

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](#)

[3] [《神经网络与深度学习》](#)

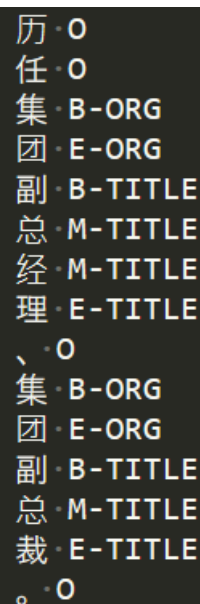
[4] Conv1d 详解：https://blog.csdn.net/sunny_xsc1994/article/details/82969867

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：



历·O
任·O
集·B-ORG
团·E-ORG
副·B-TITLE
总·M-TITLE
经·M-TITLE
理·E-TITLE
、·O
集·B-ORG
团·E-ORG
副·B-TITLE
总·M-TITLE
裁·E-TITLE
。·O

- 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：

```
class CNNmodel(nn.Module):
    def __init__(self, input_dim, hidden_dim, num_layer=3, dropout=0.2):
        super(CNNmodel, self).__init__()
        self.input_dim = input_dim
        self.hidden_dim = hidden_dim
        self.num_layer = num_layer

        self.cnn_layer0 = nn.Conv1d(self.input_dim, self.hidden_dim, kernel_size=3, padding=1)
        self.cnn_layers = nn.ModuleList([nn.Conv1d(self.hidden_dim, self.hidden_dim, kernel_size=3, padding=1) for i in range(1, num_layer)])
        self.drop = nn.Dropout(dropout)
```

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：

$$\text{Kernel_size} = 2 * \text{padding} + 1$$

调整 CNN 中的 channel 的数值，则修改 config.py 文件中 hidden_size 的值

```
class CNNConfig(object):
    num_layer = 3 # CNN层数
    hidden_size = 100 # CNN输出维度
```

4、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。

5、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, precision 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：

```
def _cal_weighted_average(self):  
  
    weighted_average = {}  
    total = len(self.golden_tags)  
  
    # 计算weighted precisions:  
    weighted_average['precision'] = 0.  
    weighted_average['recall'] = 0.  
    weighted_average['f1_score'] = 0.  
    for tag in self.tagset:  
        size = self.golden_tags_counter[tag]  
        weighted_average['precision'] += self.precision_scores[tag] * size  
        weighted_average['recall'] += self.recall_scores[tag] * size  
        weighted_average['f1_score'] += self.f1_scores[tag] * size  
  
    for metric in weighted_average.keys():  
        weighted_average[metric] /= total  
  
    return weighted_average
```

Recall 的加权均值以及 precision 的加权均值的计算式如下：

$$\text{Recall} = \text{Recall}(1) * \text{num}(1) + \text{Recall}(2) * \text{num}(2) + \dots \text{Recall}(n) * \text{num}(n) / \text{total_num}$$
$$\text{Precision} = \text{Precision}(1) * \text{num}(1) + \text{Precision}(2) * \text{num}(2) + \dots \text{Precision}(n) * \text{num}(n) / \text{total_num}$$

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

评估 bilstm 模型中...

	precision	recall	f1-score	support
M-TITLE	0.9423	0.8918	0.9163	1922
B-ORG	0.9532	0.9584	0.9558	553
B-NAME	0.9901	0.8929	0.9390	112
E-TITLE	0.9870	0.9858	0.9864	772
E-NAME	1.0000	0.9375	0.9677	112
M-NAME	0.9080	0.9634	0.9349	82
B-TITLE	0.9300	0.9288	0.9294	772
E-ORG	0.9266	0.9132	0.9199	553
O	0.9644	0.9827	0.9735	5897
M-ORG	0.9589	0.9614	0.9602	4325
label_level	0.9578	0.9579	0.9577	15100
TITLE	0.9620	0.9171	0.9390	772
ORG	0.9373	0.8644	0.8993	553
NAME	0.9897	0.8571	0.9187	112
all_ent	0.9546	0.8921	0.9223	1437

评估 bilstm+CRF 模型中...

	precision	recall	f1-score	support
B-TITLE	0.9420	0.9262	0.9340	772
B-NAME	0.9900	0.8839	0.9340	112
M-NAME	0.9506	0.9390	0.9448	82
E-TITLE	0.9883	0.9832	0.9857	772
B-ORG	0.9450	0.9638	0.9543	553
M-ORG	0.9474	0.9711	0.9591	4325
O	0.9675	0.9751	0.9713	5897
E-NAME	0.9903	0.9107	0.9488	112
M-TITLE	0.9433	0.8829	0.9121	1922
E-ORG	0.9145	0.9096	0.9121	553
label_level	0.9559	0.9560	0.9557	15100
NAME	1.0000	0.8839	0.9384	112
ORG	0.9310	0.8788	0.9042	553
TITLE	0.9555	0.9171	0.9359	772
all_ent	0.9493	0.8998	0.9239	1437

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。

2、对比 CNN 以及 CNN + CRF

评估 cnn模型 中...

	precision	recall	f1-score	support
E-TITLE	0.9733	0.9922	0.9827	772
B-TITLE	0.8646	0.9262	0.8943	772
O	0.9457	0.9456	0.9456	5897
M-TITLE	0.8808	0.8881	0.8845	1922
B-ORG	0.9586	0.8788	0.9170	553
M-ORG	0.9310	0.9140	0.9224	4325
E-NAME	0.9737	0.9911	0.9823	112
E-ORG	0.8698	0.9060	0.8875	553
M-NAME	0.8571	0.9512	0.9017	82
B-NAME	0.8729	0.9196	0.8957	112
label_level	0.9274	0.9269	0.9270	15100
NAME	1.0000	0.9196	0.9581	112
TITLE	0.9431	0.8808	0.9109	772
ORG	0.8527	0.6492	0.7372	553
all_ent	0.9173	0.7947	0.8516	1437

评估 cnn+CRF模型 中...

	precision	recall	f1-score	support
M-ORG	0.9367	0.9473	0.9419	4325
B-NAME	0.8857	0.8304	0.8571	112
E-TITLE	0.9795	0.9896	0.9845	772
M-NAME	0.9250	0.9024	0.9136	82
B-TITLE	0.9138	0.9197	0.9167	772
E-ORG	0.9043	0.9222	0.9132	553
O	0.9531	0.9540	0.9536	5897
B-ORG	0.9345	0.9295	0.9320	553
M-TITLE	0.9227	0.8939	0.9080	1922
E-NAME	0.9810	0.9196	0.9493	112
label_level	0.9409	0.9410	0.9409	15100
ORG	0.9061	0.8029	0.8514	553
NAME	1.0000	0.8125	0.8966	112
TITLE	0.9454	0.8977	0.9209	772
all_ent	0.9346	0.8546	0.8928	1437

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。

3、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

评估 cnn+CRF模型中...

	precision	recall	f1-score	support
E-TITLE	0.9820	0.9883	0.9852	772
O	0.9644	0.9513	0.9578	5897
E-ORG	0.9124	0.9042	0.9083	553
B-ORG	0.9406	0.9458	0.9432	553
E-NAME	0.9823	0.9911	0.9867	112
B-TITLE	0.9197	0.9352	0.9274	772
B-NAME	0.9910	0.9821	0.9865	112
M-NAME	0.8421	0.9756	0.9040	82
M-ORG	0.9306	0.9579	0.9441	4325
M-TITLE	0.9345	0.8975	0.9156	1922
label_level	0.9464	0.9462	0.9461	15100
NAME	0.9908	0.9643	0.9774	112
ORG	0.9273	0.8535	0.8889	553
TITLE	0.9507	0.9249	0.9376	772
all_ent	0.9452	0.9005	0.9223	1437

Layer0 的 kernel_size = 3, 剩余的 Layers 的 kernel_size = 3

训练完毕,共用时117秒.

评估 cnn+CRF模型中...

	precision	recall	f1-score	support
O	0.9644	0.9515	0.9579	5897
M-NAME	0.8526	0.9878	0.9153	82
B-ORG	0.9205	0.9421	0.9312	553
E-TITLE	0.9720	0.9896	0.9807	772
E-ORG	0.9041	0.9204	0.9122	553
M-ORG	0.9348	0.9519	0.9433	4325
M-TITLE	0.9248	0.8897	0.9069	1922
B-TITLE	0.8989	0.9326	0.9154	772
B-NAME	0.9902	0.9018	0.9439	112
E-NAME	0.9655	1.0000	0.9825	112
label_level	0.9437	0.9434	0.9434	15100
TITLE	0.9373	0.9106	0.9238	772
NAME	0.9902	0.9018	0.9439	112
ORG	0.8958	0.8391	0.8665	553
all_ent	0.9255	0.8824	0.9035	1437

Layer0 的 kernel_size = 1, 剩余的 Layers 的 kernel_size = 3

评估 cnn+CRF模型中...

	precision	recall	f1-score	support
M-ORG	0.9367	0.9473	0.9419	4325
B-NAME	0.8857	0.8304	0.8571	112
E-TITLE	0.9795	0.9896	0.9845	772
M-NAME	0.9250	0.9024	0.9136	82
B-TITLE	0.9138	0.9197	0.9167	772
E-ORG	0.9043	0.9222	0.9132	553
O	0.9531	0.9540	0.9536	5897
B-ORG	0.9345	0.9295	0.9320	553
M-TITLE	0.9227	0.8939	0.9080	1922
E-NAME	0.9810	0.9196	0.9493	112
label_level	0.9409	0.9410	0.9409	15100
ORG	0.9061	0.8029	0.8514	553
NAME	1.0000	0.8125	0.8966	112
TITLE	0.9454	0.8977	0.9209	772
all_ent	0.9346	0.8546	0.8928	1437

Layer0 的 kernel_size = 3, 剩余的 Layers 的 kernel_size = 5

评估 cnn+CRF模型中...

	precision	recall	f1-score	support
B-NAME	1.0000	0.9911	0.9955	112
O	0.9633	0.9601	0.9617	5897
M-TITLE	0.9328	0.9027	0.9175	1922
B-TITLE	0.9174	0.9352	0.9262	772
E-NAME	0.9737	0.9911	0.9823	112
M-NAME	0.9412	0.9756	0.9581	82
B-ORG	0.9320	0.9421	0.9371	553
M-ORG	0.9472	0.9533	0.9502	4325
E-ORG	0.8995	0.9222	0.9107	553
E-TITLE	0.9758	0.9935	0.9846	772
label_level	0.9498	0.9498	0.9498	15100
TITLE	0.9458	0.9262	0.9359	772
NAME	0.9821	0.9821	0.9821	112
ORG	0.9154	0.8608	0.8872	553
all_ent	0.9373	0.9054	0.9211	1437

调整 hidden_size(Layer0 的 kernel_size = 1, 剩余的 Layers 的 kernel_size = 3)

Hidden_size= 50

评估 cnn+CRF模型中...

	precision	recall	f1-score	support
B-NAME	1.0000	0.9911	0.9955	112
O	0.9633	0.9601	0.9617	5897
M-TITLE	0.9328	0.9027	0.9175	1922
B-TITLE	0.9174	0.9352	0.9262	772
E-NAME	0.9737	0.9911	0.9823	112
M-NAME	0.9412	0.9756	0.9581	82
B-ORG	0.9320	0.9421	0.9371	553
M-ORG	0.9472	0.9533	0.9502	4325
E-ORG	0.8995	0.9222	0.9107	553
E-TITLE	0.9758	0.9935	0.9846	772
label_level	0.9498	0.9498	0.9498	15100
TITLE	0.9458	0.9262	0.9359	772
NAME	0.9821	0.9821	0.9821	112
ORG	0.9154	0.8608	0.8872	553
all_ent	0.9373	0.9054	0.9211	1437

Hidden_size = 150

训练完毕,共用时141秒.

评估 cnn+CRF模型中...

	precision	recall	f1-score	support
M-ORG	0.9325	0.9480	0.9402	4325
M-NAME	0.9630	0.9512	0.9571	82
B-ORG	0.9268	0.9385	0.9326	553
B-NAME	0.8390	0.8839	0.8609	112
E-TITLE	0.9734	0.9935	0.9833	772
O	0.9566	0.9451	0.9508	5897
E-NAME	1.0000	0.9375	0.9677	112
M-TITLE	0.9152	0.8928	0.9039	1922
E-ORG	0.9007	0.9186	0.9096	553
B-TITLE	0.9084	0.9249	0.9166	772
label_level	0.9391	0.9390	0.9390	15100
TITLE	0.9432	0.9028	0.9226	772
ORG	0.9063	0.8047	0.8525	553
NAME	0.9898	0.8661	0.9238	112
all_ent	0.9330	0.8622	0.8962	1437

Hidden_size = 100

评估 cnn+CRF模型中...

	precision	recall	f1-score	support
M-ORG	0.9367	0.9473	0.9419	4325
B-NAME	0.8857	0.8304	0.8571	112
E-TITLE	0.9795	0.9896	0.9845	772
M-NAME	0.9250	0.9024	0.9136	82
B-TITLE	0.9138	0.9197	0.9167	772
E-ORG	0.9043	0.9222	0.9132	553
O	0.9531	0.9540	0.9536	5897
B-ORG	0.9345	0.9295	0.9320	553
M-TITLE	0.9227	0.8939	0.9080	1922
E-NAME	0.9810	0.9196	0.9493	112
label_level	0.9409	0.9410	0.9409	15100
ORG	0.9061	0.8029	0.8514	553
NAME	1.0000	0.8125	0.8966	112
TITLE	0.9454	0.8977	0.9209	772
all_ent	0.9346	0.8546	0.8928	1437

Hidden_size = 25

训练完毕,共用时86秒.

评估 cnn+CRF模型中...

	precision	recall	f1-score	support
M-TITLE	0.9074	0.8824	0.8948	1922
O	0.9444	0.9510	0.9477	5897
E-NAME	1.0000	0.8750	0.9333	112
M-ORG	0.9257	0.9394	0.9325	4325
M-NAME	0.9437	0.8171	0.8758	82
B-NAME	0.9885	0.7679	0.8643	112
E-TITLE	0.9707	0.9883	0.9795	772
B-TITLE	0.9099	0.9028	0.9064	772
B-ORG	0.9254	0.9421	0.9337	553
E-ORG	0.9043	0.8716	0.8877	553
label_level	0.9325	0.9325	0.9323	15100
NAME	0.9882	0.7500	0.8528	112
TITLE	0.9417	0.8782	0.9088	772
ORG	0.8955	0.7902	0.8396	553
all_ent	0.9273	0.8344	0.8784	1437

4、对比 Bi_LSTM + CRF 以及 CNN+ CRF

评估 bilstm+CRF 模型中...

	precision	recall	f1-score	support
B-TITLE	0.9420	0.9262	0.9340	772
B-NAME	0.9900	0.8839	0.9340	112
M-NAME	0.9506	0.9390	0.9448	82
E-TITLE	0.9883	0.9832	0.9857	772
B-ORG	0.9450	0.9638	0.9543	553
M-ORG	0.9474	0.9711	0.9591	4325
O	0.9675	0.9751	0.9713	5897
E-NAME	0.9903	0.9107	0.9488	112
M-TITLE	0.9433	0.8829	0.9121	1922
E-ORG	0.9145	0.9096	0.9121	553
label_level	0.9559	0.9560	0.9557	15100
NAME	1.0000	0.8839	0.9384	112
ORG	0.9310	0.8788	0.9042	553
TITLE	0.9555	0.9171	0.9359	772
all_ent	0.9493	0.8998	0.9239	1437

评估 cnn+CRF 模型中...

	precision	recall	f1-score	support
M-ORG	0.9367	0.9473	0.9419	4325
B-NAME	0.8857	0.8304	0.8571	112
E-TITLE	0.9795	0.9896	0.9845	772
M-NAME	0.9250	0.9024	0.9136	82
B-TITLE	0.9138	0.9197	0.9167	772
E-ORG	0.9043	0.9222	0.9132	553
O	0.9531	0.9540	0.9536	5897
B-ORG	0.9345	0.9295	0.9320	553
M-TITLE	0.9227	0.8939	0.9080	1922
E-NAME	0.9810	0.9196	0.9493	112
label_level	0.9409	0.9410	0.9409	15100
ORG	0.9061	0.8029	0.8514	553
NAME	1.0000	0.8125	0.8966	112
TITLE	0.9454	0.8977	0.9209	772
all_ent	0.9346	0.8546	0.8928	1437

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。

5、对比 CNN+CRF + W2V 与 CNN+CRF ; Bi_LSTM +CRF + W2V 与 Bi_lstm +CRF

评估 cnn+CRF+w2v模型中...

	precision	recall	f1-score	support
B-NAME	0.8750	0.8750	0.8750	112
B-TITLE	0.8980	0.9236	0.9106	772
M-ORG	0.9337	0.9341	0.9339	4325
E-ORG	0.8746	0.9204	0.8969	553
E-TITLE	0.9858	0.9922	0.9890	772
E-NAME	1.0000	0.9196	0.9581	112
O	0.9490	0.9496	0.9493	5897
M-TITLE	0.9185	0.8913	0.9047	1922
M-NAME	0.8295	0.8902	0.8588	82
B-ORG	0.9328	0.9295	0.9312	553
label_level	0.9359	0.9357	0.9357	15100
NAME	0.9592	0.8393	0.8952	112
TITLE	0.9479	0.8964	0.9214	772
ORG	0.8946	0.7830	0.8351	553
all_ent	0.9291	0.8483	0.8869	1437

评估 cnn+CRF模型中...

	precision	recall	f1-score	support
M-ORG	0.9367	0.9473	0.9419	4325
B-NAME	0.8857	0.8304	0.8571	112
E-TITLE	0.9795	0.9896	0.9845	772
M-NAME	0.9250	0.9024	0.9136	82
B-TITLE	0.9138	0.9197	0.9167	772
E-ORG	0.9043	0.9222	0.9132	553
O	0.9531	0.9540	0.9536	5897
B-ORG	0.9345	0.9295	0.9320	553
M-TITLE	0.9227	0.8939	0.9080	1922
E-NAME	0.9810	0.9196	0.9493	112
label_level	0.9409	0.9410	0.9409	15100
ORG	0.9061	0.8029	0.8514	553
NAME	1.0000	0.8125	0.8966	112
TITLE	0.9454	0.8977	0.9209	772
all_ent	0.9346	0.8546	0.8928	1437

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。

评估 bilstm+CRF+w2v 模型中 ...

	precision	recall	f1-score	support
M-NAME	0.9157	0.9268	0.9212	82
B-NAME	0.9798	0.8661	0.9194	112
B-TITLE	0.9363	0.9326	0.9345	772
E-ORG	0.9301	0.9150	0.9225	553
M-ORG	0.9588	0.9621	0.9604	4325
E-NAME	0.9808	0.9107	0.9444	112
M-TITLE	0.9418	0.8928	0.9167	1922
B-ORG	0.9587	0.9656	0.9622	553
E-TITLE	0.9870	0.9845	0.9857	772
0	0.9607	0.9795	0.9700	5897
label_level	0.9567	0.9568	0.9566	15100
ORG	0.9288	0.8969	0.9126	553
NAME	1.0000	0.8482	0.9179	112
TITLE	0.9483	0.9262	0.9371	772
all_ent	0.9443	0.9088	0.9262	1437

评估 bilstm+CRF 模型中 ...

	precision	recall	f1-score	support
B-TITLE	0.9420	0.9262	0.9340	772
B-NAME	0.9900	0.8839	0.9340	112
M-NAME	0.9506	0.9390	0.9448	82
E-TITLE	0.9883	0.9832	0.9857	772
B-ORG	0.9450	0.9638	0.9543	553
M-ORG	0.9474	0.9711	0.9591	4325
0	0.9675	0.9751	0.9713	5897
E-NAME	0.9903	0.9107	0.9488	112
M-TITLE	0.9433	0.8829	0.9121	1922
E-ORG	0.9145	0.9096	0.9121	553
label_level	0.9559	0.9560	0.9557	15100
NAME	1.0000	0.8839	0.9384	112
ORG	0.9310	0.8788	0.9042	553
TITLE	0.9555	0.9171	0.9359	772
all_ent	0.9493	0.8998	0.9239	1437

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，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。假设文本总共被分为n类，我们设 $t_1, t_2, \dots, t_n, f_1, f_2, \dots, f_n$ 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	t_1	f_1	...	f_1
2类	f_2	t_2	...	f_2
...
N类	f_n	f_n	...	t_n

记 $A = \text{confusion_matrix}$

定义 $\text{recall}(x) = A(x,x) / [A(1,x) + A(2,x) + \dots + A(n,x)]$ 为x类的召回率(recall);

定义 $\text{precision}(x) = A(x,x) / [A(x,1) + A(x,2) + \dots + A(x,n)]$ 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 $\text{precision}(1) = 100\%$ ，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 $\text{recall}(1) = 3\%$ ，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 $\text{precision}(1) = 10 / 80 = 12.5\%$ 。而真正的1类总共有10个，因此 $\text{recall}(1) = 100\%$ ，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，这也是最常用的指标 $F1\text{-score} = r * p / (p + r)$ ，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，进行对比分析并得出结论（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。