智慧政务中的文本分类及热点发现

摘要

随着互联网的快速发展，短时间内从浩如烟海的信息库中挖掘有用信息成为人们的迫切需要，由此应运而生了文本数据挖掘中的中文文本分类方法，它是将统计方法与计算机学习方法结合应用到文本分类中。中文文本分类是根据文本内容的主题词等属性特性将其划分到用户根据需求定义的相应类别中，一般是通过输入文本的特征向量，得到输出结果文本分类类别。

关于文本分类问题，本文首先介绍了中文文本分类的主要流程以及朴素贝叶斯分类器的理论思想，按照理论中描述的流程用 Python 的集成环境 anaconda（Jupyter）进行编程操作。首先对文本集进行 jieba 分词和删除停用词处理，然后用频率统计方法进行特征选择、TF-IDF 进行特征加权，构造朴素贝叶斯分类器进行中文文本分类并计算得精准率为 61.7%、召回率为 63.2%、F_1 值为 62.2%。

关于热点发现问题，运用 LDA 主题聚类方法，首先使用 perplexity 来计算 LDA 生成的主题数。然后以留言文档以及主题数作为输入参数，聚类得到基于 200 个主题的类。此时这 200 个类别的热度是未知的，因此建立了热度评价模型来计算每一类的热度，该模型是通过计算每一类下全部留言的热度之和求得每一类的总热度。此时将每一类的热度进行排序，可得到热度评分排名前五的留言。

本文中对于留言答复意见的评价指标分为相关性、完整性、可解释性三个方面度量。对于相关部门对留言的答复，其相关性、完整性、可解释性的界定都是非常模糊的，并不能量化具体计算，本文运用模糊综合评价法来评价答复意见。

关键词：NLP、Python、TF-IDF、LDA 主题聚类模型、模糊综合评价法

一、 问题重述

近年来，随着网络技术的发展，线上问政平台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，依靠人工进行的留言划分和热点整理存在工作量大、效率低和差错率高的问题。因此，建立基于自然语言处理技术的智慧政务系统已经成为社会治理创新发展的迫切需要。

附件中给出了群众问政留言记录以及相关部门对部分群众留言的答复意见，本文需要解决的问题有：

1. 在处理网络问政平台的群众留言时，需要按照三级标签分类体系对留言进行分类，以便将留言分派至相应的职能部门处理。请根据附件2中的群众留言数据，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. 热点问题表示某一时间段内群众集中反映的某一问题。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。本文需要解决的问题是对留言进行归类，定义合理的热度评价指标，并给出评价结果，最终给出排名前5的热点问题。

3. 针对附件4相关部门对群众留言的答复意见，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、 问题分析

2.1 问题一的分析

互联网的快速发展带来了信息来源渠道变广、信息传播速度加快的便利，但面对拥有空前规模的信息量，在短时间内对信息分类变得十分艰难，为了满足社会治理创新发展的需要，自动文本分类技术应运而生。

对于问题一，附件 2 中的群众留言是以中文文本形式存在的，而中文文本中的分词以及词性并不能被计算机直接识别，此外，这些文本的具体内容中，还包括一些对留言分类没有任何价值的语气词、副词、介词、连接词等停用词。因此，需要对数据进行预处理：利用分词算法进行分词、删除停用词。

在文本数据挖掘中的中文文本分类方法的基础上，根据文本内容的主题词等属性将群众留言划分到对应的一级标签。

2.2 问题二的分析

问题二实质上是一个热点发现问题，由于传统的相似度计算模型仅采取词频统计来表示文本，丢失了部分语义信息，影响了相似度计算，因此采用 LDA 主题聚类模型进行建模。LDA 主题模型是利用统计学的知识，分析文档集内部信息，将集合映射到基于隐含主题的特征空间上。但是在使用 LDA 主题模型时，需要事先给定主题数，本文的解决方法是使用 perplexity 来计算。我们将 LDA 主题模型运用到热点发现中，LDA 利用文档生成过程的逆过程，即根据一篇得到的文档，去寻找出这篇文档的主题，以及这些主题所对应的词。LDA 的结果是聚类后不同主题所对应的相关词，要想得到热度排名前五的具体留言，需要计算每条留言的热度，然后结合聚类结果计算属于同一类的留言的总热度，而后计算得到排名前五的留言的具体信息。

2.3 问题三的分析

对于留言答复的评价需要从相关性、完整性和可解释性三个方面进行，但是这三个方面的界定非常模糊，因此考虑使用模糊综合评价法。

三、 符号说明

3.2 符号说明

序号	*符号*	*符号说明*
1	N	单词在某文档中的频次
2	M	文档的单词数
3	D_W	出现了该单词的文档数
4	D	表示总文档数。
5	M	测试语料库的大小
6	N_d	第 d 篇文本大小（即单词个数）

四、 问题一：群众留言分类

文本分类是文本数据挖掘的方法之一。文本挖掘是一个以半结构或者无结构的自然语言文本为对象的数据挖掘，是从大规模文本数据集中发现隐藏的、重要的、新颖的、潜在的有用的规律的过程。直观的说，当数据挖掘的对象完全由文本这种数据类型组成时，这个过程就成为文本挖掘。文本挖掘也称为文本数据挖掘。[1]

本文中的中文文本分类技术主要分为以下几个步骤：

1. 中文分词：使用 python 的第三方库“jieba”进行中文文本分词
2. 去除停用词：在文本分词的基础上，删除与主要内容无关的停用词。
3. 构建词向量空间：统计文本词频，生成文本的词向量空间。
4. TF-IDF 方法：使用 TF-IDF 发现特征词，并抽取为反映文档主题的特征。
5. 分类器：使用算法训练分类器。
6. 评价分类结果：使用 F-Score 对分类方法进行评价

在实验过程中，通常将数据集分成两部分：一部分是训练文本集，它被用来训练文本分类器，训练集不仅要确保该分类器通过交叉验证方法得到的各超参数有合适的取值，并且还要保证分类器具有一定的泛化能力；另一类是测试文本集，它被用来测试训练文本集得到的分类器的泛化性能的优劣程度，优劣程度通过准确率、召回率、F1 值等指标衡量。训练文本集占整个样本数据集的 80%，测试文本集占整个样本数据集的 20%。

4.1 文本预处理

对于英文文本，每个单词之间都会有空格区分区分每个词汇，但是中文文本都是以标点符号分隔的句子的形式呈现，并不能区分每个词汇。因此，在建立模型之前需要首先对训练和测试文本集进行文本预处理。

4.1.1 文本分词处理

中文分词是指以词作为基本单元，使用计算机自动对中文文本进行词语的切

分，这样方便计算机识别出各语句的重点内容。

通过 python 程序将附件 2 中群众留言以文本形式导入，如图 1。

	留言编号	留言用户	留言主题	留言时间	留言详情	一级标签
0	24	A00074011	A市西湖建筑集团占道施工有安全隐患	2020/1/6 12:09:38	\n\t\t\t\t\t\n\t\t\t\t\tA3区大道西行便道，未管所路口至加油站路段，...	城乡建设
1	37	U0008473	A市在水一方大厦人为烂尾多年，安全隐患严重	2020/1/4 11:17:46	\n\t\t\t\t\t\n\t\t\t\t\t位于书院路主干道的在水一方大厦一楼至四楼人为...	城乡建设
2	83	A00063999	投诉A市A1区苑物业违规收停车费	2019/12/30 17:06:14	\n\t\t\t\t\t\n\t\t\t\t\t尊敬的领导：A1区苑小区位于A1区火炬路，小...	城乡建设
3	303	U0007137	A1区蔡锷南路A2区华庭楼顶水箱长年不洗	2019/12/6 14:40:14	\n\t\t\t\t\t\n\t\t\t\t\tA1区A2区华庭小区高层为二次供水，楼顶水箱...	城乡建设
4	319	U0007137	A1区A2区华庭自来水好大一股霉味	2019/12/5 11:17:22	\n\t\t\t\t\t\n\t\t\t\t\tA1区A2区华庭小区高层为二次供水，楼顶水箱...	城乡建设

图 2 数据导入示意图

然后对其利用 python 中的第三方库“jieba”进行分词处理。“jieba”分词的实质是根据文档上下文内容以及语义进行分词处理。如下图：

```
#jieba分词
jieba.load_userdict('./Desktop/Python0/newdic1.txt') # 增加自定义词库
data_cut = data_dup.apply(lambda x : jieba.lcut(x) )
print(data_cut)
```

```
0                          [A, 市, 西湖, 建筑, 集团, 占道, 施工, 有, 安全隐患]
1                  [A, 市, 在水一方, 大厦, 人为, 烂尾, 多年, ，, 安全隐患, 严重]
2                          [投诉, A, 市, A1, 区苑, 物业, 违规, 收, 停车费]
3                   [A1, 区, 蔡锷, 南路, A2, 区华庭, 楼顶, 水箱, 长年, 不洗]
4                          [A1, 区, A2, 区华庭, 自来水, 好大, 一股, 霉味]
                                    ...
9204                              [要, 解决, 医患, 矛盾, 必须, 装, 监控, ！]
9205                [两, 孩子, 一个, 是, 一级, 脑瘫, ，, 能, 再, 生育, 吗, ？]
9206    [B, 市中心, 医院, 医生, 不负责任, ，, 做, 无痛, 人流, 手术, 后, 结果...
9208                                    [K8, 县惊现, 奇葩, 证明, ！]
9209          [请问, J4, 县, 卫计委, 社会, 抚养费, 到底, 该交, 多少, 钱, ？]
Name: 留言主题, Length: 8905, dtype: object
```

图 3 jieba 分词

4.1.2 删除停用词

停用词(Stop Words)，词典译为“电脑检索中的虚字、非检索用字”。停用词大致分为两类。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，通常本身没有明确的意义，只有放入一个完整的句子中才有一定作用，如常见的“的”、“在”之类。另一类词包括词汇词，比如‘请问’等，这些词的应用十分广泛，但是对这样的词搜索引擎无法保证能给出相关的搜索结果，不能够缩小搜索范围，并且还会降低搜索效率[2]，所以通常会把这些词删除，来提高搜索性能。

在删除停用词之后，统计文章词频，结合留言内容筛选出一部分对文本分类

无用的高频词，将这些词汇补充到停用词表中，然后对新的停用词表再进行一次删除停用词的操作。这一操作可以降低特征空间维数，节省存储空间，提高分类性能。删除停用词处理结果如下图：

```
#去除停用词
stopWords = pd.read_csv('./Desktop/Python0/stopword.txt',encoding='gbk',header=None,sep='111')
#list(stopWords.iloc[:,0])
stopWords = list(stopWords.iloc[:,0])+[' ']
data_after_stop = data_cut.apply(lambda x:[i for i in x if i not in stopWords])
print(data_after_stop)
```

```
C:\Users\DELL\Anaconda3\lib\site-packages\ipykernel_launcher.py:2: ParserWarning: Falling back
does not support regex separators (separators > 1 char and different from '\s+' are interpreted
cifying engine='python'.
```

```
0                  [西湖, 建筑, 集团, 占道, 施工, 安全隐患]
1                [在水一方, 大厦, 人为, 烂尾, 多年, 安全隐患]
2                      [区苑, 物业, 违规, 收, 停车费]
3            [蔡锷, 南路, A2, 区华庭, 楼顶, 水箱, 长年, 不洗]
4                  [A2, 区华庭, 自来水, 好大, 一股, 霉味]
                          ...
9204                        [医患, 矛盾, 装, 监控]
9205                      [两, 孩子, 一级, 脑瘫, 生育]
9206    [市中心, 医院, 医生, 不负责任, 做, 无痛, 人流, 手术, 胚芽]
9208                      [K8, 县惊现, 奇葩, 证明]
9209                [J4, 卫计委, 社会, 抚养费, 该交, 钱]
Name: 留言主题, Length: 8905, dtype: object
```

图 4 去除停用词

删除停用词得到的结果为列表格式，为了方便计算机处理，需要转化为以空格分隔的字符串，如下图：

```
#保存对应的一级标签
labels = data.loc[data_after_stop.index,'一级标签']
#列表转换为字符串 空格相连
adata = data_after_stop.apply(lambda x:' '.join(x))
print(adata)
print(labels)
```

```
0                  西湖 建筑 集团 占道 施工 安全隐患
1                在水一方 大厦 人为 烂尾 多年 安全隐患
2                      区苑 物业 违规 收 停车费
3            蔡锷 南路 A2 区华庭 楼顶 水箱 长年 不洗
4                  A2 区华庭 自来水 好大 一股 霉味
                          ...
9204                        医患 矛盾 装 监控
9205                      两 孩子 一级 脑瘫 生育
9206    市中心 医院 医生 不负责任 做 无痛 人流 手术 胚芽
9208                      K8 县惊现 奇葩 证明
9209                J4 卫计委 社会 抚养费 该交 钱
Name: 留言主题, Length: 8905, dtype: object
0       城乡建设
1       城乡建设
2       城乡建设
3       城乡建设
4       城乡建设
        ...
9204    卫生计生
9205    卫生计生
9206    卫生计生
9208    卫生计生
9209    卫生计生
Name: 一级标签, Length: 8905, dtype: object
```

图 5 列表转换为字符串

与原始数据进行对比，可以看出，预处理之后的文本占用内存缩小，词语之间以空格分隔，方便处理。这说明预处理在文本分类中是十分有效且必不可少的一步。对预处理之后的新闻内容包含的词进行“词云”处理，如下图所示：

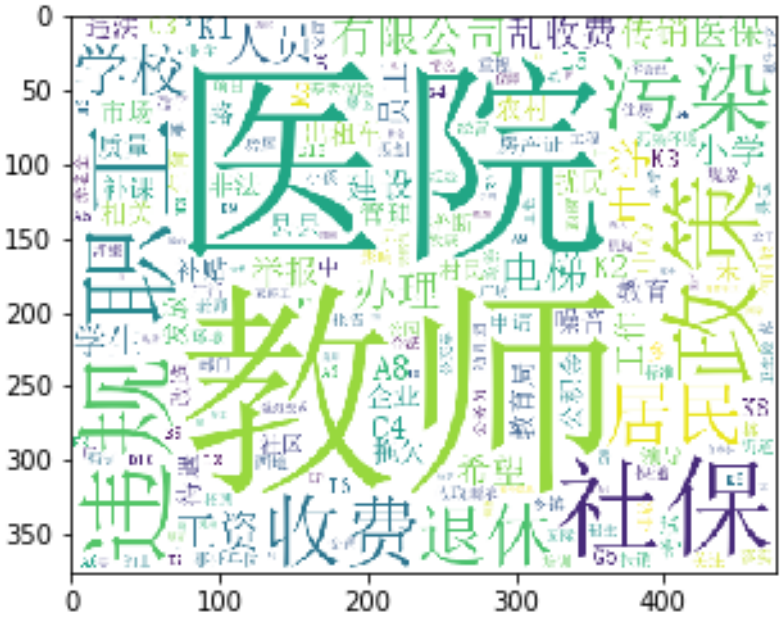

图 6 词云处理示意图

4.2 文本向量表示

计算机非常擅长处理电子表格、数据库这样的结构化数据，但是面对中文这样的非结构化数据，计算机并不能真正的像人类一样理解语言。因此，需要将非结构化数据转化为结构化数据。

4.3 特征处理阶段理论综述

经过预处理的文本集，尽管已经缩小了特征空间，但是为了提高最终的准确率和分类效率，必须进一步降低特征空间的维数。并不是所有的特征都具有预测功能，可能存在无关特征或者噪声特征，所以要进行特征选择和特征加权处理。特征选择是对特征空间中的特征词项的重要程度进行量化，然后进行特征词项的选择。

4.3.1 特征选择

文本分类问题最大的特点和困难是特征空间的高维性和稀疏性。高维稀疏的特征空间会给文本分类造成较大的影响，不仅会使文本分类具有高时间复杂度，并且会降低文本分类的性能。高维的特征空间对于绝大多数的分类算法来说都偏大，有时甚至会导致一些算法无法使用。但是，在高维特征空间中，有的特征是和分类没有关系的，甚至有一部分是误导分类的噪声。因此，在进行文本分类之前，需要降低文本特征空间的维数，同时删除文本特征空间中的噪声。

特征选择是指从原始特征几何中选取一小部分特征，组成新的特征集合的过程。显然，新的特征集合是原始特征集合的一个子集。[3]需要从原始特征集合中选取最能表示留言主题内容并且对分类最有效的特征集合。

常用的特征选择的方法有：频率统计、信息增益、卡方检验、互信息。本文所用的方法为频率统计方法，该方法有简单快速、结果比较符合实际情况的优点。

频率统计包含两部分，一部分是词频统计（TF），一部分是文档频率统计（DF）：

TF: Term frequency 即关键词词频，是指在某一文本中某一个词语出现的频率，用公式表示即为：

$$\mathrm{TF} = \frac{\mathrm{N}}{\mathrm{M}}. \tag{1}$$

表示单词在某文档中的频次 N 除该文档的单词数 M，越大越能代表文本特征。

需要实现设定一个阈值，将计算出的词频与阈值比较，如果 TF>阈值，则这个词可以选入特征项集，否则就删除。这个特征选择方法操作性强、效率高，但是他单纯的认为低频次对文本分类没有价值而过度重视高频词。实际上，有的低频词虽然出现次数少但是很具有代表性以至于它们可以反映文章的主旨内容。因此频率统计方法一般会结合其他方法一起使用进行特征选择。[4]

4.3.2 特征加权

经过特征选择后，可以选出最能代表留言内容的特征组成向量空间模型中的

文本特征集合。但是，这些特征对文本分类的影响不同，有的特征区分类别的能力较强，有的较弱，有些特征甚至有可能是噪声。因此，需要对文本特征进行加权，给有较强分类能力的特征赋予较大的权重，分类能力较差的则被赋予较小的权重，并且抑制噪声。

特征加权是指对文本特征集合中的每个特征根据其分类能力的大小赋予一定的权重。通过对特征进行加权操作，可以改善文本集合的空间分布状态。

词频本身可以表示样本的特征，一个词在文本当中出现的频率越高，说明它在文本中越重要，这个词对分类的贡献可能就越大，但是，某一些词语在正向情感和负向情感的句子中都有出现，这时候就不能只考虑词频信息，还要考虑逆文档频率（IDF）。

TF-IDF 权重策略：

IDF: Inverse document frequency 指逆向文本频率，是用于衡量关键词权重的指数，公式：

$$IDF = log\left(\frac{D}{D_W}\right), \tag{2}$$

$$TF - IDF = TF \times IDF\ . \tag{3}$$

IDF 越大代表这个词对文本分类的重要性越大。

4.4 分类器构造之朴素贝叶斯算法

朴素贝叶斯分类是以贝叶斯定理为基础，并且假设各个特征条件相互独立的，先通过已经给定的训练文本集，以特征词之间独立作为前提假设，学习从输入到输出的联合概率分布，基于学到的模型，输入 X 求出似的后验概率最大的输出结果 Y。对于相互独立的特征项t_i ，如果出现在了某文档 x 中，它属于类别 c 的条件概率可以通过下式计算：

$$P(c|x) = \frac{P(c)\prod_{i=1}^{n} P(t_i|c)}{\prod_{i=1}^{n} P(t_i)}. \tag{4}$$

其中：类别 c 的先验概率$P(c)$是根据现有的文档集所得，$P(t_i|c)$是t_i出现在

c 类别文本的条件概率；n 是文档 x 包含的词项数目。文本分类的目的在于找出文本内容最可能属于哪一个类别，这里的“最有可能”就是求得的后验概率最大。这时使用最大似然估计求$P(c)$和$P(t_i|c)$，在最大似然估计下，类别 c 的先验概率 $P(c)=\frac{N_c}{N}$，其中N_c是训练文本及中类别 c 所含的文档总数量，N 为训练文本集中全部类别下的文档总数；条件概率为：

$$P(t_i|c)=\frac{T_{ct}}{\sum T_{ct}}. \tag{5}$$

上式中的T_{ct}表示在 c 类别下包含特征项 t 的文档个数，$\sum T_{ct}$是 c 类别下所含文档的总数目。

4.5 基于 TF-IDF 的特征处理及分类器的构建

求出每个词的 TF-IDF 值，TF-IDF 越大代表该词对分类的贡献越大。通过选取全体留言的 80%作为训练文本集，20%作为测试文本集。在训练文本集上训练分类器，与测试集的结果作对比。用 python 实现的结果如下：

```
#文本数据的向量化表示
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
cv = CountVectorizer()
data_tr = cv.fit_transform(data_tr)
X_tr = TfidfTransformer().fit_transform(data_tr.toarray()).toarray()
print(X_tr)
print(X_tr.shape)

data_te = CountVectorizer(vocabulary=cv.vocabulary_).fit_transform(data_te)
X_te = TfidfTransformer().fit_transform(data_te.toarray()).toarray()
print(X_te)
print(X_te.shape)
```

```
[[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.]]
(7124, 11847)
[[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.]]
(1781, 11847)
```

图 7 TF-IDF 处理结果

图 7 显示测试集上的的平均准确率为 61.7%，平均召回率为 63.2%，平均 F1 值为 62.2%。

```
: #计算F1值
from sklearn.metrics import f1_score, precision_score, recall_score

f1 = f1_score(labels_te, labels_te_pred, average='macro' )
p = precision_score(labels_te, labels_te_pred, average='macro')
r = recall_score(labels_te, labels_te_pred, average='macro')

print(f1,p,r)
```

```
0.6224090419666404 0.617175079341594 0.6317390814040118
```

图 8 分类方法评价结果

五、问题二：热点问题挖掘

5.1 LDA 主题模型

LDA（Latent Dirichlet Allocation）是一种文档主题生成模型，也称为一个三层贝叶斯概率模型，包含词、主题和文档三层结构。所谓生成模型，就是认为每个文本的每个词都是通过“以一定概率选择了某个主题，并从这个主题中以一定概率选择某个词语”这样一个过程得到。文本到主题服从多项式分布，主题到此服从多项式分布。LDA 是一种非监督机器学习技术，可以用来识别大规模文档集或语料库中隐藏的主题信息。它采用了 bag of words 的方法，这种方法将每一篇文档视为一个词频向量，从而将文本信息转化为了易于建模的数字信息。但是词袋方法没有考虑词与词之间的顺序，这简化了问题的复杂性，但也为模型的改进提供了契机。每一篇文档都代表一些主题所构成的一个概率分布，而每一个主题又代表了很多单词构成的一个概率分布。

在本文中，对于每一条留言，定义了如下的生成过程：

1. 对每一条留言，从主题分布中抽取一个主题；
2. 从上述被抽到的主题所对应的词分布中抽取一个词；

3. 重复上述过程直至遍历留言中的每一个词。

LDA 降维的目标：将带有标签的数据降维，投影到低维空间同时满足三个条件：

1.尽可能多地保留数据样本的信息（即选择最大的特征是对应的特征向量所代表的的方向）。

2.寻找使样本尽可能好分的最佳投影方向。

3.投影后使得同类样本尽可能近，不同类样本尽可能远。

在对文本的主题特征进行研究时，我们往往要事先指定 LDA 生成的主题的数目 T，而一般的解决方法是使用困惑度 perplexity 来计算，原理如下(概率分布 perplexity)：

$$\text{perplexity}(D_{test}) = \exp\{\frac{-\sum_{d=1}^{M}\log(p(w_d))}{\sum_{d=1}^{M}N_d}\}, \quad (6)$$

其中，M 是测试语料库的大小，N_d 是第 d 篇文本大小（即单词个数）。

给某个文档先选择一个主题z，再根据该主题生成文档，该文档中的所有词都来自一个主题，gamma 是训练集学习得到的文本-主题分布。生成文档$\mathbf{w}$的概率为：

$$\text{p(w)} = \sum_z p(z)p(w|z, gamma). \quad (7)$$

perplexity 的上半部分就是生成整个文档的似然估计（表示训练集训练出的参数的生成能力）的负值，由于概率取值范围为[0,1]，按照对数函数的定义，分子值是一个大数，而分母是整个测试集的单词数目。也就是说模型生成能力越强，perplexity 值越小。[5]

使用 sklearn 的函数，在训练集上生成 gamma 和主题用来计算测试集的困惑度。可以看到，主题数目为 200 的时候模型生成能力较强。

运用 python 实现上述过程得到结果如图 9：

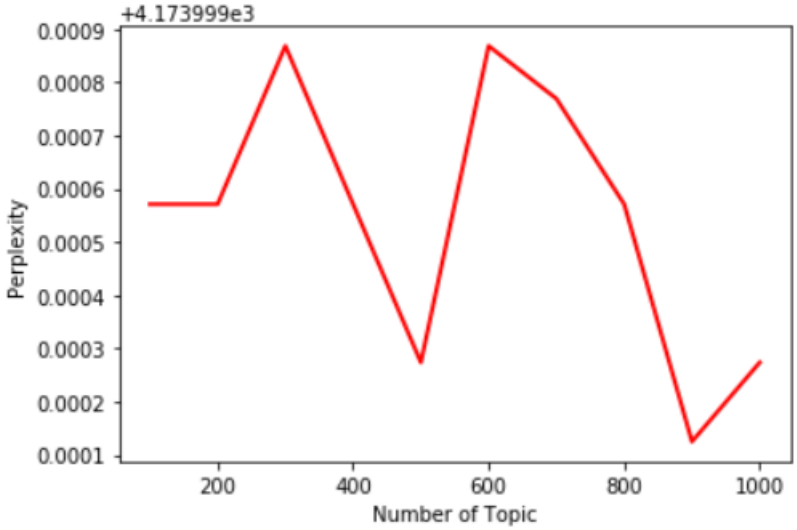

图 9 主题数与困惑度的关系

将留言文档作为算法输入并输入主题数 K=200，利用 python 算法求得留言文档关于 200 个主题的狄利克雷分布。狄利克雷的输入参数为人为设置的一个超参数，输出参数为留言文档对应各个主题的概率分布的分布。将任意主题产生各个词的概率表示出来，人为设置主题产生的各个词的数量，即设置超参数，得到各个主题产生各个词的狄利克雷分布。至此已经得到了留言文档对应各个主题的概率分布的分布（即狄利克雷分布），以及文档产生各个词的概率分布的分布。

如果已知第 i 个在文档中出现 n 次，且已知它对应各个主题的概率（这里每个词对应各个主题的概率就是留言文档对应各个主题的概率，二者同分布），求该词被各个主题产生的次数。

求出各个主题产生各个单词的数量：记作向量 m，将向量 m 作为新的参数再次带入狄利克雷分布，就又会得到每个主题产生各个词的概率，反复迭代，最终得到收敛的主体产生各个词的概率，这时，该算法就会根据输入的文档 d，找到潜在主题下的相关词。

LDA 主题聚类算法运行后得到了分为 200 个不同类，也就是 200 个不同的主题，每一个主题都对应不同的相关词。Python 实现结果中主题热度排名前五所对应的词如下图：

```
(186, '0.229*"严重" + 0.179*"影响" + 0.076*"居民" + 0.071*"生活" + 0.067*"区"')
(31, '0.289*"希望" + 0.158*"市" + 0.142*"A" + 0.084*"地铁" + 0.081*"能"')
(28, '0.194*"无" + 0.179*"投诉" + 0.117*"物业" + 0.066*"时代" + 0.052*"小区"')
(101, '0.000*"餐" + 0.000*"臭" + 0.000*"小吃店" + 0.000*"断腿" + 0.000*"肉"')
(132, '0.073*"A" + 0.047*"市" + 0.047*"还有" + 0.045*"三年" + 0.044*"黄兴"')
(126, '0.218*"中心" + 0.184*"现象" + 0.093*"乱" + 0.057*"停" + 0.031*"严重"')
(98, '0.000*"餐" + 0.000*"臭" + 0.000*"小吃店" + 0.000*"断腿" + 0.000*"肉"')
(94, '0.000*"餐" + 0.000*"臭" + 0.000*"小吃店" + 0.000*"断腿" + 0.000*"肉"')
(1, '0.304*"公司" + 0.128*"星沙" + 0.084*"县" + 0.084*"A7" + 0.071*"人行道"')
(122, '0.143*"区" + 0.135*"A3" + 0.123*"业主" + 0.107*"小区" + 0.083*"安置"')
(15, '0.000*"餐" + 0.000*"臭" + 0.000*"小吃店" + 0.000*"断腿" + 0.000*"肉"')
(7, '0.113*"尽快" + 0.109*"发放" + 0.065*"请" + 0.065*"A4" + 0.064*"的"')
(93, '0.120*"政策" + 0.096*"的" + 0.088*"装修" + 0.060*"走" + 0.059*"花园"')
(195, '0.224*"上" + 0.104*"西湖" + 0.096*"A3" + 0.089*"星城" + 0.083*"街道"')
(162, '0.309*"拆迁" + 0.098*"不" + 0.076*"A7" + 0.037*"县" + 0.028*"为何"')
(120, '0.364*"附近" + 0.099*"区" + 0.088*"夜宵" + 0.081*"摊" + 0.064*"A1"')
(5, '0.000*"江山" + 0.000*"责任" + 0.000*"质量" + 0.000*"不" + 0.000*"住户"')
(105, '0.124*"深夜" + 0.091*"扰民" + 0.089*"麻将馆" + 0.087*"中路" + 0.055*"市"')
(107, '0.344*"咨询" + 0.232*"问题" + 0.083*"A" + 0.082*"A7" + 0.079*"县"')
(86, '0.097*"很" + 0.056*"扰民" + 0.050*"范围" + 0.048*"A3" + 0.047*"天顶"')
```

图 10 热度前五主题下的相关词

仅仅利用 LDA 算法并不能得到热度大的具体的留言内容，因此需要结合热度评分模型，通过计算每一类下每一条留言的热度，最终求和得到每一类留言的总热度。最终求得热度排名前五的五个留言类别及其具体内容。

5.2 热度评分模型

热度评分模型的主要作用是从每个留言中抽取留言的各项属性，包括同类留言的个数、点赞数、反对数。这些属性对于这个留言的热度是最直接的体现。综合考虑这些属性通过计算公式得出留言的热度得分。本文运用热度排行算法计算留言的热度。算法中用来标注文章新旧程度的单位为秒。

t 为发表时间与某一固定时间的差值。留言一旦发表，时间 t 就是固定的，不会随时间改变，并且留言越新，t 值越大。发表时间和留言排名的影响可以概括为：发表时间对排名有很大影响，本算法使得新的留言比旧的留言排名靠前；留言的得分不会因为时间的流失而减少，但是新的话题会比旧的留言得分高。

令 x 代表赞成票与反对票的差值，这可以使得一些有争议的留言的热度排名比较靠后。

令 y 表示投票的方向，如果这个留言的赞成票居多，y 的值就为+1，如果反

对票多于赞成票，那么 y 就为-1。如果赞成票和反对票一样多，那么 y 的值取 0。

z 表示赞成票超过反对票的数量，代表留言的受肯定程度。结合上述几个变量，留言热度的最终得分公式为：

$$\text{Score} = log_{10}z + \frac{yt}{4500}. \tag{8}$$

$log_{10}z$ 表示赞成票超过反对票的数量越多得分越高。需要注意的是，这里用的是以 10 为底的对数，意味着 z=10 可以得到 1 分，z=100 可以得到 2 分。也就是说，前 10 个投票人与后 90 个投票人（乃至再后面 900 个投票人）的权重是一样的，即如果一个帖子特别受到欢迎，那么越到后面投赞成票，对得分越不会产生影响。而当反对票超过或等于赞成票 z=1，因此这个部分等于 0，也就是不产生得分。

本文的留言热度排序算法使用了对数函数来衡量前面的投票与其他投票的差距使其前十个好评和之后的 100 个，1000 个投票有相同的权重。

2、yt/45000

这个部分表示，t 越大，得分越高，即新留言的得分会高于老留言。它起到自动将老留言的排名往下拉的作用。 分母的 45000 秒，等于 12.5 个小时，也就是说，后一天的留言会比前一天的留言多得 2 分。结合前一部分，可以得到结论，如果前一天的留言在第二天还想保持原先的排名，在这一天里面，它得到的净赞成票必须增加 100 倍。

y 的作用是用来产生正分和负分。当赞成票超过反对票时，得分为正；当赞成票少于反对票时，得分为负；当两者相等，得分为 0。这就保证了得到大量净赞成票的留言，会排在前列；得到大量净反对票的留言，会排在最后。投票对于总分的贡献不大，但是当投票的意见倾向发生变化时（由正面评价转向负面评价），投票对于总分的作用却是决定性（y 的取值）。

六、问题三：答复意见的评价

6.1 模糊综合评价法

在客观世界中，存在着大量的模糊概念和模糊现象。一个概念往往和它对立的概念无法划出一条明确的分界，凡是涉及到模糊概念的现象被称为模糊现象。模糊性是事件本身状态的不确定性，或者说是指某些事物或者概念的边界不清楚，这种边界不清楚并不是由于人的主观认识达不到客观实际所造成的，而是事物的一种客观属性。对于相关部门对留言的答复，其相关性、完整性、可解释性的界定都是非常模糊的，并不能量化具体计算。因此，考虑用模糊数学来评价答复意见。

模糊综合评价是借助模糊数学的一些概念，对实际的综合评价问题提供一些评价的方法。具体的说，模糊综合评价就是以模糊数学为基础，应用模糊关系合成的原理，将一些边界不清、不容易定量描述的因素定量化，从多个因素对被评价事物隶属等级状况进行综合性评价的一种方法。

6.2 评价模型的建立

6.2.1 确定评价指标和评价等级

本文中对于留言答复意见的评价指标分为相关性、完整性、可解释性三个方面度量。其中相关性是指答复内容与留言主题是否相关，主要体现在答复内容是否与留言主题所对应的关键词相关；完整性是指，针对群众留言，是否做出确实具体的答复；可解释性是指人类能够理解决策原因的程度。留言答复的可解释性越高，人们就越容易理解为什么做出某些决定或预测。

评价指标为：U={相关性，完整性，可解释性}。

评价等级为：V={优秀，良好，一般，较差，差}。

6.2.2 构造评价矩阵和确定权重

可按照经验给出，可全部看成 1:1:1

七、模型评价

7.1 模型的优点

1.运用 TF-IDF 算法的优点是简单快速，结果比较符合实际情况。

2. LDA 主题聚类模型采用了词袋的方法，这种方法将每一篇文档视为一个词频向量，从而将文本信息转化为了易于建模的数字信息。

7.2 模型的缺点

1.词袋方法没有考虑词与词之间的顺序，简化了问题的复杂性。

2. 由于 Dirichlet 分布随机向量各分量间的弱相关性，使得我们假想的潜在主题之间也几乎是不相关的，这与很多实际问题并不相符。

八、参考文献

[1] https://edu.tipdm.org/course/1598

[2]张航. 基于朴素贝叶斯的中文文本分类及 Python 实现[D].山东师范大学,2018.

[3] 刘赫. 文本分类中若干问题研究[D].吉林大学,2009.

[4] 董露露.基于特征选择及 LDA 模型的中文文本分类研究与实现[D].合肥：安徽大学,2014.

[5] https://blog.csdn.net/u014449866/article/details/80218054