基于文本处理的涉疫地点自动化提取模型——疫知

摘要

新冠疫情背景下，涉疫地点信息的整合公开对于疫情防控而言至关重要。然而，涉疫地点数据多以非结构化的方式出现在疫情通报网页上，在以往这项工作主要通过人工标注实现，不仅消耗人力而且效率低下。近年来，自然语言处理作为人工智能的一个重要领域得到了飞速的发展。因此，本文通过比较不同的方法，构建了基于自然语言处理技术的涉疫地点标注标注模型，以解决这个问题。整个解题过程分为以下三个步骤：

第一步，将数据统一文本化，并基于规则抓取“来源”字段。我们对链接做一定的分析和预处理后，从链接中抽取网页正文、图片、附件等，并利用 OCR/Pandas 等多种技术工具将它们统一文本化。在得到正文信息后，我们对其做包括 html 元素清洗在内的一些的去噪处理，得到链接对应的文本数据。利用模板和规则的方案，我们可以提取到网站对应的数据来源字段。

第二步，迁移筛选符合我们问题的标注数据，构建基于 Bert-Bilstm-CRF 的地点识别模型。考虑到标注样本较小，我们面临的是一个小样本实体标注学习的问题。一方面，基于迁移学习的样本关系映射思想，我们想到了对已有标注数据集的地点标注字段进行抽取和粒度筛选，从而提高我们的标注数据量。利用腾讯 TexSmart 工具和字典模式规则，我们对现有通用数据标注集的地点字段进行了自动化粒度筛选，得到了符合我们问题标注准则的地点标注数据集。另一方面，引入预训练机制的中文语言模型，也使得我们进一步减少对数据量的依赖。基于该整合数据集和我们标注的小数据，我们构建了主流实体标注模型 BERT-BILSTM-CRF，在 50 个 epochs 后得到了 F1-Score 为 0.85 的结果。在经过包括频繁模式识别、误识别实体过滤等后处理方法的后，我们得到了更为精确的结果。

第三步，具体地点实体的行政区划归属匹配。我们提出了基于滑动窗口和字典匹配的方案，并通过引入路径加权算法提高了匹配适用度，完成了具体地点到行政区划匹配的任务。

实验最后分析并评估了涉疫地点自动化提取模型的综合表现，并给出了简要的未来计划：结合语义识别、全自动涉疫地点整合平台。

关键词：**自然语言处理、实体标注、预训练机制、迁移学习、滑动窗口**

目录

一、引言

2020 年初新冠疫情爆发，在抗击疫情的大背景下，数据的共享调配和处理是打赢疫情防控阻击战的关键一步。其中对涉疫地点数据的共享和整合，是为了进一步的疫情隔离与人口迁移管控。目前，大部分的涉疫地点信息由各地卫建委以及网络媒体发布，但这些信息都是分散在互联网上，为了进一步的聚合信息以实现信息的普及程度，需要对信息进行二次的聚合。以往，传统的整合方式是人工进行提取，这不仅消耗人力，而且存在效率低、时效差的问题。

实体识别在涉疫信息抽取的应用上为上述问题提供了解方案。近年来，自然语言处理（NLP）作为人工智能的一个重要领域得到了飞速发展，构建基于自然语言处理技术的涉疫地点信息自动化提取模型，通过端到端的处理技术，使得二次整合的过程全自动化，无需大量的人工干涉处理，直接将涉疫信息整合发布到平台上。

在实体标注领域，在有一定数据规模的情况下，深度学习方法能达到很好的效果。然而，在我们的问题上，目前缺少足够量的标注数据，也就是我们面临的是小样本学习的问题。我们将结合迁移学习的数据映射和预训练机制，来解决我们在数据量上的缺乏，提升涉疫地点提取的效果。

基于对疫情地点信息自动提取系统的理解和认识，本文将立足于以上的背景和问题，构建基于条件随机场（CRF）和预训练语言模型 BERT 的疫情地点识别模型，完成对目标信息的二次整合。

在完成对题目所给问题集的数据分析以及预处理工作后，我们对模型的时间开销、鲁棒性、实用性等方面作出综合分析。之后我们将该模型与其他主流方案相比，发现其在 F1-Score、准确率以及泛化能力上都表现出优越的效果。本文包括引言、系统模型、实验方案、实验结果、总结与展望五个部分。

二、模型框架

为了自动化地从新闻链接中抽取涉疫地点实体，我们提出了一种基于深度学习（DNN）和条件随机场（CRF）的特定粒度（行政区以下）地点实体抽取模型。该模型主要包括三个部分：数据分析与从链接到文本的预处理、地点实体的识别、地点实体行政区匹配。模型的架构图如图 1 所示

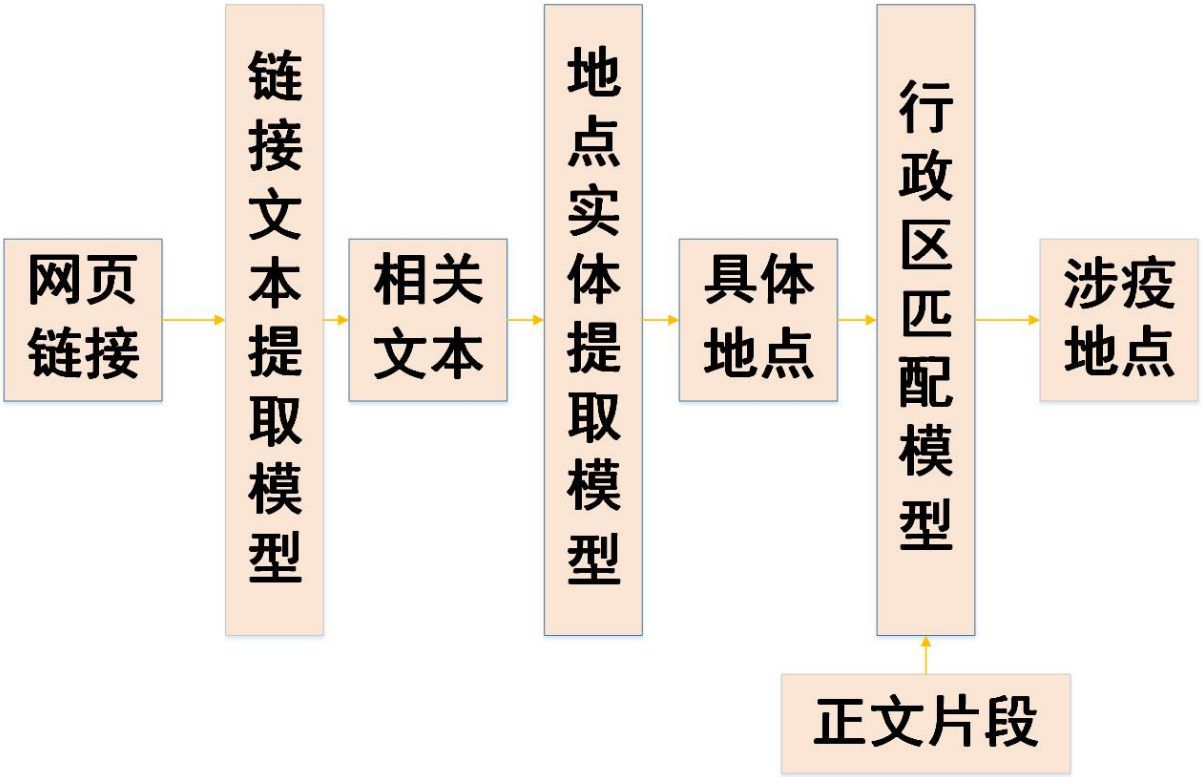

图 1 涉疫地点自动化提取模型框架

第一步：数据的分析与处理。我们对问题所给出的数据集进行统计分析，提出进行数据处理时的关键挑战，并给出相应的预处理步骤。

第二步，地点实体提取。运用迁移学习策略从大规模标注语料库中构建我们的标注数据集，摆脱数据量上的限制。引入预训练机制进一步摆脱对样本量的依赖。最后，基于整合迁移数据以及我们标注的小样本数据，我们构建了基于 BERT-BILSTM-CRF 的行政区粒度以下的地点实体识别模型。

第三步，地点实体行政区的匹配。根据从所得到的涉疫具体地点文本以及其所在的正文片段，我们采用了滑动窗口字典匹配和词向量相似度计算，构建我们的行政区匹配

模型，提取出了完整的地点实体二元组信息（行政区，具体位置）。进一步地，我们可以整合构建题目所求的六元组信息（省、市、行政区、具体位置、数据来源、链接）。

三、方案介绍

3.1 链接分析与预处理

高质量的数据集是模型匹配和优化的基础，对整个数据集进行分析处理可以促进对数据集的全面认知，从而更好地对数据进行特征工程编码表示，进一步提高数据集的质量。根据分析结果，更容易选择预处理阶段的相关参数，减少重复摸索的概率。[]

3.1.1 链接分析与预处理

根据问题所给的数据集，我们完成了数据集的分析工作，如图 2、表 1 所示。

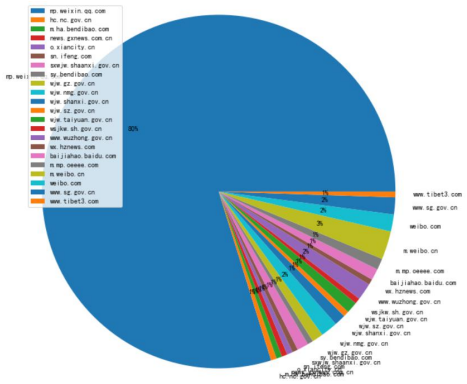

图 2 链接来源的统计分析

微信公众号	154
卫健委、政府	17
微博	8
网络新闻媒体	6

表 1. 链接来源分类统计

从图 2 的链接来源的统计分析，我们可以得到在其所给出的链接，大部分是微信公众号，数据来源主要包括政府卫健委以及新闻网络媒体。

对赛题所给出的链接数据集进行去重和去空之后，进一步对链接判断其是否能够正常访问，我们筛选得到了 185 条链接，同时，对于数据链接中的主要部分，我们对微信公众号的链接做了特别关注。对于有些微信公众号已经进行了公众号迁徙的情况，我们队也对其进行了特殊处理，将迁徙后的链接替换原本提供的链接。最终，我们得到了 185 条有效的疫情通报网页链接。

3.2 网页数据文本化提取

在这个模块中，我们设计了一套从链接到网页本的提取流程。一般情况下，涉疫地点信息会出现在正文部分的文本、图像中，有少部分的网站会将涉疫地点信息存在正文末尾的附件中。所以，我们疫情相关地点信息提取也是围绕这三个对象展开的提取和文本化。

首先是对正文部分的提取，正文部分通常包括大部分的地址实体，而非正文部分通常会包含有许多噪声信息，所以这部分抽取的准确率也将会比较显著地影响到我们后面地址识别实体的评分。

3.2.1 WEB 正文提取

在 WEB 正文提取领域，主要以下三种方法[1]：基于模板和规则的方法，这种方案对于需要抓取网页结构固定的情况已经够用了，但对于要抓取网络结构不定且难以穷举的情况，这种方法显得有些无力；基于统计和特征的方法，这类方法利用了正文所在区域的一些统计特征，比如，正文所在区域的文本密度比较大，符号密度比较小，正文所在区域的会位于网页视觉结构中的主要位置等。这类方法具有普遍意义，提取适用对象范围有了质的飞跃，但是每种方法基本上都有其特定的缺陷。基于机器学习的方法，结合大数据和已有的特征分析方法，这种方法能达到了最好的效果，但缺点是需要比较大的数据量。

在本文中，考虑到所给出的链接大部分是微信公众号链接，具有固定的网页结构，我们首先会采用基于模板和规则的方案，通过构建正则表达式和 CSS 选择器，我们就可以轻松完成数据集中大部分信息的正文提取。我们所构建的模板库和 CSS 选择器如下表 2 所示。

网站	来源	正文
南方都市报	.name	.docContent
微信公众号	#js_name	#js_content
手机版微博	.weibo-author	. weibo-text
电脑版微博	. WB_text.nick-name	. WB_text

表 2 模板库中几个网站的 CSS 选择器

对于其他链接，出于准确率的考量，我们首选采用基于视觉分块和机器学习技术的方案[2][3]，通过调用业界领先 DIFFBOT 公司提供的文本提取 API[17]提取链接中的正文，在这种方法由于一些网络因素发生失效的时候，我们会采用基于文本密度和符号密度统计的正文提取方案[4]，通过调用基于此方案的开源工具 GNE[18]，完成正文的提取。

在提取得到网页正文部分之后，我们就可以很轻松地拿到对应的正文文本、正文图片链接。对于图片链接数据，通过调用百度 AI 的 OCRAPI 进行图片中文本信息的提取。对于少量有附件的网址，考虑到其数量极少，在官网给出的数据集中只出现了一例，所以我们对其做特殊处理，人工抽取其中的文本。

3.2.2 网页信息文本化框架

最后我们将这几类抽取出的文本整合在一起，存入到数据表中。综上，我们所设计的网页信息文本提取框架如图下所示：

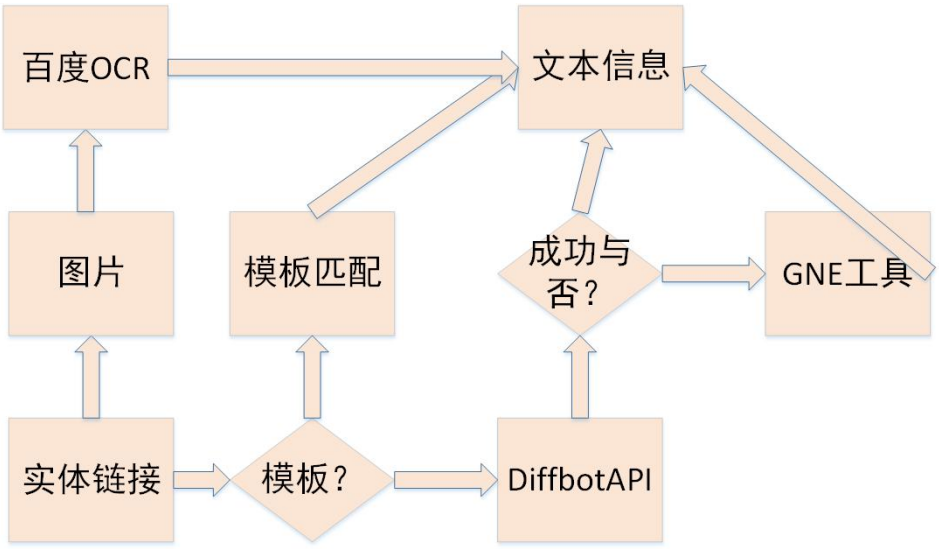

图 3 文本提取框架

3.3 基于深度学习的地点实体标注

3.3.1 实体标注

命名实体识别[5]（Named Entity Recognition，简称 NER），又称作“专名识别”，是指识别文本中具有特定意义的实体，主要包括人名、地名、机构名、专有名词等。简单的讲，就是识别自然文本中的实体指称的边界和类别。

实体标注的方法的发展可以被分作 4 个阶段[6]。阶段 1，早期阶段，主要是基于规则的方法、基于字典的方法等。阶段 2，传统机器学习阶段，代表方法有 HMM、MEMM、CRF。阶段 3，大数据驱动下的深度学习方法，如：RNN－CRF、CNN－CRF。阶段 4，近期出现的一些方法，如：注意力模型、迁移学习、半监督学习的方法

按照学习模式分，可以分为：有监督的 NER、半监督的 NER、无监督的 NER、混合方法[7]。

值得一提的是，深度学习在实体命名识别上表现出了非常不错的效果，此类方法把命名实体识别看作是序列标注来做，比较经典的方法是 LSTM+CRF、BiLSTM+CRF。

在本文中，我们采用了效果较为突出的深度学习方法作为我们的实体识别方案。

3.3.2 基于粒度筛选的数据集迁移

使用深度学习的方法，我们无法绕开数据的问题。鉴于官网给出的数据仅为测试用例，所以我们需要去找更多的训练数据，以使得我们的实体标注模型达到比较好的效果。调研现有的实体标注数据集中，已经包含有与我们问题相近的字段——地址，但是它们标注准则较粗，需要进行一定的筛选后方可使用。

在我们的问题上，我们的实体标注准则是：根据[]的行政区划等级分层，我们需要 4 级、5 级以上的行政区划地点。行政区粒度以下（**街**号，**路，**街道，**村等）（如单独出现也标记），地址标记尽量完全的，标记到最细。对于行政区粒度及以上的实体（**省**市**区）在这一阶段不希望模型识别出来，所以我们的准则是不标注这些实体。

基于迁移学习的样本关系映射思想[8]，我们通过基于字典的粒度筛选，我们构建了源数据与目的数据的映射机制，得到了符合我们标注准则的地址标注数据集。

我们选择了几个大规模实体识别标注数据集：boson 数据集、人民日报 2004 标注数据集，NERCLUE2020 细粒度实体数据集，并对他们进行了整合，抽出了其中的包含地点实体的所有数据。利用字典规则的筛选和腾讯的细粒度实体识别工具 TexSmart[19]，我们筛选了粒度在行政区以下的地址实体，并且组合成我们特定问题的标注数据集，其中包含 2368 条句子，地址实体 2829 条。

3.3.3 条件随机场

条件随机场（CRF）[9]是给定一组输入随机变量条件下另一组输出随机变量的条件概率分布模型，其特点是假设输出随机变量构成马尔科夫随机场。在实体标注中，CRF 是一种非常经典的方法。

条件随机场使用势函数和图结构上的团来定义条件概率 $P(Y \mid X)$。给定观测序列 x，链式条件随机场主要包含两种关于标记变量的团，即单个标记变量 y_i 以及相邻的标记变量 (y_{i-1}, y_i)。在条件随机场中，通过选用合适的势函数，并引入特征函数，可以得到条件概率的定义：

$$\mathrm{P(y \mid x)} = \frac{1}{Z} \exp\left(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} u_l s_l(y_i, x, i)\right)$$

$$\mathrm{Z} = \sum_{y} \exp\left(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} u_l s_l(y_i, x, i)\right)$$

已知训练数据集，由此可知经验概率分布 $\widetilde{P}(X,Y)$，可以通过极大化训练数据的对数似然函数来求模型参数。加入惩罚项后，训练数据的对数似然函数为：

$$\mathrm{L(w)} = \sum_{x,y}\left[\tilde{P}(x,y)\sum_{k=1}^{K} w_k f_k(y,x) - \tilde{P}(x,y) log Z_w(x)\right] - \sum_{k=1}^{K} \frac{w_k^2}{2\sigma^2}$$

其中的 σ 是可以调节的惩罚权重。对似然函数 $L(w)$ 中的 w 求偏导，令：$\frac{\delta L(w)}{\delta w_i} = 0$，可以依次求出 w_i。

在我们的方案中，CRF 作为模型最后一层输出层，用以对标注序列的转移矩阵进行约束，防止 DNN 网络过拟合。

3.3.4 预训练语言模型

预训练语言模型和下游任务模型分离的引入是近几年自然语言处理领域的一大突破[10]。预训练思想的本质是模型参数不再是随机初始化，而是通过一些任务（如语言模型）进行预训练，预训练属于迁移学习的范畴。通过使用大规模文本语料库进行预训练，对特定任务的小数据集微调，降低单个 NLP 任务的难度。

BERT 是预训练语言模型中的一大里程碑[11]，全称是 Bidirectional Encoder Representation from Transformers，使用双向 Transformer 的 Encoder。其使用 Masked Language Model 和 Next Sentence Prediction 作为预训练任务，两种任务使得模型能够捕捉词语和句子级别的语义，这两种通用特征的抽取能力使得它在处理下游任务时更加游刃有余。

在后续实验中发现，引入预训练语言模型，我们的模型能够一定程度减少对数据量的依赖程度，使得我们即使使用比较少的样本也可以达到较为满意的效果。

3.3.5 模型设计

我们设计了 BERT+BiLSTM+CRF 的网络结构，预训练模型采用了 12 层 BERT 模型。以下为我们的模型结构示意图：

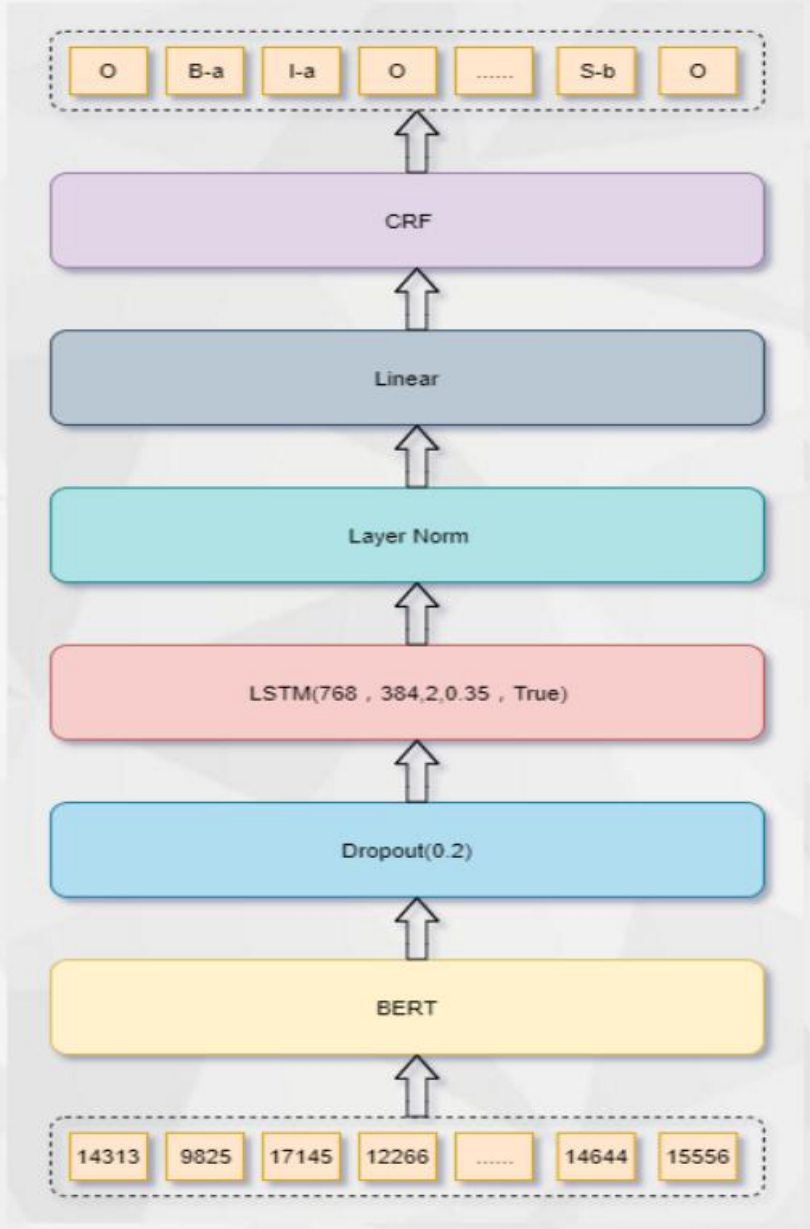

图 4 模型结构示意图

对于语言模型层，我们在收集的疫情报道语料库进行了进一步的预训练，使得语言模型能够更熟悉我们问题的文本环境。

该模型分为 18 层，前 15 层为 BERT 的 101,360,640 个不可训练参数，后面三层的训练参数为 3,295,572，总计 104,656,212 个参数。如下为每层的详细介绍：

1 INPUT: 从输入文本中抽取 token 序列、segment 序列，长度文本最长为 100 字。

2 Embedding: 从输入文本中抽取三个嵌入特征：WordPiece 嵌入、位置嵌入、分割嵌入，再将他们进行单位和整合为长度为 512 的编码向量，输入进 BERT。

3 Embedding-Dropout/Norm:对词嵌入向量进行 dropout，能够提升模型的泛化能力；进行标准化操作，能够提升模型的训练速度和稳定性。

4~15 Transformer*12：Bert 模型的 decoder，采用堆叠的 12 层 Transformer，能够有较强的特征提取能力。

16 Bilstm:双向长短时循环神经网络，能够在文本的两个方向都抽取相应的特征，在前面层抽取的通用特征上在做进一步的特征抽取。

17 Dense 层:全连接神经网络层,对 bilstm 提取的 100x256 的向量降维降至 100x64，降低后续条件随机场层的复杂度。

18 CRF 层：条件随机场层，将 100x64 的向量映射到 100x4 的标注向量，得到 4 类标注结果（O/B-LOC/I-LOC/<PAD>）。

上述流程的本质是输入序列的多分类。输入序列经过预训练的语言模型 bert 抽取通用特征后，再用 bilstm 做下游任务序列输入特征的抽取，CRF 来抽取输出序列的约束关系，最后模型输出一条概率最大的标注序列。

3.3.6 后处理

通过对数据集的观察和实验结果的反馈，我们主要采用了两种后处理方案：频繁模式挖掘、误识别实体过滤。

频繁模式挖掘：文本数据集里有一些频繁出现的模式，比如，以标点分割的实体串。通过模式匹配，挖掘出这部分遗漏的实体。

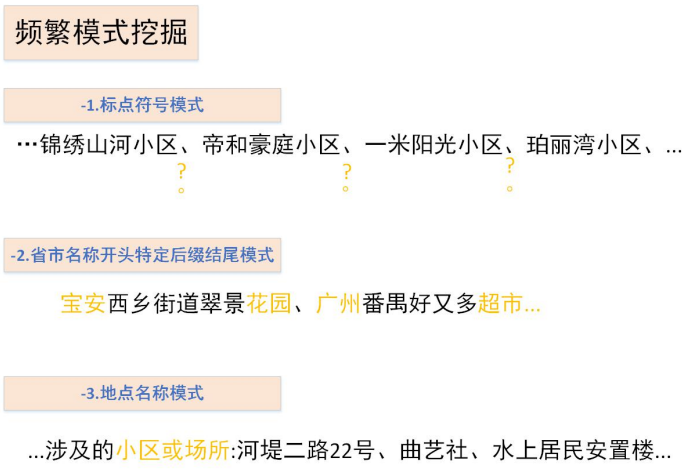

图 5 后处理-频繁模式挖掘的集中模式

误识别实体识别：我们会根据预测标签判断，一部分直接丢掉，一部分根据前后缀补全。

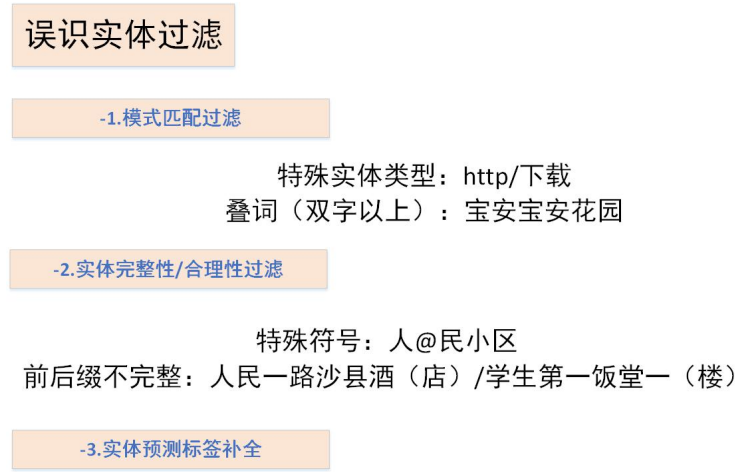

图 6 后处理-误识别实体识别的几种模式

3.4 实体行政区归属匹配

在得到涉疫的具体地点实体集合后，我们要对起行政区划归属进行匹配。我们提出了基于滑动窗口的方法，实现了地点行政区划归属的抽取。

3.4.1 基于滑动窗口地点行政区匹配

基于现有的地点到行政区匹配的算法[12]，我们做了一定的简化和改进。通过所提供的数据，我们构建了地点-行政区层次化知识树，如下图所示。

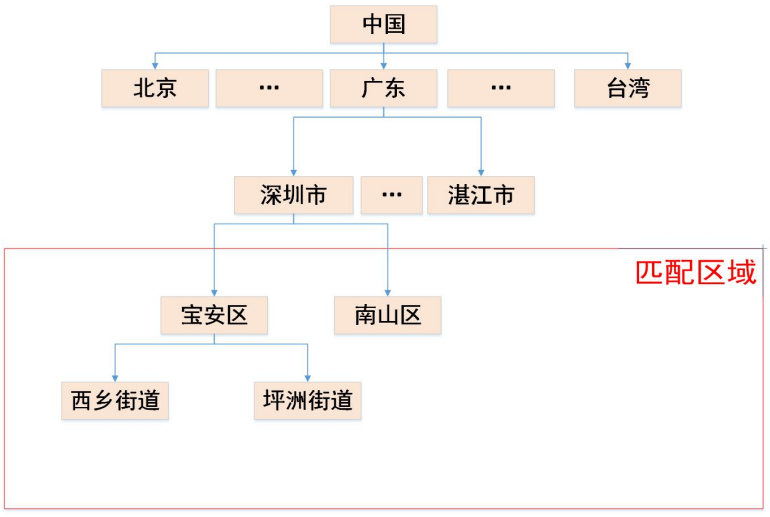

图 7 行政区划匹配模板树

首先，我们根据具体地点文本所在源文本的位置确定一个初始滑动窗口，然后对其进行滑动，直到窗口中包含有出现行政区实体模式。原方法中是根据具体地名对地点进

行索引，但考虑到地名大量的重复性和高质量数据集的缺乏，我们的改进是，仅将 3～5 级（区及街道、村以下）的字段作为匹配字段。

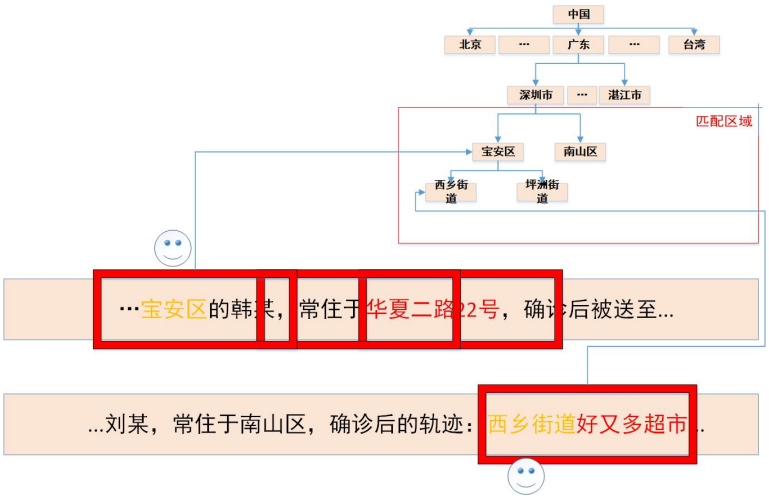

图 8 滑动窗口匹配地址实体对应行政区

当出现重名（对应行政区划树中有多条路径）的情况时，我们将会依据地点信息所在的局部上下文及上一地点行政区划归属这两个信息对不同路径进行权重加权，最终得到实体的行政区划。如图下所示：

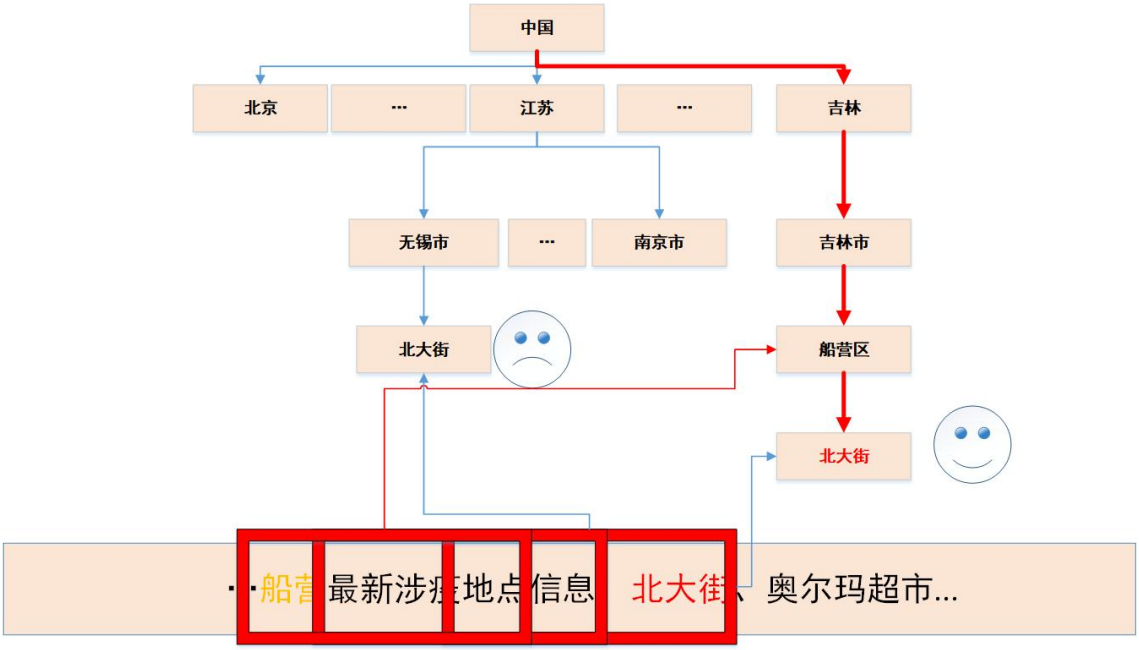

图 9 重名情况下路径加权选择最优路径

四、实验结果

4.1 实验环境

在我们的模型验证过程中，我们使用了 Colab 所提供的虚拟机，主要基于 Ubuntu 18.04.3 LTS 的操作系统，实验环境为 12.72 G 的内存容量，69 GB 的硬盘容量，Tesla T4 的 GPU，主要以 Python 为开发语言，并基于 Keras / Pytorch 开发框架完成模型的构建。

4.2 评价指标

本模型采用的评价指标主要包括准确率、召回率以及 F1-Score。

$$F1-Score=\frac{\mathrm{Pr}ecision\cdot\mathrm{Re}call}{\mathrm{Pr}ecision+\mathrm{Re}call}$$

$$\mathrm{Pr}ecision=\frac{TP}{TP+FP}$$

$$\mathrm{Re}call=\frac{TP}{TP+FN}$$

4.3 实验结果

4.3.1 网页文本抽取阶段实验结果

在预处理得到的 186 条链接上，利用上文我们构建的文本提取框架，我们一共提取 502430 字的文本。到了我们提取到的文本长度统计如下表、图所示：

	最小值	最大值	平均值	标准差
正文文本	124	13047	2242	2573
图片文本	0	7062	458	843
整合文本	205	13047	2700	2784

表 3 .提取文本字符长度的统计分析

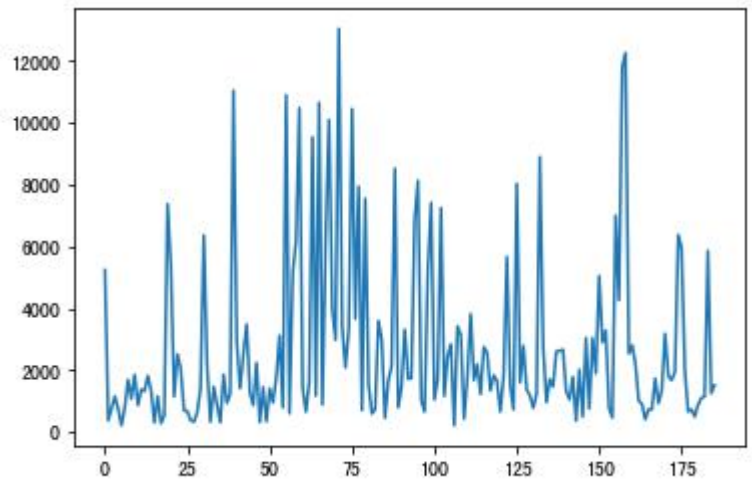

图 10 提取文本字符长度的统计图

从表中可以看到，正文普遍长度较长，平均有 2242 个字符，图片的 OCR 文本相对而言较少，平均有 458 个字符，通过拼接整合后，我们得到了 url-文本的数据表。

对正文文本的抽取结果进行手工抽验检查，我们抽取了三类比较具有代表性的样本：1 对模板对应的网站推文（微信公众号推文、南方都市报推文、新浪微博推文等）；2 政府卫检委网站推文（链接中有 gov 字段、此类我们没有设置模板）。3 不在模板库中的网络媒体推文（百度百家号、本地新闻等）

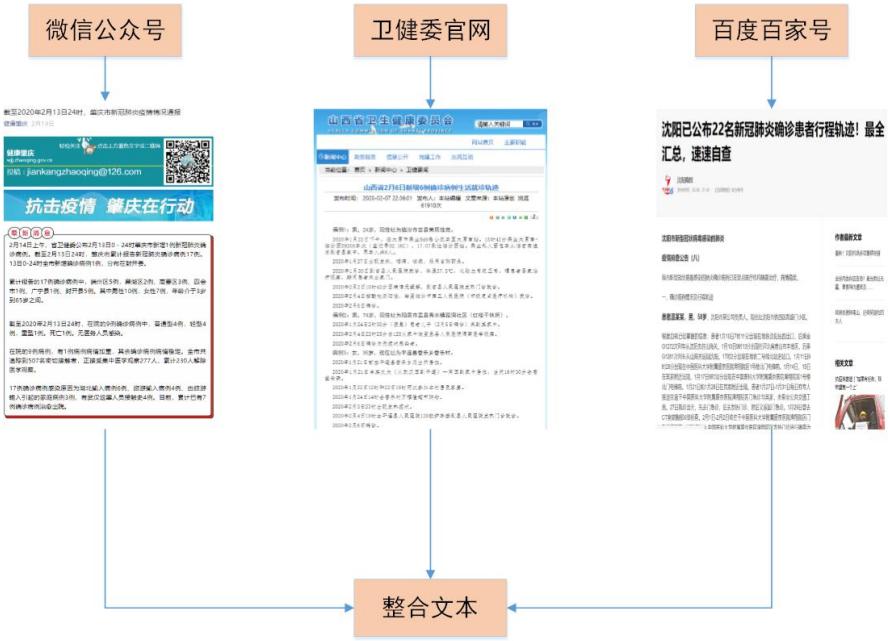

图 11 实验检验抽样案例

实验结果：对于第一类样本，我们能够 100%地抓取图片和正文中的文本信息；对于第二、三类样本，我们的抓取精度能够达到 98%左右，存在一些样本会出现评论区反客为主的现象，影响抓取效果。总体而言，正文抓取效果是非常出色的。

4.3.2 数据迁移收集阶段实验结果

在这一阶段，我们主要基于最近 CLUENER2020[17]所提供的细粒度实体数据集，并整合了其他大规模语料中粒度符合的地点字段，最后我们得到了其中包含 2368 条句子，地址实体 2829 条。进行划分后，训练集 2095 条，验证集 273 条，样本规模相对较小，后续我们会进一步处理这个问题。

4.3.3 地点实体识别阶段实验结果

为了更准确、全面地评价我们的地点识别效果，我们一方面标注了小样本的地点标注数据作为测试数据，另一方面，直观看到模型识别效果，手工筛选了以下两个不同的典型测试用例：卫健委官网文本、微信公众号推文文本。

对于我们手工标注的测试小样本，我们的实验结果如下所示：

	precision	Recall	F1-score	support
LOC	0.82	0.94	0.85	93

表 4 在抽样测试小样本上的实体识别结果

可以看出，评分已经达到了比较高的水平。为了直观得看到我们的抽取效果，我们展示出两个测试用例，链接见附录，以下为模型测试效果。

测试问题 1：

山西卫健委在 2 月 6 日疫情报道推文[13]，推文中有 17 个地点实体。

模型输出：

```
1.  '太 原 南',
2. '临 汾 西 站',
3. '太 原 南 站',
4. '吉 县 人 民 医 院',
5. '阳 泉 市 盂 县 秀 水 镇 路 洞 社 区 （ 红 楼 干 休 所',
6. '香 乐 村 万 福 隆 超 市',
7. '平 遥 县 香 乐 乡 香 乐 村',
8. '尖 草 坪 区 汇 丰 街 道 兴 华 北 小 区',
9. '吉 县 人 民 医 院',
10. '太 原 市 第 四 人 民 医 院',
11. '孟 县 人 民 医 院',
12. '山 西 万 峰 医 院',
13. '临 汾 市 吉 县 荣 辰 佳 苑',
14. '尖 草 坪 区 汇 丰 街 道 兴 华 北 小 区',
15. '沁 水 县 郑 庄 镇 中 乡 村',
16. '临 吉 高 速',
17. '吉 县',
18. '平 遥 县 人 民 医 院',
19. '县 人 民 医 院 发 热 门 诊',
20. '平 遥 县 香 乐 乡 派 出 所'
```

结果评估：

准确率：80%

召回率：94%

F1：86%

测试问题 2：
微信公众号“上海发布”2 月 10 日疫情推文[14]，含含 5 个地点实体。
模型输出：

```
'尚 海 湾 豪 庭',
'市 卫 健 委',
'四 季 绿 城 （ 北 区 ）',
'健 委',
'虹 桥 花 苑',
'城 市 超 市',
'虹 梅 路 店',
'虹 六 菜 市 场'
```

结果评估：

准确率：62.5%
召回率：100%
F1：76.9%

由以上识别结果可以看出，模型的识别召回率已经足够高了,而精确率有待进一步提高。在测试问题 1 中，我们发现了有许多医院实体，考虑到一般医院不作为涉疫地点标注，我们在后处理中会通过模式匹配将之删去。在测试 2 中，我们发现模型还识别了“市卫健委”、“建委”这样的实体，这一类的实体比较难进行统一的后处理，这提示我们在模型和数据上去进行改进。

在进行完基于频繁模式挖掘和误识实体过滤的后处理后，我们得到地点实体的数量增幅 5.2%，修复了 3%的地点实体，说明我们后处理规则的有效性。

4.3.4 行政区匹配阶段实验结果

在这一阶段，我们的实验结果表明，96%的实体都可以通过我们前文提出的框架完成行政区的匹配，对于剩下少部分的实体无法得到匹配，我们推测是有可能是上下文的行政区划关键词被拼写错等。例如，“宝安区阳光小区”被拼错为“包安区阳光小区”，在这种情况下，我们基于字典的原始方法会失效。对于这一部分实体，我们采用手工后处理。

五、总结和展望

5.1 总结

为了满足疫情地点信息的自动化标注提取，本文基于自然语言处理的相关理论和实验，构建了一个从链接到涉疫地点的自动化提取模型，让涉疫地点信息的整合更加智能化，减少人力劳动。模型主要在分为预处理及文本提取、具体地点实体识别以及具体地点行政区匹配三个阶段，在对赛题研究的基础上，我们根据研究思路撰写本论文，通过实验验证了本模型的可行性，基本实现本赛题设立的目标。

5.2 展望

5.2.1 结合语义细化识别

我们的模型有个比较强的假设，那便是输入进来的网页正文中出现的行政区以下的地点都是我们要捕捉的实体。在官网给出的问题集上的大部分文本，我们这个假设是基本成立的，但是还是存在一些反例，例如下面这种情况。

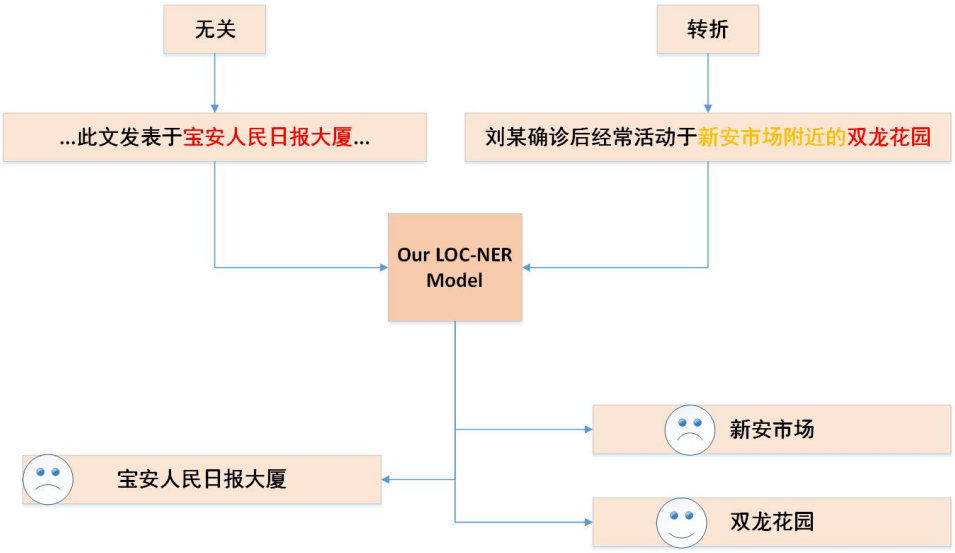

图 12 两种误识别的情况

这时，我们知道涉疫地点应该只有一个，把数据直接丢入我们的模型，由于我们的模型没有考虑到这点，就会误识别为两个实体。

更进一步的工作，应该要包括语义识别以及构建相关的数据集，以解决这种情况。

5.2.2 全自动涉疫地点整合平台

调研了目前的几个涉疫地点公布整合平台[19]，我们发现目前他们整合方式大部分靠手工，并且信息发布消息都是 2 月左右的，缺乏时效性，更进一步的工作，我们希望构建一套从抓取最新疫情通报链接到涉疫地点标注再到涉疫地点公示共享的全自动化信息共享平台。

六、参考文献

[1] 陈蕾蕾, 张如静. 面向 Web 的新闻网页正文信息抽取策略研究[J]. 电脑知识与技术, 2008, 2.
[2] 王雪梅, 陈兴蜀, 王海舟, 等. 基于标签和分块特征的新闻网页关键信息自动抽取[J]. 山东大学学报: 理学版, 2019, 54(3): 67-74.
[3] 谢方立. 基于节点类型标注的网页主题信息提取技术研究[D]. 北京: 中国农业科学院, 2016.
[4] 洪鸿辉,丁世涛,黄傲,郭致远.基于文本及符号密度的网页正文提取方法[J].电子设计工程,2019,27(08):133-137.
[5] 命名实体识别 https://en.wikipedia.org/wiki/Named-entity_recognition
[6] 一文看懂命名实体识别 https://easyai.tech/ai-definition/ner/
[7] 《*统计自然语言处理*》 宗成庆
[8] 小标注数据量下自然语言处理实战经验 https://www.cnblogs.com/jfdwd/p/11201389.html
[9] 《统计学习方法》 李航
[10] nlp 中的预训练语言模型总结 https://zhuanlan.zhihu.com/p/76912493
[11] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding https://arxiv.org/pdf/1810.04805.pdf
[12] 基于滑动窗口最大匹配算法的地址匹配方法 http://kns.cnki.net/kcms/detail/detail.aspx?filename=CN104615782A&dbcode=SCPD&v=
[13] 测试用例 1 http://wjw.shanxi.gov.cn/wjywl02/24757.hrh
[14] 测试用例 2 https://mp.weixin.qq.com/s/iWJ6omQpWP3YwGkjPANq-Q
[15] 疫情地点共享平台调用及数据搜集 https://zhuanlan.zhihu.com/p/139339580
[16] NERCLUE2020 数据集 https://github.com/CLUEbenchmark/CLUENER2020
[17] DIFFBOT ACTICLE EXTRACT API https://www.diffbot.com/
[18] GNE 网页正文提取工具包 https://pypi.org/project/gne/
[19] 腾讯 AI Lab 开放文本理解系统 TexSmart https://texsmart.qq.com