**武汉大学国家网络安全学院**

**实验报告**

**课程名称 自然语言处理**

**专业年级 2019级**

**姓 名 戴挺**

**学 号 2018302060188**

**协 作 者 无**

**实验学期 2021-2022　 学年 第一 学期**

**课堂时数 课外时数**

**填写时间 2021 年 11 月 20 日**

|  |
| --- |
| **实验介绍** |
| **【实验名称】：命名实体识别** |
| **【实验目的】：**   * 了解神经网络中的基础模型   神经网络是指一系列受生物学和神经科学启发的数学模型。主要是通过对人脑的神经元网络进行抽象，构建人工神经元，并按照一定拓扑结构来建立神经元之间的连接。前馈神经网络（FNN）是最早发明的简单人工神经网络；卷积神经网络（CNN）是一种具有局部连接、权重共享等特性的深层前馈神经网络；循环神经网络（RNN）是一类具有短期记忆能力的神经网络。   * 了解深度学习框架Pytorch的使用   PyTorch使用python作为开发语言，近年来和TensorFlow, keras, caffe等热门框架一起，成为深度学习开发的主流平台之一。PyTorch的基本元素包含张量(Tensor)、变量(Variable)、神经网络模块(nn.Module)等。   * 了解使用深度学习解决序列标注任务基本流程   序列标注是NLP领域的基础问题之一，涵盖范围非常广泛，如分词、词性标注、命名实体识别、关系抽取等等，本质上是对线性序列中每个元素根据上下文内容进行分类的问题。命名实体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，以及时间、数量、货币、比例数值等文字。  以PyTorch为例，一个常规的序列标注任务代码开发流程是：安装并导入相关的深度学习库、定义标签集合(Label set)、数据获取和预处理、定义神经网络、定义损失函数(loss function)和优化器(optimizer)、训练网络和测试网络。  **【实验环境】：**  python3 + jieba + pytorch1.0.0 + numpy  pip install -r requirement.txt   * python3   除了高性能外，拥有NumPy、SciPy等优秀的数值计算、统计分析库。TensorFlow、Caffe等著名的深度学习框架都提供了Python接口。   * jieba   jieba是一款优秀的Python第三方中文分词库，支持三种分词模式：精确模式、全模式和搜索引擎模式。   * PyTorch   PyTorch是一个针对深度学习，并且使用GPU和CPU来优化的tensor library，它是一个以Python优先的深度学习框架，不仅能够实现强大的GPU加速，同时还支持动态神经网络。   * NumPy   NumPy是Python语言的一个扩展程序库，支持大量的维度数组与矩阵运算，此外也针对数组运算提供大量的数学函数库。  **【参考文献】：**  **[1] 数据集来源：**(<https://github.com/jiesutd/LatticeLSTM>)  **[2]** [**pytorch1.0.0官方文档**](https://pytorch.org/docs/1.0.0/)  **[3]**[**《神经网络与深度学习》**](https://nndl.github.io/)  **[4] Conv1d详解：https://blog.csdn.net/sunny\_xsc1994/article/details/82969867** |
| **实验内容** |
| **【实验方案设计】：**  本次实验主要通过训练bi\_lstm模型以及cnn模型以进行命名实体识别。  数据集用的是论文ACL 2018 Chinese NER using Lattice LSTM中收集的简历数据，数据的格式如下，它的每一行由一个字及其对应的标注组成，标注集采用BME，B表示实体开头，M表示实体中间，E表示实体结尾，句子之间用一个空行隔开。该数据集就位于目录下的ResumeNER文件夹里。如下为数据及标签示例：     1. 分别使用BiLSTM和BiLSTM + CRF进行NER任务，对比实验结果。 2. 分别使用CNN和CNN + CRF进行命名实体识别（Named Entity Recognition, NER）任务，对比实验结果。 3. 调整CNN卷积核尺寸和filter/channel参数   调整CNN的卷积核尺寸的代码如下：    在这里可以调整Layer0以及剩余Layers的kernel\_size  同时需要注意，在调整kernel\_size的同时，需要调整padding的大小始终保持:  **Kernel\_size = 2 \* padding + 1**  调整CNN中的channel的数值，则修改config.py文件中hidden\_size的值     1. 分别使用BiLSTM+CRF和CNN + CRF进行命名实体识别（Named Entity Recognition, NER）任务，对比实验结果。 2. 分别在使用预训练词向量和不使用预训练词向量的情况下进行NER任务，对比实验结果。   本次实验对于模型的评估主要基于对于各个实体类型的分类识别的recall，precision以及F1-score（**在实验总结中会对这类概念做总结）**，最终对所有类别的recall，precison以及F1-score做加权平均，并用F1-score作为最终模型的评价参数。  计算的代码如下：    Recall的加权均值以及precision的加权均值的计算式如下：  Recall = Recall(1) \* num(1) + Recall(2) \* num(2) + …Recall(n) \* num(n) / total\_num  Precision= Precison(1) \* num(1) + Precision(2) \* num(2) + …Precision(n) \* num(n) / total\_num  数学上容易证明，recall加权平均值和precision的加权平均值，两者的权重分布相同时，再求调和平均值依旧等于f1-score的加权平均值。所以再求f1-score的加权平均值时，为简化计算可以不用累计求和计算。  、  **【实验结果分析】：**   1. **对比BiLSTM以及BiLSTM + CRF**       Bi\_LSTM模型在加入CRF层之后，模型的整体precision有稍微下降，recall稍微提升，最终的F1-score是稍微提升，加入CRF感觉对Bi\_LSTM模型的影响不大。但是前者运行了227s,后者运行了280s，时间和算力成本增加较大。   1. **对比CNN以及CNN + CRF**       对于模型CNN，加入CRF层之后precision以及recall都有比较显著的提升，最终F1-score也是提高不少。但是前者运行58s，后者运行了113s，时间和算力成本几乎翻倍。   1. **调整CNN卷积核尺寸和filter/channel参数()**   **Layer0的kernel\_size = 1, 剩余的Layers的kernel\_size = 5**    **Layer0的kernel\_size = 3, 剩余的Layers的kernel\_size = 3**    **Layer0的kernel\_size = 1, 剩余的Layers的kernel\_size = 3**    **Layer0的kernel\_size = 3, 剩余的Layers的kernel\_size = 5**    **调整hidden\_size( Layer0的kernel\_size = 1, 剩余的Layers的kernel\_size = 3)**  **Hiden\_size= 50**    **Hidden\_size = 150**    **Hidden\_size = 100**    **Hidden\_size = 25**     1. **对比Bi\_LSTM + CRF以及CNN+ CRF**       对比Bi\_LSTM+ CRF，CNN + CRF,观察出前者的precision，recall以及F1-score都高于后者。但是前者运行了280s，后者运行了113s。虽然模型精度上Bi-LSTM更高，但是模型性价CNN+CRF更佳。   1. **对比**CNN+CRF + W2V 与CNN+CRF；Bi\_LSTM +CRF + W2V与Bi\_lstm +CRF       使用预训练的词向量模型，反而使得模型的三个指标都稍微下降，分析原因可能是预训练的词向量模型不能完美适配这份数据集。      **对于**Bi\_LSTM+ CRF模型，使用预训练词向量模型precision略有下降，recall则略有提高，最终F1-SCORE提高。 |
| **【实验总结】：**  **1、命名实体识别简介（Named Entity Recognition）**  信息提取中的一个经典问题是识别和提取文本中对命名实体的提及。在新闻文件中，核心实体类型是人员、地点和组织；最近，NER任务已扩展到包括金额、百分比、日期和时间。标记命名实体跨度的标准方法是使用条件随机场（CRF）等判别性序列标记方法。然而，命名实体识别（NER）任务与pos等序列标记任务有根本区别，NER不仅仅是给每个token做分类标记，同时也要还原token的长度。像本次实验中，X = TITLE，ORG，NAME的分类标签都包含B-X， M - X， E - X这三类标签表示每个token的开始，中间以及结束部分。  目前的研究强调基于神经网络的序列标注，使用类似于pos标记的LSTM模型（Hammerton，2003；Huang，2015；Lample，2016）。bi-LSTM-CRF在这项任务上表现地特别好。然而，Strubell等人（2017）表明，卷积神经网络CNN可以同样准确，由于在图形处理单元（GPU）上实现ConvNets的效率高，模型训练速度显著提高。  **2、Recall, precision, F-measure & confusion matrix简介**  命名实体识别的本质是其实还是文本分类，我们人为地将一个文本序列划分为word之后，目的是识别出每个entity以及非entity的other；在识别出entity的同时，进一步识别出每个entity的start, middle, 以及end。而对于分类问题的模型评价标准，recall以及precision和F-measure是比较好的标准。  对于训练好的模型，输入一个文本序列，可以输出其识别出的分类以及对应的个数，同时文本序列本身具有人工预先标记的准确分类。假设文本总共被被分为n类，我们设t1 ,t2, …tn, f1,f2,… fn 分别为模型正确分类的1类，2类，…n类，模型错误分类的1类，2类，…n类（ps:模型正确的分类是原本为x类，且模型识别为x类；模型错误分类为原本不是x类，模型识别为x类），进一步，我们便引入confusion\_matrix。   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **模型输出分类 \ 实际真实分类** | **1类** | **2类** | **。。。** | **N类** | | **1类** | **t1** | **f1** | **。。。** | **f1** | | **2类** | **f2** | **t2** | **。。。** | **f2** | | **。。。** | **。。。** | **。。。** | **。。。** | **。。。** | | **N类** | **fn** | **fn** | **。。。** | **tn** |   记**A = confusion\_matrix**  定义**recall(x) = A(x,x) / [A(1,x) + A(2,x) + … + A(n,x)]**为x类的召回率(recall);  定义**precision(x) = A(x,x) / [A(x,1) + A(x,2) + … A(x,n)]**为x类的精确率(precision);  从recall以及precision的计算式便可以看出两个参数是从不同的角度来评价模型对于x类的识别精度。recall反映的是模型正确识别的x类占原本为x类的比率；precision反映的是模型有多少比率识别的x类中是原本为x类。  一个模型识别x类的recall和precision可以相差很大。举个例子，假设模型识别的1类有3个，且这3个确实都是1类，则precision(1) = 100 %，而原本为1类的总数有100个，该模型未将剩余97个1类识别为1类，则recall(1) = 3 % ，因此不能因为precision高而直接认为模型对某一类别可以较准确的识别；同样的道理也不能因为recall高而直接认为模型对某一类别可以较准确的识别，假设模型设别的1类有80个，且其中只有10个是真的1类，其他70个均不是一类，这时precision(1) = 10 / 80 = 12.5%。而真正的1类总共有10个，因此recall(1) = 100 %，即所有真正的1类都被正确识别。  由上面两个例子可以看出，评价一个模型对于x类的识别正确度，不可以单靠recall或者precision,应当同时保证两个参数都处于比较高的水平。这时，我们引入了F-measure的机制，对模型进行综合评分。  定义**F = (a ^ 2 + 1) \* r \* p / (a ^ 2 \* p + r)**  参数a可能根据实际的需求，对召回和精度的重要性进行了不同的权重。a>1，召回率权重更大，而a<1，精确度权重更大。当a = 1时，精度和召回量是同等的，这也是最常用的指标F1-score = r \* p / (p + r)，也就是**p和r的调和平均值**。 |
| **评语及评分（指导教师）** |
| **【评语】：**      **评分：**  **日期：** |

**附件2：**

**实验报告说明**

**1．实验名称：**要用最简练的语言反映实验的内容。

**2．实验目的**：目的要明确，要抓住重点。

**3．实验环境**：实验用的软硬件环境（配置）。

**4．实验方案设计（思路、**步骤和方法等**）**：这是实验报告极其重要的内容。包括概要设计、详细设计和核心算法说明及分析，系统开发工具等。应同时提交程序或设计电子版。

对于**设计型和综合型实验**，在上述内容基础上还应该画出流程图、设计思路和设计方法，再配以相应的文字说明。

对于**创新型实验**，还应注明其创新点、特色。

**5．实验结果分析：**即根据实验过程中所见到的现象和测得的数据，进行对比分析并做出结论（可以将部分测试结果进行截屏）。

**6．实验总结：**对本次实验的心得体会，所遇到的问题及解决方法，其他思考和建议。

**7．评语及评分：**指导教师依据学生的实际报告内容，用简练语言给出本次实验报告的评价和价值。