t_forum SET stype = " + str(count) + " WHERE
msgNo = " + str(noa)
            cursor.execute(updateSql)
            # 查询 nob
            tempSql = "SELECT nob FROM t_similar WHERE noa = " + str(noa)
            cursor.execute(tempSql)
            data = cursor.fetchall()
            for t in data:
                if tempDict.get(t['nob']) == 0:
                    q.put(t['nob'])
                    tempDict[t['nob']] = 1

            count += 1
    connection.commit()
except Exception as e:
    connection.rollback()
    connection.commit()
    print("执行失败")
    print(e.__str__())
finally:
    cursor.close()

```

## 附录八

```
import pymysql.cursors

# 第五步 热度评估、设置热度前五的问题 id
# 相同 type 的问题认为是同一类型
# 热度规则: (相同 type 问题的最高热度) + 20 + 点赞数 * 2 - 反对数
# 如果 type 为 null, 热度为: 20 + 点赞数 * 2 - 反对数

connection = pymysql.connect(host='localhost', port=3306, user='test',
                             password='test', db='teddy', charset='utf8',
                             cursorclass=pymysql.cursors.DictCursor)

cursor = None
try:
    cursor = connection.cursor()
    # 非幂等, 所以每次执行前需要清零
    sql = "UPDATE t_forum SET heat = NULL"
    cursor.execute(sql)
    sql = "SELECT msgNo, stype, oppositionCount, approvalCount FROM t_forum"
    cursor.execute(sql)
    res = cursor.fetchall()
    for i in res:
        heat = 20
        whereSql = " WHERE "
        if i['stype'] is None:
            whereSql += "msgNo = " + str(i['msgNo'])
        else:
            tempSql = "SELECT MAX(heat) AS num FROM t_forum WHERE stype = "
            + str(i['stype'])
            whereSql += "stype = " + str(i['stype'])
            cursor.execute(tempSql)
            data = cursor.fetchone()
            if data['num'] is not None:
                heat += data['num']
            heat = heat + i['approvalCount'] * 2 - i['oppositionCount']
            tempSql = "UPDATE t_forum SET heat = " + str(heat)
            cursor.execute(tempSql + whereSql)
    # 事先已观察过, 排名前五的有 61 条数据
    sql = "SELECT * FROM t_forum ORDER BY heat DESC LIMIT 61"
    cursor.execute(sql)
```

```

res = cursor.fetchall()
questionId = 0
for i in range(len(res)):
    if i == 0 or res[i]['stype'] != res[i - 1]['stype']:
        questionId += 1
    tempSql = "UPDATE t_forum SET questionId = " + str(questionId) + "
WHERE msgNo = " + str(res[i]['msgNo'])
    cursor.execute(tempSql)
    connection.commit()
except Exception as e:
    connection.rollback()
    connection.commit()
    print("执行失败")
    print(e.__str__())
finally:
    cursor.close()

```

## 附录九

```

import xlrd
import datetime
import time
excel = xlrd.open_workbook('附件 4. xlsx')
sheet1 = excel.sheet_by_name('Sheet1')
k = 0
s = 0
l = 0
m = 0
for i in range(1, sheet1.nrows):
    l += len(sheet1.row_values(i)[5])/len(sheet1.row_values(i)[4])
    m += 1
    if sheet1.cell(i, 6).ctype == 1 and sheet1.cell(i, 3).ctype == 1:
        k += 1
        dateTime1 =
time.strptime(sheet1.row_values(i)[6], '%Y/%m/%d %H:%M:%S')
        dateTime2 =
time.strptime(sheet1.row_values(i)[3], '%Y/%m/%d %H:%M:%S')
        s += (int(time.mktime(dateTime1)*1000) -
int(time.mktime(dateTime2)*1000))

```

```
a = int(s/k)
print(a/(24*60*60*1000))
print(1/m)
```