基于智慧政务的群众留言数据挖掘分析

摘要：

近年来，随着网络、科技的发展，网络问政平台逐渐成为了政府了解民意，深入民心的重要渠道，群众通过网络，将自己遇到的问题反映给政府，自然语言处理技术的智慧处理系统变成了社会治理创新发展的新趋势。本文针对群众留言分类、热点问题挖掘、答复意见评价等问题，首先对数据进行了预处理——群众留言“去重”、“去空”、中文分词、停用词过滤等。之后本文主要进行了两个方面的数据挖掘工作，一方面是利用 SLTM 模型，经过不断训练，不断提升文本分类的准确性；另一方面是利用了 TF-IDF 和 K-means 的算法进行热点问题挖掘和答复意见评价，以便政府更好的了解民情、汇聚民智、凝聚民气。

关键词：智慧政务、SLTM、TF-IDF、K-means

目录

1. 目标分解

本次数据挖掘与建模分析主要针对“智慧政务”中群众问政留言记录及相关部门对留言的答复意见等文本信息，采用不同的文本挖掘模型以期达到以下三个目标：

（1） 利用网络问政平台留言及人工分类标签数据，建立 LSTM 留言一级标签分类模型，并使用 F-score 对分类模型效果进行检测，从而实现网络平台留言自动化分类。

（2） 根据文本信息，采用聚类方法识别统一话题的不同留言，并设计指标识别某一时段内群中集中反映的热点问题，并按照要求整理相应的热点问题与留言明细。

（3） 针对相关部门的答复意见，设计指标衡量答复的相关性、完整性、可解释性等角度的答复意见的评估体系。

2. 分析方法与过程

2.1 分目标流程

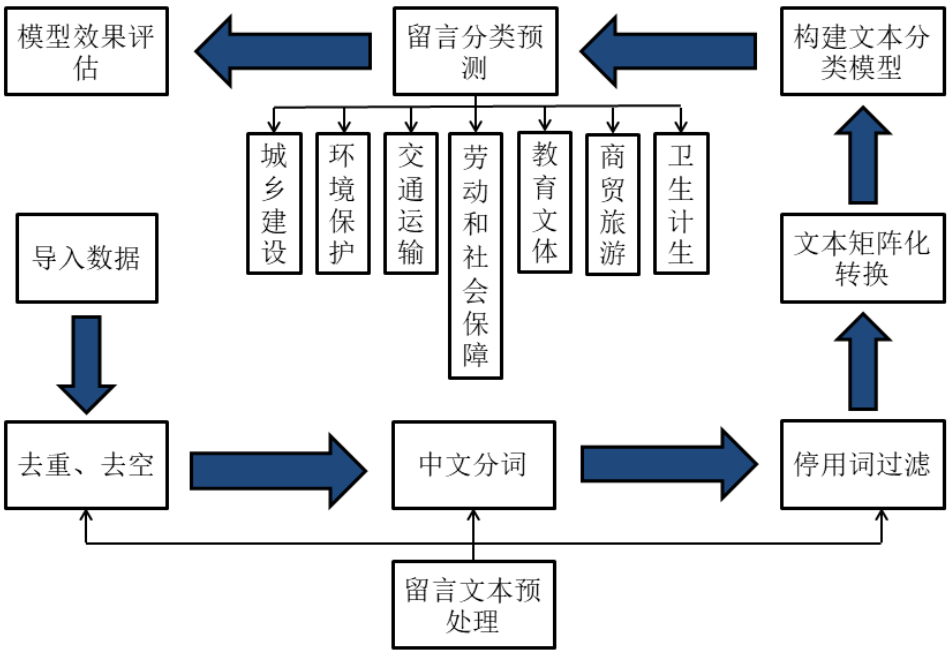

图 1 群众留言分类图

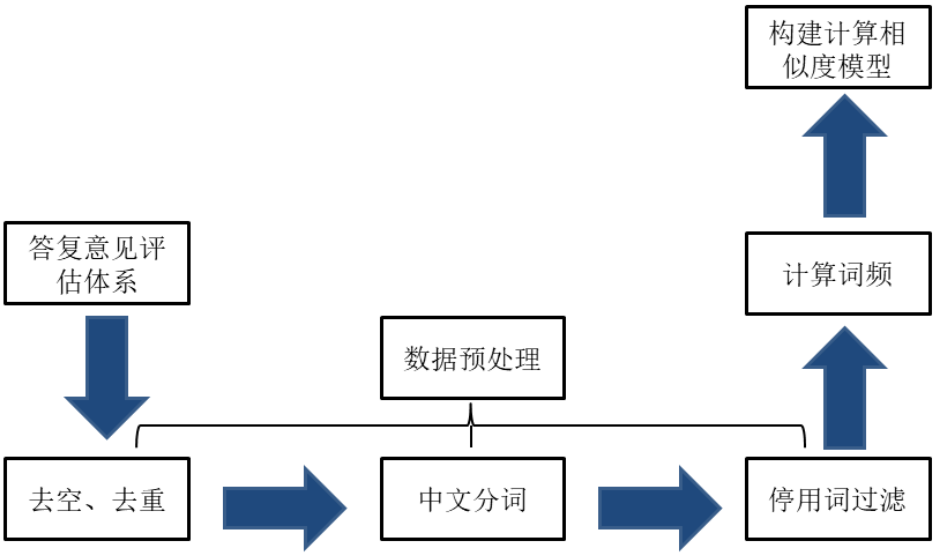

图 2 热点问题挖掘图

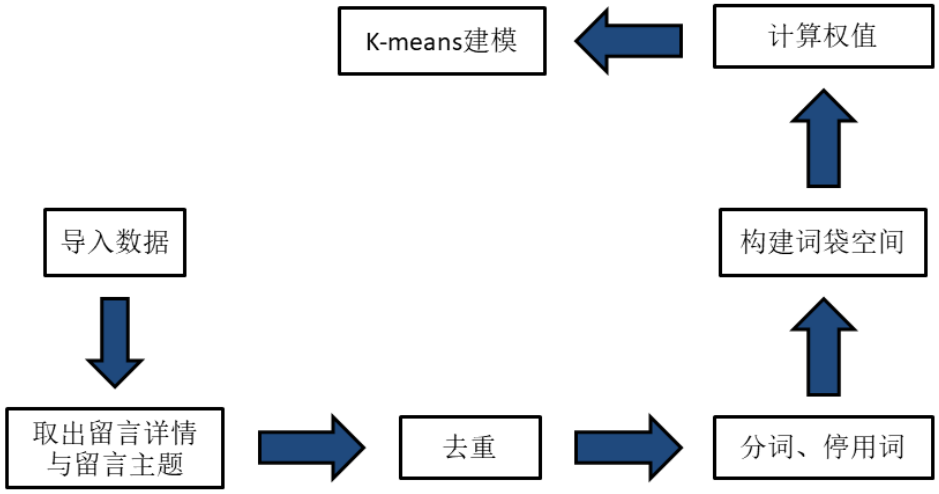

图 3 答复意见评价图

针对“智慧政务”中群众问政留言记录及相关部门对留言的答复意见等文本数据分析与挖掘，重点包括以下 5 个步骤：

步骤一：数据预处理，赛题所给的数据中有可能存在“脏数据”影响分析结果，所以需要对数据进行预处理。包括[1]去除特殊符号、重复值、空值等，文本拼接等；[2] 对留言数据进行中文分词，将留言主题、留言详情划分成多个词语进行分析；[3] 进行停用词过滤；[4]针对目标一还需进行一级标签分类编码。

步骤二：文本矩阵转化，使用 LSTM 模型对留言进行一级标签分类或者是使用聚类分析识别统一话题的留言需要将文本词语全部转化为词向量。所以在目

标一留言分类和目标二热点话题识别本文均进行文本矩阵转化操作。

步骤三：指标设计，针对目标二热点话题识别本文根据留言主题参照并改建营销学 RFM 模型设计了相应反映热度的指标，如问题留言条数、问题持续时长、问题留言时间最短间隔、最近一次问题留言间隔时长等，综合衡量问题留言的热度；针对目标三本文根据经验以及参考文献，设计了衡量相关部分回复意见的相关性、完整性、可解释性的质量评估体系。

步骤四：构建模型及模型效果评估。针对目标二留言分类，本文建立了 LSTM 分类模型，并根据混淆矩阵和 F-score 对模型效果进行评估与模型改进。

步骤五：结果分析。根据分析结果提出合理的建议与意见。

2.2 具体步骤

2.2.1 目标一：群众留言分类

1. 导入数据与数据预处理

导入群众问政留言记录数据，其中包含留言编号、留言用户、留言主题、留言时间、留言详情、一级标签 6 列数据。

```
os.getcwd()
path='E:\\项目\\C题全部数据\\'
os.chdir(path)
data=pd.read_excel('附件2.xlsx',encoding='utf-8',sep = '','')
data.head()
```

	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 4 导入数据

样本数据存在空字符、换行符、重复数据等问题，数据不规范。需要对文本进行处理，主要从以下 4 个方面对数据进行预处理。

（1）去除特殊符号

对于群众问政留言记录的文本数据中，本文发现存在空白符、换行符等特殊符号的文本内容，将其进行替换，如图 5 所示。

```
data['留言详情'] = data['留言详情'].str.replace('\n','')
data['留言详情'] = data['留言详情'].str.replace('\t','')
data
```

	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	A3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...	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	位于书院路主干道的在水一方大厦一楼至四楼人为拆除水、电等设施后，烂尾多年，用护栏围着，不但占...	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	尊敬的领导：A1区苑小区位于A1区火炬路，小区物业A市程明物业管理有限公司，未经小区业主同意...	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	A1区A2区华庭小区高层为二次供水，楼顶水箱长年不洗，现在自来水龙头的水严重霉味，大家都知道...	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	A1区A2区华庭小区高层为二次供水，楼顶水箱长年不洗，现在自来水龙头的水严重霉味，大家都知道...	城乡建设

图 5 去除空白符、换行符

（2）文本拼接

针对留言主题与留言言情，为方便后期分词处理，本文将该两者拼接成“留言”变量。

```
detial=data.留言详情
theme=data.留言主题
dat1=data.一级标签
data1=pd.concat([dat1,theme],axis=1)
data1.columns=['一级标签','留言']
data2=pd.concat([dat1,detial],axis=1)
data2.columns=['一级标签','留言']
data_new=data1.append(data2)
```

```
data_new
```

	一级标签	留言
0	城乡建设	A市西湖建筑集团占道施工有安全隐患
1	城乡建设	A市在水一方大厦人为烂尾多年，安全隐患严重
2	城乡建设	投诉A市A1区苑物业违规收停车费
3	城乡建设	A1区蔡锷南路A2区华庭楼顶水箱长年不洗
4	城乡建设	A1区A2区华庭自来水好大一股霉味

图 6 文本拼接

（3）去除重复值、空值

```
#查看缺失值
print("在一级标签列中总共有 %d 个空值." % data_new['一级标签'].isnull().sum())
print("在 留言列中总共有 %d 个空值." % data_new['留言'].isnull().sum())
data_new[data_new.isnull().values==True]
data_new = data_new[pd.notnull(data_new['留言'])]
```

```
在一级标签列中总共有 0 个空值.
在 留言列中总共有 0 个空值.
```

图 7 去除空值

```
#去重
data_new.drop_duplicates()
```

	一级标签	留言
0	城乡建设	A市西湖建筑集团占道施工有安全隐患
1	城乡建设	A市在水一方大厦人为烂尾多年，安全隐患严重
2	城乡建设	投诉A市A1区苑物业违规收停车费
3	城乡建设	A1区蔡锷南路A2区华庭楼顶水箱长年不洗
4	城乡建设	A1区A2区华庭自来水好大一股霉味

图 8 去除重复值

（4）中文分词

中文分词指的是通过某种特定的规则，将中文文本切分成一个一个单独的词。本文使用 jieba 分词器。主要程序如图 9 所示：

```
#中文分词
jieba.load_userdict('newdict1.txt')
detial_cut=data_new['留言'].apply(lambda x:' '.join(jieba.lcut(x)))
```

图 9 中文分词

分词结果示例：

分词前：A 市西湖建筑集团占道施工有安全隐患

分词后：A 市/西湖/建筑集团/占道施工/有/安全隐患

从上述结果可以看出，本文已经将群众留言切分成一个个的词语。

（5）停用词过滤

并非所有的分词结果都可以作为特征词，里面还有一些包含的信息量很低甚至没有信息量的词语，需要将它们过滤掉，否则将会影响下文的分析的正确率。在信息检索中，为节省存储空间和提高搜索效率，在处理自然语言之前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）。

本文采用了“停用词表”的过滤方法，

代码如下所示：

```
#停用词
stopwords=pd.read_table('stopwords_zh.txt',header=None,sep='\s',engine='python')
stopwords=[' ']+list(stopwords[0])
detial_after_stop=detial_cut.apply(lambda x: [i for i in x if i not in stopwords])
```

图 10 停用词表

（6）一级标签转编码

便于后期建立分类模型，本文将一级标签进行编码，转化为 lable_id（0 到 6），如图 11 所示：

```
dada=data_new.一级标签
data_new=pd.concat([dada,detial_after_stop],axis=1)
data_new['lable_id'] = data_new['一级标签'].factorize()[0]
lable_id_df = data_new[['一级标签', 'lable_id']].drop_duplicates().sort_values('lable_id').reset_index(drop=True)
lable_to_id = dict(lable_id_df.values)
id_to_lable = dict(lable_id_df[['lable_id', '一级标签']].values)
data_new.sample(10)
```

图 11 将一级标签数字化

2. 文本矩阵转化

（1）词向量化概述

文本矩阵转化的第一步就是词向量化，顾名思义，词向量化即用空间向量模型表示各个词语，进而提高计算机对自然语言的处理能力。词向量具有良好的语义特性，是表示词语特征的常用方式。文本分类分析中把对文本内容的处理简化成对一定长度的向量的处理时，通常使用较低维度的空间向量来表示词语的特征，避免数据维数灾难。词向量中每一维的值代表一个具有一定的语义和语法上解释的特征。

词向量化后便可以将评论的文本数据转化向量矩阵了。通常情况下，我们将词语 w 映射到 n 维空间向量，即 $w \in R^n$，一个文本或者句子中含有 m 个词语，把这 m 个 n 维空间向量堆放在一起，就得到整个文本或句子的空间向量模型——一个词向量矩阵 $L \in R^{mn}$。例如给定句子 c 含有 m 个词语，$1 < i < n$，W_i 为句子 c 的空间向量矩阵 L 中的第 K_i 列，即可 $W_i = Le_{k_i} \in R^n$，$e_{k_i} \in R^m$，且除了第 k_i 个分量为 1，其余分量均为 0。

将一个文本或者一句评论映射成一个词向量矩阵后，即将中文文本数据转化成计算机可以识别的信息格式，继而利用基于递归自编码的深度学习方法进行文本分类分析。

（2）文本矩阵转化过程

通过编写程序产生随机的向量词表，每个词对应一个唯一的词标识号和词向量，如图 12 所示

```
# 设置最频繁使用的50000个词
MAX_NB_WORDS = 50000
# 每条cut最大的长度
MAX_SEQUENCE_LENGTH = 250
# 设置Embeddingceng层的维度
EMBEDDING_DIM = 100

tokenizer = Tokenizer(num_words=MAX_NB_WORDS, filters='!"#$%&()*+,-./:;<=>?@[\]^_`{|}~', lower=True)
tokenizer.fit_on_texts(data_new['留言'].values)
word_index = tokenizer.word_index
print('共有 %s 个不相同的词语.' % len(word_index))
```

```
共有 4592 个不相同的词语.
```

```
X = tokenizer.texts_to_sequences(data_new['留言'].values)
#填充X,让X的各个列的长度统一
X = keras.preprocessing.sequence.pad_sequences(X, maxlen=MAX_SEQUENCE_LENGTH)

#多类标签的onehot展开
Y = pd.get_dummies(data_new['lable_id']).values

print(X.shape)
print(Y.shape)
```

```
(18420, 250)
(18420, 7)
```

图 12 矩阵转化

3. LSTM 模型与效果评估

（1）LSTM 模型概述

本文将建立 LSTM 模型对留言进行一级标签分类。LSTM 由 Hochreiter & Schmidhuber (1997) 提出，并在近期被 Alex Graves 进行了改良和推广。

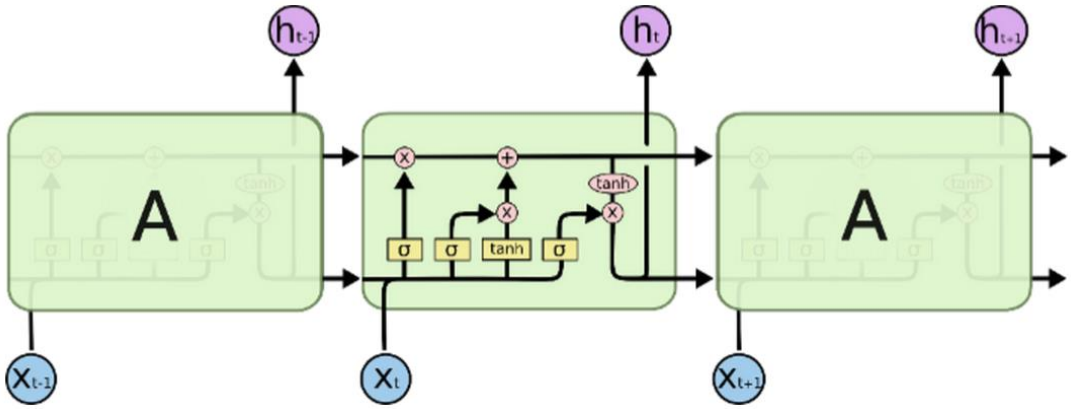

图 13 LSTM 模型

LSTM 中的重复模块包含四个交互的层。

一个 LSTM cell 有 3 个门，分别叫做遗忘门（f 门），输入门（i 门）和输出门（o 门）。要注意的是输出门的输出 ot 并不是 LSTM cell 最终的输出，LSTM cell 最终的输出是 ht 和 ct。

这三个门就是上图中三个标着 σ 的黄色的框。sigmoid 层输出 0-1 的值，表

示让多少信息通过，1 表示让所有的信息都通过。

LSTM 的输入： C_{t-1}, h_{t-1} 和 x_t

LSTM 的输出： h_t 、 C_t

◆ 忘记门：扔掉信息(细胞状态)

$$f_t = \sigma(W_f \bullet [h_{t-1}, x_{t-1}] + b_f)$$

第一步是决定从细胞状态里扔掉什么信息（也就是保留多少信息）。将上一步细胞状态中的信息选择性的遗忘 。

实现方式：通过 sigmoid 层实现的“忘记门”。以上一步的 h_{t-1} 和这一步的 x_t 作为输入，然后为 C_{t-1} 里的每个数字输出一个 0-1 间的值，记为 f_t，表示保留多少信息（1 代表完全保留，0 表示完全舍弃）

◆ 输入层门：存储信息（细胞状态）

$$i_t = \sigma(W_i \bullet [h_{t-1}, x_{t-1}] + b_i)$$

第二步是决定在细胞状态里存什么。将新的信息选择性的记录到细胞状态中。 实现方式：包含两部分，

sigmoid 层（输入门层）决定我们要更新什么值，这个概率表示为 i_t；

tanh 层创建一个候选值向量 $\tilde{C}_t$ ，将会被增加到细胞状态中。 我们将会在下一步把这两个结合起来更新细胞状态。

◆ 更新细胞状态（细胞状态）

$$\tilde{C}_t = \tanh(W_C \bullet [h_{t-1}, x_{t-1}] + b_C)$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

注意上面公式中的*是对应元素乘，而不是矩阵的乘法

更新旧的细胞状态 实现方式： f_t 表示忘记上一次的信息 C_{t-1} 的程度，i_t 表示要将候选值 $\tilde{C}_t$ 加入的程度， 这一步我们真正实现了移除哪些旧的信息（比如一句话中上一句的主语），增加哪些新信息，最后得到了本细胞的状态 C_t.

◆ 输出（隐藏状态）

$$o_t = \sigma(W_o \bullet [h_{t-1}, x_{t-1}] + b_o)$$
$$h_t = o_t * \tanh(C_t)$$

最后，我们要决定作出什么样的预测。 实现方式：

我们通过 sigmoid 层（输出层门）来决定输出的本细胞状态 C_t 的哪些部分；

然后我们将细胞状态通过 tanh 层（使值在-1~1 之间），然后与 sigmoid 层的输出相乘得到最终的输出 h_t 。

（2）训练集与测试集拆分

经上述步骤数据预处理后，各级分类标签数据统计如表 1 所示：

表 1 预处理后的留言数量

一级标签	城乡建设	劳动和社会保障	教育文体	商贸旅游	环境保护	卫生计生	交通运输
数量	4018	3938	3178	2430	1876	1754	1226

根据数据统计结果，本文将数据划分为训练集和测试集，测试集占比 10%。

```
#拆分训练集和测试集(留言详情)
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.10, random_state = 42)
print(X_train.shape,Y_train.shape)
print(X_test.shape,Y_test.shape)
```

```
(16578, 250) (16578, 7)
(1842, 250) (1842, 7)
```

图 14 测试集和训练集

（3）定义 LSTM 序列模型

[1] 模型的第一次是嵌入层(Embedding)，它使用长度为 100 的向量来表示每一个词语

[2] SpatialDropout1D 层在训练中每次更新时， 将输入单元的按比率随机设置为 0， 这有助于防止过拟合

[3] LSTM 层包含 100 个记忆单元，输出层为包含 10 个分类的全连接层

由于是多分类，所以激活函数设置为'softmax'，损失函数为分类交叉熵 categorical_crossentropy

```
#定义模型
model = Sequential()
model.add(Embedding(MAX_NB_WORDS, EMBEDDING_DIM, input_length=X.shape[1]))
model.add(SpatialDropout1D(0.2))
model.add(LSTM(100, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(7, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
print(model.summary())
```

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, 250, 100)          5000000
_________________________________________________________________
spatial_dropout1d_2 (Spatial (None, 250, 100)          0
_________________________________________________________________
lstm_2 (LSTM)                (None, 100)               80400
_________________________________________________________________
dense_2 (Dense)              (None, 7)                 707
=================================================================
Total params: 5,081,107
Trainable params: 5,081,107
Non-trainable params: 0
_________________________________________________________________
None
```

图 15 LSTM 模型

```
#训练模型(留言详情)
epochs = 5
batch_size = 64

history = model.fit(X_train, Y_train, epochs=epochs, batch_size=batch_size,validation_split=0.1,
                    callbacks=[EarlyStopping(monitor='val_loss', patience=3, min_delta=0.0001)])
```

```
Train on 14920 samples, validate on 1658 samples
Epoch 1/5
14920/14920 [==============================] - 131s 9ms/step - loss: 1.3352 - accuracy: 0.5161 - val_loss: 0.8798 - val_accuracy: 0.7201
Epoch 2/5
14920/14920 [==============================] - 134s 9ms/step - loss: 0.7832 - accuracy: 0.7531 - val_loss: 0.6782 - val_accuracy: 0.7895
Epoch 3/5
14920/14920 [==============================] - 137s 9ms/step - loss: 0.6506 - accuracy: 0.7997 - val_loss: 0.6433 - val_accuracy: 0.7955
Epoch 4/5
14920/14920 [==============================] - 137s 9ms/step - loss: 0.5261 - accuracy: 0.8387 - val_loss: 0.5742 - val_accuracy: 0.8239
Epoch 5/5
14920/14920 [==============================] - 139s 9ms/step - loss: 0.5689 - accuracy: 0.8223 - val_loss: 0.5833 - val_accuracy: 0.8191
```

图 16 模型训练

（4）模型效果评估

本步骤使用 F-Score 对分类方法进行评价，最终训练数据集的 F 值为 80.6%，测试数据集的 F 值为 0.811。

如图 17 所示：

```
# 获取confusion matrix
cm = confusion_matrix(Y_test, y_pred, labels=range(7))
cm = cm.astype(np.float32)
FP = cm.sum(axis=0) - np.diag(cm)
FN = cm.sum(axis=1) - np.diag(cm)
TP = np.diag(cm)
TN = cm.sum() - (FP + FN + TP)
# Sensitivity, hit rate, recall, or true positive rate
TPR = TP / (TP + FN)
# Precision or positive predictive value
PPV = TP / (TP + FP)
# Overall accuracy
ACC = (TP + TN) / (TP + FP + FN + TN)
# ACC_micro = (sum(TP) + sum(TN)) / (sum(TP) + sum(FP) + sum(FN) + sum(TN))
ACC_macro = np.mean(ACC) # to get a sense of effectiveness of our method on the small classes we computed this average (macro-average)
F1 = (2 * PPV * TPR) / (PPV + TPR)
F1_macro = np.mean(F1)
```

```
F1_macro
```

```
0.81097174
```

图 17 模型效果评估

依据分类结果，本文可视化展示了训练集、测试集混淆矩阵的结果混淆矩阵的主对角线表示预测正确的数量，除主对角线外其余都是预测错误的数量。从上面的混淆矩阵可以看出"劳动和社会保障"类预测最准确，为 88%。“交通运输”预测的错误数量教多，原因可能是因为此类样本量较小。整体分类结果较为满意。

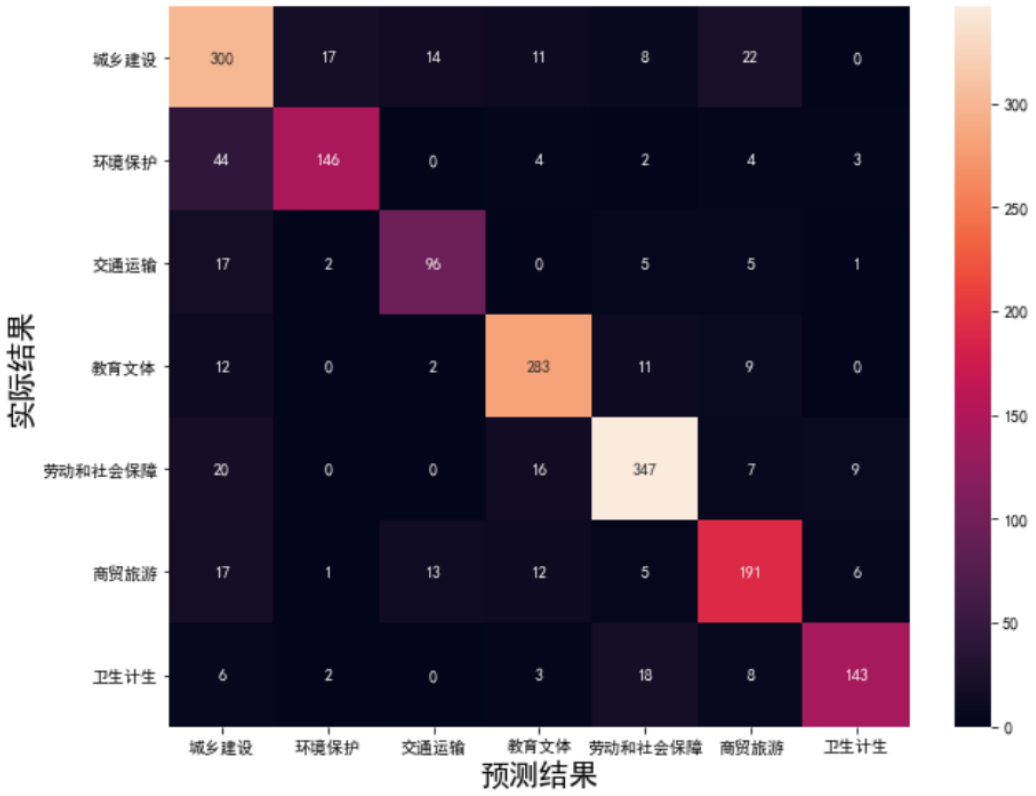

图 18 训练集混淆矩阵结果

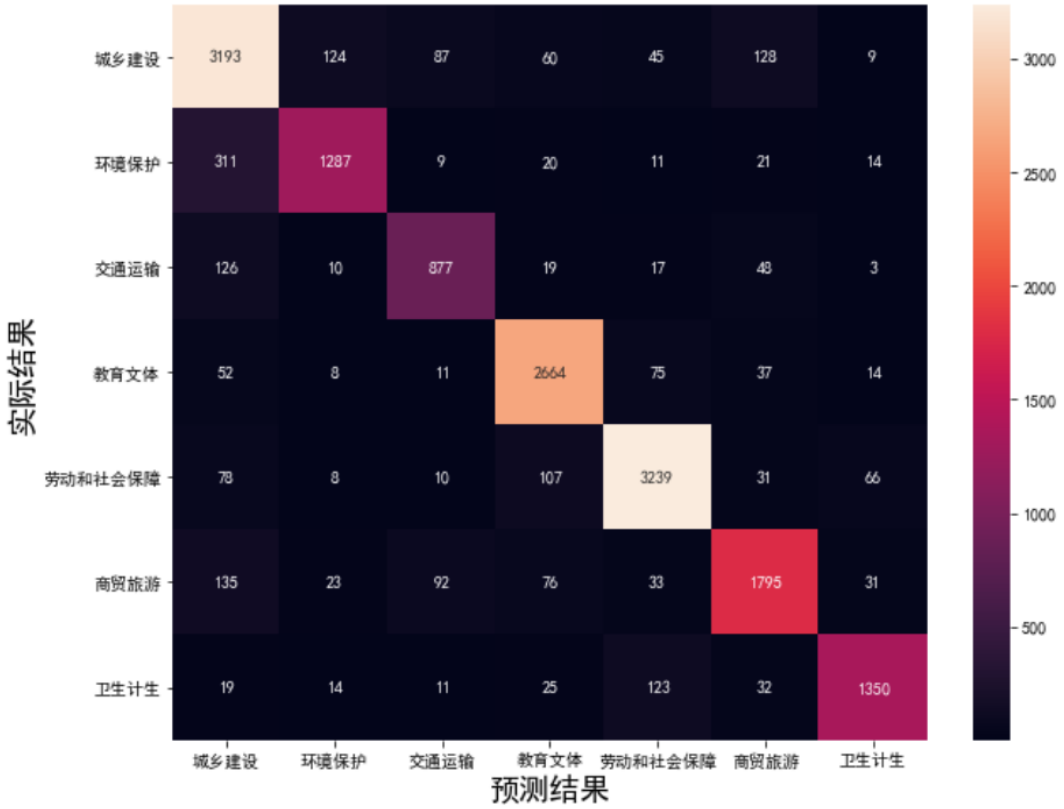

图 19 测试集混淆矩阵结果

2.2.2 目标二：热点问题挖掘

1. 导入数据与数据预处理

导入群众问政留言记录数据，其中包含留言编号、留言用户、留言主题、留言时间、留言详情、反对数、点赞数 7 列数据。

```
import os
import pandas as pd
import jieba
os.getcwd()
```

```
'C:\\Users\\[redacted]\\Desktop\\实践练习'
```

```
data=pd.read_exce[redacted]
data.head(6)
```

	留言编号	留言用户	留言主题	留言时间	留言详情	反对数	点赞数
0	188006	A000102948	A3区一米阳光婚纱艺术摄影是否合法纳税了?	2019/2/28 11:25:05	\n\t\t\t\t\t\n\t\t\t\t\t座落在A市A3区联丰路米兰春天G2栋320，...	0	0
1	188007	A00074795	咨询A6区道路命名规划初步成果公示和城乡门牌问题	2019/2/14 20:00:00	\n\t\t\t\t\t\n\t\t\t\t\tA市A6区道路命名规划已经初步成果公示文件，...	0	1
2	188031	A00040066	反映A7县春华镇金鼎村水泥路、自来水到户的问题	2019/7/19 18:19:54	\n\t\t\t\t\t\n\t\t\t\t\t本人系春华镇金鼎村七里组村民，不知是否有相关...	0	1
3	188039	A00081379	A2区黄兴路步行街大古道巷住户卫生间粪便外排	2019/8/19 11:48:23	\n\t\t\t\t\t\n\t\t\t\t\t靠近黄兴路步行街，城南路街道、大古道巷、一步...	0	1
4	188059	A00028571	A市A3区中海国际社区三期与四期中间空地夜间施工噪音扰民	2019/11/22 16:54:42	\n\t\t\t\t\t\n\t\t\t\t\tA市A3区中海国际社区三期四期中间，即蓝天璞...	0	0
5	188073	A909164	A3区麓泉社区单方面改变麓谷明珠小区6栋架空层使用性质	2019/3/11 11:40:42	\n\t\t\t\t\t\n\t\t\t\t\t作为麓泉社区麓谷明珠小区6栋居民，我们近期感...	0	0

图 20 导入数据

样本数据存在空字符、换行符、重复数据等问题，数据不规范。需要对文

本进行处理，主要从以下 4 个方面对数据进行预处理。

（1）去除特殊符号并进行文本拼接

对于群众问政留言记录的文本数据中，本文发现存在空白符、换行符等特殊符号的文本内容，将其进行替换，针对留言主题与留言言情，为方便后期分词处理，本文将该两者拼接成“data0”。如图 21 所示。

```
#取出留言主题与留言详情合并成data0
data1=data.iloc[:,2]
data2=data.iloc[:,4]
data0=data1+data2
data0=data0.str.replace('\n','')
data0=data0.str.replace('\t','')
data0=data0.str.replace('  ','')
data0=data0.str.replace(' ','')
data0.head()
```

```
0    A3区一米阳光婚纱艺术摄影是否合法纳税了？座落在A市A3区联丰路米兰春天G2栋320，一家名...
1    咨询A6区道路命名规划初步成果公示和城乡门牌问题A市A6区道路命名规划已经初步成果公示文件，...
2    反映A7县春华镇金鼎村水泥路、自来水到户的问题本人系春华镇金鼎村七里组村民，不知是否有相关水...
3    A2区黄兴路步行街大古道巷住户卫生间粪便外排靠近黄兴路步行街，城南路街道、大古道巷、一步两搭...
4    A市A3区中海国际社区三期与四期中间空地夜间施工噪音扰民A市A3区中海国际社区三期四期中间，...
dtype: object
```

图 21 去除空白符、换行符并进行文本拼接

（2）停用词过滤

并非所有的分词结果都可以作为特征词，里面还有一些包含的信息量很低甚至没有信息量的词语，需要将它们过滤掉，否则将会影响下文的分析的正确率。在信息检索中，为节省存储空间和提高搜索效率，在处理自然语言之前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）。

本文采用了“停用词表”的过滤方法，代码如下所示：

```
#分词，去除停用词
datacut = jieba.cut(data0)
stopwords=pd.read_table('[illegible]Desktop\\stopwords.txt',header=None,sep='\s',engine='python')
stopwords=[ ]+list(stopwords)
datalcut0=datalcut.apply(lambda x:[i for i in x if i not in stopwords])
```

图 22 停用词表

2.构建词袋并将出现次数转化权值

将留言详情与留言主题中的词语跟数字一一对应为，如图 23 所示：

```
#构建词袋空间
def get_all_vector(file_path,datalcut0):
    names = [ os.path.join(file_path,f) for f in os.listdir(file_path) ]
    posts = [ open(name).read() for name in names ]
    docs = []
    word_set = set()
    for post in posts:
        doc = del_stop_words(post,stop_words_set)
        docs.append(doc)
        word_set |= set(doc)
    word_set = list(word_set)
    docs_vsm = []
    for doc in docs:
        temp_vector = []
        for word in word_set:
            temp_vector.append(doc.count(word) * 1.0)
        docs_vsm.append(temp_vector)

    docs_matrix = np.array(docs_vsm)
```

图 23 词袋

TF 代表了词语在文档中出现的频率，当进行索引的时候，词语出现频率较高的文本，匹配度也会较高，但是某些停止词，例如 to 在文本中会出现相当多的次数，但这对匹配并没有起到很好的索引作用，因此需要引入另一个度量值 IDF（逆文本频率）

$$IDF = \log \frac{N}{N(x)}$$

其中 N 为语料库中文本的总数，N(x)为文本中出现单词 x 的文本数量。

某些特殊情况下，x 并未出现在语料库中（所有文本），则需要考虑将公式平滑为：

$$IDF = \log \frac{N+1}{N(x)+1} + 1$$

最终的 TF-IDF 值为：

TF−IDF(x)=TF(x)*IDF(x)

```
#出现次数转化权值tf-idf
def get_all_vector(file_path, stop_words_set):
    names = [ os.path.join(file_path,f) for f in os.listdir(file_path) ]
    posts = [ open(name).read() for name in names ]
    docs = []
    word_set = set()
    for post in posts:
        doc = del_stop_words(post, stop_words_set)
        docs.append(doc)
        word_set |= set(doc)
        #print len(doc),len(word_set)

    word_set = list(word_set)
    docs_vsm = []
    #for word in word_set[:30]:
        #print word.encode("utf-8"),
    for doc in docs:
        temp_vector = []
        for word in word_set:
            temp_vector.append(doc.count(word) * 1.0)
        #print temp_vector[-30:-1]
        docs_vsm.append(temp_vector)

    docs_matrix = np.array(docs_vsm)
```

图 24 tf-idf

3.k-means 聚类

（1）K-means 算法概述

K-Means 算法，也称 K-均值，是一种使用广泛的最基础的聚类算法

算法步骤为

[1]. 选择初始化的 k 个样本作为初始聚类中心 $c = \{a_1, a_2, \ldots, a_k\}$

[2]. 针对数据集中每个样本 x_i 计算它到 k 个聚类中心的距离并将其分到距离最小的聚类中心所对应的类中

[3]. 针对每个类别 a_j，重新计算它的聚类中心 $a_j = \frac{1}{|c_i|}\sum_{x \in c_i} x$（即属于该类的所有样本的质心）

[4]. 重复上面两步操作，直到达到某个终止条件

[5]. 迭代次数、最小平方误差 MSE，簇中心变化率

（2）K-means 聚类

```
#k-means聚类
def gen_sim(A,B):
    num = float(np.dot(A,B.T))
    denum = np.linalg.norm(A) * np.linalg.norm(B)
    if denum == 0:
        denum = 1
    cosn = num / denum
    sim = 0.5 + 0.5 * cosn
    return sim
def randCent(dataSet, k):
    n = shape(dataSet)[1]
    centroids = mat(zeros((k,n)))#create centroid mat
    for j in range(n):#create random cluster centers, within bounds of each dimension
        minJ = min(dataSet[:,j])
        rangeJ = float(max(dataSet[:,j]) - minJ)
        centroids[:,j] = mat(minJ + rangeJ * random.rand(k,1))
    return centroids

def kMeans(dataSet, k, distMeas=gen_sim, createCent=randCent):
    m = shape(dataSet)[0]
    clusterAssment = mat(zeros((m,2)))#create mat to assign data points
                                      #to a centroid, also holds SE of each point
    centroids = createCent(dataSet, k)
    clusterChanged = True
    counter = 0
    while counter <= 50:
        counter += 1
        clusterChanged = False
        for i in range(m):#for each data point assign it to the closest centroid
            minDist = inf;
            minIndex = -1
            for j in range(k):
                distJI = distMeas(centroids[j,:],dataSet[i,:])
                if distJI < minDist:
                    minDist = distJI;
                    minIndex = j
```

图 25 k-means 聚类

2.2.3 目标三：答复意见质量评估体系

1. 导入数据与数据预处理

导入群众问政留言记录数据，其中包含留言编号、留言用户、留言主题、留言时间、留言详情、答复意见、答复时间 7 列数据。

```
os.getcwd()
path='C:\\Users\\lenovo\\Desktop\\泰迪\\C题全部数据'
os.chdir(path)
data=pd.read_excel('附件4.xlsx',encoding='utf-8',sep = ",")
data.head()
```

	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
0	2549	A00045581	A2区景荟华苑物业管理有问题	2019/4/25 9:32:09	\n\t\t\t\t\n\t\t\t\t2019年4月以来，位于A市A2区桂花坪街道...	现将网友在平台《问政西地省》栏目向胡华衡书记留言反映“A2区景荟花苑物业管理有问题”的调查核...	2019/5/10 14:56:53
1	2554	A00023583	A3区潇楚南路洋湖段怎么还没修好?	2019/4/24 16:03:40	\n\t\t\t\t\n\t\t\t\t潇楚南路从2018年开始修，到现在都快一年了...	网友“A00023583”：您好！针对您反映A3区潇楚南路洋湖段怎么还没修好的问题,A3区洋...	2019/5/9 9:49:10
2	2555	A00031618	请加快提高A市民营幼儿园老师的待遇	2019/4/24 15:40:04	\n\t\t\t\t\n\t\t\t\t地处省会A市民营幼儿园众多，小孩是祖国的未来...	市民同志：你好！您反映的“请加快提高民营幼儿园教师的待遇”的来信已收悉。现回复如下：为了改善...	2019/5/9 9:49:14
3	2557	A000110735	在A市买公寓能享受人才新政购房补贴吗?	2019/4/24 15:07:30	\n\t\t\t\t\n\t\t\t\t尊敬的书记：您好！我研究生毕业后根据人才新政...	网友“A000110735”：您好！您在平台《问政西地省》上的留言已收悉，市住建局及时将您反...	2019/5/9 9:49:42
4	2574	A0009233	关于A市公交站点名称变更的建议	2019/4/23 17:03:19	\n\t\t\t\t\n\t\t\t\t建议将“白竹坡路口”更名为“马坡岭小学”，原...	网友“A0009233”，您好，您的留言已收悉，现将具体内容答复如下：关于来信人建议“白竹坡...	2019/5/9 9:51:30

图 26 导入数据

样本数据存在空字符、换行符、重复数据等问题，数据不规范。需要对文本进行处理，主要从以下 4 个方面对数据进行预处理。

（1）去除特殊符号

对于群众问政留言记录的文本数据中，本文发现存在空白符、换行符等特殊符号的文本内容，将其进行替换，

如图 21 所示。

```
data['留言详情'] = data['留言详情'].str.replace('\n','')
data['留言详情'] = data['留言详情'].str.replace('\t','')
data
```

图 27 去除空白符、换行符

（2）去除重复值、空值

```
#查看重复值
data[data.duplicated()]
data.drop_duplicates()
```

	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
0	2549	A00045581	A2区景荟华苑物业管理有问题	2019/4/25 9:32:09	2019年4月以来，位于A市A2区桂花坪街道的A2区公安分局宿舍区（景荟华苑)出现了一番乱象...	现将网友在平台《问政西地省》栏目向胡华衡书记留言反映"A2区景荟花苑物业管理有问题"的调查核...	2019/5/10 14:56:53
1	2554	A00023583	A3区潇楚南路洋湖段怎么还没修好?	2019/4/24 16:03:40	潇楚南路从2018年开始修，到现在都快一年了，路挖得稀烂用围栏围起，一直不怎么动工，有时候今...	网友"A00023583"：您好！针对您反映A3区潇楚南路洋湖段怎么还没修好的问题,A3区洋...	2019/5/9 9:49:10
2	2555	A00031618	请加快提高A市民营幼儿园老师的待遇	2019/4/24 15:40:04	地处省会A市民营幼儿园众多，小孩是祖国的未来，但民营幼儿园教师一直都是超负荷工作且收入又是所...	市民同志：你好！您反映的"请加快提高民营幼儿园教师的待遇"的来信已收悉。现回复如下：为了改善...	2019/5/9 9:49:14
3	2557	A000110735	在A市买公寓能享受人才新政购房补贴吗?	2019/4/24 15:07:30	尊敬的书记：您好！我研究生毕业后根据人才新政落户A市，想买套公寓，请问购买公寓能否享受研究生...	网友"A000110735"：您好！您在平台《问政西地省》上的留言已收悉，市住建局及时将您反...	2019/5/9 9:49:42

图 28 去除重复值

```
print("在答复意见列中总共有 %d 个空值." % data['留言详情'].isnull().sum())
print("在留言详情列中总共有 %d 个空值." % data['答复意见'].isnull().sum())
data[data.isnull().values==True]
data = data[pd.notnull(data['答复意见'])]
```

```
在答复意见列中总共有 0 个空值.
在留言详情列中总共有 0 个空值.
```

图 29 去除空值

（3）停用词过滤

并非所有的分词结果都可以作为特征词，里面还有一些包含的信息量很低甚至没有信息量的词语，需要将它们过滤掉，否则将会影响下文的分析的正确率。在信息检索中，为节省存储空间和提高搜索效率，在处理自然语言之前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）。

本文采用了“停用词表”的过滤方法，代码如下所示：

```
#停用词
stopwords=pd.read_table('stopwords_zh.txt',header=None,sep='\s',engine='python')
stopwords=[' ']+list(stopwords[0])
detial_stop=detial.apply(lambda x: [i for i in x if i not in stopwords])
```

图 30 停用词表

（6）TF-IDF 模型（计算相似度）

```
#TF-IDF
from nltk.text import TextCollection
from nltk.tokenize import word_tokenize
def compute_tf_idf_similarity(detial_stop: str, answer_stop: str) -> float:
    """
    Compute the mean tf-idf or tf
     similarity for one sentence with multi query words.
    :param query: a string contain all key word split by one space
    :param content: string list with every content relevent to this query.
    :return: average tf-idf or tf similarity.
    """
    sents = [word_tokenize(content), word_tokenize("")]  # add one empty file to smooth.
    corpus = TextCollection(sents)  # 构建语料库

    result_list = []
    for key_word in query.strip(" ").split(" "):
        if compute_type == "tf_idf":
            result_list.append(corpus.tf_idf(key_word, corpus))
        elif compute_type == "tf":
            result_list.append(corpus.tf(key_word, corpus))
        else:
            raise KeyError

    return sum(result_list) / len(result_list)
```

图 31 TF-IDF 模型

（7）构建体系（完整性、可解释性）

相关部门对留言答复意见的质量指标体系

目标层	准则层	指标层
A1相关部门对留言答复意见的质量评级体系	B1官方回应	C1留言主题
		C2留言指向
		C3答复是否解决问题
	B2动态反应	C4积极反应（回答问题）
		C5消极反应（回避问题）
		C6答复时的态度
	B3真实性	C7答复内容是否有相关条例
		C8答复内容是否落实

图 32 TF-IDF 模型

3 结论

总结本次比赛，我们根据群众留言特点，利用构建的 LSTM 模型　进行文本分析，统计分析出群众留言分类，热点问题以及答复意见的评价，实现了本次的挖掘目标：网络平台留言自动化分类、整理相应热点问题与留言明细以及针对相关部门的答复意见设计指标衡量答复的相关性，完整性，可解释性等角度的答复意见的评估体系。

在本次评论数据挖掘分析的过程中，每一步都通过程序实现，进行了大量的数据挖掘分析工作，实验中的每一步都有理有据，各个步骤之间联系密切，条理清晰且系统地完成了本次数据挖掘分析工作。但是在实验过程中依旧遇到了很多瓶颈问题，例如热点问题挖掘，本次实验中的在分词结束后，我们本来想进行每个向量对于其他向量的相似度分析，但是由于数据量的庞大，我们放弃了这个笨办法，使用 k-means 聚类是我们找到的新方法，他的计算相当方便快捷，并且省去了大量的计算步骤。这是一种巧妙地计算方法，我们在做很难解决的问题时，

我们应该先对问题进行分析，多找资料，剖析问题最终的业务诉求。

4. 参考文献

[1]. 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, 027(001):94-99.

[2]. 李芳. 文本挖掘若干关键技术研究[D]. 北京化工大学.

[3]. 周雪忠, 吴朝晖, ZHOU,等. 文本知识发现:基于信息抽取的文本挖掘[J]. 计算机科学, 2003.

[4]. 赵阳. 基于文本挖掘的高铁信号系统车载设备故障诊断[D]. 2015.

[5]. 乔良. 文本挖掘技术研究及其在信息检索中的应用[J]. 教育技术导刊, 2009.

[6]. 曾依灵, 许洪波, 白硕. 改进的 OPTICS 算法及其在文本聚类中的应用[J]. 中文信息学报, 2008(1):51-55.

[7]. 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.

[8]. 谌志群, 张国煊. 文本挖掘与中文文本挖掘模型研究[J]. 情報科學, 2007, 25(7):1046-1051.

[9]. 刘玉琴, 朱东华, 吕琳. 基于文本挖掘技术的产品技术成熟度预测%Technology maturity of product forecasting based on text mining[J]. 计算机集成制造系统, 2008, 014(003):506-510,542.