基于深度学习和自监督学习的问政信息自然语言处理模型

摘要

近年来，随着多种网络问政平台的兴起，各类与社情民意相关的文本数据量不断攀升，给过去主要依靠人工来问政信息处理的工作带来巨大挑战。本题目要求我们基于给出的群众问政留言记录，解决群众留言分类，热点问题挖掘，答复意见评价这三个问题。针对群众留言分类问题，我们使用了多种文档向量构建方法以及机器学习以及深度学习文本分类模型进行实验，最终基于 BERT 的深度学习模型以 0.932 的十折交叉验证 F1-Score 被证明是本题最优的模型。针对热点问题挖掘，我们首先分析了传统聚类模型或者主题模型不适用的原因，然后提出了基于自监督学习与隐狄利克雷分布的热点问题发现模型。我们提出了预标注和后标注两个算法，利用命名实体识别技术在两个不同阶段为数据自动打上标签，从而克服了我们发现的二支配问题，最后利用泊松分布拟合，得到了最热的 5 个热点问题。在第三问答复意见评价中，我们建立了基于回答完整性和相关性的综合评价模型，并且根据我们的主观打分给出了“质量好”和“质量差“的自动二分类，最终我们的评价模型在我们标注的数据下取得了 0.702 的 F1-Score。最后我们分析了我们的模型，以及模型和数据中存在的一些缺陷，提出了几点建议来帮助更好的建设智慧政务系统，期望它可以更好的服务于民。

关键词：深度学习，自监督学习，文本分类，命名实体识别，隐狄利克雷分布

Abstract

In recent years, with the rise of various online platforms for political inquiry, the volume of text data related to social conditions and public opinions has been increasing, which has brought great challenges to the work of political inquiry and information processing mainly relying on human beings in the past. This topic requires us to solve the following three problems: the classification of public comments, the excavation of hot issues, and the response to comments and comments based on the record of public comments. Aiming at the classification problem of crowd comments, we used a variety of document vector construction methods, machine learning and deep learning text classification models to conduct experiments. Finally, BERT's deep learning model was proved to be the optimal model in this question by cross-validation f1-score with 0.932's ten-folds. In terms of hot-spot problem mining, we first analyzed the reasons why the traditional clustering model or the topic model was not applicable, and then proposed the hot-spot problem discovery model based on self-supervised learning and Latent Dirichlet Allocation. We proposed two algorithms, pre-label and post-label, which used named entity recognition technology to automatically label the data in two different stages, thus overcoming the binary dominance problem we found. Finally, Poisson distribution fitting was used to obtain the five hottest hot issues. In the third question response opinion evaluation, we established a comprehensive evaluation model based on the answer integrity and relevance, and gave an automatic dichotomy of "good quality" and "poor quality" according to our subjective scoring. Finally, our evaluation model obtained a f1-score of 0.702 based on the marked data. Finally, we analyzed our model and the problems existing in the data in the topic, and put forward several suggestions to help build the intelligent government system, hoping that it can better serve the people.

Keywords: Deep Learning, Self-supervised Learning, Text Classification, NER, Latent Dirichlet Allocation

目录

1 引言

近年来，随着微信、微博、市长信箱、网络问政平台等渠道逐步成为政府深入群众、了解民意、改善民生的重要方式。各类与社情民意相关的文本数据量不断攀升，给过去主要依靠人工来进行留言的分类和热点问题发现的相关部门的工作带来了极大挑战。与之同时，当今大数据时代下，文本挖掘和自然语言处理技术正在蓬勃发展，极大的拓宽了人工智能的边界。

在此背景之下，本题目给出了利用文本挖掘和自然语言处理技术建立智慧政务系统的要求。具体来说需要基于给出的群众问政留言记录，解决群众留言分类，热点问题挖掘，答复意见评价这三个问题

。

我们对题目中对问题的描述在机器学习的框架下进行了分析和重定义：

1. 群众留言分类问题实际上是一个有监督的多分类问题，需要我们根据有标签的文本数据集训练一个分类器，其评价标准是分类器在测试集上的查准率和查全率。
2. 热点问题挖掘问题实际上是一个无监督的异构数据的聚类问题，数据由留言的文本信息，留言时间以及点赞数和反对数构成，需要我们首先进行聚类并给出热点评价指标并排序最后给出热点相关留言。
3. 答复意见评价是一个开放性问题，需要我们提出一套适用于问政留言答复的质量评价方案。

数据挖掘的第一步是对数据本身的特征进行了解以及预处理，在对本题目所给数据的文本特征分析中我们发现，政务留言信息有一些共性的特点。例如有 94%的留言标题可以被抽取为“地点+问题”的形式，这启示了我们在做热点问题挖掘的时候，我们可以利用命名实体识别技术来抽取留言中的关键对象来辅助我们进行聚类。以及有大量重复出现的词语如“请问”，“咨询”，“投诉”等，这些词语在一般的文本数据中有很强的语义指向性而在我们的问政留言数据中则相反，我们经过分析和实验总结了一套适用于问政留言信息的停用词表，明显改善了分类及聚类模型的性能。我们希望这份词表也可以用于更多的政务信息处理工作中。

在数据预处理环节，经过文本清洗以及中文分词后，我们对留言信息进行向量化目的是为了完成后续的分类训练和聚类。我们使用了五种向量化以及其组合的方式。分别是基于词袋（BOW）模型的向量化，基于词频-逆文档(TF-IDF)分数的向量化，基于 TextRank 分数的向量，基于语料库的 word2vec 向量化以及 BERT 方法。这五种方法贯穿了整个问题的解答过程，其中词袋模型被应用于热点问题挖掘的隐迪利克雷分布（LDA）主题模型中，TF-IDF 和 TextRank 结合的 word2vec 在传统机器学习的有监督学习中的取得了最佳性能，基于 BERT 的深度学习文本分类则是以在十折交叉验证中以 0.932 的 F1-Score 成为了第一问的最优解法。

在第一问留言信息的分类问题中，我们分传统机器学习和深度学习两个板块对问题进行了探究。在机器学习板块中我们进行的第一步是对留言的向量化表示进行数据降维处理。我们实验了主成分分析（PCA），核主成分分析（Kernel PCA）等数据降维方法。第二步，利用降维后的数据我们结合了多种机器学习算法包括支持向量机（SVM），K 近邻（KNN），逻辑回归（LR）等算法，利用十折交叉验证对这些算法的性能进行评估，最终基于 Kernel PCA 降维与使用径向基核函数的支持向量机结合取得了最高的 0.896 的 F1-Score。在深度学习板块我们使用了 BERT，在十折交叉验证中以 0.932 的 F1-Score 成为了第一问的最优解法。

第二问热点问题挖掘中，我们根据留言数据的特点，分析了传统聚类算法不适用于本题目的原因，并提出了两种针对留言中热点问题挖掘的自监督学习方法，利用了隐狄利克雷分布和层次

聚类算法，克服了存在的双支配问题，取得了良好的结果。

第三问答复意见评价中，我们利用命名实体识别和关系抽取技术，基于文档的完整性和相关性两方面，给出了并实现了一套完整的留言评价体系。

我们会在文章的第二部分“问政留言的文本特征分析与预处理”中对题目数据的特点进行分析以及给出我们的数据预处理方法；在第三部分“模型的选择与构建”介绍我们在本赛题各小题中使用的机器学习模型以及相关算法；在第四部分“实验设置”中主要介绍实验的流程，实验结果的判定标准以及相关的参数设置；第五部分“结果”中我们给出了的模型在泰迪杯给出的数据集上的性能以及和其他方法进行比较的结果；第六部分“讨论与总结”中对实验结果进行了分析并对我们的模型做出评价。

2　问政留言的预处理和文本特征分析

2.1　数据清洗和中文分词

经过对题目提供数据的浏览我们发现，附件 2，附件 3 和附件 4 中的留言信息中存在着大量的转义字符以及其他编码字符例如”\n”,”\r”,”\u2800”等，我们首先使用正则表达式对数据进行了处理，去除所有转义字符。

数据清洗后，我们使用 Jieba 分词[1]对所有附件中的文本信息进行了中文分词。分词时我们加载了百度提供的中文自然语言处理停用词表以及四川大学中文常用短语表进行去除停用词的操作。

2.2　问政信息文本特征分析与专用停用词表

经过中文分词后，我们以附件二中留言主题为例，对问政留言中的词频进行了统计，如图 1 所示。我们发现了问政信息的第一个特点，地区词出现频率较高。而地区词并不是一条留言中的关键实体，具有的区分性较弱并且大量出现，于是我们将地区词设置为停用词后，再次对词频进行了统计，结果如图 1 所示。

图 1：附件二的留言主题分词后生成的词云

去除了地区词后我们发现对于留言主题来说依旧有一些出现频次较高的词语具有较低的语义区分度，例如”小区”，“有限公司”，“幼儿园”等，在我们进行文本聚类任务时会对结果造成较大干扰，原因是这些词语经常以后缀的形式出现，例如“丽发新城小区”，“魅力之城小区”，

从依存语法的角度分析这些词语都是其对应支配词的修饰词，无法帮我们定位到关键信息，会对聚类造成一定阻碍作用。经过我们大量实验，我们总结出了一套适用于问政留言信息的停用词表，可以提高后续模型的性能，特殊停用词表可以在附件中的停用词文件夹中查看。

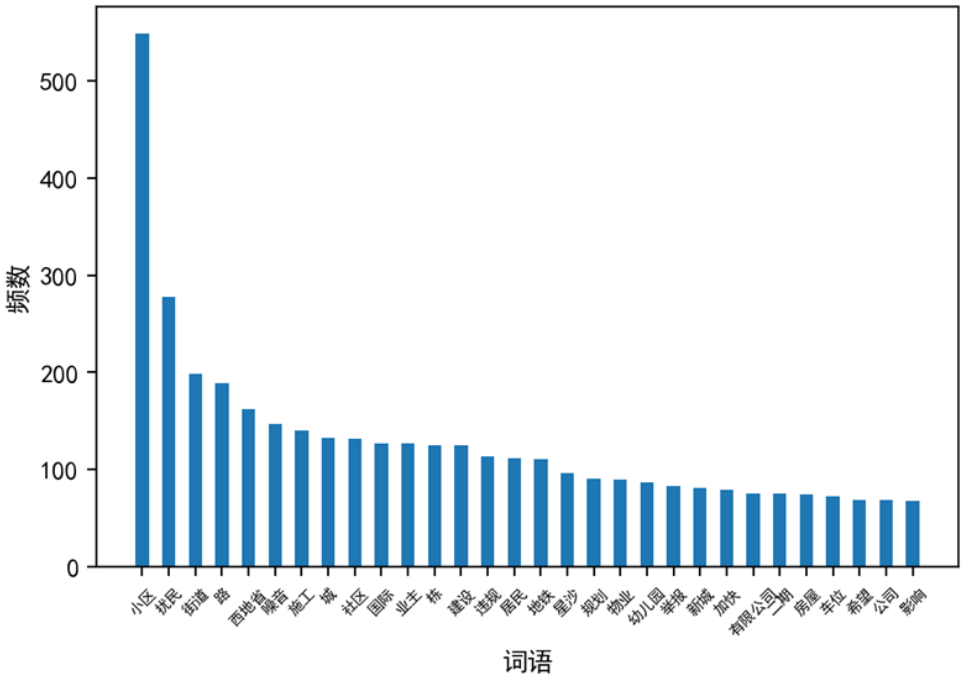


图 2：去除地区词后的留言主题词频 Top30

3　模型的选择与构建

3.1　文档向量的构建

无论是第一问的留言分类任务，还是第二问的热点问题发现任务，其都是以一条留言为单位对数据进行挖掘，因此，将由各种词语组成的文档表示成各种算法可以处理的向量是我们要做的第一步。

3.1.1　词袋模型

在词袋模型下，像是句子或者是文件这样的文字可以用一个袋子装着这些词的方式呈现，这种表现方式不考虑文法以及词的顺序，仅考虑词出现的次数。

例如以下是附件 2 留言主题中经过分词后出现的两个句子：

1)　A 市 丽发 小区 建 搅拌站 噪音 污染 严重

2)　A 市 丽发 新城 小区 侧面 建设 混泥土 搅拌站 粉尘 噪音 污染 严重

基于以上两个句子，可以构建以下的词袋，用集合表示：

BOW = {A 市，丽发，小区，建，搅拌站，噪音，污染，严重，新城，侧面，建设，混泥土，粉尘}

词袋里共有 13 个不同的词，于是以上的句子可以通过一个十三维的向量来表示称为 One-hot 编码，分别为

1) [1,1,1,1,1,1,1,1,0,0,0,0,0]
2) [1,1,1,0,1,1,1,1,1,1,1,1,1]

词袋模型这种向量表示仅考虑了文档中词的频率，在第一问文本分类问题中性能比不上采用了加权修正的 TF-IDF 向量。在第二问的隐狄利克雷分布主题模型中我们使用了词袋模型作为文档的向量表示。

3.1.2 TF-IDF

TF-IDF（词频-逆向文档频率）是一种用于信息检索和文本挖掘的加权方式。TF-IDF 是一种基于统计的加权方法，用于评估一个词语对于一个文档集合的重要程度，具体来说一个词语对于一个文档的的重要性与其在该文档中出现的次数成正比，与它在文档集合中出现的次数成反比。

语料库中第j篇文档构成的词袋中第i个词的 TF-IDF 权重的计算公式为：

$$TFIDF_{ij} = TF_{ij} \cdot IDF_{ij} \tag{1}$$

其中TF_ij表示词频，IDF_ij表示逆向文档频率：

$$TF_{ij} = \frac{n_{ij}}{\Sigma_k n_{kj}} \tag{2}$$

$$IDF_{ij} = \frac{\lg|D|}{\left|\{q: t_{ij} \in D_q\}\right|} \tag{3}$$

其中n_{ij}表示第j篇文档组成的词袋中，第i个词出现的次数。D表示所有文档构成的集合，t_{ij}表示第j篇文档词袋中的第i个词。

在生成文档向量时，向量的维度等于文档集合最终入选的不同的词的数量，对应维度的数值等于对应词在该文档中的 TF-IDF 分数，如果文档中没有对应词则为 0，这里一般会对词语的最低词频和最高词频加以限制避免维度过高带来的计算消耗以及无意义词的噪音。

由于考虑了词语对文档的重要程度，经过我们的测试，TF-IDF 加权表示在留言分类任务中的效果要明显强于普通的词袋模型。此外，TFIDF 还可以 word2vec 结合，获得更优的文档向量表示，我们会在 word2vec 小节介绍。

3.1.3 TextRank

TextRank 算法[2]原本是以一种文本排序算法，由谷歌的网页重要性排序算法 PageRank 算法改进而来，它能够给出一个特定文本中的关键词排序。TextRank 算法根据词之间的共现关系构造网络，具体来说在滑动窗内的词语见两两构建无向有权边。我们使用了 TextRank 算法为一个句子中的词语计算关键性权重。

具体来说，TextRank 算法将文本中的语法单元视作图中的节点，如果两个语法单元存在一定语法关系（例如共现），则这两个语法单元在图中就会有一条边相互连接，通过一定的迭代次数，

最终不同的节点会有不同的权重，权重高的语法单元可以作为关键词。

节点的权重不仅仅依赖于它的入度结点，还依赖于这些入度结点的权重，入度结点越多，入度结点的权重越大，说明这个结点的权重越高；

TextRank 计算关键词权重的算法的原理如下:

$$WS(V_i) = (1-d) + d * \sum_{j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in \mathrm{Out}(V_j)} w_{jk}} WS(V_j) \tag{4}$$

1) 构建词图 $G = (V, E)$，其中 V 为节点集合，由以上 S 中的不同的词构成。然后采用共现关系构造任意两个节点的边：两个节点之间存在一条边当且仅当它们对应的词在长度为 K 的窗口见共现，边的权重为两个节点共现的次数。
2) 节点 i 的权重取决于节点i的邻居节点中 $i-j$ 这条边的权重除以 j 的所有出度的边的权重乘以节点 j 的权重的积，将这些邻居节点计算的权重相加，再乘上一定的阻尼系数，就是节点 i 的权重
3) 由于计算一个点的权重需要考虑到与之相连的节点的权重，这是一个迭代的过程，我们一般设置一个最大迭代次数，迭代的传播各节点权重，直到最后算法收敛。

与 TF-IDF 类似，TextRank 也是一种关键词的加权方式，其考虑了上下文关系得出的权重，我们使用了 TextRank 结合 word2vec 的方式取得了传统机器学习算法在第一问留言分类问题中的最优解。

3.1.4 word2vec

Word2vec 是用于进行词嵌入（word embedding）的预训练模型[3,4]。在使用 One-hot、Tf-idf 或 TextRank 生成文档向量时，由于字典通常较大，因此结果向量通常维数很高并且十分稀疏。而通过 Word2vec 进行词嵌入，则可以将词语以低维稠密向量进行表示，使后续算法运行更加高效。同时，词向量还带有词语本身的特征，因此可以更好的表示近义词之间的关系。

Word2vec 使用 Skip-gram 模型训练得到词向量[3]。Skip-gram 接受一个词作为输入，并以该词周围可能出现的词作为输出。采集样本时，通常采用滑动窗口，即对句子中词 w_t 取其附近长度为$2c+1$的窗口作为该词的上下文。因此训练的目标即训练分布$\mathrm{P}(w_t | w_{t-c} \cdots w_{t-1} w_{t+1} \cdots w_{t+c})$，采用极大似然估计可以得到代价函数

$$-\frac{1}{T} \sum_{t=1}^{T} \sum_{-c \le j \le c, j \ne 0} \log P(w_{t+j} | w_t) \tag{5}$$

Skip-gram 模型采用 Softmax 函数定义分布 $P(w\_(t+j) \,|w\_t)$

$$P(w_{t+j} | w_t) = \frac{\exp({v'_{w_O}}^T v_{w_I})}{\sum_w \exp({v'_w}^T v_{w_I})} \tag{6}$$

其中 v_w 为单词 w 的输入向量，v'_w为单词 w 的输出向量。因此，该模型可以等价为一个仅有一个隐藏层并且输出层为 Softmax 层的神经网络。其接受 One-hot 编码的单词，因此隐藏层权重为 N 个 v_w，即目标词向量。在实践中，训练时可以采用哈夫曼树建立层次 Softmax 函数，

结合负采样策略以减少计算量。

在本题中我们使用了基于百度文库的 256 维的预训练词向量，在题目给出的数据中命中率达到了 95.6%。得到词向量之后，可以通过词嵌入得到文档向量。我们利用 TF-IDF 和 TextRank 对一个文档中的所有词的词向量进行了加权求和后得到了文档向量。首先对文档中每个词进行 TF-IDF 或 TextRank 处理以得到词 w 对应的权重 v_w。对权重进行降序排列，并取前 N 个词计算其词向量的加权平均，即

$$\mathrm{V}_{doc} = \sum_{i=1}^{\min\{N,W\}} \frac{v_{w_i}}{Z} \mathrm{Word2vec}(w_i) \tag{7}$$

其中，W 为文档长度，Z 为归一化因子$\sum_{i=1}^{\min\{N,W\}} v_{w_i}$。

3.2 数据降维

在获得了文档向量后，由于原始数据维度过高（TF-IDF 向量维度为 9108）需要进行数据的降维操作来避免计算量爆炸式上升以及维度灾难。值得一提的是，Word2vec 文档表示由于本身采用了低维稠密向量，维度仅为 256，无需进行额外的数据降维处理。

3.2.1 PCA

主成分分析（PCA）是一种无监督的数据降维技术，其对数据的分布有高斯假设。对于原始维度为n的数据集，希望找到 k 个正交的 n 维单位向量组成的一组基底，使得原始数据在这一组基下做线性变换后在每个维度下都具有最大的方差。

主成分分析的第一步是对数据进行中心化，即 $\boldsymbol{x}' = \boldsymbol{x} - \boldsymbol{\mu}$ 这里的 $\boldsymbol{\mu}$ 该数据集的均值超矢量，目的是将各个维度的均值化为 0 以求协方差矩阵。

用X'表示中心化后的数据集，$X' \in R^{m \times n}$，其特征的协方差矩阵为

$$Cov = \frac{1}{m} \cdot X'^{T} X' \tag{8}$$

第二步求协方差矩阵的特征值和特征向量，由于协方差矩阵为实对称矩阵，其一定有n个单位正交的特征向量。而特征向量对应的特征值即为将原始数据投影到该方向后投影点的方差。于是特征值大小排名前K的特征值对应的特征向量就是我们要找的那一组基底。

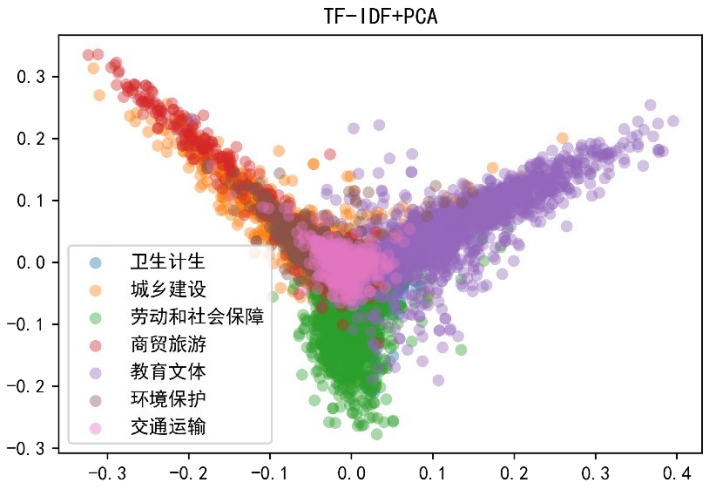


图 3：附件一中留言的 TF-IDF 向量利用 PCA 降至二维后的可视化结果

3.2.2 Kernel PCA

在核 PCA 中[5]，降维问题仍然是求矩阵的特征向量的问题，不过目标协方差矩阵是经过了核变换的协方差矩阵以对线性不可分数据进行降维。

此时的协方差矩阵为：

$$C=\frac{1}{M}\sum_{i=1}^{M}\left(\Phi\left(x_i\right)-\frac{1}{M}\sum_{j=1}^{M}\Phi\left(x_j\right)\right)\left(\Phi\left(x_i\right)-\frac{1}{M}\sum_{j=1}^{M}\Phi\left(x_j\right)\right)^{T} \tag{9}$$

其中 Φ 为非线性变换。我们可以利用核技巧(Kernel Trick)对这个协方差矩阵进行分解而不需要指定非线性变化的形式。

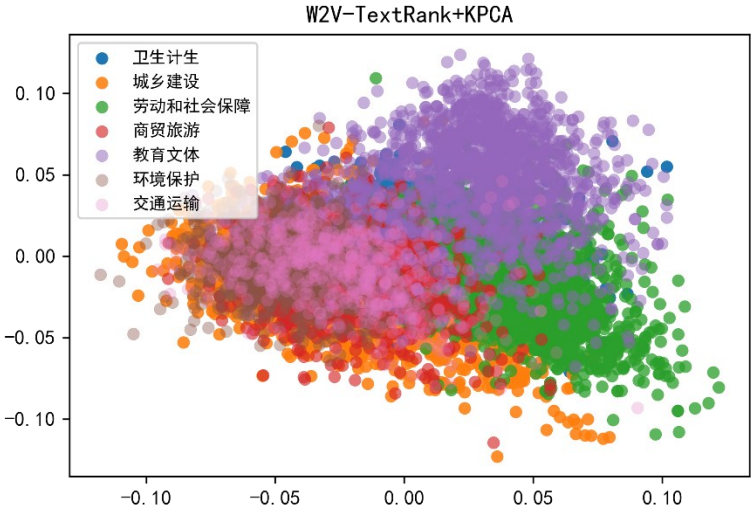


图 4：附件一中留言的 W2V-TextRank 向量利用 PCA 降至二维后的可视化结果

3.3 基于传统机器学习的文本分类模型

3.3.1 逻辑回归

针对问题中留言标签的多个类别，我们采用 SoftMax 模型来解决多分类问题:对于给定的测试输入 x，我们都可以由假设的函数F和参数θ得到在此输入x下每一个类别l的估计值P(y = l|x; θ)，为保证概率和为 1，对得到的结果进行归一化处理得到如下公式:

$$h_\theta\left(x^{(i)}\right)=\begin{bmatrix}P\left(y^{(i)}=1\middle|x^{(i)};\theta\right)\\ \vdots\\ P\left(y^{(i)}=k\middle|x^{(i)};\theta\right)\end{bmatrix}=\frac{1}{\sum_{j=1}^{k}e^{\theta_j^T x^{(i)}}}\begin{bmatrix}e^{\theta_1^T x^{(i)}}\\ \vdots\\ e^{\theta_k^T x^{(i)}}\end{bmatrix} \tag{10}$$

而 SoftMax 模型对应的代价函数为：

$$\mathrm{J}(\theta)=-\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k}y^{(i)}=jlog\frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k}e^{\theta_l^T x^{(i)}}}\right] \tag{11}$$

对应代价函数求导后有：

$$\nabla_{\theta_j}J(\theta)=-\frac{1}{m}\sum_{i=1}^{m}\left[x^{(i)}\left(\left\{y^{(i)}=j\right\}-P\left(y^{(i)}=j\middle|x^{(i)};\theta\right)\right)\right] \tag{12}$$

由此来最小化代价函数即可。

3.3.2 支持向量机

支持向量机（Support Vector Machine, SVM）[6]是一类按监督学习方式对数据进行二元分类的广义线性分类器，其决策边界是对学习样本求解的最大边距超平面。

SVM 使用合页损失函数计算经验风险并在求解系统中加入了正则化项以优化结构风险，是一个具有稀疏性和稳健性的分类器。SVM 可以通过核方法（kernel method）进行非线性分类，是常见的核学习方法之一。

支持向量机常被用于文本分类的任务，在泰迪杯给出的赛题中我们使用支持向量机在第一问中的机器学习方法中取得了最佳的结果。

3.3.3 K-近邻

K 最近邻(k-Nearest Neighbor，KNN)分类算法[7]，是一个理论上比较成熟的方法，该方法的思路是：在特征空间中，如果一个样本附近的 k 个最近(即特征空间中最邻近)样本的大多数属于某一个类别，则该样本也属于这个类别。

3.4 基于深度学习的文本分类模型

3.4.1 BERT

BERT 是一个可用于多 NLP 任务的预训练模型[8]。它采用了双向 Transformer 结构，具有相当强的泛化能力，在多种 NLP 任务中均较其它同类模型有更优的效果。实验使用 BERT 训练端到端的文本分类模型。由于模型为端到端，因此训练中的误差可控，不会出现由于模型组合带来的

误差积累。

相较于其它同类使用 Transformer 的模型，BERT 使用了双向 Transformer 结构以更好的处理文中隐含的双向关系；使用了 Masked Language Model 进行预训练；训练采用了更大规模的数据。BERT 本身学习了单词的特征，因此可用来对输入文档进行词嵌入。通过在最后加入 Softmax 层，可以使用文档特征进行分类。而 BERT 本身参数可以使用预训练得到的参数，并在实际任务的训练过程中微调（Fine-Tuning）。

3.5 基于自监督学习与隐狄利克雷分布的热点问题发现模型

3.5.1 隐狄利克雷分布

隐狄利克雷分布（LDA）是一种基于贝叶斯统计的主题模型[9]，被广泛运用在文本聚类，热点挖掘问题中。它可以将文档集中每篇文档的主题按照 概率分布的形式给出。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。此外 LDA 的另一个优点则是，对于每一个主题均可找出一些词语来描述它。

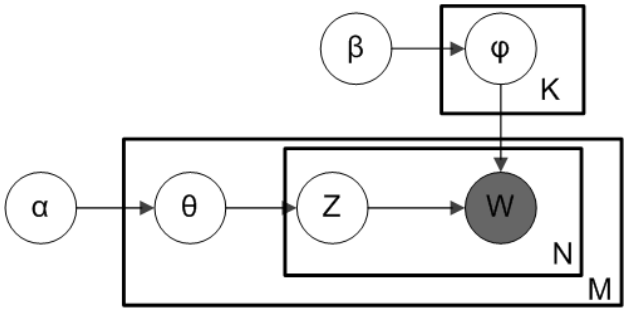


图 5:LDA 的概率图模型

LDA 是一种经典的词袋模型，即其认为一篇文章是由一组词构成的一个集合，词与词之间的顺序关系与主题无关。一篇文档可以由多个主题生成，文档中的每一个词都由其中的一个主题生成。即认为给定主题分布的情况下，一篇文章中词的分布函数满足多项式分布。而狄利克雷分布作为多项式分布的共轭先验，则是主题的所满足的分布函数，狄利克雷分布可以理解为所谓的“分布的分布”。

在 LDA 模型中一篇文档的生成方式如所示:

- 从狄利克雷分布α中取样生成文档i的主题分布θ_i
- 从主题的多项式分布θ_i中取样生成文档i的第j个词的主题$z_{i,j}$
- 从狄利克雷分布β中取样生成主题$\phi_{i,j}$的词语分布$\phi_{z_{i,j}}$
- 从词语的多项式分布$\phi_{z_{i,j}}$中采样最终生成词语$w_{i,j}$

于是模型中所有真实变量和隐变量的联合分布为：

$$p(w_i, z_i, \theta_i, \Phi|\alpha, \beta) = \prod_{j=1}^{N} p(\theta_i|\alpha)p(z_{i,j}|\theta_i)p(\Phi|\beta)p\left(w_{i,j}\middle|\phi_{z_{i,j}}\right) \tag{13}$$

最终一篇文档的单词分布的最大似然估计可以通过将上式的θ_i以及Φ进行积分以及对z_i进行求和得到

$$p(w_i|\alpha,\beta)=\int_{\theta_i}\int_{\Phi}\sum_{z_i}p(w_i,z_i,\theta_i,\Phi|\alpha,\beta) \tag{14}$$

最终可以通过马尔可夫链蒙特卡罗方法(MCMC)和吉布斯采样或者期望最大算法（EM）估计出模型中的参数。

虽然 LDA 对于广义上的文本聚类来说是一个强有力的工具，但是回到题目中来，经过我们的分析，我们发现单纯的主题模型或者基于欧氏距离度量的聚类算法无法有效的对热点问题进行聚类。原因在于问政留言信息的特殊性。在一般的文本聚类任务中，对文档有支配作用的主题只有一个，例如“科学”,“娱乐”,“军事”等，可以通过 LDA 后得到的一个文本最大概率的主题来确定这个文档的主题。而在我们的问题中，一条留言包含了“地点/人群”和“事件”这两个主题，是双支配的，任何一个主题单独都不对该留言的主题起到支配作用，而留言中出现了大量的相同事件不同地点/人群的留言，所以简单的使用聚类算法会往往会忽略其中一个而造成聚类出现偏差如表 1 所示，而若想获得明确的则需要对聚类数量进行大量调参，这是缺少实际意义的。

基于以上分析我们首次提出了利用命名实体技术的自监督文本迭代式聚类算法解决了双支配主题的问题，并在题目的数据上取得了良好的聚类结果。

表 1：使用 LDA 模型得出的某一热点问题及相关留言主题

Index	留言主题
61	A1 区桐阴里小区一直夜间施工
139	请严查 A3 区杜容路工程车环卫车乱停扰民问题
199	A 市赤岗岭地铁站晚上工地施工扰民严重
200	A7 县星沙街道吾悦广场商演噪音扰民
207	A4 区六号线楚雅路地铁施工严重扰民
338	A 市 A5 区树木岭世贸璀璨天城通宵施工噪音扰民
360	西地省人民医院凌晨施工扰民
382	A2 区瑞都商务酒店二楼茶室严重噪音扰民
…	…

3.5.2 自监督学习框架

自监督学习的基本思想是模型可以自动提取数据中的特征对未标签的样例进行标注。

我们提出了两个针对留言热点问题的自监督学习算法，分别为后标注算法和预标注算法。
这两种算法的差别在于对数据标签进行生成的时间，前者是在聚类算法运行之后对同类内的数据进行标注和等价类的划分，具有更高的召回率但是计算代价较高；后者是在聚类算法运行之前进行标注和等价类的划分，计算代价较小但是精确性不如前者。

算法 1：后标注

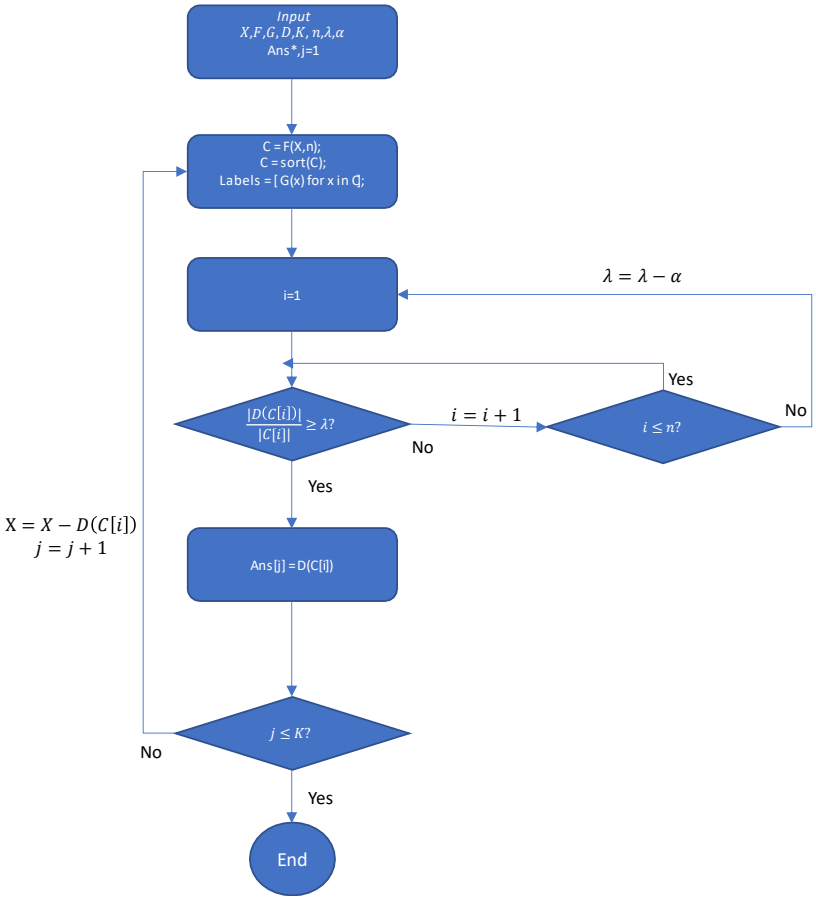
输入：原始数据X，聚类函数$F(X,n)$，标签生成函数$G(x)$，Top 问题数K, 隐含主题数n，信度阈值λ，信度衰减因子α，最大等价类划分函数D。

算法流程图如流程图 1 所示。如果我们需要挖掘出排名前K的热点问题，使用本算法我们需要进行K次聚类。

1) 每一次聚类后，首先对聚类结果类簇的集合按照类中元素数量降序排列
2) 利用命名实体识别对所有元素中的地名，人名实体进行识别，处理结果经过二次分词后作为数据的标签
3) 调用等价类划分函数，将标签具有重叠的元素作为等价类
4) 统计该类簇中最大等价类中元素数目与该类簇大小的比值，如果比值大于信度阈值，则将该类簇计入最终聚类结果进入下一轮循环；如果比值小于信度阈值，则进入下一个类簇，重复直至所有访问过类簇
5) 如果访问过所有类簇后没有符合条件的等价类，则利用信度衰减因子降低信度阈值，重新开始访问类簇
6) K轮循环结束后，最终结果集合中的K个类簇就是我们要找的K个热点问题

算法收敛性分析：

在一轮循环中决定收敛性的是信度阈值λ，其衡量的是聚类后某一类簇的聚类的可信度，如果该类中最大等价类占比超过该阈值，则认为这个类是可信的，反之认为该类不可信。如果所有类簇都被认为不可信，则该阈值会按照衰减因子α线性衰减,最终为 0. 所以在判断信度过程中一定会有类簇的信度高于阈值，从而该算法收敛。



流程图 1：后标注自监督算法流程图

算法 2：预标注

预标注算法一开始就根据命名实体识别的结果对等价类进行划分，然后在每一个等价类中进行聚类。

预标注阶段，首先对每条留言进行 NER。为了统一命名实体粒度，对于每条留言的 NER 结果做如下处理：

1) 若两个结果在原文中相邻，则进行合并。如：“A 市”“A13 区”相邻，则合并结果为“A 市 A13 区”
2) 若两个结果在原文中相隔较近，则基于规则进行判断。若间隔内容符合间隔词，则进行合并。如：原文为“桐梓坡路与麓松路”，NER 结果为“桐梓坡路”“麓松路”，其间隔词“与”符合规则，则进行合并。

如是处理的 NER 结果将包含最大量的信息，如：“A3 区青青家园小区乐果果零食炒货”“A 市江山帝景”。

预处理之后，对实体进行聚类以获得实体的等价描述。因此需要将实体映射到对应的实体向量。实体向量矩阵 M_{NER} 的计算方法为 $M_{NER} = normalise(M_{TF-IDF} \circ M_{Pos})$。其中，$M_{TF-IDF}$ 为以实体为文档计算出的 Tf-idf 文档矩阵，M_{Pos} 为实体词的位置编码矩阵，两者计算 Hadamard 积后进行正规化即可得到 M_{NER}。 M_{Pos} 引入了实体词信息单位的先验位置信息，如对于“A7 县”

“山水湾小区”“幼儿园”，“幼儿园”的层级最细，因此在分类中的权重应当最大，若不做处理则很可能出现“A7 县山水湾小区”“A7 县山水湾小区幼儿园”被聚为同一类的问题。位置信息的计算使用位移的正态分布，对于信息单位总数为 l 的实体，其 x 处信息单位的权值为

$$\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(x-l)^2}{2l^2}\right) \tag{15}$$

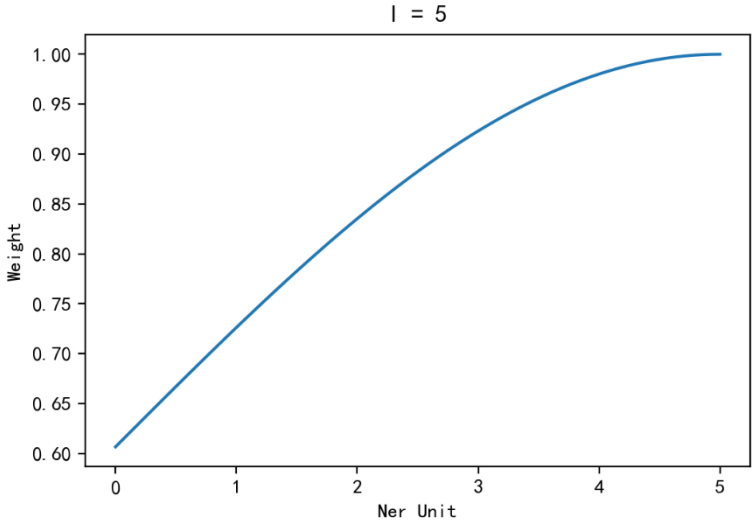


图 6：l=5 时的权值

由于无法事先确定 NER 类别数，因此采用了凝聚层级聚类算法（HAC）进行聚类。HAC 每次合并最接近的两个子类，重复直到所有数据合并为一个大类。实践中，使用欧几里得距离进行聚类，并通过对同表型距离（Cophenetic distance）设置阈值以获得 NER 类别。通过聚类，近似表述的相同实体将得到归类。在若干表述中，抽取实体组成部分（即 NER 结果中的连续信息，如：“A 市”“A13 区”“魅力之城”）最长的作为该类别的实体描述。

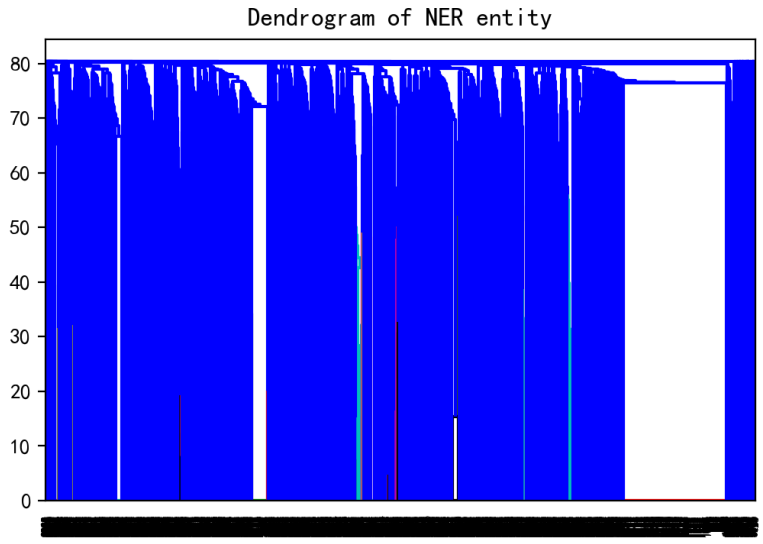


图 7：实体 NER 结果树状图

表 2：实体聚类后得到的部分实体类别（粗体为抽取结果）

类别	实体
2047	**A 市 A3 区中海国际社区三期**
2047	A3 区中海国际三期
2047	A3 区中海国际社区三期
2047	A 市中海国际社区三期
2428	A 市国王陵考古遗址公园
2428	国王陵国家考古公园
2428	A 市国王陵考古公园
2428	**A 市国王陵国家考古遗址公园**

经过 NER 聚类后，每一篇文章将会被打上多个实体标签。因此，需要通过聚类结果选出最终问题相关的实体。对每一篇文章的若干实体 $\{w_i\}$，分别计算其信息熵

$$\mathrm{H}(w_i) = -\frac{|C_i|}{\sum_j |C_j|} \log \frac{|C_i|}{\sum_j |C_j|} \tag{16}$$

并选择具最大信息熵的实体作为最终的实体标签。由此，通过实体标签唯一确定了数据集的一个划分。

完成数据集划分之后，需要在每一个类别内再次进行聚类。由于第一次分类以及包含了留言的实体信息，因此去除留言分词结果中包含实体信息的词以加大内部距离。之后通过 Tf-idf 得到

留言的文档向量，再次进行 HAC 以得到最终的分类。聚类时，由于类别粒度不同（如：“A 市”和“A 市暮云街道丽发新城小区”），因此聚类时的同表型距离阈值的设置也应当按照大类做出调整。阈值的调节策略大致如下：

1. 基于规则，若匹配一级地名则使用阈值 α。如：“A 市”“A 市 13 区”
2. 基于规则，若匹配二级地名则使用阈值 β。如：“A 市黄兴北路”“A7 县东六线榔梨段”
3. 若无类别（即未提取到实体），则使用阈值 γ
4. 其余类别，根据该类实体 w 的信息熵加权计算阈值

$$\theta - \frac{\eta}{\max\{\mathrm{H}(w_i)\}}\mathrm{H}(w) \tag{17}$$

3.5.3 基于泊松分布的热度值估计

泊松分布的概率函数为：

$$P(X=k)=\frac{\lambda^k}{k!}e^{-\lambda}, k=0,1,\cdots \tag{18}$$

泊松分布适合于描述单位时间（或空间）内随机事件发生的次数。如某一服务设施在一定时间内到达的人数，电话交换机接到呼叫的次数，汽车站台的候客人数，机器出现的故障数，自然灾害发生的次数，一块产品上的缺陷数，显微镜下单位分区内的细菌分布数等等。[]

经过我们的分析，一个问题的热度可以用其单位时间内被网民留言的次数所表示,在这个事件出现到被解决的这个时间段内可以用泊松分布进行拟合，而该事件的热度值则是拟合结果中泊松分布的参数λ 。

我们使用阶乘在实数集上的延拓 Gamma 函数来代替阶乘，于是 k 的取值拓宽到了实数域

$$P(X=k)=\frac{\lambda^k}{Gamma(k+1)}e^{-\lambda} \tag{19}$$

3.6 基于命名实体识别和语义抽取的留言回复评价模型

回复质量需要依据留言内容来做出评判，而在比较留言内容与答复意见之后，我们发现有如下问题：1、用语方式不一：市民留言中多含有口语化信息，而答复意见中都采用书面语。2、侧重点不同：市民留言多强调、咨询具体事件，而答复往往用相关政策进行回复。两者的上述区别导致留言内容与答复意见重合性少，评估困难的问题，为此我们建立了基于命名实体识别和语义抽取的留言回复评价模型。

我们从相关性和完整性对回复内容进行评定，相关性是指：回复内容与留言主题的关联程度，完整性是回复内容包含了留言语义的程度。

3.6.1 额外的数据预处理

我们发现在答复信息中，存在大量官方话语，影响留言质量的判断，因此在去除常用停用词之后，我们对词频进行统计，去除了诸如”谢谢”、”如下”等答复信息中的常用词，具体可见附件中 T3.rar 压缩包内文件。

3.6.2 命名实体识别

去除停用词后，对留言内容和答复意见分别进行命名实体识别，提取特定地点、人物，并

将内容分为有特定地点、人物交集和没有特定地点、人物交集的两个集合，作为相关的判断标准之一。

3.6.3 留言信息的关键信息提取

使用 TextRank 算法对留言内容进行关键词提取

3.6.4 相关性、完整性的判断

对答复意见分词后，使分词结果与留言关键词一一比较，用词向量之间的距离作为两词相似程度S的判断标准，若相似度大于阈值，则判断两词意思相近，遍历即得到答复意见中与留言关键词相似的词语个数和留言关键词在答复中有对应近义词的个数。而答复意见中与留言关键词相似的词语个数除以答复长度，为对应答复的相关性，留言关键词被涵盖的个数，为对应答复的完整性。

$$\mathrm{S} = \frac{\sum_{x \in Ans} \sum_{y \in Ques} Similarity\,(x, y) > \alpha}{len(Ans)} \tag{20}$$

$$C = \sum_{x \in Ques} Similarity(x, y) > \alpha \; \exists \; y \; \in Ans \tag{21}$$

在实现阶段，我们发现这样获得相关性会有两方面的局限性：第一点是当答复内容段数多时，关键词相对分散，导致计算得到的相关性低。第二点是当留言内容简要时，往往会出现一个词对应多个留言关键词的情况，针对这个问题，我们对模型进行了如下两个改进：

1、 对段落过多的答复，我们设置段落长度为固定值。
2、 当答复中有词语与留言关键词意思相近时，之后的相似词判断将不再判断此词语。

得到文章的相关性和完整性后，选取适当的阈值作为划分答复质量好坏的标准，对于相关性和完整性都小于对应阈值，且与留言内容没有对应实体重合的回复，标记为质量差的回复。

3.6.5 模型评估

经过我们的讨论，我们选取了 200 个样本进行手动标记，将质量差的答复信息标记为正样本，使用设计的模型计算召回率与 F1 值，进行模型的评估。

4 实验设置

4.1.1 第一问

我们组合了文档向量的不同表示方式，不同的降维方式，以及不同的机器学习模型，做出了一个综合性的模型评估。

经过我们的实验，我们发现将附件 2 中的留言主题和留言内容合并考虑生成文档向量比单独考虑其一的效果更好。

在构造词袋模型和 TF-IDF 向量时，我们对出现的词语出现次数的阈值进行了设置，其最低出现次数为 2，最大文档频率为 0.3，超出这个范围内的词语不被纳入文档向量的生成过程之中。在

利用 TF-IDF 和 TextRank 加权对文档的词向量求和时，我们只选取了一个句子中权重最大的十五个词进行加权求和。

特征降维时，我们使用了普通的 PCA 以及基于线性核的 KPCA 核基于 RBF 核的 KPCA，将输入数据均降至 200 维。

分类模型方面，我们选取了逻辑回归模型，基于线性核的支持向量机，基于 RBF 核的支持向量机以及 KNN。支持向量机的惩罚项系数我们设置为 1.2，KNN 算法的邻居数我们设置为 10.

深度学习模型的参数设置如表 3 所示：

表 3

项目	值
预训练模型	chinese_L-12_H-768_A-12
最长语句长度	128
批大小	4
训练算法	Adam
学习率	2e-5
总回合数	5

在模型检验方面，我们使用了十折交叉验证的方式来检验我们的模型，分类性能的评价指标我们选取的是题目中所要求的 F1-Score，其计算公式如下：

$$F_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i + R_i}$$

其中P_i为第i类的查准率，P_j为第j类的查全率。

4.1.2 第二问

在后标注法的自监督学习模型中，我们的输入数据是附件三中的留言主题构建成 one-hot 向量。使用的聚类方法是隐狄利克雷分布，其中隐含主题数我们设置为 200。标签生成函数我们使用的是 Hanlp 的命名实体识别接口，并且在提取出实体后对结果进行了二次分词。信度阈值为 0.4，衰减因子为 0.05。

预标注法的模型参数如表 4 所示

表 4

项目	值
间隔词规则	市、县、区、的、之、与、（、）、(、)
实体 HAC 同表型距离阈值	43.0
子类 HAC 同表型距离阈值 α	5.0

子类 HAC 同表型距离阈值 β	10.0
子类 HAC 同表型距离阈值 γ	5.0
子类 HAC 同表型距离阈值 θ	23.0
子类 HAC 同表型距离阈值 η	20.0

4.1.3 第三问

在使用百度提供的中文自然语言处理停用词表以及四川大学中文常用短语表进行去除停用词后，我们统计了词频，并筛选出了答复中常用且与回复内容无关的词汇，做成了词汇表并在答复中去除这类词汇。

在选取留言关键词时，我们结合了留言主题和留言内容，筛选出 20 个关键词，用来进行与答复信息词汇相似度的比较。使用词向量比较相似度时，我们经过比较和分析，选定阈值 α 为 0.45 时可以更好地提取到相近词义又不会将无关词语误判成相似词语。

最后我们给数据集中的最后 200 个留言的答复信息进行了质量划分，并以此作为测试集来判断模型参数的优劣，并对模型进行调整。

5 实验结果

5.1.1 第一问

第一问我们实验的结果如图 8 所示。通过结果可以发现基于 BERT 的深度学习模型比机器学习模型的效果要好，F1-Score 达到了 0.932。

在机器学习模型中，我们发现基于 TFIDF 加权的词向量模型是最佳的文档向量表示，Kernel PCA 和基于 RBF 核的支持向量机组合是最好的机器学习模型，F1-Score 达到了 0.896。

对文档向量模型的性能进行横向比较，我们发现基于 TFIDF 加权的词向量模型 > 基于 TextRank 加权的词向量模型 > TF-IDF 向量。这符合我们的预期，因为 TFIDF 加权的词向量即考虑了上下文，又考虑了词对于文档的重要性。

对降维方式进行横向比较，我们发现性能 Kernel PCA >普通的 PCA > 不进行特征降维，这再次印证了数据降维的重要性也提示了文档向量和标签之间存在着一定的非线性关系。

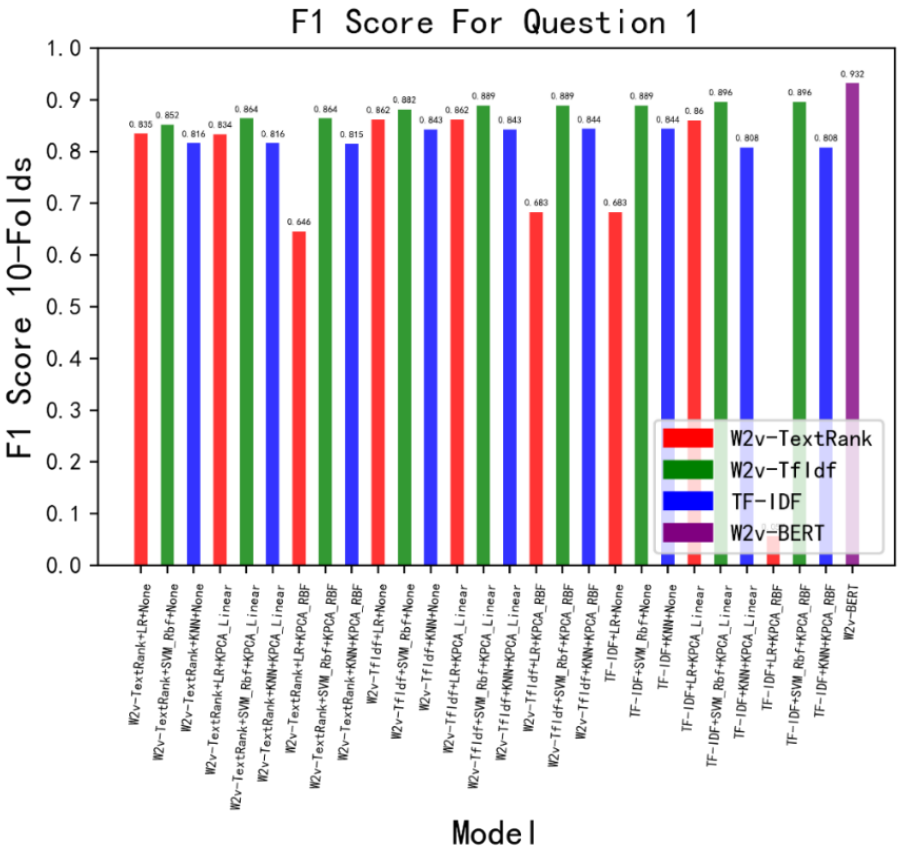


图 8：各文本分类模型的十折交叉验证准确率

深度学习模型的结果表 5 所示，其学习曲线如图 9 所示。

表 5：BERT 文本分类模型训练结果

项目	值
准确率	93.6236%
F1	0.9316645240166951
召回率	92.45232%
训练总步数	8288

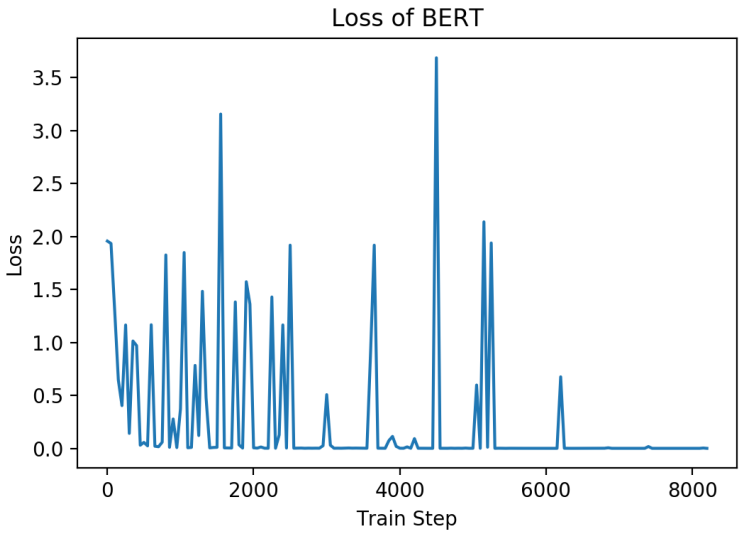


图 9：BERT 模型训练学习曲线

5.1.2 第二问

使用预标注与后标注法得到的热点问题在单位时间内发生次数的概率分布如图 9 和图 10 所示。两个方法筛选出来的热点问题有两个重合。可以看出在两种方法中，“丽发新城”小区的问题都是最热的。对两个方法的结果进行分析后我们发现预标注法由于在最开始就进行了命名实体识别的等价类划分，其划分热点问题的准确度较高，但是召回率较低，而后标注法的召回率较高，我们综合了两者的结果得出了最终的热点问题。具体的热点分析可以在附件中的热点问题表和热点问题留言明细表中查看。

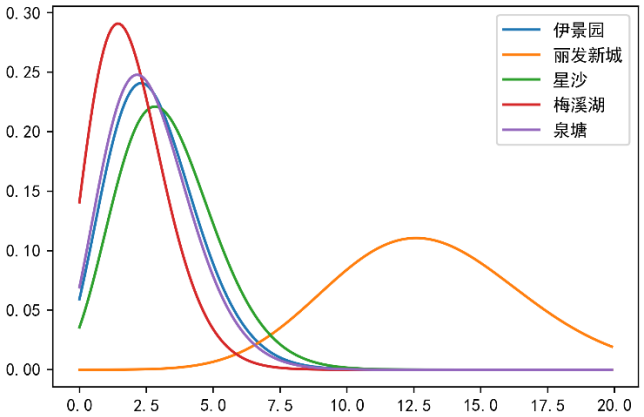


图 10：使用后标注法得到的热点问题（用地名简写代替）经过泊松分布拟合的结果，以 2 小时为基本单位

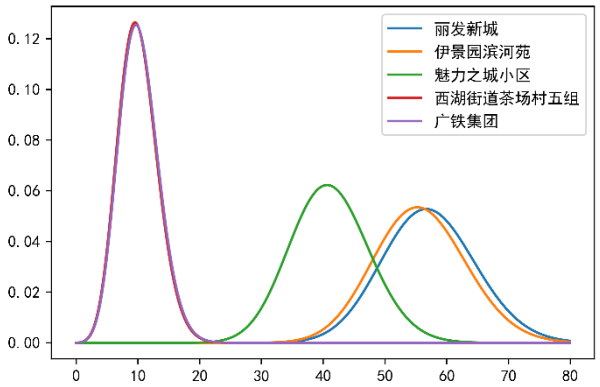


图 11：使用预标注法得到的热点问题（用地名简写代替）经过泊松分布拟合的结果，单位为天

5.1.3 第三问

去除常用停用词后得到的高频词汇 Top30 如下图所示：

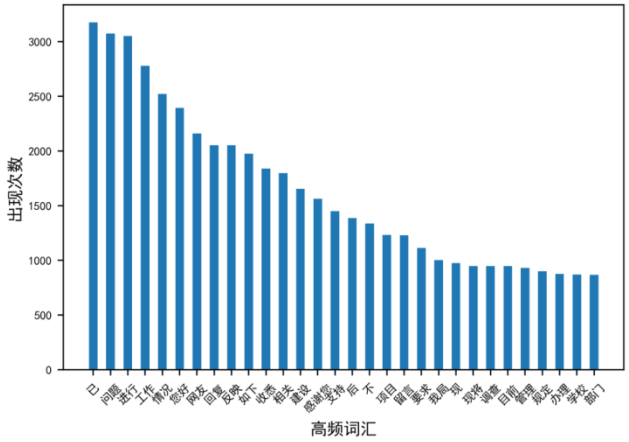


图 12：留言回复中的高频词汇

以回复质量差作为正样本，采用不同的相关性和完整性的阈值，对事先进行标签的 200 个数据进行分类，我们得到如下召回率和 F1-Score 关于相关性和完整性的分布图。

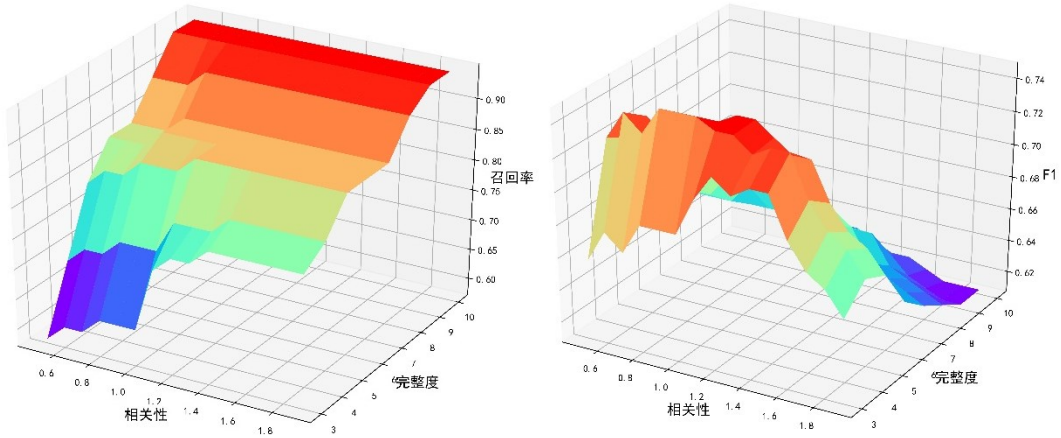


图 13：召回率（左）和 F1-Score（右）关于相关性和完整性的分布图

为了保证市民的问题得到好的回复处理，我们认为应当保证大多数质量低下的回复可以被识别的情况下选择参数，最终经过比较，我们选取相关性阈值为 1，完整度阈值为 6，得到召回率为 0.868，F1 为：0.702

6　讨论与总结

在本赛题的第一问中，最终深度学习模型以3.6%的绝对优势超越了所有的机器学习模型，获得了 0.932 的高分。这在我们的意料之中，因为基于 BERT 的模型，已经在各大自然语言处理的挑战榜单上名列头筹，对于智慧政务系统，我们可以进一步利用深度学习的成果来加快政府相关工作的效率更好的服务于民。

对于第二问我们的两个方法成功克服了我们提到的二支配问题。但是对于热点问题还有一个不可忽略的因素就是外界的知识信息，例如在省会城市中，由于人口密度大，一个问题被反馈的可能性，而在一个省较为偏远的农村地区，由于人口密度较低，留言相对较少，可能反馈的问题在我们的系统中不会被认定为热点问题，但是确实在影响着广大百姓的生活。我们认为在未来的智慧政务系统构建中，可以引入所在区域的人口密度分布信息，关键建筑物名称信息等，来更好的帮助政府相关工作人员“深入基层中去，深入百姓中去”。

第三问中，我们提出了一套留言回复的评价体系，从完整度与相关性两个方面对留言回复进行了评价，并且根据我们的主观打分给出了质量好和质量差的自动二分类。由于我们的判断存在主观性以及标注数据量不足，这个模型还有很大的成长性，我们认为这样一个留言评价系统的存在可以更好对回复人员起到提示和监督作用，可以帮助百姓政府两方更好地解决问题。

7 引用

[1] JUNYI S. fxsjy/jieba[M/OL]. [2020–05–08]. https://github.com/fxsjy/jieba.

[2] MIHALCEA R, TARAU P. TextRank: Bringing Order into Text[C/OL]//Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing. Barcelona, Spain: Association for Computational Linguistics, 2004: 404–411[2020–05–08]. https://www.aclweb.org/anthology/W04-3252.

[3] MIKOLOV T, SUTSKEVER I, CHEN K, 等. Distributed Representations of Words and Phrases and their Compositionality[J/OL]. arXiv:1310.4546 [cs, stat], 2013[2020–05–06]. http://arxiv.org/abs/1310.4546.

[4] MIKOLOV T, CHEN K, CORRADO G, 等. Efficient Estimation of Word Representations in Vector Space[J/OL]. arXiv:1301.3781 [cs], 2013[2020–05–06]. http://arxiv.org/abs/1301.3781.

[5] MIKA S, SCHÖLKOPF B, SMOLA A J, 等. Kernel PCA and De-Noising in Feature Spaces[M/OL]. KEARNS M J, SOLLA S A, COHN D A, 编//Advances in Neural Information Processing Systems 11. MIT Press, 1999: 536–542[2020–05–08]. http://papers.nips.cc/paper/1491-kernel-pca-and-de-noising-in-feature-spaces.pdf.

[6] BOSER B E, GUYON I M, VAPNIK V N. A training algorithm for optimal margin classifiers[C/OL]//Proceedings of the fifth annual workshop on Computational learning theory. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 1992: 144–152[2020–05–07]. https://doi.org/10.1145/130385.130401. DOI:10.1145/130385.130401.

[7] COVER T, HART P. Nearest neighbor pattern classification[J]. IEEE Transactions on Information Theory, 1967, 13(1): 21–27. DOI:10.1109/TIT.1967.1053964.

[8] DEVLIN J, CHANG M-W, LEE K, 等. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J/OL]. arXiv:1810.04805 [cs], 2019[2020–04–20]. http://arxiv.org/abs/1810.04805.

[9] BLEI D M, NG A Y, JORDAN M I. Latent Dirichlet Allocation[J]. Journal of Machine Learning Research, 2003, 3(Jan): 993–1022.