命名实体识别的多模型对比

尹梓琦(1120182493)[1] 崔冬航(1120182424)[2] 杨璐铭(1120182903)[3]

Abstract

命名实体识别任务已经被广泛研究,进来随着深度学习技术的深入,许多深度学习模型,性能上已经超越了传统的机器学习模型.本文,我们将应用多种方法,其中包括CRF,BiLSTM,BERT,在统一的数据集上进行对比,并总结出各自模型的优劣。本文的代码和使用说明可以在https://github.com/Heisenberg-Yin/Chinese-NER获得.

1. Introduction

命名实体识别任务是自然语言处理的一大基本任务之一，已经有许多研究方法。在(Zong)一书中总结了主流的统计学习方法，包括隐马尔科夫模型(Nabende et al., 2008),支持向量机(Arora et al., 2019),条件随机场(Song et al., 2019)。在这些书籍和论文中，统计学习方法有优秀的性能，较快的速度，但是缺点在于缺乏提升途径。在统计学习方法进入瓶颈时，深度学习方法开始爆发，并且在命名实体识别任务中表现出优异的性能。其中RNN(LSTM)(Sherstinsky, 2018)和BERT(Devlin et al., 2018)是最有影响力的两个工作。我们选择CRF,BiLSTM和BERT三个模型进行对比。

条件随机场(Conditional Random Field, CRF)是一种基于遵循马尔可夫性的无向概率图模型。它最早于2001年由(Song et al., 2019)提出，结合了最大熵模型和隐马尔可夫模型的特点，在许多序列标注任务如分词、词性标注和命名体识别任务，取得了很好的效果。相比于解决此类问题常见的隐马尔可夫模型（Hidden Markov Model，HMM）和最大熵马尔可夫模型（Maximum Entropy Markov Model，MEMM），CRF模型不仅解决了HMM的独立性假设导致的问题，而且对全局概率进行统计，不局限于局部最优解，从而解决了MEMM标注偏置的问题。

循环神经网络(RNN)解决序列问题非常适合，而长短期注意力网络(LSTM)又是RNN中性能最为优秀的一个模型。LSTM有效的解决了在RNN网络中梯度消失的问题，使得深度网络的性能大大增强。BiLSTM是LSTM的一个优化，获得更多的序列信息。

BERT(Devlin et al., 2018)模型是由Google公司于2018年提出的优秀预训练模型，在NLP领域刷新一系列下游任务的最佳成绩。他在模型Transformer(Vaswani et al., 2017)上进行改进，提出了一个双向Transformer的预训练模型。Transformer型模型和RNN型模型并不一样，前者丢弃了后者的序列特征，通过用位置编码(Positional Encoding)来替代，并且取得了更好的效果。

在本文中，我们的主要目的是加深对于命名实体任务的了解，自己实现多种模型，并且对他们的性能，速度等进行对比。本小组人员信息和分工如Table.1。

本文的组织结构如下，第二段会介绍符号表示和数据集，第三段会介绍模型，在第四段介绍我们的实验结果。在第五段，对实验进行总结，并且提出可能的改进。引文和补充材料在文末，包括实验过程训练图等。

2. Preliminaries

中文命名实体识别任务. 命名实体识别(Named Entity Recognition, NER)，主要任务是识别出文本中的人名、地名等专有名称和有意义的时间、日期等

Table 1. 人员分工与信息

姓名	学号	任务	联系电话	邮箱
尹梓琦	1120182493	实现BERT模型板块，对接模型，编写论文	18801028161	545068655@qq.com
崔冬航	1120182424	实现CRF模型板块，对接模型	18813036638	987243094@qq.com
杨璐铭	1120182903	实现LSTM板块，对接模型	18890663818	1120182903@bit.edu.cn

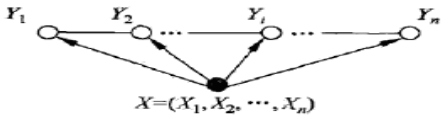


图 11.4 线性链条件随机场

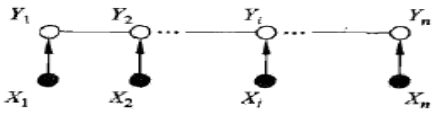
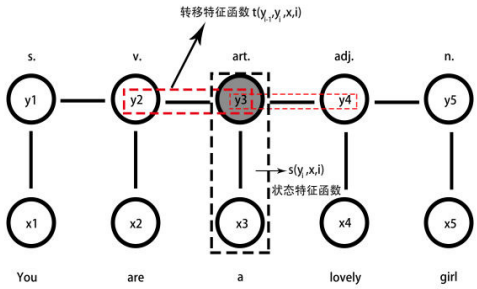


图 11.5 X 和 Y 有相同的图结构的线性链条件随机场

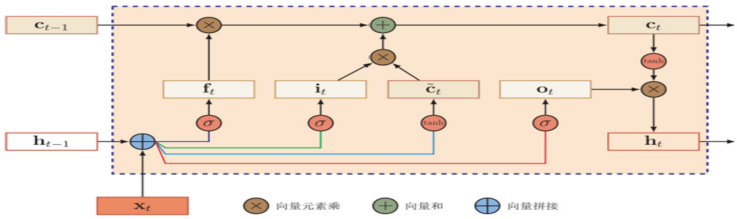
(a) CRF 1



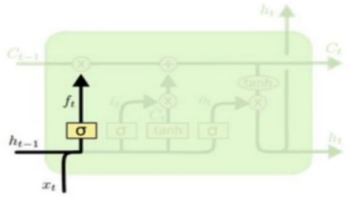
(b) CRF 2

数量短语并加以归类。命名实体识别系统是重要的自然语言处理问题：一方面，命名实体识别可以帮助识别未登录词，根据SIGHAN Bakeoff的数据评测结果，未登录词造成的分词精度损失远大于歧义。这体现了NER问题对于下游问题的间接帮助，另一方面，对关键词提取等任务来说，命名实体的类别是非常有用的文本特征，可以直接用于文本分类等问题的输出，这是NER问题对于下游问题的直接帮助。

符号. 我们将输入的词向量矩阵表示为X，可学习矩阵表示为W，$Attention$表示注意力机制，Att表示得到的注意力值。大写字母表示矩阵，加粗小写字母表示向量。

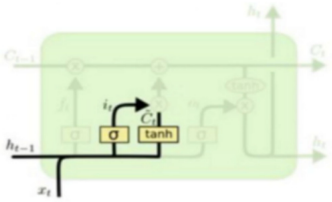


(c) LSTM.1



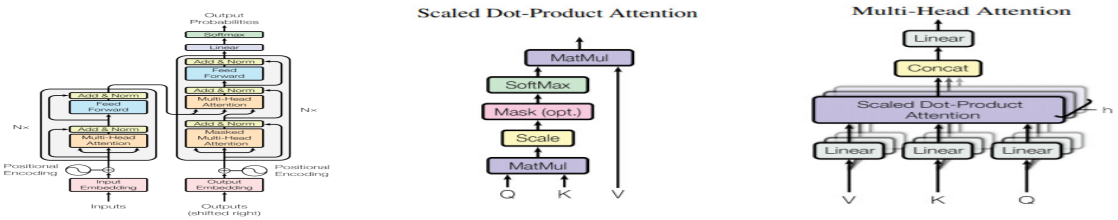
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

(d) LSTM.2



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

(e) LSTM.3



(f) 1.1 (g) 1.2 (h) 1.3

Figure 1. Transformer Model

3. Model

CRF. 条件随机场(CRF)是一种判别式无向图模型

定义. 令$G =< V, E >$表示结点和标记变量y中元素一一对应的无向图，y_v表示与结点v对应的标记变量，$n(v)$表示结点v的邻接结点，若图G的每个变量y_v都满足如下马尔可夫性：

$$P(y_v \mid x,\ y_{V/\{v\}}) = P(y_v|x,\ y_{n(v)})$$

则$(y,\ x)$构成一个条件随机场(Conditional Random Field, CRF)。

链式条件随机场.(chain-structured CRF) 如图1(a),条件概率$P(y|x)$被定义为:

$$P(y \mid x) = \frac{1}{Z}\exp(\sum_{j}\sum_{i=1}^{n-1}\lambda_j t_j(y_{i+1}, y_i, x, i)+ \sum_{k}\sum_{i=1}^{n}\mu_k s_k(y_i, x, i)) \tag{1}$$

$\lambda_j,\ \mu_k$为参数，Z为规范化因子:

$$Z = \sum_{y}\exp(\sum_{j}\sum_{i=1}^{n-1}\lambda_j t_j(y_{i+1}, y_i, x, i)+ \sum_{k}\sum_{i=1}^{n}\mu_k s_k(y_i, x, i)) \tag{2}$$

转移特征函数 $t_j(y_{i+1}, y_i, x, i)$：为转移特征函数(transition feature function)，刻画相邻标记变量$y_{i+1},\ y_i$之间的相关程度以及观测序列x对它们的影响。

状态特征函数 $s_k(y_i, x, i)$：状态特征函数（status feature function），刻画观测序列x对标记变量y_i的影响

则求解条件概率$P(y\ |x)$的问题，就转化成如何得到一组转移特征函数t_j、一组状态特征函数s_k及其分别对应的参数λ_j, μ_k。

CRF模型的简化 转移特征函数$t_j(y_{i+1}, y_i, x, i)$和状态特征函数$s_k(y_i, x, i)$统一用$F(y,\ x)$表示。同时将转移特征的权重λ_j与状态特征的权重 μ_k统一用θ表示，于是模型可以简写为：

$$\begin{aligned} P(y|x) &= \frac{1}{Z}\exp(\theta \cdot F(y_t,\ y_{t-1},\ x)) \\ &= \frac{1}{Z}\prod_{i=1}^{T}\Psi_t(y_t, y_{t-1}, x_t) \end{aligned} \tag{3}$$

$$Z = \sum_{y}\prod_{i=1}^{T}\Psi_t(y_t, y_{t-1}, x_t) \tag{4}$$

LSTM. LSTM的全称是Long Short-Term Memory，它是RNN（Recurrent Neural Network）的一种，。LSTM由于其设计的特点，非常适合用于对时序数据的建模，如文本数据。BiLSTM是Bi-directional Long Short-Term Memory的缩写，是由前向LSTM与后向LSTM组合而成。两者在自然语言处理任务中都常被用来建模上下文信息。如图Figure.1(c)。RNN只有两个输入(xt和ht-1)和一个输出，将过去的输出和当前的输入连接到一起，通过tanh激活来控制两者的输出，它只考虑最近时刻的状态。

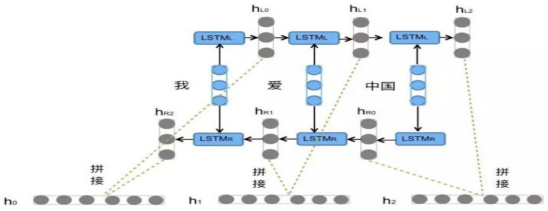
但LSTM增加了一路输入与一路输出，将信息分成三部分内容：
1.哪些细胞状态应该被遗忘
2.哪些新的状态应该被加入
3.根据当前的状态和现在的输入，输出是什么
LSTM的cell结构是通过三个门进行控制的

遗忘门. 如图Figure.1(d),LSTM第一步是决定要从上一个时刻的状态中丢弃什么信息，其是一个Sigmoid全连接组成的前馈神经网络的输出管理，将这种操作称为遗忘门(forget get layer)。这个全连接的前馈神经网络的输入是h_{t-1}和X_t组成的向量（向量拼接），输出是f_t向量，f_t向量由1和0组成，1表示能够通过，0表示不能通过。

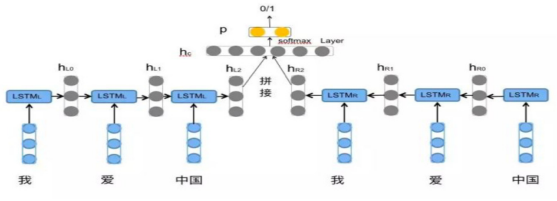
输入门. 如图Figure.1(e),输入门决定了哪些输入信息要保存到神经元状态中，首先由sigmoid层的全连接前馈网络，i(t)是决定那些值将被更新，然后tanh层输出Ct，添加到当前时刻的神经元状态中，最后根据两个神经网络结果创建一个新的神经元状态。

输出门. 如图Figure.**??**,输出门就是这个神经元最终输出什么，此时的输出是根据第三步的Ct状态来计算的，即根据一个 sigmoid层的全连接前馈神经网络过滤一部分Ct状态作为当前时刻的神经元输出。计算过程是 1.首先通过sigmoid层生成一个过滤向量。 2.然后通过一个tanh函数计算当前时刻的Ct状态向量ht(即将每个值变换到[-1,1]间) 3.通过sigmoid层的输出向量过滤tanh函数结果，即为当前时刻神经元的输出。

BILSTM. 如图Figure.2(a),我们通过正向的LSTM可以得到一句话的输出序列，如输入“我爱中国这句话”。
而BILSTM会正向依次输入“我”，“爱”，“中国”，得到三个向量h0,h1,h2，而后向的LSTM会依次输入，“中国”，“爱”，“我”得到三个向量u0,u1,u2，然后进行拼接得到[h0,u2],[h1,u1],[h1,u0].如图Figure.2(b),



(a) LSTM.5



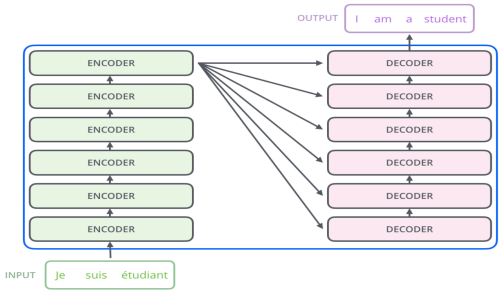
(b) LSTM.6

Figure 2. LSTM Model

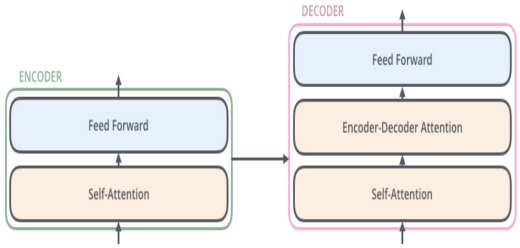
BERT. 在介绍BERT模型之前需要先介绍Transformer模型。

Transofrmer Transofrmer和LSTM等RNN型模型完全不同，他摆脱了RNN模型序列型神经元的特点，改用位置编码来表示序列信息。具体模型如图Figure.1,图片来自(Vaswani et al., 2017).Transformer就像大部分seq2seq模型一样，结构由encoder和decoder组成。Encoder由6个相同的layer组成，如图所示Figure.3(a)，可看出，有六个Encoder，六个decoder，输入为一个语句，每个encoder是下一个encoder的输入，而最后一个encoder是所有decoder的输入，底层Encoder同样为上层Encoder提供输入。最后的输出将与目标输出进行对比,得到损失并且反向传播。

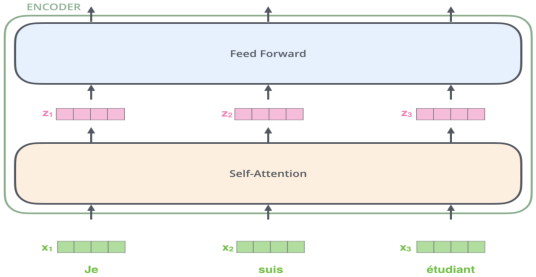
BERT Encoder-Decoder 如图3(b)，每一个Encoder由两层组成，包括自注意力(self-attention)和全连接层(feed forward)组成。而Decoder相比decoder多出一部分Encoder-Decoder attention，帮助提取Encoder输入的一部分信息。Decoder多出一个attention不难理解，因为每个Encoder只有一个输入，而Decoder都有两个输入。



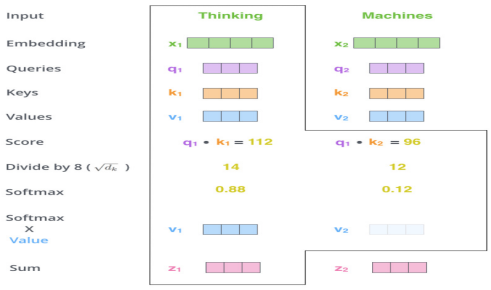
(a) 2.1



(b) 2.2



(c) 2.3



(d) 2.4

Figure 3. Transformer details

通常self-attention可以以一下方式来表示

$$Att = Attention(Q, K, V) \tag{5}$$

其中$Q = W^Q \times X$,$K = W^K \times X$, $V = W^V \times X$,这说明所有的attention均来自可学习矩阵的矩阵乘积，但是他们分别有什么直观的作用，仍然需要说明。直观的来说，Q代表查询，K表示句子输入的键值，Q代表得到的值，如图Figure.3(d)

我们输入一个词，然后计算不同的key对比value，得到他的概率值，作为他的位置信息。这样就实现了位置编码。

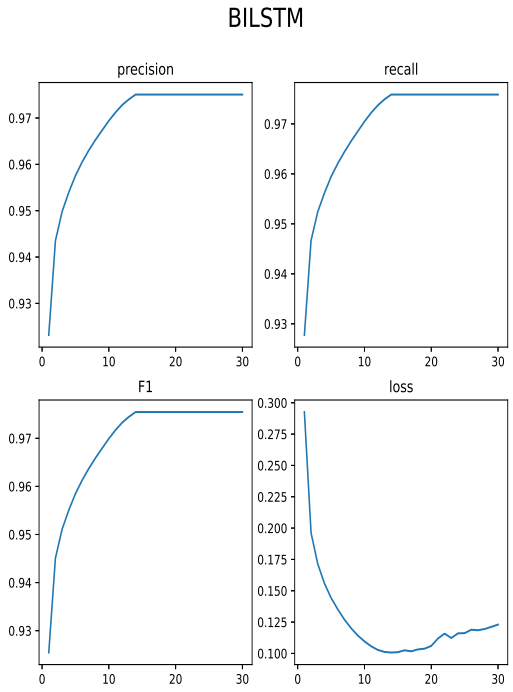
通过这个方式Transformer摆脱了RNN型神经网络的序列约束，同时保留了位置信息，从而大幅度提高了性能。

BERT Pre-traning 而BERT模型则是针对Transformer模型预训练，预训练(pre-training)的意思是，作者认为，确实存在通用的语言模型，先用文章预训练通用模型，然后再根据具体应用，用supervised 训练数据，精加工(fine tuning)模型，使之适用于具体应用。为了区别于针对语言生成的Language Model，作者给通用的语言模型，取了一个名字，叫语言表征模型 Language Representation Model。

能实现语言表征目标的模型，可能会有很多种，具体用哪一种呢？作者提议，用 Deep Bidirectional Transformers 模型。假如给一个句子“能实现语言表征[mask]的模型”，遮盖住其中“目标”一词。从前往后预测[mask]，也就是用“能/实现/语言/表征”，来预测[mask]；或者，从后往前预测[mask]，也就是用“模型/的”，来预测[mask]，称之为单向预测 unidirectional。单向预测，不能完整地理解整个语句的语义。于是研究者们尝试双向预测。把从前往后，与从后往前的两个预测，拼接在一起[mask1/mask2]，这就是双向预测 bi-directional.BERT的作者认为，bi-directional仍然不能完整地理解整个语句的语义，更好的办法是用上下文全向来预测[mask]，也就是用“能/实现/语言/表征/../的/模型”，来预测[mask]。BERT 作者把上下文全向的预测方法，称之为 deep bi-directional。

这个模型的核心是聚焦机制，对于一个语句，可以同时启用多个聚焦点，而不必局限于从前往后

的，或者从后往前的，序列串行处理。不仅要正确地选择模型的结构，而且还要正确地训练模型的参数，这样才能保障模型能够准确地理解语句的语义。BERT 用了两个步骤，试图去正确地训练模型的参数。第一个步骤是把一篇文章中，15 % 的词汇遮盖，让模型根据上下文全向地预测被遮盖的词。假如有 1 万篇文章，每篇文章平均有 100 个词汇，随机遮盖 15% 的词汇，模型的任务是正确地预测这 15 万个被遮盖的词汇。通过全向预测被遮盖住的词汇，来初步训练 Transformer 模型的参数。然后，用第二个步骤继续训练模型的参数。譬如从上述 1 万篇文章中，挑选 20 万对语句，总共 40 万条语句。挑选语句对的时候，其中 210 万对语句，是连续的两条上下文语句，另外 210 万对语句，不是连续的语句。然后让 Transformer 模型来识别这 20 万对语句，哪些是连续的，哪些不连续。这两步训练合在一起，称为预训练 pre-training。训练结束后的 Transformer 模型，包括它的参数，是作者期待的通用的语言表征模型。



(a) LSTM performance

4. Experiments

数据说明. 我们所有实验均使用SIGHAN Bakeoff2005中的PKU数据集，数据集从https://github.com/CLUEbenchmark/CLUEDatasetSearch#ner获得。SIGHAN Bakeoff2005数据集是大规模中文命名实体识别任务数据集，被广泛应用作为中文NER的Benchmark。

Metrics.

$$F_1 = \frac{2TP}{2TP + FP + FN}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

训练 训练过程中，BiLSTM有着较好的收敛速度和性能,如图4(a),BERT模型由于巨大时间复杂度和内存消耗没有进行第二次训练画图。

性能说明 我们使用F1,返回率(Recall),准确度(Precision),训练时间(Time)作为我们的性能评估,其中Time没有加入预训练时间。

5. Conclusion

尹梓琦. 通过这次大作业，加深了对于前端科学技术的理解，增强了阅读文献和写代码的能力。我们对比了多种方法，发现BERT性能上对比于LSTM其实并没有太大的提升，但是由于其预训练模型可以被反复应用，而不需要类似LSTM每次都训练，所以时间复杂度并没有上升太多。给深度学习提出了一新的思路，无监督的预训练模型。

崔冬航. 通过此次大作业，我了解了许多能够用于解决命名体识别问题的模型，对这些模型的实现原理与训练效果进行了多方位的比较，着重实现了基于CRF模型的命名体识别方法，并将其与BiLSTM模型合并优化，最终我们共同实现了一个能够投入使用的多模型命名体识别库。同时，在

Table 2. The Averaged Performance of Models

Metrics	Precision(%)	Recall(%)	F1(%)	Time(s)
CRF	0.9609	0.9155	0.9372	559
BiLSTM	0.9652	0.9665	0.9659	7485
BiLSTM+CRF	0.9662	0.9660	0.9666	
BERT	0.9684	0.9677	0.9670	12979

小组的合作开发过程中，我们也从诸多问题中寻求解决办法，如最后的代码合并的问题，经过讨论我们决定采用模块化的方式进行代码整合，这些解决问题的过程和积累的经验使我受益匪浅。

杨璐铭. 通过这次大作业，我们分析对比了CRF，BILSTM，CRF-BILSM,BERT四种模型在中文命名体识别上的表现。从结果上来看，四者同时都取得了较为不错的成绩，其中又以BERT最佳。但考虑到资源利用与算法效率方面CRF表现最为优异。因此，这几种算法都有自己适用的领域。通过这次小组合作我们学习到了更多NLP的知识，深入理解了命名体识别的原理。同时，机器学习相关的合作与其他工程的合作是不一样的，它更加需要小组成员的协调能力，而这次大作业提供了一个完美的机会，使我们受益良多。

提升与改进. 由于时间原因，我们只在一个数据上进行测试，缺乏普遍性，需要在更多的数据集上测试，例如MASR数据集。
其次，我们可以进行下游任务测试，证明这几个模型的优劣性在多项任务上保持一致，并且验证我们的猜想，即NER问题对于多个下游问题有帮助，如果下游问题性能和NER问题性能相对一致，则证实我们的猜想。

参考文献

Arora, R., Tsai, C., Tsereteli, K., Kambadur, P., and Yang, Y. A semi-markov structured support vector machine model for high-precision named entity recognition. In Korhonen, A., Traum, D. R., and Màrquez, L. (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 5862–5866. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1587. URL https://doi.org/10.18653/v1/p19-1587.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

Nabende, P., Tiedemann, J., and Nerbonne, J. Pair hidden markov model for named entity matching. In Sobh, T. M. (ed.), *Innovations and Advances in Computer Sciences and Engineering, Volume I of the proceedings of the 2008 International Conference on Systems, Computing Sciences and Software Engineering (SCSS), part of the International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering, CISSE 2008, Bridgeport, Connecticut, USA*, pp. 497–502. Springer, 2008. doi: 10.1007/978-90-481-3658-2_87. URL https://doi.org/10.1007/978-90-481-3658-2_87.

Sherstinsky, A. Fundamentals of recurrent neural net-

work (RNN) and long short-term memory (LSTM) network. *CoRR*, abs/1808.03314, 2018. URL http://arxiv.org/abs/1808.03314.

Song, S., Zhang, N., and Huang, H. Named entity recognition based on conditional random fields. *Clust. Comput.*, 22(Supplement):5195–5206, 2019. doi: 10.1007/s10586-017-1146-3. URL https://doi.org/10.1007/s10586-017-1146-3.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

Zong, C. *Statistic Natural Language Process*, pp. 1–495. Tsinghua University Press, BeiJing Tsinghua University. ISBN 978-7-302-31911-5.

A. 补充材料

借鉴博客. 最后需要提到的是，我们借鉴以下开源代码。

https://github.com/google-research/bert

https://github.com/luopeixiang/named_entity_recognition

https://github.com/lonePatient/BERT-NER-Pytorch

同时感谢以下博文，本文受到其启发或者有引用其内容

https://zhuanlan.zhihu.com/p/82312421

https://zhuanlan.zhihu.com/p/48508221

https://zhuanlan.zhihu.com/p/44121378

https://blog.csdn.net/weixin_38241876/article/details/110129336

特别感谢以下博文，本问中BERT和Transformer大部分截图节选自这篇文章

http://jalammar.github.io/illustrated-transformer/

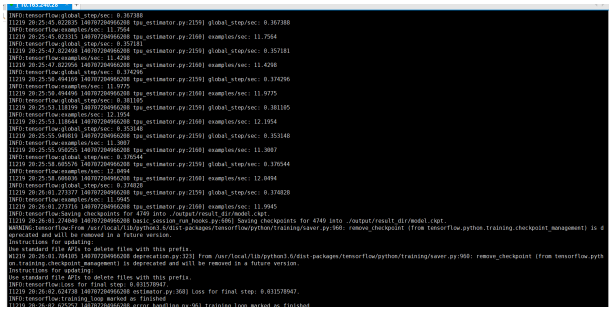
训练过程. 本段将给出一些训练过程中的材料，作为支撑材料，证明我们的训练真实性。

```
yinziqi@dell-Precision-7920-Tower:~/cuidonghang/hw2$ time python3 main.py
Reading data
CRF Model training...
CRF Model testing...
Tags            Precision       Recall          F1              Support
I_LOC           0.9438          0.9082          0.9257          332401
B_LOC           0.9598          0.9284          0.9438          210815
B_T             0.9795          0.9740          0.9767          181326
B_ORG           0.9681          0.9067          0.9364          15171
I_PER           0.9426          0.8425          0.8897          367815
O               0.9925          0.9974          0.9949          17603474
B_PER           0.9500          0.8409          0.8921          190682
I_ORG           0.9714          0.9013          0.9350          32994
I_T             0.9833          0.9786          0.9810          488323

Avg             0.9895          0.9897          0.9895          19423001
Avg without O   0.9609          0.9155          0.9372          1819527

real    9m40.067s
user    9m36.808s
sys     0m12.094s
```

(b) CRF Result

```
Epoch 33, Val Loss:0.1269
    B_PER     0.8987    0.6968    0.7850      2777
      I_T     0.9768    0.9107    0.9426      9942
    B_LOC     0.8559    0.8207    0.8379      3865
    I_ORG     0.9784    0.4905    0.6534       738
    B_ORG     0.9641    0.4879    0.6479       330
      B_T     0.9278    0.8527    0.8886      3374
        O     0.9812    0.9955    0.9883    294278
    I_LOC     0.8590    0.7652    0.8094      6164
    I_PER     0.9240    0.6940    0.7926      5398
      avg     0.9652    0.9665    0.9659    326866
Epoch 34, Val Loss:0.1287
```

(c) LSTM Result



(d) BERT Pre-Training

```
yinziqi@dell-Precision-7920-Tower:~$ python3 output.py
Tags            Precision               Recall          F1
I-LOC           0.9604          0.9685          0.9565
B-LOC           0.9669          0.9640          0.9669
B-T             0.9627          0.9632          0.9590
B-ORG           0.9581          0.9580          0.9545
I-PER           0.9581          0.9601          0.9661
B-PER           0.9648          0.9642          0.9640
I-ORG           0.9832          0.9782          0.9794
I-T             0.9905          0.9765          0.9801
The Averaged score is
                0.9684          0.9677          0.9670
```

(e) BERT Result