C 题：“智慧政务”中的文本挖掘应用

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。为了能够更好的收集民意，汇聚民心，切实解决人民群众关心的问题，推进国家治理能力和治理体系现代化，需要利用自然语言处理和文本挖掘的方法解决以下问题：

解决群众留言的分类问题，建立了基于统计的分词模型，利用结巴分词，建立在针对新闻文本的停用词表上的分词模型，对分词结果建立基于 TextRank 算法的关键词抽取模型，提取出可以用于文本分类的关键词；对提取出的大量关键词，使用支持向量机（SVM）算法，并使用 rbf kernal 高斯核函数进行分类，得到了一个准确度较高的关于留言内容的一级标签分类模型。

解决热点问题挖掘问题，关于热点问题，沿用第一问中的方法，建立基于统计针对新闻文本停用词基础上分词模型，基于 TF-IDF 算法，对热点问题中的关键词提取，基于编辑文本算法，利用 FuzzyWuzzy 工具包对文本之间的相似性进行匹配，并统计词频，将词频并结合点赞数作为热度评价指标，给出了评价结果以及排名前五的热点问题，并以表格形式存于附件中。

解决了答复意见评价问题，基于编辑距离算法，调用 Python 中 FuzzyWuzzy 工具包中的 Fuzz 库，对对留言主题和留言详情分别与答复意见做相似度匹配，得到相关性的评分；调用 time 库并结合艾宾浩斯记忆遗忘曲线给出及时性评价；利用结巴分词，计算答复中对留言关键词的包含权重百分数，衡量完整性。结合相关性、及时性和完整性，给出了答复意见质量的综合性评价方案。

关键词：文本挖掘；TF-IDF 算法；TextRank 算法；SVM 算法；编辑文本算法

目录

1. 问题重述

1.1 问题背景

近年来，随着科技的进步与社会的进步，越来越多的人们参与到和谐美丽的社会建设中来，加之我国一直在推进国家治理能力和治理体系现代化，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。数据量的增大虽然有利于政府针对民生更好的治理，但给曾经依靠人工来进行留言划分和热点整理的相关部门的工作难度加大，工作效率降低。

近年来，随着大数据、云计算、人工智能等互联网高科技术的发展，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，对提升政府的管理水平和施政效率具有极大的推动作用。随着数据量的激增，人工处理数据效率低下等影响治理能力和治理体系现代化的进程，需要我们通过收集自互联网公开来源的群众问政留言记录，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建立合理模型完成群众问题分类、热点问题挖掘、答复意见评价等问题。

问题的分类标准，是工作人员在处理网络问政平台的群众留言时，首先按照一定的划分体系，分 15 个一级标签对留言进行分类，以便后续将群众留言分派至相应的职能部门处理。

热点问题，是某一时段内群众集中反映的某一问题，及时发现热点问题，有助于相关部门进行有针对性地处理，提升服务效率。

1.2 提出问题

需要我们通过收集自互联网公开来源的群众问政留言记录，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解决下面的问题：

问题一：群众留言分类

目前，大部分网络问政平台的问题及数据处理还是依靠人工根据经验处理，存在工作量大、效率低，且差错率高等问题。要求我们根据（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。

问题二：热点问题挖掘

需要我们根据（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，给出排名前 5 的热点问题，并保存为文件“热点问题表.xls”。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”。

问题三：答复意见的评价

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，并尝试实现。

2. 问题分析

2.1 问题一的分析

问题一中，附录 2 给出的数据量庞大，要从复杂且量大的问题数据中中找到有价值的信息，有如下几个步骤。考虑到汉语的特殊性，首先要进行分词，这是文本数据挖掘的基本途径，二、对分词后的文本进行去除停用词，消去停用词后的数据是后续分析的基础。三、要根据分词后的数据，找出文本中的关键词、关键信息，结合实际情况，进行进一步的研究。四、要将处理后的数据进行分类，对不同的问题进行分类分成题目中要求的 15 类一级标签数据。

2.2 问题二的分析

问题二要求考虑一段时间内的热点问题，所谓热点问题就是人群会集中反映，并且反应后人群高度关注，我们考虑在第一问分词模型的基础上，将词频和点赞数综合考虑作为热度评价指标，统计得到排名前五的热点问题。

2.3 问题三的分析

问题三中要求给出答复的评价方案，对于现实生活中的一般问答句，答句中总会包含有几个问句中的词语或者子句，从这个特征出发，可以考虑对留言主题和留言详情分别与答复意见做相似度匹配，相似度高则相关性好。对于及时性，我们考虑计算答复-留言天数差，天数差小则相关性好对现实生活中的问答句，一般问句中都含有提问者所提出的“关键字/词”，而回答中如果对这些关键词高度包含（或者相似）时，则可认为该回答具有完整性。综合考虑上述指标，可得到评价标准。

3. 模型假设

为了简化分析，本题建立模型时做以下假设：

1、分词时，使用结巴分词有很小的几率会把一个关键词分开

2、对现实生活中的留言中，留言中都含有留言用户针对某一问题所提出的“关键字/词”，

3、对及时性进行计算，而且这里只计算答复-留言天数差，不考虑时分秒时间差。

4、对于现实生活中的一般问题答复，答复中总会包含有几个问句中的词语或者子句

5、结合艾宾浩斯记忆遗忘曲线假设当天数差<=7 天时，为及时的答复，>7 天而<15 天时，为较及时的答复，积 1 分，当>15 天时为不及时的答复，不积分。

4. 符号约定

符号	代表含义
$A_{11}A_{12}\dots A_{1n_1}$	语句 S 的分词选项
$P(A_{i1}, A_{i2}, \dots, A_{in_i})$	每个分词出现的概率
wi	表示任意一个关键词

freq(w1,w2)	任意两词 w1,w2 在语料库中相邻一起出现的次数
freq(w1)，freq(w2)	分别表示 w1,w2 在语料库中出现的统计次数
TF-IDF	
TF	词频表示关键词 w 在文档 Di 中出现的频率
IDF	逆文档频率反映关键词的普遍程度
count(w)	为关键词 w 的出现次数
|Di|	为文档 Di 中所有词的数量
g(x)	分类决策函数
K(x,z)	高斯核函数

5. 问题所用的算法模型

5.1 基于统计的分词模型

为了能在短时间内处理大量文字信息，进行文本挖掘，首先就是要完成中文的分词。中文分词(Chinese Word Segmentation)指的是将一个汉字序列切分成一个一个单独的词。分词就是将连续的字序列按照一定的规范重新组合成词序列的过程。现有的分词方法可分为三大类：基于字符串匹配的分词方法、基于理解的分词方法和基于统计的分词方法。

基于字符串匹配的分词方法又称机械分词方法，它是按照最大或最短匹配等方法，将待分析的汉字串与一个“充分大的”机器词典中的词条进行配，若在词典中找到某个字符串，则匹配成功（识别出一个词）。基于统计的分词方法是在给定大量已经分词的文本的前提下，利用统计机器学习模型学习词语切分的规律，从而实现对未知文本的切分。随着大规模语料库的建立，统计机器学习方法的研究和发展，基于统计的中文分词方法渐渐成为了主流方法，其基本原理如下：

如果有一个句子 S,它有 m 种分词选项如下：

$$A_{11}A_{12}\dots A_{1n_1};\ A_{21}A_{22}\dots A_{2n_2};\ \dots\dots;\ A_{m1}A_{m2}\dots A_{mn_m}$$

其中下标 n_i 代表第 i 种分词的词个数。如果我们从中选择了最优的第 r 种分词方法，那么这种分词方法对应的统计分布概率应该最大，即：

$$r = \underbrace{argmax}_{i} P(A_{i1}, A_{i2}, \dots, A_{in_i})$$

在 NLP 中，为了简化计算，我们通常使用马尔科夫假设，即每一个分词出现的概率仅仅和前一个分词有关，即：

$$P(A_{i1}, A_{i2}, \dots, A_{in_i}) = P(A_{i1})P(A_{i2}|A_{i1})P(A_{i3}|A_{i2})\dots P(A_{in_i}|A_{i(n_i-1)})$$

而通过我们的标准语料库，我们可以近似的计算出所有的分词之间的二元条件概率，比如任意两个词 w1,w2，它们的条件概率分布可以近似的表示为：

$$P(\omega_2|\omega_1) = \frac{P(\omega_1,\omega_2)}{P(\omega_1)} \approx \frac{freg(\omega_1,\omega_2)}{freg(\omega_1)}$$

$$P(\omega_1|\omega_2) = \frac{P(\omega_2,\omega_1)}{P(\omega_2)} \approx \frac{freg(\omega_1,\omega_2)}{freg(\omega_2)}$$

其中 freq(w1,w2)表示 w1,w2 在语料库中相邻一起出现的次数，而其中 freq(w1),freq(w2)分别表示 w1,w2 在语料库中出现的统计次数，由此可以计算出每种分词方法的概率，并将概率最大的分词方式作为结果返回。

在实际的应用中，基于统计的分词系统都需要使用分词词典来进行字符串匹配分词，同时使用统计方法识别一些新词，即将字符串频率统计和字符串匹配结合起来，因此为了能更快，更准确的实现分词，我们在这里将使用目前主流的中文分词工具，“jieba 分词”[2]来进行具体操作

“jieba 分词”过程中主要涉及如下几种算法：

（1）基于前缀词典实现高效的词图扫描，生成句子中汉字所有可能成词情况所构成的有向无环图 (DAG)；

（2）采用了动态规划查找最大概率路径, 找出基于词频的最大切分组合；

（3）对于未登录词，采用了基于汉字成词能力的 HMM 模型，采用 Viterbi 算法进行计算。

由于分词后，原文本会产各种词性的词，例如动词、介词、语气助词、名词等等，然而分词结果中我们需要的关键词并不包括一些无用的词，所以需要在使用“jieba”分词后，消去无用的停用词，在这里我们使用哈工大的停用词表，将我们读入的分过词的文本与停用词文本，进行对比，去除与停用词重合的部分，得到一个已经经过初步预处理的数据模型。

5.2 基于 TF-IDF 算法和 TextRank 算法的关键词提取模型

为了能顺利提取出文本中的关键词，我们需要确定一个可量化的指标，来完成群众留言问题详情中关键词的重要性排序，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。我们可以通过 TF-IDF 加权的各种形式，作为与群众留言详情与分级指标之间相关程度的度量或评级。[3]

TF-IDF 是信息检索领域非常重要的搜索词重要性度量；用以衡量一个关键词 w 对于文档所能提供的信息。词频（TF）表示关键词 w 在文档 Di 中出现的频率：

$$TF_{\omega,D_i} = \frac{count(\omega)}{|D_i|}$$

其中，count(w)为关键词 w 的出现次数，|Di|为文档 Di 中所有词的数量。逆文档频率（IDF）反映关键词的普遍程度，当一个词出现频率越高，其 IDF 值越低；反之，则 IDF 值越高。IDF 定义如下：

$$IDF_{\omega} = \log\frac{N}{\sum_{i=1}^{N} I(\omega, D_i)}$$

其中，N 为所有的文档总数，I(w,Di)表示文档 Di 是否包含关键词，若包含则为 1，若不包含则为 0。若词 w 在所有文档中均未出现，则 IDF 公式中的分母为 0；因此需要对 IDF 做平滑：

$$IDF_{\omega} = \log\frac{N}{1 + \sum_{i=1}^{N} I(\omega, D_i)}$$

关键词 w 在文档 Di 的 TF-IDF 值可利用下面公式计算：

$$TF - IDF_{\omega,D_i} = TF_{\omega,D_i} * IDF_{\omega}$$

由此可以看出，字词的重要性随着它在文件中出现的次数成正比增加，说明他

在区分文本属性的能力越强，但同时会随着它在语料库中出现的频率成反比下降，说明区分文本属性越低。利用 TF-IDF 可以轻松的过滤掉常见词语，保留有区分性的关键词。

由于 TF-IDF 会对分词结果有一定的依赖，如果一个关键词在分词时被切分成了两个词，那么在做关键词提取时无法将两个词黏合在一起，从而影响最终的分类结果，而如果使用 TextRank 算法做关键词提取的话，则有一定的机率将这两个被分开的关键词重新连接在一起，因此我们在 TF-IDF 算法的基础上，使用 TextRank 做优化，以提高关键词提取的准确率。

TextRank 是由 PageRank 改进而来，其算法公式如下：

$$PR(V_i) = (1-d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} PR(V_j)$$

其中 V_i 表示某个关键词，V_j 表示连接到这个关键词的其他关键词，$In(V_i)$表示所有连接到关键词的集合，Out（V_j）指关键词连接出去的集合，w_{ji} 为边的权重，也就是句子之间的相似性。（1-d）是保证马尔科夫链平稳分布的平滑项。

而针对本研究问题，我们可以将群众留言记录先做分词处理后，将所有关键词构件关键词图，选定最多可出现的关键词数值后，利用上式迭代计算各关键词权重，直至收敛，将权重倒叙排列即为候选的用于分类的关键词。在这个过程中如果在原文本中标记出相邻词组，则可以顺利提取多词关键词。

5.3 基于支持向量机(SVM)算法的关键词分类模型

完成从留言数据关键词的提取后，我们得到了大量备选关键词，而要实现问题中针对留言内容的一级标签的分类模型，需要有一个算法能对数据进行判断分析。SVM 算法可以通过超平面划分，将关键词分成我们需要的几组，用于判断留言的所属。

支持向量机（support vector machines，SVM）是一种二类分类模型。[4]它的基本模型是定义在特征空间上的间隔最大的线性分类器，如果不考虑空间维数，那么在空间中总存在一个可以将数据划分开的线性函数，也就是超平面，可以用下面的式子表示：

$$g(x) = w^T x + b = 0$$

在理论上还是计算上，在高维空间中找到超平面是完全可行的，但光是分开是不够的，SVM 算法的核心思想是尽最大努力使分开的两个类别有最大间隔，这样才使得分隔具有更高的可信度。而且对于未知的新样本才有很好的分类预测能力，为了有效的描述这个间隔，SVM 算法的办法是：让离分隔面最近的数据点具有最大的距离，算法原理如下：

$$\min_{w,b} \quad \frac{1}{2}\|w\|^2$$

$$s.t. \quad y_i(w.x_i + b) - 1 \geq 0 \quad i = 1,2,\dots,N$$

其中将所有关键词文本打散作为训练数据集

$$T = (x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$$

其中，$x_i \in \mathbb{R}^n, y_i \in +1, -1, i = 1,2,\dots,N$，$x_i$ 为第 i 个特征向量，y_i 类标记，当它等于+1 时为正例；为-1 时为负例。含有不等式约束的凸二次规划问题，可以对其使

用拉格朗日乘子法得到其对偶问题,其中 a_i 为拉格朗日算子：

$$max.W(a) = \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i=1,j=1}^{n} a_i a_j y_i y_j \, x_i^T x_j$$

$$subject \quad to \quad a_i \geq 0, \sum_{i=1}^{n} a_i y_i = 0$$

由此可分离超平面，取得分类决策函数：

$$g(x) = \sum_{i=1}^{N} a_i y_i (x_i, x) + b$$

前述方法对线性不可分的样本集无能为力，但是，如果拿到低维数据直接映射到高维的话，维度的数目会呈现爆炸性增长。所以这里需要引入核函数（kernal function）。核函数的思想是寻找一个函数，这个函数使得在低维空间中进行计算的结果和映射到高维空间中计算内积的结果相同。这样就避开直接在高维空间中进行计算，而最后的结果却是等价的。而在本题中应用的是高斯核函数如下：

$$K(x, z) = exp\left(\frac{\|x - z\|^2}{2\sigma^2}\right)$$

现在已经完成了对这个问题的建模过程。当要对一个数据点分类是，只需要把待分类的数据点带入 g(x)中，把结果和正负号对比，便可以利用分类决策函数对新的样本进行分类预测。通过判断留言中的关键词可以将用户留言按一级标签分为 15 类。

5.4 基于编辑距离算法的有关答复相关性的评价

问题三提出了对答复相关性的评价，对于现实生活中的一般问答句，答句中总会包含有几个问句中的词语或者子句，从这个特征出发，而答复内容与留言主题和留言详情若有较好的相关性，则关键词构成的字符串一定有较大的相似性，评价相似性高低是需要将答复内容的遍历留言内容，得到相似度匹配。这里我们使用编辑距离算法（Levenshtein Distance 算法），它是指两个字符串之间，由一个转成另一个的过程中需要最少编辑的次数，许可编辑的操作包括字符串中单个字符的替换、插入、删除。一般来说，如果两个字符串编辑距离越小，两个字符串之间相似程度越高，也就是相关性越好。

由于在求解过程中，距离编辑算法的不是本项目考虑的重点，在这里将调用简单易用的模糊字符串工具包（FuzzyWuzzy)来实现计算两个字符串序列之间的相似性。[5]

现对使用 FuzzyWuzzy 工具包中的 Fuzz 库的部分函数做一定的说明：

全匹配（fuzz.ratio），对位置敏感，可以比较两个长度相同字符串的相似度得分，如果遇到长度不同的字符串则返回得分最高字符串的匹配值；

非完全匹配（Partial Ratio），对位置比较敏感，效果上和全匹配类似，相似度得分会稍高一点

忽略顺序匹配（Token Sort Ratio），先以空格为分隔符，小写化所有字母，无视空格外的其它标点符号，再排序，最后匹配。因为在匹配前打乱了顺序，相似度

衡量水平较好。

去重子集匹配（Token Set Ratio），和前者相似，先 token 化，再排序，最后匹配。而且在排序的同时还会把字符串分为共有部分和多余部分，相似度衡量水平较好。

利用 FuzzyWuzzy 工具包中的各种 ratio 函数都是先通过 process 库中的函数返回需要进行模糊匹配的字符串经过处理之后的结果字符串，再对结果字符串调用全匹配或者非全匹配，得到相似度，通过相似度大小的比较衡量出答复意见与留言问题的相关性程度。

6. 问题的求解

6.1 问题一的求解

对于评论的分类，由于分类算法并不能很好地识别文字信息尤其是中文文字信息，所以我们需要将文本向量化，同时为保证处理的效率我们需要提取关键词并使用权重靠前的关键词作为文本特征参与到分类模型的拟合当中去。

我们先使用了结巴分词内的 TextRank 分词方法，建立在我们针对新闻文本的停用词表上的分词模型，对每一条评论的全文，进行提取关键词的操作。部分关键词提取结果如下：

…	……
84	押金 住房 装修 母亲 租金 工作人员 房屋 房子 交租金 中心
85	物业 业主 垃圾站 不用 垃圾 只用 露天 建好 发出 沟通
86	先锋 办事处 街道 南路 书记 道路 通车 城市 是从 望能
87	小区 车库 居住 民工 群租 住房 弹棉花 开发 农贸市场 门面
88	装修 修补 开发商 房子 楚江 新区 只能 核实 耽误 保温层
89	小区 物业公司 香海 法律 业委会 管理 业主 物业 公司 信任
90	政府 房屋 开发商 解决 渗水 外墙面 街道 质量 生活 卫生间
91	行道树 枝桠 剪枝 碰到 树枝 道路 枝应 修剪 地面 希望
92	行道树 南路 希望 改造 碰到 树枝 道路 提质 枝应 修剪
93	业主 施工 楚江 开发商 投资 物业公司 违法 江湾 私卖 利益
94	混凝土 业主 情况 质量 宜华 交房 不肯 施工 强度 资料
95	房子 借问 回家 借居 在外 危房 算不算 生计 留给 照看
…	……

（关键词提取结果）

我们设置了 Top 20 权重的关键词作为每一条评论的特征词，并使用其权重作为在矩阵中的值。经过处理，我们得到以行为评论列表，列为关键词的特征向量化矩阵（VSM），见附件。下面为部分关键词的特征向量化矩阵：

…	…	…	…	…	…	…	…	…	…
⋮	1	0.969204	0.601724	0.548887	0.456746	0.45618	0.241192	0.384345	⋮
⋮				0.748017					⋮
⋮				0.748017					⋮
⋮		0.729085		1					⋮
⋮									⋮
⋮									⋮
⋮		0.43331		0.919102					⋮
⋮		0.24825		1	0.156114				⋮
⋮		0.24825		1	0.156114				⋮
⋮		0.954389		0.651062					⋮
⋮									⋮
⋮	0.559973	1					1.440027		⋮
…	…	…	…	…	…	…	…	…	…

针对 VSM，由于其维数远远大于样本数的特性，支持向量机无疑是很好的办法。我们按训练：测试=1:3 的比例进行训练和验证，核函数我们选择了高斯核函数，对此类数据较为敏感。调整参数以后我们得到较高的准确度，建立了关于留言内容的一级标签分类模型。准确度如下：

```
[Running] python
0.995223621363439
```

6.2 问题二的求解

对于热点问题的挖掘，考虑到热点问题是一段时间内群众集中反映的问题，因此全部留言问题的文本中，热点问题在全部问题中一定具有较多的相似性，计算时也正是利用这一特点。

首先沿用问题一中的方法，建立基于统计的分词模型，并利用新闻文本的停用词表消去停用词，对分词结果进行提取关键词时，则使用 TF-IDF 算法找到常用的关键词。一定时间内，留言问题的关键词可以很好的反映特定地点或特定人群的留言问题。

调用 FuzzyWuzzy 工具包中 fuzz 库，对留言问题中的关键词，利用模糊搜索的方法，对彼此之间的相似度进行计算，并匹配，彼此之间相似度高的归为一类，找出具有相似数量最多的五个集合，按照相似数量排列。

一定时间内群众的关心程度也可以体现热点问题，因此可以用点赞数作为衡量标准，利用统计学方法，将点赞数化为一定的权重比，主要考虑相似度，结合点赞数的权重，给出最终的排名前五的热点问题，最后根据关键字定出每一组热点问题的标题。全部结果保存在附件中。

6.3 问题三的求解

6.3.1 相关性求解

对于现实生活中的一般问答句，答句中总会包含有几个问句中的词语或者子句，从这个特征出发，调用 python 中的 fuzzywuzzy 库中的 fuzz 库，对留言主题和留言详情分别与答复意见做相似度匹配。

由于 fuzz 库中的 ratio()，partial_ratio()，token_set_ratio()，

partial_token_set_ratio()， token_sort_ratio()，partial_token_set_ratio()六个函数极为相似，且在本题情况下，对 token_set_ratio()函数和 token_sort_ratio()函数以及 partial_token_set_ratio()函数 partial_token_set_ratio()函数这两对函数的返回结果是一样的，故不考虑 token_sort_ratio()，partial_token_set_ratio()两个函数。

当我们得到四个函数的返回值后，首先两两比较，对于 ratio()，partial_ratio()函数，若返回值差不小于 30，则只取较大者，对于 token_set_ratio()，partial_token_set_ratio()两个函数，若返回值差不小于 50，则只取较大者，然后对于最后保留的几个返回值通过加权平均计算相似度评分，其中 ratio()，partial_ratio()和 token_set_ratio()，partial_token_set_ratio()两对函数的权重分别为 0.3 和 0.7，每对函数平分权重。需要注意的是，大多数情况下都是带有 partial 的函数的返回值较大。

最后将得到的相似度评分储存在一个列表中等待输出保存。

6.3.2 及时性求解

对于及时性，调用 python 中的 time 库和 dateutil.parser 库中的 parse，对及时性进行计算，而且这里 只计算答复-留言天数差，不考虑时分秒时间差。

首先对附件四中的每个留言时间和答复时间调用 str()函数，使其均转化为字符串类型数据，再通过 parse()函数将每个留言时间和答复时间转化为 datetime.datetime 类型数据，最后通过(date1-date2).days 函数（date2 为留言时间，date1 为对该留言的答复时间）返回 date1 和 date2 的天数差。

最后将得到的天数差储存在一个列表中等待输出保存。

6.3.3 完整性求解

对现实生活中的问答句，一般问句中都含有提问者所提出的“关键字/词”，而回答中如果对这些关键词高度包含（或者相似）时，则可认为该回答具有完整性。

调用 python 中的 jiaba 库中的 jieba.analyse，对留言详情进行关键字提取，并将提取出来的关键字以及其对应的权重储存在 excel 文件中，这里需要注意的是，在输出文件中 1263，1585 两行中由两个空白，即缺失 1 个关键词和 1 个权重，为了数据处理方便，故人为补上关键词：No Keyword 和权重 0，然后重新从文件中读取关键词和权重数据，但这里对权重数据进行处理：由于有 5 个权重，对每个权重除以五个权重之和并乘 100，保留 3 位小数，称处理后的数据位权重百分数，以便之后度量完整性。

由于先前对于完整性的描述，故调用 python 中的 fuzzywuzzy 库中的 fuzz 库中的 partial_ratio()函数，传入的两个实参分别为关键词和答复意见，如果返回值大于 80，则认为答复意见中对该关键词进行了针对性的回答，并取其权重百分数放入一个列表中，对于 1 条答复意见进行这样的操作 5 次（因为有 5 个关键词），并取 5 个权重百分数之和。将所有的权重百分数之和储存在一个列表中等待保存输出。

当得到了相似度评分、天数差、权重百分数之和后，将数据打包输出，得到一个新的 excel 文件，见附件，其内容即为经过处理后的用于评价答复质量的数据。下面给出部分评价答复质量部分结果：

留言用户	正文-答复相似度	留言-答复时间间隔（单位：天）	权重得分	答复意见得分	答复意见质量
UU008706	22.25	18	11.796	1	一般答复意见
UU008201	14.45	13	12.945	1	一般答复意见
UU0081681	11.6	9	0	1	一般答复意见
UU0081681	15.4	9	15.56	1	一般答复意见
UU0081500	74.2	14	72.738	5	优质答复意见
UU0081057	17.5	22	28.051	0	无效答复意见
UU008162	14.4	36	0	0	无效答复意见
UU0081604	20.45	13	12.789	2	一般答复意见
UU008694	74.5	14	7.788	3	良好答复意见
UU008765	74.35	14	16.471	3	良好答复意见
UU0082119	73.9	20	56.431	3	良好答复意见
UU008233	15	15	34.021	1	一般答复意见
UU0082278	14.55	93	51.863	1	一般答复意见
UU00840	15.7	24	70.159	2	一般答复意见
UU008355	75.25	94	60.313	4	良好答复意见

部分答复质量表

6.3.4 评价标准的描述

1、相关性：

当相似度评分高于 50，积 2 分，不低于 20 而不高于 50，积 1 分，否则不积分。

2、及时性：

由艾宾浩斯记忆遗忘曲线可以知道，在 7 天后人们对于一些内容记忆量将约为刚接触时的为 25%。

故假设当天数差<=7 天时，为及时的答复，积 2 分，>7 天而<15 天时，为较及时的答复，积 1 分，当>15 天时为不及时的答复，不积分。

3、完整性：

当权重百分数之和大于 60 时，积 2 分，不大于 60 且不小于 40 时，积 1 分，否则不积分。

对于积分值在[5,6]的答复意见为优质答复意见，积分值在(2,5)的答复意见为良好答复意见，积分值[1,2]的答复意见为一般答复意见，积分值为 0 的答复意见为无效答复意见

7. 模型的优缺点

7.1 模型的优点

1、问题一中考虑到分词时可能被错误划分，再 TF-IDF 算法的基础上结合使

用 TextRank 算法以及自定义一些关键词进行关键词的提取，一定程度上提高了准确率。

2、基于 SVM 算法中使用高斯核函数可以向高维空间进行映射，解决非线性的分类，简化分析。

7.2 模型的缺点

1、TextRank 算法与 TF-IDF 算法均严重依赖于分词结果，因此是否添加标注关键词进自定义词典，将会造成准确率不同。TextRank 虽然考虑到了词之间的关系，但是仍然倾向于将频繁词作为关键词。可能影响准确率

2、基于 SVM 算法分类模型需要相当数量的训练集才能得到有效的分类器。这都对数据的准备提出了较高的要求。

3、fuzz 这几个函数评价相关性都受相同部分影响较大，需要人工判断字符串自身的情况，再选择相应的函数匹配。如果选错了，会得到和预想差别很大的结果。

8. 参考文献

[1]刘建平.文本挖掘的分词原理[EB/OL].https://www.cnblogs.com/pinard/p/6677078.html,2017-04-07.

[2]fxsjy.结巴中文分词[EB/OL].http://github.com/fxsjy/jieba,2016-05-24.

[3]zrc199021.TF-IDF 原理及使用[EB/OL].https://blog.csdn.net/zrc199021/article/details/53728499,2016-12-18.

[4] [1]野风.支持向量机（SVM）——原理篇[EB/OL].https://zhuanlan.zhihu.com/p/31886934,2019-11-07.

[5]爱吃大鱼 de 猫.FuzzyWuzzy：简单易用的字符串模糊匹配工具[EB/OL].https://www.jianshu.com/p/ed22a82b45d1,2018-12-24.