PJ2 命名实体标注 实验报告

李菲菲 18307110500 2022/12/15

I. 背景介绍

NER 实体标注任务：是一种序列标注任务，实体类别有 4 种：人名 PER, 未知 LOC, 组织 ORG 和杂项 MISC，每个词的 BIO 格式的标注为 3 种：实体的开头 B, 实体非开始词 I, 和非实体 O。

序列标注模型：经典的基于统计的模型有隐式马尔可夫模型 (HMM)，最大熵马尔可夫模型 (MEMM)，以及条件随机场 (CRF). 常用于学习上下文特征的基本神经网络有 CNN、RNN 及其变体，和基于注意力机制的 Transformer、Bert。

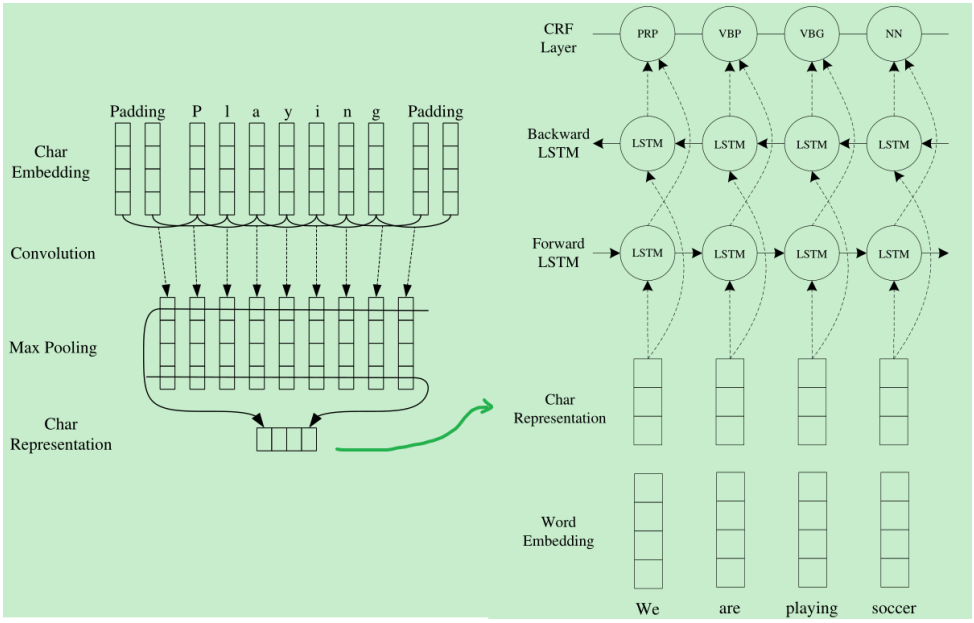
条件随机场(CRF)：基于预定义特征的线性链条件随机场可以形式化表示为

$$P(y|x) = \frac{1}{Z(x)} \exp \sum_{k=1}^{K} w_k f_k(y, x)$$

其中w_k为特征函数f_k的权重，是可学习参数；$Z(x)$是归一化因子，保证对数加权和符合概率的定义。直观理解与 HMM 的区别，即所有的可观测随机变量$x = x_1, \dots, x_T$都可以影响状态$y = y_1, \dots, y_T$. 给定 CRF，可以使用维特比算法解码出状态序列和对应概率（状态y_t即 tag of word t）：

$$v_i(t) = \max_{i=1}^{N} v_{t-1}(i) \sum_{k}^{K} w_k f_k(y_{t-1}, y_t, X, t);$$

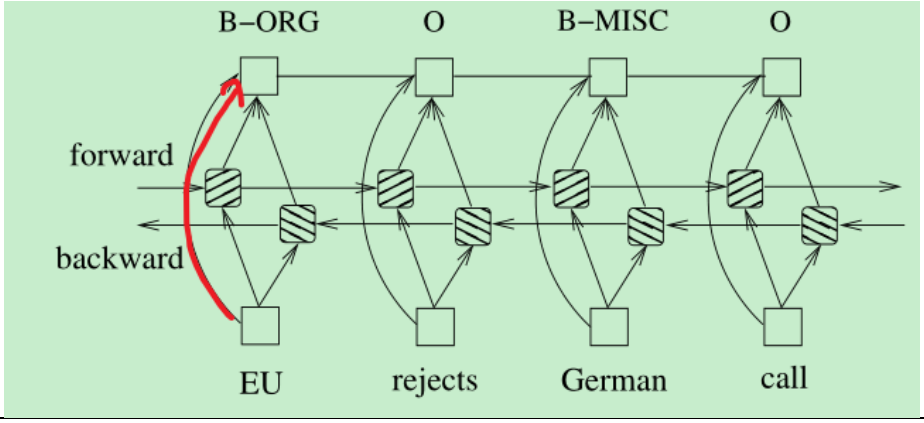
$$\Pr(\{y_1, \dots y_T\}^*) = \max_{i=1..N} v_i(T)$$



Character-level representation: 一般只依赖词嵌入和词表征的方法会在 OOV 单词上发生比较大的性能下降，很多工作的实验表明增加字符级别表征能够显著提升序列标

注任务的表现。根据如上图所示[2]的模型架构，一般的字符表征的学习方法为： 1) 使用嵌入层，将输入的词序列对应的经过 padding 的字符序列转换为字符嵌入，2) 学习 CNN 模型得到字符表征，3) 拼接到词向量中得到字符、词级别的输入向量。然后就可以按照一般的学习方法，将向量送入 RNN 中学习上下文表征、进行序列标注。

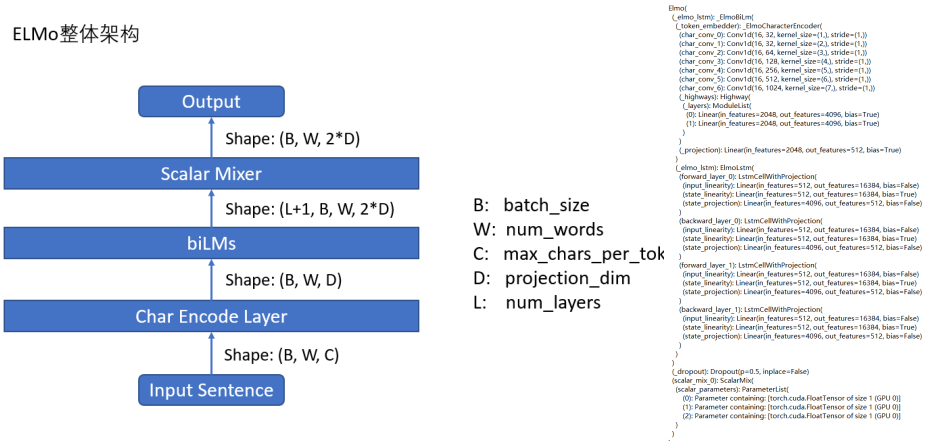
使用 CRF 和 LSTM 进行序列标注：下图所示为[1]BiLSTM-CRF 的模型架构:



对于 CRF 而言，是将手动设计的特征函数改为数据驱动学习的神经网络；对于 LSTM 而言是在输出层通过 CRF 引入了关于 Tag 的先验知识。

超参搜索：常见的超参搜索方法有网格搜索，随机搜索和贝叶斯优化。网格搜索尝试所有超参数的可能组合，是一种穷举的暴力算法；随机搜索从定义的搜索空间中随机选择一组超参进行模型训练；优化方法如贝叶斯优化方法，通过预定义一个收益函数，可以根据已经实验的参数组合预测下一个可能带来最大收益的组合。

上下文自适应的词嵌入 ELMo[4]：相比于 Word2Vec 和 GloVE 这种静态的词嵌入，ELMo 可以 1) 根据上下文学习到单词的复杂语法/语义特征，和 2) 建模单词的多义性。模型整体上是字符卷积+双向语言模型+加权求和的 Mixer 的结构，直接得到 word embedding。



II. 实验内容

训练、验证和测试集的样本数、标签数、字符、词的统计信息如表所示

	Train	Valid	Test	Total
#sample	14036	3250	3453	---
#tag	---	---	---	11
#char	---	---	---	75
#uniq word	---	---	---	17488

1. 消融实验：

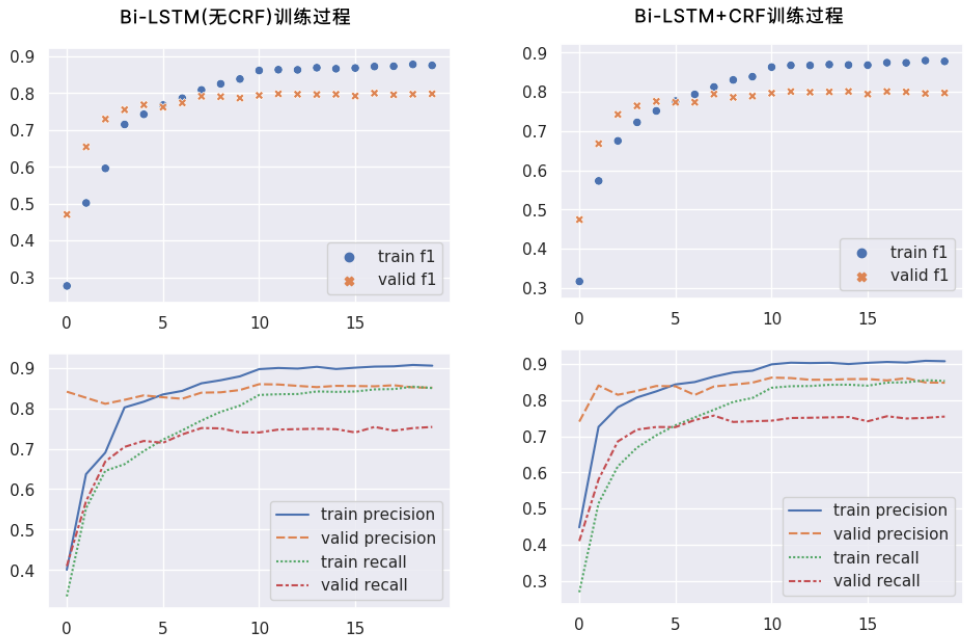
评估 CRF decoder 的作用

模型架构[1]：使用 GloVE 词向量嵌入，Bi-LSTM 作为词表征的编码器，CNN 作为字符表征的编码器。空白对照组直接将词表征后接线性层，输出 19 维中概率最大的标记；实验组将线性层的输出 logits 喂入 CRF，由 CRF 输出 seq2seq 标记。

参数：BiLSTM 的隐单元个数为 512，学习率 1e-3，丢弃率 0.5，迭代 20 次；CNN 的卷积核大小为 3. 实验结果如下表所示，实验过程的数据如下图所示。

结论：CRF 训练的速度非常慢；测试集上的 F1 略微上升；对比训练曲线，CRF 会更稳定一些。

Char-Word-decoder	Loss	F1 score	Precision	Recall	Time/Epoch
Null-BiLSTM	0.1667	**0.7999**	0.8542	0.7538	5.54s
Null-BiLSTM-CRF	0.1510	**0.8009**	0.8580	0.7534	130.98s



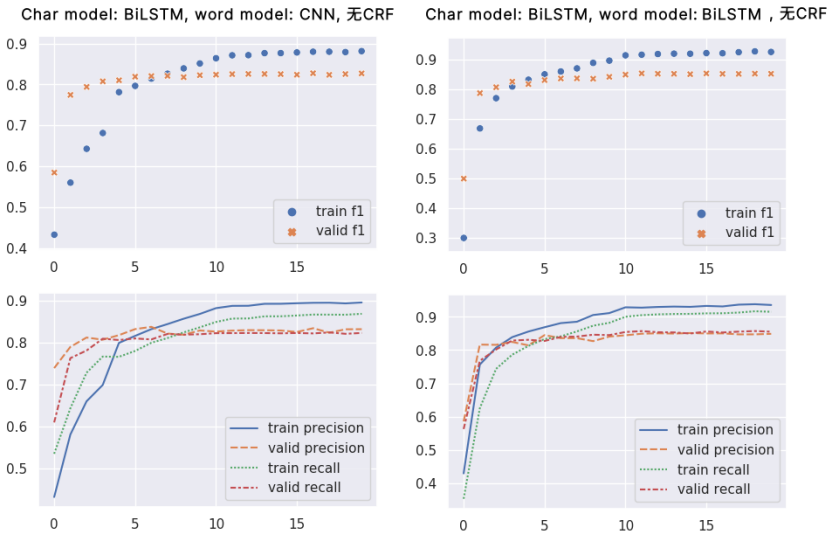
评估 char representation 的作用

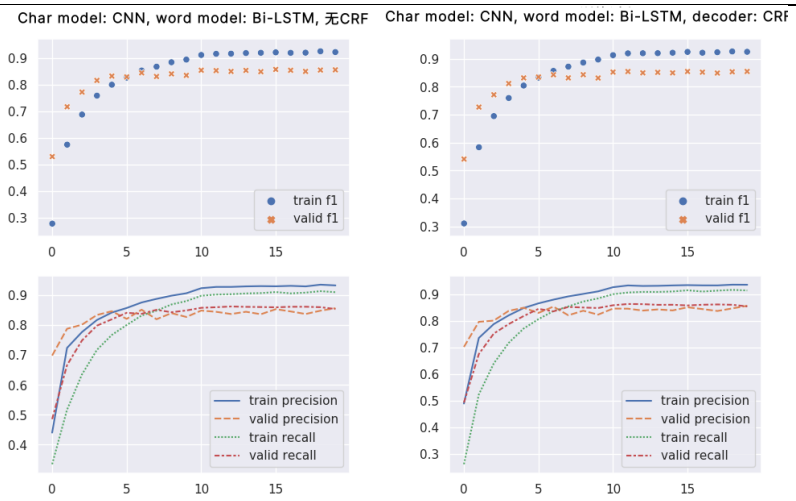
具体实现[2]：使用 Char Embedding 和 Char Encoder 层得到字符表征。对于两种结构 Encoder 结构进行实验对比。

参数： 字符嵌入为 30 维，字符特征为 10 维字符表征；两个维度均为超参；词嵌入使用 100 维 Glove。

结论：使用字符卷积和双向 LSTM 学习上下文表征的方法作为 encoder 效果最佳；decoder 使用 CRF 对损失有帮助，但是时间代价太大。

Char-Word-Decoder	Loss	F1 score	Precision	Recall	Time/Epo
Null-BiLSTM	0.1667	**0.7999**	0.8542	0.7538	5.54s
BiLSTM-CNN	0.1188	**0.8275**	0.8344	0.8220	16.26s
BiLSTM-BiLSTM	0.1022	**0.8529**	0.8492	0.8571	15.28s
CNN-BiLSTM	0.1012	**0.8566**	0.8539	0.8601	5.38s
CNN-BiLSTM-CRF	0.0924	**0.8545**	0.8568	0.8544	130.15s





2. 超参搜索

由于使用 CRF 的运行代价较大，因此仅使用 BiLSTM 进行超参搜索；由于网格法比较耗时，因此使用随机搜索

待搜索超参数和搜索空间

```
{
    'dp': [0.1, 0.3, 0.5, 0.6, 0.7], # drop out
    'bs': [10, 32, 64, 128, 256], # batch size
    'lr': [1e-4, 1e-3, 1e-2, 1e-1],
    'dow': [32, 64, 128, 256, 512], # dim of word feature
    'doc': [10, 20, 30, 40, 50], # dim of char feature
    'ks': [3,5], # kernel size of cnn
}
```

随机搜索

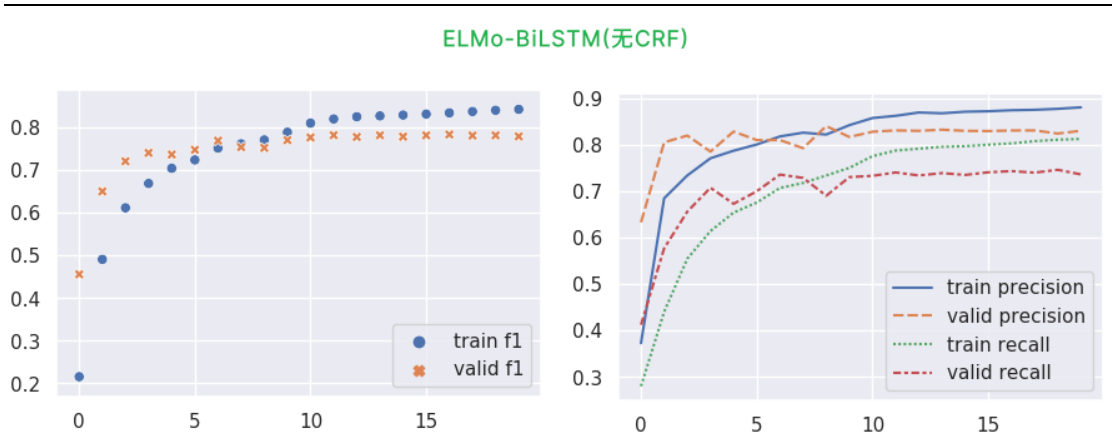
随机搜索 20 次的结果如下表，根据训练 5 次得到的最优测试集上的 F1 分数排序，综合考虑训练时间，最终选择 idx=1 的参数组

	T_train/epo	F1_test	Prec_test	Rec_test	dp	bs	lr	dow	doc	ks
0	32.896852	0.839322	0.849289	0.831880	0.1	32	0.0010	128	10	3
1	31.901880	0.833626	0.835359	0.840172	0.5	128	0.0010	512	10	3
2	24.081874	0.813706	0.833012	0.798636	0.3	128	0.0100	32	50	3
3	6.678991	0.810667	0.836838	0.789804	0.1	256	0.0010	128	30	5
4	32.712935	0.808420	0.813633	0.804612	0.6	32	0.0100	64	10	3
5	42.558396	0.808174	0.827229	0.792345	0.7	64	0.0010	128	20	5
6	90.510471	0.804901	0.810717	0.800904	0.1	10	0.0100	32	50	3
7	18.276210	0.799205	0.825254	0.778414	0.1	256	0.0010	128	30	5
8	66.326919	0.790374	0.824921	0.764346	0.5	10	0.0001	512	30	5
9	53.577368	0.754568	0.809174	0.720752	0.3	10	0.0001	64	20	3
10	53.907791	0.725933	0.802041	0.682211	0.3	10	0.0001	64	20	3
11	56.370221	0.601196	0.809552	0.544043	0.3	32	0.0001	128	30	5
12	72.576573	0.580801	0.743787	0.534276	0.3	64	0.0001	256	40	3
13	60.377568	0.548376	0.739120	0.504204	0.1	128	0.0001	512	50	3
14	44.236082	0.429423	0.577388	0.380614	0.7	64	0.0001	512	20	5
15	24.043629	0.322726	0.571486	0.271703	0.1	256	0.0001	512	20	5
16	32.819953	0.309197	0.325970	0.298737	0.5	32	0.1000	32	10	3
17	53.468475	0.162747	0.147667	0.222553	0.3	10	0.1000	128	20	3
18	52.882480	0.100477	0.091700	0.111111	0.7	10	0.1000	64	20	5
19	43.173185	0.100477	0.091700	0.111111	0.7	10	0.1000	128	10	3

3. 使用 ELMo

将字符嵌入和词嵌入改为由 ELMo 学习得到。实体标记的实验结果如下：

Char-Word-Decoder	Loss	F1 score	Precision	Recall	Time/Epo
Null-BiLSTM	0.1667	**0.7999**	0.8542	0.7538	5.54s
CNN-BiLSTM	0.1012	**0.8566**	0.8539	0.8601	5.38s
ELMo-BiLSTM	0.1787	**0.7818**	0.8292	0.7418	29.35s



III. 总结

有效的技巧：

- 比较有效的技巧是选择一组好的超参和引入 Char-level representation。
- 随机搜索方法简单且效果显著，并且可以综合评估运行时间和收敛速度（未列出默认参数和超参搜索选择之后的对比实验，但是可以根据随机搜索的表格中观察得到）
- 字符卷积引入的参数量少，且效果明显：相比于不使用 CNN，验证集上的 F1 score 可以提升 6%（0.7999->0.8566）

关于 CRF：

- CRF 在没有使用 Char 表征时可以提升标注的表现，可以提升 0.1%的 F1 score；
- CRF 缺点在于训练时间耗时比较长，目前还未尝试 CRF 的训练加速方法；
- CRF 在使用了 Char 表征后，F1 score 与不使用时相当，可以减少约 0.9%的损失

关于 ELMo：

- 仅使用 ELMo 而不使用词嵌入时，模型的 F1 大致在 0.1，推测 ELMo 的使用上出现了 bug，还未检查出具体出错位置。
- ELMo 的模型不小，参数量很大，因此无法在显存有限的 GPU 上进行 fine tune。

IV. Reference

[1] Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging[J]. arXiv preprint arXiv:1508.01991, 2015.

[2] Ma, Xuezhe and Eduard H. Hovy. “End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF.” ArXiv abs/1603.01354 (2016): n. pag.

[3] Lample, Guillaume, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami and Chris Dyer. “Neural Architectures for Named Entity Recognition.” North American Chapter of the Association for Computational Linguistics (2016).

[4] Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee and Luke Zettlemoyer. “Deep Contextualized Word Representations.” North American Chapter of the Association for Computational Linguistics (2018).

[5] https://blog.csdn.net/Magical_Bubble/article/details/89160032