深度之眼
deepshare.net

state-of-the-art模型介绍专题

导师：　　向右

深度之眼
deepshare.net

目录

深度之眼
deepshare.net

1、NER定义

命名实体识别

NER

定义

命名实体识别（Named Entity Recognition，简称NER），又称作“专名识别”，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。

命名实体识别是信息提取、问答系统、句法分析、机器翻译、面向Semantic Web的元数据标注等应用领域的重要基础工具，在自然语言处理技术走向实用化的过程中占有重要地位。

通常包括两部分：（1）实体边界识别；（2）确定实体类别（人名、地名、机构名或其他）。英语中的命名实体具有比较明显的形式标志（即实体中的每个词的第一个字母要大写），所以实体边界识别相对容易，任务的重点是确定实体的类别。和英语相比，汉语命名实体识别任务更加复杂，而且相对于实体类别标注子任务，实体边界的识别更加困难。

https://baike.baidu.com/item/命名实体识别/6968430?fr=aladdin

深度之眼
deepshare.net

2、标注方法

命名实体识别

NER

标注方法

IOB 标注法

IOB 标注法， 是 CoNLL 2003 采用的标注法， I 表示 Inside, O 表示 Outside, B 表示 Begin。而标注的 label是 I-XXX 的， I 表示这个字符在 XXX类命名实体的内部(inside)。B 用于标记一个命名实体的开始。

比如：

补 B-DRUG_EFFICACY	补 B-DRUG_EFFICACY
气 I-DRUG_EFFICACY	气 I
养 I-DRUG_EFFICACY	养 I
血 I-DRUG_EFFICACY	血 I
、O	、O
调 O	调 O
经 O	经 O
止 O	止 O
带 O	带 O

NER

标注方法

IOBS 标注法

在IOB基础之上，增加S单个实体情况的标注。S，即Single，表示单个字符。

比如：

补 B-DRUG_EFFICACY	补 B-DRUG_EFFICACY
气 I-DRUG_EFFICACY	气 I
养 I-DRUG_EFFICACY	养 I
血 I-DRUG_EFFICACY	血 I
、O	、O
但 O	但 O
苦 S-DRUG_TASTE	苦 S-DRUG_TASTE

NER

标注方法

BIOES

这是在 IOBS方法上，扩展出的一个更复杂，但更完备的标注方法。其中 B表示这个词处于一个实体的开始(Begin), I 表示内部(inside), O 表示外部(outside), E 表示这个词处于一个实体的结束为止， S 表示，这个词是自己就可以组成一个实体(Single)。

比如：

补 B-DRUG_EFFICACY	补 B-DRUG_EFFICACY
气 I-DRUG_EFFICACY	气 I
养 I-DRUG_EFFICACY	养 I
血 E-DRUG_EFFICACY	血 E
、O	、O
但 O	但 O
苦 S-DRUG_TASTE	苦 S-DRUG_TASTE

深度之眼
deepshare.net

3、state of the art模型

模型设计

NER

经典论文

深度之眼
deepshare.net

《Natural language processing (almost) from scratch》

是较早使用神经网络进行NER的代表工作之一。在这篇论文中，作者提出了窗口方法与句子方法两种网络结构来进行NER。这两种结构的主要区别就在于窗口方法仅使用当前预测词的上下文窗口进行输入，然后使用传统的NN结构；而句子方法是以整个句子作为当前预测词的输入，加入了句子中相对位置特征来区分句子中的每个词，然后使用了一层卷积神经网络CNN结构。

https://dl.acm.org/doi/pdf/10.5555/1953048.2078186

NER

经典论文

《Natural language processing (almost) from scratch》

在训练阶段，作者也给出了两种目标函数：一种是词级别的对数似然，即使用softmax来预测标签概率，当成是一个传统分类问题；另一种是句子级别的对数似然，其实就是考虑到CRF模型在序列标注问题中的优势，将标签转移得分加入到了目标函数中。后来许多相关工作把这个思想称为结合了一层CRF层，所以我这里称为NN/CNN-CRF模型。

https://dl.acm.org/doi/pdf/10.5555/1953048.2078186

NER

经典论文

《Bidirectional LSTM-CRF Models for Sequence Tagging》

经典模型，率先提出使用 Bi-LSTM + CRF 进行序列标注。

在Conv-CRF的基础上，作者将CNN结构换成了RNN结构，提出的BiLSTM-CRF在各项NLP序列标注中取得了当时的SOTA（state-of-the-art）表现，一方面BiLSTM使得模型可以同时获取前后项的特征信息，另一方面CRF使得模型能够获取句子级别的标注信息。因为CRF层的引入可以有效的解决预测标签之间的强语法依赖的问题，因此有效的避免了预测标签冲突的情况，尤其是对于NER这种标签带有强约束的任务来说。

https://arxiv.org/pdf/1508.01991.pdf

NER

经典论文

《Bidirectional LSTM-CRF Models for Sequence Tagging》

另外，作者发现BiLSTM-CRF比起其他模型更加稳健，即使不借助于Word Embedding，标注的准确度也没有大幅下降，这说明模型能够自动学习到一部分语义信息。模型结构如下图所示：

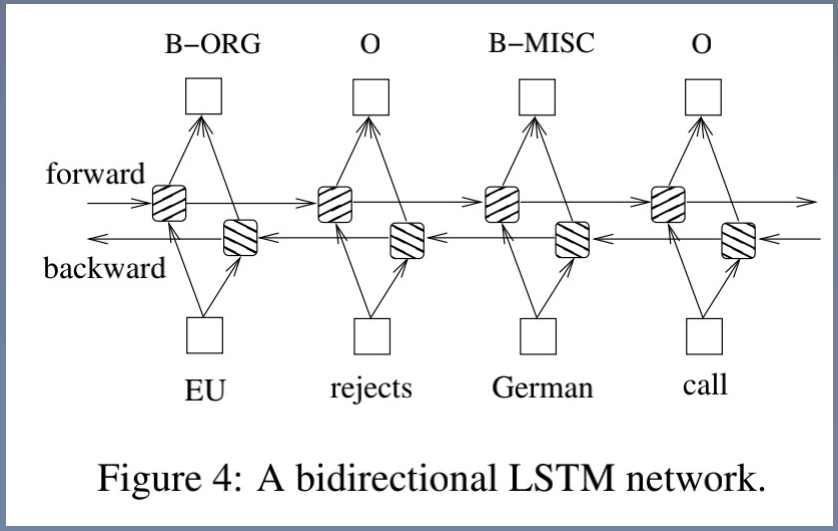


Figure 4: A bidirectional LSTM network.

https://arxiv.org/pdf/1508.01991.pdf

NER

经典论文

《Bidirectional LSTM-CRF Models for Sequence Tagging》

模型用的人工特征。分为拼写特征和语义特征。

拼写特征：开头是否大写、所有都大写、都小写、非大写开头、数字字母混合、有标点符号、前缀和后缀、撇号结尾、只有字母、不止字母、词的模式特征等。

语义特征：一元和二元特征，POS用了三元特征。

https://arxiv.org/pdf/1508.01991.pdf

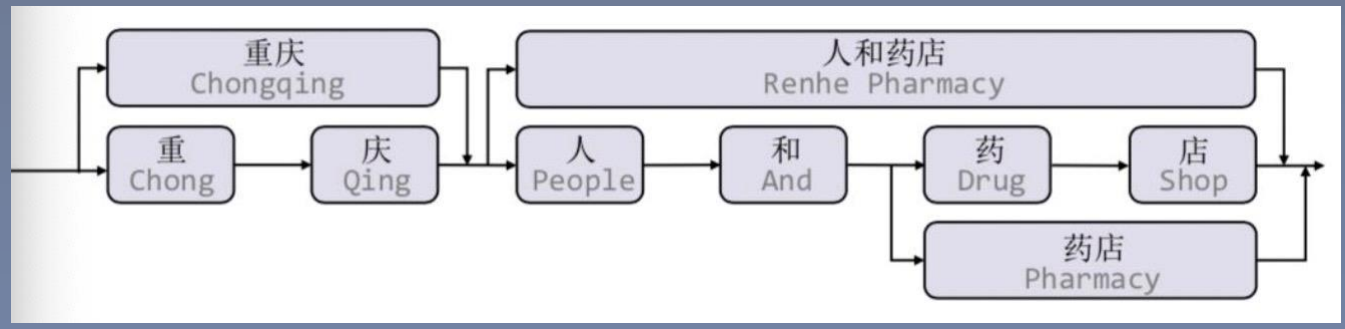
NER

Lattice

由于中文词汇的稀疏性和模糊性，导致基于字符的NER系统通常好于基于词汇（经过分词）的方法，但在基于字符的模型中引入分词信息往往能够带来性能的提升，尤其是对于NER任务来说，词汇能够提供丰富的实体边界信息。

Lattice LSTM：《Chinese NER Using Lattice LSTM（ACL2018）》

首次提出使用Lattice结构在NER任务中融入词汇信息，如下图所示，一个句子的Lattice结构是一个有向无环图，每个节点是一个字或者一个词。



https://zhuanlan.zhihu.com/p/142615620

NER

Lattice

Lattice LSTM：《Chinese NER Using Lattice LSTM（ACL2018）》

缺点：

- 不支持batch;
- 如果识别任务是识别新词，效果并不特别有效【bert这块效果还是比较显著】。
- 过多的单词会让基于字符的NER模型退化成基于单词的NER模型，这样子就会遭受分词错误的影响。

NER

Dynamic Architecture

LR-CNN:CNN-Based Chinese NER with Lexicon Rethinking (IJCAI2019)

作者提出了含有rethink机制的CNN网络解决Lattice LSTM的词汇冲突问题。

CGN: Leverage Lexical Knowledge for Chinese Named Entity Recognition via Collaborative Graph Network

LGN: A Lexicon-Based Graph Neural Network for Chinese NER (EMNLP2019)

GNN将NER任务转化为节点分类任务，可以捕捉到Lattice的有向无环结构，但是这些模型都需要LSTM作为底层编码器来编码序列的归纳偏置，这导致模型结构的复杂度较高。

https://zhuanlan.zhihu.com/p/249181095

NER

FLAT

FLAT: Chinese NER Using Flat-Lattice Transformer（ACL2020）

代码地址：https://github.com/LeeSureman/Flat-Lattice-Transformer

- ✓ Lattice-LSTM和LR-CNN采取的RNN和CNN结构无法捕捉长距离依赖，而动态的Lattice结构也不能充分进行GPU并行。

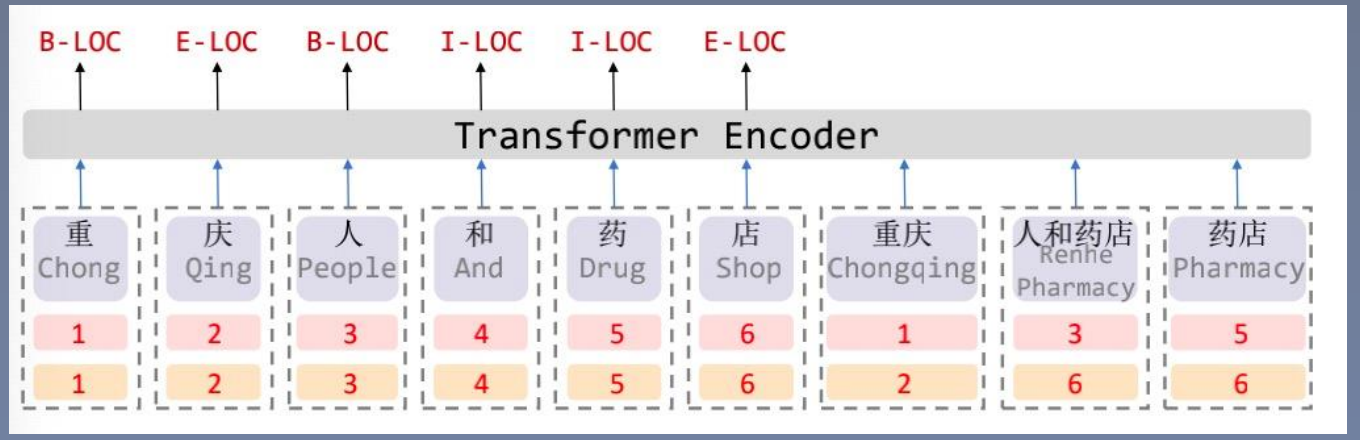
- ✓ CGN和LGN采取的图网络虽然可以捕捉对于NER任务至关重要的顺序结构，但这两者之间的gap是不可忽略的。其次，这类图网络通常需要RNN作为底层编码器来捕捉顺序性，通常需要复杂的模型结构。

https://zhuanlan.zhihu.com/p/142615620

NER

FLAT

从Transformer的position representation得到启发，作者给每一个span(字、词)增加了两个位置编码，分别表示该span在sentence中开始(head)和结束(tail)的位置，对于字来说，head position和tail position是相同的。



https://zhuanlan.zhihu.com/p/249181095

NER

FLAT

通过这种方式，可以从这样的标签序列中无损地重建Lattice结构。同时，这样扁平的结构允许我们使用Transformer Encoder，其中的self-attention机制允许任何字符和词汇进行直接的交互。

有了位置编码，容易想到可以像原始Transformer那样将字向量直接和两个位置向量相加，然后参与后续的self-attention。

$$\mathrm{Att}(\mathbf{A}, \mathbf{V}) = \mathrm{softmax}(\mathbf{A})\mathbf{V}, \tag{1}$$

$$\mathbf{A_{ij}} = \left(\frac{\mathbf{Q_i}\mathbf{K_j}^{\mathrm{T}}}{\sqrt{\mathrm{d_{head}}}}\right), \tag{2}$$

$$[\mathbf{Q}, \mathbf{K}, \mathbf{V}] = E_x[\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v], \tag{3}$$

NER

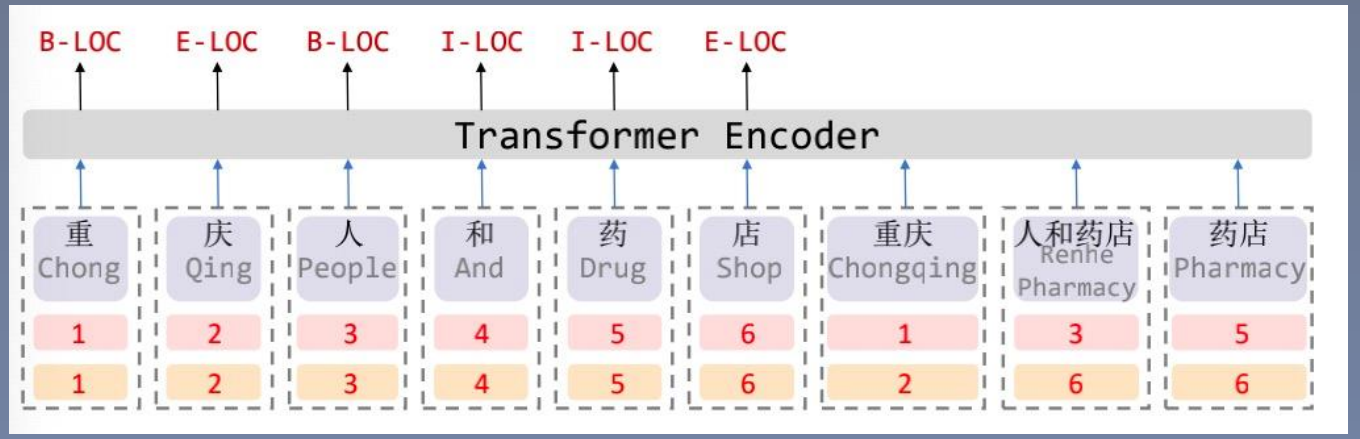
FLAT

论文中提及，上面这种方式并不算是有效编码了位置信息，也是原始Transformer在NER任务上的性能比不过BiLSTM的原因之一。针对本文提出的Flat结构，作者借鉴并优化了Transformer-XL (ACL 2019)中的相对位置编码方法，有效地刻画了span之间的相对位置信息。

NER

FLAT

span是字符和词汇的总称，span之间存在三种关系：交叉、包含、分离，然而作者没有直接编码这些位置关系，而是将其表示为一个稠密向量。作者用 head[i] 和 tail[i] 表示span的头尾位置坐标，并从四个不同的角度来计算 x_i 和 x_j 的距离：

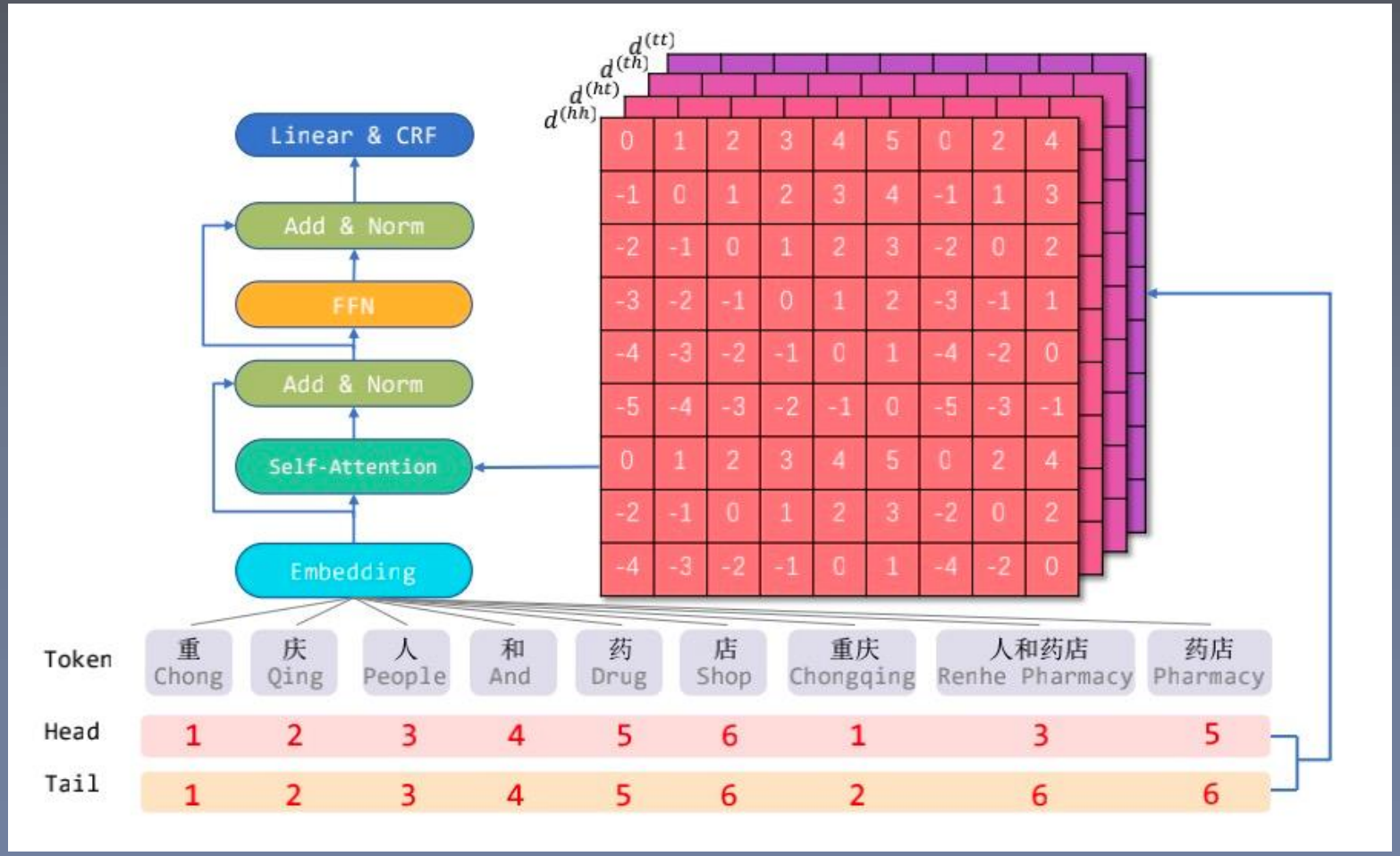


$$d_{ij}^{(hh)} = head[i] - head[j]$$
$$d_{ij}^{(ht)} = head[i] - tail[j]$$
$$d_{ij}^{(th)} = tail[i] - head[j]$$
$$d_{ij}^{(tt)} = tail[i] - tail[j]$$

NER

FLAT

此时得到四个相对距离矩阵：$d^{(hh)}, d^{(ht)}, d^{(th)}, d^{(tt)}$，其中$d_{ij}^{(hh)}$表示$x_i$的开始位置和$x_j$的开始位置的距离。



The overall architecture of FLAT.

NER

FLAT

然后将这四个距离拼接后作一个非线性变换，得到x_i和x_j的位置编码向量R_{ij}：

$$R_{ij} = \text{ReLU}(W_r(\mathbf{p}_{d_{ij}^{(hh)}} \oplus \mathbf{p}_{d_{ij}^{(th)}} \oplus \mathbf{p}_{d_{ij}^{(ht)}} \oplus \mathbf{p}_{d_{ij}^{(tt)}}))$$

W_r是可学习参数，P_d的计算公式如下，其中d是$d^{(hh)}, d^{(ht)}, d^{(th)}, d^{(tt)}$，k是位置编码中的维度索引：

$$\mathbf{p}_d^{(2k)} = \sin\left(d/10000^{2k/d_{model}}\right)$$
$$\mathbf{p}_d^{(2k+1)} = \cos\left(d/10000^{2k/d_{model}}\right)$$

NER

sinusoidal position encoding

采用正余弦方式对位置进行编码，具体的计算公式如下：

$$PE_{(pos,2i)} = sin\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

$$PE_{(pos,2i+1)} = cos\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

看到这个公式可能容易懵，先明确公式中几个字母含义：

PE ： 为二维矩阵，行表示位置，列表示编码向量

pos ： 表示词语在句子中的位置

d_{model}：表示词向量的维度

i ： 表示词向量的位置

NER

sinusoidal position encoding

整个位置编码的矩阵其实是个常量矩阵，并不存在参数需要模型去学习。因此，思考编码方式如何做到计算不同位置之间的距离？

为了便于理解，记 $f_i = 10000^{2*i/d_{model}}$，假设句长为L，pos代表第pos个位置，当 i = 0时，对应位置编码向量的第1维和第2维如下：

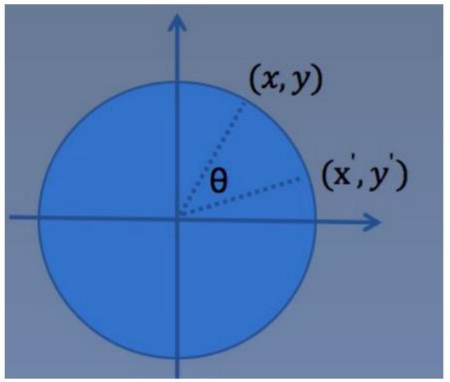
位置编码向量的第1维：$\left[sin(\frac{0}{f_0}), \ldots, sin(\frac{pos}{f_0}), \ldots, sin(\frac{L}{f_0})\right]$

位置编码向量的第2维：$\left[cos(\frac{0}{f_0}), \ldots, cos(\frac{pos}{f_0}), \ldots, cos(\frac{L}{f_0})\right]$

NER

sinusoidal position encoding

f_0是常量。如果整个位置编码只采用二维进行表示，那么第一个位置对应的位置向量也就是$[\sin(\frac{0}{f_0}), \cos(\frac{0}{f_0})]$，第 pos 位置对应位置向量就是$[\sin(\frac{pos}{f_0}), \cos(\frac{pos}{f_0})]$。下面我们看如何计算第一个位置与第 pos 位置之间距离？



第一个位置，我们记为（x,y）。第 pos 位置，记为（x’,y’）。下面是高中知识回顾：

对于任意坐标（x',y'）都可以由（x,y）旋转 θ 角得到，而与起点位置（x,y）无关，矩阵表达：

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

NER

sinusoidal position encoding

通过上面的公式，可以反推出θ，也是一个常量。

所以整合起来解释就是：不同位置之间的距离是可以被相互线性表示。即不同位置之间的相对距离便可以计算出来。

NER

sinusoidal position encoding

bert为何使用learned position embedding而非sinusoidal position encoding？

原版Transformer对learned position embedding和sinusoidal position encoding进行了对比实验，结果很相近。Sinusoidal encoding更简单、更高效、并可以扩展到更长的序列上，因此Transformer采用了sinusoidal position encoding实现。

bert采用了海量的训练数据，learned position embedding有参数，因此比常量的表示能力更强，整个Bert有种大力出奇迹的意思，数据足够多，让模型自己去学习。

NER

FLAT

这样，每一个span都可以与任意span进行充分且直接的交互，论文采用Transformer-XL (ACL 2019)中提出的基于相对位置编码的self-attention：

$$\begin{aligned}\mathbf{A}_{i,j}^{*} = & \mathbf{W}_q^{\intercal}\mathbf{E}_{x_i}^{\intercal}\mathbf{E}_{x_j}\mathbf{W}_{k,E} + \mathbf{W}_q^{\intercal}\mathbf{E}_{x_i}^{\intercal}\mathbf{R}_{ij}\mathbf{W}_{k,R} \\ & + \mathbf{u}^{\intercal}\mathbf{E}_{x_j}\mathbf{W}_{k,E} + \mathbf{v}^{\intercal}\mathbf{R}_{ij}\mathbf{W}_{k,R}\end{aligned}$$

在Transformer-XL中R是根据绝对位置编码计算得出，这里R经过了非线性变换处理。最后，用A^*替换式(1)中的A，继续vanilla Transformer计算，再将其送入CRF层进行解码得到预测的标签序列。

结 语

感谢大家参加本次的比赛直播

课下请一定要

亲自动手操作一次！

深度之眼
deepshare.net

4、互动时间

deepshare.net
深度之眼

联系我们:
电话: 18001992849
邮箱: service@deepshare.net
Q Q: 2677693114

公众号

客服微信