实习报告

北京理工大学 张益宁

目录

[1 任务说明 2](#_Toc136625976)

[2 任务要求 2](#_Toc136625977)

[2.1 任务要求 2](#_Toc136625978)

[2.2 注意事项 2](#_Toc136625979)

[3 完成思路 2](#_Toc136625980)

[3.1 实习目标确定 2](#_Toc136625981)

[3.2 相关知识点学习 3](#_Toc136625982)

[3.3 模型思路 4](#_Toc136625983)

[4 数据集介绍 5](#_Toc136625984)

[4.1 人民日报数据集 5](#_Toc136625985)

[4.2 微博数据集 5](#_Toc136625986)

[5模型搭建 5](#_Toc136625987)

[5.1 整体介绍 5](#_Toc136625988)

[5.2 预处理部分 6](#_Toc136625989)

[5.3 词向量编码部分 7](#_Toc136625990)

[5.4 神经网络部分 8](#_Toc136625991)

[5.5 预测结果部分 8](#_Toc136625992)

[6 结果分析 9](#_Toc136625993)

[6.1 评估标准 9](#_Toc136625994)

[6.2 实验结果展示 9](#_Toc136625995)

[7 困难及解决方案 9](#_Toc136625996)

[7.1 预测问题 9](#_Toc136625997)

[8 实习心得体会 9](#_Toc136625998)

## 1 任务说明

阅读相关文献，了解命名实体抽取相关任务和方法，并实现如下任务：

* + 从网络资源中下载《人民日报》和微博领域的命名实体识别数据。
  + 实现或者利用现有的命名实体识别代码和模型实现《人民日报》和微博领域的命名实体识别模型和方法。
  + 利用已有测试集或者自己构建测试集，测试《人民日报》和微博这个领域的实体抽取效果，并进行错误分析。

## 2 任务要求

### 2.1 任务要求

提交程序代码（可编译执行）和收集的（文本）数据（如果题目有要求）；撰写技术报告，报告内容包括对国内外相关研究的调研分析，自己提出的技术方法，对比实验和结果分析，参考文献等。

### 2.2 注意事项

1. 从网上爬取数据时，请务必遵循互联网数据爬取的相关规定，严禁侵犯他人知识产权或违背网站相关要求；严禁爬取和使用反动言论或有损于国家、有损于党、有损于国家领导人和历史人物的文本资料；
2. 访问ChatGPT或其它境外网址时，务必保证只用于学术研究，严禁使用政治敏感词句；严禁泄露国家秘密。一旦出现任何不当行为，行为人自己承担全部后果；
3. 课题研究中使用的任何数据，包括下载后整理的文本和自己构建的样本等，均严禁出现违背宗教、民族、伦理、道德和法律等情况。一旦出现任何不当行为，行为人自己承担全部后果；
4. 撰写技术报告时禁止使用ChatGPT或其它AI工具代写；
5. 实验数据务必真实、可信，严禁伪造或篡改实验结果。

## 3 完成思路

本次任务主要分为四个阶段：

实习目标确定——相关知识点学习——模型编写——结果分析

### 3.1 实习目标确定

首先需要考虑本次实习工作的目的是什么，能够解决什么样的问题等等。考虑到实习的期限较短，主要锻炼自己的能力那种，所以就第一次汇报关于方向确定的地方就想着是主要是练习一下自己能不能独立完成一个比较简单的命名实体识别任务工作。

本次实习中主要完成的任务为：命名实体抽取相关任务，期限为两个月，因此先确定本次实习的整体目标。首先在任务学习层面应该较为完备的了解命名实体识别任务的概念，并且查阅相关重要论文，对应用于拓展以及现状都有较好的认知。然后是方法撰写方面，从命名实体识别最早期的传统方法到目前一些先进的表现较好的方法都要有比较熟悉的认知，其中早期的简单方式：如字典和词向量处理方式应该做到完整独立的代码编写工作。对于部分先进的经典模型需要复现或独立编写的能力。

主要指定日程规划如下：

表格 1实习日程规划

|  |  |
| --- | --- |
| **周** | **任务** |
| 5.11~5.14 | 确定选题 |
| 了解命名实体识别基础知识 |
| 下载并初步分析数据集 |
| 制定日程规划 |
| 5.15~5.21 | 了解基于规则和词典的传统方法 |
| 了解深度学习相关方法 |
| 手动编写基于字典和规则的代码框架 |
| 5.22~5.28 | 对代码框架进行传统深度学习适配 |
| 训练调参 |
| 5.29~6.4 | 了解最新技术 |
| 进行对比实验 |
| 6.5~6.11 | 撰写实习报告 |
| 查漏补缺 |
| 6.12~6.18 | 查漏补缺 |

考虑到临近校内考试周，为了保证本科学业本职工作不被影响，因此在实习安排上尽量空出6月9号结课开始的考试周时间。督促自己提前完成实习任务，后续的时间均留作查漏补缺，以保证实习质量。

### 3.2 相关知识点学习

首先拿到题目的时候第一步就是去搜索相关的基础介绍，初步了解命名实体识别任务的含义，保证基础认知的正确性。

接下来主要是搜索相关的一些论文，与此同时可以着手尝试自己编写一个简单的处理代码，经过了第一周相关的知识点查询和基础学习，我决定在实验中使用构建OneHot词向量的方式加上深度学习进行encoder。选择的愿意是因为查询了相关文献时，对早期的基于词典和规则的办法比较感兴趣，所以就尝试使用深度学习自动的获取规则。

因此主要需要学习的知识点有：

* 自然语言处理概念：
* 自然语言处理相关方法：
* 词向量构建：
* 神经网络训练

### 3.3 模型思路

整个任务的处理过程如下：

1. 先把数据集扫一遍统计出高频词（初步设置为497个），构建高频词字典
2. 训练时，对每一个词构建一个500维的OneHot向量（500\*1）
3. 0~496分别对应高频词，497表示不在字典里的其他词，498表示句首，499表示句尾。
4. 然后对每一个词都链接前面一个词向量和后面一个词向量组成一个1500\*1的向量。
5. 针对这些词向量通过深度学习的方式作为encoder并进行训练。
6. 生成预测结果并进行评估。

下面对词向量构建思路详细说明：

假设我的OneHot是5维的，以识别地名为例：

对于一个经典句子——“我去天安门”的数据集标注为：

我 去 天 安 门

O O B I I

而假设我构建的高频的词典只有两个关键字：“我”和“天”。

那么“去”、“安”、“门”这三个字都会被算作其他。

我们为词典分配序号0~5:，如下所示：

0 1 2 3 4

我 天 其他 句首 句尾

那么这五个字分别单独的OneHot向量为：

我（1,0,0,0,0）

去（0,0,1,0,0）

天（0,1,0,0,0）

安（0,0,1,0,0）

门（0,0,1,0,0）

扩展后的词向量为

我（0,0,0,1,0，1,0,0,0,0，0,0,1,0,0）

去（1,0,0,0,0，0,0,1,0,0，0,1,0,0,0）

天（0,0,1,0,0，0,1,0,0,0，0,0,1,0,0）

安（0,1,0,0,0，0,0,1,0,0，0,0,1,0,0）

门（0,0,1,0,0，0,0,1,0,0，0,0,0,0,1）

后续可以根据这样获取到的词向量进行相关训练。

## 4 数据集介绍

本次任务主要使用了两个开源数据集——People's Daily dataset和Weibo NER dataset。

### 4.1 人民日报数据集

人民日报数据集标注出了三种特殊的命名实体类型：LOC(地名), ORG(机构名)和PER(人名)，使用了BIO标注法。

训练集、验证集和测试集的分割如下：

表格 2 人民日报数据集分割信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **数据集** | **句数** | **字符数** | **LOC** | **ORG** | **PER** |
| 训练集 | 20864 | 979180 | 16571 | 9277 | 8144 |
| 验证集 | 2318 | 109870 | 1951 | 984 | 884 |
| 测试集 | 4636 | 219197 | 3658 | 2185 | 1864 |

### 4.2 微博数据集

微博数据集的标注相对复杂，涵盖PER(人名), LOC(地点名), GPE(行政区名)和ORG(机构名) 四种命名实体，还为其中大类进行了细分。其中NAM特指具体的名称，如“张三”“李四”、NOM代表泛指统称代称等，如“富人”“穷人”。数据集依然使用了BIO标注法。

训练集、验证集和测试集的分割如下：

表格 3 微博数据集分割信息

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **数据集** | **句数** | **字符数** | **类型** | **PER** | **LOC** | **ORG** | **GPE** |
| 训练集 | 1350 | 73778 | NAM | 574 | 56 | 183 | 205 |
| NOM | 766 | 51 | 42 | / |
| 验证集 | 270 | 14509 | NAM | 90 | 6 | 47 | 26 |
| NOM | 208 | 6 | 5 | / |
| 测试集 | 270 | 14842 | NAM | 111 | 19 | 39 | 47 |
| NOM | 170 | 9 | 17 | / |

## 5模型搭建

### 5.1 整体介绍

首先对模型进行一个整体性的介绍：

本次采用的数据集都已经做好了分词操作，首先根据3.3节模型思路中对于词向量的构建操作，将每一个词（中文字）都生成一个词向量。如图1左半部分所示，将得到的向量交给神经网络进行训练，后面的工作等同于传统的三分类工作。最后生成的向量中哪一维度最大即为我们的预测结果。

因此可以我们的模型可以简单的分为：词向量生成部分—神经网络部分—生成预测部分。

模型如下图所示：

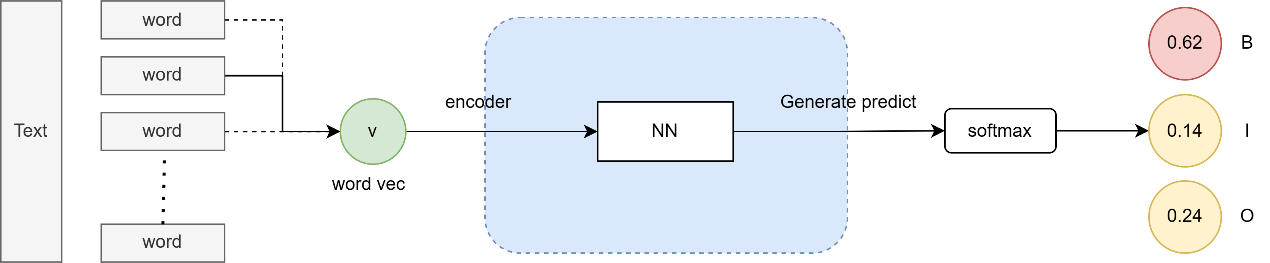


图 1 模型结构图

### 5.2 预处理部分

首先是数据集的预处理部分，因为可以看到两个数据集虽然在句子数量上比较平均，但是实际上命名实体识别处理一般而言都是对单个单词进行处理的，因此以人民日报数据集的训练集为例：

表格 4 人民日报训练集比例

|  |  |  |  |
| --- | --- | --- | --- |
| **字符数** | **LOC** | **ORG** | **PER** |
| 979180 | 16571 | 9277 | 8144 |

比如在进行地名识别时，BI和O的标注比例约为：0.017:1。而在机构名和人名的比较时这个比例会更加的极端。在训练的时候如果不加以特殊处理，那么不管采用什么方式，最终都很难得到较好的结果，大多数情况下网络都会选择把所有的标签全部预测为O以获得较低的损失。

因此我们需要对数据集进行简单的预处理，有两种选择：第一种是删除一部分全O的句子，一种是适当的复制含有BI的句子。在一般情况下，为了保证不主动丢弃数据信息，都会选择复制BI的方式。但是考虑到本次实验的完成周期以及目前我并没有很好的GPU运行环境资源，我选择删除一部分全O的标签。

我单独编写了preprocess.py作为预处理用代码。

预处理部分的伪代码如下所示：

1. **def** write(list):
2. 把list写到目标文档中
4. **for** word\_tag **in** training\_set.readlines():
5. word =切出单词和对应的标签
7. **if** word[0] == "": # 如果完成了一个句子的读入
8. **if** flag == 0:  # 如果整个句子就没有BI。
9. 有一定概率调用write
11. **if** flag == 1:  # 只要这个句子中包含B或者I
12. 调用write写入目标文档
13. temp\_list.clear()
14. flag = 0
15. **else**:
16. temp\_list.append(word) # 存入新的词和标签
17. **if** word[1] == "B-LOC" **or** word[1] == "I-LOC": # 检查是否有BI标签
18. flag = 1

因为本次实验我选择分别进行地名、人名和组织名的识别，

### 5.3 词向量编码部分

词向量的编码原则在上文3.3节中已经说明，下面主要介绍具体的代码结构，首先是生成词典，我单独编写了generate\_dictionary.py用于生成一个独立的词典文件——dictionary.txt。

生成词典的伪代码如下所示：

1. **def** Generate\_dictionary(path):
2. input\_file = 测试集的路径
3. output\_file = "dictionary.txt"路径
5. **for** sentence **in** input\_file:
6. words = 取一个词
7. **if** words[0] != "": # 如果不是空行
8. 记录这个词的出现数量
10. word\_counts = collections.Counter(wordlist)
11. word\_counts\_top = word\_counts.most\_common(497)
12. # 统计出出现频率最高的前497个词，方便和句首、句尾、其他构成500维的One-Hot
13. **for** word **in** word\_counts\_top:
14. 写出对应的词
16. 关闭输入和输出文件

可以更改word\_counts\_top的维数，从而做到生成不同大小的词典。词典的大小+3一定和OneHot向量的维数相同。

从而做到词典的维数+其他+句首+句尾=OneHot向量的维数。

以人民日报的训练集为例，统计前497个高频词生成的词典文如下图所示：

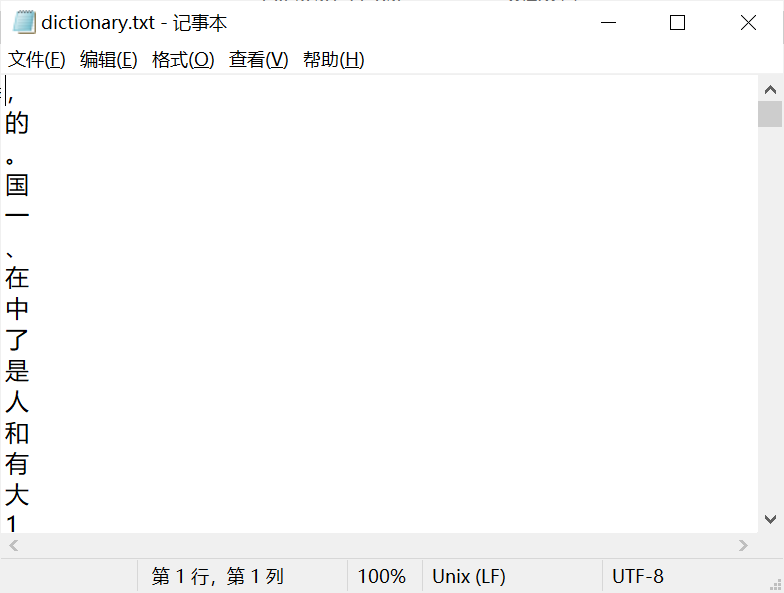


图 2 词典文件

因此可以给出OneHot向量的格式应该为：

表格 5 OneHot向量格式

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | 0 | 1 | 2 | 3 | 4 | … | 496 | 497 | 498 | 499 |
| **内容** | , | 的 | 。 | 国 | 一 |  | 防 | 其他 | 句数 | 句尾 |

综上：

比如单词：“的”，生成的OneHot向量应该为：

比如单词：“防”，生成的OneHot向量应该为：

经过词向量生成的处理，每一个字都可以处理成由499个0和1个1组成的OneHot向量，然后根据每个字的前后信息合成一个窗口大小为3的词向量。

### 5.4 神经网络部分

### 5.5 预测结果部分

## 6 结果分析

### 6.1 评估标准

本次采用经典的分类评估标准：

* 查全率（Recall）
* 查准率（Precision）
* F1分数（f1-measure）

注意到因为是三分类，这里没有使用传统的TN、TP、FN、FP的计算方式。我们认为B和I的标注为正例，O的标注为负例，因此可以规定：

* 预测结果为B和I中的一个时记为预测是正，总数为p\_sum。
* 真实标签为B和I中的一个时记为实际是正，总数为r\_sum。
* 将标签为B的词预测为B、将标签为I的词预测为I，即为Hit。

因此可以得到评估标准的计算公式：

### 6.2 实验结果展示

## 7 困难及解决方案

### 7.1 预测问题

## 8 实习心得体会