基于 LSTM、GMM 等技术的智能政务平台评价模型

摘要：

近年来，随着微信、微博、市长信箱、阳光热线等网络问政平台逐步成为政府了解民意、汇聚民智、凝聚民气的重要渠道，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。

于此同时，深度学习与自然语言处理技术的发展也日新月异，在日常生活中的应用也越来越广泛。基于此，本文构建了一个基于 LSTM、GMM 等的智能政务平台评价模型。

首先进行数据预处理。我们对问题给出的数据集进行统计分析，提出该数据集进行处理时的关键挑战，并给出相应的预处理步骤。

第二步构建文本多分类子模型。我们使用深度学习中的 LSTM（Long Short-Term Memory）长短期记忆网络，来进行文本多分类。将上一步预处理的数据进行 LSTM 的建模工作，对数据进行向量化处理，拆分训练集与测试集后建立 LSTM 的序列模型。

第三步构建高频词挖掘子模型。对预处理后的数据利用 TF-IDF 将词语转化为向量，利用 PAC 降维，之后用 GMM 聚类为 50 类，从而自定义建立热度模型。分别取每一类问题点赞数和反对数的计数统计总和、每一类问题的时间跨度进行归一化，并对后者取倒数，分别作为热度分指标，将两个分指标加和后，成为热度模型中的热度指标；

第四步构建评价答复意见子模型。将预处理的数据用 TF-IDF 进行处理，再使用同一词库对数据进行处理，得到稀疏矩阵（0-1 矩阵）。利用欧式距离进行相似度计算，并从相关性、及时性、可解释性、完整性这四个方面建立评价指标。

实验最后分析了模型结果并评估了模型的能力，并简要对模型做了总结与展望，希望未来不断加深相关学习来不断优化模型。

关键词：LSTM、GMM、TF-IDF、NPL、PCA

Evaluation model of intelligent government platform based on LSTM, GMM and other technologies

Abstract:

In recent years, with wechat, microblog, mayor's mailbox, sunshine hotline and other online political platforms gradually becoming an important channel for the government to understand the public opinion, gather people's wisdom and gather people's spirit, the amount of text data related to various social situations and public opinions has been increasing, which has brought great challenges to the work of relevant departments that used to rely mainly on human to divide messages and sort out hot spots.

At the same time, the development of deep learning and natural language processing technology is changing with each passing day, and its application in daily life is more and more extensive. Based on this, this paper constructs an intelligent government platform evaluation model based on LSTM, GMM, etc.

First, data preprocessing. We make statistical analysis of the data set given by the problem, propose the key challenges when processing the data set, and give the corresponding preprocessing steps.

The second step is to build a text multi classification sub model. We use LSTM (long short term memory) network in deep learning to classify texts. The data preprocessed in the previous step is modeled by LSTM, and the data is vectorized. After the training set and test set are split, the sequence model of LSTM is established.

The third step is to build a sub model of high frequency word mining. After preprocessing, TF-IDF is used to transform words into vectors, PAC is used to reduce dimensions, and GMM is used to cluster them into

50 categories, so as to build a custom heat model. Take the sum of the count statistics of the number of likes and objections of each kind of problem, normalize the time span of each kind of problem, and take the reciprocal of the latter as the heat sub index, and add the two sub indexes to become the heat index in the heat model;

The fourth step is to build a sub model of evaluation reply. TF-IDF is used to process the preprocessed data, and then the same thesaurus is used to process the data to get the sparse matrix (0-1 matrix). The similarity is calculated by Euclidean distance, and the evaluation indexes are established from four aspects: relevance, timeliness, interpretability and integrity.

At the end of the experiment, the results of the model are analyzed and the ability of the model is evaluated, and the model is summarized and prospected briefly. It is hoped that in the future, relevant learning will be deepened to optimize the model.

Keywords: LSTM, GMM, TF-IDF, NPL, PCA

目录

一、引言

近年来，随着微信、微博、市长信箱、阳光热线等网络问政平台逐步成为政府了解民意、汇聚民智、凝聚民气的重要渠道，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，可以较为便捷的解决上述问题，将群众的留言进行分类，并利用 NPL 进行热点问题的挖掘，最终建立评价指标对政府等相关部门的答复意见进行评价，对提升政府的管理水平和施政效率具有极大的推动作用。

通过对基于自然语言处理技术的智慧政务系统的理解与认识，本文将立足以上背景，构建基于 Keras、LSTM 模型、GMM 等的智慧政务系统，完成群众与政府之间的留言与回复问答智能交互操作。在完成对题目所给问题集的数据分析以及预处理工作后，该模型进行自身评估，具有良好效果。本文包括引言、系统模型、实验方案、实验结果、总结与展望五个部分。

二、模型框架

为了更高效的进行留言划分和热点整理，我们构建了一种基于 Keras、LSTM 模型、GMM 等的智慧政务系统，完成群众与政府之间的留言与回复问答智能交互操作。该模型主要包括四个部分：数据预处理、文本多分类、高频词挖掘、评价答复意见。

第一步：数据预处理。我们对问题给出的数据集进行统计分析，提出该数据集进行处理时的关键挑战，并给出相应的预处理步骤；

第二步：文本多分类。我们使用深度学习中的 LSTM（Long Short-Term Memory）长短期记忆网络，来进行文本多分类。将上一步预处理的数据进行 LSTM 的建模工作，对数据进行向量化处理，拆分训练集与测试集后建立 LSTM 的序列模型。定义好 LSTM 模型以后，我们开始训练数据，得出结果；

第三步：高频词挖掘。对预处理后的数据利用 TF-IDF 将词语转化为向量，利用 PAC 降维，之后用 GMM 聚类为 50 类，从而建立热度模型。分别取每一类问题点赞数和反对数的计数统计总和、每一类问题的时间跨度进行归一化，并对后者取倒数，分别作为热度分指标，将两个分指标加和后，成为热度模型中的热度指标；

第四步：评价答复意见。将“留言详情”与“答复意见”文件数据进行预处理后，对预处理后的“留言详情”用 TF-IDF 进行处理，再使用同一词库对“答复意见”进行处理，得到稀疏矩阵（0-1 矩阵）。利用欧式距离进行相似度计算，取“答复时间”与“留言时间”的时间差进行归一化来反馈回复的及时性；采用关键词计数的方法来反馈回复的可解释性与完整性。

三、方案介绍

本文在搭建模型前制定了严格的流程操作，在过程中严格执行，层层递进。首先抽取出数据进行数据清理，之后采用 jieba 分词，进行相应的预处理后完成建模的数据准备。在查询比价大量文献方法后确定建模思路进行建模，建模后得出结果进行分析，评估模型后进行评价与优化模型，最终形成本文。

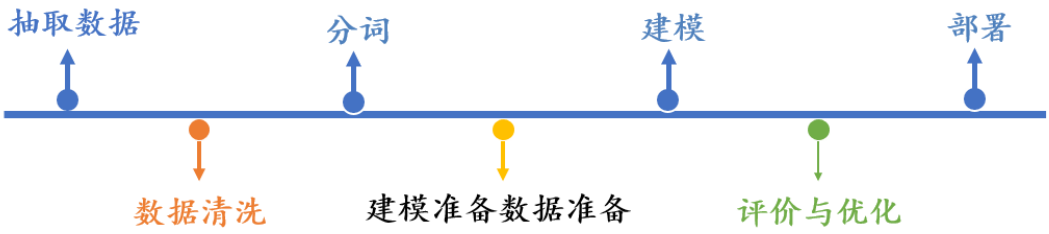

图 本文整体流程图

3.1 数据预处

在我们的模型中，我们主要采用将自然语言处理的问题要转化为机器学习的方式来进行，首先需要将自然语言进行数字化表示。例如在语音处理中，需要将音频文件转化为音频信号向量；在图像处理中，需要将图片文件转化为图片像素矩阵。

（1）去停用词

由于我们的评价内容都是中文，所以要对中文进行一些预处理工作，这包括删除文本中的标点符号，特殊符号，还要删除一些无意义的常用词(stopword)，因为这些词和符号对系统分析预测文本的内容没有任何帮助，反而会增加计算的复杂度和增加系统开销，所有在使用这些文本数据之前必须要将它们清理干净。

中文停用词包含了很多日常使用频率很高的常用词，如：吧，吗，呢，啥等一些感叹词等，这些高频常用词无法反应出文本的主要意思，所以要被过滤掉。

（2）jieba 分词

jieba 分词主要通过词典来进行分词及词性标注，两者使用了一个相同的词典。正因如此，分词的结果优劣将很大程度上取决于词典，虽然使用了 HMM 来进行新词发现。jieba 分词包整体的工作流程如下图所示：

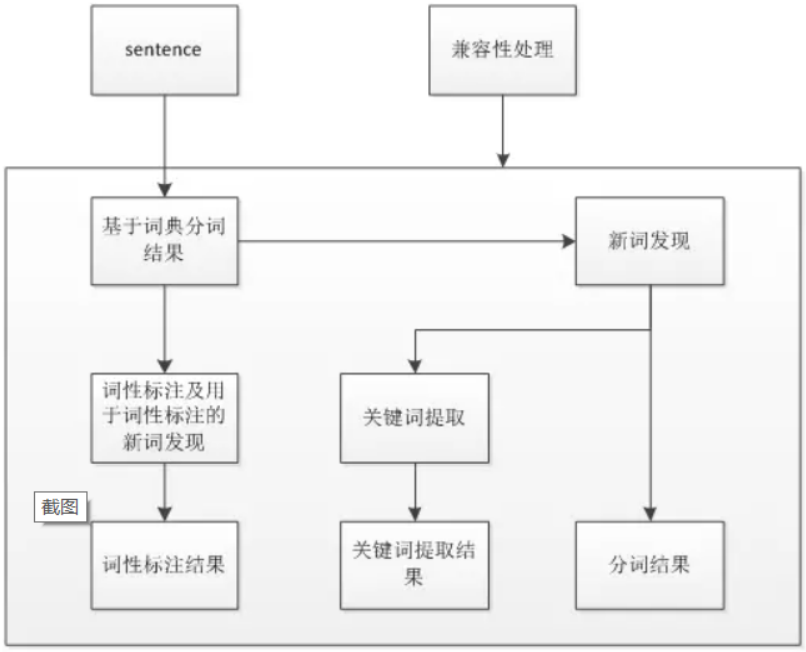

图 jieba 分词流程

除此之外，部分数据进行预处理分词时采用 viterbi 算法，用于寻找最可能的隐藏状态序列(Finding most probable sequence of hidden states)对于一个特殊的隐马尔科夫模型(HMM)及一个相应的观察序列，我们用于找到生成此序列最可能的隐藏状态序列。

我们利用概率的时间不变性，通过避免计算网格中每一条路径的概率来降低问题的复杂度。维特比算法对于每一个状态(t>1)都保存了一个反向指针(phi)，并在每一个状态中存储了一个局部概率(delta)。局部概率 delta 是由反向指针指示的路径到达某个状态的概率。

当 t=T 时，维特比算法所到达的这些终止状态的局部概率 delta’s 是按照最优（最可能）的路径到达该状态的概率。因此，选择其中最大的一个，并回溯找出所隐藏的状态路径，就是这个问题的最好答案。

因为维特比算法不是简单的对于某个给定的时间点选择最可能的隐藏状态，而是基于全局序列做决策——因此，如果在观察序列中有一个“非寻常”的事件发生，对于维特比算法的结果也影响不大。

3.2 文本多分类

在本文提出的智能政务系统模型中，我们根据深度学习中的 LSTM（Long Short-Term Memory）长短期记忆网络，来进行文本多分类。下面对此进行详细介绍。

（1）Keras

Keras 是一个由 Python 编写的开源人工神经网络库，可以作为 Tensorflow、Microsoft-CNTK 和 Theano 的高阶应用程序接口，进行深度学习模型的设计、调试、评估、应用和可视化。

Keras 在代码结构上由面向对象方法编写，完全模块化并具有可扩展性，其运行机制和说明文档有将用户体验和使用难度纳入考虑，并试图简化复杂算法的实现难度。Keras 支持现代人工智能领域的主流算法，包括前馈结构和递归结构的神经网络，也可以通过封装参与构建统计学习模型。在硬件和开发环境方面，Keras 支持多操作系统下的多 GPU 并行计算，可以根据后台设置转化为 Tensorflow、Microsoft-CNTK 等系统下的组件 。

（2）LSTM 的建模工作

长短期记忆网络（LSTM）是一种时间循环神经网络，是为了解决一般的 RNN（循环神经网络）存在的长期依赖问题而专门设计出来的，所有的 RNN 都具有一种重复神经网络模块的链式形式。在标准 RNN 中，这个重复的结构模块只有一个非常简单的结构，例如一个 tanh 层。

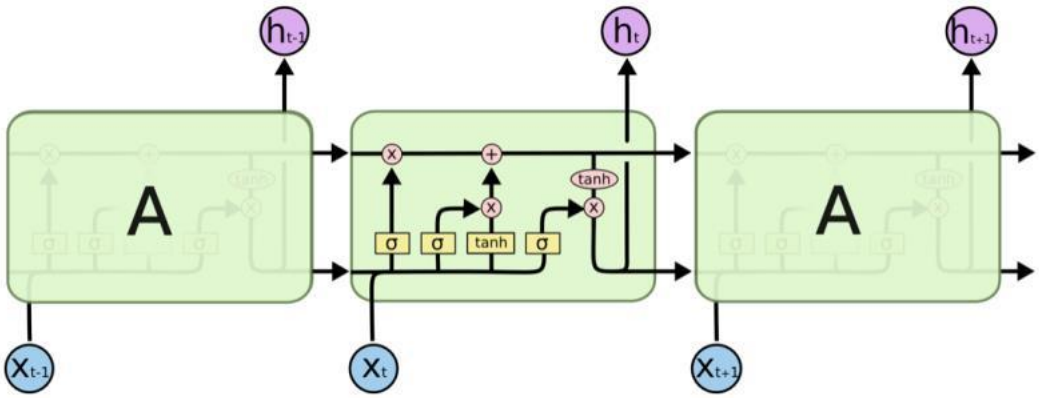

图 LSTM 模型结构原理图

我们的文本多分类子模型是在上述 Keras 这个深度学习框架基础上进行的。将预处理得出的数据评论进行向量化处理，即将每条评论转换成一个整数序列的向量，再设置最频繁使用的 5000 个词，之后设置每条评论数据的最大词语数为 250 个，若超出，则超出部分会被截去，不足则会补 0，并输出结果“共有 79138 个不相同的词语”。之后经操作将 x 为整数结构的两层嵌套 list，再进行填充，让 x 的各个列的长度统一，变成 numpy.ndarray。y 为多嘞标签的 onehot 展开，并输出 x 与 y，操作如图。

In [7]:
```
X = tokenizer.texts_to_sequences(df['cut_review'].values)
X = pad_sequences(X, maxlen=MAX_SEQUENCE_LENGTH)

Y = pd.get_dummies(df['cat_id']).values
print(X.shape)
print(Y.shape)
```

```
(9210, 250)
(9210, 7)
```

图 LSTM 的建模工作 1

之后拆分训练集与测试集，接下来定义一个基于 Keras 的 LSTM 的序列模型。模型的第一次是嵌入层(Embedding)，它使用长度为 49 的向量来表示每一个词语 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。LSTM 层包含 49 个记忆单元，输出层为包含 7 个分类的全连接层，由于是多分类，所以激活函数设置为'softmax'，由于是多分类，所以损失函数为分类交叉熵 categorical_crossentropy。

In [9]:
```
import keras
model = Sequential()
model.add(Embedding(MAX_NB_WORDS, EMBEDDING_DIM, input_length=X.shape[1]))
model.add(SpatialDropout1D(0.2))
model.add(LSTM(49, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(7, activation='softmax'))
# X_train = X_train.astype('float32')
# X_test = X_test.astype('float32')# X_train /= 255# X_test /= 255# Y_train = keras.utils.to_categorical(Y_train,7)
# Y_test = keras.utils.to_categorical(Y_test,7)
# X_train = X_train.reshape(X_train.shape[0],-1)# Y_train = keras.utils.to_categorical(Y_train, 7)
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
print(model.summary())
```

```
WARNING:tensorflow:From C:\Users\GAO\anaconda3\lib\site-packages\tensorflow\python\framework\op_def_library.py:263: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
WARNING:tensorflow:From C:\Users\GAO\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:3445: calling dropout (from tensorflow.python.ops.nn_ops) with keep_prob is deprecated and will be removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 250, 49)           2450000
_________________________________________________________________
spatial_dropout1d_1 (Spatial (None, 250, 49)           0
_________________________________________________________________
lstm_1 (LSTM)                (None, 49)                19404
_________________________________________________________________
dense_1 (Dense)              (None, 7)                 350
=================================================================
Total params: 2,469,754
Trainable params: 2,469,754
Non-trainable params: 0
_________________________________________________________________
None
```

图 LSTM 的建模工作 2

在定义好 LSTM 模型后开始训练数据，并绘制出损失函数趋势图与准确率趋势图并得出相应的实验结果。

```
epochs = 9
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)])
```

```
WARNING:tensorflow:From C:\Users\GAO\anaconda3\lib\site-packages\tensorflow\python\ops\math_ops.py:3066: to_int32 (from tensorflow.python.ops.math_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
Train on 7460 samples, validate on 829 samples
Epoch 1/9
7460/7460 [==============================] - 101s 14ms/step - loss: 1.7514 - acc: 0.3160 - val_loss: 1.3333 - val_acc: 0.4970
Epoch 2/9
7460/7460 [==============================] - 118s 16ms/step - loss: 1.2162 - acc: 0.5828 - val_loss: 1.1273 - val_acc: 0.5995
Epoch 3/9
7460/7460 [==============================] - 129s 17ms/step - loss: 0.9319 - acc: 0.6893 - val_loss: 0.8455 - val_acc: 0.6996
Epoch 4/9
7460/7460 [==============================] - 138s 18ms/step - loss: 0.6008 - acc: 0.7870 - val_loss: 0.7309 - val_acc: 0.7551
Epoch 5/9
7460/7460 [==============================] - 132s 18ms/step - loss: 0.4204 - acc: 0.8649 - val_loss: 0.7090 - val_acc: 0.7853
Epoch 6/9
7460/7460 [==============================] - 137s 18ms/step - loss: 0.2793 - acc: 0.9253 - val_loss: 0.6606 - val_acc: 0.8191
Epoch 7/9
7460/7460 [==============================] - 139s 19ms/step - loss: 0.1853 - acc: 0.9564 - val_loss: 0.6215 - val_acc: 0.8263
Epoch 8/9
```

图 LSTM 训练数据

3.3 高频词挖掘

在本文提出的智能政务系统模型中，在文本多分类后进行高频词挖掘。对预处理后的数据利用 TF-IDF 将词语转化为向量，利用 PCA 降维，之后用 GMM 聚类为 50 类，从而建立热度模型，并自定义热度模型中的热度指标进行分析。

（1）词频-逆向文件频率模型（TF-IDF）

TF-IDF（term frequency－inverse document frequency）是一种用于信息检索与数据挖掘的常用加权技术。TF 是词频(Term Frequency)，IDF 是逆文本频率指数(Inverse Document Frequency)。字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。TF-IDF 加权的各种形式常被搜索引擎应用，作为文件与用户查询之间相关程度的度量或评级。

$$TF（词频）=\frac{某个词在段落中的出现次数}{整体文章出现次数最多的词语次数}$$

$$IDF（逆文档频率）=\log(\frac{语料库中段落总数}{包含该词的段落数})$$

计算得出：

TFIDF=TF*IDF

再结合余弦定理，即可得出相似度。

（2）PCA 降维

在词频-逆向文件频率模型（TF-IDF）中已经将词语转换为向量，接下来采用 PCA 降维。PCA 算法通过舍去一部分信息之后能使得样本的采样密度增大（因为维数降低了），这是缓解维度灾难的重要手段；当数据受到噪声影响时，最小特征值对应的特征向量往往与噪声有关，将它们舍弃能在一定程度上起到降噪的效果；除此之外，PCA 不仅将数据压缩到低维，它也使得降维之后的数据各特征相互独立。PCA 降维原理如下：

（设有 m 条 n 维数据）

1）将原始数据按列组成 n 行 m 列矩阵 X

2）将 X 的每一行（代表一个属性字段）进行零均值化，即减去这一行的均值

3）求出协方差矩阵

4）求出协方差矩阵的特征值及对应的特征向量

5）将特征向量按对应特征值大小从上到下按行排列成矩阵，取前 k 行组成矩阵 P

6）即为降维到 k 维后的数据

```
In [9]: from sklearn.decomposition import PCA
        pca = PCA(n_components=0.9)
        newData = pca.fit_transform(X) #PAC降维
        newData.shape

Out[9]: (4209, 2263)
```

图 PCA 降维操作

（3）GMM 聚类

在利用 PCA 进行降维后，用 GMM 聚类为 50 类。

GMM 聚类原理：通过样本找到 K 个高斯分布的期望和方差，那么 K 个高斯模型就确定了。在聚类的过程中，不会明确的指定一个样本属于哪一类，而是计算这个样本在某个分布中的可能性。

每个 GMM 由K个 Gaussian 分布组成，每个 Gaussian 称为一个

“Component”，这些 Component 线性加成在一起就组成了 GMM 的概率密度函数，由于在对数函数里面又有加和，我们没法直接用求导解方程的办法直接求得最大值。为了解决这个问题，我们采取之前从 GMM 中随机选点的办法：分成两步，类似于 K-means 的两步。估计数据由每个 Component 生成的概率（并不是每个 Component 被选中的概率）。重复迭代前面两步，直到似然函数的值收敛为止。

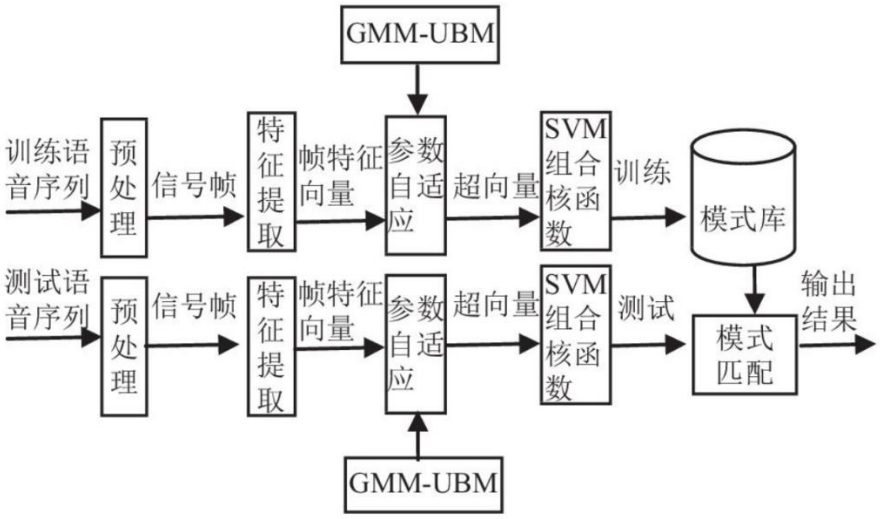

图 GMM 原理

```
In [10]: from sklearn.mixture import GaussianMixture
         clf = GaussianMixture(n_components=50)
         clf.fit(newData)
         result = clf.predict(newData)

In [11]: result = pd.DataFrame(result)
         data3new = data3.loc[data3_after_stop.index,:]
         print(result.shape, data3new.shape)

         (4209, 1) (4209, 7)
```

图 GMM 聚类操作

（4）建立热度模型

对预处理后的数据利用 TF-IDF 将词语转化为向量，利用 PAC 降维，之后用 GMM 聚类为 50 类后，建立热度模型，根据定义的热度指标进行热词挖局。

热度分指标 1：每一类问题点赞数和反对数的计数统计总和进行归一化

```
re1 = data3new['点赞数'].groupby(data3new['labels']).sum() + data3new['反对数'].groupby(data3new['labels']).sum()#热度1
re1 = (re1-re1.min())/(re1.max()-re1.min())#标准化
re1
```

图 热度分指标 1 定义

热度分指标 2：每一类问题的时间跨度进行归一化，并取倒数

```
re2 = pd.to_datetime(data['留言时间']).groupby(data['labels']).max() - pd.to_datetime(data['留言时间']).groupby(data['labels']).min()#热度2
re2 = (re2-re2.min())/(re2.max()-re2.min())#标准化
re2 = 1/re2
```

图 热度分指标 2 定义

热度分指标=热度分指标 1+热度分指标 2

3.4 评价答复意见

在热词挖掘后，我们进行智能政务系统模型中最后一个子模型的建立——评价答复意见模型。

我们将“留言详情”与“答复意见”文件数据进行预处理后，对预处理后的“留言详情”用 TF-IDF 进行处理，再使用同一词库对“答复意见”进行处理，得到稀疏矩阵（0-1 矩阵）。利用欧式距离进行相似度计算，取“答复时间”与“留言时间”的时间差进行归一化来反馈回复的及时性；采用关键词计数的方法来反馈回复的可解释性与完整性，可解释性根据回复中的文件名称、条例、法案、文献资料等名词词频评判；完整性通过前期调查、采取措施、给出的建议结论这三方面以 1:4:2 的比例进行衡量。从而建立了评价答复意见模型并进行分析。

```
#可解释性
key1 = ['实施办法','实施意见','实施方案','工作方案','根据','按照','出台','签订','规定','颁布']
#完整性
key2_1 = ['调查','经查','据查','经核实']#前期调查
key2_2 = ['召开会议','政策','下一步']#采取措施
key2_3 = ['您可','不宜','应该']#给出建议或结论
```

图 评判名词

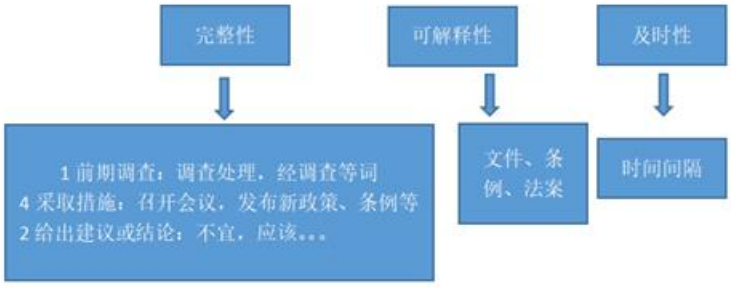

图 评价回复意见模型

四、实验结果

4.1 实验环境

在我们的模型验证过程中，我们主要基 Ubuntu 16.04 的操作系统，实验环境为 128G 的内存容量，8T 的固态硬盘容量，Intel i7 的 GPU，主要以 Python 为开发语言，并基于 Keras / Tensorflow 开发框架完成模型的构建。

4.2 实验结果

（1）群众留言多分类结果

	一级标签	该类总数
0	城乡建设	2009
1	劳动和社会保障	1969
2	教育文体	1589
3	商贸旅游	1215
4	环境保护	938
5	卫生计生	877
6	交通运输	613

图 群众留言分类结果

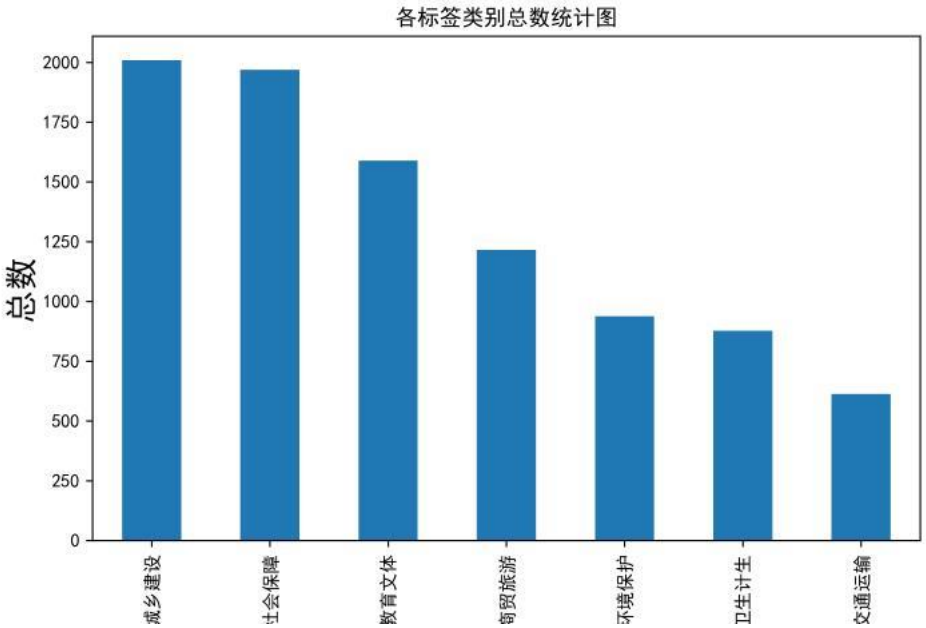

图 各标签类别总数统计图

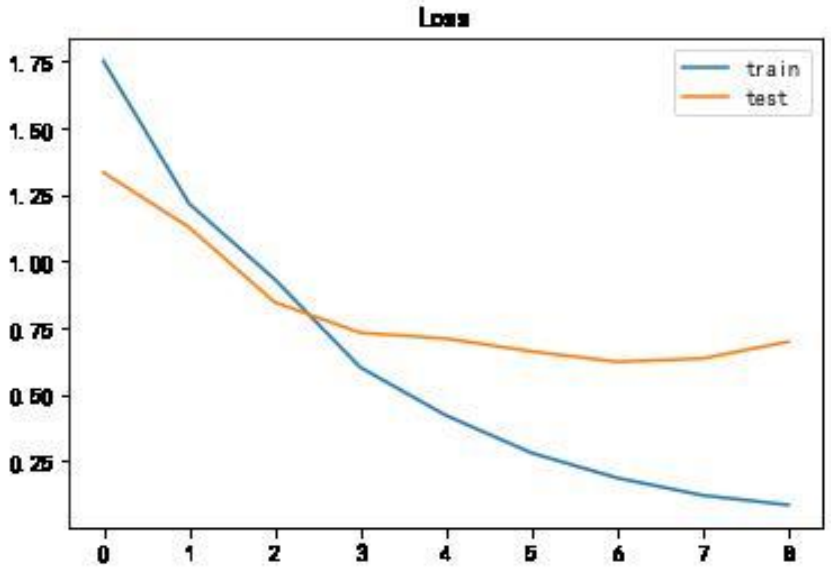

图 损失函数趋势

从上图中我们可以看见,随着训练周期的增加,模型在训练集中损失越来越小,这是典型的过拟合现象,而在测试集中, 损失随着训练周期的增加由一开始的从大逐步变小,再逐步变大。

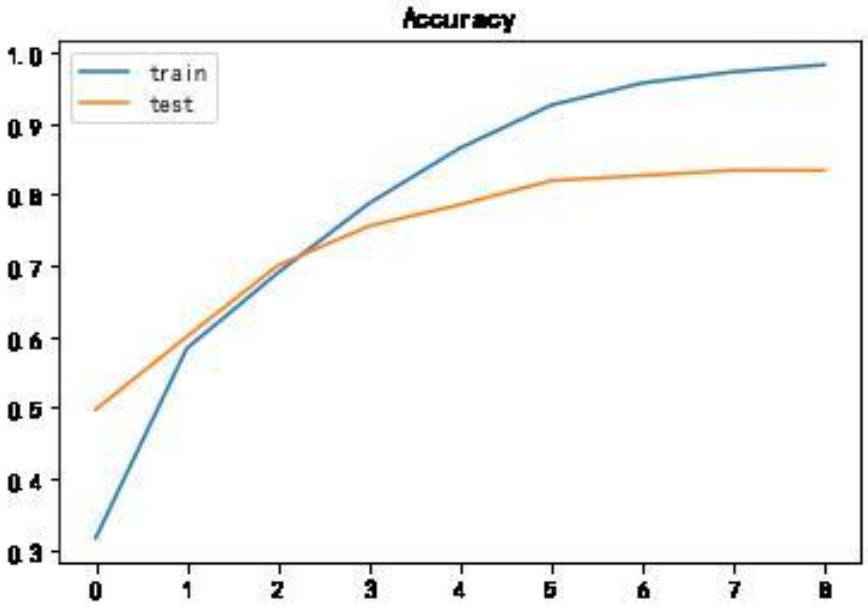

图 准确率趋势图

从上图中我们可以得出：随着训练周期的增加，模型在训练集中准确率越来越高，这是典型的过拟合现象；而在测试集中，准确率随着训练周期的增加

由一开始的从小逐步变大，再逐步变小。

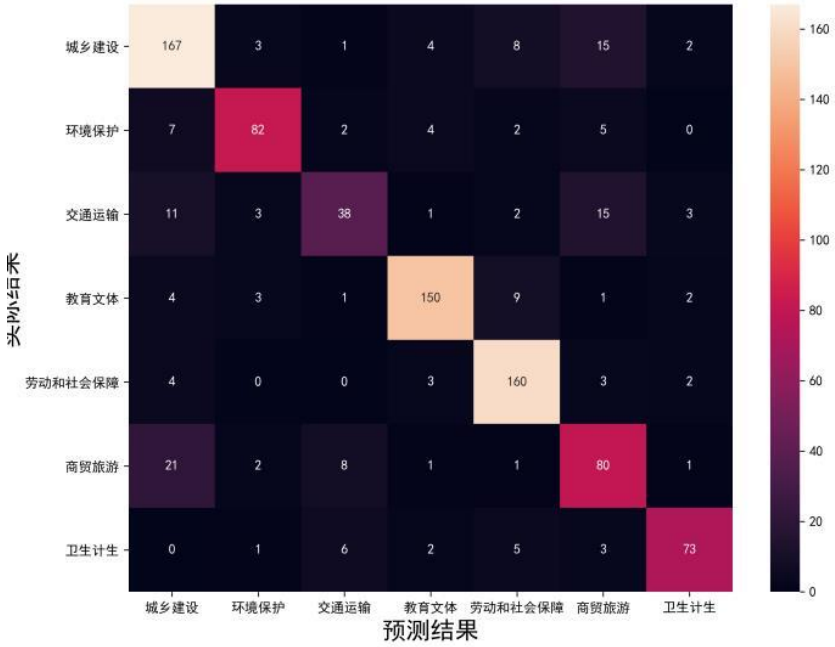

图 预测结果

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 “交通运输”类预测的错误数量较多，预测不是很准确，其他类预测整体良好。

accuracy 0.8143322475570033

	precision	recall	f1-score	support
城乡建设	0.78	0.83	0.81	200
环境保护	0.87	0.80	0.84	102
交通运输	0.68	0.52	0.59	73
教育文体	0.91	0.88	0.90	170
劳动和社会保障	0.86	0.93	0.89	172
商贸旅游	0.66	0.70	0.68	114
卫生计生	0.88	0.81	0.84	90
accuracy			0.81	921
macro avg	0.80	0.78	0.79	921
weighted avg	0.81	0.81	0.81	921

图 准确率与 F1 指标得分

而多分类模型一般不使用准确率(accuracy)来评估模型的质量，因为 accuracy 不能反应出每一个分类的准确性，因为当训练数据不平衡(有的类数据很多,有的类数据很少)时，accuracy 不能反映出模型的实际预测精度,这时

候我们就需要借助于 F1 分数、ROC 等指标来评估模型。

从中可以看出“交通运输”类的 F1 得分较差，“文体教育”类的 F1 得分最高。模型整体质量较高。

（2）热点问题挖掘结果

经过热词挖掘子模型的建立与应用分析，得出以下结果（下图仅仅列出热度前十）。

	热度排名	问题ID	热度指数	时间范围	地点/人群	问题描述
0	1	23	1.793199	2018-10-27-2020/1/7	A3区一米阳光婚纱艺术摄影	是否合法纳税
1	2	43	1.623894	2019-12-04-2020/1/7	A7县春华镇金鼎村	水泥路、自来水到户的问题
2	3	37	1.168777	2019/1/10-2019/9/16	A4区秀峰街道1094	围墙外渣土车日夜运渣土填农田，严重扰民
3	4	22	0.974249	2019-08-01-2020/1/7	A4区北辰小区	非法住改商问题
4	5	26	0.860843	2019-08-20-2020/1/6	A2区云峰路水墨林溪苑二期	电梯覆盖信号差
5	6	24	0.756695	2019-11-25-2020/1/6	A市A1区纬二路	有两种处罚标准
6	7	36	0.735325	2019/1/15-2020/1/7	A2区黄兴路步行街大古道巷住户	卫生间粪便外排
7	8	47	0.718492	2019-11-13-2020/1/6	A市万家丽南路丽发新城居民区	附近搅拌站扰民
8	9	5	0.676673	2019-08-24-2020/1/2	A市保利麓谷林语小区居民	保利地产商及物业无良
9	10	30	0.667559	2019-12-05-2020/1/6	A6区道路命名规划初步成果公示	城乡门牌问题
10	11	10	0.660442	2019-08-29-2020/1/7	A市南郡明珠	业主都没有房产证

图 热词挖掘模型结果

（3）评价答复意见的结果

根据评价答复子模型的建立与应用分析，得出相应结果。

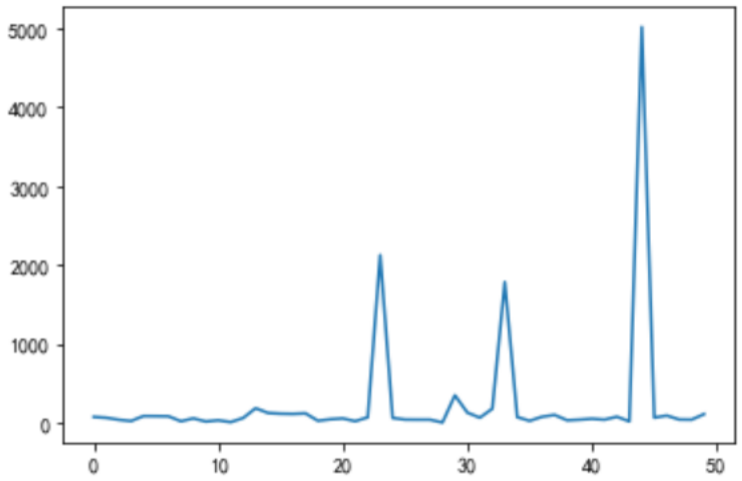

图 各分类点赞数总和

如图所示为前十条数据的答复根据相关性、及时性、可解释性、完整性的评价。

	相关性	及时性	可解释性	完整性
0	0.306351	0.013121	0.000000	0.018519
1	0.467170	0.012702	0.000000	0.009259
2	0.438813	0.012716	0.181818	0.000000
3	0.501540	0.012736	0.090909	0.000000
4	0.352902	0.013530	0.000000	0.018519
5	0.552432	0.026765	0.000000	0.000000
6	0.612775	0.035275	0.090909	0.009259
7	0.304174	0.024578	0.000000	0.000000
8	0.271034	0.013988	0.000000	0.009259
9	0.410660	0.013994	0.000000	0.000000
10	0.109957	0.060954	0.000000	0.009259

图 评价回复子模型结果

五、 总结与展望

5.1 总结

为了迎接近年来，因微信、微博、市长信箱、阳光热线等网络问政平台逐步成为政府了解民意、汇聚民智、凝聚民气的重要渠道，而使得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。本文根据人工智能的相关理论好实验，构建了一个智能政务处理系统模型，从而提升政府的管理水平和施政效率，更好的解决民生问题。

模型主要包括四个部分：数据预处理、文本多分类、高频词挖掘、评价答复意见，在对赛题研究的基础上，我们根据研究思路撰写本论文，通过实验验证了本模型的可行性，基本实现本赛题设立的目标。

5.2 展望

（1）模型优化

本文有些模型尚未成熟，思路尚浅，有待进一步提高逻辑性与严密性。同时一些模型有待优化与提升，希望日后可以加强相关研究，改善模型。

（2）构建以用户满意度为导向的智慧政务评价系统

仅通过政府回复的文本内容来判断其回复质量的高低易产生一定的倾斜，使判断出现较大误差，而以用户满意度为导向的智慧政务评价系统则可以结合用户对回复的及时评价，对相关问题解决的质量作出有效评定。

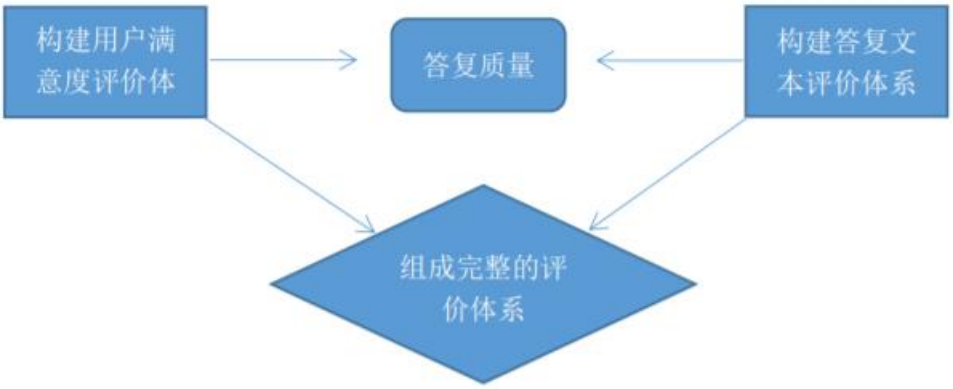

图 构建以用户满意度为导向的智慧政务评价系统

六、参考文献

[1]陈志,郭武.不平衡训练数据下的基于深度学习的文本分类[J].小型微型计算机系统,2020,41(01):1-5.

[2]薛金成,姜迪,吴建德.基于 LSTM-A 深度学习的专利文本分类研究[J].通信技术,2019,52(12):2888-2892.

[3]王颖. 基于 SERVQUAL 模型的智慧政务服务质量评价研究[D].华中科技大学,2019.

[4]https://blog.csdn.net/zhengxqq27/article/details/90481590

[5]https://blog.csdn.net/weixin_42608414/article/details/89856566?utm_medium=distribute.wap_relevant.none-task-blog-BlogCommendFromBaidu-2&depth_1-utm_source=distribute.wap_relevant.none-task-blog-BlogCommendFromBaidu-2

[6]第六届“泰迪杯”数据挖掘挑战赛优秀作品《基于双重注意力机制与 Bi-LSTM 的智能阅读系统》

[7]第六届“泰迪杯”数据挖掘挑战赛优秀作品《一种基于潜在语义索引和卷积神经网络的智能阅读模型》

[8]https://blog.csdn.net/juanjuan1314/article/details/77961961

[9]https://baike.baidu.com/item/tf-idf/8816134

[10]https://blog.csdn.net/kamendula/article/details/51568895

[11]https://www.cnblogs.com/mindpuzzle/archive/2013/04/24/3036447.html

[12]https://blog.csdn.net/zouxiaolv/article/details/100590725