基于朴素贝叶斯和 NLP 以及主成分分析的“智慧政务”中的文本挖掘应用

摘要

随着网络问政平台逐步成为政府部门处理政务的重要渠道，从大量各类社情民意相关的文本数据中，通过自然语言处理技术以及运用数据挖掘的方法来处理留言分类和热点整理工作对提升政府的管理水平和施政效率带来极大方便。

针对问题 1，根据所给数据，建立关于留言内容的一级标签分类模型。分析数据可知，留言是文本型数据，需要采用自然语言处理（NLP）方法，要建立分类模型，就涉及到机器学习的分类模型。在建立模型前，运用自然语言处理计数对文本型数据进行预处理工作。然后需要将得到的非结构化数据转化为结构化数据，将文本量化。本题采用的是 TF-IDF 方法。然后，将附件 2 中的数据划分训练集和测试集，训练集用于模型训练。选用了三个分类模型分别用数据进行训练，在不断调整后得到测试集分类准确率如下：高斯朴素贝叶斯模型为准确率为 0.66，互补朴素贝叶斯模型准确率为 0.85，决策树模型的准确率为 0.338。由于准确率的局限性，又采用 F1-Score 评价标准对三个模型进行评价。三个模型分别得分为：GaussianNB：0.65，ComplementNB：0.86，DecisionTree：0.3，由此可知互补朴素贝叶斯的分类效果最好。

针对问题 2，从数据集中提取热点问题。热点问题可以理解为某一时间段或某一区域内群众集中反映的问题。在一段确定的时间段内，地域和人群很大程度上代表了热点问题的内容。大致可以分为三个流程来完成：首先，根据留言主题用 FoolNltk 提取出留言所提及的地点、根据地点将所有留言分组后按照留言数量对地点进行降序排列，结果见表 2-2。其次，建立针对留言的留言热度指标：经过 Jieba 提取留言主题关键词、使用 doc2bow 工具、TF-IDF 模型求得某条留言与所有留言相似度之和，将结果与留言点赞数的数量级加权相乘作为留言热度，结果见表 2-6。最后，将每个地点中的留言按照热度降序排列，选取第一条留言提取问题主要内容作为该地点的热点问题，对每个地点的留言热度求和作为热点问题的热度指数，统计留言的最早时间和最晚时间作为热点问题的时间范围。提取出热点问题之后，根据留言编号将所有留言纳入相应的热点问题，得出热点问题明细表。

针对问题 3，对答复意见给出评价。对答复意见的质量评价，从答复的相关性、完整性、可解释性等角度衡量。相关性通过计算留言和答复的相似性来量化，可得到每个回复对于留言的相似度，结果见表 3-1；完整性通过提取文本中的关键词，统计答复意见中出现在留言详情中的关键词个数，并计算复现率，复现率越高，回答越完整。结果见表 3-4；可解释性认为如果答复中出现相关法律，政策，则答复具有理论支撑，即具有可解释性。通过统计每条回复中的法律条文总数，来量化完整性，结果见表 3-5。在三个指标都量化出来后，选用主成分分析模型，对三个因素降维处理，得到每条答复意见的主成分因子得分，通过这个得分来综合衡量答复意见的质量。得分越高的，质量越高，结果见表 3-6。

关键词：NLP、TF-IDF 算法、分类模型、热点挖掘、主成分分析

Abstract

As the online political inquiry platform has gradually become an important channel for government departments to handle government affairs, from a large number of various types of social data and public opinion-related text data, natural language processing technology and data mining methods are used to process message classification and hotspot sorting work. The management level and efficiency of administration have brought great convenience.

For question 1, according to the given data, establish a first-level label classification model about the content of the message. It can be seen from the analysis of the data that the message is text-based data, which requires the use of natural language processing (NLP) methods. To establish a classification model, it involves machine learning classification models. Before building the model, the natural language processing count is used to preprocess the text data. Then you need to convert the resulting unstructured data into structured data to quantify the text. This question uses the TF-IDF method. Then, divide the data in Annex 2 into a training set and a test set, and the training set is used for model training. Three classification models were selected for training with data, and after continuous adjustment, the classification accuracy of the test set was as follows: the accuracy rate of the Gaussian Naive Bayes model is 0.66, the accuracy rate of the complementary Naive Bayes model is 0.85, and the decision tree model The accuracy rate is 0.338. Due to the limitation of accuracy, the three models were evaluated using F1-Score evaluation standard. The scores of the three models are: GaussianNB: 0.65, ComplementNB: 0.86, DecisionTree: 0.3, which shows that the classification effect of complementary naive Bayes is the best.

For problem 2, extract hot issues from the data set. Hot issues can be understood as problems that are collectively reflected by people in a certain period of time or in a certain area. Within a certain period of time, the region and the crowd largely represent the content of hot issues. It can be roughly divided into three processes to complete: First, use FoolNltk to extract the locations mentioned in the message according to the subject of the message, group all the messages according to the location, and then sort the locations in descending order according to the number of messages. The results are shown in Table 2-2. Secondly, establish a message popularity index for the message: extract the subject keyword of the message through Jieba, use the doc2bow tool, TF-IDF model to find the sum of the similarity between a certain message and all messages, and weight the result with the order of the number of likes of the message. Multiplied as the message popularity, the results are shown in Table 2-8. Finally, arrange the messages in each location in descending order of popularity, select the first message to extract the main content of the problem as the hotspot problem at that location, sum up the message popularity of each location as the hotness index of the hotspot problem, count the earliest messages Time and latest time are used as the time range of hot issues. After extracting the hot issues, all messages are included in the corresponding hot issues according to the message number, and a list of hot issues is obtained.

For question 3, give comments on the answers. The evaluation of the quality of the responses is measured from the perspectives of relevance, completeness, and interpretability of the responses. The relevance is quantified by calculating the similarity of the message and the reply, and the similarity of each reply to the message can be obtained. The results are shown in Table 3-1; the completeness is extracted from the keywords in the text, and the statistical comments appear in the message details The number of keywords and calculate the recurrence rate, the higher the recurrence rate, the more complete the answer. The results are shown in Table 3-4. Interpretability believes that if relevant laws and policies appear in the response, the response has theoretical support, that is, it is interpretable. Quantify the completeness by counting the total number of legal provisions in each reply. The results are shown in Table 3-5. After the three indicators are quantified, the principal component analysis model is selected to reduce the dimension of the three factors, and the principal component factor score of each reply is obtained, and the score is used to comprehensively measure the quality of the reply. The higher the score, the higher the quality. The results are shown in Table 3-6. For example, the overall score of the first comment is 0.399.

Keywords: NLP, TF-IDF algorithm, classification model, hot spot mining, principal component analysis

目录

摘要.....1

1 挖掘背景与目标.....5

2 问题分析.....6

2.1 问题一的分析.....6

2.2 问题二的分析.....6

2.3 问题三的分析.....8

3 模型假设与符号说明.....9

3.1 模型假设.....9

3.2 符号说明.....9

4 分析方法与过程.....9

4.1 问题 1 分析方法与过程.....9

4.2 问题 2 分析方法与过程.....18

4.3 问题 3 分析方法与过程.....29

5 总结.....32

参考文献.....34

附录.....35

I.问题一程序代码.....35

II.问题二程序代码.....35

III.问题三程序代码.....41

1 挖掘背景与目标

1.1 挖掘背景

传统城市和政府按照业务、管理职责分别设立相关的部门，各部门各司其职，对群众反馈留言进行处理，存在严重的部门壁垒。并且随着信息通讯技术的发展，政府面临电子化、信息化压力。所以政务系统智能化转型成为推动政务发展的必不可少的发展趋势。

随着微信、微博、市长信箱、阳光热线等网络问政平台的广泛应用，政府可以通过这些平台更好的了解民意、汇聚民智、凝聚民气，人民也可以更好的对政府进行建议以及合法表达自己的诉求。以往对于少量数据都是依靠人工进行留言划分和热点整理等工作，但随着大数据、人工智能的发展，数据量的极大攀升，海量的文本数据再通过人工进行处理就显得尤为浪费时间以及耗费人力。所以，建立基于自然语言处理技术（NLP）结合数据挖掘技术的智慧政务系统已成为社会处理发展的新趋势，并且采用智慧政务系统会给更富的管理水平和施政效率的提升带来极大推动。

目前智慧政务系统主要在处理群众留言数据，进行留言分类，热点问题挖掘，评论回复等方面对人工依赖性较大。对于留言分类问题，工作人员按照一定的划分体系对留言进行分类，当前大部分电子政务系统仍依靠人工根据经验处理，存在工作量大、效率低，且差错率较高的问题。热点问题挖掘，答复意见评价等，也有相同问题。如何降低大部分电子政务系统的人工依赖性，实现智能化处理是解决问题的关键。

智慧政务的建设是实现智慧政务升级发展的突破口，是政府从“管理型”走向“服务型、智慧型”的必然产物。

1.2 挖掘目标

（1）参考附件 1 提供的内容分类三级标签体系，根据附件 2 提供的关于留言内容的一级标签分类数据，建立关于留言内容的一级标签分类模型。并使用 F-Score 对分类方法进行评价。

（2）参考附件 3 提供的留言详情数据，将某一时间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体系，并给出评价结果。

（3）参考附件 4 提供的数据，根据相关部门对留言答复意见，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。

2 问题分析

2.1 问题一的分析

问题一中，要求建立关于留言内容的一级标签分类模型。可以利用机器学习里面的分类算法。对附件 2 的数据进行分析，发现留言主题，留言内容等都是文本，要将文本信息量化分类，运用自然语言处理技术。在建立模型前进行数据预处理工作，包含抽取数据、数据清洗、分词、去停用词、建模准备数据准备（文本数据转化为数字信息）等。再建立多个分类模型，通过训练数据对模型进行训练，再对得到的测试数据进行准确度计算以及 F1-Score 得分比较，最后对模型进行评价和优化。

2.2 问题二的分析

问题二主要任务是从数据集中提取热点问题。热点问题可以理解为某一时间段或某一区域内群众集中反映的问题。在一段确定的时间段内，地域和人群很大程度上代表了热点问题的内容。虽然一个地点可能同时反应多个问题，但它们的热度差距通常比较大，可以根据热度选取最具代表性的一个热点问题。按照这样的逆向思路，可以将问题二分为三个流程：首先，根据留言主题提取出留言所提及的地点、根据地点将相似的留言分组、将地点按照留言数量进行降序排列。其次，对每个地点的留言计算每条留言的留言热度、将每个地点中的留言按照热度降序排列，选取第一条留言提取问题主要内容作为该地点的热点问题。最后，对每个地点的留言热度求和作为热点问题的热度指数，统计留言的最早时间和最晚时间作为热点问题的时间范围。

这种做法的好处是几乎不用人工干预、挖掘出的热点问题有很强的地域性。其关键是要寻找一个针对留言计算热度的合理方法。题目所要求处理的数据集如表 1 所示。

表 1 问题二 部分原始数据表

留言编	留言用户	留言主题	留言时间	留言详	反对数	点赞数
188006	A0001029	A3 区一米	2019/2/2	座落在 A	0	0

188007	A0007479	咨询 A6 区	2019/2/1	A 市 A6	0	1
188031	A0004006	反映 A7 县	2019/7/1	本人系	0	1
188039	A0008137	A2 区黄兴	2019/8/1	靠近黄	0	1
188059	A0002857	A 市 A3 区中	2019/11/	A 市 A3	0	0

从表 1 中可以看出中很少有直接的留言热度的信息，留言点赞数可以纳入考虑，但是经过分析发现不同留言之间的点赞数（见图 1）差别太大，只能将少数留言点赞数高的留言区分出来区分出来，甚至不足以形成题目所要求的五个热点的主题。

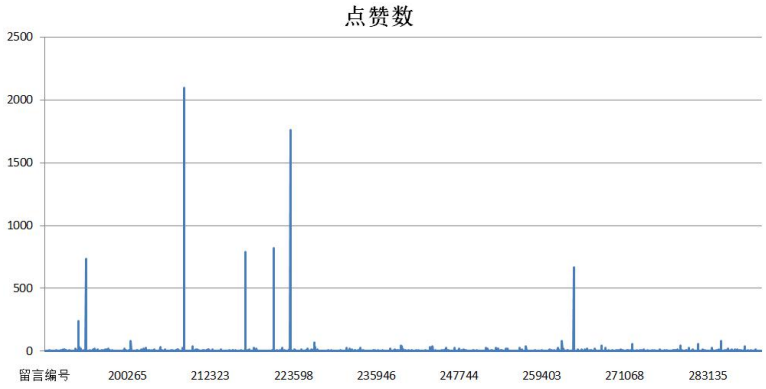

图 1 各留言的点赞数条形图

如图 1 所示，所有留言中最大点赞数超过两千，少数 6 条留言超过 500 点赞，而绝大多数留言的点赞数都接近于 0。不能只关心极少数点赞数高的留言，因为有许多热点问题可能就隐藏在低点赞数的留言之中。因此，首先需要从点赞数的角度建立“影响深度指标”，代表留言所描述问题影响的严重程度。点赞数越高，该地点留言问题越严重，影响深度指标就越大。另外，还需要从留言问题出现频次的角度建立“影响广度指标”，代表留言所描述问题影响的传播程度。频次越高，该问题牵涉的范围就越大，影响广度指标就越大。最后，将两个指标结合，作为最终的留言热度指数。

对于留言问题的影响深度指标建立，考虑先把留言按照点赞数的数量级划分成多个组，每个组赋予不同的权值，这样做对点赞数接近 0 的留言比较公平，缩小了指标在留言之间的差距(见图 2)，而保留了一定的区别。

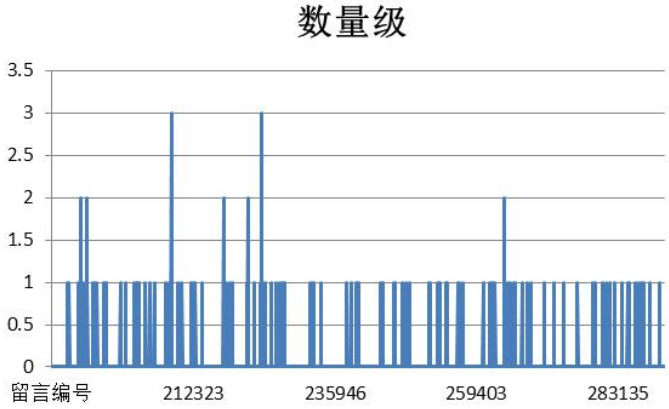

图 2 各留言点赞数数量级

如图 2 所示，数量级单位以 10 为底数，10^0 的留言居多，少数高点赞数留言调整到了同一数量级。因此，影响深度指标可考虑为和数量级成正比的数值。

留言问题的影响广度指标，主要是建立在留言主题频次基础之上的。但是对同一个问题的留言主题表达千变万化，无法直接统计留言主题的频次。换个角度，一条留言所对应的问题影响若很广，这条留言就会和数据集中很多留言都相似，因为有更多的不同人和他发出相似的言论。也就是说，如果能计算一条留言和所有其他留言的相似度，那么就可以用与所有留言相似度之和来表示影响广度。

至此，完成问题二任务流程所需要的指标全部分析完毕。得出热点问题表之后，可根据留言编号将所有留言纳入对应的热点问题，得到热点问题明细表。

2.3 问题三的分析

针对问题 3，对答复意见的质量给出一套评价方案并实现。对答复意见的评价可考虑从相关性、完整性、可解释性等角度建立指标，进行量化，结合统计学方法给出评价方案。相关性从答复意见的内容是否与问题相关考虑，可以采取对留言详情和答复意见做相似度分析，得到两个文本的相似度来对相似性进行量化；完整性可以通过提取留言详和答复意见的关键词，通过答复意见中提取出的关键词在留言详情中的关键词所占的比例，来反映答复意见的完整性，答复意见中包含留言详情中的关键词越多，比例越大，完整性越高；可解释性考虑答复意见中是否给出内容的相关解释，通过对数据的分析，发现答复较全面，更具可解释性的回复评论中都包含一条及以上的相关的法律条令，政策等理论支撑来解释可解释性，出现相关政策越多的，理论支持越强，对内容可解释性越强，以每条回复中出现的政策条数计数来进行简单量化。最后采用主成分分析法，将量化后

的三个对答复内容的影响因素进行主成分分析，得到一个衡量答复意见好坏的最终的主成分，来对答复意见进行评价。

3 模型假设与符号说明

3.1 模型假设

（1）假设所给数据的真实性，可靠性。

（2）假设回复内容的法规内容真实有效。

（3）假设忽略的所有数据均对问题的分析不产生影响。

（4）隐马尔可夫模型中隐藏的马尔可夫链在任意时刻的状态只依赖与前一时刻的状态，与其他时刻的状态及预测无关。

（5） 隐马尔可夫模型中假设任意时刻的观测只依赖于该时刻的马尔可夫链状态，与其他观测及状态无关。

3.2 符号说明

表 2 符号说明表

符号	含义
TP	被判定为正样本，事实上也是正样本
FP	被判定为正样本，但事实上是负样本
TN	被判定为负样本，事实上也是负样本
FN	被判定为负样本，但事实上是正样本

4 分析方法与过程

4.1 问题 1 分析方法与过程

问题 1 的留言归类问题是一个文本挖掘过程

1-1 文本挖掘流程

4.1.1 数据预处理

题目要求处理的原始数据如图 1-5 所示。

留言编号	留言用户	留言主题	留言时间	留言详情	一级标签
24	A00074011	A市西湖建筑集团占道施工有安全隐患	20/1/6 12:09:	围墙内。每天尤其	城乡建设
37	U0008473	在水一方大厦人为烂尾多年，安全隐患严	20/1/4 11:17:	着，不但占用人行道	城乡建设
83	A00063999	投诉A市A1区苑物业违规收停车费	19/12/30 17:06	央。不知程明物业如	城乡建设
303	U0007137	A1区蔡锷南路A2区华庭楼顶水箱长年不洗	19/12/6 14:40:	，霉是一种强致癌	城乡建设
319	U0007137	A1区A2区华庭自来水好大一股霉味	19/12/5 11:17:	，霉是一种强致癌	城乡建设
379	A00016773	投诉A市盛世耀凯小区物业无故停水	19/11/28 9:08:	业不是为业主服务的	城乡建设
382	U0005806	咨询A市楼盘集中供暖一事	19/11/27 17:14	月亮岛片区近年规划	城乡建设

图 1-2 未处理前留言数据

建模前需要先对文本数据做预处理，其流程可以表述为图 1-3。

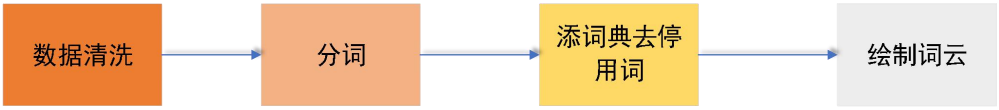

图 1-3 预处理流程图

（1）数据清洗

对留言主题的内容运用正则表达式，去除空格、字母、数字、无意义字母与数字组合标点符号，以及数据去重等操作。

（2）分词

中文分词是指以词为基本单位，使用计算机自动对中文文本进行词语的切分，即使词之间有空格，这样方便计算机识别出各语句的重点内容。此处采用结巴分词对清洗后的数据进行处理。

结巴分词算法是基于前缀词典实现高效的词图扫描，生成句子中汉字所有可能成词情况所构成的有向无环图（DAG）。采用了动态规划查找最大概率路径，找出基于词频的最大切分组合。对于未登录词，采用基于汉字成词能力的 HMM 模型，使用了 Viterbi 算法。此外，结巴分词支持用户添加自定义词典，对于具体问题，加入自定义的词典，可以增加分词的准确性。对于留言主题，便添加了出现频次较高的地点，人群，以提高分词的准确性。如地点“润和紫郡”，如果不自行添加分词词典，默认分词库将会将其拆分为“润和”，“紫郡”两个词。

（3）添词典去停用词

停用词，词典译为“电脑检索中的虚字、非检索用字”。停用词一定程度上相当于过滤词，不过过滤词的范围更大些。停用词大致可分为两类。一类是人类

语言中包含的功能词，语气助词、副词、介词、连接词等，通常自身并无明确的意义，只有将其放入一个完整的句子中才有一定作用。另一类词包括词汇词，这些词应用广泛，但是对这样的词搜索引擎无法保证得到真正相关的搜索结果，难以帮助缩小搜索范围，同时还会降低搜索的效率，所以通常将其除去以提高搜索性能。除了引用常规的停用词词典外，还需要根据不同的研究主题人为地判断和选择合适的停用词语。

（4）绘制词云

词云图是文本结果展示的有力工具，通过词云图的展示可以对留言主题数据分词后的高频词予以视觉上的强调突出效果，使得阅读者可以一眼获取主旨信息。

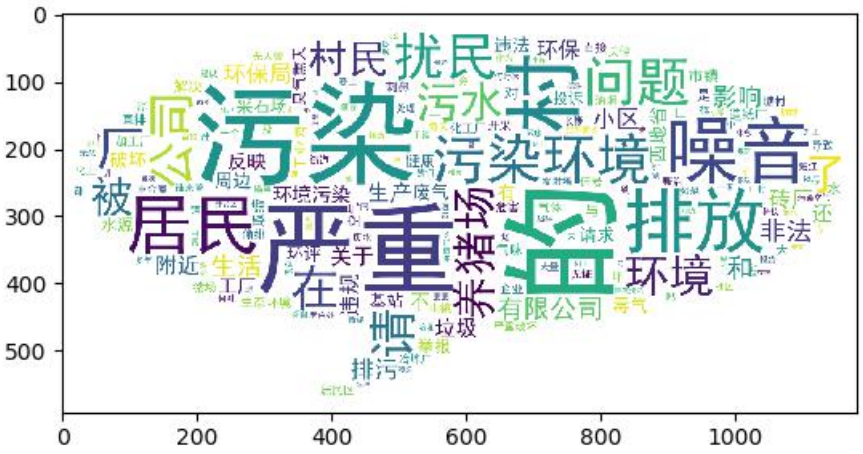

图 1-4 环境保护类词云

从图 1-4 可以看出，环境保护类出现的高频词是“环境污染”，“严重”，“排放”等词。

4.1.2 文本的向量表示

文本向量表示的目的是让计算机理解语言，将非结构化数据转化为结构化数据。向量表示方法包含 One-Hot 表达和 TF-IDF 权重策略。由于 One-Hot 表达忽略了句子的词频信息，所以，这里选用的是 TF-IDF 权重策略。

TF-IDF 的相关概念可以这样理解：①需要一个重要性调整系数，衡量一个词是不是常见词。如果某个词比较少见，但它在此文章中多次出现，那它很可能反映了这篇文章的特性，即关键词。②按统计学语言表达，即在词频的基础上，对

每个词分配一个“重要性”权重。最常见的词给与最小的权重，较常见的词给予较小的权重，较少见的词给予较大的权重。这个权重叫做“逆文档频率”（IDF），它的大小与一个词的常见程度成反比。③“词频”（缩写为 TF）④“逆文档频率”（缩写为 IDF）⑤将“词频”和“逆文档频率”相乘，就可以得到一个词的 TF-IDF 值，某个词对文章的重要性越高，它的 TF-IDF 值就越大。TF-IDF 关键词计算的流程为：

①计算词频：

$$\text{词频（TF）=某个词在文章中的出现次数} \tag{1}$$

不同的文章总的字数不一样，为了便于查找，通常进行一个“标准化”。

$$\text{词频}（TF）=\frac{\text{某个词在文章中的出现次数}}{\text{文章的总词数}} \tag{2}$$

②计算逆文档频率

需要一个语料库（corpus）,用来模拟语言的使用环境。指经科学取样和加工的大规模电子文本库。逆文档频率计算方式为：

$$\text{逆文档频率}（IDF）=\log(\frac{\text{语料库的文档总数}}{\text{包含该词的文档数}+1}) \tag{3}$$

如果一个词越常见，那么它的分母越大，逆文档频率就越小，接近于 0。分母加 1，是为了避免分母为 0（所有文档不包含该词）。log 表示对得到的值取对数。

③计算 TF-IDF

TF-IDF 与一个词在文档中的出现次数成正比，与该词在整个语言中的出现次数成反比。计算出文档的每个词的 TF-IDF 值，然后按降序排列，取排在最前面的几个词。

$$TF-IDF=\text{词频}（TF）\times\text{逆文档频率}（IDF） \tag{4}$$

4.1.3 构建分类器模型

机器学习的分类模型有很多，本题选用了常见的三类分类模型对训练数据进行模型训练，再用训练好的模型对测试数据进行测试。选择的三类分类模型分别为高斯朴素贝叶斯（Gaussian Naive Bayes）模型、决策树模型（Decision Tree）、互补朴素贝叶斯（ComplementNB/CMB）模型。

（1）高斯贝叶斯模型

高斯朴素贝叶斯（Gaussian Naive Bayes）模型是建立在朴素贝叶斯模型之上的。而朴素贝叶斯分类器是基于贝叶斯定理与特征条件独立同分布假设的分类方法。其计算公式为：

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{5}$$

贝叶斯定理是基于假设的先验概率、给定假设下观察到不同数据的概率，提供了一种计算后验概率的方法。独立同分布是指随机过程中，任何时刻的取值都为随机变量，如果这些随机变量服从同一分布，并且相互独立，那么这些随机变量是独立同分布。

$$P(x \mid c_k) = P(x_1, L, x_n \mid c_k) = \prod_{j=1}^{n} P(x_j \mid c_k) \tag{6}$$

可以从三个不同层面来理解贝叶斯原理，首先对于贝叶斯网络分类器，若某一待分类的样本 D，其分类特征为 $x = (x_1, x_2, ..., x_n)$,则样本 D 属于类别 y_j 的概率：

$$P(Y = y_j \mid X = x_1, L, X = x_n), (j = 1, L, m) \tag{7}$$

公式（7）应满足：

$$\max P(Y = y_j \mid X = x) \tag{8}$$

结合贝叶斯公式：

$$P(Y = y_j \mid X = x) = \frac{P(X = x / Y = y_j)P(Y = y_j)}{P(X = x)} \tag{9}$$

可以得到：

$$\max P(Y = y_j \mid X = x) \Rightarrow \max P(Y = y_j)P(X = x \mid Y = y_j) \tag{10}$$

因此估计 $P(Y = y_j \mid X = x)$ 的问题就转化为如何基于训练集数据 D 来估计先验概率 $P(Y = y_j)$ 和条件概率 $P(X = x \mid Y = y_j)$，这是第一个层面。

第二个层面，可以设 μ_n 是 n 次独立试验中事件 A 发生的次数，且事件 A 在每次试验中发生的概率为 P，则对任意正数 ε，有：

$$\lim_{n\to\infty} P\left(\left|\frac{\mu_n}{n} - p < \varepsilon\right|\right) = 1 \tag{11}$$

该定律是切比雪夫大数定律的特例，其含义是，当 n 足够大时，事件 A 出现的频率将几乎接近于其发生的概率，即频率的稳定性。当 n 足够大时，事件 A 出现的频率将几乎接近于其发生的概率，即概率的稳定性。根据大数定理，当训练集包含充足的独立同分布样本时，$P(Y=y_j)$可通过各类样本出现的频率来进行估计。此为第二个层面，第三个层面可在朴素贝叶斯算法中，假设样本独立同分布，此时：

$$\begin{aligned} &P(X=x \mid Y=y_j) = P(X=x_1, \mathrm{L}, X=x_{\mathrm{n}} \mid Y=y_j) \\ &= \prod_{i=1}^{n} P(X=x_i \mid Y=y_j) \end{aligned} \tag{12}$$

所以有，

$$\begin{aligned} &\max P(Y=y_j) P(X=x \mid Y=y_j) \\ &= \max P(Y=y_j) \prod_{i=1}^{n} P(X=x_i \mid Y=y_j) \end{aligned} \tag{13}$$

若共有 m 种标签，只需计算$P(y_k)P(x_1,\cdots,x_n \mid y_k), k=1,2,\cdots,m$取最大值作为预测的分类标签。

$$\hat{y} = \arg\max_k P(y) \prod_{i=1}^{n} P(x_i \mid y_k) \tag{14}$$

可以看出，的朴素贝叶斯只能处理离散数据，当$x_1,\cdots,x_n$是连续变量时，可以使用高斯朴素贝叶斯（Gaussian Naive Bayes）完成分类任务。

当处理连续数据时，一种经典的假设是：与每个类相关的连续变量的分布是基于高斯分布的，故高斯贝叶斯的公式是：

$$P(x_i = v \mid y_k) = \frac{1}{\sqrt{2\pi\sigma_{y_k}^2}} \exp\left(-\frac{v-\mu_{y_k}}{2\sigma_{y_k}^2}\right) \tag{15}$$

其中$\mu_y, \sigma_{y_k}^2$表示全部属于类y_k的样本中变量 x 的均值和方差。

还有一种贝叶斯模型称为多项式朴素贝叶斯，多项式朴素贝叶斯通常被用于处理多分类问题，比起原始的朴素贝叶斯分类效果有较大的提升。公式为：

$$P(x_i \mid y_k) = \frac{N_{y_k i} + \alpha}{N_y + \alpha n} \tag{16}$$

其中 $N_{y_k i} = \sum_{x \in T} x_i$ 表示在训练集中 y_k 具有特征 i 的样本的的数量，$N_y = \sum_{i=1}^{|r|} N_{yi}$ 表示训练集 T 中类 y_k 的特征总数。平滑系数 $\alpha > 0$ 防止零概率出现，当 $\alpha = 1$ 称为拉普拉斯平滑，而 $\alpha < 1$ 称为 Lidstone 平滑。

(2)互补朴素贝叶斯模型

ComplementNB 是标准多项式朴素贝叶斯(MNB)算法的一种改进，特别适用于不平衡数据集。具体来说，ComplementNB 使用来自每个类的补充的统计信息来计算模型的权重。

(3)决策树模型

决策树采用自顶向下的递归方式，在内部节点进行属性值的比较，并根据不同的属性值从该结点向下分支，最终得到叶节点是学习划分的类。

决策树是一种树状结构，它的每一个叶节点对应着一个分类非叶节点对应着在某个属性上的划分，根据样本在该属性上的不同取值将其划分成若干个子集。对于非纯的叶节点，多数类的标号给出到达这个节点的样本所属的类。构造决策树的核心问题是在每一步如何选择适当的属性对样本进行拆分。对一个分类问题，从已知类标记的训练样本中学习并构造出决策树是一个自上而下，分而治之的过程。本文选用 CART 决策树以基尼指数为准则来选择划分属性的决策树。

基尼值：

$$\begin{aligned} Gini(D) &= \sum_{k=1}^{|y|} \sum_{k' \neq k} p_k p_{k'} = \sum_{k=1}^{|y|} p_k \sum_{k' \neq k} p_{k'} \\ &= \sum_{k=1}^{|y|} p_k (1 - p_k) = 1 - \sum_{k=1}^{|y|} {p_k}^2 \end{aligned} \tag{17}$$

基尼指数的公式为：

$$Gini\_index(D,a) = \sum_{v=1}^{V} \frac{|D^v|}{D} Gini(D^v) \tag{18}$$

基于 CART 决策树的分类算法流程可以表述为：

①根据基尼指数公式（18）找出基尼指数最小的属性 a_*。

②计算属性a_*的所有可能取值的基尼值$Gini(D^v),v=1,2,\cdots,V$，选择基尼值最小的取值a_*^v作为划分点，将集合 D 划分为 D1 和 D2 两个集合（节点），其中 D1 集合的样本为$a_*=a_*^v$的样本，D2 集合为$a_*\neq a_*^v$的样本。

③对集合 D1 和 D2 重复步骤①和步骤③，直到满足停止条件。

4.1.4 分类结果与模型评估

(1)分类模型原理

按照 4.1.1-4.1.3 节处理后，训练出了三个分类模型。通过 F1-Score 和准确率（accuracy）来对分类模型进行评估。

Confusion Matrix		真实值	
		P	N
预测值	P'	TP	FP
	N'	FN	TN

图 1-5 混淆矩阵

准确率指的是分类正确的样本占总样本个数，即:

$$accuracy=\frac{n_{corrent}}{n_{total}} \tag{19}$$

其中，$n_{corrent}$表示被正确分类的样本个数，n_{total}表示样本总数。综合上图混淆矩阵，准确率公式还可表示为：

$$accuracy=\frac{TP+TN}{TP+TN+FP+FN} \tag{20}$$

准确率是分类问题中最简单直观的评价指标，但存在明显缺陷，比如正负样本的比例不均衡，设样本中正样本占 90%，负样本占 10%，那分类器只需要一直预测为正，就可以得到 90% 的准确率，但其实际性能是非常低下的。于是采用 F1-Score 来提升预测准确度与性能。F1-Score 是精确率和召回率的加权平均，计算公式为：

$$F1=\frac{2PR}{P+R} \tag{21}$$

查准率的计算公式为：

$$P = \frac{TP}{(TP + FP)} \tag{22}$$

查全率的计算公式为：

$$R = \frac{TP}{(TP + FN)} \tag{23}$$

P 体现了模型对负样本的区分能力，P 越高，模型对负样本的区分能力越强。R 体现了模型对正样本的识别能力，R 越高，模型对正样本的识别能力越强。

F1-Score 是两者的综合，F1-Score 越高，说明模型越稳健。

(2)分类结果

模型名称	SCORE/ACC	F_SCORE
GaussianNB	0.66	0.65
DecisionTree	0.338	0.3
ComplementNB	0.85	0.86

图 1-6 模型分类的 acc/F1-Score

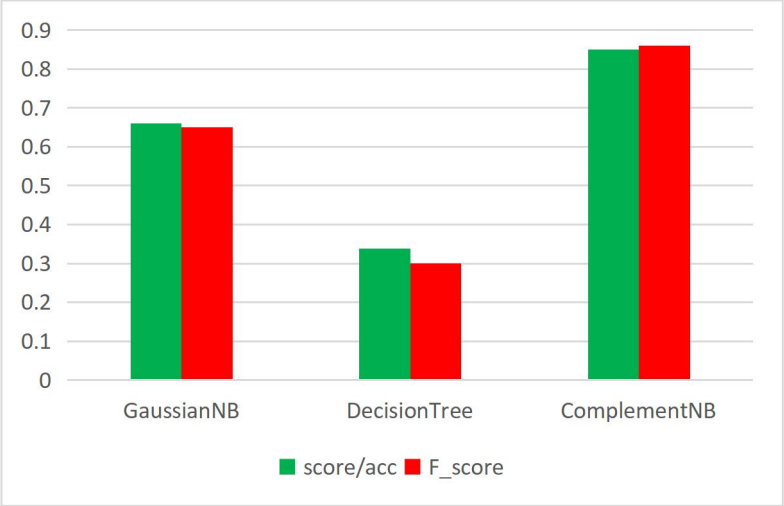

图 1-7 模型分类的 acc/F1-Score 对比图

从图 1-6、1-7 中可以看出，三个分类模型中，分类准确率以及 F1-Score 分数最高的是互补朴素贝叶斯模型。

因为 ComplementNB 使用来自每个类的补充的统计信息来计算模型的权重。CNB 的发明者通过实验结果表明，CNB 的参数估计比 MNB 的参数估计更稳定。此外，在文本分类任务上，CNB 通常比 MNB 表现得更好(通常是相当大的优势)。所以三个模型中 CNB 的分类效果最好，达到 86%。

4.2 问题 2 分析方法与过程

问题二的求解分为三个步骤：首先，根据留言主题提取出地点、根据地点将相似的留言分组。其次，对每个地点的留言计算每条留言的留言热度、选取一条留言提取问题主要内容作为该地点的热点问题。最后，计算热点问题的热度指数，统计留言的最早时间和最晚时间作为热点问题的时间范围。解决问题的思路可以用一张流程图表示，见图 2-1。

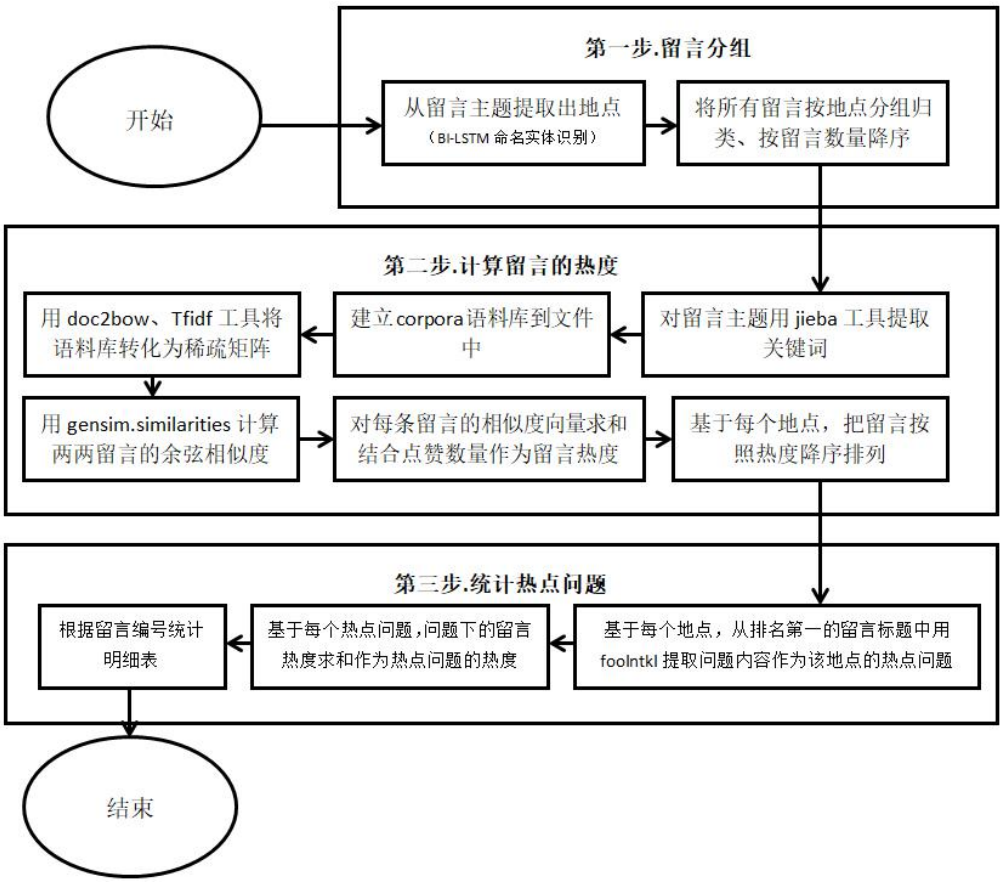

图 2-1 挖掘热点问题流程图

图 2-1 中 doc2bow、jieba、gensim、foolntkl 皆为 Python3.7 环境下实验可用到的工具。三个步骤的详细解答分别在：**4.2.1 留言分组**、**4.2.2 计算针对留言的热度**、**4.2.3 统计热点问题**小节中叙述。

4.2.1 留言分组

在一段确定的时间段内，热点问题发生的地点一定程度上可以代表热点问题的内容。即使一个地点可能同时反应多个问题，它们的热度差距通常也比较大，可以根据热度选取最具代表性的一个热点问题。因此需要对原始数据集(见表 1)

按地点进行分组，但在分组之前首先需要从留言主题中提取地点信息并附加到数据集中。提取地点信息一个可行的方法就是命名实体识别(Named Entity Recognition，NER)，它是信息提取、句法分析、问答系统等应用领域的重要基础工具[5]，

目前命名实体识别的主要方法可以归纳 3 类：基于规则的方法、基于统计的方法和基于深度学习的方法[6]。基于规则的方法需要领域专家来参与，这样对语言的要求高，可移植性不好。基于统计的方法主要使用统计学的方法来完成命名实体识别任务，比如隐马尔可夫模型(hidden markov models，HMM)[7]、条件随机场模型(conditional random field，CRF)[8]等，而基于深度学习的方法，不需要认为设定特征，能够从原始数据中自主的学习，因此可以减少人为对数据的干扰，找到更深层次和更加抽象的特征。本文使用的就是基于深度学习的模型，短期记忆网络(long short-term memory，LSTM) [5]。

由于循环神经网络(recurrent neural network,RNN)[9]存在梯度消失的问题，RNN 会丧失学习到远距离信息的能力。为了解决这个问题，Hochreiter 等提出了长短期记忆网络。LSTM 通过巧妙的设计来避免长期依赖问题。它与一般的 RNN 结构上并没有本质的区别，只是使用了不同结构的隐藏单元结构，在 LSTM 中存在一个被称为 Cell 的结构，它由 3 个门结构和 1 个细胞状态组成[6]，如图 2-2 所示。

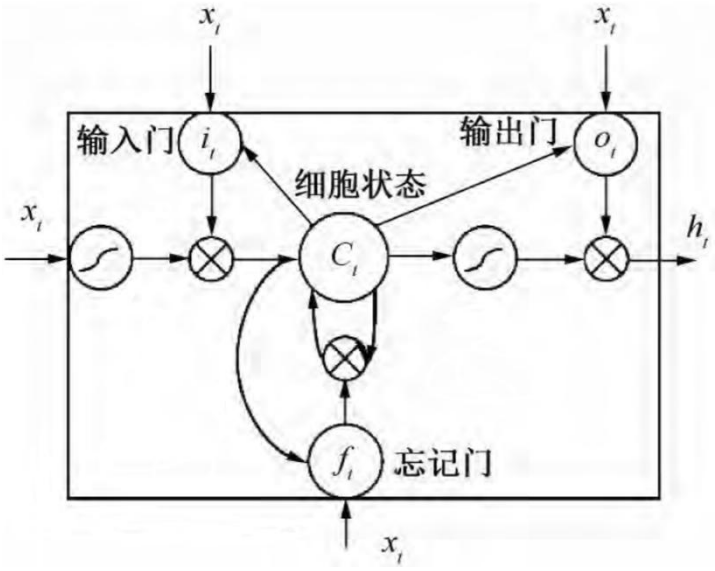

图 2-2 LSTM 记忆单元结构

从图 2-2 可以看出，LSTM 的记忆单元主要由输入门、输出门和忘记门组成。

输入门主要用于控制哪部分内容可以输入到细胞状态中，忘记门主要用于控制细胞状态中内容的存储，是继续存储细胞中 的历史内容或者是忘记细胞中的内容；输出门主要是用来控制哪些细胞状态中的内容可以被输出。LSTM 单元的具体工作流程为[6]：

$$f = \sigma(\mathrm{W}_f[h_{t-1}, x_t] + b_f) \tag{24}$$

$$i_t = \sigma(\mathrm{W}_i[h_{t-1}, x_t] + b_i) \tag{25}$$

$$o_t = \sigma(\mathrm{W}_o[h_{t-1}, x_t] + b_o) \tag{26}$$

$$\widetilde{C}_t = \tanh(\mathrm{W}_c[h_{t-1}, x_t] + b_c) \tag{27}$$

$$C_t = f_t C_{t-1} + i_t \widetilde{C}_t \tag{28}$$

$$h_t = o_t \tanh(C_t) \tag{29}$$

式中：f_t、i_t、o_t 分别为 t 时刻的遗忘门、输入门和输出门的输出；$\widetilde{C}_t$、C_t 分别为 t 时刻细胞中存储的内容；C_t 为 t 时刻 cell 中存储旧内容的输出；$\widetilde{C}_t$ 为 t 时刻 cell 中存储新内容的输出；x_t 、h_t 分别为 t 时刻的输入向量和隐藏层的向量；σ 表示使用的激活函数是 sigmoid 函数；W_i、b_i 分别为不同门的时候对应的权重矩阵和偏置向量。单向的 LSTM 神经网络模型，只能获取句子的上文信息，对于句子后文的上下文信息无法获取[6]。 如图 2-3 所示。

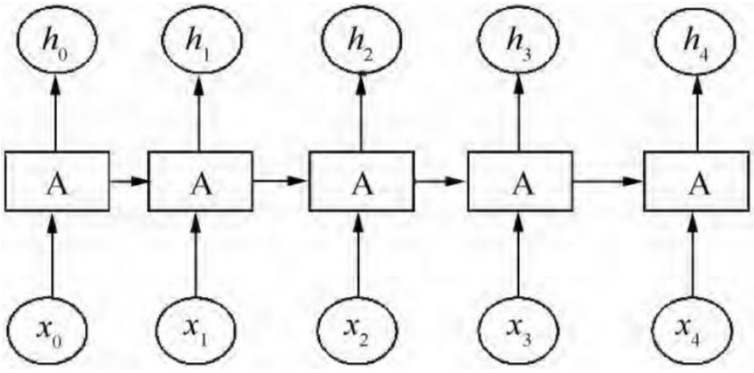

图 2-3 单向 LSTM 神经网络结构

比如输入“爱吃”，当网络接收到“爱”的时候， 由于无法获得“爱”字后面的上下文信息，因此就可能会把“爱”和“吃”这 2 个字拆分成为 2 个词这样就导致了错误，这也是单向 LSTM 的一个局限性。

为了弥补单向 LSTM 无法获取句子后面的上下文信息，本文最终使用了双向长短时期神经网络(BI-LSTM) 的模型。BI-LSTM 神经网络模型，可以从前向和后向 2 个方向对句子进行建模，这样既能保存前面的上下文信息，也能同时考虑到句子未来的上下文信息，使其在中文命名实体识别任务中可以取得更好的效果[10]。具体网络结构如图 2-4 所示。

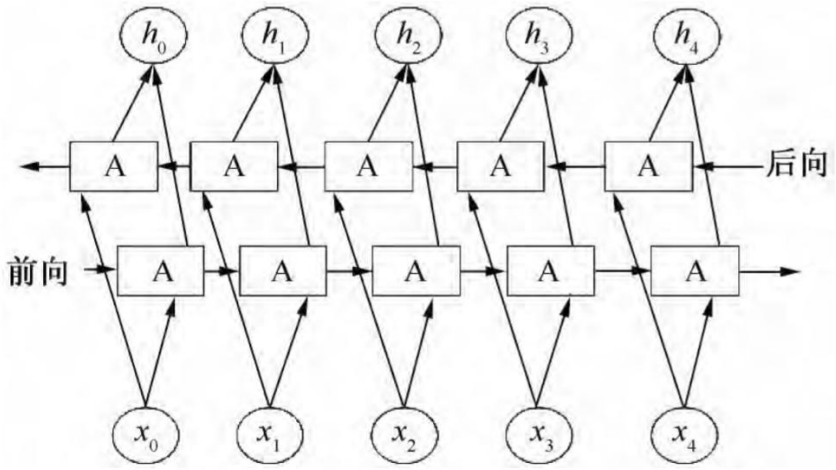

图 2-4 BI-LSTM 神经网络结构

从图 2-4 可以看出，BI-LSTM 模型分别从前向和后向对句子进行建模，然后将前向和后向的结果相 加起来作为下一层的输入，以这样的方式实现了对句子的双向建模。在 Python 编程领域，可以用 foolnktl 工具实现使用 BI-LSTM 的命名实体识别。使用 foolnktl 识别命名实体的完整过程见图 2-5 所示。

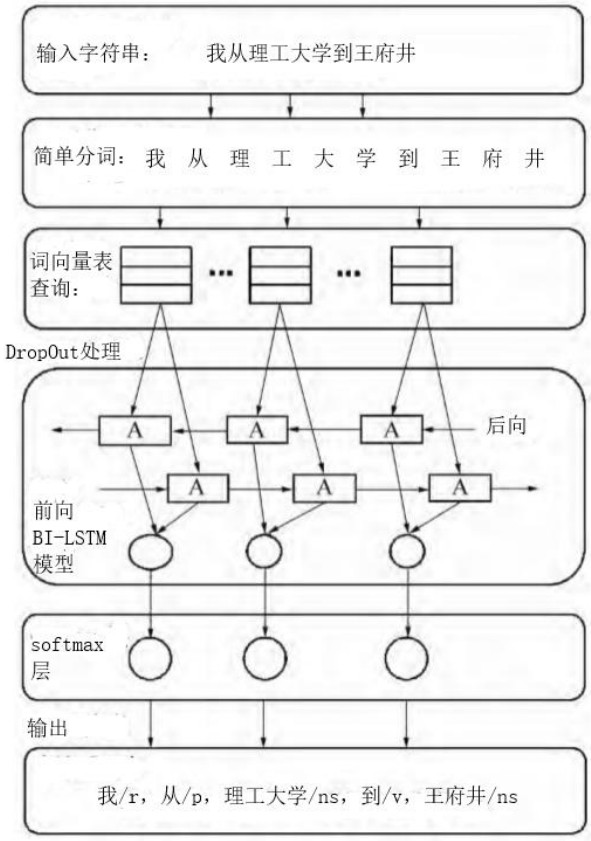

图 2-5 foolntkl 识别过程

首先将原始数据划分成单个词的组合，然后将句子按照每个词的顺序在词表中进行查找，找到每个词在词表中的相应位置，这里词表是所有原始数据组成的词汇表，每个词汇分配一个单独的 ID。再经过词表转换处理，然后句子中的每个词都会被表示为一个 one-hot 类型的词向量，然后再将词向量送入到 BI-LSTM 神经网络模型进行训练，最终实现中文命名实体识别。图 2-5 的例子的输出中 ns 表示该词属于地名。

针对问题所给数据集中的留言主题进行命名实体识别、保留地名并按照地名分组，部分结果见表 2-1。

表 2-1 提取留言涉及地点结果

留言主题	地点
A 市伊景园滨河苑捆绑销售车位	伊景园滨河苑
投诉 A 市伊景园滨河苑捆绑销售车位	伊景园滨河苑
投诉 A 市伊景园滨河苑捆绑车位销售	伊景园滨河苑
A 市伊景园滨河苑项目捆绑销售车位	伊景园滨河苑
关于伊景园滨河苑捆绑销售车位的投诉	伊景园滨河苑

投诉伊景园滨河苑项目违法捆绑车位销售	伊景园滨河苑
投诉伊景园滨河苑捆绑销售车位问题	伊景园滨河苑
伊景园滨河苑捆绑车位销售合法吗？！	伊景园滨河苑
A2 区丽发新城附近修建搅拌厂噪音、灰尘污染	丽发新城搅拌厂
A5 区劳动东路魅力之城小区临街门面烧烤夜宵摊	劳动东路魅力之城小区
A5 区劳动东路魅力之城小区临街门面烧烤夜宵摊	劳动东路魅力之城小区
关于伊景园滨河苑捆绑销售车位的维权投诉	伊景园滨河苑
A5 区劳动东路魅力之城小区油烟扰民	劳动东路魅力之城小区

注：完整结果见文件 TopicHotnessWithLocation.csv

从表 2-1 中第二列表示留言主题经过命名实体识别之后的结果，由于只保留了命名实体识别中的地名 ns，所以不包含标点符号。分组完毕之后，将地点相同地点的留言合并到一起，见表 2-2。

表 2-2 每个地点的留言表

地点	该地点下的留言
A 市伊景园滨河苑	A 市伊景园滨河苑捆绑销售车位', '伊景园滨河苑', ']
	投诉 A 市伊景园滨河苑捆绑车位销售', '伊景园滨河苑', '
	关于伊景园滨河苑捆绑销售车位的投诉', '伊景园滨河苑
劳动东路魅力之城小区	A5 区劳动东路魅力之城小区油烟扰民', '劳动东路魅力之
	A5 区劳动东路魅力之城小区临街门面烧烤夜宵摊', '劳动
	A5 区劳动东路魅力之城小区一楼的夜宵摊严重污染附近的
A 市经济学院	A 市经济学院强制学生实习', 'A 市经济学院', '经济学院
	A 市经济学院强制学生外出实习', 'A 市经济学院', '经济
	A 市经济学院寒假过年期间组织学生去工厂工作', 'A 市经

注：完整数据见 GroupByLocation.csv

4.2.2 计算留言的热度

按照 2.2 节问题分析中的思路，热度指标由影响广度指标和影响深度指标构成，首先计算影响广度指标。计算广度指标需要知道两两留言标题之间的相似度，计算相似度之前需要把留言标题向量化，也称作文本的表示。文本的表示有很多种方法，比如基于计数统计的独热编码、词袋方法[11]、基于频率的 TF-IDF 方法、基于神经网络的 word2vec 方法等等。本文使用的是 doc2bow 模型，可以将词组

转化为稀疏向量。

(1)jieba-TFIDF 提取关键词

为了得到 doc2bow 模型所需要的词组，需要一句话转化为多个词语的组合，称为分词或词条化。分词方法有基于规则、基于统计、基于语义、基于理解四个种类。本文使用的 jieba 分词是基于统计的分词方法，是对隐马尔可夫模型(Hidden Markov Model, HMM)的开源实现。隐马尔可夫模型是关于时间序列的概率模型，描述了这样一个过程：先由隐藏的马尔科夫链随机生成的不可预测的随机状态序列，再由产生的状态生成一个可观测的随机序列过程。其中，把由隐马尔可夫链随机生成的状态序列，称为状态序列(State Sequece)；把每个状态生成观测组成的随机序列称为观测序列(Observation Sequence)。因此，HMM 就是定义了观测序列 x 和状态序列 y 的联合概率 $p(x,y)$。由于状态序列是一个马尔可夫链，而且状态序列不可见，因此称该过程为隐马尔科夫过程[12]。HMM 包含了一个五元组:

$$\lambda = \{A, B, \pi, Q, V\}$$

其中，$Q = \{q_1, q_2, \cdots, q_N\}$ ，称为状态值集合，N 为可能的状态数。$V = \{v_1, v_2, \cdots, v_M\}$ ，称为观测值集合 M 为可能的观测数。$A = [a_{ij}]$ ，称为转移概率矩阵，a_{ij} 表示从状态 i 到状态 j 的概率。$B = [b_j(k)]$ 称为观测概率矩阵，$b_j(k)$ 表示在状态 j 的条件下生成观测 v_k 的概率。π 称为初始状态概率向量。

在模型中，隐藏的马尔可夫链由状态转移概率矩阵 A 与初始状态概率向量 π 确定，该隐藏的马尔科夫链用来生成不可观测的状态序列。从状态序列生成观测序列是由观测概率矩阵 B 确定的。在把留言标题分词的过程中，可以将状态值 Q 设置为 $\{B, E, M, S\}$，分别表示某个词的 $\{(B\text{egin}), (End), (Middle), (\text{Sin}\,gle)\}$：开始、结尾、中间、独自成词，每个留言标题即为一段观测序列。比如“A 市 B 区噪音扰民”通过 HMM 求解得到的状态序列“BEBEBMME”，分词结果为“A 市/B 区/噪音扰民”。这样就把分词问题抽象为了一个解码问题，对于观测到的字符穿序列 $X = \{x_1, x_2, \cdots, x_T\}$ ，求解最大条件概率 $\max P(y_1, \cdots, y_T \mid X)$ ，每一个 x_i 对应的状态为 y_i。给定一个观测序列，通过 Viterbi[13]算法来找到使得条件概率最大的

状态序列。

分词完毕之后，根据 4.1 节介绍的 TF-IDF 算法计算每个词汇的 TF-IDF 值，该值代表了词汇的重要程度。然后根据 TF-IDF 提取出每条留言主题中的关键词，调用 gensim.corpra 建立语料库，见图 2-6。

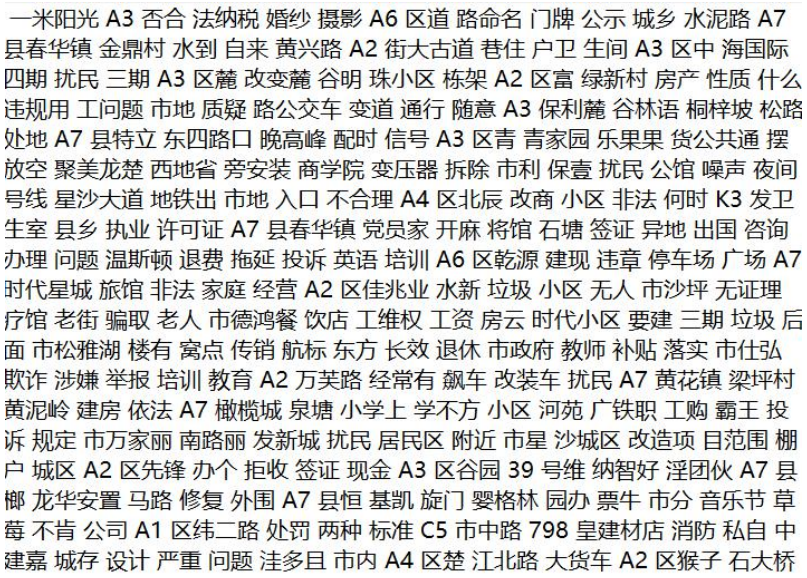
一米阳光 A3 否合 法纳税 婚纱 摄影 A6 区道 路命名 门牌 公示 城乡 水泥路 A7 县春华镇 金鼎村 水到 自来 黄兴路 A2 街大古道 巷住 户卫 生间 A3 区中 海国际 四期 扰民 三期 A3 区麓 改变麓 谷明 珠小区 栋架 A2 区富 绿新村 房产 性质 什么 违规用 工问题 市地 质疑 路公交车 变道 通行 随意 A3 保利麓 谷林语 桐梓坡 松路 处地 A7 县特立 东四路口 晚高峰 配时 信号 A3 区青 青家园 乐果果 货公共通 摆 放空 聚美龙楚 西地省 旁安装 商学院 变压器 拆除 市利 保壹 扰民 公馆 噪声 夜间 号线 星沙大道 地铁出 市地 入口 不合理 A4 区北辰 改商 小区 非法 何时 K3 发卫 生室 县乡 执业 许可证 A7 县春华镇 党员家 开麻 将馆 石塘 签证 异地 出国 咨询 办理 问题 温斯顿 退费 拖延 投诉 英语 培训 A6 区乾源 建现 违章 停车场 广场 A7 时代星城 旅馆 非法 家庭 经营 A2 区佳兆业 水新 垃圾 小区 无人 市沙坪 无证理 疗馆 老街 骗取 老人 市德鸿餐 饮店 工维权 工资 房云 时代小区 要建 三期 垃圾 后 面 市松雅湖 楼有 窝点 传销 航标 东方 长效 退休 市政府 教师 补贴 落实 市仕弘 欺诈 涉嫌 举报 培训 教育 A2 万芙路 经常有 飙车 改装车 扰民 A7 黄花镇 梁坪村 黄泥岭 建房 依法 A7 橄榄城 泉塘 小学上 学不方 小区 河苑 广铁职 工购 霸王 投 诉 规定 市万家丽 南路丽 发新城 扰民 居民区 附近 市星 沙城区 改造项 目范围 棚 户 城区 A2 区先锋 办个 拒收 签证 现金 A3 区谷园 39 号维 纳智好 淫团伙 A7 县 榔 龙华安置 马路 修复 外围 A7 县恒 基凯 旋门 婴格林 园办 票牛 市分 音乐节 草 莓 不肯 公司 A1 区纬二路 处罚 两种 标准 C5 市中路 798 皇建材店 消防 私自 中 建嘉 城存 设计 严重 问题 洼多且 市内 A4 区楚 江北路 大货车 A2 区猴子 石大桥

图 2-6 从关键词建立的语料库

注：完整数据见文件 keywords_text1.txt

从图 2-6 中可以看到，语料库中不存在标点符号、停用词汇，TF-IDF 提取的关键词中不会包含停用词，为 doc2bow 的使用做了铺垫。

(2)计算留言相似度

有了语料库，就可以用 doc2bow 结合语料库内容算出每条留言的词袋，结果见表 2-3。

表 2-3 部分留言的词袋

留言编号	词袋区					
25641	(0, 1)	(35, 1)	(37, 1)	(38, 1)	(98, 1)	(876, 1)
25642	(25, 1)	(40, 1)	(774, 1)	(1430, 1)	(6194, 1)	(6195, 1)
25643	(17, 1)	(37, 1)	(135, 1)	(762, 1)	(1582, 1)	(3133, 1)
25644	(86, 1)	(107, 1)	(677, 1)	(698, 1)	(1110, 1)	(1111, 1)
25645	(486, 1)	(1337, 1)	(6200, 1)	(6201, 1)	(6202, 1)	(6203, 1)
25646	(3356, 1)	(6204, 1)	(6205, 1)	(6206, 1)	(6207, 1)	(6208, 1)
25647	(270, 1)	(6209, 1)				
25648	(7, 1)	(836, 1)	(2178, 1)	(4456, 1)	(4496, 1)	(6210, 1)
25649	(17, 1)	(1931, 1)	(6213, 1)	(6214, 1)	(6215, 1)	(6216, 1)

注：完整结果见文件 bags.csv。

从表 2-3 可以看出，doc2bow 生成的词袋长度是不同的，以编号为 25647 的留言为例，词袋区的取值(270,6209)表示该留言的两个关键词分别在语料库的第 270

和第 6209 个位置出现。同时可以把词袋看作是稀疏向量，词袋区中有取值的位置在稀疏向量中对应的值为 1，其余为 0，这样就显现了将留言主题转化为向量，同时为留言相似度的计算节约不少时间和空间成本。有了文本向量后，计算两个留言的相似度就转化为了求解两个向量的相似度。计算相似度有很多方法，如欧式距离和余弦相似度。欧氏距离更加接近人们平常所说的距离，如果是平面上的两个点 $A(x_1,y_1)$ 和 $B(x_2,y_2)$，那么 A 与 B 的欧式距离就是：

$$\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$$

而余弦相似度用向量空间中两个向量夹角的余弦值作为衡量两个个体间差异的大小。相比距离度量，余弦相似度更加注重两个向量在方向上的差异，而非距离或长度上。图 2-7 可以形象地说明两个概念的差异：

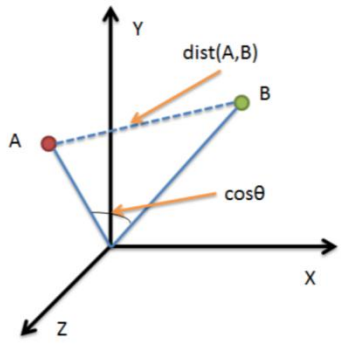

图 2-7 两种距离的区别

从图 2-7 可以看出，欧氏距离衡量的是空间各点的绝对距离，跟各个点所在的位置坐标直接相关；而余弦距离衡量的是空间向量的夹角，更加体现在方向上的差异，而不是位置。如果保持 A 点位置不变，B 点朝原方向远离坐标轴原点，那么这个时候余弦距离是保持不变的（因为夹角没有发生变化），而 A 与 B 两点的距离显然在发生变化，这就是欧式距离与余弦相似度的不同之处。欧氏距离能够体现个体数值特征的差异，但对绝对数值的变化比较敏感[14]，高纬度向量计算需要时间长，所以本文选择的是余弦相似度作为相似度的度量标准，python3.7 中计算得两两留言相似度矩阵，见表 2-4

表 2-4 留言相似度矩阵

相似度	A 市经济学院寒假过年期间组织学生去工厂	A 市经济学院组织学生外出打工合理	A 市经济学院强制学生实	A 市经济学院强制学生外出	A 市经济学院体育学院变相强制
A 市经济学院寒假过年	1	0.48348972	0.41822	0.367463	0.259616

A市经济学院组织学生	0.483489722	1	0.47468	0.639264	0.294666
A市经济学院强制学生	0.41822356	0.47468758	1	0.878628	0.71172
A市经济学院强制学生	0.367463291	0.63926446	0.87862	1	0.625338
A市魅力之城油烟太烦	0.000615898	0.00066644	0.08172	0.075337	0.07642

注：完整数据见文件 sim_M1.csv

从表 2-4 中可以看出，表格前 4 行 4 列描述的都是经济学院强制学生实习的问题，这四条留言的语义是相近的，相似度也较高。而表格 2-4 第 5 条留言描述的是魅力之城噪音扰民的问题，与前 4 个留言语义不相关，因此相似度也很低。通过进一步对比也可以发现，整个 4376×4376 的矩阵中，语义不相关的留言相似度通常达不到 0.1。这说明相似度的计算流程是有效果的。

(3)计算广度指标

计算出相似度之后，按照 2.2 节的分析，针对每一条留言把与之相似度大于 0.1 的相似度全部求和，作为留言热度的影响广度指标，结果见表 2-5。

表 2-5 影响广度指标

留言编号	影响广度	留言主题
218709	13.77691	A市伊景园滨河苑捆绑销售车位
196264	12.95729	投诉A市伊景园滨河苑捆绑车位销售
223247	12.95729	投诉A市伊景园滨河苑捆绑销售车位
230554	12.95729	投诉A市伊景园滨河苑捆绑车位销售
283879	10.86048	A市伊景园滨河苑项目捆绑销售车位
244243	10.85825	关于伊景园滨河苑捆绑销售车位的投诉
251844	10.28863	投诉伊景园滨河苑项目违法捆绑车位销售
258037	10.09166	投诉伊景园滨河苑捆绑销售车位问题
205277	9.03702	伊景园滨河苑捆绑车位销售合法吗？！
264944	8.180631	A2区丽发新城附近修建搅拌厂噪音、灰尘污染
246598	7.790138	A5区劳动东路魅力之城小区临街门面烧烤夜宵摊
360103	7.790138	A5区劳动东路魅力之城小区临街门面烧烤夜宵摊
266213	6.856887	咨询A3区西湖街道茶场村五组的拆迁规划
276460	6.845273	A市伊景园滨河苑捆绑销售车位是否合理？
189739	6.51348	请问A3区西湖街道茶场村五组是如何规划的
214447	6.51348	A3区西湖街道茶场村五组是如何规划的？

注：完整数据见文件 topichotness1.csv

从表 2-5 中可以看到，这种影响广度指标的计算方式可以使语义相近的留言主题排序在一起。影响广度指标越大的相似留言出现的通常越多。计算好影响广度指标之后，需要计算影响深度指标，从而计算出综合的留言热度。

(4)计算深度指标、热度指标

按照 2.2 节的分析，影响的深度和留言点赞数有关，本文用留言点赞数的数量级再加 1 作为留言深度指标。去数量级的目的是缩小使深度指标取值的差距，避免一定的盲目性。举个例子，点赞数为 30 的留言所反应的问题通常情况不会是点赞数为 1 的留言所反映问题严重程度的 30 倍，取数量级再之后是后者是前者严重程度的 2 倍，更加符合逻辑常理。进一步分析发现，热度指标是一个综合指标，可以认为是问题在深度和广度两个不同层面上的共同反应，也可以看作是沿着问题的广度延申，将问题的严重程度发大。

因此可以直接采用广度指标 × 深度指标作为留言的热度指标，理解为根据严重程度被放大的留言频率。各留言的热度指标如表 2-6 所示。

表 2-6 留言热度表

针对留言的热度排名	热度	留言主题
89	4.126997471	反映 A 市地铁 3 号线松雅西地省站地下通道建设
90	4.098442554	A 市提升规划建设水平带动经济发展
91	4.098442554	给 A 市提升规划建设水平带动经济发展的些许建议
92	4.035616934	反映 A 市人才租房购房补贴问题
93	4.015935481	咨询 A 市人才购房及购房补贴实施办法等相关问题
94	4	举报 A 市 A3 区柏家塘小区私营 ktv 严重扰民
95	3.95	A 市 A5 区汇金路五矿万境 K9 县存在一系列问题

经过计算之后，点赞数最高的留言目前排名 95，可以算作是热度比较普通的留言。如果只考虑深度，该留言是当之无愧的第一名，如果只考虑广度，该留言约为 1210 名。综合考虑两个方面的时候使之排名提高了 1115 名。单独考虑的时候差别大的原因主要是数据集不够大，如果数据集足够大，那么高点赞数留言的相似留言不会只有那么寥寥几条，深度和广度是相辅相成的。

4.2.3 统计热点问题

将每个地点中的留言按照热度降序排列，选取第一条留言提取问题主要内容作为该地点的热点问题，用某个地点中最热门的一条留言来反应该地点的热点问题。然后，对每个地点的留言热度求和作为热点问题的热度指数，统计留言的最早时间和最晚时间作为热点问题的时间范围，得到热点问题表，见表 2-7。

表 2-7 热点问题表（前 5）

问题	热度指数	时间范围	人群地点	问题描述

ID				
1	173.5284	2019-07-28 13:09:08 至 2019/8/7 19:52:14	伊景园滨河苑	捆绑车位销售合法吗？！
2	40.67357	2019/8/12 13:15:05 至 2019/8/9 16:47:36	A市经开区泉星公园	建议A市经开区泉星公园项目规划进一步优化
3	38.60788	2019-07-21 10:29:36 至 2019/08/01 16:20:02	劳动东路魅力之城小区	魅力之城小区油烟扰民
4	37.95947	2019-11-15 08:56:46 至 2020-01-26 19:47:11	丽发新城小区	小区附近的搅拌站噪音严重扰民
5	34.77013	2019/1/6 20:36:34 至 2019/9/12 8:30:47	西湖街道茶场村五组	咨询A3区街道茶场村五组的拆迁规划

注：完整数据见文件 form1.csv

然后根据地点将表 2-2 和表 2-7 连接，统计出热点问题明细表，见表 2-8。

表 2-8 热点留言明细表

问题ID	留言编号	留言用户	留言主题	留言时间	留言详情	点赞数	反对数
1	234633	A909194	无视消费者权益的A市伊景园滨河苑车	2019/8/20 12:34	伊景园滨河苑项目商品房，广铁集团丁	0	0
1	286304	A909196	无视职工意愿、职工权益的A市伊景园	2019/8/23 10:23	广铁集团与A市政府及A市政工程有限公	0	0
1	209571	A909200	伊景园滨河苑项目绑定车位出售是否合	2019/8/28 19:32	广铁集团铁路职工定向商品房伊景园滨	0	0
1	239032	A909169	请维护铁路职工权益取消伊景园滨河苑	2019/9/1 10:03	广铁集团和A市政开发有限公司协商，	0	0
1	244528	A909235	伊景园滨河苑开发商强买强卖！	2019/8/21 19:05	A市广铁集团伊景园滨河苑商品房本来	0	0
1	258386	A909185	A市伊景园滨河苑欺压百姓	2019/8/28 0:00	A市伊景园·滨河苑强制要求购房者捆	0	0
1	268626	A909186	A市伊景园滨河苑坑害购房者	2019/8/10 12:31	A市伊景园·滨河苑违规涨价18.5万元	0	0
1	255507	A909195	违反自由买卖的A市伊景园滨河苑车位	2019/8/20 12:34	广铁集团铁路职工定向商品房伊景园滨	0	0
2	238692	A00080342	建议A市经开区泉星公园项目规划进一	2019/8/12 13:15	目前A市经济技术开发集团有限公司	0	0
2	278545	A00036841	给A市经开区泉星公园项目规划进一步	2019/8/26 13:00	目前A市经济技术开发集团有限公司的	0	0
2	278281	A00032698	建议A市经开区泉星公园项目规划进一	2019/8/22 13:23	目前A市经济技术开发集团有限公司泉	0	0
2	289574	A00080342	对A市经开区泉星公园项目规划再进一	2019/8/16 13:33	目前A市经济技术开发集团有限公司的	0	0
2	226408	A00080342	A市经开区泉星公园项目规划需优化	2019/8/9 16:47	目前A市经济技术开发集团有限公司的	0	0
3	236798	A00039089	A5区劳动东路魅力之城小区油烟扰民	2019/7/28 12:49	尊敬的政府：A5区劳动东路魅力之城小	0	0
3	360101	A324156	A5区劳动东路魅力之城小区油烟扰民	2019/7/28 12:49	尊敬的政府：A5区劳动东路魅力之城小	4	0
3	284147	A909113	A5区劳动东路魅力之城小区一楼的夜宵	2019/7/21 10:29	局长：　你好，A5区劳动东路魅力之	0	0
3	360107	A0283523	A5区劳动东路魅力之城小区一楼的夜宵	2019/7/21 10:29	局长：　你好，A5区劳动东路魅力之	3	0
3	272122	A909113	A5区劳动东路魅力之城小区一楼的夜宵	2019/8/1 16:20	局长：　你好，A5区劳动东路魅力之	0	0
3	360108	A0283523	A5区劳动东路魅力之城小区一楼的夜宵	2019/8/1 16:20	局长：　你好，A5区劳动东路魅力之	6	0

如表 2-8 所示，同一个问题 ID 中的留言主题是类似的，不会发生判断错误的情况，但是不排除有漏掉一些留言这种情况。这部分留言是因为命名实体识别对地点的识别不准确造成的，当然和留言用户的表达也有很大关系，这些因素对挖掘主要热点问题是否有影响、影响有多大，都是亟待解决和改进的方向。

4.3 问题 3 分析方法与过程

4.3.1 相关性评价

对于答复意见的相关性评价，考虑答复意见的内容是否与问题相关，可通过对比两条文本相似度量化。

将留言主题和留言详情相加，结合成完整的留言，然后对答复意见和留言进行相似度计算，先计算出词频（TF 值），然后对所得的词频向量计算余弦相似

度。相似度越高的两句话的相关性也就越大。两个向量间的余弦值可以通过使用欧几里得点积公式求出：

$$a \cdot b = \|a\|\|b\| \cos\theta \quad (30)$$

给定两个属性向量，A 和 B，其余弦相似性θ由点积和向量长度给出，如下所示：

$$similarity = \cos\theta = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} * \sqrt{\sum_{i=1}^{n} (B_i)^2}} \quad (31)$$

这里的A_i,B_i分别代表向量 A 和 B 的各分量。相似度结果展示：

留言编号	留言和答复意见的相似度
28611	0.5437
28612	0.343
28613	0.5058
28614	0.3013
28615	0.6019

图 3-1 相似度结果

图 3-1 中相似度是指结合后的留言和答复意见的余弦相似度。

4.3.2 完整性评价

对留言的答复意见，从答复的完整性来评价答复意见的质量，可通过提取留言详情和答复意见的每条数据的关键词进行匹配来衡量。如果答复意见中提取出的关键词与留言详情中提取出的关键词相匹配，匹配数量越多的，表明答复意见中对留言详情的回复内容越完整。用两条数据中匹配上的关键词总数/留言详情中的关键词总数，得到关键词复现率，以复现率来衡量完整性。

（1）Tf-idf 进行关键词提取

Tf-idf 是用一种统计学的方法来衡量一个词语在文本中的重要程度，常被用于信息提取、文本挖掘等场景中。该算法的核心便是计算出一个文本中的某个词语的 tf 值与 idf 值。从留言详情、留言标题中提取的关键词分别见图 3-2、3-3。

留言编号	答复意见中的关键词
28611	['物业公司','小区','投票','业主','业委会','高昂',''
28612	['几下','一段','潇楚','挖机','这路','一个圈','来台
28613	['幼儿园','教师','民营','惠普型','有何','更是','工
28614	['公寓','研究生','落户','新政','尊敬','购房','请问
28615	['马坡岭','小学','白住坡','路口','更名','取消','保

图 3-2 留言详情提取的关键词示例

留言编号	答复意见中的关键词
28611	['业主大会','业委会','业主','停车','胡华恒','区景
28612	['施工','平塘','排水','换填','土方','管线','集镇']
28613	['民办','幼儿园','待遇','教师','学前教育','教职工'
28614	['购房','房屋交易','补贴','管理中心','首次','公寓'
28615	['马坡岭','小学','公交站点','白住坡','站名','公交

图 3-3 答复意见提取的关键词示例

(2)关键词匹配统计与复现率计算

$$复现率=\frac{key_{both}}{key_{total}} \tag{32}$$

其中 key_{both}：表示答复中的关键词在留言中的个数。 key_{both}：表示留言详情中的总关键词个数。统计结果见图 3-4：

留言编号	复现词个数	复现率
28611	6	0.3
28612	1	0.05
28613	4	0.2
28614	5	0.33
28615	6	0.85

图 3-4 复现词统计个数与复现率

4.3.3 可解释性评价

对于留言的答复意见，从答复意见的可解释性角度来评价答复意见的质量，可解释性认为答复意见中应给出内容的相关解释。

可以通过统计每条答复意见里的引入的文献资料数目，以文献资料数目来判定答复是否合情合理，具有理论依据。文献资料数目越多，可以认为该答复意见的可解释性越强。

留言编号	文献数目
28611	1
28612	0
28613	1
28614	2
28615	0

图 3-5 文献统计结果

4.3.4 建立主成分分析模型

在前三步计算中，已经将答复意见分别从相关性、完整性、可解释性等角度对答复意见的质量进行衡量，并已将三个指标量化，现在进行 PCA（主成分分析），得出每一行量化指标的降维结果，主成分因子得分。得到一个综合评价指标，来

评价答复意见的质量。

主成分分析（Principal Component Analysis，PCA）， 是一种统计方法。通过正交变换将一组可能存在相关性的变量转换为一组线性不相关的变量，转换后的这组变量叫主成分。

主成分分析的原理可以看成原来变量重新组合成一组新的相互无关的几个综合变量，同时根据实际需要从中可以取出几个较少的总和变量尽可能多地反映原来变量的信息的统计方法叫做主成分分析或称主分量分析，也是数学上处理降维的一种方法。

主成分分析是设法将原来众多具有一定相关性（比如 p 个指标），重新组合成一组新的互相无关的综合指标来代替原来的指标。通常数学上的处理就是将原来 p 个指标作线性组合，作为新的综合指标。

最经典的做法就是用 $F1$（选取的第一个线性组合，即第一个综合指标）的方差来表达，即 $Va(rF1)$越大，表示 $F1$ 包含的信息越多。因此在所有的线性组合中选取的 $F1$ 应该是方差最大的，故称 $F1$ 为第一主成分。如果第一主成分不足以代表原来 p 个指标的信息，再考虑选取 $F2$ 即选第二个线性组合，为了有效地反映原来信息，$F1$ 已有的信息就不需要再出现再 $F1$ 中，用数学语言表达就是要求 $Cov(F1,F2) = 0$，则称 $F2$ 为第二主成分，依此类推可以构造出第三、第四，……，第 p 个主成分。

经过拟合后得到的主成分模型中的主成分公式为：

$$Y = 0.028X_1 + 0.01X_2 + 0.99X_3$$

将数据代入主成分公式后，得到各答复意见的答复得分，见图 3-6。

留言编号	答复得分
28611	0.399
28612	-0.6
28613	0.39
28614	1.39
28615	-0.59

图 3-6 答复得分

答复得分取值可正可负，其取值越大表示针对该留言的答复意见的相关性、完整性、可解释性总体上更佳。

5 总结

本文的主要工作是利用自然语言处理技术以及机器学习模型来处理“智慧政务”中的文本挖掘应用问题。包含对群众留言分类、热点问题挖掘、对答复意见进行评价等问题的解决。

对于留言分类，主要运用了自然语言处理技术，来对文本型的数据进行预处理，将非结构化数据转换为能用于分类模型的结构化数据。再利用训练数据训练分类模型，用测试数据测试模型，得到86%的准确率。可改进之处是，可利用交叉验证模型，对模型的参数进行多次调整，筛选，会得到更好的预测效果。

对于热点挖掘问题，根据留言主题，利用 FoolNLtk 提取出留言所涉及的地点、人群，并以此将所有留言分组后按照留言数量对地点/人群进行降序排序。然后建立针对留言的留言热度指标：经过 jieba 提取留言主题关键词，并求得某条留言与所有留言相似度之和，将结果与留言点赞数的数量级加权相乘作为留言热度。最后将每个地点中的留言按热度降序排列，选取一条留言，提取主要内容作为该地点的热点问题。对每个地点的留言热度求和作为热点问题的热度指数，统计留言的最早时间和最晚时间作为热点问题的时间范围。提取出热点问题之后，根据留言编号将所有留言纳入相应的热点问题，得出热点问题明细表。可改进之处是，对于有漏掉一些留言这种情况，这部分留言是因为命名实体识别对地点的识别不准确以及留言用户的表达多样造成的，这些因素对挖掘主要热点问题是否有影响、影响有多大，是亟待解决和改进的方向。

对于答复意见的质量评价问题，从答复的相关性、完整性、可解释性等角度来评价，将三个指标分别根据相似度、关键词复现率、法律法规参考数量来量化，最后运用一个主成分分析模型，将三个指标因素降维为一个主成分，计算出主成分因子得分，即为答复意见的质量评价准则，得分越高的，答复意见的质量越好的可能性越大。待改进之处是，法律法规总数的提取只要是按照出现《》号的法规内容的提取，但一些非正式的如“按**登记条件第几条规定”等，由于表达的形式过于多样化，提取的难度较大，所以造成部分可能具有可解释性的回复内容的漏掉。这是亟待解决和改进的地方。

结合自然语言处理技术和机器学习模型来通过电子设备批量处理政务，将会给政府的管理水平和施政效率的提升带来极大推动。

参考文献

[1] 张良均,王路,谭立云,苏剑林.Python 数据分析与挖掘实战.北京：机械工业出版社，2015.12
[2] 王丽坤,王宏,陆玉昌.文本挖掘及其关键技术与方法[J].计算机科学,2002,29(12):12-19.
[3] 黄晓斌,赵超.文本挖掘在网络舆情信息分析中的应用[J].情报科学,2009(1):94-99.
[4] 拓凯渊.基于文本挖掘的线上旅游评论分析.西部皮革,2020,42(2):121-123.
[5] 刘璟. 中文命名实体识别方法研究[J]. 电脑知识与技术, 2019, 15(09):185-186.
[6] 柏兵, 侯霞, 石松. 基于 CRF 和 BI-LSTM 的命名实体识别方法[J]. 北京信息科技大学学报:自然科学版, 2018, 33(06):30-36.
[7] 秦欢, 门业堃, 于钊,等. 基于隐马尔科夫和主成分分析的电网数据词典构建[J]. 电力大数据, 2019, 22(01):22-27.
[8] 许也, 申柏希, 徐翔,等. 基于条件随机场的非规范化中文地址解析方法[J]. 地理与地理信息科学, 2019, 35(02):18-24.
[9] 张兆晨, 冀俊忠. 基于循环神经网络的时序 fMRI 数据分类方法研究[J]. 小型微型计算机系统, 2018, 39(07):52-56.
[10] 刘宇鹏,栗冬冬.基于 BLSTM-CNN-CRF 的中文命名实体识别方法[J/OL].哈尔滨理工大学学报,2020(01):115-120.
[11] 黄春梅,王松磊.基于词袋模型和 TF-IDF 的短文本分类研究[J].软件工程,2020,23(03):1-3.
[12] 张响亮, 王伟, 管晓宏. 基于隐马尔可夫模型的程序行为异常检测[J]. 西安交通大学学报, 2005(10):22-25.
[13] 曾佐祺, 李赞. 基于 Viterbi 算法的 GMSK 信号解调性能分析与仿真[J]. 重庆邮电大学学报:自然科学版, 2008, 20(2):132-138.
[14] 吴桂玲. 基于欧氏距离和余弦相似度特征选择的入侵检测模型[J]. 中小企业管理与科技旬刊, 2010, 2010(2):231-232.
[15] 林海明,杜子芳.主成分分析综合评价应该注意的问题[J].统计研究,2013,30(08):25-31

附录

I. 问题一程序代码

（1）文本数据预处理

```
import pandas as pd
import jieba


def data_process(file='附件2（关于留言内容的一级标签分类模型）.xlsx'):
    data = pd.read_excel(file, header=0)
    new_data = data[['一级标签', '留言主题', ]]

    jieba.load_userdict('userdic1.txt')
    data_cut = new_data['留言主题'].apply(lambda x: jieba.lcut(x))

    stopWords = pd.read_csv('stopwords.txt', sep='stopword', header=None)
    stopWords = ['区', '镇', '乡', '县', '市', ' ', '  ', ',', '，', '、', '（', '）', '！', '？', '-', '"', '"', ':', '：',
                 ' '] + list(stopWords.iloc[:, 0])
    data_after_stop = data_cut.apply(lambda x: [i for i in x if i not in stopWords])
    labels = new_data.loc[data_after_stop.index, '一级标签']
    adata = data_after_stop.apply(lambda x: ' '.join(x))

    return adata, data_after_stop, labels
```

（2）补充贝叶斯模型进行分类

```
from data_process import data_process
from sklearn.externals import joblib
from sklearn.naive_bayes import ComplementNB
from sklearn.metrics import f1_score
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer

adata, data_after_stop, labels = data_process()
data_train, data_test, labels_train, labels_test = train_test_split(adata, labels, test_size=0.2)

countVectorizer = CountVectorizer()
data_train = countVectorizer.fit_transform(data_train)
X_train = TfidfTransformer().fit_transform(data_train.toarray()).toarray()

vocabulary = countVectorizer.vocabulary_
data_test = CountVectorizer(vocabulary=vocabulary).fit_transform(data_test)
X_test = TfidfTransformer().fit_transform(data_test.toarray()).toarray()

model = ComplementNB()
model.fit(X_train, labels_train)
print("f_score: " ,f1_score(labels_test, model.predict(X_test),average='macro'))

joblib.dump(model,'ComplementNB_model.model')
```

（3）词云图绘制

```
from data_process import data_process
from wordcloud import WordCloud
import matplotlib.pyplot as plt

adata, data_after_stop, labels = data_process()

word_fre = {}
for i in data_after_stop[labels == '环境保护']:
    for j in i:
        if j not in word_fre.keys():
            word_fre[j] = 1
        else:
            word_fre[j] += 1

mask = plt.imread('duihuakuan.jpg')
wc = WordCloud(mask=mask, background_color='white', font_path=r'C:\Windows\Fonts\simhei.ttf')
wc.fit_words(word_fre)
plt.imshow(wc)
plt.savefig('环境保护.png')
```

II. 问题二程序代码

（1）计算相似度矩阵和留言的热度

```
import jieba
import jieba.analyse
import jieba.posseg as pseg
from gensim.models import word2vec
import csv
import gensim
import numpy as np
from gensim import corpora,similarities,models


def get_topics(fujian2 = "附件 3.csv"):#提取所有留言主题
    topics=[]
    with open(fujian2) as f:
        reader = csv.reader(f)
        next(reader)
        for i in reader:
            topics.append(i[2])
    return topics
def get_keyword( sentence=""):#针对一个句子或者文本
    keywords = jieba.analyse.extract_tags(sentence, topK=10)# 提取关键词
    return keywords
def build_vocabulary(fujian2="附件 3.csv"):#建立语料库
    topics = get_topics(fujian2)
    for i in topics:
        keywords = get_keyword(i)
        word_list.append(keywords)
    dictionary = corpora.Dictionary(word_list)
    return dictionary
def get_id(fujian2 = "附件 3.csv"):
    ids = []
    with open(fujian2) as f:
        reader = csv.reader(f)
        next(reader)
        for i in reader:
            ids.append(i[0])
    return ids
def get_time(fujian2 = "附件 3.csv"):
    times=[]
    with open(fujian2) as f:
        reader = csv.reader(f)
        next(reader)
        for i in reader:
            times.append(i[3])
```

```
    return times
def get_deepth(fujian2 = "附件 3.csv"):
    jis = []
    zans=[]
    with open(fujian2) as f:
        reader = csv.reader(f)
        next(reader)
        for i in reader:
            zan = float(i[6])
            zans.append(float(zan))
            if zan>=1000:
                jis.append(4)
            elif zan>=100:
                jis.append(3)
            elif zan >= 10:
                jis.append(2)
            else:
                jis.append(1)
    return jis
word_list=[]
dic = build_vocabulary()
corpus = [dic.doc2bow(words) for words in word_list]
tfidf = models.TfidfModel(corpus)
topics = get_topics()
ids = get_id()
jis = get_deepth()
times =get_time()
featurenum=len(dic.token2id.keys())
index=similarities.SparseMatrixSimilarity(tfidf[corpus],num_features=feature
num)

sum_list= []
with open("sim_M1.csv",'w',newline="") as f:
    indexx = 0
    for i in topics:
        list1 = get_keyword(i)
        test_corpus = dic.doc2bow(list1)
        sim = index[tfidf[test_corpus]]
        list2 = sim.tolist()
        list3 = [0]
        for j in list2:
            if j>0.5:
                list3.append(j)
        sum_list.append(sum(list3)*jis[indexx])
```

```
            f.write(str(list2).strip('[').strip(']'))
            f.write("\n")
            indexx = indexx + 1

hot_data=[]
with open("topichotness1.csv",'w',newline="") as f:
    sum_list1 = sorted(enumerate(sum_list), key=lambda x: x[1],reverse=True)


    for i in sum_list1:
        f.write(str(ids[int(i[0])]))
        f.write(",")
        f.write(str(i[1]))
        f.write(",")
        f.write(str(times[int(i[0])]))
        f.write(",")
        f.write(str(topics[int(i[0])]))
        f.write("\n")
        fittopic = topics[int(i[0])]
        fitid = ids[int(i[0])]
        fittime = times[int(i[0])]
        place="".join(jieba.analyse.extract_tags(sentence=fittopic,topK=10))


        hot_data.append((fitid,i[1],fittime,fittopic,place))
```

(2) 命名实体识别提取地点

```
import fool
import csv


def foolfunc(s=""):
    res = fool.analysis(s)
    list1 = res[0][0]
    try:
        list2 = res[1][0][0]
        LocationPeaple=list2[3]
    except:
        LocationPeaple=""
    Question=""
    for (i,j) in list1:
        if('ns' in str(j) ):
            pass;
        else:
            Question = Question+i
```

```
    return(LocationPeaple,Question)

data = []
with open("topichotness1.csv",'r',newline="") as f:
    reader = csv.reader(f)
    for i in reader:
        sub_data = []
        sub_data.append(i[0])
        sub_data.append(i[1])
        sub_data.append(i[2])
        sub_data.append(i[3])
        sub_data.append(foolfunc(i[3])[0])
        sub_data.append(foolfunc(i[3])[1])
        data.append(sub_data)
with open("TopichotnessWithLocation.csv",'w',newline="") as f:
    writer = csv.writer(f)
    for i in data:
        writer.writerow(i)
```

(3) 统计热点问题

```
import csv
import numpy as np
hot_location={}

import numpy as np


places=[]
with open("TopichotnessWithLocation.csv",'r',newline="") as f:
    reader = csv.reader(f)
    for i in reader:
        id = i[0]
        hotness=i[1]
        time = i[2]
        title = i[3]
        place=i[4]
        des = i[5]
        index = 0
        if place not in places:
            places.append(place)
    dict1={}
    for i in  places:
        dict1[i]=[]
```

```
25.
26. with open("TopichotnessWithLocation.csv", 'r') as f:
27.     for key in dict1:
28.         f.seek(0)
29.         reader = csv.reader(f)
30.         for i in reader:
31.             if i[4]==key:
32.                 dict1[key].append(i)
33. dict2 =sorted(dict1.items(),key = lambda item:len(item[1]),reverse=True)
34. form1=[]
35. with open("form1.csv",'w',newline="") as f:
36.     writer = csv.writer(f)
37.     list3=[]
38.     for i in dict2:
39.         place=i[0]
40.         lists = i[1]
41.         try:
42.             question = sorted(lists,key=lambda x:x[1],reverse=True)[0][5]
43.         except:
44.             question = sorted(lists, key=lambda x: x[1], reverse=True)[0][5]

45.         SUM=0
46.         maxtime = ""
47.         mintime = "2020/10/6 21:35:50"
48.         for j in lists:
49.             SUM=SUM+float(j[1])
50.             if j[2]>maxtime:
51.                 maxtime=j[2]
52.             if j[2]<mintime:
53.                 mintime=j[2]
54.         if len(place)>4:
55.             list3.append(([SUM,("".join(mintime)+" 至 "+str(maxtime)),place,
    question]))
56.     list3 = sorted(list3,key=lambda x:x[0],reverse=True)
57.     writer.writerow(["问题 ID","热度指数","时间范围","人群地点","问题描述"])
58.     Qid=[1]
59.     for line in list3:
60.         writer.writerow((Qid+line))
61.         form1.append((Qid+line))
62.         Qid[0]=Qid[0]+1
63. list5=[]
64. for i in form1:
65.     list4 = dict1[i[3]]
66.     for j in list4:
```

```
        list5.append([i[0],j[0],j[3],j[2]])
list6=[]
with open("附件 3.csv",'r') as f:
    reader = csv.reader(f)
    sub_list=[]
    for i in reader:
        sub_list.append(i)
    for i in list5:
        Qid=i[0]
        Jid=i[1]
        User = ""
        Title =i[2]
        Time=i[3]
        Detail=""
        Up=0
        Down=0
        for j in sub_list:
            if j[0]==Jid:
                User = j[1]
                Detail=j[4]
                Up=j[5]
                Dowm=j[6]
        list6.append([Qid,Jid,User,Title,Time,Detail,Up,Down])
with open("form2.csv",'w',newline="") as f:
    writer=csv.writer(f)
    writer.writerow(['问题 ID','留言编号','留言用户','留言主题','留言时间','留言详情','点赞数','反对数'])
    for i in list6:
        writer.writerow(i)
```

III. 问题三程序代码

（1）相似度计算

```python
import re
import numpy as np
from scipy.linalg import norm
from sklearn.feature_extraction.text import TfidfVectorizer
import pandas as pd

def tf_similarity(s1, s2):
    def add_space(s):
        return ' '.join(list(s))

    # 将字中间加入空格
    s1, s2 = add_space(s1), add_space(s2)
    # 转化为TF矩阵
    cv = TfidfVectorizer(tokenizer=lambda s: s.split())
    corpus = [s1, s2]
    vectors = cv.fit_transform(corpus).toarray()
    # 计算cos
    return np.dot(vectors[0], vectors[1]) / (norm(vectors[0]) * norm(vectors[1]))

data = pd.read_excel('./附件4（留言及答复意见）.xlsx', header=0)
data['留言详情'] = data['留言详情'].apply(lambda x: re.sub('\n\t\t\t\t\t\n\t\t\t\t\t', '', x)).apply(
    lambda x: re.sub('\n\t\t\t\t', '', x))
data['留言'] = data['留言主题'] + data['留言详情']

adata = []
adata_2 = []
#相似度
for i in range(len(data)):
    adata.append(tf_similarity(data['留言'][i], data['答复意见'][i]))
#文献数目
for i in range(len(data)):
    adata_2.append(data['答复意见'][i].count('《'))
data['相似度'] = adata
data['文献数目'] = adata_2

data.to_csv('data.csv', encoding="utf_8_sig")
```

（2）完整度（复现率）计算

```python
import pandas as pd
import jieba
from jieba import analyse
import csv
import re
import numpy as np

data = pd.read_excel('H:/其他/泰迪杯/C题全部数据/附件4（留言及答复意见）.xlsx', header=0)
mind = data['留言主题']
answer = data['答复意见']
content = data['留言详情']

def word_deal(words):
    words = words.dropna().apply(lambda x: re.sub('\n', '', x))
    words = words.dropna().apply(lambda x: re.sub('\t', '', x))
    words = words.apply(lambda x: re.sub('^[A-Z]+[0-9]*"', '', x))
    words = words.apply(lambda x: re.sub('[0-9]*', '', x))
    words_cut = words.apply(lambda x:jieba.lcut(x)) # 分词
    stopWords = pd.read_csv('H:/其他/泰迪杯/C题全部数据/stopword.txt', encoding='utf-8', sep='hh', header=None)
    stopWords = ['？', '！', '"', '"', '、', '（', '的', '?', '了', '请','网友','市民','同志','年月日', '该项'] + list(stopWords.iloc[:, 0])
    words_afterStop = words_cut.apply(lambda x: [i for i in x if i not in stopWords])
    words_join = words_afterStop.apply(lambda x: ' '.join(x))
    return words_join

def tfidf_word(text):
    tfidf = analyse.extract_tags
    text_key = text.apply(lambda x: tfidf(x))
    return text_key

mind_key = tfidf_word(word_deal(mind))
content_key = tfidf_word(word_deal(content))
answer_key = tfidf_word(word_deal(answer))
```

```
def reply_count_rate(content, answer):
    countList = {}
    count = 0
    rate = 0.0
    for x in content:
        # 在content_key中匹配关键词
        nums = len(re.findall(x, str(answer)))
        if nums:
            countList[x] = nums
        count = len(countList)
        rate = count/len(content)
    return count,rate

countL = {}
rateL = {}
for i in range(len(content)):
    countL[i] = reply_count_rate(content_key[i], answer_key[i])[0]
    rateL[i] = reply_count_rate(content_key[i], answer_key[i])[1]

with open('H:/其他/泰迪杯/C题全部数据/keyword_rate.csv', 'w') as f:
    [f.write('{0},{1}\n'.format(key, value)) for key, value in rateL.items()]
```

（3）主成分分析

```
import pandas as pd
from sklearn.externals import joblib
from sklearn.decomposition import PCA

#参数初始化
data1 = pd.read_csv('./keyword_count.csv', header=0)
data2 = pd.read_csv('./data.csv', header=0)

data = pd.concat([data2['相似度'],data1['key_rate'],data2['文献数目']],axis=1)

pca = PCA(1)  #选取累计贡献率大于95%的主成分（1个主成分）
pca.fit(data)
print(pca.components_) #返回模型的各个特征向量
print(pca.explained_variance_ratio_)#返回各个成分各自的方差百分比(也称贡献率)

joblib.dump(pca,'pca.model')
pca_load = joblib.load('pca.model')
pca_load.fit(data)
low_data = pca_load.transform(data)    #降低维度(因子得分)
data['PCA'] = low_data

data.to_csv('pca.csv', encoding="utf_8_sig")
```