招联金融
MUCFC.COM

WWW.MUCFC.COM

ner技术方案总结

黄凯 20230224

目录 | Contents

问题定义

概述:
NER全称是命名实体识别（Named Entity Recognition, NER），旨在识别文本中专有名词，如位置、组织和时间。

举例:
输入：小明在北京大学的燕园看了中国男篮的一场比赛

输出：B-PER，E-PER，O，B-ORG，I-ORG，I-ORG，E-ORG，O，B-LOC，E-LOC，O，O，B-ORG，I-ORG，I-ORG，E-ORG，O，O，O，O

其中，“小明”以PER，“北京大学”以ORG，“燕园”以LOC，“中国男篮”以ORG为实体类别分别挑了出来。

标注方法:
序列标注:
1.BIO：标识实体的开始、中间和非实体部分
2.BMES：增加S单个实体情况的标注
3.BIOES：增加E实体的结束标识
指针标注:
1.单指针（二维）
2.双指针（一维）

应用场景:

知识图谱、文本理解、对话系统、信息检索、槽位抽取

词典匹配

概述:

词典匹配是工业界最常用的NER技术，尤其是垂直领域的NER任务。经过离线实体库不断的丰富完善累积后，在线使用词典匹配进行实体识别目前美团基于实体库在线ner识别率可以达到92%。

重点工作：实体离线挖掘、字符匹配算法

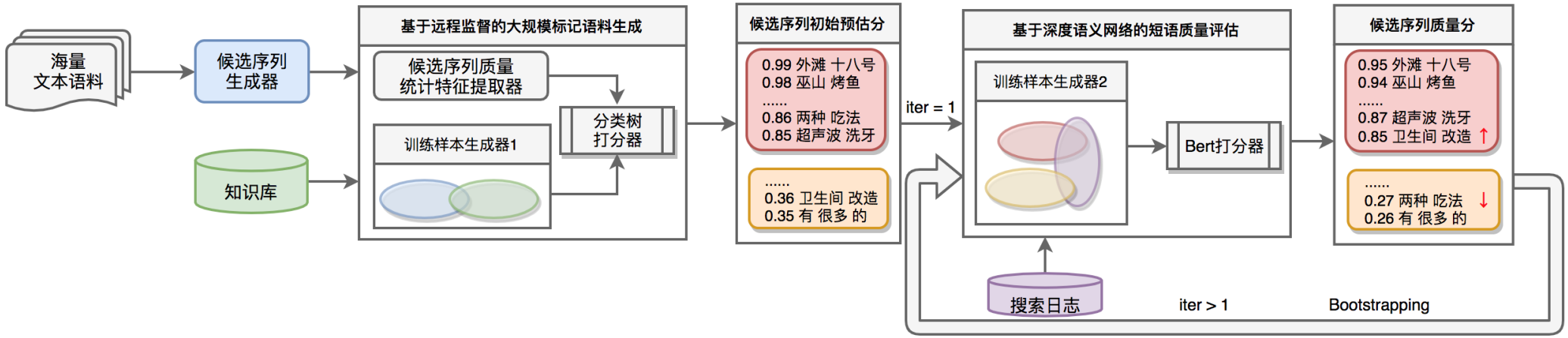
优点：速度快，可解释性强，且精度高
缺点：长期维护、未登录实体

词典匹配—实体离线挖掘

实体来源：

1.领域UGC、会话文本等非结构化数据；2.用户搜索日志；3.百科词条、领域信息库

美团基于用户UGC垂直领域的实体挖掘方案



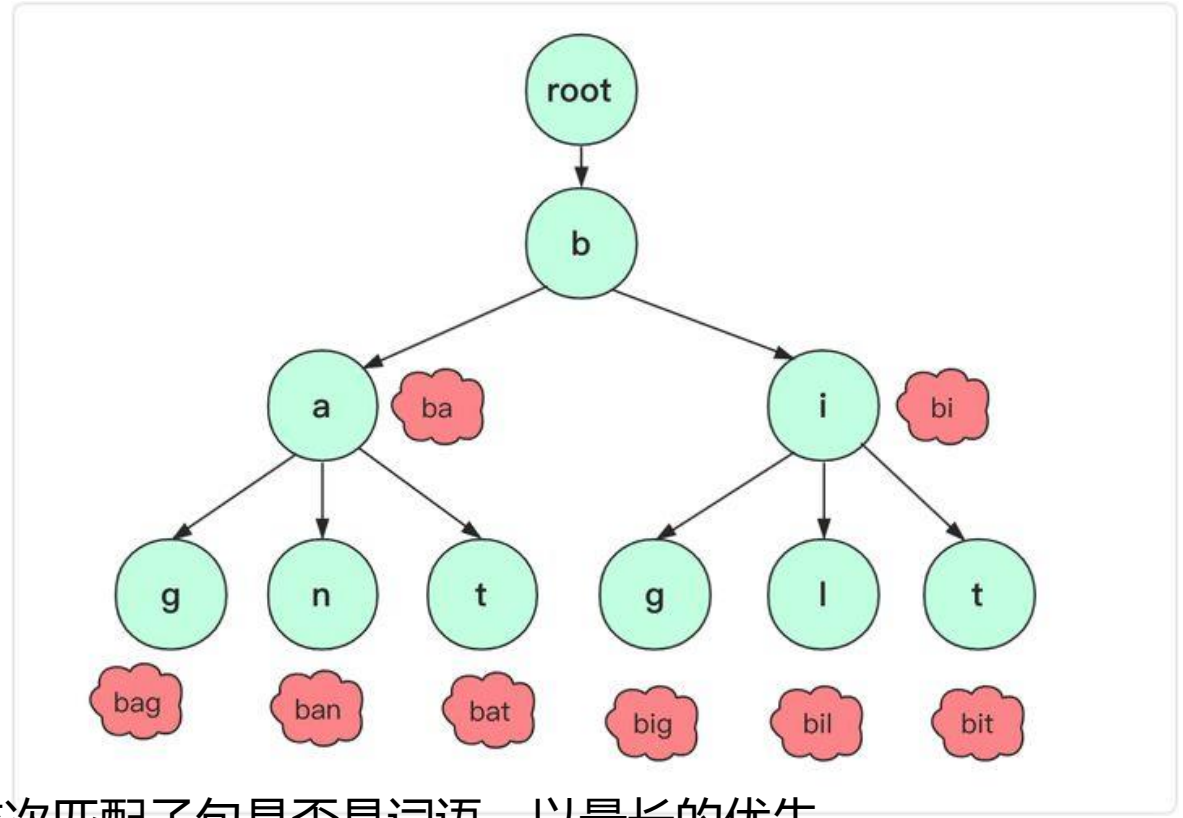
Step1：候选序列挖掘。频繁连续出现的词序列，是潜在新型词汇的有效候选。

Step2：基于远程监督的大规模有标记语料生成。利用领域已累积实体词典作为远程监督词库，将Step1中候选序列与实体词典的交集作为训练正例样本。将基于n-gram的候选短语中不匹配任何知识库的高质量短语的短语放在负向池中。

Step3: 基于深度语义网络的短语质量评估。我们利用搜索日志数据对Step2中生成的大规模正负例池数据进行远程指导，将有大量搜索记录的词条作为有意义的关键词。我们将正例池与搜索日志重合的部分作为模型正样本，而将负例池减去搜索日志集合的部分作为模型负样本，进而提升训练数据的可靠性和多样性。

词典匹配—字符匹配算法

1.构建字典树



2.双向匹配算法
正向最大匹配：从前往后依次匹配子句是否是词语，以最长的优先。
后向最大匹配：从后往前依次匹配子句是否是词语，以最长的优先。

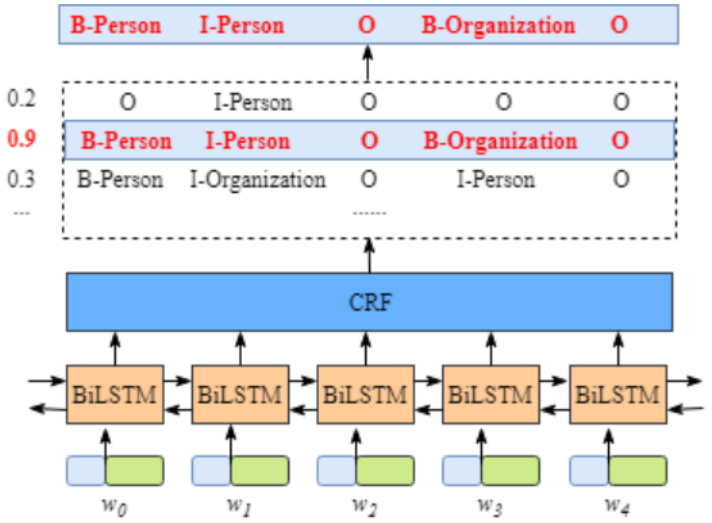
双向最大匹配原则：
覆盖 token 最多的匹配。
句子包含实体和切分后的片段，这种片段+实体个数最少的。

NER模型

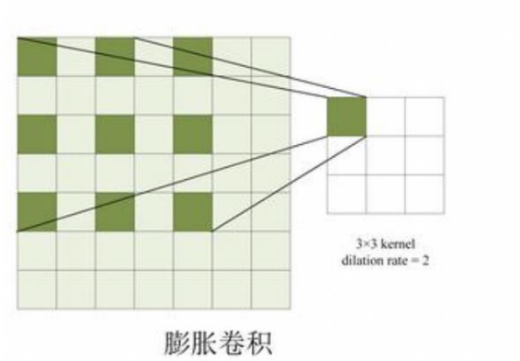
任务类型	模型	提出时间	备注
序列标注	BiLSTM-CRF	2015	
序列标注	IDCNN-CRF	2017	卷积
序列标注	Lattice LSTM	2018	词汇增强
序列标注	BERT-CRF	2019	
序列标注	BERT-cascade-CRF	2020	流式NER
序列标注	FLAT	2020	词汇增强
指针标注	Biaffine	2020	
指针标注	global-pointer	2022	
指针标注	NER_MRC	2022	

NER模型——NN-CRF框架

框架图：



IDCNN:引入空洞卷积，去掉池化层，最大化保留全局信息



膨胀卷积

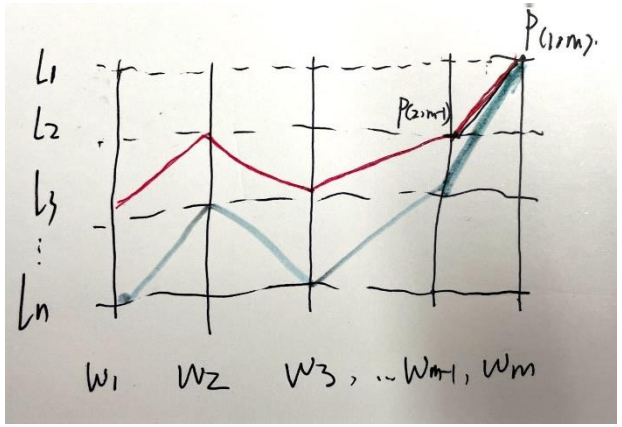
NN-CRF中CRF计算过程:

目标：给每一个可能的标注序列算一个分数，通过训练，使得那个唯一的真实的标签序列得分最高

假设文本长度m，标签为n，则优化目标：

$$Prob = \frac{e^{p_{realpath}}}{e^{p_1} + e^{p_2} + \ldots + e^{p_{m^n}}} \longrightarrow -log(Prob) = log(e^{p_1} + e^{p_2} + \ldots + e^{p_{m^n}}) - p_{realpath}$$

如何计算某一种路径的分数?
对于第m个字符的输出，对于标签为1的路径节点$p_{1,m}$:



$$p_{1,m} = \log \sum_{i=1}^{n} e^{p_{i,m-1} + e_{m1} + t_{i1}}$$

e_{m1}为w_m对标签l_1的发射概率，由NN网络提供；
t_{i1}为l_i到l_1的转移概率，由crf提供；
由以上公式，可以递归求出所有路径的分数，
其中的主要参数：
发射概率矩阵E_{m*n}
转移概率矩阵T_{n*n}

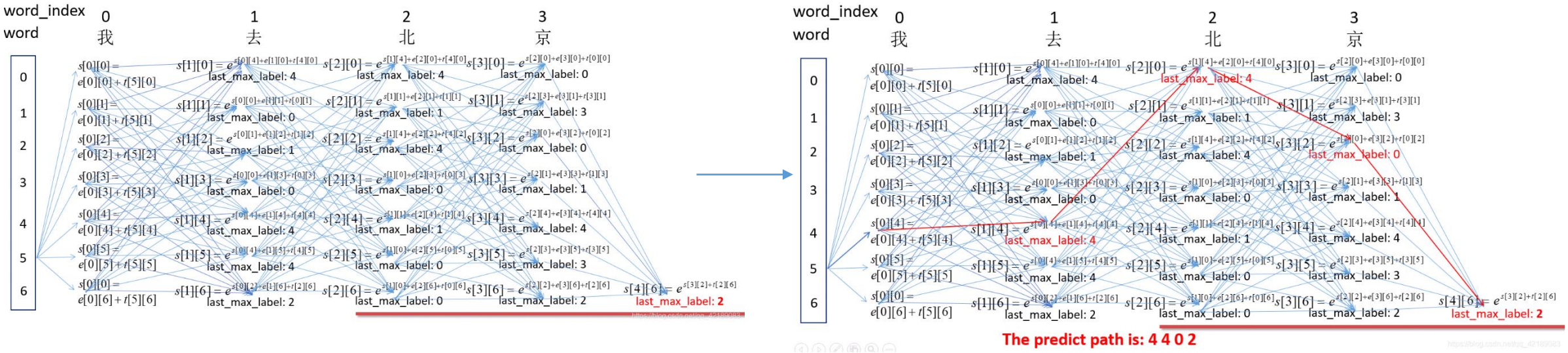
NER模型——NN-CRF框架

假设文本长度m，标签个数为n，则发射矩阵和转移矩阵：

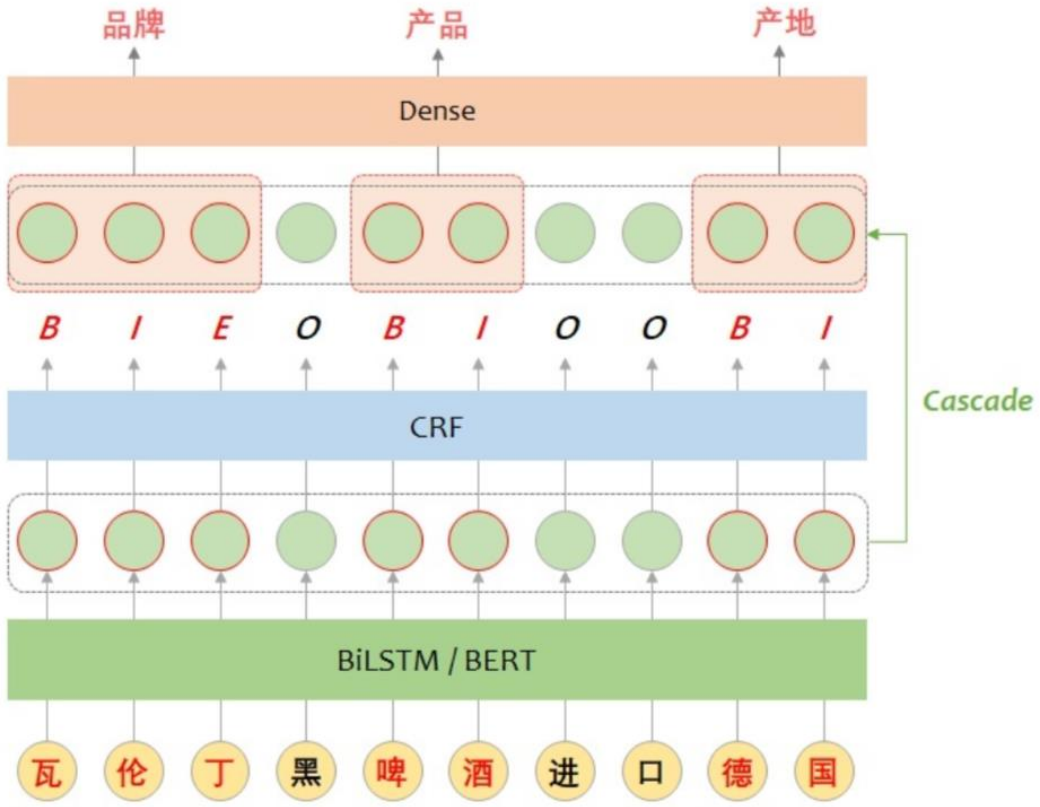
$$\begin{bmatrix} e_{1,1} e_{2,1} & \dots e_{m,1} \\ e_{1,2} e_{2,2} & \dots e_{m,2} \\ \dots \\ e_{1,n} e_{2,n} & \dots e_{m,n} \end{bmatrix} \qquad \begin{bmatrix} t_{1,1} t_{2,1} & \dots t_{n,1} \\ t_{1,2} t_{2,2} & \dots t_{n,2} \\ \dots \\ t_{1,n} t_{2,n} & \dots t_{n,n} \end{bmatrix}$$

已知发射矩阵和转移矩阵，模型如何输出最佳路径？

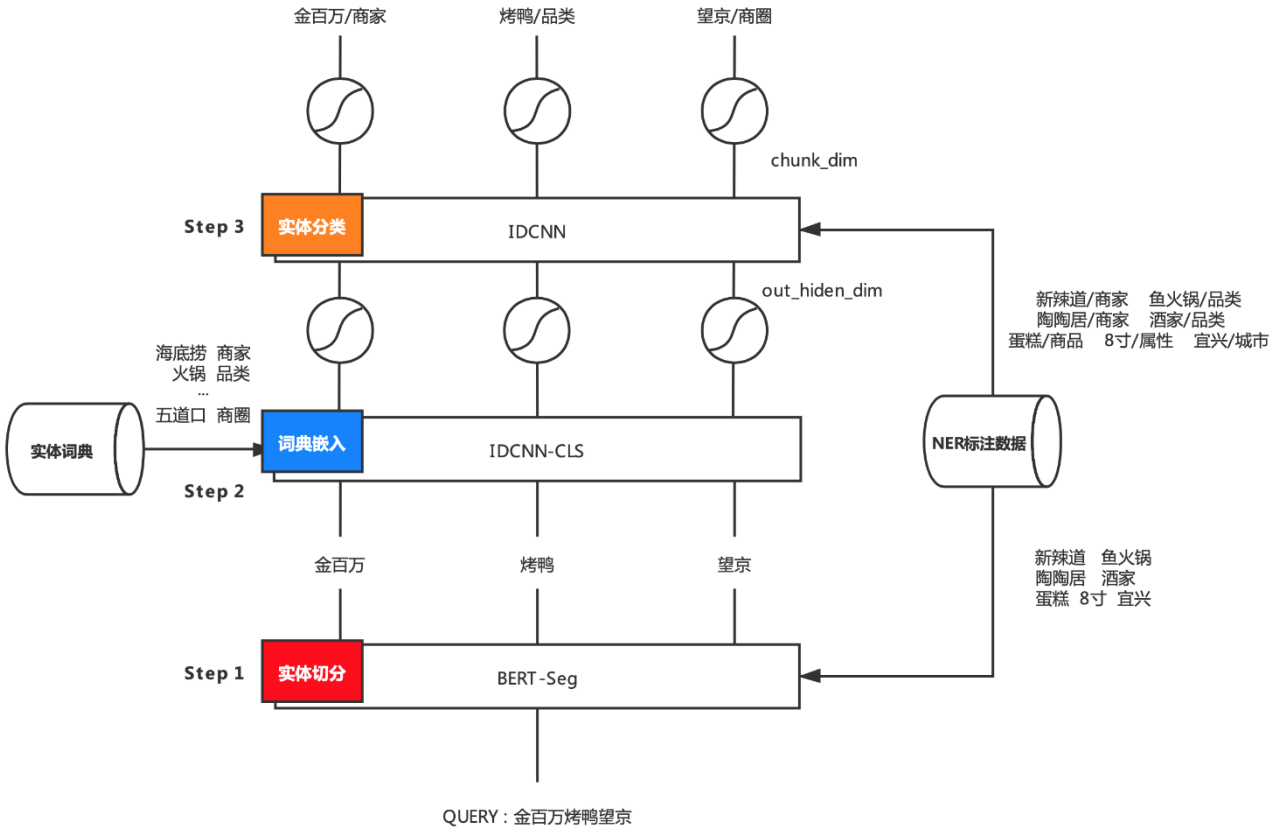
维特比算法（动态规划）：每一个节点均为当前节点的最优路径



NER模型—— cascade-CRF框架



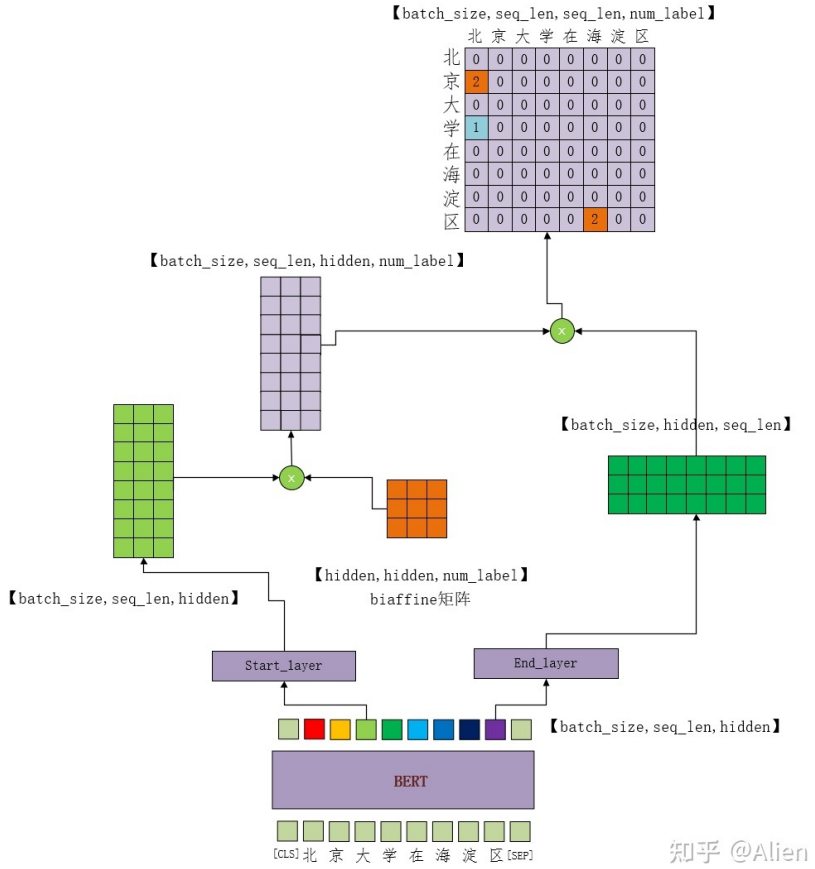
美团方案：



NER模型—— biaffine

概述：

单指针标注，引入评分矩阵，对所有的spans进行评分，最后达到实体抽取的目的。



核心：引入biaffine评分矩阵，矩阵大小为l*l*c，
l为句子的长度，c是标签个数，则biaffine的计算公式如下：

$$
\begin{aligned}
h_s(i) &= \mathrm{FFNN}_s(x_{s_i}) \\
h_e(i) &= \mathrm{FFNN}_e(x_{e_i}) \\
r_m(i) &= h_s(i)^{\top} \mathrm{U}_m h_e(i) \\
&\quad + W_m(h_s(i) \oplus h_e(i)) + b_m
\end{aligned}
$$

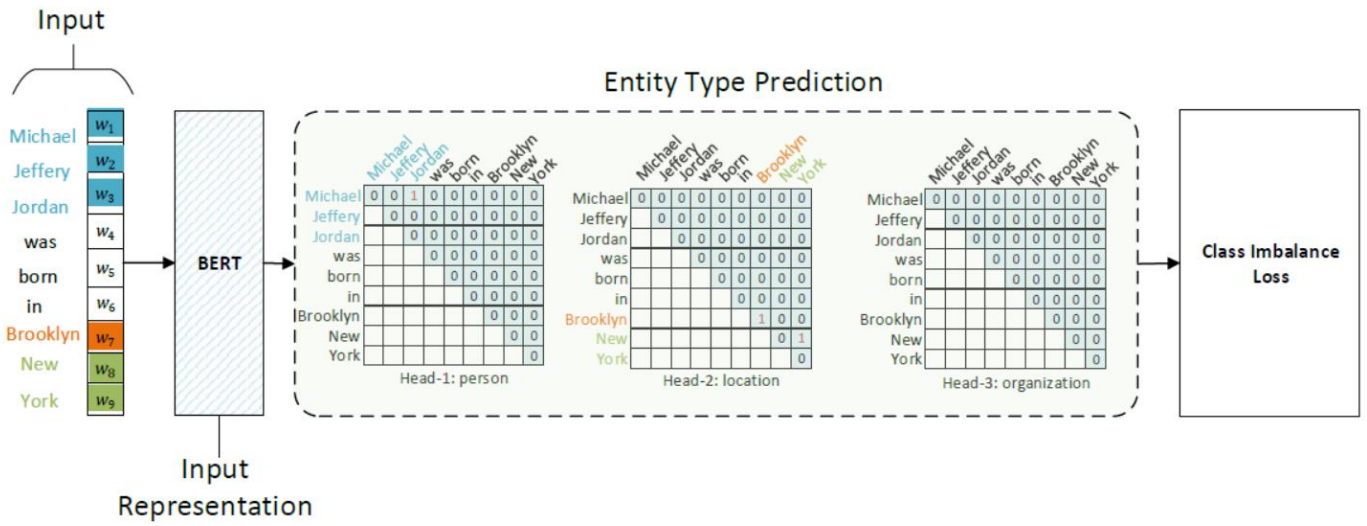
其中s_i和e_i是实体的开始和结束索引，U_m是d×c×d张量，W_m是2d×c矩阵，b_m是偏差，评分张量（rm） l×l×c

矩阵计算：

```
# Mapping matrix
bilinear_map = tf.get_variable(
    'bilinear_map', [vector_set_1_size, output_size, vector_set_2_size],
    initializer=initializer)

...
vector_set_1 = tf.reshape(vector_set_1, [-1, vector_set_1_size])

# [v1, r, v2] -> [v1, r*v2]
bilinear_map = tf.reshape(bilinear_map, [vector_set_1_size, -1])

# [b*n, v1] x [v1, r*v2] -> [b*n, r*v2]
bilinear_mapping = tf.matmul(vector_set_1, bilinear_map)

# [b*n, r*v2] -> [b, n*r, v2]
bilinear_mapping = tf.reshape(
    bilinear_mapping,
    [batch_size, bucket_size * output_size, vector_set_2_size])

# [b, n*r, v2] x [b, n, v2]T -> [b, n*r, n]
bilinear_mapping = tf.matmul(bilinear_mapping, vector_set_2, adjoint_b=True)

# [b, n*r, n] -> [b, n, r, n]
bilinear_mapping = tf.reshape(
    bilinear_mapping, [batch_size, bucket_size, output_size, bucket_size])
return bilinear_mapping
```

NER模型—— global-pointer

特点：
1.指针网络
2.多标签分类问题
3.引用multi-head构建评分矩阵
4.ROPE旋转式编码引入位置信息
5.数据不平衡处理



基于multi-head打分矩阵计算公式：

$$h_1, h_2, ...h_n = PLM(x_1, x_2, ...x_n).$$

$$q_{i,\alpha} = W_{q,\alpha} h_i + b_{q,\alpha},$$

$$k_{i,\alpha} = W_{k,\alpha} h_i + b_{k,\alpha},$$

$$s_\alpha(i,j) = q_{i,\alpha}^\top k_{j,\alpha}$$

1.$[h_1,h_2,\cdots,h_n]$为长度为n的文本经过编码得到的向量序列
2.$q_{i,\alpha}$和$k_{i,\alpha}$分别为接入FFN网络后的矩阵
3.$S_\alpha(i,j)$为实体类型α的打分矩阵。
4.multi-head的个数为实体类别的个数

ROPE旋转式编码

$$\begin{aligned} s_\alpha(i,j) &= (\mathcal{R}_i q_{i,\alpha})^\top (\mathcal{R}_j k_{j,\alpha}) \\ &= q_{i,\alpha}^\top \mathcal{R}_i^\top \mathcal{R}_j k_{j,\alpha} \\ &= q_{i,\alpha}^\top \mathcal{R}_{j-i} k_{j,\alpha} \end{aligned}$$

数据不平衡处理

$$\log\left(1 + \sum_{(i,j)\in P_\alpha} e^{-s_\alpha(i,j)}\right) + \log\left(1 + \sum_{(i,j)\in Q_\alpha} e^{s_\alpha(i,j)}\right)$$

NER模型—— ner-mrc框架

概述：将NER的序列标注任务看作一项MRC任务，此方法的优点1.可以引入query先验知识；2.对于不同类别的嵌套实体回答不同的独立问题，可以解决嵌套实体问题。

方法步骤：

1.构造三元组集合（content,query,answer）：

例如 “我在招联金融工作一年半了”，实体为“招联金融”，则三元组集合为：（“我在招联金融工作一年半了”，“请在文本中找出公司名称”，“招联金融”），输入到bert中的格式[CLS] query [SEP] content [SEP]

2.构造T_{start}矩阵和T_{end}矩阵，则，每个字符分别作为start和end的概率值由以下两公式得到：

$$p_{start} = softmax_{each\ row}(E.T_{start})$$

$$p_{end} = softmax_{each\ row}(E.T_{end})$$

3.获得start指针和end指针的位置：

$$\hat{I}_{\text{start}} = \{i \mid \text{argmax}(P_{\text{start}}^{(i)}) = 1, i = 1, \cdots, n\}$$
$$\hat{I}_{\text{end}} = \{j \mid \text{argmax}(P_{\text{end}}^{(j)}) = 1, j = 1, \cdots, n\}$$

4.Start指针和end指针匹配模型：

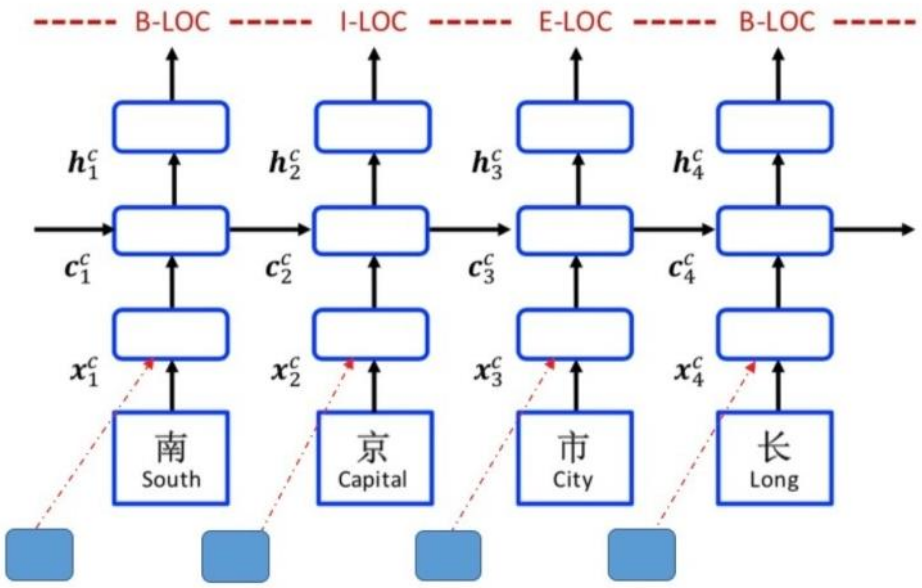
$$P_{i_{\text{start}}, j_{\text{end}}} = \text{sigmoid}(m \cdot \text{concat}(E_{i_{\text{start}}}, E_{j_{\text{end}}}))$$

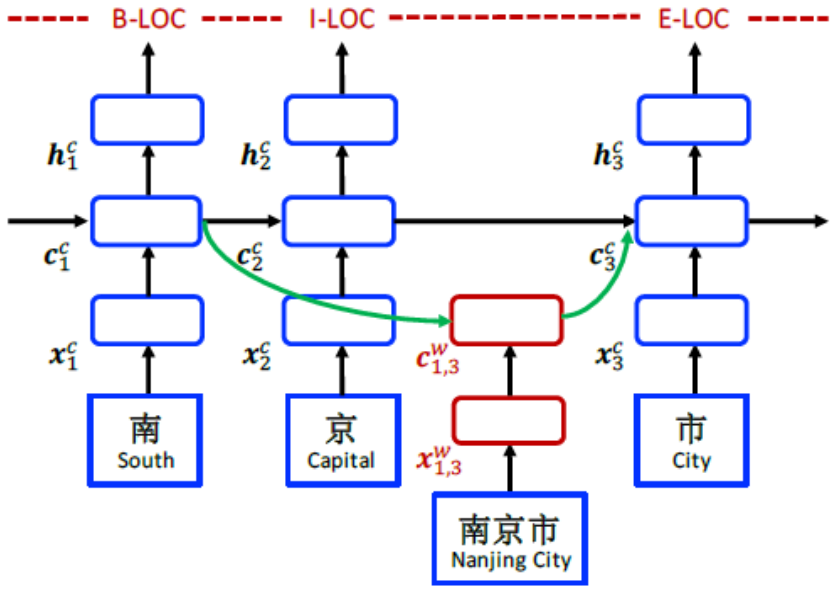
Loss:

$$\mathcal{L}_{\text{start}} = \text{CE}(P_{\text{start}}, Y_{\text{start}})$$
$$\mathcal{L}_{\text{end}} = \text{CE}(P_{\text{end}}, Y_{\text{end}})$$
$$\mathcal{L}_{\text{span}} = \text{CE}(P_{\text{start,end}}, Y_{\text{start, end}})$$

$$\mathcal{L} = \alpha\mathcal{L}_{\text{start}} + \beta\mathcal{L}_{\text{end}} + \gamma\mathcal{L}_{\text{span}}$$

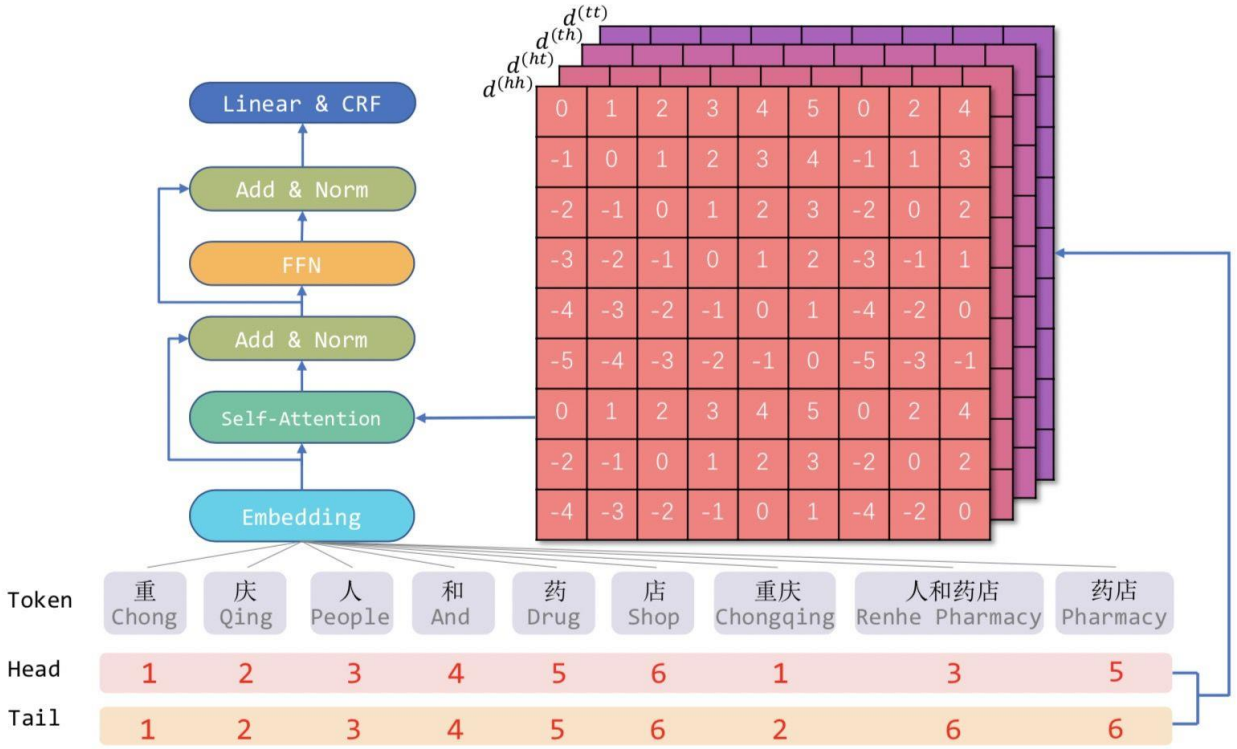
Lattice LSTM





NER问题扩展——词汇增强

FLAT



引入相对位置：

$$
\begin{aligned}
d_{ij}^{(hh)} &= head[i] - head[j], \\
d_{ij}^{(ht)} &= head[i] - tail[j], \\
d_{ij}^{(th)} &= tail[i] - head[j], \\
d_{ij}^{(tt)} &= tail[i] - tail[j],
\end{aligned}
$$

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

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

公司业务场景

1.信息脱敏，在对话文本中涉及个人隐私信息需要脱敏处理。

2.催收会话文本信息抽取：如承诺日期、发薪日期、电话号码、出生年月等实体。

3.专有实体的扩充和维护。

感谢聆听