一种面向多需求的PDF文档信息抽取方法

余厚金 毛先领* 黄河燕

北京理工大学计算机学院，北京 100081
yuhoujin@outlook.com {maoxl,hhy}@bit.edu.cn

摘要. 目前，PDF已成为电子文档发行和数字化信息传播的一个标准，其广泛应用于学术界的交流以及各类公告的发行。如何从非结构化的PDF文档中抽取结构化数据是知识图谱领域所面临的一大挑战。本文利用Adobe公司开发的 Acrobat DC SDK对PDF进行格式转换，从半结构化的中间文件进行信息抽取。相比已有方法，Acrobat导出的中间文件保存了更完整更准确的表格和文本段落信息，能应用于不同需求的信息抽取任务。在CCKS 2019公众公司公告评测中，我们的方法位列总成绩第三名。

关键字： PDF, Acrobat DC SDK, 信息抽取.

1 引言

便携文档格式（Portable Document Format，PDF）是Adobe公司创建的一种主要用于电子初版的文件规范系统。由于PDF文档具有良好的显示效果、跨平台性和文档信息的独立性以及较好的安全性，所以当前互联网中的大多数科技论文，各类公告，采用PDF文档作为存储格式[1]。如何挖掘这些PDF文档中信息，基于内容进行信息抽取是知识图谱领域所面临的一大挑战。针对这个问题，CCKS 2019举行了评测任务——公众公司公告信息抽取。该评测主要目标是针对公告文件（PDF格式）中的信息抽取。

对PDF文档进行信息抽取一般分两个阶段：（1）内容抽取。PDF文档中的文本大致可分为两类：文本段落和文字流表格。由于PDF格式是面向显示的，本身缺乏其内容的结构化信息，使得对PDF进行自动化内容抽取变得十分困难，这将直接影响信息抽取的性能。因此，如何准确地保留PDF文档中的显示出的结构化信息，如段落、表格等，并减少噪声成为一个关键问题。（2）信息抽取。文本段落信息抽取和传统信息抽取任务大致相同，不同之处在于表格内容信息抽取。要挖掘表格中的信息，除了表格本身，还需结合表格前后的上下文。

传统的PDF格式文档内容抽取一般都是通过人工方式，这种方法只适用于小规模文档集的处理。随着文档集的增大，该方法效率太低，因此利用软件自

* 通讯作者

动化提取成为主要趋势。现有的PDF文本内容抽取的开源软件很多，但大多数软件存在以下问题：（1）通用性较低。这些软件大多只能识别文字，而忽略了文字的位置信息，并且这些软件只能针对某一种信息进行抽取，无法适用于不同需求的信息抽取任务。（2）提取效果差。PDF文档一般包含文本、图片和表格等内容以及页眉、页脚和脚注带来的噪声。要较好地从PDF文档中原样抽取内容变得十分困难。目前的开源软件提取效果千差万别，离落地生产还差很远。（3）解析速度慢。解析上百页的复杂PDF文档，这些软件往往需要一分钟甚至更久的时间，无法处理大规模文档集。

为了解决上述问题，本文利用Acrobat DC SDK，首先对PDF文档进行内容抽取。可根据需要，选择转换的格式，如XML，HTML，TXT和Excel等。转换得到的中间文件，不同程度保留了原PDF文档表格和文本段落信息。利用这些半结构化信息，可按需提取PDF文档中的内容，然后信息抽取。以上几种格式，均有成熟的解析库，实现内容抽取并不困难。

值得一提的是，尽管Acrobat DC SDK是付费软件，但其性能优越，目前已有的开源软件很难达到这样的性能。另一方面，Adobe公司提供Adobe PDF Library SDK，该SDK支持C++，C#，.NET和Java接口，可部署到不同平台的服务器上[2]。

在本次评测中，我们将公告文件（PDF格式）转换成XML，XML文件对文字流表格和文本段落进行了标记。对于任务一，我们使用BeautifulSoup4[1]查找<Table>标签，获取PDF中所有的表格；然后根据表格的上下文，确定其名称，抽出符合条件的表格。对于任务二，我们使用BeautifulSoup4抽出所有文本段落并分句，利用Bi-LSTM-CRF进行命名实体识别，然后结合规则抽取信息点。评测结果显示，我们的方法位列总成绩第三。

注意：由于PDF内容抽取参数设置不当，在表格抽取任务中，有1个测试用例输出为空（共10个用例），这影响了我们在这个任务中的表现，本来F1值可以达到0.96左右（理论值可达到0.99），这个分数能在该项任务中排名第三。实际评测结果为0.887，排名第五。

下面，我们将介绍如何使用Acrobat DC SDK对PDF进行内容抽取。

2 基于Acrobat DC SDK的PDF内容抽取系统

2.1 Acrobat DC SDK简介

Acrobat DC SDK是一组工具，可帮助开发与Acrobat技术交互的软件。 SDK包含头文件，类型库，简单实用程序，示例代码和文档。可以通过以下几种方式开发与Acrobat和Acrobat Reader集成的软件：（1）JavaScript：在单个PDF文档或外部编写脚本，以扩展Acrobat或Acrobat Reader的功能。（2）插件：创建动态链接并扩展Acrobat或Acrobat Reader功能的插件。（3）交互式通信：编写一个单独的应用程序进程，使用应用程序间通信来控制Acrobat功能。 Acrobat DC SDK支持Windows和Apple Mac OS环境中的开发 [2]。

[1] https://pypi.org/project/beautifulsoup4

Acrobat DC SDK提供以下功能：提取内容（只能提取文本段落），导出，创建和操作表单，提供搜索和索引等。本次评测主要使用导出功能，利用中间文件抽取内容。

2.2 实验设置

下面介绍开发环境与生产环境。

我们在Windows 10上进行项目开发，系统分为转换器和控制器。转换器使用C#编写，通过调用Acrobat DC SDK的导出功能API，把PDF转换成设定的格式。控制器使用Python编写，用于控制转换时间、格式等参数，以及异常处理。完成开发后，我们将系统部署到阿里云服务器上，操作系统是Windows Server 2016，Web API采用Flask框架。

本项目已经开源，可从网上下载[2]。

2.3 PDF内容抽取

图1是系统架构图，分为转换器和控制器两部分。

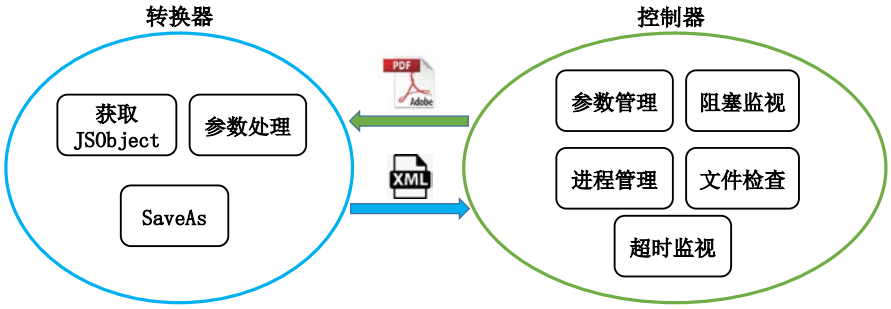

图 1. 基于Acrobat DC SDK的PDF内容抽取系统架构图。左侧为转换器，包含3个模块，是PDF内容抽取的核心。右侧为控制器，包含5个模块，用于控制整个格式转换过程。解析中间文件（XML等），即可抽取PDF文档的内容。

转换器接受由控制器传入的参数，使用Acrobat库中AcroAVDoc获取一个PDF文件对象，然后通过该文件对象并获取JSObject。最后构造参数列表，通过反射机制，调用SaveAs函数进行格式转换。

控制器接受包含PDF文件路径信息、转换格式和转换时间上限等参数，并通过管道将参数传给转换器。

*进程管理模块：*由于Acrobat DC SDK不支持并行，为了防止转换失败以及Adobe Acrobat DC打开过多PDF文件，必须在转换成功或者失败时杀掉Adobe Acrobat DC进程，进程管理模块用于杀掉相关进程防止对后续转换产生影响。

[2] https://github.com/houking-can/pdf-converter

阻塞监视模块：由于某些PDF文档可能不规范，转换过程可能会出现弹窗，需要点击确认按钮，这会阻塞转换器进程，阻塞监视通过调用win32api，监视相应窗口，一旦检测到窗口立刻发送鼠标左键点击事件，模拟点击确认动作，解除阻塞。

文件检查：转换一旦开始，就会生成相应的文件（实际上还未完成转换），需要不断检查该文件是否完整，文件检测通过即表示转换完成。

超时监视：设置适当的转换时间上限，一旦超时未完成转换，杀死相关进程，停止转换，防止个别文件耗时太久或因某种原因被阻塞，影响总体进度。

2.4 转换格式的比较

我们对不同转换格式进行了比较，如表1所示。

表 1. 转换格式之间的比较.

格式	转换速度	能否直接提取表格	信息完整性	解析难度	解析速度
XML	快	**是**	好	**容易**	快
Word	慢	**是**	较好	一般	慢
Excel	较快	否	**很好**	较难	较快
TXT	**很快**	否	一般	难	**很快**
HTML	慢	**是**	很好	**容易**	较慢

格式简介：（1）XML对文本段落和文字流表格进行了结构化表示，可通过解析XML找到某个节点前后的节点，这有利于确定信息点上下文；缺点是对于某些不规范的表格识别成图片。（2）Word实质内部是基于XML的，保留了字体，样式等更多的格式信息；现有解析Word的工具包只能获取所有的段落和表格，无法获取他们的上下文；同XML一样，Word会对不规范表格的识别成图片。（3）Excel通过设置相同长度的单元格，把PDF中的每一行都填充到单元格中，保留了表格和段落的位置信息，可通过规则抽取出表格；所有格式中，信息保留最完整。（4）TXT只保留了PDF中的所有文本，去掉了位置信息，适合抽取不包含表格的PDF文档，转换速度很快。通过规则也能抽取表格，但准确率较低。（5）HTML相比XML，保留了更多的排版信息，转换速度慢，同时解析速度也较慢。

各项说明：（1）转换速度：TXT和XML转换速度较快，其他格式保留更多的排版，字体等格式信息，转换速度较慢。（2）能否直接提取表格：XML，Word和HTML可使用库直接抽取表格，Excel可通过规则抽取表格。（3）信息完整性：Excel信息保留最完整，其他格式对不规范的PDF有不同程度的信息损失。（4）解析难度：Excel和TXT需要制定相应的规则来抽取信息点，其他格式都有相应的库，可直接抽取信息点。（5）解析速度：XML和TXT保留的格式信息最少，解析快。

综上所述，抽取PDF文档中的表格可选择XML和Excel格式，中小规模文档集可选择Excel（召回率更高），XML更适合大规模文档集（效率和效果兼顾）。以上的几种转换格式不适合抽取公式类的文本，实际上公式太过符号化很难进行信息抽取。

3 公众公司公告信息抽取

随着金融科技的发展和全球资本市场的不断扩大，在金融领域， 每天都有海量的数据产生，而与之形成强烈对比的是有限的人力以及人脑所能处理信息的极限能力。因此，亟需引入新的技术来提高信息处理效率[3]。本次评测任务分为两个子任务，分别抽取PDF中的文字流表格和文本段落中的信息点。

在本次评测中，我们先将PDF转换成XML，然后解析XML抽取表格和信息点。下面我们分别介绍这两个任务使用的方法。

3.1 表格中的信息点提取

任务介绍：公众公司定期报告中财务报表信息点提取（包括合并资产负债表，母公司资产负债表，合并利润表，母公司利润表，合并现金流量表和母公司现金流量表），除表头部分外，对其中每一行采取原样提取原则[3]。

表格抽取：我们使用BeautifulSoup4查找<Table>标签获取所有表格，然后遍历所有表格，根据表格的上下文确定表格的名称，最后抽取表格。表格抽取架构如图2所示。

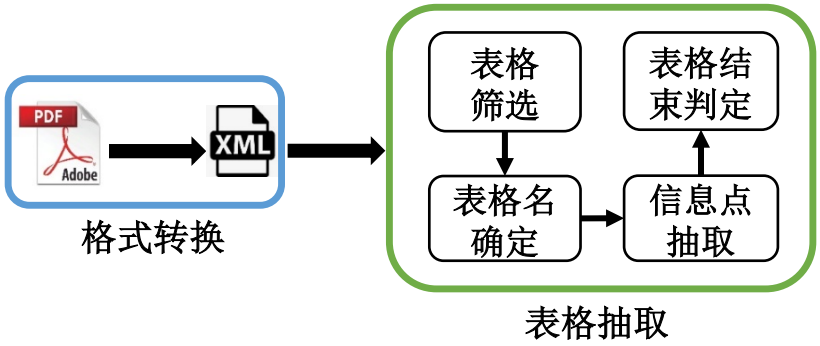

图 2. 表格抽取架构图

XML文件中表格区域的树结构如下：

```
<Table>
  <TR>
    <TD>项目</TD>
    <TD>附注</TD>
```

```
        <TD>本期金额</TD>
        <TD>上期金额</TD>
      </TR>
    </Table>
```

<TR>标签对应表格的一行，<TD>对应表格中一列。
下面介绍表格抽取相关模块。

表格筛选：筛选符合条件的表格。通过标签<Table>抽出的表格包含文档中所有的表格，我们通过表头（表格中第一行）初步塞选出符合条件的表格。任务中需要抽取的表格表头都有“项目”，“附注”字样。

表格名确定：通过查找筛选出来的表格前面节点的文本，确定表格的名称，以及表格中数字的单位。我们设置了查找范围，最多查找表格前3个文本段落，降低表格名出错的概率。

信息点抽取：确定了表格名后需要按行抽取，通过<TR>标签找到表格所有行。需要抽取的表格都是4列，对每个<TR>节点，查找<TD>标签得到4个数据并删除空白符，按顺序填入即可。

表格结束判定：PDF文档中存在跨页的表格，会导致同一个表格在XML中被拆分。经过观察，BeautifulSoup4通过标签查找的节点是按文档出现顺序排序的，即被拆分的表格是处于相邻位置的。可通过下一个表格的表头来判断其是否属于当前表格。为了增强鲁棒性，我们对每个表统计了平均长度，并设置了最大表格长度。另一方面，我们构建了项目名称字典，当抽取的表格项目名称80%出现在字典中，即断定该表格是需要抽取的表格。

在训练集上，我们的方法F1值达到了0.95，理论最佳F1值达到了0.99（忽略附注中的空格，以及不区分0和0.00）。

3.2 文本段落中的信息点提取

任务介绍：提取 “人事变动”类型公告，需要从中提取出离职高管信息及继任者信息[3]。包括离职高管姓名，离职高管性别，离职高管职务，离职原因，继任者姓名，继任者性别和继任者职务。

基于Bi-LSTM-CRF的命名实体识别：我们把这个任务建模为序列标注问题，首先进行命名实体识别，然后结合规则抽取信息点。

训练数据生成：评测数据只提供JSON格式的信息点，因此需要生成序列标注的训练数据。观察到JSON中的信息点几乎都是从文本中原样抽取（除了合并项），我们使用简单的启发式规则对文本段落进行BIO标注。本次任务我们需要识别3类实体：人名，原因，职位。标注过程如下：

（1） 抽取XML中的文本段落，除去空白符，分句。
（2） 抽取JSON中的信息点，得到所有子串。

（3） 遍历所有句子，每个句子所有字初始标记为O。对每个子串，查找其在句子中的所有位置，分别标注B和I。

注：这种方式不能标注包含合并项的信息点，因为无法匹配到该子串。

*命名实体识别模型：*我们使用Bi-LSTM-CRF神经网络模型[4]进行命名实体识别。模型架构图如图3所示。

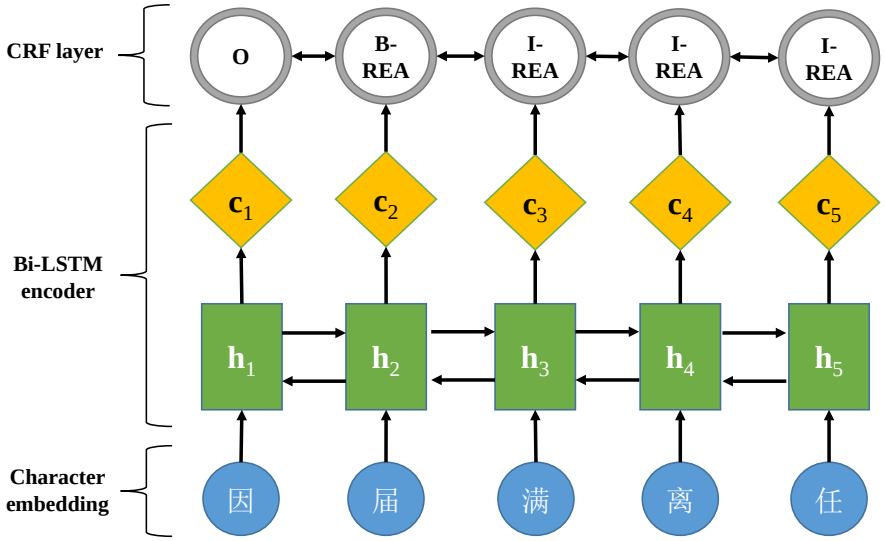

图 3. Bi-LSTM-CRF模型架构图

对于一个句子，该句子中的每个字具有属于集合{O，B-PER，I-PER，B-REA，I-REA，B-TIT，I-TIT}的标签。

第一层是字嵌入层，我们使用金融新闻预训练的词向量[3]，由参考文献[5]发布。第二层Bi-LSTM层可以有效地使用过去和未来的输入信息并自动提取特征。第三层CRF层给每个句子中的字打上BIO标签。最后一层没有使用Softmax，因为Softmax给每个位置打标签是独立的，可能会得到不合法的标记序列。相比Softmax，CRF层可以为最终预测的标签添加一些约束以确保它们有效，在训练过程中，CRF层可以自动从训练数据集中学习这些约束。

我们使用Tensorflow实现Bi-LSTM-CRF，训练并保存最佳的模型用于预测。完成命名实体识别后，需要抽取信息点，下面介绍如何进行句子级别信息抽取。

*句子级别信息抽取：*使用命名实体识别的结果，结合正则表达式进行信息抽取。默认一个句子只有一条的离职信息，对于多个人的情况，本次评测没有考虑这种情况。我们将需要抽取的信息分为离职和聘任。

[3] https://github.com/Embedding/Chinese-Word-Vectors

（1）离职：匹配“(因|由于).*?(申请辞去|辞去|不在担任|不再担任)”正则表达式的句子包含离职的信息点。

a) 离职原因：模型对离职原因识别率很高，如果句子中识别出原因的实体，直接作为最终结果。如果没有识别出，使用正则表达式抽取。
b) 离职高管姓名：定位描述离职的动词（申请辞去，辞去，不在担任，不再担任）的位置，结合模型识别出的人名实体，向前查找，匹配上的第一个人作为结果。如果模型没有识别出人名实体，通过先生，女士这两个词确定姓名。确定方式为，从先生或女士开始，向前查找，找到在全文中出现两次以上的最长子串。
c) 离职高管性别：查找姓名后出现描述性别的词（先生，女士）。
d) 离职高管职务：从离职的动词向后查找，找到最近的一个职位。模型识别的职位实体可能不完整，需要结合正则表达式补全。

（2）聘任：匹配“(聘任|提名|选举|增补).*(为|担任|出任) ”正则表达式的句子包含聘任的信息点。

a) 继任者姓名：同离职一样。
b) 继任者性别：同离职一样。
c) 继任者职务：使用模型识别出的实体，结合正则表达式。

抽取信息点后需要删除重复的信息点。对同一个职务，如果同时出现离职和聘任，则合并。我们的规则系统还能继续完善，以解决跨句信息抽取的问题。

4 相关工作

我们将分别介绍PDF内容抽取和命名实体识别的相关工作。

4.1 PDF内容抽取

关于PDF内容抽取的研究是因实际需求而生，参考文献[6]研究了如何实现PostScript文件与PDF文件间的数据转换。而参考文献[7]则通过分离内容和版式，对版式元素进行提取和封装成对象，提出格式转换通用的技术框架及其文本转换、图像转换、表格转换的技术方案。参考文献[8]提出了一种基于规则与SVM 相结合的PDF 论文抽取方法。该方法充分利用规则方法与机器学习在信息抽取时的优点。参考文献[9]研究了结构化PDF文档与XML文档之间的对应关系，以及利用标签定位PDF文档内容的方法。参考文献[10]利用PDFBox[4]将PDF转为XML，设计了面向医疗知识的PDF文本内容抽取系统。以上文献都是直接解析PDF源文件，相比Adobe公司的软件，丢失很多结构化信息，不利于抽取具有复杂结构的数据，比如表格。我们提出的方法通用性更强，适用不同需求的信息抽取任务。

[4] https://pdfbox.apache.org

4.2 命名实体识别

命名实体识别（Named Entity Recognition，NER）是信息抽取和信息检索中一项重要的任务，其目的是识别出文本中表示命名实体的成分，并对其进行分类。早期的命名实体识别主要使用基于规则[11-13]和统计机器学习[14-17]的方法。近年来，源于神经网络模型的深度学习技术成为机器学习领域新的热潮。使用词向量作为特征，是最为简单有效的方法[18]。更多研究力求借鉴和改进现有的模型和方法，如 LSTM与CRF 相结合的模型[4]，本文使用的就是这种方法。除此之外，卷积神经网络（CNN）[19]也被用来解决NER问题。

5 结论与展望

关于PDF文档内容抽取，在此之前我们已经做了一年的研究，尝试过很多开源工具和软件，效果都无法满足实际需求，直到找到了基于Acrobat DC SDK的方法。在这次评测我们取得了不错的成绩，同时也有一些遗憾。在以后的工作中，我们将继续完善PDF内容抽取系统，并探索其他领域复杂文档的信息抽取任务。

参考文献

1. 宋艳娟[1], 张文德[2]. 基于XML的PDF文档信息抽取系统的研究[J]. 现代图书情报技术, 2005(9).
2. Acrobat SDK Overview, https://www.adobe.com/devnet/acrobat/overview.html.
3. 任务描述, https://www.biendata.com/competition/ccks_2019_5.
4. Huang Z , Xu W , Yu K . Bidirectional LSTM-CRF Models for Sequence Tagging[J]. Computer Science, 2015.
5. Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, Xiaoyong Du, Analogical Reasoning on Chinese Morphological and Semantic Relations, ACL 2018.
6. 孙殷，王鹏.PostScript 文件与PDF 文件间数据转换[J].微型机与应用，2013，32(11)：19-21.
7. 张杰.数字出版中PDF 与EPUB 格式转换技术研究[D].杭州电子科技大学， 2016.
8. 李雪驹，王智广，鲁强.一种规则与SVM 结合的论文抽取方法[J].计算机技 术与发展，2017，(10)：24-29.
9. Mathew R. B. Hardy，David F. Brailsford. Mapping and Displaying Structural Transformations between XML and PDF. In ： Proceedings of the 2002 ACM symposium on Document engineering. Mclean ， Virginia ， USA. 2002. New York，USA: ACM Press，2002. 95~102.
10. 刘现营. 面向医疗知识的PDF文本内容抽取系统设计与实现[D].
11. Collins M, Singer Y. Unsupervised models for named entity classification[C]// Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 1999: 100-110.
12. Cucerzan S, Yarowsky D. Language independent named entity recognition combining morphological and contextual evidence[C]// Proceedings of the 1999 Joint SIGDAT Conference on EMNLP and VLC, 1999: 90-99.

13. Mikheev A, Moens M, Grover C. Named entity recognition without gazetteers[C]// Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 1999: 1-8.
14. 张晓艳, 王挺, 陈火旺. 命名实体识别研究[J]. 计算机科学, 2005, 32(4): 44-48.
15. Bikel D M, Miller S, Schwartz R, et al. Nymble: a high-performance learning name-finder[C]// Proceedings of the Fifth Conference on Applied Natural Language Processing. Stroudsburg: Association for Computational Linguistics, 1997: 194-201.
16. Borthwick A E. A maximum entropy approach to named entity recognition[D]. New York: New York University, 1999.
17. McCallum A, Li W. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons[C]// Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL, Stroudsburg: Association for Computational Linguistics, 2003,4: 188-191.
18. Cherry C, Guo H Y. The unreasonable effectiveness of word representations for Twitter named entity recognition[C]// The 2015 Annual Conference of the North American Chapter of the ACL. Stroudsburg: Association for Computational Linguistics, 2015: 735-745.
19. Dong X S, Qian L J, Guan Y, et al. A multiclass classification method based on deep learning for named entity recognition in electronic medical records[C]// Proceedings of the 2016 New York Scientific Data Summit. IEEE, 2016: 1-10.