“智慧问政”-基于深度学习的智能政务系统

摘要

随着互联网技术的不断发展和政府的问责机制不断完善，利用深度学习和数据挖掘建立智慧政务系统逐渐成为社会治理创新发展的新趋势。本文构建了基于的智能问政模型，利用自然语言处理和文本挖掘技术挖掘群众留言的特点。

针对问题一，在对附件 2 数据中留言详情部分进行数据分析、数据清洗和预处理后，我们利用谷歌开源的预训练模型 BERT 训练数据。BERT 采用预训练-微调机制，使得我们的下游分类任务的训练成本大大缩小，在经过参数的微调后，模型在验证集上的 F1-Score 取得了 0.926 的分值。

针对问题二，利用基于 RoBERTa 的命名实体识别模型抽取每个留言的特定实体，并在此基础上进行留言分组，然后利用 tf-idf 模型计算文本之间的余弦相似度，从而将相同主题留言归类，最后根据定义的热度指标 Heat 将所有问题排序，并提取排名前五的问题的相关信息（比如时间范围）。

针对问题三，考虑答复与留言的相关性，通过答复建议进行清洗、分词后得到结构化的文本，并将其经过 LDA 主题模型提取答复主题。对答复主题词和留言主题放入 BERT 输入表示层，得到答复主题词、向量后计算余弦相似度得到相似度得分。考虑答复的完整性，答复意见去除标点，空字符后得到的统计文本长度，一般来说长度越长信息量越大，越完整，归一长度化后得到答复完整性得分。考虑答复的及时性，统计答复时间与留言时间时间间隔，并评予时间等级，归一化时间等级后得到答复及时性评分，最后使用加权和计算总得分得出答复质量。

关键字：BERT　　命名实体识别　　tf-idf　LDA 主题模型

Abstract

With the development of Internet technology and the continuous improvement of the government's accountability mechanism, the use of deep learning and data mining to establish a smart government system has gradually become a new trend of social governance innovation and development. This article builds a model based on intelligent questioning, using natural language processing and text mining techniques to mine the characteristics of the masses' message.

For question 1, after performing data analysis, data cleaning and preprocessing on the details of the message in the attachment 2 data, we use Google open source pre-training model BERT training data. BERT uses a pre-training-fine-tuning mechanism, which greatly reduces the training cost of our downstream classification tasks. After fine-tuning the parameters, the model's F1-Score on the validation set scored 0.926.

For question two, we use the named entity recognition model based on RoBERTa to extract specific entities of each message, and group message based on this, and then use the tf-idf model to calculate the cosine similarity between the texts, so as to group the same topic messages. Class, and finally sort all the questions according to the defined heat index Heat, and extract the relevant information (such as the time range) of the top five questions.

For question three, we consider the relevance of the reply and the message, get the structured text after cleaning and word segmentation through the suggestion of the reply, and extract the reply topic through the LDA topic model. Put the response subject words and message subjects into the BERT input presentation layer, and after obtaining the response subject words and vectors, calculate the cosine similarity to obtain the similarity score. Considering the completeness of the reply, the punctuation and the length of the statistical text obtained after the null characters are removed from the reply opinion. Generally speaking, the longer the length, the greater the amount of information and the more complete the response length score is obtained after normalizing the length. Considering the timeliness of the response, the time interval between the response time and the message time is counted, and the time grade is evaluated. After normalizing the time grade, the timeliness of the response is scored. Finally, the weighted sum is used to calculate the total score to obtain the quality of the response.

key word : BERT,Named Entity Recognization tf-idf

一、简介

1.1 背景

近年来，随着互联网技术的不断发展和政府的问责机制不断完善，网络问政平台如微信、微博、市长信箱等逐步成为政府了解民意、集结民智的重要渠道，各类社情民意相关的文本数据量也随之不断攀升，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。利用大数据、数据挖掘、人工智能等技术，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 问题重述

二、基于 BERT 的文本分类模型

针对群众留言的一级分类模型，我们选择现如今在自然语言处理领域 11 项任务中均取得出色结果的 BERT 模型进行留言分类。BERT 模型相对于循环神经网络 RNN 更可以捕捉长距离依赖，对比其他预训练模型，可以获取到真正意义上的双向上下文信息。结合上述原因，我们选择了 BERT 来训练留言分类。

2.1 Transformer 模型

2017 年由谷歌提出的 Transformer 模型采用全 attention 结构取代了长短期记忆网络（Long Short-Term Memory，LSTM）在 seq2seq 模型上取得了出色的成绩。普遍来讲，最具有竞争力的神经序列转换模型其整体架构都是编码器-解码器结构（Encoder-Decoder structure），编码器 encoder 将输入序列$(x_1,\dots,x_n)$映射到另一序列空间表示$z=(z_1,\dots,z_n)$上，然后由解码器 decoder 将 z 映射到一个元素的符号的输出序列$(y_1,\dots,y_m)$。每一步先前生成的符号表示将作为下一步的附加输入。Transformer 在遵循上述的整体架构上，同时为编码器和解码器使用了堆叠式自注意力层（stacked self-attention layer）和逐点全连接层。Transformer 的模型结构如图 2.1 所示。编码器由 N=6 个独立的层堆叠组成，解码器也同样由 6 个层组织。对于解码器，每个独立的层都有 2 个子层，第一层是 self-attention 层以及多头机制，第二层是简单的逐点全连接层。两个子层中都采用残差连接（Residual Connection），使得子层输出的公式变为：

$$Output(x)=LayerNorm\big(x+Sublayer(x)\big)$$

在这里 LayerNorm(x)表示对 x 的归一化，而 Sublayer(x)表示 x 经过子层后的输出。解码器在包含编码器中同样含有的两个子层结构外，还在中间插入了多头注意力层。每个子层同样在最后的归一化上增加残差连接。在这些的基础上，自注意力子层还增加了 mask 机制用以确保在预测当前位置时只取决于小于该位置的输出。

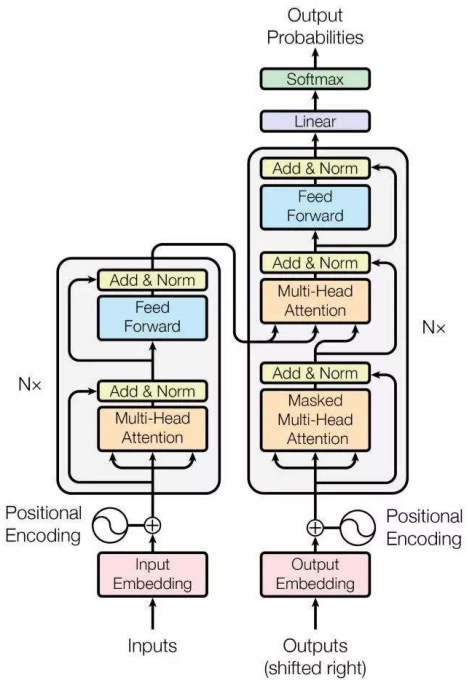

图 2.1 Transformer 结构

2.1.1 自注意力机制 self-attention

区别于 RNN 等传统提取器，BERT 等 Transformer 模型采用的是注意力机制 attention 来对文本序列进行特征提取。针对输入序列的每一个位置，self-attention 机制令模型同样关注其它位置的单词来找到能够有助于更好地编码该位置。

在我们输入一个句子到模型中时，第一步是对句子中的每一个单词进行词嵌入 embedding，由 embedding 生成三个向量：query、key 和 value 向量，这三个向量由 embedding 分别同三个矩阵 W_Q、W_K 和 W_V 相乘得到。如图 2.2 所示。

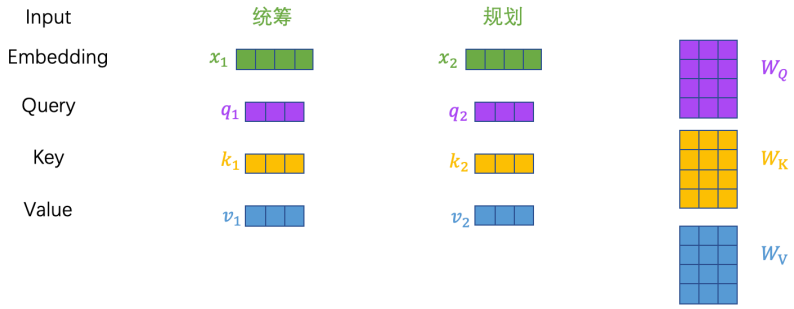

图 2.2 每个单词的输入向量同三个矩阵W_Q、W_K、W_V分别相乘得到对应的 query、key 和 value 向量

接下来，对于每个位置，我们都计算其他位置对该位置的贡献度，该贡献度的分值由目前我们对应的位置的 q 向量点乘其它位置对应的 k 向量得到。例如我们计算所有单词对第一个单词的贡献度，那么第一个单词对其的贡献度分值是 q1 • k1，第二个单词对第一个单词的贡献度是 q1 • k2。得到的结果经过缩放处理（除以 k 向量维度的平方根）和 softmax 层

之后，得到的 softmax score 与 value 向量相乘，这里我们希望能够保持我们所关注的单词与该位置的联系，去除不相关的词汇，如将它们乘以极小的数值，最后将这些带有权重的 value 向量相加起来，就得到了该 position 的 self-attention 表达。

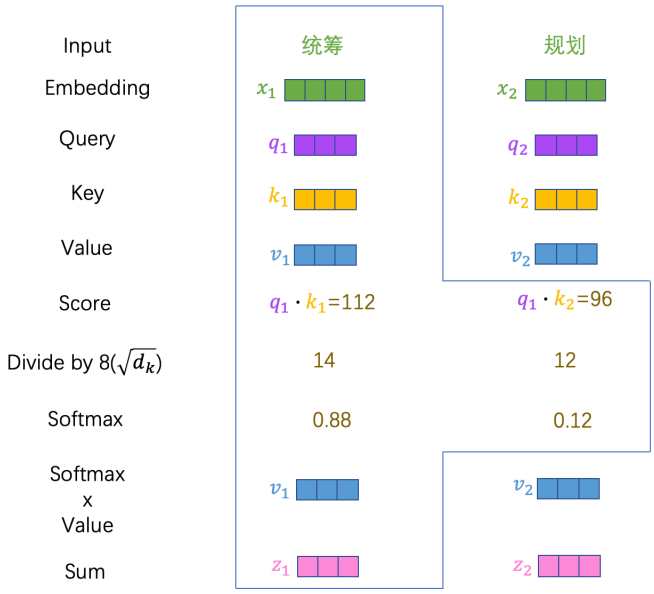

图 2.3 self-attention 机制流程（以第一个单词“统筹”为例，这里我们假设 q/k/v 向量的维度大小为 64）

针对整个输入序列，我们构建词嵌入矩阵，矩阵 X 的每一行代表输入序列的一个单词，该矩阵同权重矩阵W_Q、W_K、W_V相乘得到 Query、Key 和 Value 矩阵。

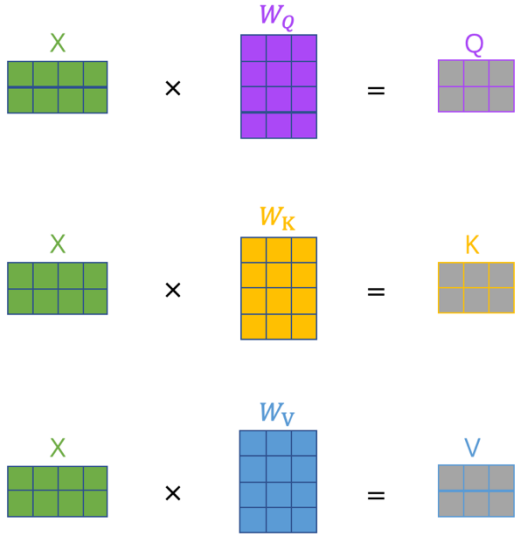

图 2.4 词嵌入矩阵 X 经过转换得到相应的矩阵 Query、Key 和 Value

通过矩阵我们可以把图 2.3 所示的过程压缩成一个公式表达，如图 2.5 所示。这就是 self-attention 机制。

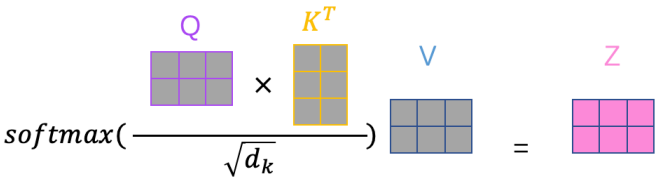

图 2.5 self-attention 的矩阵形式

2.1.2 多头机制

尽管上述的 self-attention 机制帮助模型更好的关注序列上的其它位置对某词的贡献，它仍旧被该词本身所主导，而其它词在 z 上的编码非常小。在 self-attention 层上增加的多头机制（multi-headed mechanism）解决了该问题，同时，它为注意力层提供了多个表示子空间，这有助于提升注意力层的性能。

多头机制下，我们会拥有多个 Query、Key 和 Value 矩阵（通常在 Transformer 模型中有 8 个）。在经过训练后，每一个集合都将输入的嵌入式表示映射到不同的表示子空间。

接下来，我们合并这多个矩阵，然后乘以一个W^O权重矩阵，得到最终的结果 Z。

2.1.3 逐点全连接网络

逐层神经网络分别地应用在每一个位置，其包含了两次线性变换和一次 ReLU 激活函数：

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b2$$

需要注意的是对于同一层别的不同输入序列位置，他们共享同一参数，而不同的全连接层参数并不相同，这一思想同卷积神经网络（Convolutional Neural Network，CNN）类似。

2.1.4 Embedding 和 Softmax 层

同其它的转导模型一样，Transformer 模型将输入和输出转换成维度为d_{model}的向量。对于解码器输出，用常见的线性变换和 softmax 方程将其转换成预测下一输出标志的概率。两个 embedding 层以及 softmax 层之前的线性层共享同一参数矩阵，但在经过 embedding 层时需要将参数乘以$\sqrt{d_{model}}$。

2.1.5 位置编码表示序列顺序

模型中还需要考虑的是输入序列的顺序，而这也是自然语言处理中至关重要的地方。Transformer 在输入的 embedding 里额外增加了一个向量表示，该向量遵循模型学习的特定模式，帮助其学习到每个单词的位置表示或者不同词在序列中的距离，这便是位置编码（positional encodings）。位置编码的维度同样是d_{model}大小，因此它可以和之前的 embedding 相加。位置编码的公式采用正弦曲线版本，公式为：

$$PE_{(pos,2i)} = sin\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

$$PE_{(pos,2i+1)} = cos\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

pos 表示单词的位置，i 表示的是维度。正弦曲线版的位置编码可以令模型外推到比训练集中的序列更长的序列。

2.2 BERT-从 Transformer 模型得来的双向编码表征模型

2018 年谷歌又推出了新的采用多层双向 Transformer 编码器模型 BERT(Bidirectional

Encoder Representations from Transformer）在自然语言处理领域引起了巨大反响，在包括文本分类、命名实体识别等 11 项 nlp 任务中都打破之前记录，取得了 state-of-art 的结果。

2.2.1 模型架构

BERT 在架构上一共分为两步：预训练（pre-trainning）和微调（fine-tuning）。除了输出层面，这两个过程都是用的同样的网络结构。在预训练里，模型在不同的预训练任务训练无标签的数据，得到的权重参数将作为接下来微调的初始化参数，并在微调过程中根据已有标签的数据不断进行 fine-tune。简而言之，任何下游任务（downstream task）在初始化时都采用同样的参数，但其最终的 fine-tuned 模型是不同的。

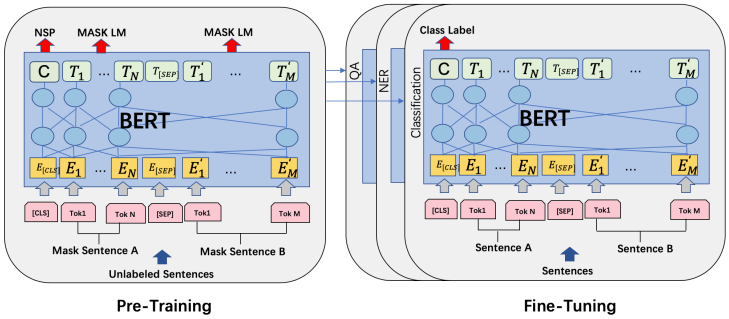

图 2.6 BERT 的整体架构

BERT 是一个多层双向 Transformer 编码器，采用双向的 self-attention（即注意两边的上下文）。作者指明 L 表示神经网络层数，H 表示每个隐藏单元的维度，A 表示 self-attention 头数。BERT 有 2 种大小的模型，分别是 BERT-base（L=12, H=768, A=12)和 BERT-large（L=24, H=1024, A=16)。我们采用的是 base 版本的中文预训练模型。

2.2.2 输入/输出表示

BERT 模型采用 WordPiece 做词嵌入。WordPiece 通过双字节编码算法（Byte-Pair Encoding，BPE）演变，其主要思想在于将单词拆分成 subword 使得词汇表变得更精简清晰。WordPiece 模型受数据驱动，基于构建的数据集合训练语言模型，从所有可能的 subword 单元选择新的单元，新的单元在加入到模型中可以最大程度地增加训练数据的似然值。重复这一步骤直到达到 subword 的期望大小或者概率增量低于某个阈值。

给定训练语料库，WordPiece 首先将词汇拆分成字符片段，而拆分的字符序列需要对应之前的词汇位置在首个字符之前添加符号“_”，这便于之后的解码恢复序列。选择 D 个 wordpiece，在根据所选的 wordpiece 进行分割时所训练产生的语料库在 wordpiece 的数量上是最小的。

WordPiece 很好的维护了字符灵活性以及词汇的高效性。在 BERT 中采用 WordPiece 做词嵌入，对应的词汇表包含 30000 个词汇标识（token）。在每个序列前面增加一个标识[CLS]，对应的最后的隐藏状态被用作分类任务的聚合序列表征。句子组合被压缩成单个句子，我们采用两种方法区分这些分句，一种是每个分句之间增加标识[SEP]，一种是学习每个词汇对应的 embedding 来区分该词汇是否属于分句 A 还是分句 B。在这里我们采用第一种方式。对于给定的词标识，如图 2.7 所示，它的表示由标识嵌入（token embedding）、分割嵌入（segment embedding）和位置嵌入（position embedding）的加和构成。

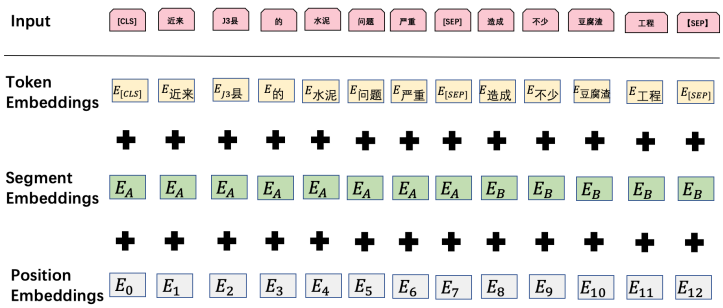

图 2.7 BERT 输入表示

2.2.3 预训练 pre-training

BERT 采用两种无监督方法训练模型，不同于传统的 left-to-right 或者 right-to-left 语言模型。

方法 1:Mask LM　显然，深度的双向模型比单向模型或者像 ELMo 模型两个方向模型的堆叠性能更优，但标准的条件语言模型只能单向训练。在用模型训练 nlp 任务时，人们需要关注词汇的上下文信息。BERT 采用了 mask 方法，在训练过程中随机将 15%的 WordPiece 标识转为标识[mask]，预训练的任务就是预测被替代的标识。

方法 2: NSP（Next Sentence Prediction）　这一任务涉及的是问答（Question Answering，QA）和自然语言推理（Natural Language Ingerence，NLI），该任务基于对上下句子的关系理解，主要目标是给出上一个句子 A，预测下一个句子 B，因为和题目无关在这里不详细介绍。

2.2.4 微调 Fine-tuning

Transformer 中的自注意力机制使得下游的微调变得直接了断。对于每个任务，只需要将特定的输入输出插入到 BERT 中然后对参数进行端到端的微调。

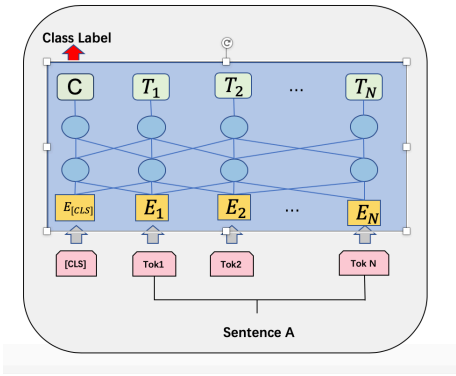

图 2.8 BERT 分类任务的 fine-tuning 结构

2.3 训练模型

2.3.1 预处理

在给出的数据集中，原始数据需要经过分析和处理才能作为 BERT 的输入进行训练。

附件 1.xlsx 将群众留言分为了 15 个一级类别。我们将附件 2.xlsx 的数据读取并进行统计，得到的统计信息如图 2.9 所示：

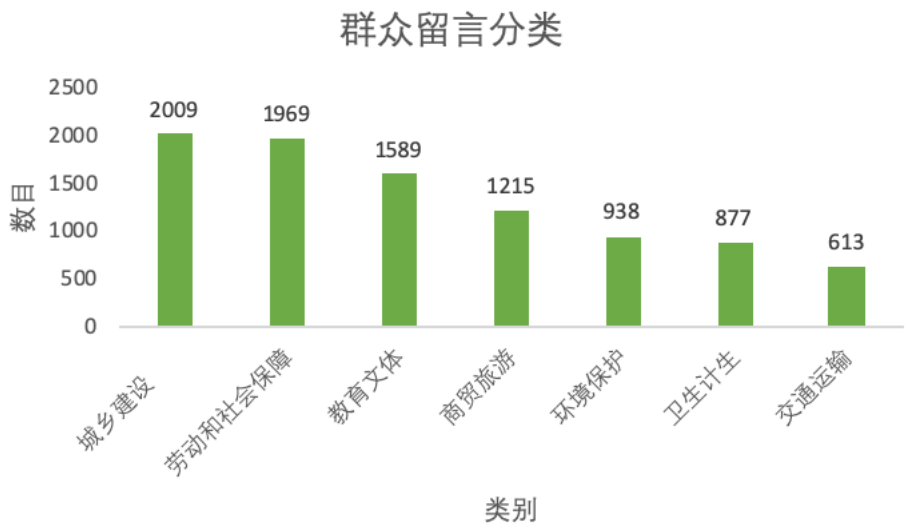

图 2.9 群众留言分类分布

实际的群众留言只存在于 7 个类别当中，分布数量不均匀。我们将实际训练模型的标签定义为这 7 个类别标签，将原文件的顺序进行 shuffle 打乱，使得留言样本随机排序。

通过分析我们看到，群众留言在实际生活中多有口语化用词较多、符号不规范的特点。口语化用此多也就表示高频词汇和单字出现较多，比如“我、的、地、了、吗”等，这一类词出现广泛，而实际意义较少，我们称为“停用词（stop words）”。针对停用词，在预处理阶段我们构建停用词表，在训练过程中将其删除，避免对文本尤其是短文本训练造成负面影响。停用表我们采用谷歌提供的中文停用词表。针对符合不规范，我们利用正则表达，剔除特定的特殊符号如转义字符、空格等。完成数据清洗，提取出留言详情作为训练样本，分类作为训练标签。这便是预处理的全部过程。

2.3.2 实验环境

实验环境配置如表 2.1 所示：

CPU	Intel(R) Xeon(R)
显卡	Titan XP
操作系统	Ubuntu 16.04
Python	3.7
CUDA	10.1
cuDNN	7.6.5
tensorflow	1.14.0
pytorch	1.4.0

表 2.1 实验环境

2.3.3 训练过程

（1）模型的准备：我们采用谷歌提供的 BERT-base-Chinese 模型。BERT-base-Chinese 包含以下文件：

- bert_model.ckpt：模型变量载入文件。
- bert_config.json：模型参数的设置文件。
- vocab.txt：中文训练文本词典。

（2）数据准备：将上述做好预处理数据集分为训练集和验证集，样本数量比例为 8:2，保存为 train.tsv 和 dev.tsv 文件，格式为 UTF-8，存放在 dataset 文件夹中。

（3）代码修改：修改 bert 源代码包中的 run_classifier.py 文件，创建群众留言分类的自定义 processor 类，在该类中设定数据集文件名以及标签类别，在源代码的 processor 词典中添加该定义类。

（4）配置训练脚本，设定训练模型，任务名称，数据路径和输出路径，以及超参数设定。

2.3.4 训练结果

本文模型采用查准率（Precision，P）和查全率（Recall，R）二者的调和平均数 F1-Score 来评价模型效果。为了说明，这里首先给出混淆矩阵的定义：

真实值 预测值	正类	负类
正类	TP	FP
负类	FN	TN

表 2.2 混淆矩阵

- TP（True Positive）：正类项目被预测为正类
- FP（False Positive）：负类项目被预测为正类
- FN（False Negative）：正类项目被预测为负类
- TN（True Negative）：负类项目被预测为负类

（1） 查准率（Precision，P）：预测值为正类的样本中预测准确的比例

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

（2） 查全率（Recall，R）:真实值为正类的样本中预测准确的比例

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

（3） F1-Score：查准率和查全率的调和平均数

$$F1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i + R_i}$$

（4） 准确率（Accuracy，Acc）：全部样本中预测正确的比例

$$Acc = \frac{TP}{TP + FP + TN + FN}$$

模型在验证集上的性能指标如表 2.3 所示：

指标	值
P	0.927
R	0.925
$F1$	0.926
Acc	0.928
$Loss$	0.267

表 2.3 模型性能指标

三、民生热点挖掘模型

及时发现民生热点，可以及时调动有关部门进行关注并有针对性地处理，从而大大提升行政效率。民生热点有时效性、集中性的特点。某一时段内，特定的人群可能会在特定的地点集中遇到民生问题并反映，而对留言问题的点赞或者反对也反映了人们的认同度，也因此隐含了问题的热度信息。

3.1 热点挖掘流程

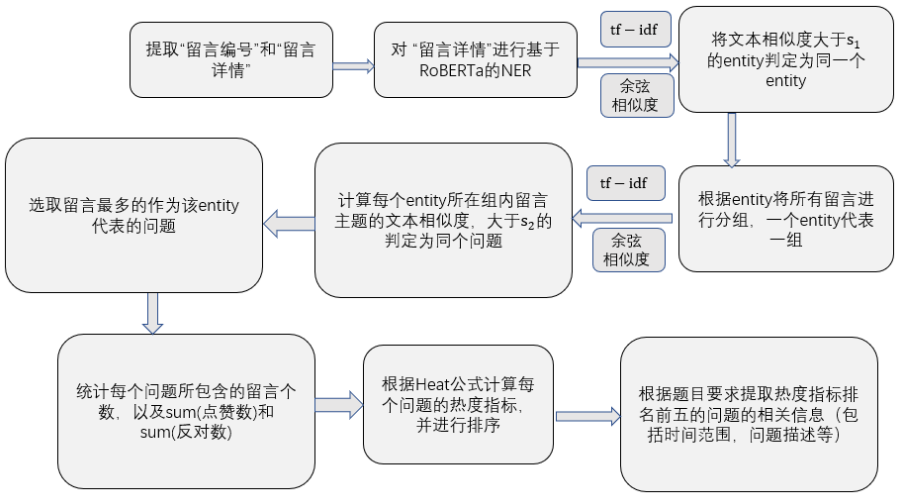

图 3.1 民生热点挖掘流程

3.2 基于 RoBERTa 的命名实体识别 NER

3.2.1 命名实体识别

命名实体识别（Named Entity Recognition，NER）指在文本中抽取具有特殊意义的实体如人名、地点、机构等。我们认为，热点民生话题具有一定的集中性，在群众留言里会集中反映特定的地点、组织机构、职位和人群等，因而挖掘热点的第一步是要实现文本的自动识别抽取地点和人物。

NER 在本质上是序列标注问题，其数据标注方式遵照序列标注问题的方式，我们采用 BMES 标注法。其含义为：

- B，Begin，开始位置

• M，Middle，中间位置
• E，End，结束位置
• S，Single，单字
如“星沙小区”这一地点名，其标注为：
星　B
沙　M
小　M
区　E

3.2.2 CLUENER 细粒度命名实体识别

CLUENER2020 是开源的中文命名实体识别数据集，不同于大多数数据集采用三种实体标注，该数据集实现了十个实体识别，能更好地区分特定实体。

数据的 10 个标签类别为：地址（address），书名（book），公司（company），游戏（game），政府（government），电影（movie），姓名（name），组织机构（organization），职位（position），景点（scene）。

数据分布为训练集 10748 个样本，验证集 1343 个样本。我们的附件 3 作为测试集，有 4326 个样本。

3.2.3 RoBERTa

RoBERTa（Robust Optimized BERT）于 2019 年由华盛顿大学提出，在原 BERT 上进行改进。经证明，RoBERTa 在命名实体识别上的效果要更优于 BERT。

RoBERTa 相较于原先的架构上做了以下三个改进：
（1）移除 NSP 损失，这使得下游任务的性能略有提升。
（2）动态 mask 机制，避免训练迭代时相同位置的 mask 重复多次。
（3）训练时采用更大的 batch size，BERT 通常设定 batch 大小为 256，而 RoBERTa 采用 2K 到 8K 不等的 batch。

3.2.4 训练

CLUENER 数据集以 json 文件存储数据，附件 3 留言详情预处理同 2.3.1，最终以 test.json 格式存储。包含留言 id 和留言详情。

实验环境以及训练过程同章节二相同。

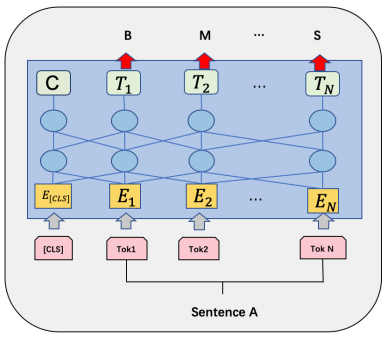

图 3.2 RoBERTa-NER 架构，与 BERT 基本类似

3.3 NER 结果处理

3.3.1 NER 任务结果文件数据预处理

NER 结果文件中每行对应一个留言，包含留言编号，以及留言详情中识别出的实体位置标识,提取每一行的留言编号和识别出的实体 entity，构成字典。

3.3.2 文件数据处理

提取文件每一行的留言编号，并且计算每个留言的（点赞数-反对数）。构造留言编号和（点赞数-反对数）的字典，留言编号和留言主题的字典，留言编号和留言时间的字典，以便于后续使用。

3.3.3 TF-IDF 算法

在计算文本相似度时，我们使用 TF-IDF 模型。

tf即词频$term\ frequency$，表示词w在文档d中出现次数$count(w,d)$和文档d中总词数$size(d)$的比值：

$$tf(w,d) = \frac{count(w,d)}{size(d)}$$

idf即逆向文档频率$inverse\ document\ frequency$，表示文档总数$n$与词$w$所出现文件数$docs(w,D)$比值的对数：

$$idf(w) = log(\frac{n}{docs(w,D)})$$

根据

$$weight(t,d) = (1 + logtf(w,d)) * idf(w)$$

作为权重来将每一篇文档表示为一个权重向量，并根据余弦相似度作为文档之间的相似度。

3.3.4 热度指数 Heat

将 NER 结果中的实体作为依据，对每条留言进行初步分组。考虑到现实生活中对于同一个地址，不同的人表述方式可能会有细微差别的问题，我们设置参数s_1作为判定两个实体是否为同一个地址的阈值，如果两个 address 的文本相似度大于s_1，则判定二者为同一个地址。

包含地址相同的留言初步划分为一组。考虑到组内，虽然地址相同，但是反映问题可能有多个的问题，我们设置参数s_2作为判定两个留言主题所反映的是否为同一个问题的阈值，如果两个留言主题的文本相似度大于s_2，则判定二者为同一个问题。

遍历计算每一个组内相互之间文本相似度大于s_2的留言个数最多的一类问题作为该组所在地址所代表的问题，并统计该问题所包含的留言的点赞数和反对数，该问题的热度指标公式如下:

$$Heat = c + \log\left(\sum_{i=1}^{c}(like_i - dislike_i)\right)$$

C 表示该问题包含的留言个数，$like_i$表示该问题中包含的留言i的点赞数，$dislike_i$表示相应的反对数。

根据$Heat$对所有的问题进行热度排序，热度指标高，代表该问题为热点问题。

3.3.4 结果生成

找出热度指标 Heat 排名前五的问题包含的全部留言编号，利用 list, dictionary, tuple 等数据将相应的留言按照问题顺序转移到“热点问题留言明细表”中。

统计 Heat 排名前五的问题的信息，比如时间范围，如果该问题的留言只有一个，则该留言的留言时间作为该问题的时间范围。选择该问题包含的留言列表中的第一个留言的留言主题作为该问题的问题描述。

热点问题表和热点详情表的部分结果分别如表 3.1 和表 3.2 所示。

热度排名	问题 ID	热度指数	时间范围	地点/人群	问题描述
1	1	6.133656	2019-11-18 至 2019-12-15	A 市暮云街道丽发新城社区	投诉小区附近搅拌站噪音…
2	2	5.822908	2019/1/2 至 2019/1/11	A 市经开区东六线以西	问问 A 市经开区东六线以…
3	3	5.752133	2019/8/19	A 市 A5 区汇金路五矿万境 K9 县 24 栋	A 市 A5 区汇金路五矿万境 K9 县存在一系…
4	4	5.642219	2019/4/11	梅溪湖金毛湾	反映 A 市金…
5	5	5.145567	2019/2/25	西地省政府	严惩 A 市 58…

表 3.1 热点问题表

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
1	255008	A909208	投诉小区附近搅拌站噪音…	2019-11-18 12:23:22	暮云街道丽发新城边上在…	0	0
1	266665	A00096279	投诉小区附近搅拌站噪音…	2019-12-04 17:23:22	开发商把特大型搅拌站，…	0	0
1	208285	A909205	投诉小区附近搅拌站噪音…	2019-12-15 12:32:11	尊敬的领导,我是 A 市暮…	24	0
1	261072	A909207	投诉小区附近搅拌站噪音…	2019-11-23 23:12:22	投诉 A 市暮云街道丽发新城…	9	2
2	233542	A00080329	问问 A 市经开区东六线以西…	2019/1/2 20:27:26	A 市经开区东六线以西，泉塘…	24	0

2	239670	A00080329	问问 A 市经开区东六线以西…	2019/1/11 15:46:04	A 市经开区东六线以西，泉塘…	41	0
…	…	…	…	…	…	…	…

表 3.2 热点详情表部分结果

四、答复质量评价模型

4.1 数据预处理

4.1.1

在附件四给出的答复意见中，在末尾都出现了年月日时间信息，考虑到表格中已经有具体答复时间，是无意义字符，因此处理数据时将其年月日时间信息删除。因为无意义字符的去除，在答复意见中出现了空值，但选择保留，设置为空值。考虑到在在后续处理中答复意见和留言主题 txt 文件行数一一对应的关系，去除答复意见文本中中文半角空格，中文换行符等符号，合成一段文本。

4.1.2 对答复意见进行中文分词

在对答复意见进行挖掘分析之前，为了方便非结构化的文本信息转换为计算机结构化语言，对答复意见进行了中文分词。采用 python 的中文分词包 jieba 进行分词。jieba 采用了基于前缀字典的实现的高效词图扫描，生成句子中汉字所有可能成词的有有向无环图（DAG），在进行中文分词的同时还去除了停用词，停用词表来自哈工大停用词表，而且添加了答复意见表中特定的表达信息量小且频率高的词汇，如网友，用户，答复如下等词汇。

4.1.3 对答复时间和留言时间

在评价答复质量之前，考虑到回复与留言时间差也是评价质量的一大指标，因此提取留言时间和答复时间，保留到年月日。根据最大时间最小时间差值计算每个等级的时间区间，评出等级。

4.2 提取答复意见主题

4.2.1 词袋模型

词袋模型（Bag-of-words model，BOW model）最早出现在自然语言处理和信息检索领域。该模型忽略掉文本的语法和语序等要素，将其仅仅看作是若干个词汇的集合，文档中每个单词的出现都是独立的。BoW 使用一组无序的单词来表达一段文字或一个文档。采用分词和去除停用词过后的文本用 Bow 模型形成稀疏向量，在用稀疏向量构建 docume-term 矩阵，形成 lda 模型的语料。

4.2.2 LDA 主题模型

通过 Bow 得到预料之后，我们就可以对答复内容进行主题提取，LDA 主题模型公式为：

$$p(\text{词语} \mid \text{文档}) = \sum_{\text{主题}} p(\text{词语} \mid \text{主题}) \times p(\text{主题} \mid \text{文档})$$

给定一系列文档，通过对文档进行分词，计算各个文档中每个单词的词频就可以得到左边这边”文档-词语”矩阵。主题模型就是通过左边这个矩阵进行训练，学习出右边两个矩阵。

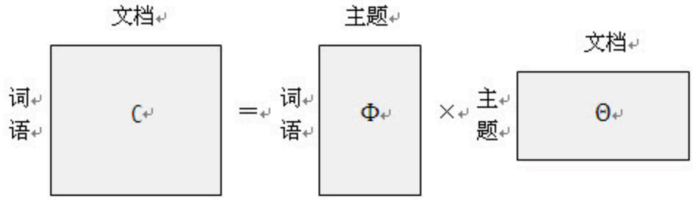

图 4.1 LDA 模型

计算文档-词汇矩阵，N 个文档组成的语料库（D_1，D_2，・・・・・・，D_n），由 V 个词组成的词汇表。矩阵中的值表示了词 W_j 在文档 D_i 中出现的频率，主题用 Z 表示，下面对语料库中的每一个 *word* 随机指派一个主题编号 Z_i，统计每个 Z_i 下出现的 *word* 次数，可得一个主题—词汇矩阵。

计算主题-词汇矩阵，此时可得出词分布 $\vec{\theta}$，几即每个词出现在该主题下概率 $\vec{\theta}_i$（$w_1 p_{1i}$，・・・・・・，$w_n : p_{1n}$）

$$p_{1i} = \frac{\text{每个} z_i \text{下出现的} word \text{次数}}{\text{该} z_i \text{下的 word 总数}}$$

此时还可得出每个主题属于每个词的概率分布 $\vec{w}_i$（$z_i : p_{2i}$，・・・・・・，$z_n : p_{2n}$）

$$p_{2i} = \frac{\text{每个} w_i \text{属于每个主题的次数}}{\text{该} w_i \text{的总个数}}$$

计算文档-主题矩阵，统计每个词代表的主题在每一个文档中出现的次数，可得出文档-主题矩阵

4.3 答复质量的总分

4.3.1 答复意见和留言主题相似度分数

bert 的输入表示（input representation）能够在一个 token 序列中明确地表示单个文本句子或一对文本句子，。对于给定 token，其输入表示通过对相应的 token、segment 和 position embeddings 进行求和来构造。因此采用 bert 模型的输入表示部分对留言主题和答复意见进行文档的向量表示，并且计算对应留言主题和答复意见的向量的余弦相似度，得到分数在[0,1]间的留言主题和答复意见相似度分数。

$$sim\_score_i = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

其中 A 为留言主题向量矩阵，B 为答复意见向量矩阵

4.3.2 答复意见完整性分数

答复意见的完整性要求能从各个方面较详细的解决留言中提出的民生问题，而往往文本的长度和解决问题的详细程度正相关，长度越长也意味着解决问题的角度越多。因此统计每个答复意见的长度，每个长度计算得分，通过归一化得到最后结果，计算公式如下：

$$dcom\_score_i = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}$$

其中x_i=ln^{l_i}，l_i为第 i 篇答复意见的长度。

4.3.3 回复及时性分数

在民生调查中发现业主等待回复的时间和业主的满意度成反比，时间越长，业主满意度越低，直接影响了答复意见质量，因此将回复时间减去留言时间得到回复时间间隔，得到回复时间间隔分布图，划分合理的时间等级，分为 1-5 级，级别越高回复越快分数越高。最后将等级也映射到[0,1]之间,公式如下：

$$re\_score_i = log^{level_i}$$

其中 $level_i$为每个答复时间的等级，log底数取 5

4.3.4 总评分

三个得分都已归一化，所以在结合三个因素时进行了较简单的加权和，公式如下:

$$score_i = \alpha sim\_score_i + \beta dcom\_score_i + \gamma re\_score_i$$

其中$\alpha + \beta + \gamma = 1$,尝试所有的权重后发现$\alpha = 0.5$，$\beta = 0.2$，$\gamma = 0.3$效果较好。

4.4 结果分析

对所有留言主题进行中文分词后统计每个词出现的的频率，生成词云，如图 4.2 所示：

图 4.2 留言主题词云

从词云中可以看出:反映给物业的留言主要是咨询相关事情，和小区问题反映，以及建设小区建议。其中 A5、A7、B9、A8 区小区问题较多较为突出，分析后得到大致原因为小区较老旧或者房产商不负责任导致房子和周围环境质量堪忧，这几个小区很多问题物业不及时解决，多次反映，所以问题聚集。留言的主要内容和民生息息相关，涉及教育、交通、住房等问题。

着重对问题较突出的 A5、A7、B9、A8 区小区答复意见分数进行分析。计算得到改几个所有答复意见得分的平均值，结果如表 4.1 所示：

区	答复意见平均得分
A5	0.78
A7	0.77
B9	0.74
A9	0.78
所有小区平均分：0.76	

表 4.1 答复意见表

通过计算可以看到所有问题较多小区都比较接近小区平均分，答复评分一般，答复质量不是很让人满意，这跟人的感性分析相同，答复质量评分具有一定合理性。关注答复时间和留言时间的差值，统计情况如图 4.3 所示。

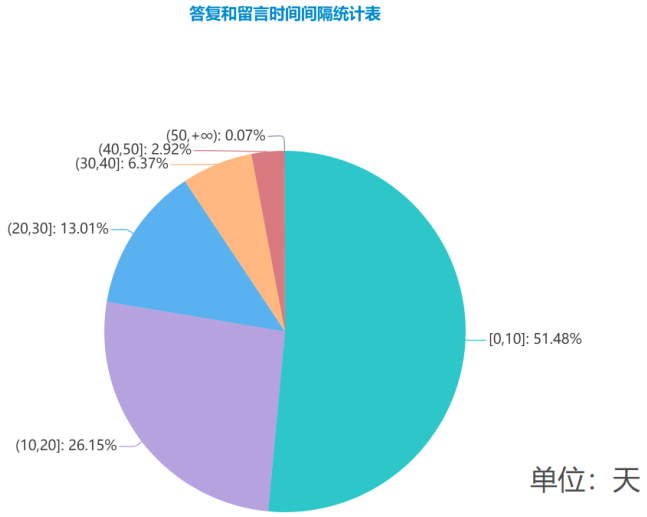

图 4.3 答复时间和留言时间差值

在[0,10]天就立即回复的物业比较多占比百分之五十，四十天内占大部分，因此以十天为单位的回复及时性等级划分具有合理性。通过图片可看出所有的物业还是比较重视小区问题的反映，及时作出答复。

六、参考文献

[1] 唐晓波，邱鑫. 面向主题的高质量评论挖掘模型研究[J]. 现代图书情报技术，2015, No.260,No.261(Z1):112-120.

[2] 王振振，何明，杜永萍. 基于 LDA 主题模型的文本相似度计算[J]. 计算机科学，2013，040(012):229-232.

[3] Chao G L , Lane I . BERT-DST: Scalable End-to-End Dialogue State Tracking with Bidirectional Encoder Representations from Transformer[C]// Interspeech 2019. 2019.

[4] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. [cs.CL] 2019

[5] Xu L , Tong Y , Dong Q , et al. CLUENER2020: Fine-grained Named Entity Recognition Dataset and Benchmark for Chinese[J]. 2020.