# “智慧政务”中的文本挖掘应用

## 摘要

随着互联网的迅速普及和移动互联网的兴起，网民数量和网站数量都在急剧增长，网络的社会影响在日趋扩大。新闻网站、社交网站、微博网站都包含了大量的、动态更新的新闻信息，它们已经成为人们获取最新资讯的重要来源。面对海量的信息，单凭人工的力量很难进行全面的采集和整理。因此，如何通过计算机去自动采集、整理和分析海量的热点问题数据便具有重要的研究意义。

对于问题一，先用 Rstudio 将定义一级分类标签的标准进行划分，并将同一一级分类所对应的二级、三级分类作为分类词库，同时建立对应分词词库。再根据词库与留言主题和留言详情进行匹配，根据权重确定所处的标签体系，并在 Matlab 使用 F-Score 方法对上述分类方法进行评价。

对于问题二，提取留言主题中所带有的地区为标准进行划分。将划分后的留言中，将两两留言主题分词进行匹配。以匹配词数的高低和点赞数与反对数总和作为热点问题标准，并将热点问题的具体留言找出。

对于问题三，建立 0-10 分的评价体系，从相关性、完整性、可解释性、及时性这四个方面进行评分，每个方面占总评分的 25%，对答复意见的内容和留言详情的内容进行相关性分析、完整性分析和及时性分析得出最终的得分。

**关键词：**自然语言处理；文本挖掘；Matlab；Rstudio

# 一、问题重述

1. 群众留言分类

在处理网络问政平台的群众留言时，工作人员首先按照一定的划分体系对留言进行分类，以便后续将群众留言分派值相应的职能部门处理。目前大部分电子政务系统还是依靠人工根据经验处理，存在工作量大、效率低，且差错率高等问题。根据附件 2 给出数据，建立关于留言内容的一级标签分类模型，并使用 $F-\text{Score}$ 对分类方法进行评价。

（$F-\text{Score}$：$F_1=\frac{1}{n}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i+R_i}$，其中 $P_i$ 为第 $i$ 类的查准率，$R_i$ 为第 $i$ 类的查全率。）

2. 热点问题挖掘

某一时段内群众集中反映的某一问题可称为热点问题，如“XXX 小区多位业主多次反映入夏以来小区楼下烧烤店深夜经营导致噪音和油烟扰民”。及时发现热点问题，有助于相关部门进行有针对性地处理，提升服务效率。根据附件 3 将某一时段内反映特定地点或特定人群问题的留言进行归类，定义合理的热度评价指标，并给出评价结果，并保存为文件“热点问题表.xls”，同时给出相应热点问题对应的留言信息，并保存为“热点问题留言明细表.xls”。

3. 答复意见的评价

针对附件 4 相关部门对留言的答复意见，从答复的相关性、完整性、可解释性等角度对答复意见的质量给出一套评价方案，并尝试实现。

# 二、问题背景

近年来，随着微信、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、凝聚民气的重要渠道，各类社情民意相关的文本数据量不断攀升，给以往主要依靠人工来进行留言划分和热点整理的相关部分的工作带来了极大挑战。同时，随着大数据、云计算、人工智能等技术的发展，建立基于自然语言处理技术的智慧政务系统已经是社会治理创新发展的新趋势，对提升政府的管理水平和施政效率具有极大的推动作用。互联网与政务服务

图 1：互联网＋政务示意图

的结合越来越紧密，并不断创新政务信息公开，拓宽政务传播渠道。为公众提供简单、快捷、无技术与操作障碍的服务，实现了服务的普惠化。

## 三、数据说明

附件 1 中包含 517 条数据，每条数据对应有一级分类、二级分类、三级分类，其中一级分类中，有 15 种不同一级分类，其中每种一级分类所对应三级分类数据量如下表。

**表 1：附件 1 中一级分类对应三级分类数据量**

| 一级分类 | 数据量 | 一级分类 | 数据量 |
|---|---|---|---|
| 城乡建设 | 65 | 卫生计生 | 27 |
| 农村农业 | 56 | 商贸旅游 | 25 |
| 政法 | 55 | 环境保护 | 24 |
| 教育文体 | 45 | 交通运输 | 22 |
| 劳动和社会保障 | 42 | 纪检监察 | 21 |
| 民政 | 38 | 国土资源 | 19 |
| 党务政务 | 32 | 科技与信息产业 | 15 |
| 经济管理 | 31 | | |

附件 2 包含 9210 条数据，附件 3 有 4326 条数据，附件 4 有 2816 条数据。附件 2、附件 3、附件 4 每条数据都包含留言编号、留言用户、留言主题、留言时间、留言详情。其中附件 2 数据还包含一级标签，附件 3 包含反对数、点赞数，附件 4 包含答复意见、答复时间。

## 四、问题分析

对问题一：需要建立起一级标签与留言的对应关系。以附件 1 中三级分类为一级标签的分类表标准，建立分类标准，并根据分类标准建立对应分类词库与分词词库。以附件 2 中留言主题和留言详情为匹配目标，用上述词库对留言进行匹配，并将匹配结果与实际结果用 $F-S\text{core}$ 方法进行评价。

对问题二：将带有 A1-A9 的留言主题数据作为地区分类进行划分，将同一地区的所有留言主题进行分词，比对其中的动词与名词的相同数量以确定相关度，对比其中的相关度并进行匹配，以匹配词数的高低和点赞数与反对数总和作为热点问题标准，提取其中最高的五条数据。生成对应的热点问题表和热点问题留言

明细表，其中热度指数=分词相关度+关注数。

对问题三：需要建立一个 0-10 分的评价体系，以 0 分起始分，从留言答复的相关性、完整性、可解释性、及时性这四个方面进行加分，每个方面占总评分的 25%。将附件四中的答复意见的内容和留言详情的内容进行相关性分析、完整性分析和及时性分析，利用权重进行加分，最终得出评价分数。

# 五、数据假设

1.假定所有所给数据真实有效。

2.假定附件 1 中除其他项的三级分类涵盖附件 2 中所有一级标签对应的三级分类。

3.假定附件 2、附件 3、附件 4 中留言主题与与留言详情均存在对应关系。

# 六、数据预处理

## 6.1 数据预处理

自然语言处理与常规纯数字类型处理均需要处理空值、重复值、去除噪声项等。空值、重复值处理方式可等同于常规处理，噪声项中可对泛指而无实际意义的项进行处理。

### 6.1.1 “其他”分类

在自然语言识别中，出现“其他”需要识别时，一般会出现例如：其他等存在某等同行为、或是存在并列的其他项等，便于人工识别的前后文或是同级项。对于题目中附件 1 中二级分类、三级分类存在的其他分类而言，并没有存在能提高其他项识别度的同类项，或是已包含于其他同级分类中。此时，若使用计算机语言识别自然语言，编写计算机能识别其他分类的代码难度较高且识别正确率无同一判断标准，故其他项在此处为无效数据，需对该数据进行一定代替或舍弃该数据。现将附件 1 中三级分类为“其他”的数值进行处理。

在附件 1 中出现其他分类数据在二级分类与三级分类中，一级分类并不存在其他分类，且其他分类在出现的二级与三级分类中存在三种情况：

情况一：二级分类与三级分类同时为其他分类；

情况二：仅二级分类为其他分类；

情况三：仅三级分类为其他分类；

其中，情况一与情况三占多数，情况二有且仅有一条，于第 306 条：农村农

业-其他-农村“八大员问题”（为方便后续对附件 1 中一级二级三级分类同时说明，下文出现形式均以：一级分类-二级分类-三级分类形式表现）。

针对情况一，二级分类与三级分类均为其他分类，有效数据仅为一级分类，并通过对该情况提取可知，情况一均存在于所有一级分类中，故可舍弃该数据；针对情况二，有效数据为一级、三级分类，且三级分类不与其他三级分类重复，故保留该数据或是以三级分类数据代替二级分类；针对情况三，同情况二。

### 6.1.2 重复分类

使用 Rstudio 对所有源数据进行处理可发现不存在重复数据，但若对 5.1.1 中其他使用二级或三级分类代替，则会产生重复数据。对于附件 1 中重复数据，即重复的标签可以删除，对于后续操作出现的重复数据，不影响权重可删除。

## 七、问题一数据处理

### 7.1 附件 1 数据处理

附件 1 中一级、二级、三级分类逻辑层级如右图所示。由右图可知，若能确定留言问题分类的三级分类，则能确定所对应的一级分类。故可以将三级分类作为一级标签的标识符。

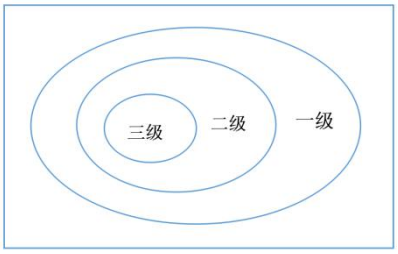


图 2：附件 1 逻辑层级图

### 7.1.1 分类词库标准

若要将三级分类作为唯一标识，则所有的三级分类需不重复，或重复但不影响唯一标识。在本题中，实际情况与之相反，如右图所示。即存在相同名称的三级分类但所对应的二级分类或一级分类可能不同。基于该实际情况，对三级分类的重复情况进行如下的分类标准。

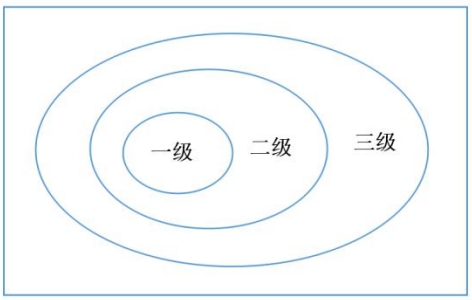


图 3：实际情况分类层级图

*基于三级分类重复情况的一级分类评判标准

（以下三级分类均在同一一级分类下）

a. 三级分类均不重复

b. 三级分类重复，一级分类也重复

c. 三级分类重复，一级分类不重复，二级分类也不重复

d. 三级分类重复，二级分类也重复，一级分类不重复

根据上述重复情况标准，建立词库标准。

*基于三级分类建立分类词库标准

1.对于仅存在 a、b 两种情况的分类词库，仅以三级分类为准，建立一级标签词库。

2.对于存在 c 情况但不存在 d 情况的分类词库，将对应二级分类添加到三级分类，建立一级标签词库。

3.对于存在 d 情况的分类词库，将一级、二级、三级分类均添加为一级标签词库。

根据上述词库标准，使用 Rstdio 软件对一级分类进行划分为 1、2、3 三种。如下表所示。

**表 2：词库标准划分表**

| 一级分类 | 词库分类 |
|---|---|
| 党务政务 | 1 |
| 环境保护 | 1 |
| 纪检监察 | 1 |
| 交通运输 | 1 |
| 科技与信息产业 | 1 |
| 民政 | 1 |
| 商贸旅游 | 1 |
| 卫生计生 | 1 |
| 政法 | 1 |
| 国土资源 | 2 |
| 农村农业 | 2 |
| 教育文体 | 2 |
| 劳动和社会保障 | 2 |
| 城乡建设 | 3 |
| 经济管理 | 3 |

### 7.1.2 建立分类词库

对一级分类为党务政务、环境保护、纪检监察等，以三级分类为准，将所有

三级分类作为词库，可得对应词库。对一级分类为国土资源、农村农业、教育文体等，将二级分类与三级分类作为词库，可得对应词库。对一级分类为城乡建设、经济管理，将一级、二级、三级分类作为词库，可得对应词库。对上述词库进行重复处理删除。

**表 3：党务政务分类词库**

| 三级分类 | |
|---|---|
| 制度建设 | 民族宗教 |
| 组织建设 | 工会工作 |
| 作风建设 | 共青团工作 |
| 党的建设 | 妇联工作 |
| 廉政建设 | 群众团体 |
| 港澳台侨 | 互联网信息监管 |
| 侨务 | 宣传舆论 |
| 港澳事务 | 精神文明建设 |
| 台湾事务 | 广播影视管理 |
| 外交外事 | 出版管理 |
| 国防外交 | 新闻管理 |
| 军事国防 | 政治体制 |
| 宗教事务 | 统战工作 |
| 民族政策 | 政治体制改革 |
| 民族事务 | 政协工作 |
| 民族宗教 | 人大工作 |

### 7.1.3 建立分词词库

考虑到词库中可能存在部分分词，但不存在源词可能，例如：三级分类为“安全隐患”，而留言问题中描述为“该地区安全存在隐患”，若仅以安全隐患进行匹配，则无法获取到正确结果，而使用分词匹配，即同时匹配安全与隐患，则可以获得正确结果。

使用 R 语言的中文分词包 jiebaR，将词库中的词进行分词并建立对应分词库（若在分词过程中，jiebaR 包中分词不足以囊括所需的词库，可以将对应的词库进行扩充，此处暂时不进行该操作，需对应熟悉人员进行）。

在分词过程中，重复 6.1.2 的处理，对“其他”分类和重复分类进行删除，同时为减少匹配多余的可能，将所有分词后字符长度为 1 的分词在匹配时不进行匹配。

**表 4：党务政务分词词库**

| 党务政务分词词库 | | | | | |
|---|---|---|---|---|---|
| 制度 | 建设 | 组织 | 作风 | 党的建设 | 廉政建设 |
| 港澳台侨 | 侨务 | 港澳 | 事务 | 台湾 | 外交 |
| 外事 | 国防 | 军事 | 宗教事务 | 民族 | 政策 |
| 民族事务 | 民族宗教 | 工会工作 | 共青团 | 工作 | 妇联 |
| 群众团体 | 互联网 | 信息 | 监管 | 精神文明 | 广播 |
| 影视 | 管理 | 出版 | 新闻 | 政治体制 | 统战工作 |
| 政治 | 体制改革 | 政协 | 人大 | 宣传 | 舆论 |

### 7.2 附件 2 数据处理

#### 7.2.1 问题一匹配机制

针对问题一对附件 2 中留言贴一级标签，即建立附件 1 中一级分类与附件 2 中留言主题或留言详情多对多的映射关系。其中附件 1 中一级分类所属定义域在 7.1 附件 1 数据处理中已完成，并且映射关系可看做简单的存在与否对附件 2 进行一级标签定义。根据已有的分类词库与分词词库可分别对附件 2 中留言详情与留言主题进行存在与否的匹配。

#### 7.2.2 权重问题

在匹配过程中，仅以存在与否进行匹配，在匹配结果中不以词库的大小做加权处理。若考虑权重则会出现以下情况。

情况 1：留言为：“学校 XXX 存在 XXX 问题，XXX 存在 XXX 问题。而安全隐患问题对于学校安全是重中之重，希望有关部门予以重视对待。”

对于情况 1，安全隐患是需要赋予的标签，所对应的一级标签为城乡规划，而过多的学校场景，包括教师等特定词汇出现时，可能会对结果产生偏差，从而导致原有匹配为安全隐患变为教育文体。

情况 2：假定一留言同时匹配 2 次城乡规划和教育文体。

若进行加权处理，由表一：附件 1 中一级分类对应三级分类数据量可得，城乡规划对应的数据量为 65，而教育文体为 45，加权后，一级标签必定属于教育文体，而实际情况可能为城乡规划。

针对可能出现的情况 1、2，对匹配过程与结果不进行加权处理，对应的若出现多重标签的可能时，在计算结果时将所得结果乘上标签数量的倒数。

### 7.2.3 匹配结果

以 7.2.1 中匹配机制为准，进行对应的匹配过后，生成留言主题-分类词库、留言主题-分词词库、留言详情-分类词库、留言详情-分词词库四个匹配结果。对于该匹配结果由 7.2.2 可知，不能使用权重计算，故将对应的匹配结果中选取各自最大值，并将出现次数最多的最大值作为最终的匹配结果，若同一匹配结果中出现最大值相同或不同匹配结果相同次数的最大值，则保留结果，并在计算评价指标时乘上标签数量的倒数。

| V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 | V11 | V12 | V13 | V14 | V15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 14 | 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 15 | 15 | 14 | 0 | 14 | 14 | 0 | 0 | 14 | 14 | 14 | 8 | 0 | 0 | 14 |
| 14 | 14 | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 11 | 0 | 10 | 0 |
| 14 | 15 | 12 | 14 | 14 | 14 | 4 | 14 | 14 | 14 | 14 | 12 | 14 | 11 | 14 |
| 15 | 0 | 14 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 13 | 0 | 14 | 0 |
| 14 | 4 | 14 | 0 | 0 | 0 | 9 | 0 | 0 | 0 | 0 | 14 | 0 | 0 | 0 |
| 0 | 14 | 0 | 0 | 0 | 0 | 14 | 0 | 0 | 0 | 0 | 15 | 0 | 0 | 0 |
| 0 | 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 14 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 14 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

图 4：部分匹配结果

### 7.2.4 $F-Score$

使用 $F-Score$ 方法对上述匹配结果进行计算。具体公式：$F_1=\frac{1}{n}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i+R_i}$，

其中 $P_i$ 为第 $i$ 类的查准率，$R_i$ 为第 $i$ 类的查全率。

查准率=（检索出的相关信息量/检索出的信息总量）x100%，

查全率=（检索出的相关信息量/系统中的相关信息总量）x100%

根据上述方法，使用 MATLAB 软件，对上述匹配结果进行评价，结果为 0.0278。

## 八、问题二数据处理

### 8.1 附件 3 预处理

对于问题二需要提取附件 3 中热点问题，即对附件 3 中，综合留言主题或留言详情中同一地区出现重复问题，以及反对数与点赞数最高的数据。

使用 R 语言将所有数据导入，可知有 4326 条数据，其中留言编号为 189856 的留言主题仅为“）”，数据出现缺失且无法处理，故删除。

在剩余的 4325 条数据中，对于留言主题，使用 R 语言对所有留言主题进行

分词处理可得“附件 3-留言详情-分词.csv”中 45343 条数据。观察数据可知，留言详情所有的字符数一般情况原大于留言主题（约在 10 倍以上），留言详情分词数必定远大于留言详情数，时间效率上不推荐直接使用留言详情，这里仅对前 3000 条留言详情数据给出分词结果，有 600947 条，数据分词时间超过 1 个小时。

由常理可知，留言主题一般情况可概括留言详情中所发生的具体细节，故以留言主题作为热点问题提取的一个标准指标。同时，对一个问题的反对数与点赞数可以看成对该问题的关注数。综上，将留言主题与关注数作为热点问题的反映指标。

### 8.2 地区分类处理

由于题目未给出所有地区的分类，无法快捷有效提取所有留言中地区，在分词表中 name 为 eng 的值中，包括 A1-A9 地区等可快捷分辨地区数据，故暂以带有 A1-A9 的留言主题数据作为地区分类。分类后数据有 2277 条数据，其中数据条数如下表所示。

**表 5：所属地区留言数据量**

| 所属地区 | 数据量 | 所属地区 | 数据量 | 所属地区 | 数据量 |
|---|---|---|---|---|---|
| A1 | 205 | A2 | 264 | A3 | 434 |
| A4 | 235 | A5 | 176 | A6 | 144 |
| A7 | 682 | A8 | 84 | A9 | 53 |

### 8.3 关注数

对于反对数与点赞数，可归类为对该留言的关注数，其中关注数=反对数+点赞数+1（发表留言的人算一个关注）。通过对关注数进行初步分析可知关注数最大值为 2098，最小值为 1，中间波动过大，为此将关注进行取对数计算。使用 Rstudio 取对数后，数据在$[0,\ln 2098]$中，波动值较小。

### 8.4 相关问题

对于留言主题需要提取问题比较复杂，但可以通过对比留言主题有效分词之间的相关性。表述同一件事情时，不同人可能出现表述不同，但对于关键词，例如关键性地点，有关人事物等名词和动词的表述差异并不会出现过多差异。故将同一地区的所有留言主题进行分词，并比对其中的动词与名词的相同数量以确定

相关度，如果不存在任何相关度，则可以认为不相关，如果存在相关度，则对比其中的相关度，并提取其中最高的五条数据。

由于在 8.2 中已做地区分类处理，在此以留言主题中存在 A1 的分词作为作为热点问题提取的范围。其中前五条数据是留言编号为 204464、207199、277127、272187、252074。

### 8.5 热点问题

由 8.4 可知，热点问题所在留言编号，其对应留言部分信息如下：

**表 6：热点问题部分信息**

| 留言编号 | 地点/人群 | 问题描述 |
| --- | --- | --- |
| 204464 | A5 区东塘街道新 A1 区之都佳苑小区 | 小区违建问题 |
| 207199 | A1 区马王堆沁园小区 b 区 | 公共停车位被个人占用 |
| 277127 | A1 区马王堆陶瓷城 B 区 | 车位被占用严重,消防通道被堵 |
| 272187 | A1 区工商银行靠近 A1 区广场 1 号口 | 半夜施工、噪音大 |
| 252074 | A4 区 A1 区中路名富公寓 | 物业乱收费 |

由此可以生成对应的热点问题表和热点问题留言明细表，其中热度指数=分词相关度+关注数。

## 九、问题三数据处理

### 9.1 问题三评价方案

针对所回答问题可以从相关性、完整性、可解释性、及时性四方面来对答复意见进行评分。

及时性可以将答复时间减去留言时间的天数作为标准计算；相关性可以用留言主题与答复意见的相关性进行计算，该相关性可以使用分词相关计算；完整性可以从对答复意见中的回答问题是否完整来计算；可解释性可从答复意见是否对解决问题有效来进行判断。

有的答复意见是“已转交给相关部门”，关于这类答复意见直接给予 6 分的评价分数，对于剩下的数据进行相关性分析、完整性分析和及时性分析。

以 0 分为起始分数，依次进行相关性分析、完整性分析和及时性分析，评价分数进行依次变化，形成最终的评价分数。

### 9.2 相关性分析

相关性分析包括相关性和可解释性这两方面，对附件四的留言详情进行分词处理，将分词处理的结果与答复意见进行匹配得出相关性指数。

若相关性指数在 0%-20%之间，则评价分数加 1 分；（不包括 0%）

若相关性指数在 20%-40%之间，则评价分数加 2 分；

若相关性指数在 40%-60%之间，则评价分数加 3 分；

若相关性指数在 60%-80%之间，则评价分数加 4 分；

若相关性指数在 80%-100%之间，则评价分数加 5 分。

### 9.3 完整性分析

完整性是指问题详情中问题的个数和答复意见中所回答的有关问题的个数进行对比。

若所回答的问题个数为 0，则评价分数不加分；

若所回答的问题个数少于应回答的问题个数，则评价分数加 1.5 分；

若所回答的问题个数大于或等于应回答的问题个数，则评价分数加 2.5 分。

### 9.4 及时性分析

及时性是指答复时间与留言时间之间的差距，将答复时间和留言时间都转换为以天为单位（以 2011 年 1 月 1 日为起始日期）。计算相差时间，相差时间=答复时间-留言时间，取相差时间的平均数作为衡量数值。

若相差时间低于平均数，则评价分数加 2.5 分；

若相差时间等于平均数，则评价分数加 1.5 分；

若相差时间高于平均数，则评价分数不加分。

## 十、结果分析

对于问题一，优点：

1.该建模方便理解，并符合自然语言识别方法；

2.使用代码运行可节省很多人工工作量。

缺点：

1.所建立词库不够完善。

附件 1 中一级分类与附件二中留言主题与留言详情映射关系应为类单纯存在与否的关系。在自然语言分类中，关键词与原数据之间关系与聚点关系相同，聚点本身必须存在于大多源数据之间，但聚点本身可不存在于源数据之间。例如：垃圾堆放问题属于环境保护问题，但环境保护的标签中并不存在有垃圾堆放的分类，对应的附件 1 中为卫生计生-公共卫生-其他。

2.匹配时可对词库匹配进行优先级选择。

通过对具体的附件 2 数据观察可知，并非所有附件 1 中一级分类都被使用到，而此时多余的匹配库可能对匹配结果造成影响。

改进：

1. 针对缺点 1 可以从附件 1 中其他分类更具体，或是用自然语言识别对附件 2 中提取关键词与附件 1 中分类进行比对，亦或是对相应的一级分类直接建立词库。

2. 针对缺点 2 可以对需要匹配的数据进行预匹配处理，将可能出现在对应的库范围缩小，从而提高匹配精度。

对于问题二：优点：

1.以地区作为划分，能更好具体到问题地区。

2.分词处理对问题匹配速度上有较好提升。

缺点：

1. 地区划分表不够完善。

2. 分词可以做更近一步优化。

改进：

1.地区划分表可以找专业人士，或是根据数据部分人工操作，达到更好的划分效果。

2.分词优化上可以对分词词库进行升级，保证分词出来的结果更贴合实际问题的关键词。

## 十一、参考文献

[1] 刘星星,何婷婷,龚海军,等. 网络热点事件发现系统的设计[J]. 中文信息学报, 2008,(6)： 80-85.

[2] 张启宇,朱玲,张雅萍. 中文分词算法研究综述[J]. 情报探索, 2008,000(011): 53-56.

[3] 郑天奇. 基于关键词分析的微博群体阅读特征分析[J]. 图书情报研究， 2018,39(2)：

60-68.
[4] 侯亚君. R 语言在数据挖掘中的运用[J]. 晋城职业技术学院学报, 2014,7(2): 63-65.
[5] 易剑波. 基于文本挖掘的电商用户评价分析与系统实现[D]. 江苏： 东南大学, 2017.
[6] 毛宇. 中医药症状的中文分词与句子相似度研究[D]. 浙江: 浙江大学,2017.

# 附录

一、Rstudio 代码

1.词库标准划分表

```
#清空变量
rm(list=ls())
#设置工作路径
setwd("D:\\2020 泰迪杯\\C 题示例数据")
#显示文件，判断文件读取方式
list.files()
#加载读取所需要的包
# install.packages("openxlsx")
library(openxlsx)
#读取附件 1
fj1<-read.xlsx("D:\\2020 泰迪杯\\源数据\\C 题全部数据\\附件 1.xlsx",sheet=1)
#清除空数据
fj1<-na.omit(fj1)
#数据格式化
sufj1<-fj1
class(sufj1)
str(sufj1)
#（若非 character 型则转化为 character 型）
# sufj1$一级分类<-as.character(sufj1$一级分类)
# sufj1$二级分类<-as.character(sufj1$二级分类)
# sufj1$三级分类<-as.character(sufj1$三级分类)
#分类为其他的数据，计算机识别难度较高且识别时间可能过长舍弃该数据
#删除二级分类、三级分类数值均为其他
for(i in 1:nrow(sufj1)){
  if(sufj1[i,2]=='其他' & sufj1[i,3]=='其他'){
    sufj1<-sufj1[-i,]
  }
}
#三级分类为其他时，将二级分类值赋值给三级分类
for(i in 1:nrow(sufj1)){
  if(sufj1[i,3]=='其他'){
    sufj1[i,3]<-sufj1[i,2]
  }
}
#删除因此生成的完全重复行
# install.packages("dplyr")
library(dplyr)
sufj1<-sufj1%>%distinct()
#标记三级分类重复但一级分类不重复的行
for(i in 1:nrow(sufj1)){
```

```
  a1<-sufj1[i,1]
  a3<-sufj1[i,3]
  for(j in (i+1):nrow(sufj1)){
    b1<-sufj1[j,1]
    b3<-sufj1[j,3]
    if(a3==b3 & a1!=b1){
      sufj1[i,4]<-0
      sufj1[j,4]<-0
    }
  }
}
#提取标记的行
fj1_dsflcf<-na.omit(sufj1[sufj1$V4==0,])
fj1_dsflcf<-fj1_dsflcf[,-4]
#生成三级分类重复一级分类不重复行的表格
# write.csv(fj1_dsflcf,"附件一中三级分类重复一级分类不重复.csv",row.names = F)
#标记二级、三级分类仍存在重复同时一级分类不重复的行，若二级分类不同，则将二级分类做为前缀加在三级分类中
sufj1<-sufj1[,-4]
for(i in 1:nrow(sufj1)){
  a1<-sufj1[i,1]
  a2<-sufj1[i,2]
  a3<-sufj1[i,3]
  for(j in (i+1):nrow(sufj1)){
    b1<-sufj1[j,1]
    b2<-sufj1[j,2]
    b3<-sufj1[j,3]
    if(a3==b3 & a2==b2 & a1!=b1){
      sufj1[i,4]<-0
      sufj1[j,4]<-0
    }
    if(a3==b3 & a2!=b2 & a1!=b1){
      sufj1[i,4]<-paste0(sufj1[i,2],sufj1[i,3])
      sufj1[j,4]<-paste0(sufj1[j,2],sufj1[j,3])
    }
  }
}
#提取标记的行
fj1_dsdefl_cf<-na.omit(sufj1[sufj1$V4==0,])
fj1_dsdefl_cf<-fj1_dsdefl_cf[,-4]
fj1_dsflcfdeflbcf<-na.omit(sufj1[sufj1$V4!=0,])
fj1_dsflcfdeflbcf<-fj1_dsflcfdeflbcf[,-4]
#生成二级、三级分类重复一级分类不重复的表格
# write.csv(fj1_dsdefl_cf,"附件一中二级、三级分类重复一级分类不重复.csv",row.names = F)
#生成三级分类重复一级、二级分类不重复的表格
```

```
# write.csv(fj1_dsflcfdeflbcf,"附件一中三级分类重复一级、二级分类不重复.csv",row.names = F)
sufj1<-sufj1[,-4]
#根据建立词库标准，对一级分类进行标记
D<-as.data.frame(fj1_dsdefl_cf$一级分类)
D<-D %>% distinct()
D$`fj1_dsdefl_cf$一级分类`<-as.character(D$`fj1_dsdefl_cf$一级分类`)
C<-as.data.frame(fj1_dsflcfdeflbcf$一级分类)
C<-C%>%distinct()
str(C)
C$`fj1_dsflcfdeflbcf$一级分类`<-as.character(C$`fj1_dsflcfdeflbcf$一级分类`)
for(i in 1:nrow(C)){
  C[i,2]<-1
}
for(i in 1:nrow(C)){
  a<-C[i,1]
  for(j in 1:nrow(D)){
    b<-D[j,1]
    if(a==b){
      C[i,2]<-0
    }
  }
}
C<-C[C$V2==1,]
C<-as.data.frame(C[,-2])
AB<-as.data.frame(sufj1$一级分类)
AB<-AB%>%distinct()
str(AB)
AB$`sufj1$一级分类`<-as.character(AB$`sufj1$一级分类`)
for(i in 1:nrow(AB)){
  AB[i,2]<-1
}
for(i in 1:nrow(AB)){
  a<-AB[i,1]
  for(j in 1:nrow(C)){
    b<-C[j,1]
    if(a==b){
      AB[i,2]<-0
    }
  }
  for(j in 1:nrow(D)){
    b<-D[j,1]
    if(a==b){
      AB[i,2]<-0
    }
```

```
  }
}
AB<-AB[AB$V2==1,]
AB<-as.data.frame(AB[,-2])
#词库标准划分表
ckbzhfb<-data.frame()
str(AB)
str(C)
str(D)
AB$`AB[, -2]`<-as.character(AB$`AB[, -2]`)
C$`C[, -2]`<-as.character(C$`C[, -2]`)
D$`fj1_dsdefl_cf$一级分类`<-as.character(D$`fj1_dsdefl_cf$一级分类`)
k<-1
for(i in 1:nrow(AB)){
  ckbzhfb[k,1]<-AB[i,1]
  ckbzhfb[k,2]<-1
  k<-k+1
}
for(i in 1:nrow(C)){
  ckbzhfb[k,1]<-C[i,1]
  ckbzhfb[k,2]<-2
  k<-k+1
}
for(i in 1:nrow(D)){
  ckbzhfb[k,1]<-D[i,1]
  ckbzhfb[k,2]<-3
  k<-k+1
}
colnames(ckbzhfb)[1]<-"一级分类"
colnames(ckbzhfb)[2]<-"词库分类"
write.csv(ckbzhfb,"词库标准划分表.csv",row.names = F)
2.分类词库、分词词库
#词库
dir.create("问题 1")
setwd("D:\\2020 泰迪杯\\运行\\问题 1")
dir.create("分类词库")
dir.create("分词词库")
setwd("D:\\2020 泰迪杯\\运行\\问题 1\\词库")
# install.packages("jiebaR")
library(jiebaRD)
library(jiebaR)
wk = worker()
#建立对应词库
#AB
```

```
for(i in 1:nrow(AB)){
  #取分类为 AB 的建立源词库
  a<-AB[i,1]
  fj1_dwzw<-sufj1[sufj1$一级分类==a,]
  fj1_dwzw<-as.data.frame(fj1_dwzw$三级分类)
  #删除原数据词中的重复
  fj1_dwzw<-fj1_dwzw%>%distinct()
  colnames(fj1_dwzw)[1]<-"三级分类"
  fj1_dwzw$三级分类<-as.character(fj1_dwzw$三级分类)
  #考虑到源数据词可能不需要全覆盖，即可能出现部分源数据词，在此对源数据词进行分词
  fj1_dwzw_fc<-data.frame()
  k<-1
  for(m in 1:nrow(fj1_dwzw)){
    b<-wk[fj1_dwzw[m,1]]
    for(m in 1:length(b)){
      fj1_dwzw_fc[k,1]<-b[m]
      k<-k+1
    }
  }
  #删除分词中的重复
  fj1_dwzw_fc<-fj1_dwzw_fc%>%distinct()
  colnames(fj1_dwzw_fc)[1]<-"分词"
  a<-AB[i,1]
  #输出对应的词库
  setwd("D:\\2020 泰迪杯\\运行\\问题 1\\分类词库")
  write.csv(fj1_dwzw,paste0(a,".csv"),row.names = F)
  setwd("D:\\2020 泰迪杯\\运行\\问题 1\\分词词库")
  write.csv(fj1_dwzw_fc,paste0(a,"_分词.csv"),row.names = F)
}
#C
for(i in 1:nrow(C)){
  #取分类为 C 的建立源词库
  a<-C[i,1]
  fj1_dwzw<-sufj1[sufj1$一级分类==a,]
  fj1_dwzw1<-data.frame()
  for(m in 1:nrow(fj1_dwzw)){
    fj1_dwzw1[m,1]<-fj1_dwzw[m,2]
  }
  for(m in 1:nrow(fj1_dwzw)){
    fj1_dwzw1[m+nrow(fj1_dwzw),1]<-fj1_dwzw[m,3]
  }
  fj1_dwzw<-fj1_dwzw1
  #删除原数据词中的重复
  fj1_dwzw<-fj1_dwzw%>%distinct()
```

```
  colnames(fj1_dwzw)[1]<-"三级分类"
  fj1_dwzw$三级分类<-as.character(fj1_dwzw$三级分类)
  #考虑到源数据词可能不需要全覆盖，即可能出现部分源数据词，在此对源数据词进行分词
  fj1_dwzw_fc<-data.frame()
  k<-1
  for(m in 1:nrow(fj1_dwzw)){
    b<-wk[fj1_dwzw[m,1]]
    for(m in 1:length(b)){
      fj1_dwzw_fc[k,1]<-b[m]
      k<-k+1
    }
  }
  #删除分词中的重复
  fj1_dwzw_fc<-fj1_dwzw_fc%>%distinct()
  colnames(fj1_dwzw_fc)[1]<-"分词"
  a<-C[i,1]
  #输出对应的词库
  setwd("D:\\2020 泰迪杯\\运行\\问题 1\\分类词库")
  write.csv(fj1_dwzw,paste0(a,".csv"),row.names = F)
  setwd("D:\\2020 泰迪杯\\运行\\问题 1\\分词词库")
  write.csv(fj1_dwzw_fc,paste0(a,"_分词.csv"),row.names = F)
}
#D
for(i in 1:nrow(D)){
  #取分类为 D 的建立源词库
  a<-D[i,1]
  fj1_dwzw<-sufj1[sufj1$一级分类==a,]
  fj1_dwzw1<-data.frame()
  for(m in 1:nrow(fj1_dwzw)){
    fj1_dwzw1[m,1]<-fj1_dwzw[m,1]
  }
  for(m in 1:nrow(fj1_dwzw)){
    fj1_dwzw1[m+nrow(fj1_dwzw),1]<-fj1_dwzw[m,2]
  }
  for(m in 1:nrow(fj1_dwzw)){
    fj1_dwzw1[m+2*nrow(fj1_dwzw),1]<-fj1_dwzw[m,3]
  }
  fj1_dwzw<-fj1_dwzw1
  #删除原数据词中的重复
  fj1_dwzw<-fj1_dwzw%>%distinct()
  colnames(fj1_dwzw)[1]<-"三级分类"
  fj1_dwzw$三级分类<-as.character(fj1_dwzw$三级分类)
  #考虑到源数据词可能不需要全覆盖，即可能出现部分源数据词，在此对源数据词进行分词
  fj1_dwzw_fc<-data.frame()
```

```
  k<-1
  for(m in 1:nrow(fj1_dwzw)){
    b<-wk[fj1_dwzw[m,1]]
    for(m in 1:length(b)){
      fj1_dwzw_fc[k,1]<-b[m]
      k<-k+1
    }
  }
  #删除分词中的重复
  fj1_dwzw_fc<-fj1_dwzw_fc%>%distinct()
  colnames(fj1_dwzw_fc)[1]<-"分词"
  a<-D[i,1]
  #输出对应的词库
  setwd("D:\\2020 泰迪杯\\运行\\问题 1\\分类词库")
  write.csv(fj1_dwzw,paste0(a,".csv"),row.names = F)
  setwd("D:\\2020 泰迪杯\\运行\\问题 1\\分词词库")
  write.csv(fj1_dwzw_fc,paste0(a,"_分词.csv"),row.names = F)
}
```

3.匹配

```
#匹配
setwd("D:\\2020 泰迪杯\\C 题全部数据")
fj2<-read.xlsx("附件 2.xlsx",sheet=1)
#加载匹配所需库
# install.packages("tidyverse")
library(tidyverse)
ppztfljg<-data.frame(ckbzhfb$一级分类)
ppztfcjg<-data.frame(ckbzhfb$一级分类)
ppxqfljg<-data.frame(ckbzhfb$一级分类)
ppxqfcjg<-data.frame(ckbzhfb$一级分类)
sufj2<-fj2
for(i in 1:nrow(sufj2)){
  fj2_lyzt<-sufj2[i,3]
  fj2_lyxq<-sufj2[i,5]
  for(hang in 1:nrow(ckbzhfb)){
    fljg<-0
    fcjg<-0
    xqfljg<-0
    xqfcjg<-0

    ppfl<-read.csv(paste0("D:\\2020 泰迪杯\\运行\\问题 1\\分类词库\\",ckbzhfb[hang,1],".csv"))
    ppfc<-read.csv(paste0("D:\\2020 泰迪杯\\运行\\问题 1\\分词词库\\",ckbzhfb[hang,1],"_分词.csv"))
    ppfl$三级分类<-as.character(ppfl$三级分类)
    ppfc$分词<-as.character(ppfc$分词)
    for(j in 1:nrow(ppfc)){
```

```
      a<-ppfc[j,1]
      ppfc[j,2]<-nchar(a)
    }
    for(j in 1:nrow(ppfl)){
      a<-ppfl[j,1]
      flag<-grepl(a,fj2_lyzt)
      if(flag){
        fljg<-fljg+1
      }
    }
    ppztfljg[hang,i+1]<-fljg
    for(j in 1:nrow(ppfc)){
      a<-ppfc[j,1]
      flag<-grepl(a,fj2_lyzt)
      if(flag & ppfc[j,2]!=1){
        fcjg<-fcjg+1
      }
    }
    ppztfcjg[hang,i+1]<-fcjg
    for(j in 1:nrow(ppfl)){
      a<-ppfl[j,1]
      flag<-grepl(a,fj2_lyxq)
      if(flag){
        xqfljg<-xqfljg+1
      }
    }
    ppxqfljg[hang,i+1]<-xqfljg
    for(j in 1:nrow(ppfc)){
      a<-ppfc[j,1]
      flag<-grepl(a,fj2_lyxq)
      if(flag & ppfc[j,2]!=1){
        xqfcjg<-xqfcjg+1
      }
    }
    ppxqfcjg[hang,i+1]<-xqfcjg
  }
}
#将对应结果输出为 csv
setwd("D:\\2020 泰迪杯\\运行\\问题 1")
for(i in 1:15){
  row.names(ppxqfcjg)[i]<-as.character(ppxqfcjg[i,1])
}
ppxqfcjg<-ppxqfcjg[,-1]
```

```
for(i in 1:15){
  row.names(ppxqfljg)[i]<-as.character(ppxqfljg[i,1])
}
ppxqfljg<-ppxqfljg[,-1]
for(i in 1:15){
  row.names(ppztfcjg)[i]<-as.character(ppztfcjg[i,1])
}
ppztfcjg<-ppztfcjg[,-1]
for(i in 1:15){
  row.names(ppztfljg)[i]<-as.character(ppztfljg[i,1])
}
ppztfljg<-ppztfljg[,-1]
write.csv(ppxqfcjg,"匹配详情分词结果.csv",row.names = F)
write.csv(ppxqfljg,"匹配详情分类结果.csv",row.names = F)
write.csv(ppztfcjg,"匹配主题分词结果.csv",row.names = F)
write.csv(ppztfljg,"匹配主题分类结果.csv",row.names = F)
```

4.匹配结果提取

```
ztfljg<-data.frame()
ztfcjg<-data.frame()
xqfljg<-data.frame()
xqfcjg<-data.frame()
for(i in 1:ncol(ppztfljg)){
  a<-max(ppztfljg[,i])
  if(a==0){
    ztfljg[1,i]<-0
  }else{
    k<-1
    for(j in 1:nrow(ppztfljg)){
      b<-ppztfljg[j,i]
      if(b==a){
        ztfljg[k,i]<-j
        k<-k+1
      }
    }
  }
}
for(i in 1:ncol(ppztfcjg)){
  a<-max(ppztfcjg[,i])
  if(a==0){
    ztfcjg[1,i]<-0
  }else{
    k<-1
    for(j in 1:nrow(ppztfcjg)){
      b<-ppztfcjg[j,i]
```

```
      if(b==a){
        ztfcjg[k,i]<-j
        k<-k+1
      }
    }
  }
}
for(i in 1:ncol(ppxqfljg)){
  a<-max(ppxqfljg[,i])
  if(a==0){
    xqfljg[1,i]<-0
  }else{
    k<-1
    for(j in 1:nrow(ppxqfljg)){
      b<-ppxqfljg[j,i]
      if(b==a){
        xqfljg[k,i]<-j
        k<-k+1
      }
    }
  }
}
for(i in 1:ncol(ppxqfcjg)){
  a<-max(ppxqfcjg[,i])
  if(a==0){
    xqfcjg[1,i]<-0
  }else{
    k<-1
    for(j in 1:nrow(ppxqfcjg)){
      b<-ppxqfcjg[j,i]
      if(b==a){
        xqfcjg[k,i]<-j
        k<-k+1
      }
    }
  }
}
jg<-data.frame()
for(i in 1:ncol(ztfljg)){
  k<-1
  for(j in 1:nrow(ztfljg)){
    if(!is.na(ztfljg[j,i])){
      jg[k,i]<-ztfljg[j,i]
      k<-k+1
```

```
    }
  }
  for(j in 1:nrow(ztfcjg)){
    if(!is.na(ztfcjg[j,i])){
      jg[k,i]<-ztfcjg[j,i]
      k<-k+1
    }
  }
  for(j in 1:nrow(xqfljg)){
    if(!is.na(xqfljg[j,i])){
      jg[k,i]<-xqfljg[j,i]
      k<-k+1
    }
  }
  for(j in 1:nrow(xqfcjg)){
    if(!is.na(xqfcjg[j,i])){
      jg[k,i]<-xqfcjg[j,i]
      k<-k+1
    }
  }
}
for(i in 1:nrow(jg)){
  for (j in 1:ncol(jg)) {
    if(is.na(jg[i,j])){
      jg[i,j]<-0
    }
  }
}
write.csv(jg,"匹配结果.csv",row.names = F)
# jg<-read.csv("jg.csv")
#手动输入一个与标准词库划分表相同的序号表
a<-read.xlsx("a.xlsx")
# str(sufj2)
# str(a)
# str(jg)
zjg<-data.frame()
for(i in 1:nrow(sufj2)){
  for(j in 1:nrow(a)){
    if(sufj2[i,6]==a[j,1]){
      zjg[20,i]<-a[j,2]
    }
  }
}
for(i in 1:ncol(jg)){
```

```
  a<-as.data.frame(table(jg[,i]))
  for (j in 1:nrow(a)) {
    if(a[j,1]==0){
      a[j,2]<-0
    }
  }
  b<-max(a$Freq)
  k<-1
  for (j in 1:nrow(a)) {
    if(a[j,2]==b){
      zjg[k,i]<-a[j,1]
      k<-k+1
    }
  }
}
for(i in 1:nrow(zjg)){
  for (j in 1:ncol(zjg)) {
    if(is.na(zjg[i,j])){
      zjg[i,j]<-0
    }
  }
}
# zjg<-read.csv("zjg.csv")
zjglong<-data.frame()
for(i in 1:ncol(zjg)){
  k<-0
  for(j in 1:nrow(zjg)){
    if(zjg[j,i]!=0){
      k<-k+1
    }
  }
  zjglong[1,i]<-k
}
for(i in 1:ncol(zjg)){
  zjg[21,i]<-zjglong[1,i]
}
write.csv(zjg,"总结果.csv",row.names=FALSE)
```

4.问题二代码

```
rm(list = ls())
dir.create("D:\\2020 泰迪杯\\运行\\问题 2")
setwd("D:\\2020 泰迪杯\\运行\\问题 2")
library(openxlsx)
fj3<-read.xlsx("D:\\2020 泰迪杯\\源数据\\C 题全部数据\\附件 3.xlsx",sheet = 1)
fb_fj3<-fj3
```

```
fb_fj3<-fb_fj3[fb_fj3$留言编号!="189856",]
library(jiebaR)
library(jiebaRD)
library(tidyverse)
library(pacman)
wk<-worker()
tag_worker<-worker(type = "tag")
#总数据测试代码，时长过长舍弃
# fj3_lyzt_fc<-data.frame()
# k<-1
# for(i  in 1:nrow(fb_fj3)){
#   cn<-fb_fj3[i,3]
#   tag_result<-tagging(cn,tag_worker) %>% enframe()
#   if(nrow(tag_result)!=0){
#     for (j in 1:nrow(tag_result)) {
#       fj3_lyzt_fc[k,1]<-tag_result[j,1]
#       fj3_lyzt_fc[k,2]<-tag_result[j,2]
#       k<-k+1
#     }
#   }
# }
#
# fj3_lyxq_fc<-data.frame()
# k<-1
# for(i  in 1:nrow(fb_fj3)){
#   cn<-fb_fj3[i,5]
#   tag_result<-tagging(cn,tag_worker) %>% enframe()
#   if(nrow(tag_result)!=0){
#     for (j in 1:nrow(tag_result)) {
#       fj3_lyxq_fc[k,1]<-tag_result[j,1]
#       fj3_lyxq_fc[k,2]<-tag_result[j,2]
#       k<-k+1
#     }
#   }
# }
# write.csv(fj3_lyzt_fc,"附件 3-留言主题-分词_改.csv",row.names=FALSE)
# write.csv(fj3_lyxq_fc,"附件 3-留言详情-分词.csv",row.names=FALSE)
#
# fj3_lyzt_fc<-data.frame()
# n<-1
# for(i  in 1:nrow(fb_fj3)){
#   cn<-fb_fj3[i,3]
#   tag_result<-tagging(cn,tag_worker) %>% enframe()
#   k<-1
```

```
#   if(nrow(tag_result)!=0){
#     for (j in 1:nrow(tag_result)) {
#       fj3_lyzt_fc[k,n]<-tag_result[j,1]
#       fj3_lyzt_fc[k,n+1]<-tag_result[j,2]
#       colnames(fj3_lyzt_fc)[n]<-paste0(fb_fj3[i,1],"name")
#       colnames(fj3_lyzt_fc)[n+1]<-paste0(fb_fj3[i,1],"value")
#       k<-k+1
#     }
#     n<-n+2
#   }
# }
#
# for(i in 1:nrow(fj3_lyzt_fc)){
#   for(j in 1:ncol(fj3_lyzt_fc)){
#     if(is.na(fj3_lyzt_fc[i,j])){
#       fj3_lyzt_fc[i,j]<-0
#     }
#   }
# }
#
# fj3_lyzt_fc<-read.csv("附件 3-留言主题-分词_改.csv")
# ppjg<-data.frame()
# # fb_fj3<-fb_fj3[fb_fj3$留言编号!="189856",]
# for(i in 1:4325){
#   pptmp1<-data.frame(as.character(fj3_lyzt_fc[,paste0("X",fb_fj3[i,1],"name")]),
#                      as.character(fj3_lyzt_fc[,paste0("X",fb_fj3[i,1],"value")]))
#
#   for(j in 1:4325){
#     pptmp2<-data.frame(as.character(fj3_lyzt_fc[,paste0("X",fb_fj3[j,1],"name")]),
#                        as.character(fj3_lyzt_fc[,paste0("X",fb_fj3[j,1],"value")]))
#     flag<-0
#     flag1<-0
#     for(n in 1:nrow(pptmp1)){
#       a1<-pptmp1[n,1]
#       a2<-pptmp1[n,2]
#       if(a2!="0"){
#         flag1<-flag1+1
#         for(m in 1:nrow(pptmp2)){
#           b1<-pptmp2[m,1]
#           b2<-pptmp2[m,2]
#           if(a1==b1 & a2==b2){
#             flag<-flag+1
#             break
#           }
```

```
#         }
#       }else
#         break
#     }
#     if(flag1!=0){
#       ppjg[i,j]<-flag/flag1
#     }
#     if(i==1){
#       colnames(ppjg)[j]<-fb_fj3[j,1]
#     }
#   }
#   rownames(ppjg)[i]<-fb_fj3[i,1]
# }
# write.csv(ppjg,"附件 3—留言主题-匹配结果.csv",row.names=FALSE)
#划分地区
dir.create("D:\\2020 泰迪杯\\运行\\问题 2\\地区划分表")
setwd("D:\\2020 泰迪杯\\运行\\问题 2\\地区划分表")
for (m in 1:9) {
  k<-1
  tmp<-data.frame()
  for(i in 1:nrow(fb_fj3)){
    if(grepl(paste0("A",m),fb_fj3[i,3])){
      for (j in 1:ncol(fb_fj3)) {
        tmp[k,j]<-fb_fj3[i,j]
      }
      k<-k+1
    }
  }
  write.csv(tmp,paste0("附件 3—留言主题-A",m,".csv"),row.names=FALSE)
}
setwd("D:\\2020 泰迪杯\\运行\\问题 2")
tmp<-data.frame()
k<-1
for (m in 1:9) {
  for(i in 1:nrow(fb_fj3)){
    if(grepl(paste0("A",m),fb_fj3[i,3])){
      for (j in 1:ncol(fb_fj3)) {
        tmp[k,j]<-fb_fj3[i,j]
      }
      tmp[k,8]<-paste0("A",m)
      k<-k+1
    }
  }
}
```

```
for(i in 1:ncol(fb_fj3)){
  colnames(tmp)[i]<-colnames(fb_fj3)[i]
}
colnames(tmp)[8]<-"所属地区"
#关注数
str(tmp)
for (i in 1:nrow(tmp)) {
  tmp[i,9]<-tmp[i,6]+tmp[i,7]+1
  tmp[i,10]<-log(tmp[i,6]+tmp[i,7]+1)
}
min(tmp$V9)
max(tmp$V9)
min(tmp$V10)
max(tmp$V10)
colnames(tmp)[9]<-"关注数"
colnames(tmp)[10]<-"关注数取对数"
write.csv(tmp,"附件 3—留言主题-地区划分总表.csv",row.names=FALSE)
# tmp<-read.csv("附件 3—留言主题-地区划分总表.csv")
table(tmp$所属地区)
#对应分词
dir.create("D:\\2020 泰迪杯\\运行\\问题 2\\地区划分分词")
setwd("D:\\2020 泰迪杯\\运行\\问题 2\\地区划分分词")
for (m in 1:9) {
  tmp1<-tmp[tmp$所属地区==paste0("A",m),]
  tmp2<-data.frame()
  n<-1
  for(i in 1:nrow(tmp1)){
    cn<-tmp1[i,3]
    tag_result<-tagging(cn,tag_worker) %>% enframe()
    k<-1
    if(nrow(tag_result)!=0){
      for (j in 1:nrow(tag_result)) {
        tmp2[k,n]<-tag_result[j,1]
        tmp2[k,n+1]<-tag_result[j,2]
        colnames(tmp2)[n]<-paste0(tmp1[i,1],"name")
        colnames(tmp2)[n+1]<-paste0(tmp1[i,1],"value")
        k<-k+1
      }
      n<-n+2
    }
  }
  for(i1 in 1:nrow(tmp2)){
    for(j1 in 1:ncol(tmp2)){
      if(is.na(tmp2[i1,j1])){
```

```
        tmp2[i1,j1]<-0
      }
    }
  }
  write.csv(tmp2,paste0(paste0("A",m),"分词.csv"),row.names=FALSE)
}
dir.create("D:\\2020 泰迪杯\\运行\\问题 2\\地区划分分词匹配结果")
setwd("D:\\2020 泰迪杯\\运行\\问题 2\\地区划分分词匹配结果")
for (m in 1:9) {
  ppjg<-data.frame()
  tmp2<-tmp[tmp$所属地区==paste0("A",m),]
  tmp1<-read.csv(paste0("D:\\2020 泰迪杯\\运行\\问题 2\\地区划分分词\\A",m,"分词.csv"))
  for(i1 in 1:ncol(tmp1)){
    tmp1[,i1]<-as.character(tmp1[,i1])
  }
  for(i in 1:nrow(tmp2)){
    pptmp1<-data.frame(tmp1[,paste0("X",tmp2[i,1],"name")],
                       tmp1[,paste0("X",tmp2[i,1],"value")])
    for(i1 in 1:ncol(pptmp1)){
      pptmp1[,i1]<-as.character(pptmp1[,i1])
    }
    for(j in 1:nrow(tmp2)){
      pptmp2<-data.frame(tmp1[,paste0("X",tmp2[j,1],"name")],
                         tmp1[,paste0("X",tmp2[j,1],"value")])
      for(i1 in 1:ncol(pptmp2)){
        pptmp2[,i1]<-as.character(pptmp2[,i1])
      }
      flag<-0
      for(n in 1:nrow(pptmp1)){
        a1<-pptmp1[n,1]
        a2<-pptmp1[n,2]
        if(a1=="n"|a1=="v"){
          for(m1 in 1:nrow(pptmp2)){
            b1<-pptmp2[m1,1]
            b2<-pptmp2[m1,2]
            if(b1!="0"){
              if(a1==b1 & a2==b2){
                flag<-flag+1
                break
              }
            }else{
              break
            }
          }
```

```
        }
        ppjg[i,j]<-flag
      }
      if(i==1){
        colnames(ppjg)[j]<-tmp2[j,1]
      }
    }
    rownames(ppjg)[i]<-tmp2[i,1]
  }
  write.csv(ppjg,paste0(paste0("A",m),"分词匹配结果.csv"),row.names=FALSE)
}
tmp1<-read.csv("A1 分词匹配结果.csv")
tmp2<-data.frame()
for (i in 1:nrow(tmp1)) {
  flag<-0
  for(j in 1:ncol(tmp1)){
    if(i!=j & tmp1[i,j]!=0){
      flag<-flag+tmp1[i,j]
    }
  }
  tmp2[i,1]<-flag
}
tmp1<-read.csv("D:\\2020 泰迪杯\\运行\\问题 2\\地区划分表\\附件 3—留言主题-A1.csv")
for(i in 1:nrow(tmp2)){
  rownames(tmp2)[i]<-tmp1[i,1]
}
write.csv(tmp2,"D:\\2020 泰迪杯\\运行\\问题 2\\A1 匹配表.csv")
```

二、MatLab 代码

1. F-score 评价

```
%需要将总结果第一行数据删除作为总结果-改
zjg=csvread("D:\2020 泰迪杯\运行\问题 1\总结果-改.csv");
pj=0;
flag=0;
for k = 1:15
    cql=0;
    czl=0;
    flag1=0;
    for i = 1:9210
        if zjg(20,i)==k
            flag1=flag1+1;
        end
    end
    if flag1~=0
```

```
        for i = 1:9210
            for j = 1:19
                if zjg(j,i)~=0
                    if zjg(j,i)==k
                        cql=cql+1;
                        if zjg(j,i)==zjg(20,i)
                            czl=double(czl+1/(zjg(21,i)));
                        end
                    end
                end
            end
        end
    end
    cql=double(czl/cql);
    czl=double(czl/flag1);
    if (cql+czl)~=0
        pj=pj+double(2*(cql*czl)/(cql+czl));
        flag = flag+1;
    end
end
pj/flag
```