基于群众留言的文本数据挖掘

摘要

近年来，网络问政平台逐渐成为政府了解民意、汇聚民智、凝聚民气的重要渠道，于是越来越多人在平台上留言。但是工作人员在处理留言时，需要将留言按一定的划分体系，随着类与民意相关的样本数据的递交数量不断增多，工作人员的工作量暴增，且人工划分留言的效率低以及差错率高。另外，工作人员需要及时发现热点问题，以进行针对性的处理。因此，利用文本数据挖掘对于解决留言划分以及挖掘热点问题有重要的意义。

针对问题一，首先需对文本数据进行预处理：对数据进行缺失值处理；之后清理除数字、字母、汉字以外的其他字符；利用 pyltp 进行中文文本分词；构建 word2vec 模型；接着去除已分词文本中的停用词，通过 Tokenizer 将文本转化为数字列表，结合已建立的 word2vec 模型构造了词嵌入矩阵。经过数据预处理后，我们构建了文本分类卷积神经网络模型 TextCNN，在词嵌入后，对文本向量进行卷积核大小为 3，4，5 的卷积操作，经过最大池化后将池化的向量连接起来。之后通过 Flatten 层，将向量转变成一维，通过 Dropout 层防止过拟合，最后经过全连接层输出分类结果。训练过程中我们以改进的交叉熵也即 focal loss 作为损失函数，最终得出的 F-Score 有 0.8799，分类效果较佳。

针对问题二，首先需对数据进行预处理，步骤与问题一类似。然后以 IF-IDF 的方式构造文本向量，由此构建出基于余弦相似度的 single-pass 聚类模型。该模型不需事先知道聚类个数，其思想是每次读取新数据都和已经读取并聚类的数据进行比较，果余弦相似度大于某个阈值，就将其归为一类，否则自成一类。利用该模型我们将群众留言反映的同一个问题聚成一类，最后聚成了 1056 个类别，再通过我们构建的热度评价指标，计算出热度排名，得出热点问题。

针对问题三，需要构建答复意见的质量的评价方案，即建立评价模型。基于答复的相关性、完整性、可解释性、及时性这四个角度，建立答复意见质量的得分方程。其中，相关性通过留言文本与答复文本的相关性衡量，完整性通过答复文本对留言的命名实体覆盖率衡量，可解释性通过留言文本的命名实体数量与答复文本篇幅进行衡量，及时性通过答复时间与留言时间差进行衡量。分别求出答复的相关性、完整性、可解释性、及时性的得分，以它们得分的加权和作为答复意见质量的得分。

关键词：word2vec、TextCNN、TF-IDF、single-pass

Text data mining on mass messages

Summary

In recent years, the online political inquiry platform has gradually become an important channel for the government to understand public opinion and gather people's wisdom and morale. Therefore more and more people leave messages on the platform. However, when processing the messages, the staff needs to divide the messages according to a certain system. As the data grows rapidly, the workload of the staff increases sharply, and the efficiency of manual message division is low and the error rate is high. In addition, the staff needs to find out hotspot issues in time for targeted treatment. Therefore, using text mining is of great significance to solve the problem of message division and mining hot spots.

For the first problem, we preprocessed the text data: dealt with the missing data, then deleted characters except numbers, letters, and Chinese characters, used pyltp to segment Chinese text, constructed word2vec model, removed the stop words, converted the text into a list of numbers through Tokenizer, and then constructed a word embedding matrix in combination with the established word2vec model. After data preprocessing, we construct a text classification convolutional neural network model TextCNN. After word embedding, the text vector is convolved with a convolution kernel size of 3, 4, and 5 and we connected the pooled vectors after maximum pooling. Later, through the Flatten layer, the vector is transformed into one dimension, and the Dropout layer is used to prevent overfitting, and finally the classification result is output through the fully connected layer. During the training process, we used the improved cross entropy, also known as focal loss, as the loss function. The resulting F-Score is 0.8799, which indicates a satisfying classification effect.

For problem two, the data needs to be preprocessed as well, same as problem one. Then constructed the text vector using IF-IDF method, thus constructed a single-pass clustering model based on cosine similarity. The model does not need to know the number of clusters in advance. The principle behind it is to compare the new data with the data that has been read and clustered each time. And if the cosine similarity is greater than a certain threshold, it is classified as the same class. Otherwise, it is a new category. Using this model, we grouped the messages reflecting a same problem into one category, and then 1,056 categories were formed. Next, we calculated the popularity ranking through the heat evaluation index we constructed, and obtained hot issues.

In response to question three, it is asked to construct an evaluation plan for the quality of the replies, that is, to establish an evaluation model. Based on the four angles of relevance, completeness, interpretability, and timeliness of the reply, a scoring equation for the quality of the reply opinion is established. Among them, the relevance is measured by the relevance of the message text and the reply text, the integrity is measured by the coverage of the named entity of the message, the interpretability is measured by the number of named entities of the message text and the length of the reply text, and the timeliness is measured by the difference between reply time and message time. The scores of the relevance, completeness, interpretability and timeliness of the reply were obtained respectively, and the weighted sum of their scores was used as the score of the quality of the reply opinion.

Key Words： word2vec、TextCNN、TF-IDF、single-pass

目录

1、挖掘背景与目标

1.1 挖掘背景

近年来，网络问政平台逐渐成为政府了解民意、汇聚民智、凝聚民气的重要渠道，但是各类与民意相关的样本数据的递交数量不断增多，使得以人工进行留言分类和整理的相关部门面临极大的挑战。同时大数据、云计算等技术的发展，所建立的基于自然语言处理技术的智慧政务系统对于提升政府的管理水平和施政效率发挥极大的作用，因此建立智慧政务系统是社会治理创新发展的新趋势。

1.2 挖掘目标

本次建模基于收集自互联网公开来源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，利用自然语言处理和文本挖掘的方法达到以下三个目标：

（1） 群众留言分类

对于已知的留言内容以及一级标签，通过训练建立了留言内容的一级标签分类模型，该模型能够帮助工作人员自动对留言内容进行划分，达到减少工作量、提高效率、提高划分准确率的目的。并对所分类的结果进行评价。

（2） 热点问题挖掘

某一时段内群众集中反映的某一问题可称为热点问题，因此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，同时利用所定义的热点评价指标得出排名前 5 的热点问题以及对应的留言信息。

（3） 答复意见评论

利用相关部门对留言的答复意见，基于答复的相关性、完整性、可解释性等角度对答复意见的质量建立一套评价方案，并尝试实现查看效果。

2、问题分析

2.1 问题 1 的分析

1、对于网络问政平台的群众留言的分类：

(1) 由于所给的文本数据可能存在无用或多余的信息，因此需要对所给的文本数据进行预处理，包括：

① 缺失值处理

② 其他符号（除数字、字母和汉字之外）的清理

在进行文本数据分类时，若不清理与文本不相关的符号，则会在求解时造成非常大的阻碍，造成求解效率慢，求解结果准确率低。

③ 文本分词

文本由句子组成，而句子中由词语组成，对文本数据进行分类实则上是对文本数据的关键词进行分类，因此对文本数据进行分词很有必要。

④ 去除停用词

在文本数据中有可能存在出现频率高但信息率低的词语，去除停用词，提取的信息将会更有价值，对于文本分类结果也会更加准确。

(2) 将词语转换为向量

我们需要把文本数据转换成模型能够识别的方式，因此若想将所建立的模型应用于文本数据，则必须将文字转换为数字特征，将每条文本转换为数字列表。

2、利用求解得到的分类结果，根据 F-sore 指标对分类结果进行评价，同时可根据数据的实际情况，增加其他几个评价指标进行辅助性评价。

2.2 问题 2 的分析

题目要求我们对于留言进行分类，并根据热度的大小进行排序，因此涉及两个问题，一是聚类，而是排序。

1、留言聚类

（1）对于所得到的数据，首先需进项数据预处理（与问题一类似）。

（2）若要对多条文本数据进行聚类，必须要找到每条文本数据的特点，即“关键词”，因此需要寻找一种方法来评估一个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，该字词就是聚类的关键。

（3）聚类的方式

① 传统的聚类方式需要先确定聚类的个数以及阈值，而后再进行聚类。但这对于文本数据的聚类显然不太现实。

② 若聚类的方式是按照一定顺序依次读取文本数据，每次读取的新数据都和已经读取并聚类的数据进行比较，若按照一定规则找到相应的近似组别，则将这个新数据归入这个类中，反之，将这个新数据视为一个新类。反复执行，直到所有的数据都读完。这种方法不仅不需要事先知道聚类数，且也能保证聚类的精度。

（4）聚类的标准

在确定聚类的方式之后，我们需要考虑聚类的标准，怎么样判定一个文本数据是否应该归入已有的类别，实则也可以这样理解，怎样衡量两个文本之间的相似程度，因此，衡量文本之间的相似程度是确定聚类标准的关键。

2、根据热度对已经聚类的结果进行排序，因此热点的定义很重要。根据生活常识，一条新闻的热度高不高，取决于其点赞数，反对数以及讨论的时间跨度，因此可根据上述因素来制定热度的标准。

2.3 问题 3 的分析

基于答复的相关性、完整性、可解释性、及时性这四个角度，构建答复意见的质量的评价方案，即建立评价模型。需考虑在这四个角度中，答复与留言的关系，利用它们之间蕴含的数学关系建立数学公式求取在各个角度的评分，最后可以四个角度的评分加权和作为答复意见质量的评价得分，利用该得分对答复意见的质量进行评价

3、分析方法与过程

3.1 问题 1 的分析方法与过程

3.1.1 总体流程

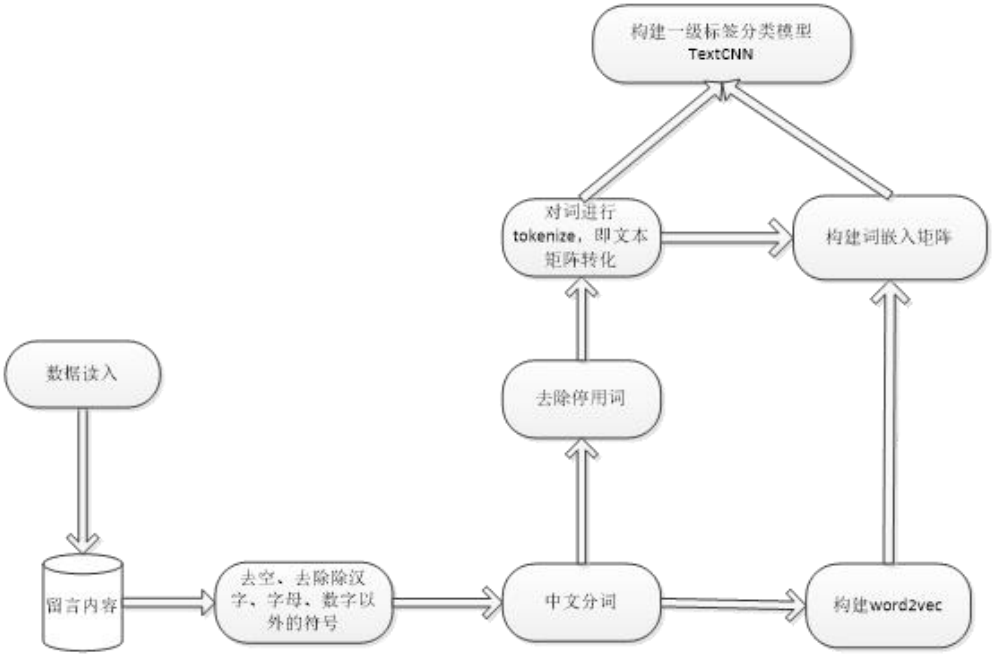

图 1 群众留言分类的总体流程图

3.1.2 具体步骤

3.1.2.1 读入数据

利用 pandas 读入附件的数据，查看前五条数据，见图 2：

	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\n\n\n\n\n\n\n\n\n\n\nA3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\n\n\n\n\n\n\n\n\n\n\n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\n\n\n\n\n\n\n\n\n\n\n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\n\n\n\n\n\n\n\n\n\n\nA1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\n\n\n\n\n\n\n\n\n\n\nA1区A2区华庭小区高层为二次供水，楼顶水箱...	城乡建设

图 2 pandas 读入附件前 5 条数据的结果

由图 2 可以看到存在许多'\t', '\n'这样的符号。

3.1.2.2 数据预处理

1、查看是否有缺失值

由图 3 可得，附件 1 每一列中都有 9210 条数据且没有缺失值。

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9210 entries, 0 to 9209
Data columns (total 6 columns):
留言编号    9210 non-null int64
留言用户    9210 non-null object
留言主题    9210 non-null object
留言时间    9210 non-null object
留言详情    9210 non-null object
一级标签    9210 non-null object
dtypes: int64(1), object(5)
memory usage: 431.8+ KB
```

图 3 缺失值查询结果

2、清理除数字、字母、汉字以外的其他字符

图 3 为清理结果：

	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目施工...	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用人行道...	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利用业...	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水是我...	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水是我...	城乡建设

图 3 清理结果

3、使用哈工大的 pyltp 分词工具进行分词

以第一条数据："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目施工围墙内。每天尤其上下班期间这条路上人流车流极多，安全隐患非常大。强烈请求文明城市 A 市，尽快整改这个极不文明的路段。" 为例：

（1）采用两种方式分词：

① jieba 分词的结果如下：

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 项目 施工 围墙 内 每天 尤其 上下班 期间 这 条 路上 人
流 车流 极 多 安全隐患 非常 大 强烈 请求 文明城市 A 市 尽快 整改 这个 极 不 文
明 的 路段

图 4 jieba 分词的结果

② pyltp 分词的结果如下：

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 项目 施工 围墙 内 每天 尤其 上下班 期间 这 条 路上 人
流 车流 极 多 安全 隐患 非常 大 强烈 请求 文明 城市 A 市 尽快 整改 这个 极 不 文
明 的 路段

图 5 pyltp 分词的结果

（2） 结果分析

对比分词结果可得，对于一些命名实体，如：“未管所”、“路灯杆”，“燕子山”、“安置房”等，jieba 分词并不能正确识别分类，而 pyltp 分词达到正确识别分类的要求，因此基于两种分词工具的分词效果，选择采用 pyltp 分词工具进行分词。

4、去除停用词

① 停用词的两个特征：一是极其普遍、出现频率高；二是包含信息量低，对文本标识无意义。例如：“的”，“了”,“我”，“我们”，“你们”。

② 去除停用词能够使模型训练速度更快，还能提高分类准确性。

③ 本文采用《哈工大停词表》，将一些高频词去掉，对比结果如下：

```
array(['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', '占用', '人行', '道路', '而且', '护栏', '锈迹', '斑斑', '随时', '可能',
       '倒塌', '危机', '过往', '行人', '和', '车辆', '安全', '请求', '有关', '部门', '牵头',
       '处理'], dtype='<U4')
```

图 6 未去除停用词前的效果

```
array(['位于', '书院', '路', '主干道', '在水一方', '大厦', '楼', '楼', '人为', '拆除', '水电',
       '设施', '后', '烂尾', '年', '护栏', '围', '占用', '人行', '道路', '护栏', '锈迹',
       '斑斑', '随时', '倒塌', '危机', '过往', '行人', '车辆', '安全', '请求', '部门', '牵头',
       '处理'], dtype='<U4')
```

图 7 去除停用词后的效果

5、用 word2vec 预训练词向量

(1) 当语料库比较大，则使用常规方法如 onehot 编码、词袋模型去表示词向量时，词向量的维度将会非常大，使得训练过程变得非常缓慢。若要解决词向量维度问题，则需要一种低维稠密的向量来表征词向量。

(2) Word2vec 可将词转换成向量。这是一种通过神经网络算法来训练 N-gram 语言模型，并在训练过程中求出 word 对应的 vector 的方法，而且它所生成的向量低维稠密，能够解决词向量高维度问题，因此我们采用 word2vec 预训练词向量。

6、使用 Tokenizer 将文字转换成数字特征，将每条文本转换为数字列表

使用 Keras 的 Tokenizer 模块实现转换：

(1) 当我们创建了一个 Tokenizer 对象后，使用该对象的 fit_on_texts()函数，对所输入文本的每个词利用词频进行编号，词频越大，编号越小。

(2) 根据每个词的编号将数据集中的每条文本转换为数字列表，使用该对象的 texts_to_sequences()函数，将每条文本转变成一个向量。

(3) 由于每句话的长度不唯一，需要将每句话的长度设置一个固定值。将超过固定值的部分截掉，不足的在最前面用 0 填充。

以第二、第三条文本为例，使用 Keras 的 Tokenizer 模块实现转换，所得到的文本向量结果见图 8 和图 9：

```
array([    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,
           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,
           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,
           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,
           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,
           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,
           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,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,   704,  5013,   252,  5272,
       35819,  2723,   245,   245,  2631,   690,  2096,   307,    15,
        6042,     6,  5541,  4090,  1193,  3222,   490,  5541, 15344,
        8693,  1699,  3351,  6043,  3659,  1665,   447,    72,   163,
           9,  2847,    74])
```

图 8 第二条文本向量结果

```
array([ 2289,    29,   176,    95,    48,   368, 17307,    56,    20,
         473,    95,  5131,  6635,  5131,   307,   518,    40,   286,
         351,  3725,    24,    29,    29,  1673,    95,    48,    54,
         356,    15,    20,    80,   226,    26,   176,  1988,  2116,
       15345,    95,  3725,    29,   268,   719,   254,   101,    40,
        6224,    95,  7653,   171,  1106, 35826,   254,   363,    15,
       17308,  1572,  1106,   210,   171,  1646,    62,  1110,   210,
        1262,   465,  1359,  4091,  4331,  1646,    22,    77,    24,
         732,   919,   210,  1382,    26,    29,    72,   118,   848,
         442,   922,    12,    29,    34,    95,   222,   276,   608,
          95,  5544,   186,    37,   469, 15345,    95,   571,   247,
         453,   731,    12, 25137,  5545,  1646,   240,  2230,    26,
       10915,   108,   103,  1646, 35827,    30,  1956,   240,    29,
         240,   102,    95,  2945,    52,    26,    29,   648,    31,
          27,    40,  2154, 35828,  1188,  1646,   537,    34,  6840,
         537,    26,    29,  6840,    34,  6840,  1188,  1646,   102,
          95,  2945,    74,   240,   366,    29,   176,     6, 15345,
          95,  5544,    27,  1646, 35829,   233,    24,   404,    29,
        1189,    30,    95,   270,  2478,  5543,    97,   233,    24,
         357,   195,  1646,  2978,  1111,  1029,  3175,    36,     1,
        1557,   101,   110,   853,    34,   240,  2352,   104,    95,
        1558,  1499,   546,  5543,   637,  1111,  1029,   546,  7339,
           8,  2754,  2882,  2848,  1557,   101,    24,  1305,   268,
         357,  2814,    59,  1825,   806,   323,    51,  2946,  4410,
         127,   101,    48,   182,  3352,    56,   219,   504,  2194,
       10916,    20,    34,   673,    29,   567,   949,    26,  1557,
         127,   101,    29,  1673,    29,  3398,   474, 25138,    29,
         567,   949,    26,  1557,   127,   101,  1557,   127, 15346,
        1725,    29,  1465,   182,   648,   474,    95,    82,    19,
        1222,   195, 15345,    95,  1152,    26,  1557,   101,   110,
        7654,    26,  1433,    31,    52,  3725,  3223,    29,   356,
        1499,  1736,    24,   101,   110,    22,   256,  2754,    59,
          24,    78,   806,  3301,  1097, 20328,    26,   366,    29,
         416,   303,  1153])
```

图 9 第三条文本向量结果

3.1.3 TextCNN 模型的建立与求解

1、模型介绍

CNN 通常被认为是 CV 领域且用于计算机视觉方向的工作。但在 2014 年，Yoon Kim 针对 CNN 的输入层做了一些变形，提出了文本分类模型 TextCNN。与传统图像的 CNN 网络相比, TextCNN 在网络结构上没有任何变化（甚至更加简单了）

图 10 为 TextCNN 的模型框架：

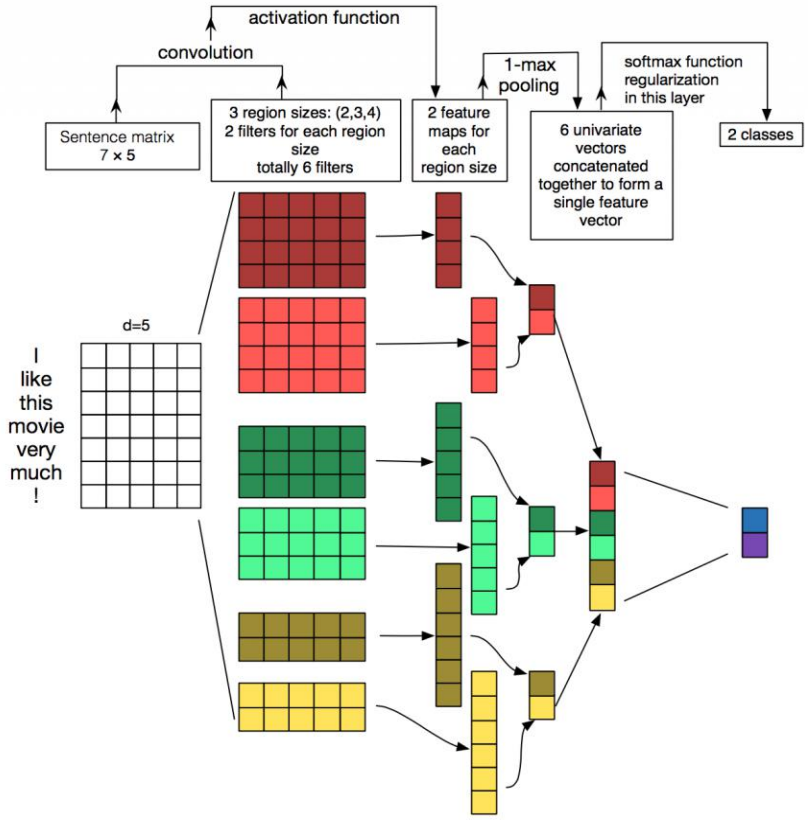

图 10 TextCNN 的模型框架

（1）上图很好地诠释了模型的框架，从图可得，TextCNN 实际只有一层卷积和一层 max-pooling，最后将输出外接 softmax 来进行 n 分类。

（2）假设我们有一些句子需要对其进行分类

句子中每个词是由 n 维词向量组成的，即输入矩阵大小为 m*n，其中 m 为句子长度，CNN 需要对输入样本进行卷积操作：

① 对于文本数据，filter 不再横向滑动，仅仅是向下移动，有点类似于 N-gram 在提取词与词间的局部相关性。

② 上图中共有三种步长策略，分别是 2,3,4，每个步长都有两个 filter（实际训练时 filter 数量会很多）。在不同词窗上应用不同 filter，最终得到 6 个卷积后的向量。

③ 而后对每一个向量进行最大化池化操作并拼接各个池化值，最终得到这个句子的特征表示，将这个句子向量丢给分类器进行分类，至此完成整个流程。

（3）TextCNN 的网络结构

① 嵌入层 (Embedding layer)

通过一个隐藏层，将 one-hot 编码的词投影到一个低维空间中。其本质上是特征提取器，能够在指定维度中编码语义特征，使得语义相近的词，它们的欧氏距离或余弦距离也比较小。

② 卷积层 (Convolution Layer)

A. 在处理图像数据时，CNN 使用的卷积核的宽度和高度的一样的，但是在 text-CNN 中，卷积核的宽度是与词向量的维度一致。这是由于我们输入的每一行向量代表一个词，在抽取特征的过程中，词作为文本的最小粒度。而高度和 CNN 一样，可以自行设置（通常取值 2,3,4,5），高度就类似于 n-gram。

B. 由于我们的输入是一个句子，句子中相邻的词之间关联性很高，因此，当我们用卷积核进行卷积时，不仅考虑了词义而且考虑了词序及其上下文。

③ 池化层 (Pooling Layer)

A. 因为在卷积过程中我们使用了不同高度的卷积核，使得我们通过卷积层后得到的向量维度会不一致，所以在池化层中，我们使用 1-Max-pooling 对每个特征向量池化成一个值，即抽取每个特征向量的最大值表示该特征，而且认为这个最大值表示的是最重要的特征。

B. 当我们对所有特征向量进行 1-Max-Pooling 之后，还需要将每个值给拼接起来，得到池化层最终的特征向量。

C. 在池化层到全连接层之前可以加上 dropout 防止过拟合。

④ 全连接层 (Full connected Layer)

以 softmax 作为激活函数，这一层起到“分类器”的作用

2、模型建立

基于实际的问题需求，我们提出了一个 CNN 文本分类模型：

（1）该模型使用预训练的词向量矩阵进行词嵌入，经过 3 层卷积并进行池化，再合并 3 层池化后提取得到特征向量，而后进行 Flatten 处理和 Dropout，最后经过全连接层输出各个属于各个类别的概率值。模型架构如图 11 所示：

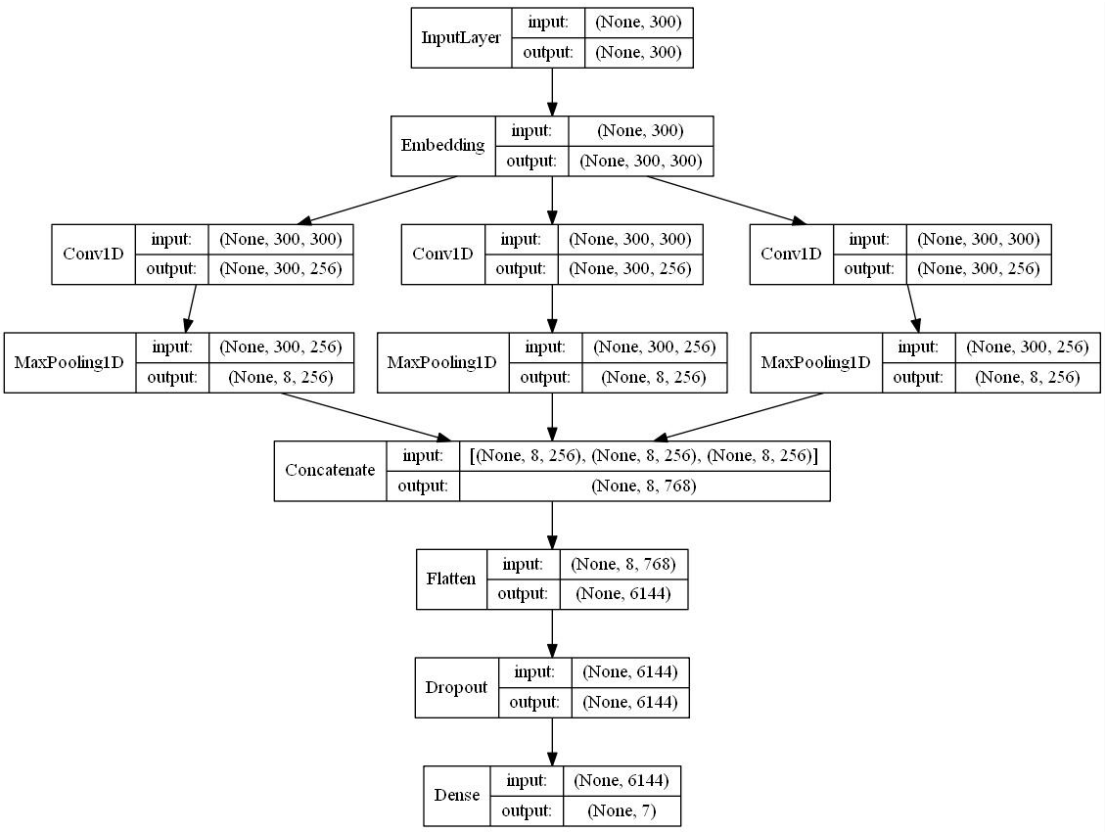

图 11 CNN 文本分类模型的架构

（2）模型分为 8 层：

① InputLayer：模型输入编码后的文本数据，这里我们限制每个句子的最大长度为 300，多余的截断，未满 300 的补 0。

② Embedding：对文本数据进行词嵌入，该层会在每次迭代中训练词向量，训练出来的词向量可以更好的适应自然语言处理任务。

③ Conv1D：对文本数据进行卷积核大小为 3，4，5 的卷积操作，提取文本特征，本文采用的是 ReLU 作为激活函数，可以防止反向传播中的梯度问题(梯度消失、梯度爆炸)。

④ Maxpooling1D：对数据进行降维，抽取每个特征向量的最大值表示该特征，降低了全连接层的复杂度 。

⑤ Concatenate：将池化后的向量连接起来。

⑥ Flatten：用于将数据“压平”，即把多维的输入一维化，起到从卷积层到全连接层的过渡作用

⑦ Dropout：让部分神经元随机失活，这样子就不会使模型太依赖于某些局部特征，可以防止过拟合。

⑧ Dense：全连接层，起到“分类器”的作用。

（3）损失函数

这里以交叉熵作为损失函数。

以二分类为例：

$$L = -y\log y' + (1-y)\log(1-y') = \begin{cases} -\log y', y = 1 \\ -\log(1-y'), y = 0 \end{cases}$$

y'是经过激活函数的输出，所以在 0-1 之间。可见交叉熵对于正样本而言，输出概率越大损失越小。对于负样本而言，输出概率越小则损失越小。

（4）评价指标

对于所建立模型的分类结果，采用 F-Score（公式如下）作为主要评价指标，同时以精确率、混淆矩阵作为辅助评价指标。

$$F_1 = \frac{1}{\mathrm{n}} \frac{2P_i R_i}{P_i + R_i} (P_i: 第i类查准率、R_i：第i类查全率)$$

3、模型结果

利用所建立的模型进行留言分类，同时对于分类结果进行评价：

（1）F-Score 的评价结果，见图 12

由图可得经过 15 词 Epoch 后，准确率达到 0.847，F1_Score 值为 0.846。

```
Epoch 1/15
7828/7828 [==============================] - 11s 1ms/step - loss: 1.6056 - acc: 0.4239
Epoch 2/15
7828/7828 [==============================] - 7s 952us/step - loss: 0.9345 - acc: 0.6906
Epoch 3/15
7828/7828 [==============================] - 7s 947us/step - loss: 0.7004 - acc: 0.7639
Epoch 4/15
7828/7828 [==============================] - 7s 946us/step - loss: 0.5608 - acc: 0.8164
Epoch 5/15
7828/7828 [==============================] - 7s 956us/step - loss: 0.4687 - acc: 0.8516
Epoch 6/15
7828/7828 [==============================] - 7s 957us/step - loss: 0.4005 - acc: 0.8740
Epoch 7/15
7828/7828 [==============================] - 7s 954us/step - loss: 0.3540 - acc: 0.8964
Epoch 8/15
7828/7828 [==============================] - 7s 955us/step - loss: 0.3088 - acc: 0.9116
Epoch 9/15
7828/7828 [==============================] - 7s 957us/step - loss: 0.2651 - acc: 0.9255
Epoch 10/15
7828/7828 [==============================] - 7s 957us/step - loss: 0.2294 - acc: 0.9388
Epoch 11/15
7828/7828 [==============================] - 7s 956us/step - loss: 0.2058 - acc: 0.9447
Epoch 12/15
7828/7828 [==============================] - 7s 955us/step - loss: 0.1800 - acc: 0.9562
Epoch 13/15
7828/7828 [==============================] - 7s 956us/step - loss: 0.1591 - acc: 0.9617
Epoch 14/15
7828/7828 [==============================] - 7s 958us/step - loss: 0.1396 - acc: 0.9713
Epoch 15/15
7828/7828 [==============================] - 8s 960us/step - loss: 0.1225 - acc: 0.9755
1382/1382 [==============================] - 1s 630us/step
test_accuracy:  0.8473227206946454
F1_Score:  0.8468703182573682
```

图 12　F-Score 的评价结果

（2）混淆矩阵的评价结果，见图 13：

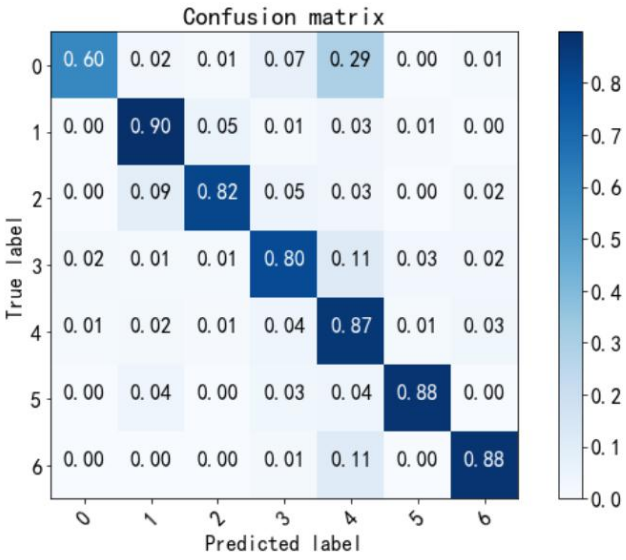

图 13　混淆矩阵的评价结果

3.1.4 结果分析与改进

1、结果分析

（1）由混淆矩阵的定义(见表 1)可得，黄色阴影部分表示模型正确分类的概率。

表 1 混淆矩阵的定义

混淆矩阵		真实分类			
		分类 1	分类 2	…	分类 n
预测分类	分类 1	A11	A12	…	A1n
	分类 2	A21	A22	…	A2n
	…	…	…	…	…
	分类 n	An1	An2	…	Ann

（2）由图 13 可得，第一类的正确分类的概率较低，通过观察数据的分布我们发现这是由于文本数据不均衡所造成的。文本中各类标签的占比如图 14 所示：

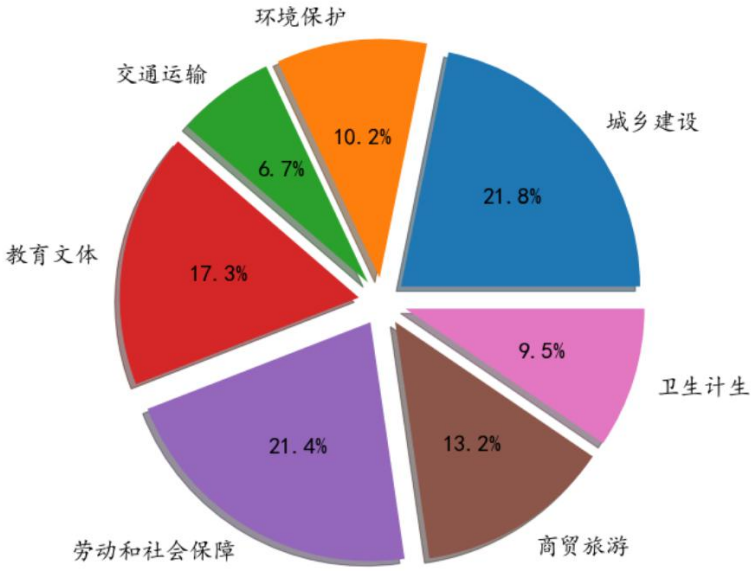

图 14　文本中各类标签的占比情况

由图可得，占比低的文本类与占比高的文本类的文本数量差距最高可达到三倍以上，因此若想提高第一类的正确分类概率，必须解决文本数据不均衡的问题。

2、模型改进

（1）对于文本数据不均衡，常规的采样方法以及 SMOTE 数据合成并不适用。为了解决样本不均衡问题，本文采用 KaiMing He 等人提出的 Focal loss 作为损失函数，这个损失函数是在标准交叉熵损失基础上修改得到的。该函数可以通过减少易分类样本的权重，使得模型在训练时更专注于难分类的样本。

（2）改进交叉熵损失函数

改进后的交叉熵，即 Focal loss：

$$L_{\text{fl}} = \begin{cases} -(1-y')^{\gamma}\log y', y=1 \\ -y'^{\gamma}\log(1-y'), y=0 \end{cases}$$

A. 首先在原有的基础上加了一个因子，其中 gamma>0 使得减少易分类样本的损失。使得更关注于困难的、错分的样本。

例：gamma 为 2

a. 对于正类样本而言，预测结果为 0.95 肯定是简单样本，所以(1-0.95)的 gamma 次方就会很小，这时损失函数值就变得更小。而预测概率为 0.3 的样本其损失相对很大。

b. 对于负类样本而言，同样，预测 0.1 的结果应当远比预测 0.7 的样本损失值要小得多。

c. 对于预测概率为 0.5 时，损失只减少了 0.25 倍，所以更加关注于这种难以区分的样本。这样减少了简单样本的影响，大量预测概率很小的样本叠加起来后的效应才可能比较有效。

B. 加入平衡因子 alpha，用来平衡正负样本本身的比例不均：

$$L_{\text{ft}} = \begin{cases} -\alpha(1-y')^{\gamma}\log y', y=1 \\ -(1-\alpha)\ y'^{\gamma}\log(1-y'), y=0 \end{cases}$$

3、模型改进后的结果

图 15 和图 16 分别为 F-Score、混淆矩阵（使用 focal loss 求解）评价的结果：

（1）由图可得，使用 focal loss 作为损失函数，准确率达到 0.87988，F-Score 值为 0.87995。F-Score 提升了 3 个百分点，第一类的错分率有了明显下降。这说明损失函数成功的放大了第一个类别的权重，会使模型更重视第一个 label 的正确预测。

（2）模型改进后，较好地处理文本数据不平衡所产生的问题，使得结果更优化。

```
Epoch 1/15
7828/7828 [==============================] - 13s 2ms/step - loss: 1.3285 - acc: 0.5327
Epoch 2/15
7828/7828 [==============================] - 12s 2ms/step - loss: 0.6367 - acc: 0.7954
Epoch 3/15
7828/7828 [==============================] - 12s 2ms/step - loss: 0.4016 - acc: 0.8760
Epoch 4/15
7828/7828 [==============================] - 12s 2ms/step - loss: 0.2766 - acc: 0.9203
Epoch 5/15
7828/7828 [==============================] - 12s 2ms/step - loss: 0.1826 - acc: 0.9525
Epoch 6/15
7828/7828 [==============================] - 12s 2ms/step - loss: 0.1289 - acc: 0.9687
Epoch 7/15
7828/7828 [==============================] - 12s 2ms/step - loss: 0.0807 - acc: 0.9867
Epoch 8/15
7828/7828 [==============================] - 12s 2ms/step - loss: 0.0572 - acc: 0.9907
Epoch 9/15
7828/7828 [==============================] - 12s 2ms/step - loss: 0.0435 - acc: 0.9940
Epoch 10/15
7828/7828 [==============================] - 12s 2ms/step - loss: 0.0312 - acc: 0.9971
Epoch 11/15
7828/7828 [==============================] - 12s 2ms/step - loss: 0.0257 - acc: 0.9978
Epoch 12/15
7828/7828 [==============================] - 12s 2ms/step - loss: 0.0240 - acc: 0.9967
Epoch 13/15
7828/7828 [==============================] - 12s 2ms/step - loss: 0.0307 - acc: 0.9943
Epoch 14/15
7828/7828 [==============================] - 12s 2ms/step - loss: 0.0202 - acc: 0.9971
Epoch 15/15
7828/7828 [==============================] - 12s 2ms/step - loss: 0.0157 - acc: 0.9982
1382/1382 [==============================] - 1s 611us/step
test_accuracy:  0.8798842257597684
F1_Score:  0.8799501021156385
```

图 15　F-Score 的评价结果

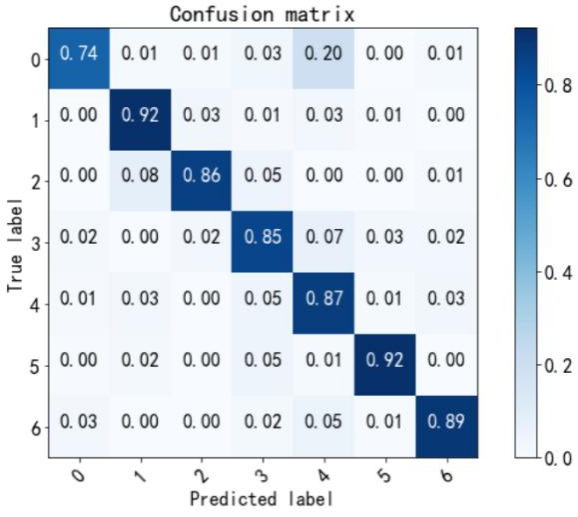

图 16　混淆矩阵（使用 focal loss 作为损失函数）评价结果

1、模型的收敛性

以训练过程中训练集的损失变化以及验证集的损失变化作为评价模型的收敛性的依据，结果如图 17 所示：

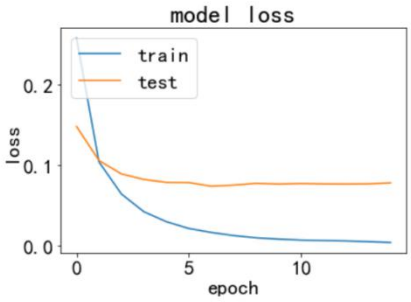

图 17　训练集的损失变化以及验证集的损失变化

从图中显然发现，随着 epoch 次数的增加，在训练集上的 loss 以及在验证集上的 epoch 都趋向于一个稳定值，说明我们的算法是高效且收敛的。

3.2 问题 2 的方法与流程

3.2.1 总体流程

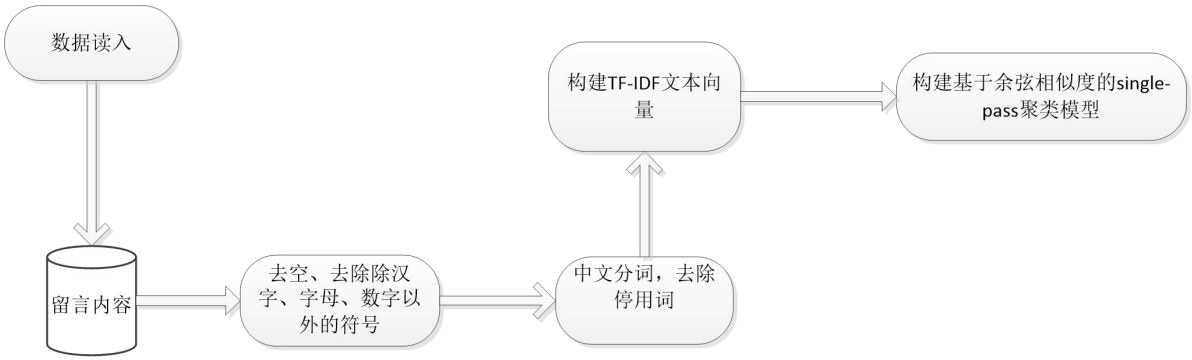

图 18　问题二的总体流程图

3.2.2 具体步骤

3.2.2.1 数据预处理

数据预处理方式与问题一的处理方式类似，则不具体描述。

3.2.2.2 构建 TF-IDF 文本向量

1、TF-IDF

（1）概述

① TF-IDF 又称为词频-逆文档频率，是一种统计方法，用以评估一个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。

② 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。

③ 主要思想：若某个单词在一篇文章中出现的频率 TF 高，而在其他文章中出现频率低，则认为此词或者短语具有很好的类别区分能力，适合用来分类。

（2）TF

① TF 是 Term Frequency 的缩写，即词频，表示词条在文本中出现的频率。

② 公式：

$$tf_{ij} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

即

$$TF_{\mathrm{w}} = \frac{\text{在某一类中词条w出现的次数}}{\text{该类中所有词条的数目}}$$

（3）IDF

① IDF 是 Inverse Document Frequency 的缩写，即逆文档频率，由总文件数目除以包含该词语的文件的数目，再将得到的商取对数得到。

② 如果包含词条 t 的文档越少，IDF 越大，则说明词条具有很好的类别区分能力。

③ 公式：

$$idf_i = \log \frac{|D|}{|\{j : t_{\mathrm{i}} \in d_j\}|}$$

其中：

$|\mathrm{D}|$：语料库中的文件总数；

$|\{\mathrm{j} : \mathrm{t_i} \in \mathrm{d_j}\}|$：包含词语 t_i 的文件数目（即 $n_{i,j} \neq 0$ 的文件数目）。若该词语不在语料库中，就会导致分母为零，因此一般情况下使用 $1+|\{\mathrm{j} : \mathrm{t_i} \in \mathrm{d_j}\}|$

（3）TF-IDF 实际上是 TF 与 IDF 相乘，即：

$$TF\text{-}IDF = TF \times IDF$$

2、构造 TF-IDF 文本向量

利用题目所给的数据，对留言内容构造 TF-IDF 文本向量，以第二条数据为例，结果如下图：

```
[(1, 0.07474649098311782),
 (5, 0.10893738626607714),
 (8, 0.3154442978518353),
 (16, 0.07335445160461108),
 (18, 0.24239936505655563),
 (39, 0.275839177452104),
 (40, 0.9007926696961249),
 (41, 0.41763399911674604),
 (42, 0.2982626292304681),
 (43, 0.38579951273882235),
 (44, 0.5073959250350408),
 (45, 0.3600265768328639),
 (46, 0.3076176608922802),
 (47, 0.5214805660332634),
 (48, 0.3833671540359587),
 (49, 0.2589787829776743),
 (50, 0.5073959250350408),
 (51, 1.299476345785946),
 (52, 0.8043617609799412),
 (53, 0.8043617609799412),
```

图 19 第二条数据构造 TF-IDF 文本向量的部分结果

由图可得，小括号左边是词编号，右边则是词的 TF-IDF 值。（这里只截取部分）

3.2.3 基于余弦相似度的 single-pass 聚类模型的建立

1. 模型介绍

（1）余弦相似度

① 可将其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：

a. 若夹角为 0 度，意味着方向相同、线段重合

b. 若夹角为 90 度，意味着形成直角，方向完全不相似

c. 若夹角为 180 度，意味着方向正好相反

则，可通过夹角的大小，来判断向量的相似程度。夹角越小，就代表越相似。

以二维空间为例，如图 20 所示：

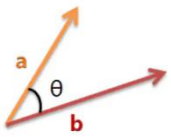

图 20 二维空间的夹角

② 公式

$$\cos\theta = \frac{\sum_{i=1}^{n}(A_i \times B_j)}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_j)^2}} = \frac{A \bullet B}{|A| \times |B|}$$

（2）Single-pass

① 对于传统的聚类，需要确定聚类个数，聚类的阈值也不好确定，例如K-means，dbscan。

② 对于 single-pass 来说，其主要思想是按一定顺序依次读取数据，每次读取的新数据都和已经读取并聚类的数据进行比较：

a. 若按照一定规则找到相应的近似组别，则将这个新数据归入这个类中，

b. 若没有，则将这个新数据视为一个新类。

依照上述规则反复执行，直到所有的数据都读完，整个过程只对数据进行一次读取（single）。因此 single-pass 不需要事先知道聚类数，其流程图见图 21：

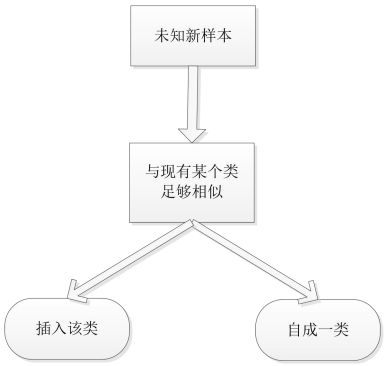

图 21 single-pass 的流程图

2、模型建立

利用所给的留言内容所求解得到的基于 TF-IDF 的文本向量，构建基于余弦相似度的 single-pass 聚类模型：

（1）规则

对于未知样本与已有的第 i 类中的文本向量之间的余弦相似度 $\cos\theta$：

$$\cos\theta \begin{cases} > 0.1, & \text{未知样本与第i类文本向量聚成一类} \\ \leq 0.1, & \text{未知样本自成一类} \end{cases}$$

（2）算法的伪码流程图如下图：

```
Begin
    输入阈值0.1
    输入文本特征向量
    将第一个文本特征向量作为第一个类，并设为该类的中心
    While(存在未遍历的文本特征向量)
    {
        While(存在未遍历的类中心)
        {
            计算文本特征向量与类中心的相似度
            记录最大的余弦相似度的类与相似度值
        }
        If(最大相似度值 > 0.1)
        {
            将该文本向量加入到最大相似度的类中
            更新该类的中心
        }
        else{
            新建一个类
        }
    }
End
```

图 22 模型的算法伪代码

（3）构建热度评价指标

① 假设

- k：聚成的某个类中的样本数
- P：为该类中所有留言内容的点赞数之和
- N：该类中所有留言内容的反对数之和
- Begin：该类中留言内容最早出现的时间
- End：该类中留言内容最晚出现的时间

② 热度评价指标

$$\mathrm{h}=\frac{w_1k+w_2P-w_3N}{begin-end}$$

其中，$\mathrm{w}_1,\mathrm{w}_2,\mathrm{w}_3$分别为$\mathrm{k},P,N$的权重，且设$\mathrm{w}_1=1,\mathrm{w}_2=0.5,\mathrm{w}_3=0.5$；注意 begin-end 需要手动换算成天数。

③ 意义

当某类的样本数越大、点赞数之和越大、(begin-end)越小时，则在一小段时间内，居民集中反映某一个热点问题，即其热度 h 也就越大，反之越小。

3.2.4 模型结果

1. 聚类结果

（1）根据所构建的模型对数据进行聚类，从中提取主题关键字与中心句，而后将聚类结果按 k 值进行降序排序，选取前五个，结果如下图：

.....
得到的主题数量有: 1056 个 ...
.....

【主题索引】:30
【数量】: 59
【主题关键词】: 车位,职工,销售,捆绑,广铁集团,购房,购买,领导,铁路职工,业主
【主题中心句】:
投诉：武广新城片区伊景园滨河苑为广铁集团的定向商品房，在未取得预售资格强行逼迫职工缴纳18.5万购房款且不签购房合同，工地停工1年多，现广铁集团又强烈要求捆绑购买12万车位一个，不买就取消购房资格，捆绑车位销售明显与中央文件及法律法规不符，但辛辛苦苦一辈子的老百姓房都买不起，要贷款，何来钱买车位
A市武广新城片区下的伊景园.滨河苑是广铁集团铁路职工的定向商品房，之前已经统一交了18.5万的认购款，但没有正规的合同，现在集团下发文件，强制要求职工再交12万的车位费，不交就取消购房资格，捆绑销售车位
商品房伊景园滨河苑项目是由A市政府办牵头为广铁集团铁路职工定向销售的楼盘，作为集团的一名退休员工，我深深感觉到政府对铁路员工的关怀，在广铁辛苦几十年，住的是职工原来配置的福利房，遇到这样一个利好消息十分激动，现在到了缴纳认购款的时候，却被告知除了要缴纳18.5万的认购款还要缴纳12万元的钱买车位，不然就取消购房资格

图 23　聚类排序第一类的结果

【主题索引】:35
【数量】: 58
【主题关键词】: 幼儿园,小区,业主,孩子,教育局,家长,相关,要求,开发商,幼儿
【主题中心句】:
尊敬的A市委市政府领导：我们是西地省A市A3区咸嘉湖街道臼鹤咀社区高心麓城小区业主，我小区原配套幼儿园被开发商租赁给巴罗比幼儿园无证办学长达7年之久，经小区业主投诉后，今年6月17日，A3区教育局对巴罗比幼儿园做出了"责令你园停止办学，退还所收费用，并对举办者处违法所得一倍的罚款，请在接到本处罚决定书起15日内停止一切招生及教育教学行为，退还所收费用"的行政处罚（岳教民幼处字[2019]1号），截止至2019年7月25日，园方无视政府部门行政处罚，不愿移交本该属于全体业主的配套幼儿园给区教育局，仍继续在招生、教学，并广泛散布今年9月幼儿园将转成普惠制幼儿园谣言迷惑广大业主，严重损害了广大业主的利益
我是A3区高心麓城小区业主，我小区配套幼儿园无证办学八年之久（经A3区教育局证实为无证办学），但仍然在继续办学中，且收费高昂（最低收费为6000元每月）而高心麓城小区为平民小区，业主根本无法承担这起这么昂贵的学费，经业主投诉被A市政法频道曝光后，A3区教育局工作人员表示今年9月份，小区幼儿园将被强制转为普惠制幼儿园，可据小区业主了解到，幼儿园已在招收下半年生源，收取的预收学费仍为近7000元每月，现要求A3区教育局公布高心麓城小区幼儿园转普惠制幼儿园工作情况，并回复今年9月是否小区适龄儿童都能进入小区幼儿园读书，享受普惠制幼儿园的待遇
我是A3区桐梓坡路臼鸽咀社区高心麓城小区的业主，今年9月，在我们小区业主多次投诉，强烈要求取缔无证办园的芭比罗幼儿园在我小区办园后，A3区教育局对我小区业主公示了关于要求芭比罗幼儿停招招生办学且将我小区配套幼儿园举办成公办园的A3区教育局红头文件，时至今日，事情已过去3个月之久，但芭比罗幼儿园根本没有搬出，停止办学，还打着下学期会在此地办普惠园的幌子在咸嘉新村广场虚假招生

图 24　聚类排序第二类的结果

【主题索引】:66
【数量】: 41
【主题关键词】: 国家,加快,背景,市委,市政府,建设,中心,刻不容缓,大力,城市
【主题中心句】:
作为内陆大省的A市自古以来就是国家重要的中心城市，但随着改革开放以来一批沿海城市和内陆强省会城市发展极为迅速，先后获得国家中心城市且开放程度飞快，外资国际巨头企业总部铺天盖地，作为省会A市还是小农经济以本土品牌为主，在国家大力倡导经济转型升级的大背景下不知市委市政府相关部门有何具体政策和行动加快国家中心城市的步伐，否则西地省人口流失一落千丈
地处中部大省的A市常年人口流失严重，在周边省市已相继落下多个国家中心城市，且国家新区招商引资都在上万亿，地铁建设一年开通好几条，A市似乎没点动静，在人口出生率快速下降和产业饱和度高的大背景下，加快产业布局尤其是国际行业巨头总部落地刻不容缓，同时加快交通基础设施建设力度，希望市委市政府加快国家中心城市建设有何具体政策和行动
在国家大力倡导加快制造业转型升级的大背景下，作为内陆大省A市常年都是低端落后本土企业为主，人口流失严重，近年周边城市外资巨头总部布天盖地，希望市委市政府加快招商引资尤其是外资巨头总部刻不容缓有何具体行动

图 25　聚类排序第三类的结果

【主题索引】:31
【数量】: 40
【主题关键词】: 搅拌站,小区,居民,影响,噪音,新城,相关,问题,生活,居民区
【主题中心句】:
开发商把特大型搅拌站，水泥厂从绿心范围内搬迁到A市暮云街道丽发新城的居民区，产生的噪音和扬尘等环境污染的问题，严重影响了周边居民的正常生活
我是A市暮云街道丽发新城小区的一名业主,向领导反映开发商在居民区附近建搅拌站,每天噪音影响根本无法睡觉,请领导关注,早日撤销该搅拌站.
开发商在A市暮云街道丽发新城社区附近百米范围内修建搅拌厂，整天尘土飞扬，噪声嘈杂，不仅污染了社区环境，还严重影响了我们业主的身心健康，所有业主都苦不堪言，恳请将搅拌站搬离居民区，换我们一个安静整洁健康的生活环境

图 26　聚类排序第四类的结果

【主题索引】:4
【数量】: 33
【主题关键词】: 施工,夜间,扰民,相关,部门,噪音,工地,投诉,居民,进行
【主题中心句】:
我们天天给A2区城管打投诉电话，打了无数个电话，一遍又一遍重复着长大建设集团彻夜施工噪音扰民，百姓无法正常休息，请求政府干预制止，然而每次得到都是千篇一律的回复"我们会派人处理"或者"今天他们办理了夜间施工证"，结果依旧是施工从来没有停止过，最好的效果是城管来人时停一会儿，走后即马上又继续开工
我是A市A2区莲花小区的居民，A2区竹塘西路—新姚路施工项目在2019年12月18日夜间23:00开始一直在施工，机器轰鸣，噪音大到令人抓狂，施工项目周边(最近的小区位置不超过100米)都住了莲花小区和银杏小区的居民，在这样大的噪音下，居民的生命健康都受到了很大的损伤，更别提夜间休息不好第二天还要上班的情况.....再者，如果用专业的仪器测量，这个施工产生的噪音应该远远超过了国家规定的《中华人民共和国环境噪声污染防治法》的夜间噪音限制(55分贝)
A市A3区银杉路阳光丽城小区北面道路施工，从今年5月份直到现在几乎每天通宵作业，压路机晚上可以把整栋高楼像地震一样咄咄作响，道路施工带起的扬尘致使小区靠北窗户都不敢开，道路施工大片大片的黄土黄沙裸露在外，小区居民无数次拨打12345热线，最后都是城管象征性来一下停工不超过一刻钟马上更加变本加厉，A市作为所谓的文明城市，就在距离市政府直线距离不到一公里如此野蛮施工野蛮作业，造成周边环境尘土飞扬，噪声通宵扰民，请求相关部门处理此事，如果让我来为A市文明城市投票我百分百投否决票，跟一个省会的文明相差甚远，深表失望

图 27　聚类排序第五类的结果

由上图可得，一共聚成 1056 个类，按 k 值排名前五类的 k 值分别为 59, 58, 41,

40, 33。

（2）按照题目要求，将聚类后的数据按类别写入表格，构造“热点问题留言明细表”，第一类的部分结果如下图（由于第一类的 k 值较大，因此只展示部分结果）：

	A	B	C	D	E	F	G	H
1	问题ID	留言编号	留言用户	留言主题	留言时间	留言详情	点赞数	反对数
2	1	188801	A909180	投诉滨河苑针对广铁职工购房的	2019-08-01 00:00:00	尊敬的张市长，您好！我叫李建义，来自湖北仙桃，虽然已经在广铁集团	0	0
3	1	190337	A0009051	关于伊景园滨河苑捆绑销售车位	2019-08-23 12:22:00	投诉伊景园.滨河苑开发商捆绑销售车位！A市武广新城片区下的伊景园.滨	0	0
4	1	191001	A909171	A市伊景园滨河苑协商要求购房	2019-08-16 09:21:33	商品房伊景园滨河苑项目是由A市政府办牵头为广铁集团铁路职工定向销	12	1
5	1	192739	A909188	请政府救救广铁集团的职工吧	2019-09-01 20:32:26	实在搞不懂买个单位福利房-伊景园滨河苑这么麻烦，认购都都认购了好几	0	0
6	1	195511	A909237	车位捆绑违规销售	2019-08-16 14:20:26	对于伊景园滨河苑商品房，A市广铁集团违规捆绑车位销售至今，买房必须	0	0
7	1	195528	A0006953	A3区车塘河路公园尚小区物业强	2019-04-17 10:47:26	A市A3区车塘河路公园尚小区物业强制买车位，地下停车场只能买了车位	0	0
8	1	195995	A909199	关于广铁集团铁路职工定向商品	2019-08-10 18:15:16	尊敬的市政府领导，您好！我是广铁集团基层职工，我要反应的问题是关	0	0
9	1	196264	A0009508	投诉A市伊景园滨河苑捆绑车位	2019-08-07 19:52:14	A市伊景园.滨河苑现强制要求购房者捆绑购买12万车位一个，不买就取消	0	0
10	1	196475	A0003378	反映A7县星沙M9县城二期停车	2019-10-21 17:19:04	尊敬的县领导：星沙M9县城二期3838户，地下停车位1870个，车位比（1	1	3
11	1	199190	A0009508	关于A市武广新城违法捆绑销售	2019-08-01 22:32:26	武广新城为铁广集团的定向商品房，在未取得预售资格强行逼迫职工缴纳	0	0
12	1	200085	A0001042	A市市政建设开发有限公司对广	2019-08-19 11:34:11	我是广铁的一名职工，对于A市市政建设开发有限公司操作广铁职工的伊景	9	2
13	1	204960	A909192	家里本来就困难,还要捆绑买卖车	2019-08-21 18:12:20	我是广铁集团铁路职工,因家人身体欠佳,常年就医家里早已经是捉襟见肘,捉	0	0
14	1	205277	A909234	伊景园滨河苑捆绑车位销售合法	2019-08-14 09:28:31	广铁集团强制要求职工购买伊景园滨河苑楼盘时捆绑购买12万一个的车位	1	0
15	1	205982	A909168	坚决反对伊景园滨河苑强制捆绑	2019-08-03 10:03:10	我坚决反对伊景园滨河苑捆绑销售车位！原本广铁集团与市政府和开发商	2	0
16	1	207243	A909175	伊景园滨河苑强行捆绑车位销售	2019-08-23 12:16:03	您好！A市武广新城片区的伊景园滨河苑是广铁集团为职工提供定向商品房	0	0
17	1	209506	A909179	A市武广新城坑客户购房金额并	2019-08-02 16:36:23	您好！由A市广铁集团发起的定向商品房伊景园滨河苑项目存在坑害客户的	0	0
18	1	209571	A909200	伊景园滨河苑项目绑定车位出售	2019-08-28 19:32:11	广铁集团铁路职工定向商品房伊景园滨河苑项目原本是由A市政府办主持为	0	0
19	1	212323	A0002070	广铁集团要求员工购房时必须同	2019-07-11 00:00:00	尊敬的领导：您好！我是一名就职于广铁集团的普通员工，首先十分感谢	0	0
20	1	213584	A909172	投诉A市伊景园滨河苑定向限价	2019-07-28 13:09:08	投诉A市伊景园滨河苑定向限价商品房项目，广州铁路集团公司与开发商联	0	0
21	1	214975	A909182	关于房伊景园滨河苑销售若干问	2019-08-22 00:00:00	尊敬的领导，您好！感谢您在百忙之中抽出时间阅读我的信件。我是广铁	3	0
22	1	218709	A0001066	A市伊景园滨河苑捆绑销售车位	2019-08-01 22:42:21	伊景园滨河苑作为广铁集团定向商品房，取符预售证，不与广铁购房者签	1	0
23	1	218739	A909184	A市伊景园.滨河苑欺诈消费者	2019-08-24 00:00:00	A市伊景园滨河苑强行捆绑车位销售，诱骗购房者交付车位定金，还不写入	0	0
24	1	220534	A0007509	投诉武广新城伊景园滨河苑为广	2019-08-12 12:37:28	投诉：武广新城片区伊景园滨河苑为广铁集团的定向商品房，在未取得预	0	0
25	1	222209	A0001717	A市伊景园滨河苑定向限价商品	2019-08-28 10:06:03	广铁集团与伊景园滨河苑开发商沆瀣一气，严重侵害了购房职工的合法权	0	0
26	1	223247	A0004475	投诉A市伊景园滨河苑捆绑销售	2019-07-23 17:06:03	关于铁广集团铁路职工定向商品房伊景园滨河苑项目是由A市政府办牵头为	0	0
27	1	223722	A0004751	投诉A市中欣楚天熙苑不履行产	2019-03-15 21:54:03	本人于去年3月在中欣-楚天熙苑售楼部签订了购买一个产权车位的协议，	0	0
28	1	224767	A909176	伊景园滨河苑车位捆绑销售！广	2019-07-30 14:20:08	伊景园滨河苑车位捆绑销售！广铁集团做个人吧！我辛辛苦苦攒钱买个房	0	0
29	1	225479	A0004380	A市市政建设开发有限公司违规	2019-07-05 01:55:26	A市市政建设开发有限公司违规操作铁广职工住宅项目，2500多名职工盼	0	0
30	1	229731	A0004380	A市广铁管辖范围内取得购房资	2019-07-02 11:38:30	为什么现在开发商要绑定销售车位？这貌似违反规定的吧，在A市广铁管辖	0	0

图 28　第一类的部分结果

2、热度评价结果

利用所定义的热度评价指标，选出热度排名前五的热点问题：

① 对于之前已经聚好的类别，选出 K 值排名前 10 的类别，从这些类中筛选掉不合理的样本

比如：第一类中的第 10 条：“反映 A7 县星沙 M9 县城二期停车问题”，这明显不属于第一类。

② 根据所定义的热度评价指标，计算得出排名前五名的热点问题的 k 值、点赞数总和、反对数总和以及时间跨度(天)，最后计算得到热度 h，结果见表 2：

表 2　热度排名前五的热点问题的相关情况

热度排名	问题 ID	K 值	点赞数总和	反对数总和	时间跨度（天）	热度 h
1	1	56	35	3	61	1.180
2	2	58	144	2	344	0.331
3	3	41	60	3	259	0.268
4	4	40	25	2	203	0.252
5	6	33	66	4	358	0.179

3、命名实体识别

对排名前五的留言内容进行命名实体识别，找出热点问题发生的地点/人群，这里我们使用哈工大的 pyltp 工具进行命名实体识别，再对识别的结果进行拼接整理，最后得出热点问题表，结果如下图：

	A	B	C	D	E	F
1	热度排名	问题ID	热度指数	时间范围	地点/人群	问题描述
2	1	1	1.18	2019/07/02至2019/09/01	A市武广新城片区	伊景园滨河苑捆绑车位销售
3	2	2	0.331	2019/01/07至2019/12/18	A市A1，A2，A3，A5，A6等区的幼儿园	幼儿园收费不合理以及不改为普惠
4	3	3	0.268	2019/01/04至2020/09/19	A市	请求加强A市的建设力度
5	4	4	0.252	2019/07/03至2020/01/15	A市A2区丽发新城小区	搅拌厂噪音严重扰民
6	5	6	0.179	2019/01/03至2019/12/27	A市万科金域蓝湾开发商	房子质量不达标
7						

图 29　排名前五的热点问题命名实体识别的结果

3.3 问题 3 的方法与流程

3.3.1 总体流程

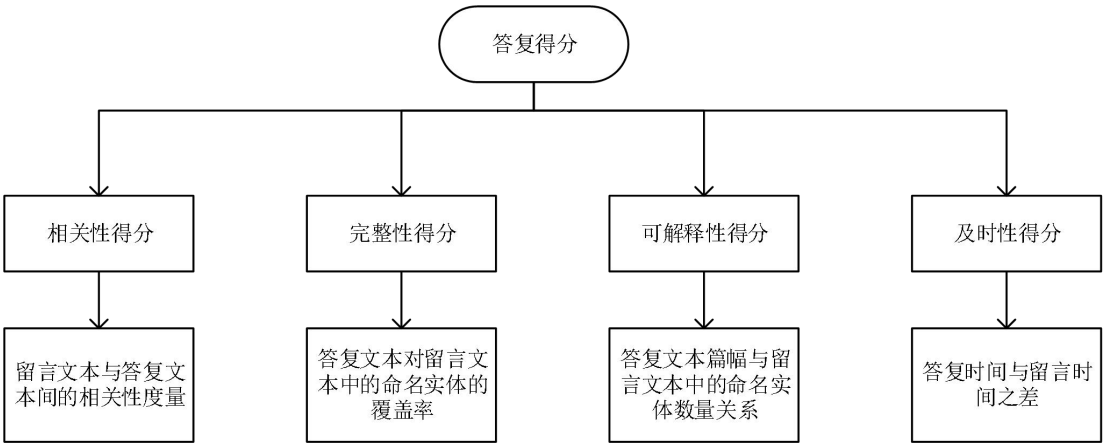

图 30　问题三的总体流程图

3.3.2 构建质量得分计算方程

质量得分计算方程如下：

$$\text{Score} = \alpha_1 S_1 + \alpha_2 S_2 + \alpha_3 S_3 + \alpha_4 S_4$$

其中，S_1，S_2，S_3，S_4 依次为相关性、完整性、可解释性、及时性得分，α_1，α_2，α_3，α_4 为其对应的权重系数，且 $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1$。

3.3.3 指标的构建

3.3.3.1 相关性

1、相关性的理解

相关性可理解为留言详情与答复意见两个文本间的相似度，因此可直接计算两个文本相似度作为相关性得分。

2、文本间的相似度的计算

（1）文本的向量化处理

① 无监督文本向量化的方法

a. 基于词向量的词袋模型的方法：平均词向量、TFIDF 加权平均词向量、SIF 加权平均词向量等。

b. 基于学习模型的方法：PV-DM 与 PV-DBOW、基于 Encoder-decoder 的 Skip-Thought Vectors、Quick-Thought vectors、n-grams embedding 等。

虽然各种方法都有其优缺点，但是由于在此题的背景下没有明确标准，无法比较各种方法在此处的优劣，同时考虑到此处留言详情与答复意见篇幅均相对较短，所以此处只使用了其中一种方法：平均词向量。

② 平均词向量

在切分出来的词都是同等重要的假设下，利用文本中的词的向量，求出它们的平均向量且以此作为文本的向量：

a. 利用 word2vec 方法将词语转变为向量

首先使用 Python 的 jieba 库中集成的 textRank 方法将文本切分成词语，而后利用 gensim 库中的 word2vec，将切分完成的文本传入。由于留言的多样性，许多低频词都属于重要词汇，故利用 skip-gram 算法，不对低频词进行过滤，将词转为词向量。

b. 利用求得的词向量，分别对每一条留言及答复中由 textRank 方法切出来的所有词的向量求平均得到文本向量。

（2）相似度计算

相似度的计算方式有多种：余弦相似度、闵可夫斯基距离、皮尔森相关系数、Jaccard 相似性系数等，且：

a. 余弦相似度用向量空间中两个向量夹角的余弦值作为衡量两个个体间差异的大小，适合 word2vec 模型向量化的数据。

b. 闵可夫斯基距离较适合 TF-IDF 向量化后的数据或者提炼出来的主题模型数据。皮尔森系数是衡量线性关联性的程度。

c. Jaccard（杰卡德）相似性系数主要用于计算符号度量或布尔值度量的样本间的相似度。

显然，本题更适合使用余弦相似度作为相似度的度量标准。利用余弦相似度计算公式：

$$\cos\theta = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2}}$$

求解每一对留言与答复间的相似度，其中 x_i， y_i 分别为两个向量的第 i 个分量上的值。求得 $\cos\theta$ 分布在[-1,1]之间。

为了得分的计算方便，此处利用 min-max 标准化方法：

$$S_1 = \frac{\cos\theta - \min}{\max - \min} = \frac{\cos\theta + 1}{2}$$

将 $\cos\theta$ 的值映射到区间[1,0]上得到 S_1，且以 S_1 作为相似性得分，部分结果见图 31：

6.781515727143402339e-01
5.174454467475456676e-01
6.0217828628133596l6e-01
6.571638925477578663e-01
8.204586464315388739e-01
5.467003245773675024e-01
5.325643075940511562e-01
6.528770157516240458e-01
7.383216164457524489e-01
7.519845898959044428e-01
6.891875437272618754e-01
6.755630031414755488e-01
6.703813078502190992e-01
6.822783161116247053e-01
5.776989119653975768e-01
6.224497711343082162e-01
5.163419790641240459e-01
5.809541320842711576e-01
6.700841028927444487e-01
5.234853257293190509e-01
5.760922368377370262e-01
7.699550780493452651e-01
7.134772413025740612e-01

图 31　相关性得分的部分结果

3.3.3.2 完整性

1、完整性的理解

完整性可理解为回复的内容对留言所提到的问题的覆盖率，可利用留言问题的主体与答复的主体的相关性去衡量，进一步理解为回复的内容对留言详情中的

命名实体的覆盖率，可由回复的内容对留言详情中每一个命名实体的覆盖率加权求和求得。

2、完整性的求解

（1）利用 jieba 库中的 TF-IDF 分词方法（相关原理上文已作解释此处不再赘述），对留言以及答复进行分词，只保留其中的名词性的词语，并利用词的 TF-IDF 值，计算他们在这些词语中所占的比例作为权重，即：

$$w_i = \frac{tf - idf_i}{\sum tf - idf_i}$$

其中 w_i 为该留言中第 i 个词语的权重，$tf\text{-}idf_i$ 为第 i 个词语的 *TF - IDF* 值。

（2）根据余弦相似度公式，计算留言中每个词语与答复中每个词语的相似度，取其最大值作为答复对该命名实体的覆盖率，即：

$$sim_i = \max\{\cos\theta(w_i, w_j), w_j \in \text{答复文本}\}。$$

其中，sim_i 为答复对留言中的第 i 个命名实体的覆盖率，$\cos\theta(w_i, w_j)$ 为留言中第 i 个命名实体与答复中第 j 个词的相似度。

（3）由留言中每个命名实体的加权和可得该留言的答复的覆盖率，即完整性：

$$S_2 = \sum w_i \cdot sim_i$$

部分结果见图 32：

5.780315713598116467e-01
1.958435821528816401e-01
3.823074460782018846e-01
5.310076333630692202e-01
4.138621423233564411e-01
1.364271487856666409e-01
3.822862073058279075e-01
3.727349160390603378e-01
2.509755680173518799e-01
4.029353574943512206e-01
3.379294046303767396e-01
6.114886920272725179e-01
2.848259450290871464e-01
1.292817969809721967e-01
5.015229711737332341e-01
4.704367407062740036e-01
1.692012866055618014e-01
1.650203100270015988e-01
3.659131265951792167e-01
6.219928008484552212e-02
1.799662597639524186e-01

图 32　完整性得分的部分结果

3.3.3.3 可解释性

1、可解释性的理解

可解释性可以理解为可执行性，即答复是否有敷衍般的答复，是否有明确的解决步骤、流程。

2、可解释性的计算

（1）在留言的命名实体数量一定时，篇幅长的答复中往往可解释性较强，且答复的可解释性的增长速度应该是随着篇幅长度的增加而降低的（解决方案数量从0到1的答复可解释性的增加要大于从1到2，与经济学中的边际效益递减现象相似），则构造以下公式以衡量答复的可解释性：

$$S'_3 = \arctan(\frac{L}{n})$$

其中，L为答复文本进行分词并去停用词后剩下词的数量，n为留言的命名实体数量。

（2）由于S'_3的值域为$[0,\frac{\pi}{2}]$，为了计算方便，使用min-max标准化方法将S'_3的值映射到区间[0,1)上，即：

$$S_3 = \frac{2S'_3}{\pi}$$

S_3即为答复的可解释性，部分结果见图33：

6.771710655658095268e-01
9.740299348986242167e-01
8.013478156017629361e-01
8.820038302649430006e-01
9.028204193049421677e-01
6.928224959584591280e-01
5.501554273111252114e-01
8.715541022864693455e-01
7.951672353008665262e-01
5.235677377406828814e-01
8.001407260882059669e-01
9.734895107734576891e-01
4.895645450473825044e-01
2.951672353008665262e-01
9.070457097329713836e-01
8.373123939862959775e-01
7.878500888834932825e-01
7.577621168183131806e-01
8.440417392452614909e-01
7.422378831816868194e-01

图33　可解释性得分的部分结果

3.3.3.4 及时性

1、及时性的理解

顾名思义，及时性即回复留言耗费时间长短。

2、及时性的计算

假设当人们在不超过一定等待时长时，对于等待时长的厌恶度服从均匀分布，设答复时间差（天）为$t = t_2 - t_1$，由下列公式计算每个答复的及时性得分S_4：

$$S_4 = \begin{cases} \dfrac{90 - t}{90}, t \le 90 \\ 0, t > 90 \end{cases}$$

及时性得分S_4部分结果见图 34：

8.3333333333333703e-01
8.4444444444444420e-01
8.4444444444444420e-01
8.4444444444444420e-01
8.3333333333333703e-01
6.5555555555555580e-01
5.5555555555555802e-01
6.8888888888888840e-01
8.2222222222221877e-01
8.2222222222221877e-01
2.2222222222222099e-01
6.6666666666666297e-01
8.2222222222221877e-01
9.4444444444444198e-01
8.1111111111111160e-01
2.4444444444444364e-01
9.2222222222222765e-01
8.0000000000000444e-01
8.5555555555555136e-01
9.0000000000000222e-01

图 34　及时性得分的部分结果

3.3.4 部分结果展示

利用所建立的质量得分计算方程，求取回复的质量得分，部分结果如下图：

	Score
0	0.691672
1	0.632941
2	0.657569
3	0.728655
4	0.742619
5	0.507876
6	0.50514
7	0.646514
8	0.651672
9	0.625177
10	0.51237
11	0.731802
12	0.566749
13	0.512793
14	0.699345
15	0.543661
16	0.598904
17	0.575934
18	0.683899

图 35　回复的质量得分的部分结果

4、模型的展望和总结

1、模型优点

（1）对群众留言详情进行文本挖掘，构建了基于 word2vec 的 TextCNN 卷积神经网络文本分类模型，较好地实现对留言的一级标签划分，能够极大地提高相关部门工作人员的工作效率，同时减少工作强度。

（2）构建基于 TF-IDF 计算文本相似度的 single-pass 聚类模型，对群众反映的同一个问题进行聚类，快速地定位热点问题。

2、模型展望

（1）对于问题一的各个类别数据不均衡

① 利用 focal loss 损失函数能够减轻数据不均衡带来的影响，但也仅限于减轻，并不能完全解决，从改进后打印出的混淆矩阵可发现该情况。因此最好能够平衡各个类别的数据，但是一般的采样并不适合文本数据。

② 要实现文本类别均衡，或许可以分析样本量少的类别，通过文本句法依赖，文本词性标记分析词的相关属性，同时采用同义词替换的方式生成新的样本，则不但可以平衡各类别的数据，而且还能提升模型分类的准确率。

（2）对于问题二的聚类模型

① 由于 single-pass 是单遍聚类，因此样本的输入顺序会影响聚类质量，

比如一个新样本，当它与某一个类的相似度达到阈值，就会归于该类，但这个样本可能与之后的类更相似。这个是 single-pass 的特性，没有什么好的方法可以改变。

② 但是可以通过对文本特征进行更加细致的关键词抽取，使用 Jaccard 系数作为文本的相似性度量，以此来提高聚类质量。

5、参考文献

[1] Kim Y . Convolutional Neural Networks for Sentence Classification[J]. Eprint Arxiv, 2014.

[2] Lin, T. Y. , Goyal, P. , Girshick, R. , He, K. , & Dollár, Piotr. (2017). Focal loss for dense object detection. IEEE Transactions on Pattern Analysis & Machine Intelligence, PP(99), 2999-3007.

[3]党燕,许志伟,刘利民,王宇,赵思远.基于 Single-Pass 算法的网络舆情文本增量聚类算法研究[J]. 内蒙古工业大学学报(自然科学版),2017,36(05):364-372.

[4]刘春磊,武佳琪,檀亚宁.基于 TextCNN 的用户评论情感极性判别[J].电子世界,2019(03):48+50.