“智慧政务”中的文本挖掘应用

摘要

近年来，随着微信、微博、市长信箱等网络问政平台逐步成为政府了解民意的重要渠道，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门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。

本文主要解决群众留言文本分类、热点问题挖掘以及答复意见评价三个问题。针对问题一，我们通过 jieba 分词库将群众的留言进行分词，然后将文本向量化表示，最后通过 SVM 支持向量机模型进行文本分类，最终正确率达到 89%。针对问题二，我们首先使用 LDA 主题模型，提取出群众留言中的主题词，然后进行人工合并得到有关的热点问题。针对问题三，我们利用得到的文本向量计算出相关部门对留言的答复意见与群众留言主题之间的文本相似度，然后构建答复意见的质量的评价指标，从而反映答复意见的效果。

关键词：文本分析 信息提取 LDA 模型 SVM 向量机 文本向量化

Text Mining Application in "Intelligent Government Affairs"

Abstract

In recent years, as online questioning platforms such as WeChat, Weibo, and the mayor's mailbox have gradually become an important channel for the government to understand public opinion, the amount of text data related to various social conditions and public opinion has continued to rise, giving manual message division and hot spots in the past Organizing the work of related departments has brought great challenges. At the same time, with the development of big data, cloud computing, artificial intelligence and other technologies, the establishment of a smart government system based on natural language processing technology has become a new trend in the development of social governance innovation, which has a great impact on improving the management level and efficiency of government Promote the role.

This article mainly solves the three problems of mass message text classification, hotspot problem mining, and answer evaluation. For problem one, we segment the message of the masses through the jieba word segmentation library, then vectorize the text, and finally classify the text through the SVM support vector machine model, and the final accuracy rate reaches 89%. For problem two, we first use the LDA topic model to extract the subject words in the mass message, and then manually merge to get the relevant hot issues. For question three, we use the obtained text vectors to calculate the text similarity between the relevant department's response to the message and the subject of the mass message, and then construct an evaluation index of the quality of the response to reflect the effect of the response.

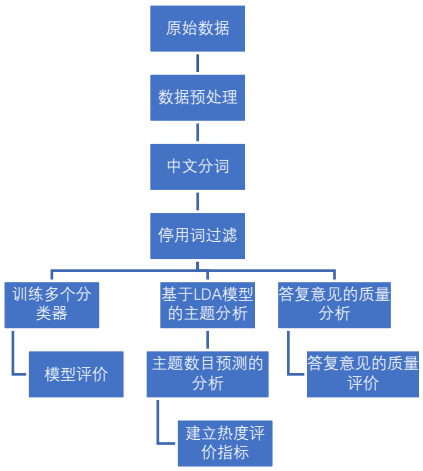
Key Words: text analysis, Stacked AutoEncoder(SAE), Latent Dirichlet Allocation(LDA), SVM(Support Vector Machines), Text vectorization

1 挖掘目标

本次建模针对收集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，在对目标文本进行基本的探索分析、预处理、中文分词和停用词过滤后，通过文本向量化建立词袋模型提取 TF-IDF 特征，然后基于各种传统机器学习方法进行文本分类的预测；以及建立 LDA 模型和主题选取的可视化，实现对群众留言的分类和热点问题挖掘以及答复评价。

2 分析方法和过程

2.1 总体流程



本论文的分析流程课大致分为以下 n 步：

1) 对文本数据进行基本的预处理，包括结巴分词，停用词过滤和添加保留词；
2) 文本向量化表示，建立词袋模型，训练语料库；
3) 运用多种预测模型对文本进行分类和聚类；

2.2 具体步骤

2.2.1 数据探索分析

本文使用的实验数据是收集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，去重后留言数目总计 9210 条。其中关键文本项为留言详情、一级标签和答复意见，如表 1 所示：

表 1 留言数据分布表

城乡建设	2009
劳动和社会保障	1969
教育文体	1589
商贸旅游	1215
环境保护	938
卫生计生	877

交通运输	613
Name:一级标签, dtype:int64	

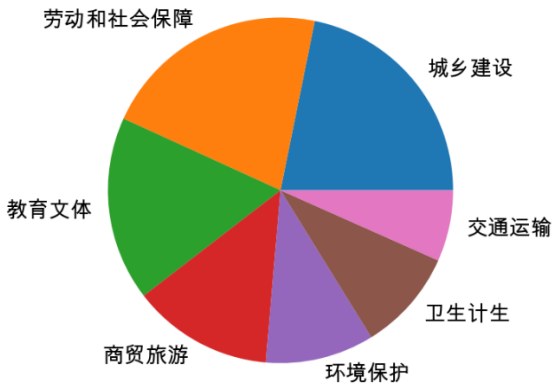


图 1 数据分布饼状图

2.2.2 留言文本的预处理

取到留言文本后，我们首先对留言数据进行简单的预处理。留言内容中存在大量含义价值低的语气词、助词和礼貌用词等，若将这些词语业带入进行分词、词频统计乃至语义提取等，会徒增计算量且对分析结果造成极大的噪声影响，得到的结果质量也必然存在隐患。因此，在运用这些留言文本数据之前必须要先进行预处理，把大量无用重复的文本去除。

我们运用 python3.7 对这些文本数据的预处理主要由两个步骤组成：文本去重、去除非汉字字符。

预处理后的留言文本避免了同一个人可能出现的重复留言，并且保留纯中文的文本数据以便后期的分词处理。

2.2.3 中文分词

在汉字语言中，只有字、句和段落有明显的分界符来进行基础的划界，相比“句子”而言，“词”和“词组”的边界模糊，没有一个形式上的分界符号，甚至在不同语境中存在不同的分词情况。因此，进行中文的文本挖掘时，首先应该对文本进行合适的分词，即将一串连续的字符串按照一定形式的规范，重新组合成电脑可识别的词序列的过程。

分词的好坏对后续的文本挖掘算法有这确定性的影响，若分词效果不理想，即使后续的模型或算法再优秀也无法达到满意的效果。

本文采用目前最主流且做最好的 Python 中文分词组件——“结巴”分词库，我们选择结巴库中默认的精准模式对附件 2 中的留言详情数据进行中文分词。

部分结果示例如下：

◆ 地铁 号线 施工 导致 万家 丽路 锦楚 国际 星城 三期 一个月 停电 10 来次 每次 通知 断电 居民 小孩 被困 电梯 每次 停电 十几个 小时 影响 居民 生活
◆ 尊敬 胡书记 远鑫逸园 西北边 堆放 垃圾 长期 清理 影响 周边 居民 生活质量 影响 周边 环境卫生 物业管理 渠道 每次 清理 一点 不能根除 垃圾 居民 生活 影响 影响 市容 市貌 督促 根治 垃圾场 改建 绿化带 美化环境
◆ 强烈建议 C5 市路 两侧 绿化带 加宽 米 种上 高大 树 穿行 机动车 侵占人行道 夏天 遮阳 行人 爆晒 绿化 美化 城市 造福 市民 一举多得
◆ 杨 书记 你好 关注 黄兴 大道北 延线 工程进度 民工 工资 发放 情况 中铁二局 公司 拖欠 工程款 过多 已致 全线 大面积 停工 尚未 支付 民工 工资 材料 款令 包工头 苦不堪言

2.2.4 停用词过滤

在中文分词这一步之后，将初始的文本数据处理为多个词的无序集合，即所谓的“词袋”。但文本中含有大量重复但在文本中含义表达并无意义的词，应当进行删除，以消除这些词对文本挖掘工作的不良影响，此类词称为停用词。

停用词有两个特征:其一是使用十分广泛,甚至是过于频繁,比如中文的“我”、“就”、“和”几乎在每个文档上都会出现；其二是出现频率很高，但实际意义又不大，这一类主要包括了语气助词、副词、介词、连词等，通常自身并无明确意义，只有将其放入一个完整的句子中才有一定作用的词语。如常见的“的”、“在”、“和”、“接着”之类。

文本采用基于停用词表的稳步停用词过滤方式,将我们的分词结果与停用词表中的自定义词进行匹配，若匹配成功，则进行删除处理。

结果示例如下：

区大道 西行 便道，未管所 路口 至 加油站 路段，人行道 包括 路灯杆，被 圈 西湖 建筑 集团 燕子 山 安置房 项目 施工 围墙 内 。每天 尤其 上下班 期间 这 条 路上 人流 车流 极多，安全隐患 非常 大。强烈 请求 文明城市 A 市，尽快 整改 这个 极 不文明的 路段 。

（停用词过滤前）

大道 西行 道 未管 路口 加油站 路段 人行道 包括 路灯 杆 圈 西湖 建筑 集团 燕子 山 安置房 项目 施工 围墙 上下班 期间 条 路上 人流 车流 安全隐患 文明城市 整改 不 文明 路段

（停用词过滤后）

2.2.5 保留专有名词

```
jieba.load_userdict(file_name) # file_name 为自定义词典的路径
```

结巴库支持导入自定义的词典，以便包含 jieba 词库里没有的词。虽然 jieba 有新词识别能力，但是自行添加新词可以保证更高的正确率，来增强歧义

纠错能力。部分自定义保留词示例如下：

- 安置房
- 书院路
- 火炬路
- 公共面积
- 盛世耀凯
- 拨款
- 雾霾
- 广场舞

- 中央广场
- 楚江新区
- 嘉庆路
- 五一大道
- 占道经营
- 烧烤摊
- 城管队
- 燕子山

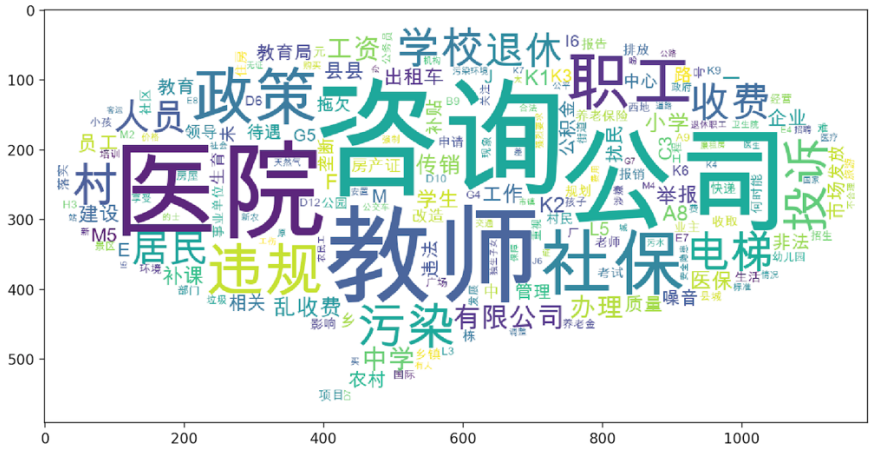


图 2 留言文本词频的词云图

2.2.6 文本特征向量化

文本表示是自然语言处理中的基础工作，文本表示的好坏直接影响到整个自然语言处理系统的性能。文本向量化就是将文本表示成一系列能够表达文本语义的向量，是文本表示的一种重要方式。目前对文本向量化大部分的研究都是通过词向量化实现的，也有一部分研究者将句子作为文本处理的基本单元，于是产生了 doc2vec 和 str2vec 技术。

词袋(Bag Of Word)模型是最早的以词语为基础处理单元的文本项量化方法。该模型产生的向量与原来文本中单词出现的顺序没有关系，而是词典中每个单词在文本中出现的频率。

本文采用 TF-IDF 技术来进行文本特征向量化就是为了克服词频统计技术的缺陷而产生的，它引入了“逆文档频率”概念，它衡量了一个词的常见程度，TF-IDF 的假设是：如果某个词或短语在一篇文章中出现的频率高，并且在其他文章中很少出，那么它很可能就反映了这篇文章的特性，因此要提高它的权值。

得词频稀疏矩阵过程如下图所示：

```
In [10]: from sklearn.feature_extraction.text import CountVectorizer, TfidfVector
         cv = CountVectorizer().fit(tmp)
         cv_data = cv.transform(tmp)
         # cv.vocabulary_
         cv_data.toarray()

Out[10]: array([[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]], dtype=int64)
```

图 3 Jupyter 运行截图

3 研究方案及实施

3.1 问题一

本题要求根据附件 2 给出的数据，建立关于留言内容的一级标签分类模型。“留言主题”相比“留言详情”语言更加简练，但不利于模型的训练，容易使结果过拟合。经思考，小组决定主要使用“留言详情”列训练多种分类器，并比较训练结果选择最优分类模型。

3.1.1 分类器模型

a) MultinomialNB 朴素贝叶斯模型

贝叶斯分类是一类分类算法的总称，这类算法均以贝叶斯定理为基础，故统称为贝叶斯分类。而朴素朴素贝叶斯分类是贝叶斯分类中最简单，也是常见的一种分类方法。

而朴素贝叶斯中的多项式模型常用于文本分类，特征是单词，值时单词的出现次数。

$$P(x_i|y_k)=\frac{N_{y_k x_i}+\alpha}{N_{y_k}+\alpha n} \quad P(x_i|y_k)=\frac{N_{y_k x_i}+\alpha}{N_{y_k}+\alpha n}$$

其中$N_{y_k x_i}$是类别y_k下特征x_i出现的总次数；N_{y_k}是类别y_k下所有特征出现的总次数。具体代码与运行结果如下：

```
In [13]: model_nb = MultinomialNB().fit(cv_train, y_train)
         model_nb.score(cv_test, y_test)

Out[13]: 0.8628302569670648
```

图 4 Jupyter 运行截图

b) K 近邻分类器

K 近邻(k-Nearest Neighbor，KNN)分类算法的核心思想是如果一个样本在特征空间中的 k 个最相似(即特征空间中最邻近)的样本中的大多数属于某一个类别，则该样本也属于这个类别。KNN 算法可用于多分类，KNN 算法不仅可以用于分类，还可以用于回归。通过找出一个样本的 k 个最近邻居，将这些邻居的属性的平均值赋给该样本，作为预测值。具体代码与运行结果如下：

```
In [14]: model_knn = KNeighborsClassifier().fit(cv_train, y_train)
         model_knn.score(cv_test, y_test)

Out[14]: 0.5595367354325009
```

图 5 Jupyter 运行截图

c) linearSVC 线性分类支持向量机

支持向量机分类器(Support Vector Classifier)是根据训练样本的分布，搜索所以可能的线性分类器中最佳的那个，决定分类边界位置的样本并不是所有训练数据，是其中的两个类别空间的间隔最小的两个不同类别的数据点，即“支持向量”。从而可以在海量甚至高维度的数据中，筛选对预测任务最为有效的少数训练样本。具体代码与运行结果如下：

```
In [15]: model_svc = LinearSVC().fit(cv_train, y_train)
         model_svc.score(cv_test, y_test)

         d:\program\python3.7.0\lib\site-packages\sklearn\svm\_base.py:947
         : ConvergenceWarning: Liblinear failed to converge, increase the
         number of iterations.
           "the number of iterations.", ConvergenceWarning)

Out[15]: 0.8726022439377489
```

图 6 Jupyter 运行截图

3.1.2 模型评估

1. MultinomialNB 朴素贝叶斯模型：

```
In [17]: y_pre_nb = model_nb.predict(cv_test)
         print(classification_report(y_true=y_test, y_pred=y_pre_nb))
         #print(confusion_matrix(y_true=y_test, y_pred=y_pre_nb))

                       precision    recall  f1-score   support

              交通运输       0.87      0.48      0.62       183
          劳动和社会保障       0.87      0.92      0.89       601
              卫生计生       0.91      0.84      0.87       251
              商贸旅游       0.89      0.77      0.82       364
              城乡建设       0.80      0.90      0.85       614
              教育文体       0.88      0.93      0.91       461
              环境保护       0.89      0.93      0.91       289

          accuracy                           0.86      2763
         macro avg       0.87      0.82      0.84      2763
      weighted avg       0.87      0.86      0.86      2763
```

图 7 Jupyter 运行截图

2. K 近邻分类器

```
In [18]: y_pre_knn = model_knn.predict(cv_test)
         print(classification_report(y_true=y_test, y_pred=y_pre_knn))
         #print(confusion_matrix(y_true=y_test, y_pred=y_pre_knn))
```

	precision	recall	f1-score	support
交通运输	0.56	0.34	0.43	183
劳动和社会保障	0.72	0.76	0.74	601
卫生计生	0.57	0.57	0.57	251
商贸旅游	0.27	0.79	0.40	364
城乡建设	0.78	0.41	0.53	614
教育文体	0.91	0.58	0.71	461
环境保护	0.94	0.27	0.42	289
accuracy			0.56	2763
macro avg	0.68	0.53	0.54	2763
weighted avg	0.70	0.56	0.57	2763

图 8 Jupyter 运行截图

3. linearSVC 线性分类支持向量机

```
SVC: 0.8870792616720955
SVC:              precision    recall  f1-score   support

        交通运输       0.85      0.80      0.82       111
    劳动和社会保障       0.90      0.93      0.92       395
        卫生计生       0.91      0.87      0.89       160
        商贸旅游       0.85      0.81      0.83       256
        城乡建设       0.86      0.89      0.88       426
        教育文体       0.92      0.94      0.93       295
        环境保护       0.91      0.87      0.89       199

avg / total       0.89      0.89      0.89      1842


Process finished with exit code 0
```

图 9 PyCharm CE 运行截图

3.2 问题二

某一时段内群众集中反映的某一问题可称为热点问题，而及时发现热点问题，有助于相关 部门进行有针对性地处理，提升服务效率。 对于无监督学习的文本主题聚类的问题，我们选择字数少意图简明的留言主题和数据量大、有利于机器学习的留言详情，进行聚类主题模型的训练。通过三种中文文本聚类方法，当各个 topic 之间的相似度的最小的时候，就算找到了相对合适的主题个数。

3.2.1 聚类主题模型

a) 基于 K-Means 的主题聚类
b) 基于 DBSCAN 的主题聚类
c) 基于 LDA 模型的主题聚类

LDA 算法的输入是一个文档的集合 D={d1, d2, d3, ... , dn}，同时还需要聚类的类别数量 m；然后会算法会将每一篇文档 di 在 所有 Topic 上的一个概率值 p；这样每篇文档都会得到一个概率的集合 di=（dp1，dp2，...，dpm)；同样的文档中的所有词也会求出 它对应每个 Topic 的概率，wi = （wp1，wp2，wp3，...，wpm)；

这样就得到了两个矩阵，一个是文档到主题，另一个则是词到主题。其公式如下：

$$p(z_i = k|\vec{\mathbf{z}}_{\neg i}, \vec{\mathbf{w}}) \propto \frac{n^{(k)}_{m,\neg i} + \alpha_k}{\sum_{k=1}^{K}(n^{(t)}_{m,\neg i} + \alpha_k)} \cdot \frac{n^{(t)}_{k,\neg i} + \beta_t}{\sum_{t=1}^{V}(n^{(t)}_{k,\neg i} + \beta_t)}$$

算法步骤：

1. 对文档集合中的每篇文档 d，做分词，并过滤掉无意义词，得到语料集合 W ={w1,w2, ..., wx}。
2. 对这些词做统计，得到 p（wi|d）。
3. 为语料集合 W 中的每个 wi ，随机指定一个主题 t，作为初始主题。
4. 通过 Gibbs Sampling 公式， 重新采样 每个 w 的所属 主题 t， 并在语料中更新 直到 Gibbs Sampling 收敛。

收敛以后得到 主题-词 的概率矩阵，这个就是 LDA 矩阵，而 文档-主题的的概率矩阵也是能得到的，统计后，就能能得到文档-主题的概率分布。运行结果如下：

```
LatentDirichletAllocationClustering
Loading model from cache /var/folders/rj/n86np0g53jxgmc4mt8cyvdm40000gp
Loading model cost 0.809 seconds.
Prefix dict has been built successfully.
Topic #0: 教师 工资 退休 拖欠 企业 县县 养老金 农民工 教育局 人员
Topic #1: 西地 标准 重视 就业 计划生育 请问 g4 教师 路面 情况
Topic #2: k2 西地 k1 开通 希望 司机 报名 中心 网上 液化气
Topic #3: 违法 建议 建筑 违规 规划 有限公司 a7 项目 居民 部门
Topic #4: 传销 小学 学校 中学 学生 违规 补课 举报 培训 老师
Topic #5: 出租车 的士 市场 价格 乱收费 城区 垄断 涨价 收费 期间
Topic #6: 社保 小孩 西地 缴纳 孩子 缴费 农村 查询 信息 m2
Topic #7: 小区 电梯 质量 房产证 业主 房屋 安全隐患 a3 开发商 故障
Topic #8: 解决 请求 员工 招聘 公司 报告 考试 养老保险 西地 污染
Topic #9: 办理 社区 社保 失业 手续 污染 i6 违建 噪音 环保局
Topic #10: 人员 西地 事业单位 工作 待遇 退休 独生子女 职工 公务员 工伤
Topic #11: a8 县市 收取 教育 教育局 垄断 k1 操作 档案 h3
Topic #12: 请问 医院 申请 c4 建议 医生 建设 廉租房 补贴 发放
Topic #13: 收费 不合理 c3 工资 加班 职工 易俗 请求 排污 k4
Topic #14: 景区 k8 医院 卫生院 旅游 工作人员 发生 建议 违规 公园
Topic #15: 投诉 补课 公司 市二中 西地 l4 县城 职工 劳动合同 虚假
Topic #16: 咨询 政策 医保 生育 报销 新农 相关 二胎 事宜 西地
Topic #17: 职称 教师 西地 工作 整治 卫生 评审 学院 西地省 m5
Topic #18: 公积金 住房 建设 贷款 建议 公交车 交通 广场 公园 规划
Topic #19: l5 k9 排放 污染 购房 养猪场 出台 乱收费 农村 班车
```

图 10 文档-主题的概率分布

4 参考文献

[1]自然语言处理—LDA 主题聚类模型
https://www.cnblogs.com/zongfa/p/9557682.html
[2] word2vec 的几种实现
https://www.jianshu.com/p/972d0db609f2
[3]文本聚类的无监督学习
https://github.com/zhangfazhan/text_clustering