基于相似度分析与 TF-IDF 模型的智慧政务模型

摘要

自然语言处理和文本挖掘技术日新月异，在生活中也被广泛应用。本文构建了基于相似度分析和 TF-IDF 模型的智慧政务模型。对于网络问政平台中民众的留言详情进行一级标签分类，对于热点问题进行归类，定义合理的热度评价指标和评价结果，对于相关部门留言的答复意见的质量的评价方案，模型可以定位并给出答案。

在数据预处理阶段，我们从数据中抽取语句进行分词、去停用词和添加新词处理。利用词频统计计算出七个标签的历史常用词作为相似度分析的一部分，并通过 text2vec 获得词向量表示。

为了对留言进行一级标签分类，我们利用相似度分析与 TF-IDF 模型对留言详情建立一级标签分类模型，为了解分类效果，利用 F_1-score 评价模型 $F_1 = \frac{2PR}{P+R}$，因模型随机分配训练集与测试集，所以需要多次运行才能客观了解 F_1-score。从多次运行的结果可看出，模型中的每个标签 F_1-score 平均值都能达到 75%以上。

对热点问题挖掘，我们对留言主题利用 Python 中的 fuzzywuzzy 库进行字符串模糊匹配以及相似度分析，并搜索排序，得到相似字符串种类且每条留言与之的相似度。我们规定阈值为相似度大于 60%，且相似字符串数量大于 4 条的留言主题为热点问题，并规定热度指标为各留言问题在单位时间内的提到的次数，公式为：$\mathrm{f} = \frac{s}{t}$，并作出热度指标 TOP5 的热点问题且各热点问题的详细留言。

对于答复意见的质量评价，我们从四个方面出发，分别为：相关性、完整性、可解释性和时效性。基于满分为 100 分，四个方面的权重相等均为 25 分。利用相似度分析计算留言详情和答复意见的相关性，计算提前规定的格式与答复意见的相似度，不难发现相似度越高相关性越强，完整性中相似度达到 98%以上则说明答复意见完全符合规范。为了评分模型的直观表达，可解释性与时效性我们做出规定：答复中法律法规数量为 0 时分数为 0 分、数量为 1 时为 21.25 分、数量在 2 到 5 条时为 23.75 分、数量在 5 条以上为 25 分，时间以数据总体的四分位数作为分界点计算分数，其中 5 天以内回复为 25 分、5 天到 11 天回复为 21.25 分、11 天到 22 天回复为 18.75 分、22 天以后回复为 12.5 分。四者的结合便是我们对答复意见质量的评分结果。

关键字：相似度分析 TF-IDF 模型 数据预处理 词频统计 F_1-score 评价模型 字符串模糊匹配

Abstract

Natural language processing and text mining technologies are changing rapidly and are widely used in daily life. This paper constructs an intelligent government model based on similarity analysis and tf-idf model. For the details of people's comments on the network platform, the model can be used to classify the first-class labels, classify hot issues, define reasonable heat evaluation indicators and evaluation results, and evaluate the quality of comments left by relevant departments. The model can be used to locate and give answers.

In the data preprocessing stage, we extract statements from the data for word segmentation, stop words and add new words. As part of similarity analysis, word frequency statistics were used to calculate the historical common words of seven tags, and word vector representation was obtained by text2vec.

In order to carry out the first-level label classification of the comments, we use similarity analysis and tf-idf model to establish the first-level label classification model for the details of the comments. In order to understand the classification effect, we use the f1-score evaluation model: $F_1 = \frac{2PR}{P+R}$. Because the model randomly assigns the training set and the test set, it needs multiple runs to objectively understand f1-score. It can be seen from the results of multiple runs that the average value of each label f1-score in the model can reach more than 75%.

For hot issues mining, we use the fuzzywuzzy library in Python to carry out string fuzzy matching and similarity analysis for message topics, and search for sorting, to get similar string types and the similarity degree of each message. We set the threshold value as the similarity degree greater than 60% and the number of similar strings greater than 4 as the hot topic, and set the heat index as the number of times each message question is mentioned in unit time, the formula is: $\mathrm{f} = \frac{s}{t}$, and make the hot topic of the TOP5 heat index and the detailed message of each hot issue.

For the quality evaluation of replies, we start from four aspects: relevance, integrity, interpretability and timeliness. Based on the full score of 100 points, the weight of all four aspects is equal to 25 points. By using similarity analysis to calculate the correlation between message details and reply comments, and calculate the similarity between the format stipulated in advance and reply comments, it is not difficult to find that the higher the similarity is, the stronger the correlation is. If the similarity reaches more than 98% in the integrity, it indicates that the reply comments completely conform to the specification. Visual expression of the model in order to score, interpretability and timeliness we make provisions: number of laws and regulations in response to 0 score of 0 points, quantity, amount to 1 to 21.25 points in 2 to 5, 23.75 points, the number in the above article 5 to 25 points and time in data overall score quartile as a cut-off point, reply within five days of 25 points, five days to 11 days to reply to 21.25 points, 11 to 22 days replies for 18.75 points, after 22 days of 12.5 points. The combination of the four is the result of our rating of the quality of the responses.

key word: similarity analysis tf-idf model data preprocessing word frequency statistics F_1-score evaluation model String fuzzy matching

目录

一、简介

1.1 挖掘意义

近年来，随着各大媒体渠道的逐渐开放化，微博、市长信箱、阳光热线等网络问政平台逐渐成为政府了解人民民情民意的重要渠道，从而实现科学决策、民主决策，真正做到全心全意为人民服务。随着网络的日益普及，互联网在中国民众的政治、经济和社会生活中扮演着日益重要的角色，成为中国公民行使知情权、参与权、表达权和监督权的重要渠道。

网络民意的凸显，一方面在于网媒的发展、公民意识的成长，另一方面更在于执政者对于网络民意的日益重视。2008 年 6 月 20 日，胡锦涛总书记到人民网强国论坛同网友在线交流，成为“中国第一网民”，重视程度可见一斑。据相关调查，超七成公众认为网络表达将成中国民主建设的新通道，近六成人认为有助于拉近政府与民众距离。

1.2 挖掘目标

我们要构建一个智慧政务模型。模型可以减少人工对进行留言划分和热点整理的相关部门的工作，以及提升政府的管理水平和施政效率。具体用在情景上就是可以对群众留言进行有效分类，且能将热点问题进行挖掘，便于政府部门对民意的了解。并且能够对政府回答群众的答复意见进行评价，便于政府提高答复意见水平，更便民。

二、数据预处理

2.1 分词

由于中文文本的特点是词与词之间没有明显的界限，从文本中提取词语时需要分词，本文采用 Python 开发的一个中文分词模块——jieba 分词[1]，对问题和回答中的每一句话进行分词进行中文分词。jieba 分词用到的算法为最短路径匹配算法该算法首先利用词典找到字符串中所有可能的词条,然后构造一个有向无环图。其中,每个词条对应图中的一条有向边,并可利用统计的方法赋予对应的边长一个权值，然后找到从起点到终点的最短路径，该路径上所包含的词条就是该句子的切分结果。

2.2 停用词过滤

停用词是指在信息检索中，为节省存储空间和提高搜索效率，在处理自然语言数据（或文本）之前或之后会自动过滤掉某些字或词，这些字或词即被称为 Stop Words（停用词）。这些停用词都是人工输入、非自动化生成的，生成后的停用词会形成一个停用词表。但是，并没有一个明确的停用词表能够适用于所有的工具。

2.3 词频统计

在 jieba 分词之后利用 pandas 库中的函数进行词频统计，统计出 7 类一级标签词频最高的词语作为标签。

2.4 向 jieba 库中添加新词

jieba 库中对于每类问题可能会有不同的新词需要添加，把这些新词利用库中函数添加进库中。

2.5 text2vec

text2vec 包是由 Dmitriy Selivanov 于 2016 年 10 月所写的 R 包。此包主要是为文本分析和自然语言处理提供了一个简单高效的 API 框架。由于其由 C++所写，同时许多部分（例如 GloVe）都充分运用 RcppParallel 等包进行并行化操作，处理速度得到加速。并且采样流处理器，可以不必把全部数据载入内存才进行分析，有效利用了内存，可以说该包是充分考虑了 NLP 处理数据量庞大的现实。text2vec 包也可以说是一个文本分析的生态系统，可以进行词向量化操作（Vectorization）、Word2Vec 的“升级版 GloVe 词嵌入表达”、主题模型分析以及相似性度量四大方面，可以说非常的强大和实用强大和实用[2]。

三、模型的构建

3.1 群众留言分类

3.1.1 一级标签分类模型

基于“用 Python 进行简单的文本相似度分析”[3]，在全部数据中抽取出 70%个数据，利用 gensim 包和 TF-IDF 模型分析对于留言详情进行一级标签分类。

权重策略文档中的高频词应具有表征此文档较高的权重，除非该词也是高文档频率词[4]。

1. TF：Term frequency 即关键词词频，是指一篇文档中关键词出现的频率

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

（N :单词在某文档中出现的频次；M :该文档的单词数）

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

$$IDF = \log(\frac{D}{D_w}) \tag{2}$$

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

（D :总文档数；D_w：出现了该单词的文档数）

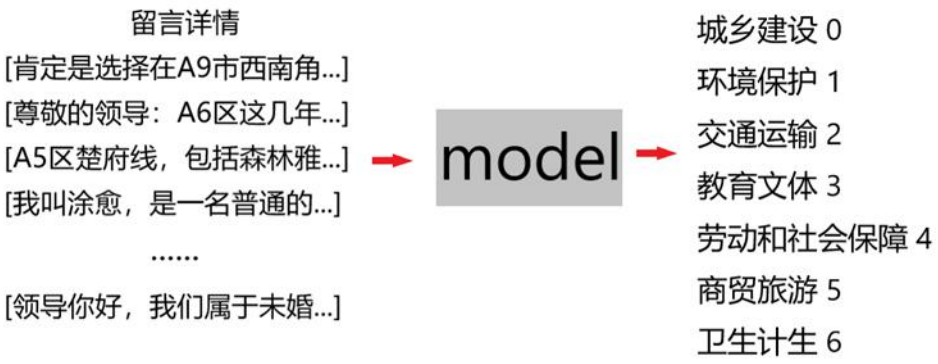

图 1：一级标签概念模型

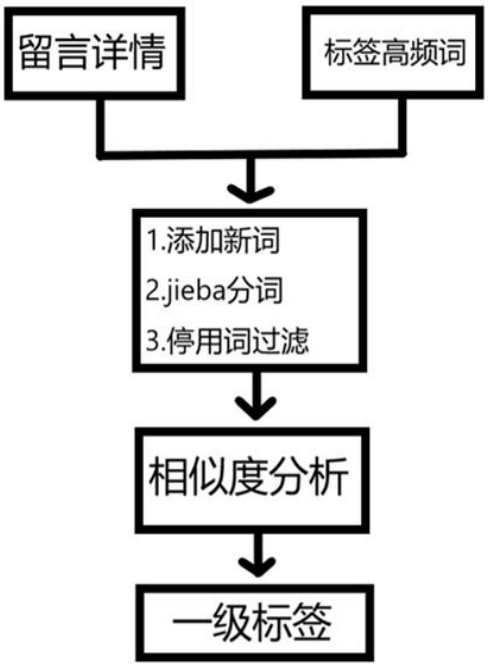

图 2：一级标签模型制作流程图

表 1：新一级标签例子

留言编号	留言用户	留言主题	留言时间	留言详情	一级标签
1775	U0002150	给 A9 市城...	2019/7/4...	肯定是选...	交通运输
1783	U0004763	请 A6 区政...	2019/7/4...	尊敬的领...	城乡建设
1827	U000613	A5 区楚府...	2019/7/1...	A5 区楚府...	商贸旅游
...	...	...	...	...	...

表 2：每种一级标签的个数

一级标签	个数
城乡建设	1119
环境保护	775
交通运输	643
教育文体	1273
劳动和社会保障	1139
商贸旅游	852
卫生计生	798
合计	6600

3.1.2 对分类方法进行评价

首先，我们把多分类模型化成多个一对多的二分类模型。

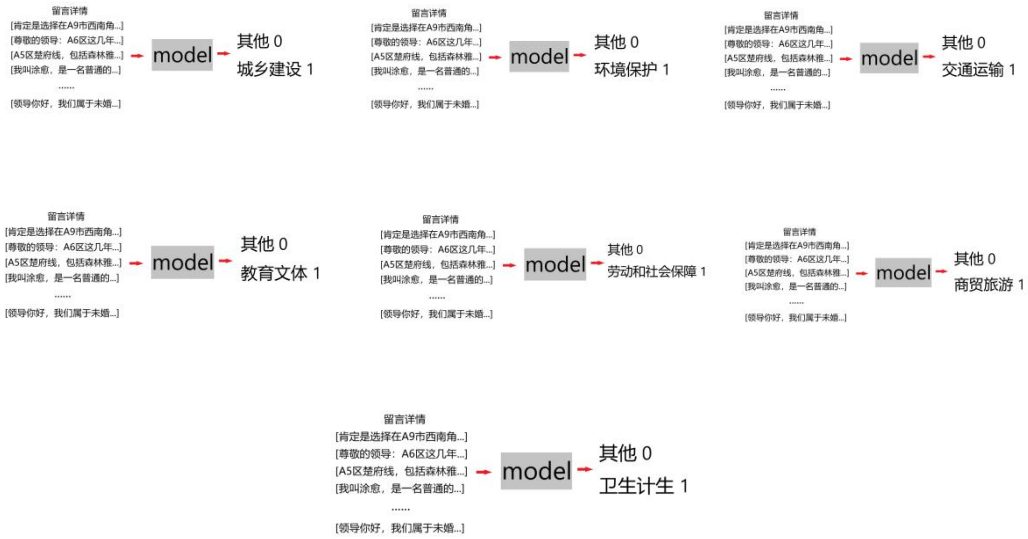

图 3：七个二分类模型

然后，利用朴素贝叶斯精度[5]和精确率（Precision）、召回率（Recall）这两者的调和的平均数 F_1-score 来评价我们模型的表现效果，其中因为模型随机分配训练集和测试集，为了更加精确了解模型的效果，可多运行几次。指标的详细定义如下：为了放便后面符号的说明定义一个混淆矩阵，如表：

表 3：混淆矩阵

	相关	不相关
被检测到的	TP	FP
未被检测到的	FN	TN

1. *TP* (True Positive):正类项目被判定为正类
FP (False Positive):负类项目被判定为负类
FN (False Negative):正类项目被判断为负类
TN (True Negative):正类项目被判断为负类

2. 精准率(Precision):

是衡量某一检索系统的信号噪声比的一种指标，即检出的相关文献与检出的全部文献的百分比。

$$P = \frac{TP}{(TP + FP)} \tag{4}$$

3. 召回率(Recall):

召回率(Recall)是检索出的相关文档数和文档库中所有的相关文档数的比率，衡量的是检索系统的查全率。

$$R = \frac{TP}{(TP + FN)} \tag{5}$$

4. F_1-score：

分类的 F_1 值就是准确率和召回率的调和平均值，具体计算公式为：

$$F_1 = \frac{2PR}{P + R} \tag{6}$$

表 4：F_1 分数和朴素贝叶斯精度

城乡建设	F_1.1	F_1.2	F_1.3	F_1.4	Avg
	0.76	0.76	0.71	0.78	0.7525
	朴素贝叶斯.1	朴素贝叶斯.2	朴素贝叶斯.3	朴素贝叶斯.4	Avg
	0.77	0.75	0.73	0.79	0.76
环境保护	F_1.1	F_1.2	F_1.3	F_1.4	Avg
	0.81	0.82	0.81	0.79	0.8075
	朴素贝叶斯.1	朴素贝叶斯.2	朴素贝叶斯.3	朴素贝叶斯.4	Avg
	0.82	0.81	0.82	0.79	0.81
交通运输	F_1.1	F_1.2	F_1.3	F_1.4	Avg
	0.75	0.76	0.81	0.82	0.785
	朴素贝叶斯.1	朴素贝叶斯.2	朴素贝叶斯.3	朴素贝叶斯.4	Avg
	0.78	0.8	0.82	0.83	0.8075
教育文体	F_1.1	F_1.2	F_1.3	F_1.4	Avg
	0.82	0.85	0.85	0.83	0.8375
	朴素贝叶斯.1	朴素贝叶斯.2	朴素贝叶斯.3	朴素贝叶斯.4	Avg
	0.82	0.85	0.84	0.83	0.835
劳动和社会保障	F_1.1	F_1.2	F_1.3	F_1.4	Avg
	0.81	0.78	0.79	0.8	0.795
	朴素贝叶斯.1	朴素贝叶斯.2	朴素贝叶斯.3	朴素贝叶斯.4	Avg
	0.82	0.8	0.8	0.81	0.8075

商贸旅游	F_1.1	F_1.2	F_1.3	F_1.4	Avg
	0.76	0.79	0.75	0.77	0.7675
	朴素贝叶斯.1	朴素贝叶斯.2	朴素贝叶斯.3	朴素贝叶斯.4	Avg
	0.76	0.79	0.78	0.78	0.7775
卫生计生	F_1.1	F_1.2	F_1.3	F_1.4	Avg
	0.85	0.9	0.87	0.88	0.875
	朴素贝叶斯.1	朴素贝叶斯.2	朴素贝叶斯.3	朴素贝叶斯.4	Avg
	0.85	0.9	0.87	0.88	0.875

3.2 热点问题挖掘

根据 Levenshtein Distance 算法，又叫 Edit Distance 算法，是指两个字符串之间，由一个转成另一个所需的最少编辑操作次数。许可的编辑操作包括将一个字符替换成另一个字符，插入一个字符，删除一个字符。一般来说，编辑距离越小，两个串的相似度越大。我们选择运用 Python 中的 fuzzywuzzy 库，此库不仅可用于计算两个字符串之间的相似度，而且还提供排序接口能从大量候选集中找到最相似的句子。我们导入了两个模块：fuzz，process，fuzz 主要用于两字符串之间匹配，process 主要用于搜索排序。

建模步骤：导入 FuzzyWuzzy 库 ，运用 process.extract()函数，此可从列表 ListS 中找出 Top n 与 S1 最相似的句子，进行字符串模糊匹配。我们运用此函数来自定义一个匹配函数，返回需要匹配的字符与对照列表最相近的字符串，并输出他的匹配度。

我们规定阈值为匹配度高于 60，且输出相似留言高于 4 条的留言为热点问题并将其整理为字典输出，再运用值反查键的形式，查出这些热点问题在原表的位置并将其输入 Excel 表中。

对表进行再次处理为各个热点问题表，并规定热度指标为单位时间内提的留言次数。

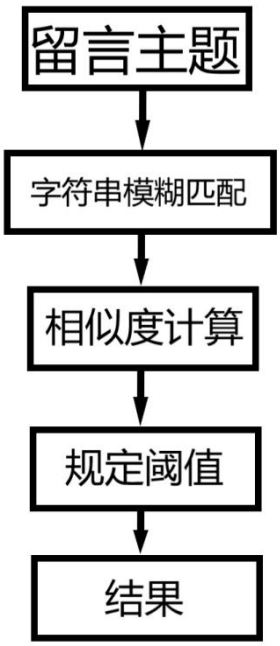

图 4：热点问题流程图

$$f = \frac{s}{t} \quad (7)$$

f：热度指标

s：同一个问题的留言总数

t：同一个问题最后一个留言的时间-第一个留言的时间

由这个留言指标，我们提出了留言问题 TOP5：

表 5：留言问题 TOP5

热度排名	问题 ID	热度指数	时间范围	地点/人群	问题描述
1	1	0.241	2019/08/22 至 2020/01/06	A 市 A2 区丽发新城小区居民	A 市 A2 区丽发新城小区遭搅拌站严重污染及扰民
2	2	0.117	2019/01/15 至 2019/08/26	A 市购房者	A 市购房问题
3	3	0.115	2019/01/05 至 2019/12/27	A 市人群	A 市地铁公交相关意见
4	4	0.094	2019/01/04 至 2019/12/31	A 市人群	A 市城市建设相关意见
5	5	0.0640	2019/07/07 至 2019/09/01	A 市伊景园购房者	A 市伊景园滨河苑车位捆绑销售

3.3 答复意见评价模型

1.相关性：我们对“留言详情”和“答复意见”做相似度分析，相似度越高

就代表了答复意见相关性越好。

2.完整性：首先要先规定好一种答复意见规范格式，例如：我们采用的规范格式是：网友“xxx”您好！您的留言已收悉。现将有关情况回复如下：xxx。感谢您对我们工作的支持、理解与监督！x 年 x 月 x 日。对此格式取出留言详情中相应部分的文字做相似度分析，在此，我们规定：

表 6：结果规定

参数	含义
98%以上	完全符合规范格式
85%到 98%	基本符合规范格式
85%以下	不符合规范格式

3.可解释性：为了答复意见的权威性和可解释性，意见中常常会采用法律法规来支持意见，所以我们利用 Python 计算出每条答复意见中所使用了多少条法律法规来表现答复意见的可解释性，采用的法律法规越多表示的是答复意见可解释性越好，对此我们做出以下规定

表 7：数量对应的得分

数量	得分
0	0
1	21.25
2-5	23.75
5	25

4.时效性：相关部门答复时，时效性也是市民很关心的重要一点，因为很多留言没有被及时反馈，就有一定的可能造成损失，所以对此我们利用答复时间和留言时间的差的四分位数做出以下规定：

表 8：及时性规定

天数	含义	得分
5 天以内	回复及时	25
5 天到 11 天	回复较及时	21.25
11 天到 22 天	回复稍及时	18.75
22 天以后	回复不及时	12.5

由以上标准，我们给四个标准平均分配权重，利用 Python 得到的评分结果为：

表 9：评分结果

留言编号	留言用户	留言主题	留言时间	留言详情	答复意见	答复时间	时间差	答复评分
2549	A000...	A2 区...	201...	2019...	现将...	2019...	15.23	87.46
2554	A000...	A3 区...	201...	潇楚...	网友...	2019...	14.74	62.20
2555	A000...	请加...	201...	地处...	市民...	2019...	14.76	85.46
2557	A000...	在 A...	201...	尊敬...	网友...	2019...	14.78	88.13
...	...	...	...	...	...	...	...	...

四、实验平台和数据来源

4.1 实验平台

使用到的 Python 库有：

1. pandas[6]：基于 NumPy 的一种工具，该工具是为了解决数据分析任务而创建的。Pandas 纳入了大量库和一些标准的数据模型，提供了高效地操作大型数据集所需的工具。pandas 提供了大量能使我们快速便捷地处理数据的函数和方法。

2. jieba：在数据预处理中给中文分词。

3. text2vec：为文本分析和自然语言处理提供了一个简单高效的 API 框架。

4. re：是 Python 的标准库，主要用于字符串匹配。

5. sklearn：sklearn (scikit learn)是一个常用的 Python 库，封装了大量的常用机器学习算法，有以下几个特点：对数据挖掘和数据分析来说，是简单且高效的工具；人人都可以调用，且在不同的业务场景下，可以复用；基于 scipy，numpy 和 matplotlib；开源的库，可以商业使用。sklearn 中主要有以下 6 个模块：分类，回归，聚类，降维，模型选择，预处理。

6. gensim[7]：是一个免费的 Python 库，它可以用来从文档中自动提取语义主题，并且尽可能地做到轻松（对人）高效（对电脑）。Gensim 致力于处理原始的、非结构化的数字文本（普通文本）。Gensim 中用到的算法，如潜在语义分析（LatentSemantic Analysis，LSA）、隐含狄利克雷分配（Latent Dirichlet Allocation，LDA）或随机预测（RandomProjections）等，是通过检查单词在训练语料库的同一文档中的统计共现模式来发现文档的语义结构。这些算法都是无监督算法，也就是无需人工输入——你仅需一个普通文本的语料库即可。一旦这些统计模式被发现了，所有的普通文本就可以被用一个新的、语义代号简洁地表示，并用其查询某一文本与其他文本的相似性。

7. FuzzyWuzzy[8]：是一个简单易用的模糊字符串匹配工具包。它依据 Levenshtein Distance 算法 计算两个序列之间的差异。Levenshtein Distance 算法，又叫 Edit Distance 算法，是指两个字符串之间，由一个转成另一个所需的最少编辑操作次数。许可的编辑操作包括将一个字符替换成另一个字符，插入一个字符，删除一个字符。一般来说，编辑距离越小，两个串的相似度越大。fuzzywuzzy 模块，不仅可用于计算两个字符串之间的相似度，而且还提供排序接口能从大量候选集中找到最相似的句子。我们导入了两个模块：fuzz，process，fuzz 主要用于两字符串之间匹配，process 主要用于搜索排序。

4.2 实验数据来源

选取了出题方的附件 2 中 70%的数据、附件 3 和附件 4 中的全部数据，还有我方从制作出的模型中整理出来的数据。

五、模型评价

5.1 模型优点

1.模型结果较为直观
2.模型可有效减少人工工作量
3.在热点问题挖掘中，利用字符串模糊匹配，能较为高效的得出热点问题归类。

5.2 模型缺点

1.在一级标签分类模型中，随着标签的增加数据量也会随着增加，代码的运行速度便会变慢。
2.在 F_1-score 中，因随机分配训练集和测试集，如果遇到一级标签较少的话精度便会降低。
3.在热点问题挖掘中，进行的字符串模糊匹配及输出相似度中，存在一部分是同一主题但相似度不高于 60%的情况，在模糊匹配中应该加强改进。
4.在答复意见质量评分模型中可解释性主观因素较强，且完整性中 text2vec 不能做到完全匹配。

六、参考文献

[1] https://github.com/fxsjy/jieba
[2] 自然语言处理-text2vec
https://www.jianshu.com/p/ed8dc1fa2963
[3] 用 Python 进行简单的文本相似度分析
https://blog.csdn.net/xiexf189/article/details/79092629?ops_request_misc=%257B%2522request%255Fid%2522%253A%252215877932681919516250438 1%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.57662%2522%257D&request_id=158779326819195162504381&biz_id=0&utm_source=distribute.pc_search_result.none-task-blog-2~all~first_rank_v2~rank_v25-2
[4] TF-IDF 及其算法
https://www.cnblogs.com/biyeymyhjob/archive/2012/07/17/2595249.html
[5] 朴素贝叶斯以及三种常见模型推导
https://www.jianshu.com/p/b6cadf53b8b8
[6] pandas
https://baike.baidu.com/item/pandas/17209606?fr=aladdin
[7] http://radimrehurek.com/gensim/intro.html
[8] Python fuzzywuzzy 模块 模糊字符串匹配详细用法
https://blog.csdn.net/sunyao_123/article/details/76942809?ops_request_misc=&request_id=&biz_id=102&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0