基于自然语言处理群众留言与神经网络学习评价相关答复

摘　要

近年来，随着大数据与人工智能的发展，政务领域也迈入智慧化转型阶段，并打造出服务型、智慧型政府。本文以从海量政务大数据中收集的群众问政留言记录，以及相关部门对部分群众留言的答复意见作为研究对象，利用自然语言处理技术处理相关文本信息，用数据挖掘从大量数据中挖掘出重要数据，用分类和聚类算法整理数据建立模型，通过进行神经网络训练学习，对群众留言的答复意见作出评价。

针对问题一，首先是读取已经获取到的数据，从中获取留言主题和留言详情这两个字段中的数据，本文使用了 sklearn 库中数据集划分函数 StratifiedShuffleSplit，实现了将数据集按相应的比例划分成测试集和训练集，再使用 split(x, y)，按照 y 的值将数据集分为训练集或测试集，保证训练集和测试集中各类 y 值所占的比例与原数据集相同。将获取到的训练集数据和测试集数据进行数据的预处理，使用 pandas 中的 dropna 函数，滤除缺失数据。利用 jieba 分词获取到数据中的特征词，并去除其中的停用词，再将特征词与所对应的标签整合，使带有标签的数据写入磁盘中，用于后续模型的训练。本文中采用了 FastText 文本分类算法，fastText 不需要预训练好的词向量，fastText 会自己训练词向量，本文利用 fasttext 训练模型，将带有标签的训练集数据用于训练我们的监督学习的分类器，最后使用测试集来评估模型，后续再进行了模型优化，在预处理阶段，将数据进行正则化。

针对问题二，目的是对热点问题的挖掘，我们可以对附件三给出的留言主题、留言详情以及时间列使用 sklearn 库中数据集划分函数 StratifiedShuffleSplit 对数据进行划分归类，将获取到的数据进行预处理，使用 pandas 中的 dropna 函数，滤除缺失数据，利用 jieba 去除其中的停用词，使用 nulk 库中 word_tokenize 函数对处理过的留言主题进行实体识别，获取地点或人群以及问题描述，同样使用 word_tokenize 函数对附件三中的时间列进行提取，再将它们保存到“热点问题表.xls”中；接着使用 TextRank 算法对处理过的留言详情进行文本摘要提取，得到热点问题留言主题和留言详情，并将其保存在“热点问题留言明细表.xls”中。

针对问题三，由附件 4 相关部门对留言的答复意见，要求我们从答复的相关性、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。我们可由相关性、完整性等定义入手，相关性，即答复意见的内容是否与问题相关；完整性，即是否满足某种规定。对于相关性，我们可以使用 gensim 文本相似度分析算法对附件四中的留言详情与答复意见进行文本相似

度计算，使用 TF-IDF 模型进行训练。为了保证相关性的准确度，我们还利用欧式距离、余弦距离和基于词向量的余弦相似度算法进行并行计算，得出文本相似度，再与 gensim 文本相似度分析算法进行比较，得出相关结论。对于完整性是否满足某种规定，我们这里则判定所给的句子是否通顺，使用 N-Gram 来判断句子是否通顺。

关键词：sklearn 数据集划分；jieba 分词；FastText-文本分类算法；gensim-文本相似度分析算法

Natural Language Processing and Neural Network Learning Evaluation

Abstract

In recent years, with the development of big data and artificial intelligence, the field of government affairs has entered the stage of intelligent transformation, and created a service-oriented and intelligent government. In this paper, we use natural language processing technology to process relevant text information, use data mining to extract important data from a large number of data, use classification and clustering algorithm to organize data to build a model, and train and learn through neural network , comment on the response to the message.

To solve the first problem, firstly, read the data that has been obtained, and obtain the data in the two fields of message subject and message details. In this paper, we use the data set division function stratifiedsufflesplit in the sklearn database to realize the division of the data set into test set and training set according to the corresponding proportion, and then use split (x, y) According to the value of Y, the data set is divided into training set or test set to ensure that the proportion of various Y values in training set and test set is the same as the original data set. The training set data and test set data are preprocessed, and the dropna function in pandas is used to filter the missing data. Using the Jieba segmentation to get the feature words in the data, and remove the stop words, and then integrate the feature words with the corresponding tags, so that the data with tags can be written to disk for the subsequent model training. In this paper, the fasttext text classification algorithm is used. Fasttext does not need pre trained word vectors, and fasttext will train word vectors by itself. In this paper, the fasttext training model is used to train our supervised learning classifier with labeled training set data. Finally, the test set is used to evaluate the model, and then the model is optimized. In the pre-processing stage, the data is input into the Row regularization.

Aiming at the second problem, the purpose is to mine the hot issues. We can use the data set partition function stratifiedsufflesplit in the sklearn database to partition and classify the message topics, message details and time columns given in Annex 3, and pre process the acquired data. We can use the dropna function in pandas to filter out the missing data and use the Jieba to remove the stop words In nulk database, the word "tokenize" function is used to identify the entity of the processed message subject, obtain the location or crowd and the description of the problem. In the same way, the word "tokenize" function is used to extract the time column in Annex 3, and then save them to the "hot issues table. XLS"; then the

textrank algorithm is used to extract the text summary of the processed message details, and the hot issues are obtained The topic and details of the message are saved in the "hot issues message list. XLS".

In response to question 3, the relevant departments in Annex 4 ask us to provide a set of evaluation scheme for the quality of the reply from the perspective of relevance, integrity and interpretability of the reply. We can start with the definition of relevance and integrity, that is, whether the content of the reply is related to the question; integrity, that is, whether it meets certain requirements. For the relevance, we can use gensim text similarity analysis algorithm to calculate the text similarity of the message details and reply comments in Annex 4, and use TF-IDF model for training. In order to ensure the accuracy of the correlation, we also use the Euclidean distance, cosine distance and cosine similarity algorithm based on word vector to calculate the text similarity, and then compare with gensim text similarity analysis algorithm to get the relevant conclusions. As for whether the integrity meets certain requirements, we will determine whether the given sentence is smooth or not, and use n-gram to determine whether the sentence is smooth or not.

Keywords:sklearn data set partition；Jieba word segmentation；fasttext-text classification algorithm；gensim-text classification algorithm

目 录

1. 挖掘目标

1.1 挖掘背景

近年来，随着网络问政的兴起，政府微信公众号和政务微博已日益成为群众反映问题、汇集网络民意的平台，也成为各级党委、政府倾听网络民意、解决问题的平台,各类社情民意相关的文本数据量不断攀升,无疑给政府相关部门在处理问题上带来巨大的挑战。

以前，在处理网络问政平台的群众留言时，工作人员首先按照一定的划分体系对留言进行分类，以便后续将群众留言分派至相应的职能部门处理，但由于数据量很大，人工处理起来，第一是处理起来不方便，其次就是效率比较低，不能将群众所反映的问题得到及时的解决，再则就是差错率高。

随着云计算、大数据、物联网、人工智能等新一代信息技术的不断成熟，它们开始与政务工作进行全方位融合，政务运行和政务服务更加有针对性和效率，使人人在网上，事事在格上，实时感知社情民意，洞察大众冷暖，从而开启了一个全新的智慧政务时代。

因此题目需要使用基于自然语言处理技术对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进行数据挖掘，对数据进行清洗和整理，建立有效的文本分类模型，针对相关部门对留言的答复意见，通过相关性，完整性，可解释性等描述把指标量化出来，做一套评价方案，使答复令人满意。

1.2 挖掘目标

参考附件 1 提供的内容分类三级标签体系对留言进行分类,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。

根据附件 3 将某一时段内反映特定地点或特定人群问题的留言进行归类，定义合理的热度评价指标，并给出评价结果，按表 1 的格式给出排名前 5 的热点问题，并保存为文件“热点问题表.xls”。按表 2 的格式给出相应热点问题对应的留言信息，并保存为“热点问题留言明细表.xls”。

针对附件 4 相关部门对留言的答复意见，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。

2. 问题分析

2.1 问题一的分析

对于问题一的分析，由于题目需要根据附件 2 建立关于留言内容的一级标签分类模型，就会使用到文本分类算法以及相关模型的训练，在此之前要对附件 2 中的数据进行测试集和数据集的划分，将训练集用于模型的训练，根据测试集评价模型的结果，再考虑是否进行模型的优化，让结果达到理想状态，在数据划分之前，由于所研究的对象是留言文本信息，就会使用到自然语言处理中 jieba 分词对文本信息进行处理，处理过后就能对数据进行多方面的操作，包括数据清洗，异常数据的处理，停用词的去除以及特征词的获取等。

2.2 问题二的分析

对问题二的分析，要求我们根据附件三将某一时段内反映特定地点或特定人群问题的留言进行归类，并将归类结果保存在相应的表格里。首先是对附件三中的留言主题、留言详情进行数据清洗，异常数据的处理，停用词的去除以及特征词的获取等，再进行文本分类算法，得到相应的文本分类器，接着对留言主题生成的文本生成器使用实体识别算法，获取地点/人群以及问题描述，同样对附件三中的时间列使用实体识别算法获取时间范围;对留言详情生成的文本生成器使用 TextRank 算法进行文本摘要提取，获取热点问题留言明细表中的留言主题与留言详情。

2.3 问题三的分析

对问题三的分析，由附件 4 相关部门对留言的答复意见，要求我们从答复的相关性、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。可由相关性、完整性等定义入手，相关性即：答复意见的内容是否与问题相关；完整性即：是否满足某种规定。于相关性，我们可以使用 gensim 文本相似度分析算法对附件四中的留言详情与答复意见进行文本相似度计算，使用 TF-IDF 模型进行训练;对于完整性：是否满足某种规定，我们这里则判定所给的句子是否通顺，使用 N-Gram 来判断句子是否通顺。

3. 数据预处理

3.1 滤除缺失数据

本文将数据处理成了 pandas 能够读取和操作的文件，对于缺失数据的处理，本文就采用了 pandas 中的 dropna()函数，由于缺失的数据比较少，就将缺失数据所在的行数据全部删除，但对其结果的影响很小，删除数据后能获取到完整的数据列表，而且减少了后续数据的挖掘以及处理的时间和内存要求。

3.2 去除停用词

停用词主要包括英文字符、数字、数学字符、标点符号及使用频率特高的单汉字等，在文本分析中，去除停用词是文本分析中一个预处理方法，能够过滤掉分词结果中的噪声，在本文中是获取文本数据中的特征词，停用词的过滤去除数据的冗余性，减少数据所占用内存空间，大大提高了数据处理的高效性以及训练模型的准确性。

3.3 数据正则化

数据正则化主要用于两个方面，第一用于结巴分词中去除停用词，第二用于优化已经训练完成的模型，例如本文中将所有数据中的大写转化为了小写，这样就会提高模型训练的结果。

3.4 中文分词

将中文语句切割成单独的词组，可以把分词算法分为四大类：基于规则的分词方法，基于统计的分词方法，基于语义的分词方法，基于理解的分词方法；在这里使用 jieba 分词，jieba 分词分为三种：精确模式，试图将句子最精确地切开，适合文本分析；全模式，将句子中所有的可能成词的词语都扫描出来，速度非常快，但是不能解决歧义；搜索引擎模式：在精确模式的基础上，对长词再次切分，提高召回率，适用于搜索引擎分词。我们使用精确模式，利用 cut 函数将句子精确地切开，并统计词频，为后续的数据处理做准备。

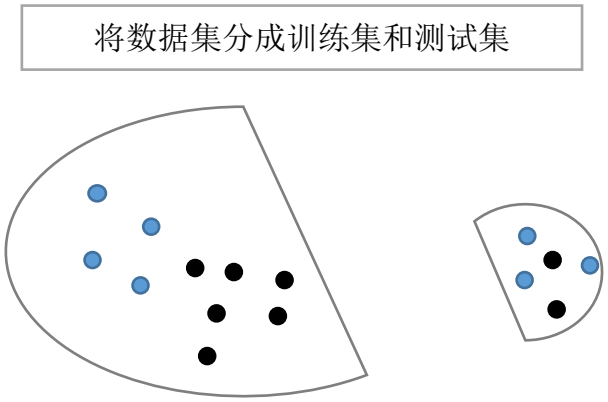
4. 问题求解

4.1 建立分类模型

4.1.1 数据集的划分

本文采用了 sklearn 库中数据集划分函数 StratifiedShuffleSplit（），数据集在进行划分之前，首先是需要进行打乱操作，否则容易产生过拟合，模型

泛化能力下降。StratifiedShuffleSplit（）函数将数据集中大部分数据划分为训练集，小部分划分为测试集。



1.函数用法

```
from sklearn.model_selection import StratifiedShuffleSplit
StratifiedShuffleSplit(n_splits=1, test_size=0.2, train_size=0.8, random_state=5)
```

2. 相关参数的说明

（1）参数 n_splits 是将训练数据分成 train/test 对的组数，根据项目的需要进行划分，其默认参数值为 10。

（2）参数 test_size 和 train_size 是用来设置 train/test 对中 train 和 test 所占的比例。

（3）参数 random_state 控制是将样本随机打乱,是随机数种子，和 random 中的 seed 种子一样，保证每次抽样到的数据一样，便于调试。

3. 函数的作用

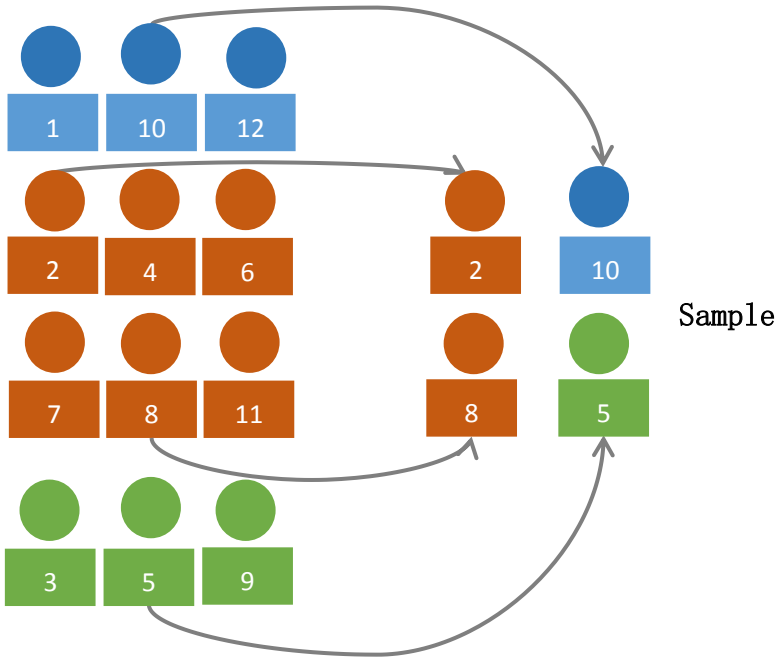
（1）其产生指定数量的独立的 train/test 数据集划分数据集划分成 n 组，n 的值会影响到模型训练的结果，对 n 的改变（尽量是增加 n 的值）可以优化模型，让结果达到预想状态。

（2）首先将样本随机打乱，然后根据设置参数划分出 train/test 对。

（3）其创建的每一组划分将保证每组类比比例相同。即第一组训练数据类别比例为 2:1 则后面每组类别都满足这个比例。

4.obj.split(X, y)对数据集的处理

使用 StratifiedShuffleSplit（）函数划分数据集,通过参数构建 StratifiedShuffleSplit()对象,再使用对象的 split 方法分割，返回分组后数在原数组中的索引。同时 split(X, y)能够按照 y 的值将数据集分为训练集或测试集，保证训练集和测试集中各类 y 值所占的比例与原数据集相同。



4.1.2 jieba 分词及特征词的获取

在做文本数据挖掘的时候，第一步要做文本的预处理就是分词，随着 NLP 技术的日益成熟，开源实现的分词工具越来越多，如 Ansj、盘古分词等，本文采用了 Jieba 分词来处理文本信息。

Jieba 分词结合了基于规则和基于统计这两类方法，具有高性能、准确率、可扩展性等特点，属于概率语言模型分词。其中的中文分词要稍微的困难一些，和英文不同，中文语句是由连续的字符组成序列后呈现的，没有像英文一样的分隔符。

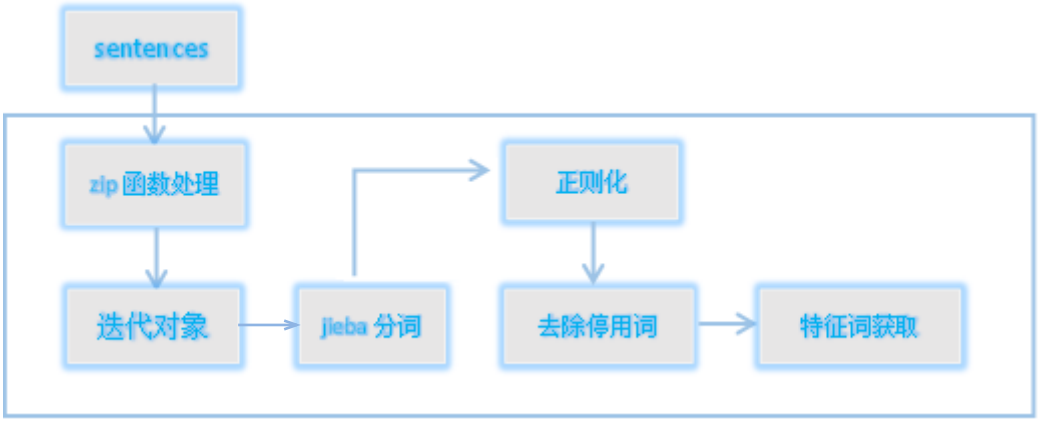
本文采用 jieba 分词主要用于获取特征词，其步骤如下：

1. 从数据集中获取所需进行分词的对象。

2. 使用 zip 函数将需进行分词的对象作为参数，将对象中对应的元素打包成一个 tuple,然后返回一个可迭代的 zip 对象，用 jieba 分词中精准模式处理可迭代的 zip 对象。

3. 使用停用词词表将获取到的分词去除其中的停用词。

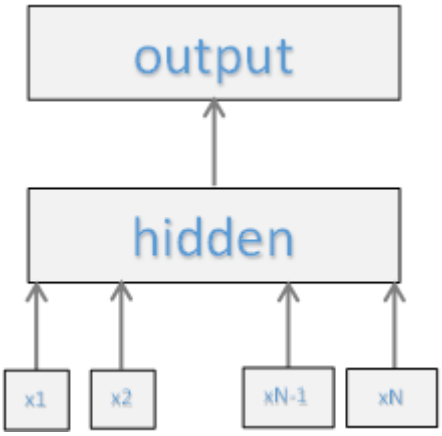
4. 处理所获取到的特征词。



4.1.3 创建文本分类模型

本文使用 FastText 文本分类算法建立文本分类模型，fastText 是一个快速文本分类算法，其主要思想基于 word2vec 中的 skip-gram 模型，在训练文本分类模型的同时，也将训练出字符级 n-gram 词向量。

fastText 的模型架构和 word2vec 中的 CBOW 模型的结构很相似。CBOW 模型是利用上下文来预测中间词，而 fastText 是利用上下文来预测文本的类别。从本质上来说，word2vec 是属于无监督学习，fastText 是有监督学习。但两者都是三层的网络（输入层、单层隐藏层、输出层），具体的模型结构如下：

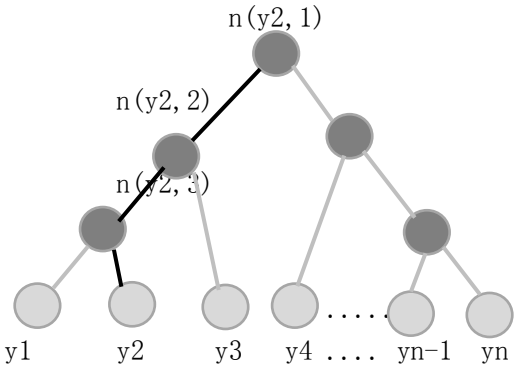


上面图中 xi 表示的是文本中第 i 个词的特征向量，该模型的负对数似然函数如下：

$$-\frac{1}{N}\sum_{n=1}^{N} y_n \log(f(BAx_n))$$

上面式子中的矩阵 A 是词查找表，整个模型是查找出所有的词表示之后取平均值，用该平均值来代表文本表示，然后将这个文本表示输入到线性分类器中，

也就是输出层的 softmax 函数,式子中的 B 是函数 f 的权重系数。softmax 计算概率分布,它的基本思想是使用树的层级结构替代扁平化的标准 Softmax，使得在计算 P(y=j)时，只需计算一条路径上的所有节点的概率值，无需在意其它的节点,分层 softmax 结构：



树的结构是根据类标的频数构造的 Huffman Tree。

n 个不同的类标组成所有的叶子节点，n-1 个内部节点作为内部参数，从根节点到某个叶子节点经过的节点和边形成一条路径，路径长度被表示为 L(yj)，P(yj)就可以被写成：

$$p(y_j)=\prod_{y=1}^{L(y_j)-1}\sigma([n(y_j,l+1)=LC(n(y_j,l))]\cdot\theta n(y_j,l)^T X)$$

其中：

$\sigma(\cdot)$ 表示 sigmoid 函数，里面是判断该节点是否是左孩子，若是，则 1，否则-1；

$LC(n)$ 表示 n 节点的左孩子；

$[x]$ 是一个特殊的函数，被定义为 $[x]=\begin{cases}1 & \text{if x=true}\\ -1 & \text{otherwise}\end{cases}$

$\theta n(yj,l)$ 是中间节点 $n(yj,l)$ 的参数；

X 是 Softmax 层的输入

。4.1.4 模型的评估和优化

一个比较好的文本分类模型必须要经得起真实数据的验证,本文将文本分类模型直接在数据集中所划分的测试集上进行测试，本文是通过 F1-Score 来评估模型，是反映模型的稳健型，是精确率和召回率的调和平均数，最大为 1，最小为 0，其公式如下：

$$F1 = \frac{2TP}{2TP + FN + \mathrm{FP}} = \frac{2 \cdot \mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

在分类模型中，将每个样本的预测值与对应的实际值作比较，TP(true positive)所表示的是预测为正，实际也为正，将正类预测为正类数，FP(false positive)表示预测为正，实际为负，将负类预测为正类数，FN(false negative)表示预测为负，实际为正，将正类预测为负类数，TN(true negative)，预测为负，实际也为负，将负类预测为负类数，其中 Precision（精确率）,Recall（召回率）的表示如下：

$$\mathrm{Recall} = \frac{TP}{TP + FN} \quad , \quad \mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + FP}$$

由以上三个公式可知,recall 体现了分类模型对正样本的识别能力,recall 越高，说明模型对正样本的识别能力越强，precision 体现了模型对负样本的区分能力，precision 越高，说明模型对负样本的区分能力越强 F1-score 是 Precision 和 Recall 的综合，F1-score 越高，则表示分类模型越稳健。

针对第一题，其模型第一次测试结果 90%，这个结果可以说明分类模型需进行优化，本文主要是在数据预处理阶段，将所有大写转化为小写，在使用 jieba 分词之前，先用正则表达式处理数据，用于后续更好，更高效去除停用词，从而获取到更高质量的特征词，经过这两方面的处理，其结果增加了 3 个百分点。模型再一次优化可以在数据集划分的时候将参数 n_splits 的值调高，从而实现模型的优化。

4.2 挖掘热点数据

4.2.1 文本分类与关键词提取

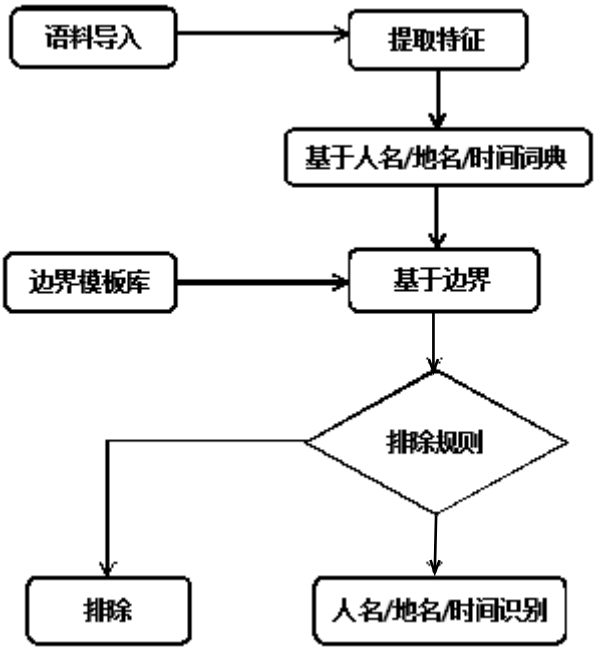
对附件三给出的留言主题、留言详情以及时间列使用 sklearn 库中数据集划分函数 StratifiedShuffleSplit（）对数据进行划分归类，将获取到的数据进行预处理，使用 FastText 文本分类算法建立留言主题文本分类器与留言详情文本分类器。对留言详情文本分类器使用 TextRank 算法，通过把文本分割成若干组成单元(单词、句子) 并建立图模型, 利用投票机制对文本中的重要成分进行排序, 仅利用单篇文档本身的信息即可实现关键词提取、文摘。TextRank 一般模型可以表示为一个有向有权图 G=(V,E),由点集合 V 和边集合 E 组成,E 是 V×V 的

子集。图中任两点 Vi,Vj 之间边的权重为 wji,对于一个给定的点 Vi,In(Vi)为指向该点的点集合,Out(Vi)为点 Vi 指向的点集合。点 Vi 的得分定义如下:

$$WS\left(V_i\right)=\left(1-d\right)+d*\sum_{V_i\in Out\left(V_i\right)}\frac{W_{ji}}{\sum_{V_i\in Out\left(V_i\right)_i}W_{ij}}WS\left(V_j\right)$$

4.2.2 实体识别

命名实体识别是信息提取、问答系统、句法分析、机器翻译等应用领域的重要基础工具，在自然语言处理技术走向实用化的过程中占有重要地位。一般来说，命名实体识别的任务就是识别出待处理文本中三大类（实体类、时间类和数字类）、七小类（人名、机构名、地名、时间、日期、货币和百分比）命名实体。在本小问中，对附件三中生成的留言主题文本分类器使用实体识别与模式匹配算法进行人群/地点与时间范围的提取。主要流程图如下图所示：



人名/地名/时间识别：人名以“王菲”为例，粗分结果是“始##始, 王, 菲, 末##末”，很明显，粗分过程并不能识别正确的人名，因为“王菲”这个词并不存在于一元语言模型词典中；地名以“重庆”为例，粗分结果是“始##始, 重, 庆, 末##末, ”；时间以“2019/08/18”为例，粗分结果是“始##始, 2019, 08, 18, 末##末。

HMM 模型：人名识别是一个 HMM 的求解问题，所以需要建立 HMM 模型，分为下面五大步骤，

1. 观测序列

观测序列是我们能看到的显状态序列，这个例子里是“始##始, 王, 菲, 末##末”。

2. 隐状态

隐状态是下面的标注集(红色部分)：

编码	代码	意义
B	Pf	姓氏
C	Pm	双名的首字
D	Pt	双名的末字
E	Ps	单名
F	Ppf	前缀
G	Plf	后缀
K	Pp	人名的上文
L	Pn	人名的下文
M	Ppn	两个中国人名之间的成分
U	Ppf	人名的上文和姓成词
V	Pnw	人名的末字和下文成词
X	Pfm	姓与双名的首字成词
Y	Pfs	姓与单名成词
Z	Pmt	双名本身成词
A	Po	以上之外其他的角色

红色部分是标签，在本文中，我会混用“标签”“隐状态”“tag”这三个词，不再赘述。

这十五种标签分别对应于 15 个不重复的整型数字：

0 1 2 3 4 5 6 11 12 13 23 24 25 100 101

按照 result += (char) (tag + 'A')来映射的，不过 100 和 101 会被映射到字母表之外：

0-a 1-b 2-c 3-d 4-e 5-f 6-g 11-l 12-m 13-n 23-x 24-y 25-z 100-Å 101-Æ

3.初始概率

初始概率指的是一个隐状态随机出现的概率，可以用某个隐状态的频度除以所有隐状态的频度来计算。每个标签的频度可以从词典中查到：

6937450 92626 69241 70479 14295 870 869 65949 78874 14025 1238 3351 5397 329805 0

4.转移概率

转移概率是指前面的隐状态固定，后面的隐状态是 X 时候的概率。

5.求解 HMM：模式匹配

通过模式匹配来发现，ICTCLAS 中用到的模式串有：

BBCD:姓+姓+名 1+名 2
BBE: 姓+姓+单名
BBZ: 姓+姓+双名成词
BCD: 姓+名 1+名 2
BE: 姓+单名
BEE: 姓+单名+单名
BG: 姓+后缀
BXD: 姓+姓双名首字成词+双名末字
BZ: 姓+双名成词
B: 姓
CD: 名 1+名 2
EE: 单名+单名;
FB: 前缀+姓
XD: 姓双名首字成词+双名末字
Y: 姓单名成词

通过上面的实体识别算法，我们可以提取出所需要的人群/地点以及时间。

4. 2. 3 摘要提取

将大量的文本进行处理，产生简洁、精炼内容的过程就是文本摘要。

Extraction 是抽取式自动文摘方法，通过提取文档中已存在的关键词，句子形成摘要。"自动摘要"就是要找出那些包含信息最多的句子。句子的信息量用"关键词"来衡量。如果包含的关键词越多，就说明这个句子越重要。Luhn 提出用"簇"（cluster）表示关键词的聚集。所谓"簇"就是包含多个关键词的句子片段。本小问利用抽取式自动文摘方法对附件三中处理过的留言详情进行摘要提取。

上图就是 Luhn 原始论文的插图，被框起来的部分就是一个"簇"。只要关键词之间的距离小于"门槛值"，它们就被认为处于同一个簇之中。Luhn 建议的门槛值是 4 或 5。也就是说，如果两个关键词之间有 5 个以上的其他词，就可以把这两个关键词分在两个簇。下一步，对于每个簇，都计算它的重要性分值。

簇的重要性=（包含的关键词）^2/簇的长度

通过该算法可以确定关键词的重要性。然后，找出包含分值最高的簇的句子（比如 5 句），把它们合在一起，就构成了这篇文章的自动摘要。

基于 TextRank 的自动文摘算法具体步骤如下：

1.预处理：将输入的文本或文本集的内容分割成句子得，构建图 G=(V,E)，其中 V 为句子集，对句子进行分词、去除停止词，得，其中是保留后的候选关键词。

2.句子相似度计算：构建图 G 中的边集 E，基于句子间的内容覆盖率，给定两个句子，采用如下公式进行计算：

$$\text{Simiarity}\left(S_{i,}S_j\right)=\frac{\left|\left\{t_k \nu t_k \in S_i \Lambda t_k \in S_j\right\}\right|}{\log\left(\left|S_i\right|\right)+\log\left(\left|S_j\right|\right)}$$

3.句子权重计算：根据公式，迭代传播权重计算各句子的得分；

4.抽取文摘句：将 3 得到的句子得分进行倒序排序，抽取重要度最高的 T 个句子作为候选文摘句。

5.形成文摘：根据字数或句子数要求，从候选文摘句中抽取句子组成文摘。

4.3 评价答复意见

4.3.1 相关性评价模型

问题三中要求我们针对附件 4 相关部门对留言的答复意见，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。首先我们从相关性入手，建立起相关性评价模型。相关性即答复意见的内容是否与问题相关，因此对附件四中留言详情与答复意见进行文本相似度分析，使用 Python 相似度计算和 gensim-文本相似度分析算法。

1．Python 相似度计算

（1）基于词向量的余弦相似度算法。首先假设将留言详情看为 A 句，答复意见看为 B 句，分别对它们进行 jieba 分词，列出所有的词，计算词频， 写出词频向量；

如：句子 A：(1，1，2，1，1，1，0，0，0)

句子 B：(1，1，1，0，1，1，1，1，1)

（2）我们可以把它们想象成空间中的两条线段，都是从原点（[0, 0, …]）出发，指向不同的方向。两条线段之间形成一个夹角，如果夹角为 0 度，意味着方向相同、线段重合,这是表示两个向量代表的文本完全相等；如果夹角为 90 度，意味着形成直角，方向完全不相似；如果夹角为 180 度，意味着方向正好相反。因此，我们可以通过夹角的大小，来判断向量的相似程度。夹角越小，就代表越相似。

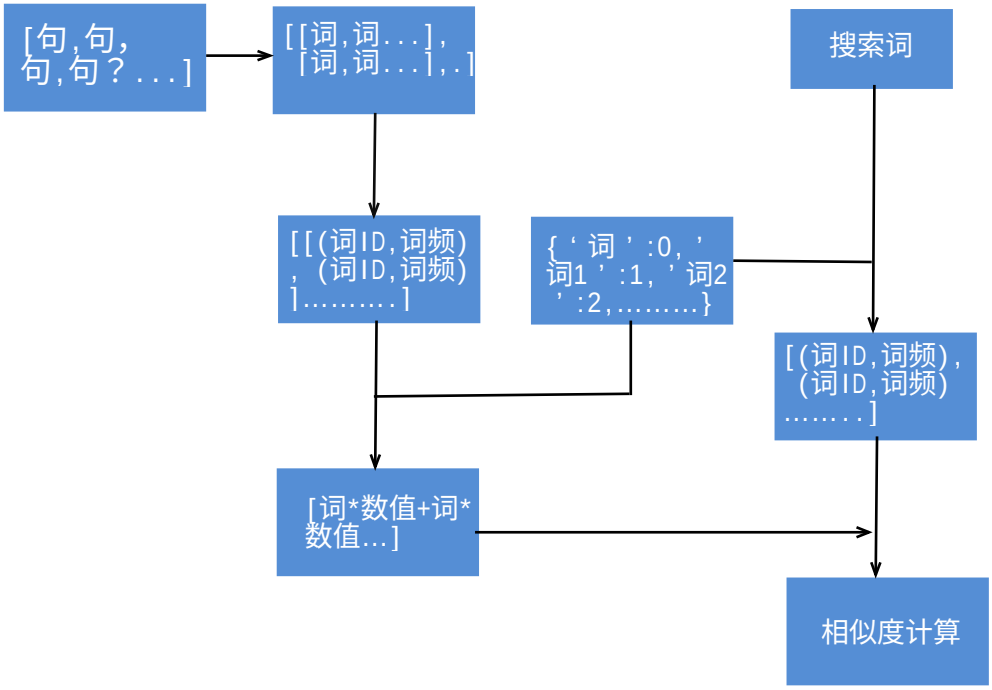
$$\cos(\theta)=\frac{\sum_{i=1}^{n}\left(X_i \times Y_i\right)}{\sqrt{\sum_{i=1}^{n}\left(X_i\right)^2}\times\sqrt{\sum_{i=1}^{n}\left(Y_i\right)^2}}$$

由上面公式可计算出余弦值，计算结果夹角的余弦值越接近于 1，说明句子 A 和句子 B 越基本相似的。

（3）再结合欧式距离、余弦距离进行并行计算，得出文本相似度结果。

2. gensim 文本相似度分析算法

（1）gensim 使用流程；



（2）使用 jieba 库中的 cut 函数生成分词列表；

Texts = [list(jieba.cut(text)) for text in texts]

如：['我'],['们'],['是'],['G'],['市'],['残'],['疾'],['人']

（3）基于文本集建立词典，获取特征数；

corpora.Dictionary：建立词典，dictionary = corpora.Dictionary(texts)

len(dictionary.token2id)：词典中词的个数：

num_features = len(dictionary.token2id)

（4）基于词典建立语料库，语料库即存放稀疏向量的列表；

doc2bow 函数：将所有单词取集合，并对每个单词分配一个 ID 号；转换成稀疏向量，搜索词也转成稀疏向量。

corpus = [dictionary.doc2bow(text) for text in texts]

如：[[(0,1)],[(1,1)],[(1,1)],[(1,1)],[(1,1)],[(1,1)]…]

（5）用语料库训练 TF-IDF 模型：TF-IDF 是一种统计方法，用以评估一字词对于一个文件集或一个语料库中的其中一份文件的重要程度。

```
tfidf = models.TfidfModel(corpus)
new_vec = dictionary.doc2bow(list(jieba.cut(keyword)))
```

（6）相似度计算

```
index = similarities.SparseMatrixSimilarity(tfidf[corpus],
num_features); sim = index[tfidf[new_vec]]。
```

3. 结合 Python 相似度计算算法结果与 gensim 文本相似度分析算法结果对留言的答复意见从相关性角度给出评价。

4.3.2 完整性评价模型

1. 完整性即通顺与否的判定，即给定一个句子，要求判定所给的句子是否通顺。在本小问运用 N-Gram 判断句子是否通顺。首先使用 pyltp 用于分词和词性标注，首先加载分词和词性标注模型；加载训练数据，并对数据进行分词和词性标注，在句首句尾分别加上<s>和</s>作为句子开始和结束的标记。

2. 测试数据的加载方式与训练数据一致不再赘述，接下来就是对训练数据中标签为 0 的数据进行 1gram 和 2grams 的词性频率计数。

3.由于句子长短不一，计算出来的句子的概率差距甚远，所以需要对相同长度的句子进行一个聚类，然后用计算出来的概率值除以句子字长，这样才能保证句子的概率基本保持在一个较小的范围内，设置的阈值才能较好地将不同类型句子区分开来。

4.统计训练集中 2-Grams 概率的最小值，最大值，以及平均值，其中平均值将被用作判断句子好坏的阈值。

5.接着计算测试集中每个句子基于 2-Grams 的除以字长的概率值，然后与由训练集计算得到的与其等字长类的平均概率值进行对比。如果训练集中没有找到与测试集中某个句子的等长的句子，则测试集中该句子概率值直接去总体训练样本计算得到的概率值进行对比。最后对留言的答复意见从完整性角度给出评价。

5. 不足与改进

1.针对第一题，其模型几次测试结果在90%左右，但是这个结果可以说明分类模型需进行优化，本文主要是在数据预处理阶段，将所有大写转化为小写，在使用 jieba 分词之前，先用正则表达式处理数据，用于后续更好，更高效去除停用词，从而获取到更高质量的特征词，经过这两方面的处理，其结果增加了 3 个百分点。模型再一次优化可以在数据集划分的时候将参数 n_splits 的值调高，从而实现模型的优化。

2.针对第二题，文本实体识别方法提取人群与地点，存在一些标注语料老旧，覆盖不全的问题，因为近年来起名字的习惯用字与以往相比有很大的变化，以及各种复姓识别、国外译名、网络红人、虚拟人物和昵称的涌现；其次，基于统计机器学习的方法中，最大熵模型结构紧凑训练时间复杂性非常高，有时甚至导致训练代价难以承受，另外由于需要明确的归一化计算，导致开销比较大。我们应该多加运用半监督的学习和无监督的学习方法，采用未标注语料集等方法将逐步解决语料库不足的问题。

3.针对第三问，对于完整性评价模型中使用的 N-Gram 来判断句子是否通顺，其中使用 2-Grams 方法的瓶颈效果并不是太好，当下深度学习在 NLP 中应用很火热，未来可以深入学习以下深度学习在 NLP 中的应用然后再回过头来用深度学习的视角来重新看待这个问题；bert 参数没调好，但是目前对 bert 了解比较少不知该怎么调，而时间有限所以也就没进一步深入了，后续有时间学习一下 bert 再回过头来优化模型。

6. 参考文献

[1] python3 使用 fasttext 进行文本分类（一定要用 linux ）
https://blog.csdn.net/bingheshidai_1234/article/details/90056772,2016 年 5 月 12 日

[2] python 实现完整的 K-means 文本聚类算法,
https://blog.csdn.net/weixin_43718084/article/details/90231783,2019 年 9 月 15 日

[3] 韩中庚，数学建模方法及其应用（第二版），北京：高等教育出版社，2009 年

[4] 朱建宇. K 均值算法研究及其应用[D].大连:大连理工大学计算机软件与理论,2013.

[5] 许丽利.聚类分析的算法其应用[D].长春:吉林大学应用数学, 2010.

[6] 唐东明.聚类分析及其应用研究[D].成都:电子科技大学计算机应用技术，2006.

[7] 高滢.多关系聚类分析方法研究[D].长春:吉林大学计算机应用技术, 2008.

[8] 刘丽.聚类算法研究与应用[D].无锡:江南大学计算机应用技术，2013.

[9] 陈衡岳.聚类分析及聚类结果评估莫法研究[D].沈阳:东北大学计算机应用技推广赚钱 2006.

[10] 杨守建，陈恳.基于 Hopfield 神经网络的交通标志识别[].计算机工程与科学，2011,33(8):132-137.

[11] 仲云飞，梅一韬，吴邦彬，陈端.遗传算法优化 BP 神经网络在大坝扬压力预测中的应用.水电能源科学，2012(6):98-101.

[12] 刘晓强，杨燕华，赵跃.基于人工神经网络的软土地基沉降预测技术研究[].水道港口，2015, 36(6):574-577

[13]王玉龙，崔玉，李鹏，李锐.基于小波分析改进的神经网络模型电力系统负荷预测[J].电网与清洁能源，2015, 31(2):16-20