“智慧政务”中的文本挖掘应用

摘要

近年来，向网络问政平台反映民意的留言日益增加，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，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。本文构建了一个智慧政务系统，能进行留言分类和热点问题整理，并能对留言答复进行评价。

数据预处理阶段是文本挖掘问题的必经之路，本文对数据源文件进行数据清洗、分词、去停用词，为后来的模型建立打下基础。为了解决留言分类问题，本文采用 word2vec 构建词向量，并将其放入深度学习的 LSTM 模型中进行训练并对留言进行分类，用 F1 分数对其进行评价，对测试集的测试精度为 0.8，验证了模型的有效性。

本文采用 TF-IDF 算法来对数据预处理后的留言构建词向量，并赋予其权重，再建立向量空间模型，运用余弦相似度算法得出其相似度；运用聚类分析中 silhouette Coefficient(s)和 Davies-Bouldin Index (DBI)两个指标来获得分类最佳的余弦相似度，将大于此余弦相似度的留言视为相似，统计出 TOP5 热点留言问题。

本文从相关性、完整性以及及时性三个角度出发构建出一套对留言答复的评价体系。相关性和完整性运用 LDA 模型和余弦相似度相结合计算其相关程度，及时性则是评价留言答复和留言的时间差，本文构建了一整套体系，对相关性和完整性以及及时性分别打分，并得出它们的综合得分。

关键词：智慧政务,word2vec,LSTM,TF-IDF,余弦相似度,聚类分析

Abstract

In recent years, the increasing number of messages that reflect public opinion on the Internet questioning platform has brought great challenges to the work of relevant departments that used to manually divide messages and organize hotspots. The establishment of smart government systems based on natural language processing technology It is a new trend of innovation and development of social governance, and it has a great role in promoting the government's management level and governance efficiency. This article builds a smart government system that can classify messages and sort out hot issues, and can evaluate the responses to messages.

The data preprocessing stage is the only way for the text mining problem. In this paper, data cleansing, word segmentation, and stop word removal are performed on the data source files to lay the foundation for the subsequent model establishment. In order to solve the problem of message classification, this paper uses word2vec to construct a word vector, and put it into the LSTM model of deep learning to train and classify the message, evaluate it with F1 score, and the test accuracy of the test set is 0.8 The validity of the model.

In this paper, the TF-IDF algorithm is used to construct word vectors for the data pre-processed messages and give them weights. Then, a vector space model is established, and the similarity is obtained by using the cosine similarity algorithm; silhouette Coefficient (s) in cluster analysis And Davies-Bouldin Index (DBI) to obtain the best cosine similarity of the classification. Messages greater than this cosine similarity are regarded as similar, and the TOP5 hot message problem is counted.

This article constructs a set of evaluation system for message responses from the perspectives of relevance, completeness and timeliness. Relevance and completeness use the combination of LDA model and cosine similarity to calculate the degree of relevance. Timeliness is to evaluate the time difference between message reply and message. And get their overall score.

Key words：Smart government, Word2vec, LSTM, TF-IDF, Cosine similarity, Cluster analysis

目录

1 挖掘背景与目标

1.1 挖掘背景:

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，对提升政府的管理水平和施政效率具有极大的推动作用。

1.2 挖掘目标:

我们要构建一个基于自然语言处理技术的智慧政务系统。这个系统能够做到以下三点：

（1） 对网络问政平台的群众留言进行自动分类，能够有效改善以往依靠人工进行留言分类存在的工作量大、效率低且差错率高的问题；

（2） 能及时发现热点问题并生成一张热点问题明细表，有助于相关部门进行有针对性的处理，提升服务效率；

（3） 能够对相关部门的留言回复进行评价，有助于相关部门发现自己的不足加以改善。

2 . 问题分析

2.1 问题一的分析

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。首先，对附件 2 中的数据进行数据清洗，去除无效数据；其次，对清洗后的数据进行分词、去停用词；然后使用 word2vec 构建语料库，并对词组向量化；最后搭建 LSTM 模型，对向量化后的数据进行分类。

2.2 问题二的分析

根据附件 3 给出的数据，将某一时间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，生成一张热点问题表以及一张热点问题留言明细表。首先，对附件 3 进行数据清洗，去除无效数据；其次，对对清洗后的数据进行分词、去停用词；然后用 tf-idf 构建词向量，然后将 tf-idf 算法和余弦相似度一同使用于向量空间模型中，计算出留言之间的相似度，并统计出热点 TOP5 问题。

2.3 问题三的分析

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、及时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。对答复相关性及完整性的分析：和问题二相似采用 LDA 模型，提取出留言中的主题和答复意见的主题，并对其进行相似度匹配，相似度越高，相关性及完整性评分较高。对答复及时性的分析，运用数据处理，对时间进行转换，并将留言答复时间与留言时间相减，运用量化标准，得出得分。

3.数据预处理

由于三个问题中给出的原始数据都是 xlxs 表格数据，且表格中的文字都是中文，其中存在着大量的冗余信息、非结构化数据以及机器不可识别数据；这将会严重的影响数据挖掘和建模的执行效率，甚至可能会导致建模无法进行。因此需要对给出的原始数据进行如下处理：数据清洗、分词、去停用词。

3.1 数据清洗

数据清洗主要是删除原始数据中的一些无关数据和重复数据，如“/n”等，使得原始数据只剩下中文。

3.2 分词

由于中文文本的特点是词与词之间没有明显的界限，从文本中提取词语时需要分词，本文采用 Python 开发的一个中文模块——jieba 分词[1]，对每一条留言主题及留言详情进行中文分词。

Jieba 分词共有三种分词模式：（1）精确模式，试图将句子最精确地切开，适合文本分析；（2）全模式，把句子中所有的可以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，适合用于搜索引擎分词。Jieba 有着三种算法：（1）基于 Trie 树结构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，使用了 Viterbi 算法。本文主要采用精确模式对中文文本进行切分，用到的算法为算法（1），其中，每个词条对应有向无环图中的一条有向边，并可利用统计的方法富裕对应的边长一个权值，然后找到从起点到终点的最短路径，该路径上所包含的词条就是该句子的切分结果。

3.3 去停用词

在文本处理中，停用词即功能词，是指那些功能普遍，但却没有什么含义的词，它们通常是一些单字，单字母以及高频的单词，比如中文中的“的、了、地、吗”等，英文中的“the、this、an、a”等。为了避免停用词对后来的数学建模进行干扰，一般在数据预处理阶段就将其删除。本文所用的停用词，取自四川大学机器智能实验室停用词表[2]。

4.问题一的解决

对于问题一，在对其进行数据预处理之后，然后使用 word2vec 构建语料库以及词向量，并将其划分为训练集和测试集，再构建 LSTM 模型对其进行训练，对向量化后的数据进行分类。

4.1 word2vec

Word2vec 是 Mikolov 在 2013 年提出的用于快速有效地训练词向量的模型[3].这些模型为浅层双层的神经网络，用来训练以重新构建语言学之词文本。网络以词表现，并且需猜测相邻位置的输入词，在 word2vec 中词袋模型假设下，词的顺序是不重要的。训练完成后，Word2vec 模型可用来映射每个词到一个向量，可用来表示词对词之间的关系，该向量为神经网络之隐藏层。Word2vec 依赖 skip-grams 或连续词袋（CBOW）来建立神经词嵌入。Skip-gram 把一个词从词窗剔除。在 skip-grams 下给定 n 词围绕着词 w，word2vec 预测一个句子中其中一个缺漏的词 c，即以机率p(c/w)来表示。相反地，CBOW 给定词窗中的文本，预测当前的词p(w/c)（如图 1 所示）。我们使用的是 CBOW 对文本进行预测。

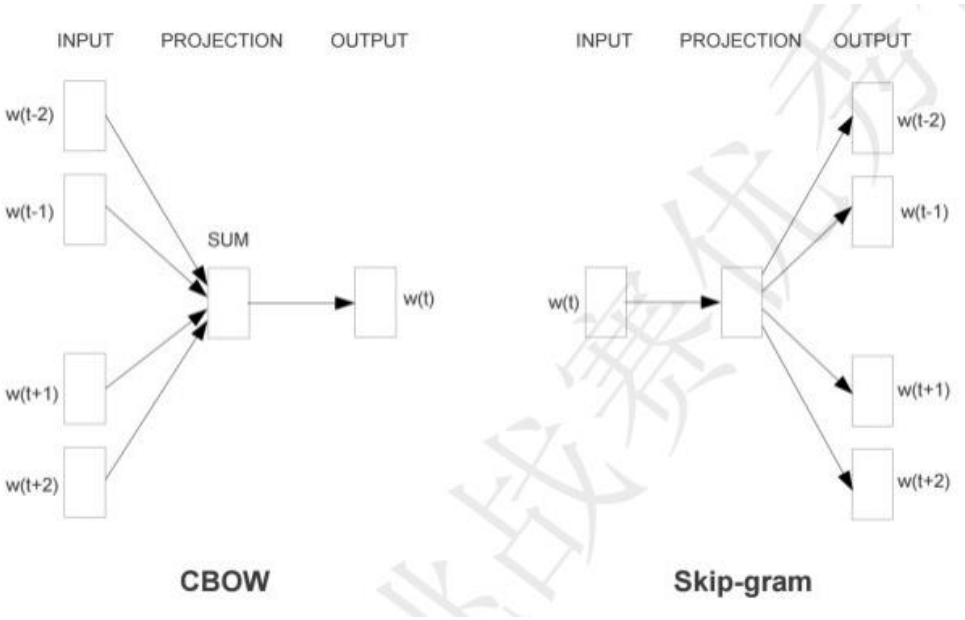

图 1 两种 word2vec 算法网络示意图

为了能够中文文本数据输入 LSTM 神经网络模型进行训练，我们要将中文文本转换为能够被计算机理解的数字形式，而 word2vec 能很好的完成这个任务。首先利用 join 函数构建语料库（图 2），并将 Word2vec 中的词库转换为数字（图 3），建立一个 200 维的向量数组，且每个词代表一定的权重（如图 4 所示）；其次将语料库文本向量化，；最后将向量化后的文本与 Word2vec 词库转换的数字一一匹配；即将中文文本转换为数字，同时将标签转换为独热编码形式，将句长设定为统一巨长，本文采用的句长为每个数据的平均句长，方便后来的 LSTM

模型训练。

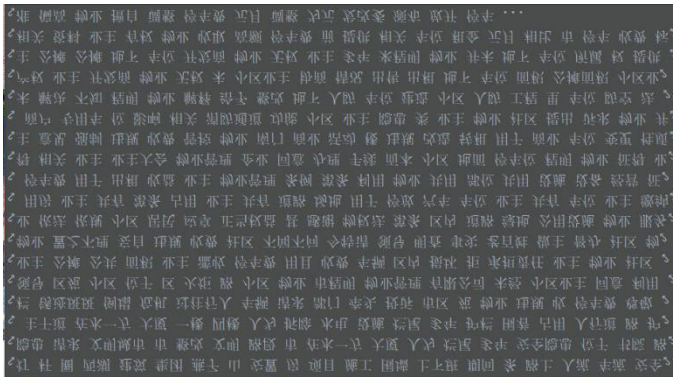

图 2 附件 2 数据语料库

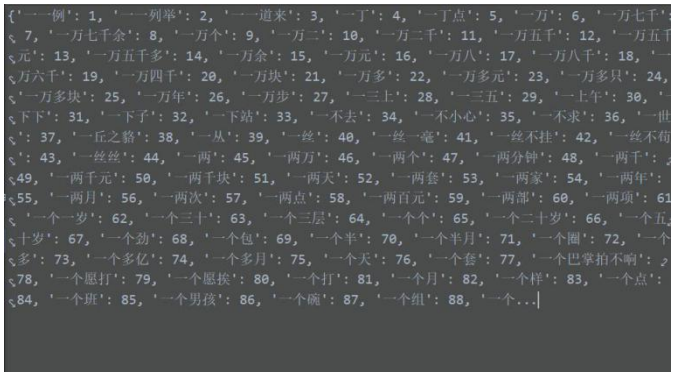

图 3 word2vec 词库转为数字

	0	1	2	3	4	5
0	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
1	-0.01313	0.00504	0.00534	0.01843	0.00393	-0.01270
2	-0.00131	0.00427	-0.00049	0.00789	0.00587	-0.00615
3	-0.00318	0.00073	-0.00213	0.00977	0.00674	-0.00550
4	0.00268	0.00858	0.00080	0.00911	0.00509	0.00013
5	-0.00592	0.00665	0.00123	0.02155	0.02287	-0.02366
6	-0.01237	0.11552	-0.00987	0.13862	0.05662	-0.02544
7	-0.01017	0.00621	0.00141	0.02848	0.00963	-0.01450
8	0.00118	0.00642	-0.00067	0.00739	0.00046	-0.00258
9	-0.00208	0.01236	-0.00263	0.02381	0.00776	-0.01108
10	0.00512	0.00764	-0.00109	0.00165	-0.00267	0.00056
11	-0.00004	0.00651	-0.00604	0.02153	0.01032	0.00483
12	-0.00589	0.00971	0.00201	0.02313	0.01416	-0.00367
13	0.00557	0.00836	-0.00256	-0.00048	-0.00144	-0.00083
14	0.00241	0.01801	0.00424	0.00715	0.00279	-0.00557
15	0.00552	0.00449	-0.00046	0.00792	0.00097	0.00443
16	-0.00517	0.08086	0.02437	0.05586	0.00360	-0.06186
17	0.00263	0.01403	-0.00034	0.00358	0.00018	0.00373
18	0.00553	0.00851	0.00052	0.00011	0.00054	0.00260

图 4 200 维的向量数组及每个词的权重

4.2 训练集和测试集

Sklearn（全称 Scikit-Learn）是基于 Python 语言的机器学习工具。它建立在 NumPy，SciPy，Pandas 和 Matplotlib 之上，在 sklearn 中，有六大任务模块：分别是分类、回归、聚类、降维、模型选择和预处理，如

图 5 所示。

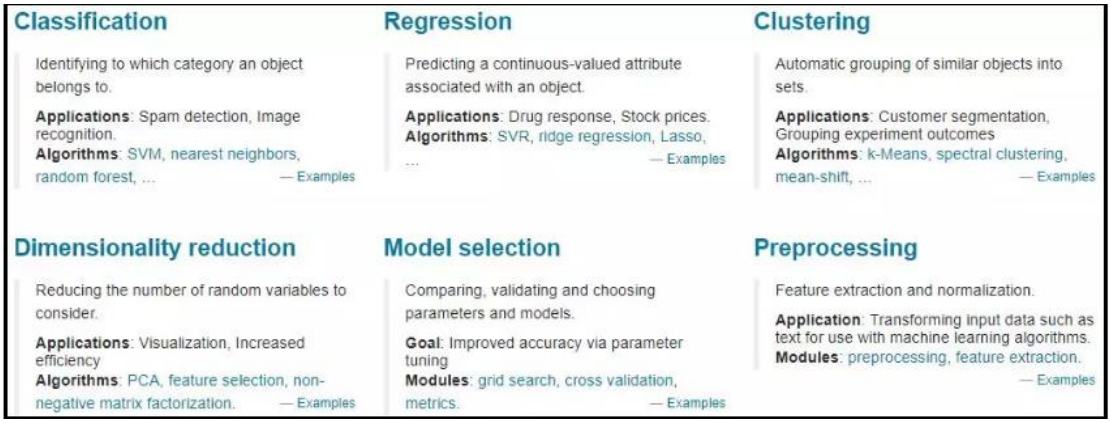

图 5 Sklearn 库的六大模块

本文划分训练集和测试集的方法主要是调用了 sklearn 库中的 train-test-split 函数，将训练集和测试集划分为 8:2.

4.3 LSTM 模型

4.3.1 RNN 与 LSTM

循环神经网络（Recurrent Neural Network,RNN[3]）近年来由于其良好的性能代替深度神经网络（Deep Neural Network,DNN）成为主流自然语言处理建模方案，相对于 DNN，RNN 在隐层上增加了一个反馈，即 RNN 隐层的输入有一部分是前一级的隐层输出，这使 RNN 能够通过循环反馈看到当前时刻之前的信息，赋予了 RNN 记忆功能。这些特点使得 RNN 非常适用于建模。

LSTM（Long Short-Term Memory）是长短期记忆网络，是一种时间递归网络（RNN），主要是为了解决长序列训练过程中的梯度消失和梯度爆炸问题。相比于普通的 RNN，LSTM 能够在更长的序列中有更好的表现。LSTM 已经在科技领域有了多种应用。基于 LSTM 的系统可以学习翻译语言、控制机器人、图像分析、文档摘要、语音识别图像识别、手写识别、控制聊天机器人、预测疾病、点击率和股票、合成音乐等等任务。LSTM 通过刻意的 STM 通过刻意的设计来避免长期依赖问题。记住长期的信息在实践中是 LSTM 的默认行为，而非需要付出很大代价才能获得的能力。所有 RNN 都具有一种重复神经网络模块的链式的形式。在标准的 RNN 中，这个重复的模块只有一个非常简单的结构，例如一个 tanh 层

（图 6）。LSTM 同样是这样的结构，但是重复的模块拥有一个不同的结构，它在算法中加入了一个判断信息有用与否的“处理器”，这个处理器作用的结构被称为细胞。一个细胞中被放置了三扇门，分别叫做输入门、遗忘门和输出门如图 7 所示，这些精心设计的“门”结构实现了 LSTM 遗忘或增加信息能力。一个信息进入 LSTM 的网络当中，可以根据规则来判断是否有用。只有符合算法认证的信息才会留下，不符的信息则通过遗忘门被遗忘。

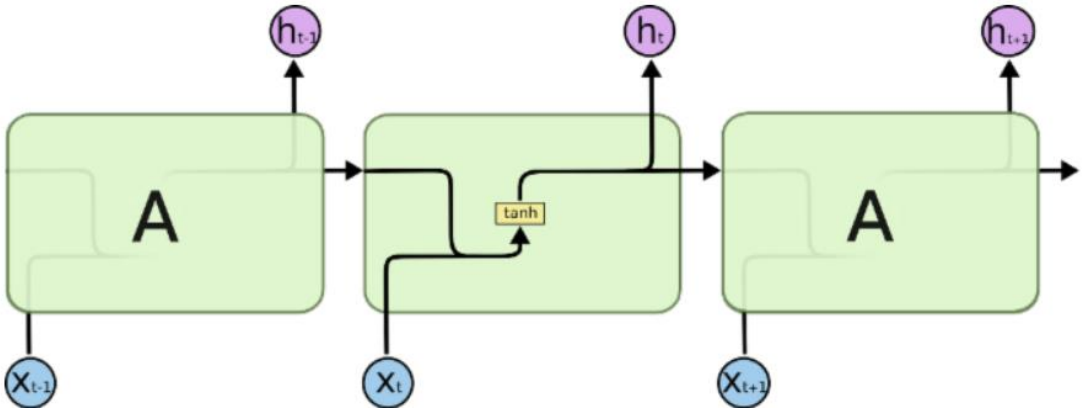

图 6 标准 RNN 中的重复模块包含单一的层

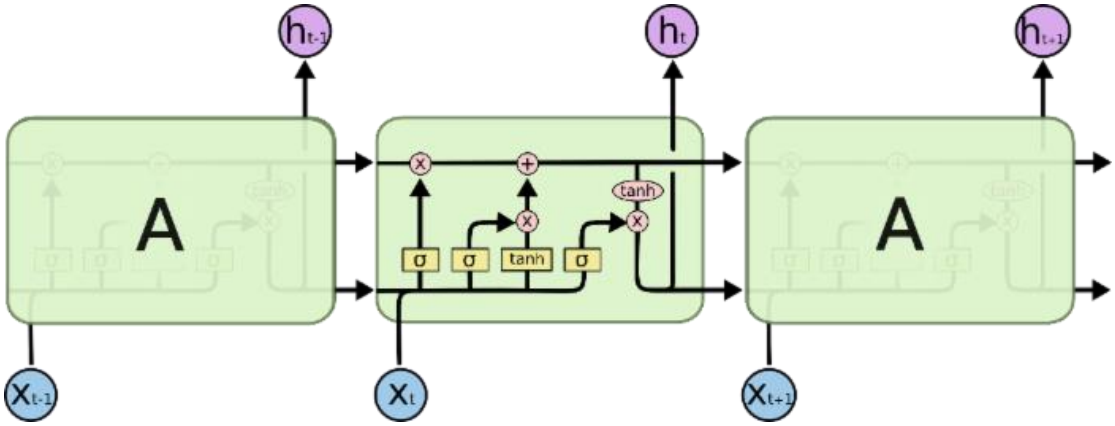

图 7 LSTM 中的重复模块包含四个交互的层

某一时间步 t 的输入门（it）和遗忘门（ft）都以输入变量（即前文建立的 Word2vec 词向量）、上一个时间步 t-1 的输入向量（ht-1）和偏执（b）作为输入，并通过激活函数得到响应值。

忘记门层：忘记门层决定了 LSTM 何时会从系统状态中丢弃信息。其公式为：

$$f_t = \delta(W_f \cdot [h_{t-1}, x_t] + b_f)$$

当我们在输入中看到新的主语时，可以通过使用忘记门层忘记旧的主语，提高语义的准确度。

输入门层：输入门层确定了 LSTM 将要把什么样的信息保存在新的细胞状态中。其计算公式为：

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\overline{c_t} = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$
$$c_t = f_t * c_{t-1} + it * \overline{c_t}$$

输出门层：最终 LSTM 使用输出门层确定需要输出的值，其计算公式为：

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

4.3.2 LSTM 参数设定

LSTM 参数如表 1 所示。

表 1 LSTM 模型参数

输入特征维度（词向量）	200
神经元数量	20
输出类别	13
每次传入样本数	64
迭代次数	10

4.4 评价指标及结果

对于本文中的模型，我们采用召回率（Recall）和精确率（Precision）的调和平均数 F1-Score 来评价我们模型的表现效果。以上指标的详细定义如下：为了方便后面符号的说明定义一个混淆矩阵，如表 2 所示。

表 2 混淆矩阵

	相关	不相关
被检测到的	TP	FP
未被检测到的	FN	TN

（1）召回率（Recall）

召回率（Recall）是检索出的相关文档数和文档库中所有的相关文档树的比率，衡量的是检索系统的查全率。

$$R = \frac{TP}{(TP + FN)}$$

（2）精确率（Precision）

精确率（Precision）是检出的相关文档数与检出的全部文档数的百分比，衡量的是检索系统的正确率。

$$P = \frac{TP}{TP + FP}$$

（3）F1-Score

分类的 F1 值就是精确率和召回率的平均值，具体计算公式为：

$$F1 = \frac{2PR}{P + R}$$

根据 LSTM 模型训练数据可得，测试集的 F1 分数为 0.8。

5.问题二的解决

对于问题 2，在对附件 3 进行数据预处理后，同样利用 join 函数构建语料库，再采用 TF-IDF 算法构建词向量，并利用余弦相似度对构建完成的词向量进行训练，计算出留言之间的相似度，并采用 DBI 和 DI 两个指标对其进行分析，选择一个合适的相似度，统计出最热的五个留言主题，并生成一张热点问题表及一张热点问题留言明细表。

5.1 TF-IDF 算法

5.1.1 简介

TF-IDF（term frequency－inverse document frequency）[4]是一种用于信息检索与数据挖掘的常用加权技术。TF 意思是词频(Term Frequency)，IDF 意思是逆文本频率指数(Inverse Document Frequency)。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。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。

5.1.2 原理

TFIDF 的主要思想是：如果某个词或短语在一篇文章中出现的频率 TF 高，并且在其他文章中很少出现，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。TFIDF 实际上是：TF * IDF，TF 词频(Term Frequency)，IDF 逆向文件频率(Inverse Document Frequency)。TF 表示词条在文档 d 中出现的频率。IDF 的主要思想是：如果包含词条 t 的文档越少，也就是 n 越小，IDF 越大，则说明词条 t 具有很好的类别区分能力。如果某一类文档 C 中包含词条 t 的文档数为 m，而其它类包含 t 的文档总数为 k，显然所有包含 t 的文档数 n=m+k，当 m 大的时候，n 也大，按照 IDF 公式得到的 IDF 的值会小，就说明该词条 t 类别区分能力不强。但是实际上，如果一个词条在一个类的文档中频繁出现，则说明该词条能够很好代表这个类的文本的特征，这样的词条应该给它们赋予较高的权重，并选来作为该类文本的特征词以区别与其它类文档。这就是 IDF 的不足之处. 在一份给定的文件里，词频（term frequency，TF）指的是某一个给定的词语在该文件中出现的频率。这个数字是对词数(term count)的归一化，以防止它偏向长的文件。（同一个词语在长文件里可能会比短文件有更高的词数，而不管该词语重要与否。）对于在某一特定文件里的词语来说，它的重要性可表示为：

$$\mathrm{t}, f_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

以上式子中分子是该词在文件中的出现次数，而分母则是在文件中所有字词的出现次数之和。

逆向文件频率（inverse document frequency，IDF）是一个词语普遍重要性的度量。某一特定词语的 IDF，可以由总文件数目除以包含该词语之文件的数目，再将得到的商取以 10 为底的对数得到，如下式所示。

$$\mathrm{idf}_i = \lg \frac{|D|}{|\{j : t_i \in d_j\}|}$$

其中|D|：语料库中的文件总数

$|\{j:t_i \in d_j\}|$：包含词语的文件数目（即的文件数目）如果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$|\{j:t_i \in d_j\}|$作为分母。

然后再计算 TF 与 IDF 的乘积。

$$\text{tf-idf}_{i,j} = \text{tf}_{i,j} \times idf_i$$

某一特定文件内的高词语频率，以及该词语在整个文件集合中的低文件频率，可以产生出高权重的 TF-IDF。因此，TF-IDF 倾向于过滤掉常见的词语，保留重要的词语。

5.1.3 TF-IDF 在本文的应用

本文使用 tf-idf 作为构建词向量的算法，将附件三的语料库转换为一个个词向量，并赋予它们权重值。

5.2 余弦相似度算法

余弦相似度[5]用向量空间中两个向量家教的余弦值作为衡量两个个体间差异的大小。余弦值越接近 1，则表明夹角越接近 0 度，也就是两个向量越相似，这就叫“余弦相似度”。

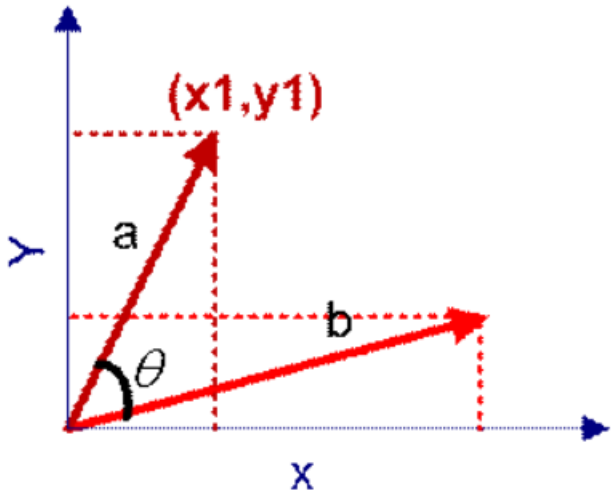

图 8 向量夹角表示

我们知道，对于两个向量，如果他们之间的夹角越小，那么我们认为这两个向量是越相似的。余弦相似度就是利用了这个理论思想。它通过计算两个向量的余弦值来衡量向量之间的相似度值（如图 8 所示）。余弦相似度公式如下：

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

将 TF-IDF 算法构建的词向量代入余弦相似度计算公式中，可以得出每个词之间的余弦相似度，即得出各个留言之间的余弦相似度。

5.3 余弦相似度取值

在选择聚类算法之前，首先来了解什么样的聚类结果是比较好的。我们希望同一个簇内的样本尽可能相似，不同簇的样本尽可能不同，也就是说聚类结果的“簇内相似度”高且“簇间相似度”低[6]。

考虑聚类结果的簇划分，$C=\{C_1,\ C_2,...,\ C_K\}$ 定义：

$$\mathrm{aug}(C)=\frac{2}{(|C|(|C|-1))}\sum_{1\le i\le j\le|C|}\mathrm{dist}(x_i,x_j)$$

$$\mathrm{diam}C=\max\sum_{1\le i\le j\le|C|}\mathrm{dist}(x_i,x_j)$$

$$d_{\min}(C_i,C_j)=\min_{x_i\in C_i,x_j\in C_j}dist(x_i,x_j)$$

$$d_{cen}(C_i,C_j)=dist(\mu_i,\mu_j)$$

其中，μ 代表簇 C 的中心点；$aug(C)$ 代表簇 C 内样本的平均距离；$diamC$ 代表簇 C 内样本间的最远距离；$d_{\min}(Ci,Cj)$ 对应于簇 C_i 和簇 C_j 最近样本间的距离；$d_{cen}(C_i,C_j)$ 对应于簇 C_i 和 C_j 中心点间的距离。基于以上公式可导出下面两个常用的聚类性能度量内部指标：

DB 指数（Davies-Bouldin Index，简称 DBI）

$$DBI=\frac{1}{k}\sum_{i=1}^{k}\max_{j\ne i}\left(\frac{aug(C_i)+aug(C_j)}{dcen(C_i,C_j)}\right)$$

轮廓系数（Silhouette Coefficient，简称 S）

$$S = \frac{d_{\min}(C_i, C_j) - aug(C_i)}{\max(d_{\min}(C_i, C_j), aug(C_i))}$$

DB 指数的计算方法是任意两个簇内样本的平均距离之和除以两个簇的中心点距离，并取最大值，DBI 的值越小，意味着簇内距离越小，同时簇间的距离越大；

S 的计算方法是同簇样本到彼此间距离的均值减去样本到除自身所在簇外的最近簇的样本的均值再除以两者间的最大值，s 取值在[-1, 1]之间如果 s 接近 1，代表样本所在簇合理，若 s 接近-1 代表 s 更应该分到其他簇中。本文采用 DBI 与 S 两个性能指标来判断余弦相似度的取值，即将 DBI 与 S 值作为纵坐标，余弦相似度值作为横坐标画出两条曲线（如图 9 和图 10 所示），取 DBI 的突变点并结合 S 图，最后选取 0.42 作为余弦相似度的取值。

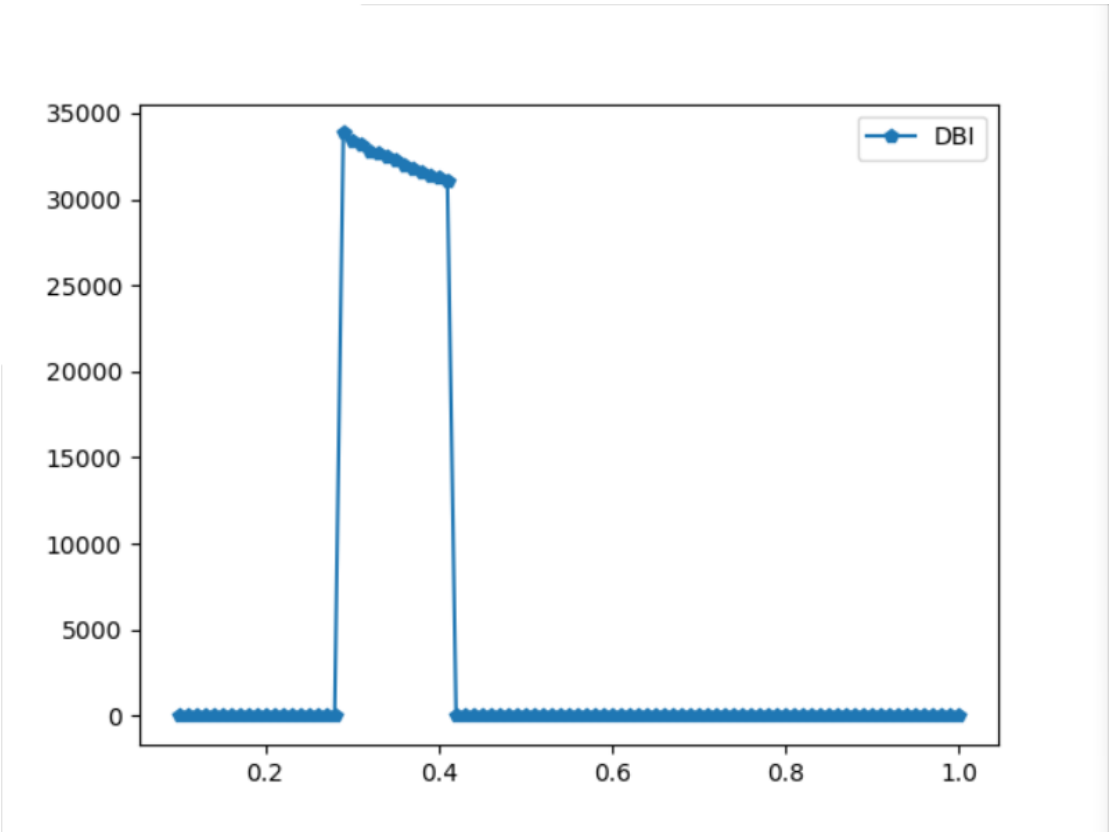

图 9 DBI 与余弦相似度关系图

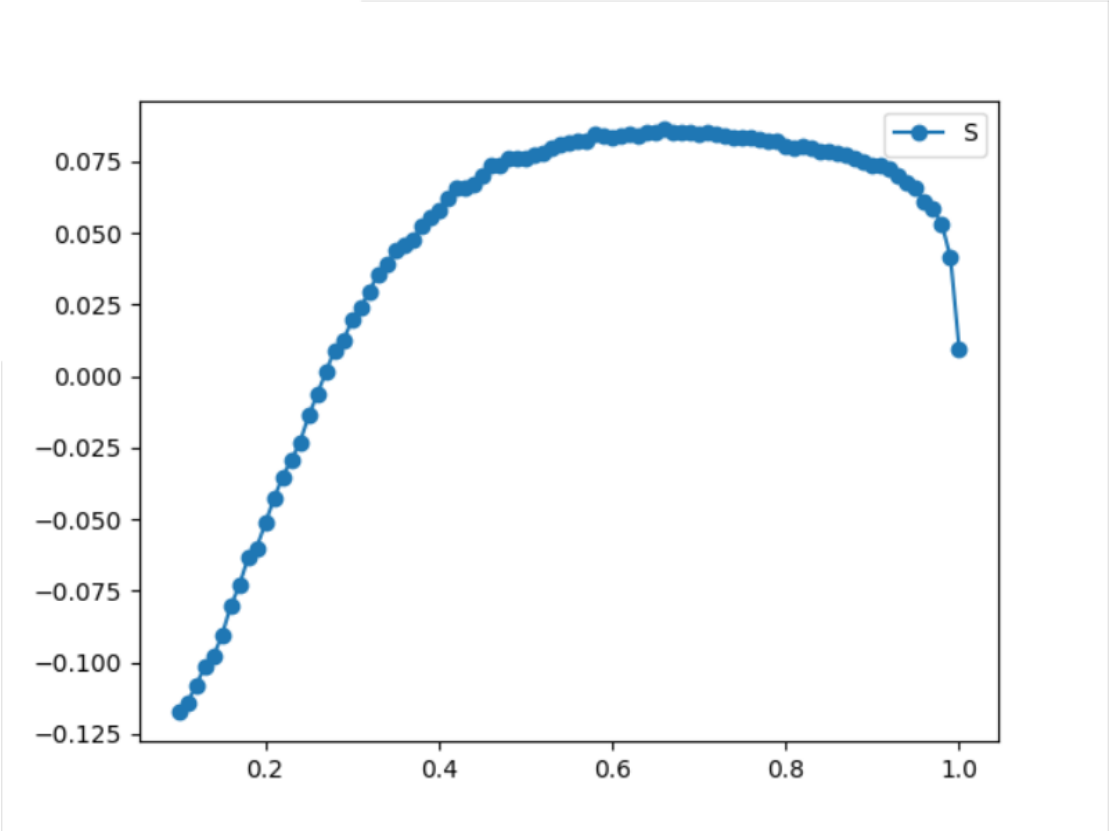

图 10 S 与余弦相似度关系图

5.4 统计与生成热点问题表

得出余弦相似度之后，我们把大于上述所取的余弦相似度的两条留言认为相似，即反映为同一热点问题，将所有相似留言进行统计，并将其点赞数与反对数进行加和，得出每个热点问题的热点值，取排名前五的热点问题利用 Pandas 库中的 Dataframe 模块生成热点问题表及热点问题留言详情表。

6.问题三的解决

对于问题三，我们主要是从留言答复的相关性、完整性以及及时性来对留言回复进行打分。

6.1 答复性与完整性

首先同样是对附件 4 进行数据预处理，再利用 LDA 模型提取出每条留言详情的主题和其对应的留言答复的主题，并对其进行余弦相似度匹配，通过留言与其对应的留言回复的余弦相似度来判断其留言回复的相关性与完整性。

6.1.1 主题模型

主题模型是一种典型的词袋模型，即它认为一篇文档是由一组词构成的一个集合，词与词之间没有顺序以及先后关系。一篇文档可以包含多个主题，文档中每一个词都由其中的一个主题生成。主题模型算法是文本处理与数据挖掘中一个非常重要的方法，它可以有效地从文本语义中提取主题信息。目前，主题模型已经被广泛应用于文本分析领域。主题模型是对文字隐含主题进行建模的方法，它克服了传统信息检索中文档相似度计算方法的特点。

6.1.2 LDA 模型

LDA（Latent Dirichlet Allocation）主题模型，它可以将文档集中每篇文档的主题按照改了分布的形式给出。同时它是一种无监督学习算法，在训练时不需要手工标注的训练集，需要的仅仅是文档集以及指定主题的数量 k 即可。本文主要是通过调用 LDA 模型直接得出每条留言以及留言回复的主题。

6.1.3 相关性及完整性计算

将得出的留言及留言回复的主题先使用 TF-IDF 转换为词向量，并赋予其权重，再使用余弦相似度方法计算其相似度，并根据其相关性及完整性评价标准（如表 3 所示），即可得出其相关性及完整性得分，具体过程和问题二类似。

表 3 相关性及完整性评价标准

相似度	得分
0.8-1	100
0.6-0.8	80

0.4-0.6	60
0.2-0.4	40
0.05-0.2	20
0-0.05	0

6.2 及时性

对于留言答复的及时性，我们自己先制定了一套评价标准，如表 4 所示。

表 4 及时性评价标准

回复时间	得分
七天内	100
十五天内	80
一个月内	60
三个内	40
半年内	20
一年内	0

利用 pandas 库提取出附件 4 中的留言时间及留言回复时间，并对其进行数据预处理，并将其两两相减，并根据评价标准，得出留言答复及时性得分。

6.3 综合得分

我们制定了一个综合得分标准，相关性与完整性占综合得分的 70%，及时性占综合得分的 30%，并根据这个标准，得出每条留言回复的综合得分。

7 参考文献

[1]https://github.com/fxsjy/jieba

[2]http//www.scu.edu.cn/cs/xsky/gjhy/webinfo/2014/12/1416879433948314.htm

[3]Mikolov T,Karafiat M,Burget L,et al.Recurrent neural network based language model[C].Interspeech.2010,2:3.
[4]https://baike.sogou.com/v73026138.htm?fromTitle=tf-idf
[5]https://www.cnblogs.com/dsgcBlogs/p/8619566.html
[6]https://blog.csdn.net/leaf_zizi/article/details/82684921