CoNLL2003 命名实体识别项目报告

徐迪 202028013229112

June 28, 2021

.

一、 前言

NER 全称是命名实体识别（Named Entity Recognition, NER），目标是识别文本中感兴趣的实体，如位置、组织和时间。NER 是属于自然语言处理中的序列标注任务 (sequence tagging)，序列标注中除了 NER，还有如词性（POS）标记和分块（Chunking）等。已识别的实体可以在各种下游应用程序中使用，比如根据患者记录去识别和信息提取系统，也可以作为机器学习系统的特性，用于其他自然语言处理任务。NER 总结的看其实就是提取出属于预定义类别的文本片段，它可能是通用性的，也可能是用户定义好的类型，属于特定的领域。

二、 项目介绍

项目中的数据集使用的是 CoNLL2003 英文数据集，数据集包含训练集（14041 个样本）、验证集（3250 个样本）和测试集（3453 个样本）。数据集中的目标实体分为人名（PER）、地名（LOC）、机构名（ORG），其他实体（MISC）。它使用标准的 BIOUL 实体标注方式，因此标注中包含 (B/L/U/I)-(PER/LOC/ORG/MISC) 这十六种标注以及 O 表示其他，共十七种标注类型。

项目的模型任务便是准确地识别出每一个实体以及它的类型，训练任务会在训练集和验证集上进行，通过验证集来优化模型参数，最后会在测试集上进行测试，计算 Accuracy、Precision、Recall 和 F1 等指标来评估模型。

通过调研 NER 的发展历程，我了解到 NER 的发展经历了从基于规则的线性模型，到后来的监督学习方法（HMM，DT，CRF 等），再到近年来的深度学习方法的大流行。为了对比研究各类 NER 方法在 CoNLL2003 数据集上的效果，以及研究不同模型之间的异同和优劣，这里分别实现了传统的监督学习方法 HMM 和 CRF 来进行命名实体的识别，同时使用卷积神经网络、循环神经网络，以及预训练模型 RoBERTa 方法分别进行实体的识别。

在下一个部分的研究进展中，本文对 NER 方面的研究发展过程进行了简单的综述，方便理解命名实体识别领域的研究趋势；在实验模型介绍部分，本文针对各个模型的原理进行了简要的介绍和总结对比，希望能从模型原理上解释实验的结果差异；在实验结果部分，对各个模型的结果进行了总结和分析；最后，本文对命名实体识别项目做了一个简要的总结。

项目地址：https://github.com/HanielF/ner_with_allennlp

三、 研究进展

命名实体识别 NER 的任务目标是给出一个命名实体的起始和终止边界，并给出命名实体的类别。一般而言完成 NER 任务的方法分为基于规则、基于无监督方法、基于特征的机器学习方法和基于深度学习的方法四种。其中一般领域性比较强，数据量很少的 NER 任务会用规则，其余基本上都是机器学习或者深度学习。尤其是在数据量比较充足的时候，深度学习一般都可以获得比较不错的效果。

在基于规则的 NER 任务中，需要手工指定符合条件的词及其对应的类别。具体使用的规则包括特定领域词典、同义词典、句法词汇模板和正则表达式等等。其优点在于不需要进行数据标注，但是指定规则工作量大，需要不断维护，同时迁移成本较高，常用的 NER 系统包括 LaSIE-II, NetOwl 等。当词汇表足够大时，基于规则的方法能够取得不错效果。但总结规则模板花费大量时间，且词汇表规模小，且实体识别结果普遍高精度、低召回。基于无监督的 NER 学习方法中使用聚类的方法，根据文本相似度进行不同实体类别组的聚类，同样不需要标注数据，但得到的结果准确度有限。常用到的特征或者辅助信息有词汇资源、语料统计信息（TF-IDF）、浅层语义信息（分块 NP-chunking）等。基于特征的有监督学习方法中，NER 任务可以视为机器学习 token 级别的多分类任务或序列标注任务，需要标注数据，同时一般结合精心设计的特征，包括词级别特征、文档特征和语料特征等等。常用的 NER 机器学习模型包括隐马尔可夫模型 HMM、决策树 DT、最大熵模型 MEM、最大熵马尔科夫模型 HEMM、支持向量机 SVM、条件随机场 CRF 等。

深度学习 NER 受益于 DL 非线性，相比于传统线性模型可以学到更为复杂并对模型有益的特征，端到端过程得以实现，近年来成为主流研究方向。近年来，使用大规模的语料数据进行模型的预训练，然后在下游任务进行任务和数据导向的微调，成为了自然语言处理领域各个任务的主流方法。得益于大规模的语料，以及自注意力机制的存在，预训练模型能够很好地学习语言本身具有的含义，并且通过无监督的训练方式得到一个具备潜在语义信息的编码，这种通用的包含语义的编码能够适应大部分下游任务，也具有更强的表示能力。最具代表性的是 BERT，它的提出是从监督学习方法到无监督预训练模型的转折。随后各类方法大多都是在 Transformer 基础上，调整预训练任务和策略等。比较有名的有 GPT 系列，XLNET，ALBERT，RoBERTa 等，它们的核心模块都是自注意力机制，通用特点是使用大规模语料训练大规模的语言模型，提取通用的语言特征。目前大多数方法都是基于深度学习模型训练得到文本的隐语义编码，然后通过结合 CRF 模型得到预测结果。这样充分结合了深度学习强大的编码能力和传统机器学习的分类能力，也避免了深度学习模型在预测命名实体类别时，出现不可控的情况，例如 B-PER 后面不可能出现 B-PER，这是一种规则约束，而单纯的深度学习模型是有可能出现这种情况的，结合 CRF，通过学习得到一系列规则约束，避免了这些问题。

除了模型方面，近年来的研究偏向于结合不同级别的语义编码，从字符级别，到词级别，到实体级别的编码，不同模型之间的区别可能就在于如何提取这些编码，以及如何后处理这些编码。LUKE 模型便是在其基础上对实体级别的编码进行预训练，并且使用 BERT 语言模型的掩码机制和新的训练任务，结合实体级别的自注意力机制，最终在多个实体相关的任务上获得了 SOTA 效果。

目前在命名实体识别领域，实体识别效果最好的是使用 ACE 模型 (Automated Concatenation of Embeddings) 结合文档级上下文语义，模型的目标是找到更好的 embedding 拼接方式，并且使用神经网络结构搜索的方式来自动化寻找拼接方式这一过程，而不是人为定义好

拼接哪些 embeddding。同时论文中使用了强化学习方法对模型进行训练，对于好的拼接方式，模型将会得到一个奖励，反之会得到一个惩罚，这样一个强化学习的奖励机制能够让模型学习到具体应该拼接哪些 embedding。

四、 整体框架介绍

1. Data

数据读取预处理部分，主要包括 tokens 表示 word level 的处理，具体为对词进行小写处理，token_characters 表示 character-level 的处理，type 是 dataset_reader 的读取类型，设置为 conll2003。

2. Embedding

模型的第一层是词嵌入层，利用随机初始化的 embedding 矩阵将句子中的每个字由 one-hot 向量映射为低维稠密的字向量，其中每个维度都表示隐含的特征维度。单词的字符级表示与预训练得到的词向量连在一起作为最终的词表示。数据预处理采用 word level 进行，label 的编码格式设置为 BIOUL。

3. Tag decoder

模型的第二层是 CRF 层，进行句子级的序列标注。CRF 层的参数是标签之间转移得分矩阵，进而在为一个位置进行标注的时候可以利用此前已经标注过的标签。条件随机场 CRF 利用全局信息进行标记用于解码。在预测当前标签时使用邻居的标签信息 NER 中，CRF 模型关注整个句子的结构，是一个输出和输出直接相连的无向图，产生更高精度的标签。

4. Trainer

训练器相关的参数的设置。根据不同的模型，训练算法使用 SGD（随机梯度下降法）或者 Adam（自适应矩估计）等。其中 LSTM+CRF 模型用前向和后向 LSTM 各一个独立层，维度为 100，并加入了剔除率为 0.5 的 dropout。

五、 实验模型介绍

1. HMM

隐马尔科夫模型（Hidden Markov Model，以下简称 HMM）在语言识别，自然语言处理，模式识别等领域都有广泛的应用。HMM 作为一个经典的机器学习模型，本项目也实现了用 HMM 进行命名实体识别的任务。

HMM 模型主要能够用于两个场景：

1. 项目的问题是基于序列的，比如时间序列，或者状态序列

2. 项目的问题中有两类数据，一类序列数据是可以观测到的，即观测序列；而另一类数据是不能观察到的，即隐藏状态序列

对于 HMM 模型，首先假设 Q 是所有可能的隐藏状态的集合，V 是所有可能的观测状态的集合，即：$Q = q_1, q_2, ..., q_N, V = v_1, v_2, ..., v_M$, 其中，N 是可能的隐藏状态数，M 是所有的可能的观察状态数。在本项目中，CoNLL2003 使用 BIOUL 标注方式，因此共有 17 种隐状态，观测状态数量就是词汇数量。对于一个长度为 T 的序列，$I = i_1, i_2, ..., i_T$ 对应的状态序列, $O = o_1, o_2, ..., o_T$ 是对应的观察序列。HMM 通过状态转移矩阵 A、发射矩阵 B 已经初始化矩阵 π 来进行隐状态的转移和观测状态的预测。

这主要是基于两个条件假设：

1. 齐次马尔科夫链假设。它决定了如何计算状态转移矩阵 A，即任意时刻的隐藏状态只依赖于它前一个隐藏状态，这样假设的好处是模型简单，便于求解，缺点则是某一个隐藏状态不仅仅只依赖于前一个隐藏状态。若时刻 t 的状态为 q_i，时刻 $t+1$ 的状态为 q_j，则从 t 到 t+1 的状态转移概率 $a_{ij} = P(i_{t+1} = q_j | i_t = q_i)$

2. 观测独立性假设。它决定了如何计算发射矩阵 B，即任意时刻的观察状态只仅仅依赖于当前时刻的隐藏状态，这也是一个为了简化模型的假设。若 t 时刻的隐状态是 $i_t = q_j$，观测状态为 $o_t = v_k$，则在 q_j 的条件下生成 v_k 的概率是 $b_{jk} = P(o_t = v_k | i_t = q_j)$

对于初始化的隐藏状态概率分布矩阵 π，$\Pi = [\pi(i)]_N$，其中，$\pi = P(i_1 = q_i)$

因此，对于一个 HMM 模型，只需要先统计得到初始化矩阵 Π，状态转移矩阵 A，观测矩阵，也叫发射矩阵 B，就可以通过初始化第一个时间步，然后使用维特比算法，不断转移到一个概率最大的隐状态对应当前的观测值，然后便可以通过观测状态序列得到隐状态序列。

2. CRF

条件随机场 CRF 是一种基于统计的序列标记和分割数据的方法，是用于序列标注问题的无向图模型，在给定需要标记的观测序列条件下，计算序列的联合概率。条件随机场的建立过程中，首先定义一个特征函数集，每个特征函数都以标注序列作为输入，提取特征作为输出.

条件随机场使用对数线性模型来计算给定观测序列下状态序列的条件概率 $p(s|x; w)$。w 是条件随机场模型的参数，可以视为每个特征函数的权重。CRF 模型的训练其实就是对参数 w 的估计。模型训练结束之后，对给定的观测序列 x ，可得到其最优状态序列，解码后得到最终结果。

在预测当前标签时使用邻居的标签信息 NER 中，CRF 模型关注整个句子的结构，是一个输出和输出直接相连的无向图，产生更高精度的标签。同时，使用条件随机场 CRF 可解决 tagging 之间不独立的问题。对每种生成的 tag 序列，这里采用打分的方式代表该序列的好坏，分数越高代表当前生成的 tag 序列表现效果越好。

3. BiLSTM+CRF

RNN 对处理序列问题有天然的优势存在，传统的 RNN 由于存在梯度消失问题，对长距离的语义关系提取的效果并不好，直到 LSTM 的提出，使得大量自然语言处理问题中开始应用 LSTM 这一循环神经网络模型。但 LSTM 对句子进行建模存在一个问题：无法编码从后到前

的信息。例如“这个餐厅脏得不行，没有隔壁好”，这里的“不行”是对“脏”的程度的一种修饰，BiLSTM（Bi-directional Long Short-TermMemory）则是由前向 LSTM 与后向 LSTM 组合而成。通过 BiLSTM 可以更好的捕捉双向的语义依赖。而 BiLSTM 通过综合前向与后向 LSTM 分别捕捉到了每个词上文与下文的信息，一定程度上解决了这个问题。

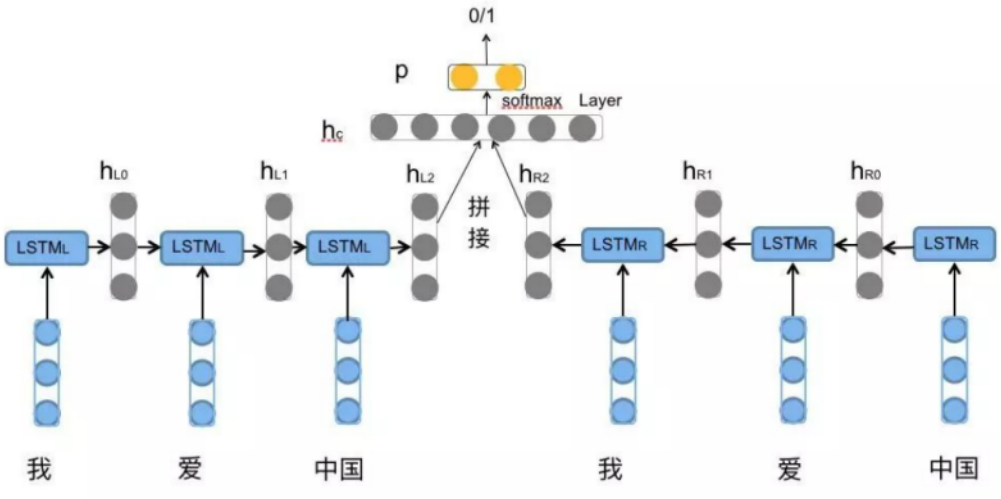

图 1: BiLSTM 结构图

这里主要是用 BiLSTM 来提取字符级别的特征编码，以及对编码进行序列级别特征提取，得到一个特征融合后的结果，再通过全连接层得到对所有 tag 的 logit，最后使用 CRF 进行实体的识别预测。

4. CNN+BiLSTM+CRF

对于分词任务，当前词的标签基本上只与前几个词有关联。BiLSTM 在学习较长句子时，可能因为模型容量问题丢弃一些重要信息，因此在模型中加了一个 CNN 层，用于提取当前词的局部特征，加入了 CNN 的模型如下。

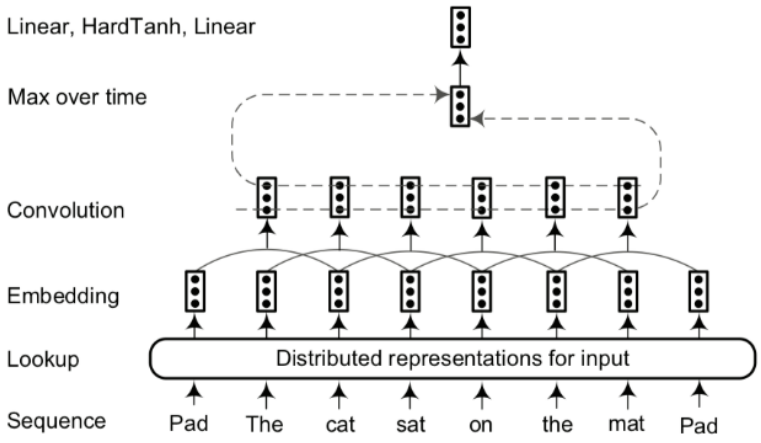

图 2: 基于句法的 CNN 网络结构

句子经过 embedding 层，一个 word 被表示为 N 维度的向量，随后整个句子表示使用卷积编码，进而得到每个 word 的局部特征，卷积层的输出大小与输入的句子长度有关，为了获取固定维度的句子表示，常常使用最大池化操作得到整个句子的全局特征。

5. Transformer

项目中使用的 BERT 和 RoBERTa 都是基于 Transformer 的 encoder，并且在 Transformer 的基础上提出了特定的预训练任务，因此在这里对 Transformer 进行一个系统的介绍，后续的 BERT 模型和 RoBERTa 模型则会相对较少的描述，而是更偏重在项目本身的模型结构上。

Transformer 是 Vaswani 等人在《Attention Is All You Need》中提出的，目标是为了提高机器翻译的效率，他们使用了创新的 self-attention 机制，用于替代 RNN 循环神经网络，能够高效的并行计算所有位置上的隐藏状态，同时克服了 RNN 中一直存在的梯度消失问题。Transformer 主体分为 encoder 和 decoder，是比较经典的 seq2seq 模型结构，如图 3.3 所示，后续的 BERT 和 GPT 系列都是只采用了编码器和解码器中的一种，并且都在各类自然语言处理任务中刷新了 SOTA。Transformer 是实验中的模型 BERT 和 RoBERTa 的基础，下面主要对 Transformer 中的多头自注意力机制、带有位置编码的输入、Layer Normalization、残差连接这几个部分进行简要的介绍。

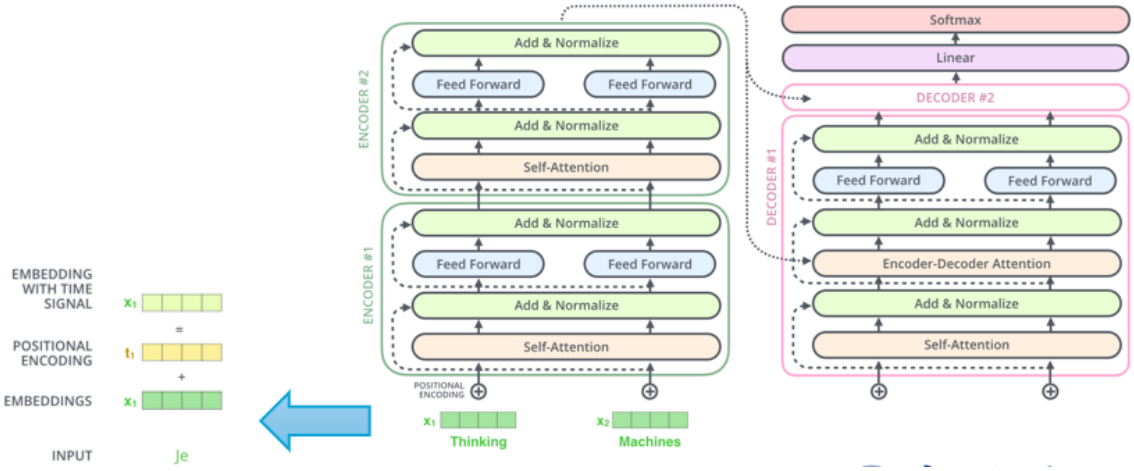

图 3: encoder-decoder 结构图

把 encoder 和 decoder 展开后如图 6所示，是由两个 encoder 和一个 decoder 的模型结构。可以看到在 encoder 中主要有 positional encoding、self-attention、残差连接、layer Normalization。decoder 与 encoder 的不同主要在于在输入中加入 encoder 的输出

Transformer 中最有名的还是 self-attention，它在文本处理中替代了 RNN，不用受限于 RNN 每个时间步对前一个时间步的依赖，以及解决了长距离的梯度消失问题。在自注意力机制被提出之前，已经有了传统的注意力机制，研究者发现文本不同位置的 token 或者图片中的 block 都应该有不同的权重，对结果的影响同样有不同的权重。传统的 attention 机制通常是对隐藏层进行矩阵运算得到，比如对 LSTM 的 hidden state 进行 dot product 或者一些 attention 变种如 hard attention，soft attention，global attention，local attention 等。Self-Attention 的实现主要是通过三个矩阵实现，分别是 Query 矩阵、Keys 矩阵、Values 矩阵，Query 和 Key 矩阵用于计算权重，Values 矩阵代表这个位置 token 真正的内容，它们分别是通过对输入进行线性变换得到。对于传统 attention 中的权重 energy 计算，self-attention 使用 Query 矩阵和 Key 矩阵进行点乘运算，并经过一个 softmax 归一化得到。所以从这个角度来说，传统的 attention 机制也是 self-attention 的一种特殊情况。得到权重之后便可以用它对 Values 矩阵进行加权平均，然后得到 self-attention 层的输出 Z 矩阵。Multi-Head Attention 其实就是对上面的操作进行多次。会有一组 QKV 向量。Multi-head 是有概率学习

到文本的不同语义的。

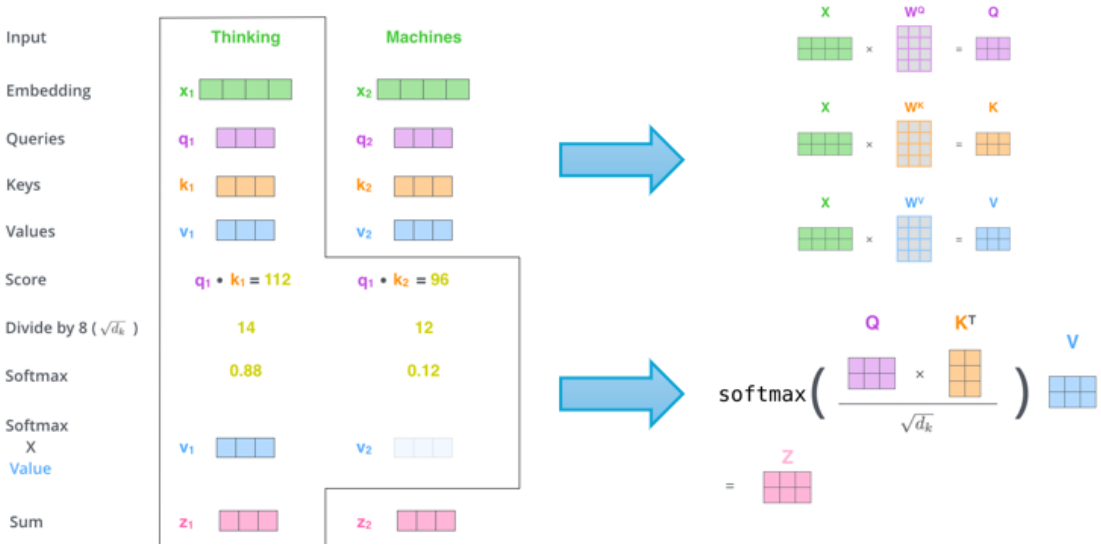

图 4: self-attention 运算过程

Transformer 中在进行 self-attention 之后，会进行残差连接的操作，就是加上之前的输入。而 Layer Normalization 其实就是对每个样本的不同特征进行 Normalization，之前的 BatchNorm，是对所有 batch 样本的每个特征进行 Normalization，Batch Normalization 还不可以用于 RNN 模型，因为同一个节点在不同时刻的分布是明显不同的。Transformer 的残差连接其实是将输入通过 self-attention 之后，得到输出 z1 和 z2，然后和残差连接的输入 x1,x2，加起来，然后经过 layer normalization 层之后输出给全连接层。全连接层也是有一个残差连接和一个 LayerNorm 层，最后再输出给上一层。

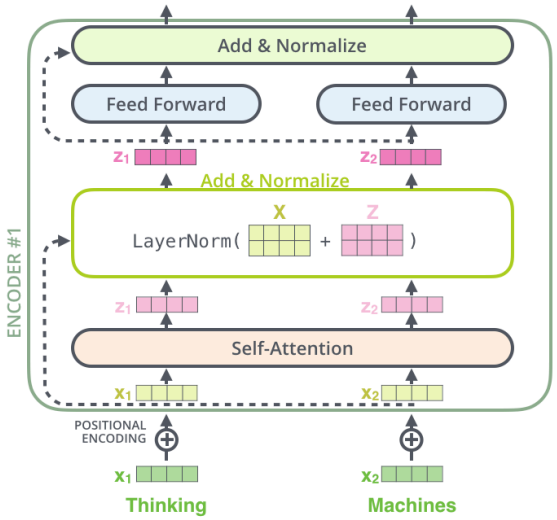

图 5: 残差连接和层级归一化过程

6. BERT+BiLSTM+CRF

BERT+BiLSTM+CRF 模型中，BERT 负责学习输入句子中每个字和符号到对应的实体标签的规律，学得句子中每个字符最大概率的实体标注，过程中考虑每个字符的上下文信息，而

CRF 负责学习相邻实体标签之间的转移规则，通过引入 CRF 解决输出的最大分数对应的实体标注依然可能有误的问题。

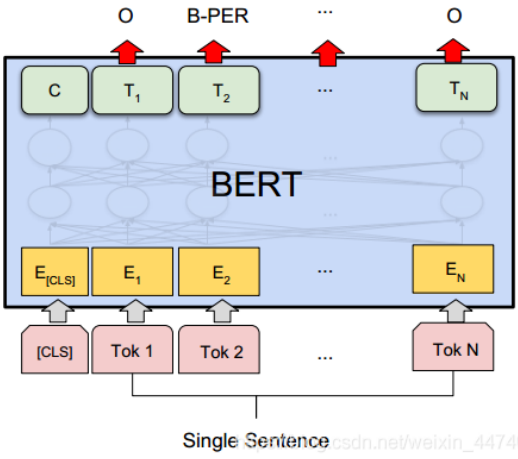

图 6: 基于 BERT 的序列标注结构

通过 BERT 学习序列的状态特征得到一个状态分数，该分数直接输入到 CRF 层，省去了人工设置状态特征模板的过程。其中，状态分数是根据训练得到的 BERT 模型的输出计算出来的，转移分数通过 CRF 层提供的转移分数矩阵计算。

7. RoBERTA+BiLSTM+CRF

RoBERTa 是在 BERT 的一个继承和优化，从模型上来说，RoBERTa 基本没有什么太大创新，主要是在 BERT 基础上做了几点调整：

1. 预训练的时间更长，batch size 更大，training step 减少，实验效果相当或者更好些；
2. 移除了 next predict loss 预训练任务，相比于 BERT，采用了连续的 full-sentences 和 doc-sentences 作为输入（长度最多为 512）；
3. 预训练训练的序列更长；
4. 动态调整 Masking 机制，即每次输入到序列的 Masking 都不一样
5. text encoding，基于 bytes 的编码可以有效防止 unknown 问题
6. 训练数据更多，数据集从 16G 增加到了 160G，训练轮数比 BERT 有所增加；在结果上，RoBERTa 超越了 BERT 和 XLNet，这里也选择对它作为 BERT 的对比实验模型，探究不同预训练任务以及 masking 策略在对下游情感分析任务上的影响。

命名实体识别预测的过程中，输入方面，使用了 RoBERTa 为数据集进行编码，由于 RoBERTa 使用 word pieces 级别的编码，而实际使用中使用的是词级编码，因此是对每个词进行了 word pieces 级别的切分，然后对切分的结果使用 RoBERTa 编码，然后再对每个词的 Word pieces 进行一个拼接，得到词级别的编码。模型的编码器使用的还是 BiLSTM，提取上下文的特征，最后使用全连接层对每个类目进行一个回归，得到对应类目的 logit。最后再使用 CRF 模型限制输出的形式，得到最终的输出结果。

相对于其他模型，甚至 BERT 模型本身来说，RoBERTa 结合 CRF 强大的地方并不是 encoder 或者 decoder 有特殊的处理，而是在输入的 embedding 阶段，有更强的语义表示，或者说更能代表实体本身的含义。从词级别的拆分来看，也可以用这种方法扩展到实体级别的编码，通过多层级的的编码融合，得到更强的更具有表征意义的输入，对模型的训练是非常有效的。

六、 实验结果分析

1. 数据集

命名实体识别 NER 任务中，本项目使用来自 CoNLL 2003 共享的的英文数据进行实验，该数据集包含四种不同类型的命名实体：PERSON、LOCATION、ORGANIZATION 和 MISC。这里使用 BIOUL 编码格式，因为之前的研究中相对于默认的 BIO 编码格式存在显著改进。

2. 实验结果

本文对所选的传统机器学习模型 HMM 和 CRF，以及以深度学习模型 CNN、BiLSTM、BERT、RoBERTa 为主的共六个模型的性能进行比较，除传统机器学习模型之外，基于所有这些模型都使用斯坦福大学的 GloVe 词嵌入和相同的超参数运行。

System	tagging accuracy	F1 scores
HMM	90.18	59.32
CRF	92.00	69.36
BiLSTM+CRF	96.89	85.15
CNN+BiLSTM+CRF	96.65	84.69
BERT+BiLSTM+CRF	97.95	90.44
RoBERTa+BiLSTM+CRF	98.02	91.22

表 1: 对比不同 NER 模型下的标注准确率和 F1 得分

本文中的模型可以通过 GloVe 嵌入获得 91.22 的最佳 F1 分数。使用了各种机器学习分类器的组合，根据表 1 结果，BLSTM-CRF 模型明显优于 CRF 模型，表明句子特征提取对于命名实体识别任务很重要。BiLSTM+CRF 略微优于 CNN+BiLSTM+CRF，可能是使用了 GloVe 不同的词嵌入。然而，BERT+BiLSTM+CRF 明显优于 BiLSTM+CRF，预训练模型 BERT 的引入大大提高了准确度，而针对 BERT 模型的改进模型 RoBERTa 性能也有所提升。

```
.
├── allennlp-source-code # allennlp源码
├── configs # 模型配置文件
├── data # 数据集
├── models # 生成的模型目录，不包含大模型，只保存了训练日志和结果
│   ├── model_bert # BERT+BiLSTM+CRF
│   ├── model_bilstm_crf # BiLSTM+CRF
│   ├── model_cnn # CNN+BiLSTM+CRF
│   ├── model_crf # CRF only
│   ├── model_hmm # HMM only
│   └── model_roberta_crf # RoBERTa+BiLSTM+CRF
├── ner_modules # 包含HMM和CRF tagger
│   ├── crf_tagger.py # CRF tagger
│   ├── hmm.py # 第一版HMM模型
│   └── hmm2.py # 第二版HMM模型
└── report # 项目报告
```

图 7: 源码目录结构和介绍

七、 实验源码和运行说明

1. 目录结构和介绍

2. 运行方式

1. clone 到本地
   - git clone https://github.com/HanielF/ner_with_allennlp.git
2. 主要依赖环境介绍
   - $python = 3.7$
   - $pytorch == 1.8.1$
   - $allennlp == 2.4.0$
   - $allennlp - models == 2.4.0$
3. 切换到项目路径：cd ner_with_allennlp
4. HMM 运行方式：python ner_modules/hmm2.py
5. CRF/CNN/BiLSTM/BERT/RoBERTa 运行方式
   - python run.sh crf
   - python run.sh cnn
   - python run.sh bilstm_crf
   - python run.sh bert
   - python run.sh roberta_crf

八、 总结

在对 CoNLL2003 数据集进行命名实体识别的过程中，我通过阅读相关论文，对命名实体识别的方法有了全局上的了解，并且了解了整个时间线上 SOTA 的发展变化过程，这让我对命名实体识别这一任务的背景和目标都有了充分的认识。

在传统的实体识别方法上，这里选用了经典的隐马尔科夫模型和条件随机场，它们基于对训练集数据的统计分析完成测试集上面的实体预测。而在深度学习的模型选择上，这里侧重使用基于 Transformer 的两个预训练模型，分别是 BERT 和 RoBERTa，它们都是通过对大规模语料的预训练，提取出语言本身具有的潜在含义，然后再迁移到命名实体识别这一下游任务中。对比基于 CNN 和 BiLSTM 的模型，从准确率和 F1 分数上也可以看到有很大的提升，这也充分证明了预训练模型的有效性，以及模型的初始化编码对模型能力上的提升作用。