

命名实体识别的基本方法介绍及实现

孙相会

1971654

一 摘要

最近通过阅读一些经典顶会论文，对命名实体识别的一些基本方法有了一个大致的了解。本文主要介绍了一些在命名实体识别方向上的一些基本的同时也是非常典型的方法，包括隐马尔科夫模型 (HMM) 和条件随机场 (CRF)，以及基于深度学习的一些神经网络架构如双向长短期记忆循环神经网络 (BiLSTM)，以及用卷积神经网络 (CNN) 或者 BiLSTM 来提取字符层面的特征后联合预训练的词向量送进 BiLSTM 这一架构。实验的数据和代码放在<https://github.com/xianghuisun/NERs>

关键词：HMM; CRF; BiLSTM; CNN

二 简短介绍

命名实体识别是序列标注类的任务之一，目的是对于给定的句子预测出句子中每一个单词对应的标签。如图所示，B-ORG 代表的是一个单词，它是一个组织名的开头。B-PER 代表的是一个单

```
the 0
Commission B-ORG
's 0
chief 0
spokesman 0
Nikolaus B-PER
van I-PER
der I-PER
Pas I-PER
told 0
a 0
news 0
briefing 0
. 0
```

词，它是人名的开头，I-PER 则代表人名结尾。o 代表其它类型单词，也就是这个单词既不是组织名，也不是地点名同时也不是人名。

我用的数据集是最经典的 CoNLL2003 命名实体识别数据集，共有九种标签。

```
I-ORG
O
B-MISC
I-LOC
I-PER
I-MISC
B-LOC
B-PER
B-ORG
```

三 模型简介

隐马尔科夫模型

隐马尔科夫模型是可以用于序列标注的一种时序概率图模型，描述的是由一组隐藏的状态序列（在 ner 里面就是各个单词对应的标签）生成一个观测序列（就是各个单词）。序列的每一个位置就是一个时刻。

假设 x_1, x_2, \dots, x_n 代表句子中的 n 个单词， y_1, y_2, \dots, y_n 就代表这 n 个单词对应的标签。要计算的是 $\max(P(y_1, y_2, \dots, y_n | x_1, x_2, \dots, x_n))$ 的概率。HMM 通过贝叶斯公式

$$P(y_1, y_2, \dots, y_n | x_1, x_2, \dots, x_n) = \frac{P(x_1, x_2, \dots, x_n | y_1, y_2, \dots, y_n) P(y_1, y_2, \dots, y_n)}{P(x_1, x_2, \dots, x_n)}$$

将计算 $P(y_1, y_2, \dots, y_n | x_1, x_2, \dots, x_n)$ 改为计算 $P(x_1, x_2, \dots, x_n | y_1, y_2, \dots, y_n) P(y_1, y_2, \dots, y_n)$ 。对于 $P(x_1, x_2, \dots, x_n)$ 是给定的输入，可以忽略。

HMM 有两个基本的假设

1. 齐次马尔科夫假设，即当前时刻的状态仅仅依赖于前一个时刻的状态。也就是说 t 时刻第 i 个单词的标签仅仅依赖于第 $i-1$ 个单词对应的标签。因此 $P(y_1, y_2, \dots, y_n)$ 可以写成

$$P(y_1, y_2, \dots, y_n) = P(y_1) p(y_2 | y_1) p(y_3 | y_2) \cdots P(y_n | y_{n-1})$$

2. 观测独立性假设，即当前时刻的观测仅仅依赖于当前时刻的状态，与其它时刻的状态无关。也就是说 t 时刻，单词 i 出现的概率仅仅与它所对应的标签有关，与其它单词和标签无关。因此 $P(x_1, x_2, \dots, x_n | y_1, y_2, \dots, y_n)$ 由观测独立性假设就可以得出

$$P(x_1, x_2, \dots, x_n | y_1, y_2, \dots, y_n) = P(x_1 | y_1) P(x_2 | y_2) \cdots P(x_n | y_n)$$

在 HMM 中 $P(x_i | y_i)$ 也叫发射概率，意思是在标签为 y_i 的时候，单词 x_i 出现的概率，显然这可以通过训练数据求出来。 $P(y_i | y_{i-1})$ 也叫转移概率，指的是当前单词被标记为 y_{i-1} ，而它的下一个单词被标记为 y_i 的概率。这两个概率都可以通过遍历训练数据集中每一个句子以此来统计相应的频数，用频率来代替概率。

从而 $P(y_1, y_2, \dots, y_n | x_1, x_2, \dots, x_n)$ 就可以写成 $P(x_1 | y_1) p(y_1) p(x_2 | y_2) p(y_2 | y_1) \cdots P(x_n | y_n) P(y_n | y_{n-1})$ ，HMM 有三个概率矩阵，用 A, B, π 来表示。这三个矩阵分别记录着状态转移概率，发射概率以及初始概率。如下图 1 和图 2 所示。

这就是每一个标签转移到下一个标签的概率，例如从 I-PER 转移到 O 的概率是 0.943921。这就是每一种标签的初始概率，例如所有句子的第一个单词是 O 的概率是 0.5791372。发射概率矩阵

	B-ORG	I-LOC	I-MISC	O	B-MISC	I-PER	B-PER	I-ORG	B-LOC
B-ORG	2.894821e-03	1.608234e-08	1.608234e-08	0.595851	9.649404e-04	1.608234e-08	4.824702e-04	3.996462e-01	1.608234e-04
I-LOC	9.033420e-04	1.047877e-01	9.033420e-08	0.888889	2.710026e-03	9.033420e-08	9.033420e-08	9.033420e-08	2.710026e-03
I-MISC	6.743736e-03	9.633909e-08	2.861271e-01	0.698458	3.853563e-03	9.633909e-08	2.890173e-03	9.633909e-08	1.926782e-03
O	2.426802e-02	6.399795e-10	6.399795e-10	0.888842	1.830981e-02	6.399795e-10	3.317014e-02	6.399795e-10	3.541007e-02
B-MISC	9.373168e-03	2.929115e-08	2.513181e-01	0.710896	9.080257e-03	2.929115e-08	1.698887e-02	2.929115e-08	2.343292e-03
I-PER	2.298322e-08	2.298322e-08	2.298322e-08	0.943921	2.298322e-08	5.607905e-02	2.298322e-08	2.298322e-08	2.298322e-08
B-PER	1.530222e-08	1.530222e-08	1.530222e-08	0.343841	1.530222e-04	6.555470e-01	1.530222e-08	1.530222e-08	4.590665e-04
I-ORG	1.651982e-03	2.753304e-08	2.753304e-08	0.660793	8.259911e-04	2.753304e-08	8.259911e-04	3.356277e-01	2.753304e-04
B-LOC	1.422475e-03	1.480797e-01	1.422475e-08	0.845377	3.840683e-03	1.422475e-08	1.422475e-04	1.422475e-08	1.137980e-03

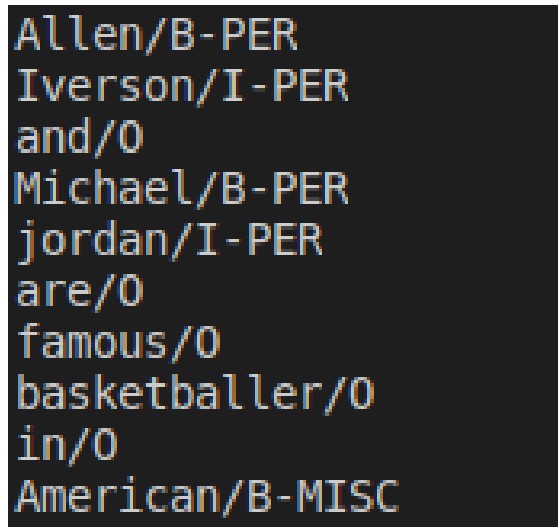
图 1 状态概率转移矩阵

	O
B-ORG	1.747223e-01
I-LOC	7.213966e-09
I-MISC	7.213966e-09
O	5.791372e-01
B-MISC	3.614197e-02
I-PER	7.213966e-09
B-PER	9.702784e-02
I-ORG	7.213966e-09
B-LOC	1.129707e-01

图 2 初始概率转移矩阵

比较大就不给出图了, $B[i][j]$ 的意思就是 j 这个单词被标记为标签 i 的概率。HMM 的预测问题通常利用 *viterbi* 算法, 核心思想是最优路径如果通过节点 v_i , 那么从初始节点 v_0 到节点 v_{i-1} 的路径也一定是最优路径。根据这一思想, 就可以不必穷举所有可能的路径再找出概率最大的路径了。只需要从前向后, 每一步都是找概率最大的路径上的节点, 那么这个节点一定在最优路径上。

现在假设给出这样的句子 "Allen Iverson and Michael Jordan are famous basketballer in American"。首先在发射概率矩阵中找到单词 Allen 被标记为这九种标签的概率, 分别乘以对应的九种标签的初始概率, 同理找到单词 Iverson 在发射概率矩阵中的九种概率, 用前面的 Allen 的九种概率分别乘以九种状态转移概率。例如以 I-PER 为第一个标签, 那么 Allen 的九种概率分别乘以对应的转移到 I-PER 的概率 (假设 Allen 被标记为 B-PER 的概率乘以 $A[B-PER][I-PER]$ 大于 Allen 其他标签的概率乘以相应的转移到 I-PER 的概率, 那么就认为 Iverson 被标记为 I-PER 的情况下是从 Allen 被标记为 B-PER 转移过来的), 还有其余的九种情况也都要计算出来。由此可见, 从第一个单词到第二个单词之间有九种路径, 同理第二个单词到第三个单词也有九种可能的路径 (可见如果不是 *viterbi* 算法的思想, 那么每两个单词之间就有 81 种路径)。以此类推到最后一个单词 American, 假设算得 American 被标记为 O 的概率最大, 而且是从前一个单词 in 被标记为 O 的情况下转移过来的, 那么就认为 American 被标记为 O, O 就是最优路径上的最后一个节点, 前一个节点也是 O, 以此从后向前推, 就得到了最优路径。下图就是结果



```
Allen/B-PER
Iverson/I-PER
and/O
Michael/B-PER
jordan/I-PER
are/O
famous/O
basketballer/O
in/O
American/B-MISC
```

神经网络模型

最常用的用于序列标注任务的神经网络架构就是 BiLSTM-CRF。事实上 BiLSTM 已经可以很好的建模输出的标签和输入的单词之间的关系, 但是在序列标注问题中, 输出的标签之间是有一定关系约束的, 例如 B-PER 后面接 I-LOC 就很不合理, 仅用 softmax 层分类不会考虑到这种约束关系, 而 CRF 用了一个状态概率转移矩阵可以很好的解决这个问题。

	B-MISC	B-LOC	I-LOC	I-ORG	I-PER	O	I-MISC	
B-MISC	-1.263990	-1.345764	-4.079100	-4.067087	-3.698221	-0.198945	2.326010	-0.7
B-LOC	-0.220864	-2.315734	2.774562	-3.196770	-3.412635	0.309631	-3.054955	-1.1
I-LOC	-0.576073	-0.954766	1.776856	-3.255663	-2.157637	0.206763	-1.765944	-1.4
I-ORG	-1.828231	-2.419062	-3.448389	2.039526	-3.740747	-0.556875	-3.966177	-2.9
I-PER	-2.142168	-1.758618	-1.485961	-3.182490	0.636386	-0.024485	-2.191404	-2.3
O	0.887695	0.827331	-3.137301	-3.970531	-2.978096	0.281402	-3.587938	0.6
I-MISC	-1.054047	-1.255401	-2.184670	-3.385370	-2.433785	-0.293422	2.505166	-0.8
B-ORG	-1.697226	-2.567379	-3.493662	2.514851	-3.665789	-0.314679	-3.532480	-2.2
B-PER	-2.609368	-2.198691	-3.157818	-3.632676	2.289206	-1.082377	-2.942393	-3.0

图 3 训练后的 CRF 概率转移矩阵

从图中可以看到，矩阵中的数据值不代表概率，但是可以明显的看出来 B-LOC 转移到 I-LOC 的值是 2.774562，B-MISC 转移到 I-MISC 的值是 2.326010，而 B-LOC 转移到 I-MISC 的值是-3.054955，B-MISC 转移到 I-LOC 的值是-4.079100。这说明了 CRF 的概率转移矩阵学习到了输出层面上各个标签的转移概率之间的关系。下表就是所有模型的结果汇总。

模型架构	correct/total	f1 值
HMM	0.895079	0.6140245
BiLSTM	0.89199903	0.62274653
BiLSTM-CRF	0.9028061168	0.63009822302
BiLSTM-CRF(glove)	0.896527549	0.63349347
BiLSTM-CNN-CRF(glove)	0.9190669149	0.67904301964
BiLSTM-BiLSTM-CRF(glove)	0.9554123	0.81911336

表 1 模型结果汇总

BiLSTM 就是仅用 BiLSTM 和 softmax 分类层。BiLSTM-CRF 意思就是输出层分类用的是 CRF。后面带有的 glove 的意思是利用了 glove 预训练词向量，没有的就代表嵌入矩阵是随机初始化的。BiLSTM-CNN-CRF 意思是考虑到了字符层面的表示，用 CNN 来提取字符层面的特征，BiLSTM-BiLSTM-CRF 就是先用一层 BiLSTM 来提取字符层面的特征，然后联合预训练的词向量再送进 BiLSTM 内。correct/total 的意思是统计了所有的预测正确的标签数量/所有的标签的数量，f1 值就是根据定义由精确率和召回率所计算的。

可以看到最好的结果来自于模型结构为 BiLSTM-BiLSTM-CRF(glove) 模型，f1 值达到 0.81911336，当然 f1 值的计算是我根据定义自己写的计算方法，可能有误，但是就算有误，其他模型的架构也是同样的方法计算 f1 值，因此仍然可以说明 BiLSTM-BiLSTM-CRF(glove) 这种模型的结果是优于其它的模型。

参考文献

- [1] Neural Architectures for Named Entity Recognition
- [2] Bidirectional LSTM-CRF Models for Sequence Tagging
- [3] End-to-end Sequence Labeling via Bidirectional LSTM-CNNs-CRF
- [4] Named Entity Recognition with Bidirectional LSTM-CNNs
- [5] A Survey on Deep Learning for Named Entity Recognition
- [6] 李航 统计学习方法