

第八章 知识图谱

戴洪良

计算机科学与技术学院/人工智能学院

hongldai@nuaa.edu.cn



从文本中抽取知识

- **命名实体识别**
- 细粒度实体分类
- 关系抽取
- 事件抽取

命名实体识别 (Named Entity Recognition, NER)

3月20日消息，微软旗下语音识别子公司Nuance今日发布一款 AI 临床笔记软件，命名为DAX Express，主要面向医护人员。

目标输出：

实体提及 ¹	位置	类别
3月20日	(0, 4)	日期
微软	(8, 9)	机构
Nuance	(20, 25)	机构
DAX Express	(46, 56)	产品

- 识别文本中的实体（如人名、地名、机构名、产品名）并分类
 - 要求1：确定实体的位置（Span）
 - 要求2：实体分类
- 要识别的实体类别根据实际应用场景预先设定

¹ 英文为Entity Mention, 在没有歧义的情况下也可简称为实体

NER数据集

- CoNLL 2003
 - 英文和德文新闻文本； PER、 LOC、 ORG、 MISC四种类别
- ACE 2005
 - 英文、 中文、 阿拉伯语； 数据源包括broadcast news, newsgroups, telephone conversations等； PER、 LOC、 ORG、 FAC等7种类别； 包含嵌套（nested） 实体
- GENIA
 - 生物医疗文本； DNA、 RNA、 cell_type、 protein、 cell_line五类
- MSRA NER
 - 中文； 新闻文本； PER, LOC, ORG三类
- Weibo NER
 - 中文； 微博文本； PER, LOC, GPE, ORG四类

NER方法

- 基于规则和词典的方法
- 基于传统机器学习的方法
- 基于神经网络的方法
- 基于迁移学习的方法
- 基于预训练模型的方法
- 结合规则与神经网络模型的方法
- 基于大模型的方法

基于规则和词典的实体识别

- 思路：使用实体词典或规则匹配文本中的实体
- 基本流程：
 - 准备词典
 - 预处理
 - 句子划分；分词；词性标注等
 - 实体识别
 - 词典匹配；规则匹配

基于规则和词典的实体识别

- 准备词典
 - 词典中应包含实体词及对应的实体类别
 - 构建方法：人工构建；利用已有知识库或ontology、术语词典；基于标注数据构建等

例：

实体词	实体类别
nuclear complexes	<i>protein</i>
A6H monoclonal antibody	<i>protein</i>
CCAAT sequences	<i>DNA</i>
ORF2	<i>DNA</i>
TAL1 gene	<i>DNA</i>
activator cells	<i>cell_line</i>
adult erythroid cells	<i>cell_type</i>

基于规则和词典的实体识别

- 预处理

Nitric oxide decreases cytokine-induced endothelial activation.



分词加词性标注

JJ NN VBZ NN HYPH VBN JJ NN .
Nitric oxide decreases cytokine - induced endothelial activation .

基于规则和词典的实体识别

- 预处理

例：使用hanlp做分词及词性标注

```
import hanlp

# 加载模型
HanLP = hanlp.load(hanlp.pretrained.mtl.CLOSE_TOK_POS_NER_SRL_DEP_SDP_CON_ELECTRA_SMALL_ZH)
# 处理文本
result = HanLP(['我们经过了南京市长江大桥'])

print(result['tok/fine']) # 分词结果
print(result['pos/pku']) # 词性标注结果
```

我们经过了南京市长江大桥



我们	经过	了	南京市	长江	大桥
r	v	u	ns	ns	n

基于规则和词典的实体识别

- 预处理
 - 划分句子；分词；词性标注等
- 作用：
 - 以词为单位匹配，避免错误地将词切开
 - 基于词性标注，提升词典匹配准确率（只匹配名词或名词短语）
 - 可基于词性标注结果编写匹配规则

基于规则和词典的实体识别

- 匹配规则
 - 可用正则表达式、正则表达式加代码判断等

规则	类别
i[3579]	CPU
\d\d?GB?内存	内存
\dTB?(固态)?硬盘	硬盘
\d{3,5}元	价格

惠普星Book Pro
i5, 16G内存, 1T硬盘, 5399元左右
优势:轻薄、2.8K高刷屏、散热好、颜值高
金属机身
注意:无独显不适合玩大型游戏, 接口少



惠普星Book Pro
i5, **16G内存**, **1T硬盘**, **5399元**左右
优势:轻薄、2.8K高刷屏、散热好、颜值高
金属机身
注意:无独显不适合玩大型游戏, 接口少

基于规则和词典的实体识别

- 优点

- 无需标注训练数据和训练机器学习模型
 - 实践中可考虑用基于规则和词典的方法自动标注大量数据，再加少量人工标注数据训练深度学习模型
- 对于一些固定且无歧义的实体词，可以准确识别
- 通过扩充词典可应对新出现的实体
 - 前提是已知该实体对应的实体词
- 识别结果可解释

- 缺点

- 无法识别不被词典和规则覆盖的实体
 - 影响召回率
- 受自然语言多样性影响，多数情况下不能保证识别质量
- 不易维护

基于传统机器学习的方法

- 主要的有：
 - 隐马尔可夫模型 (Hidden Markov Model, HMM)
 - 最大熵马尔可夫模型 (Maximum Entropy Markov Models, MEMM)
 - 条件随机场 (Conditional Random Fields, CRF)
 - 支持向量机 (Support Vector Machine, SVM)

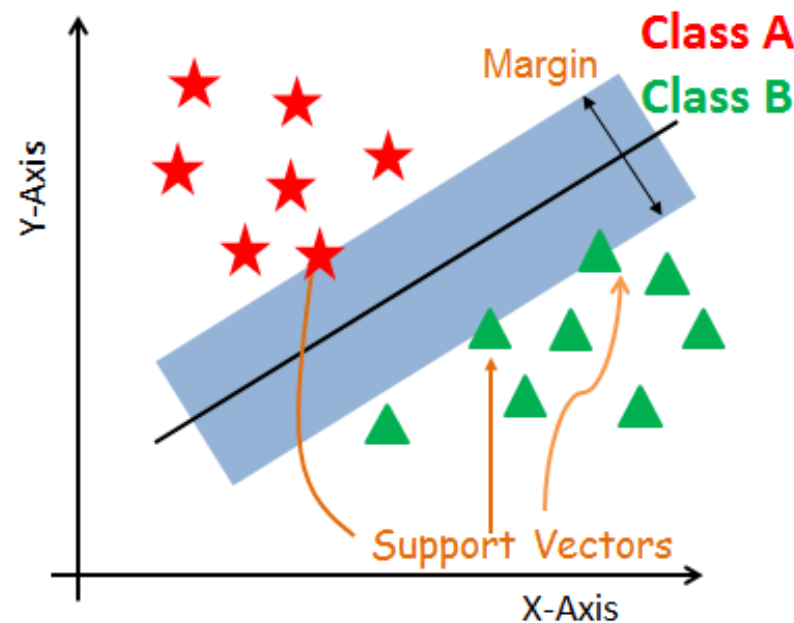
将NER转化为分类问题

- 序列标注
 - 通过序列标注将NER转化为对文本中词的分类问题
 - 序列标注体系主要有IO、BIO、BIOES等

Words	IO	BIO	BIOES
New	I-LOC	B-LOC	B-LOC
York	I-LOC	I-LOC	I-LOC
City	I-LOC	I-LOC	E-LOC
mayor	O	O	O
Eric	I-PER	B-PER	B-PER
Adams	I-PER	I-PER	E-PER
went	O	O	O
to	O	O	O
Mexico	I-LOC	B-LOC	S-LOC

基于支持向量机 (SVM) 的NER

- 支持向量机
 - 基本思想：选择以最大间隔 (margin) 将两类数据点分开的超平面得到分类器



线性可分情况下的优化问题：

$$\min_{\mathbf{w}, b} \frac{\mathbf{w}^T \mathbf{w}}{2}$$

subject to: $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1$ (\forall data points \mathbf{x}_i).

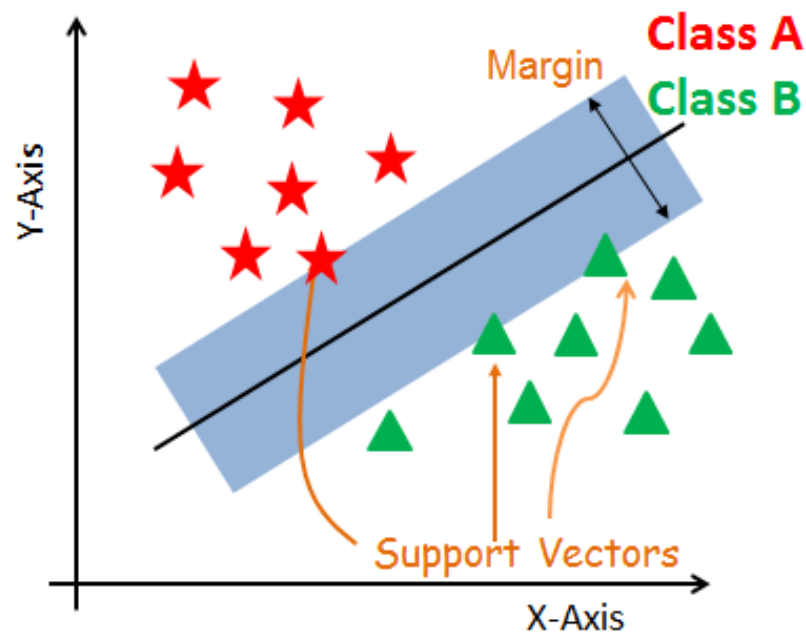
基于支持向量机 (SVM) 的NER

- 支持向量机

- 基本思想：选择以最大间隔 (margin) 将两类数据点分开的超平面得到分类器

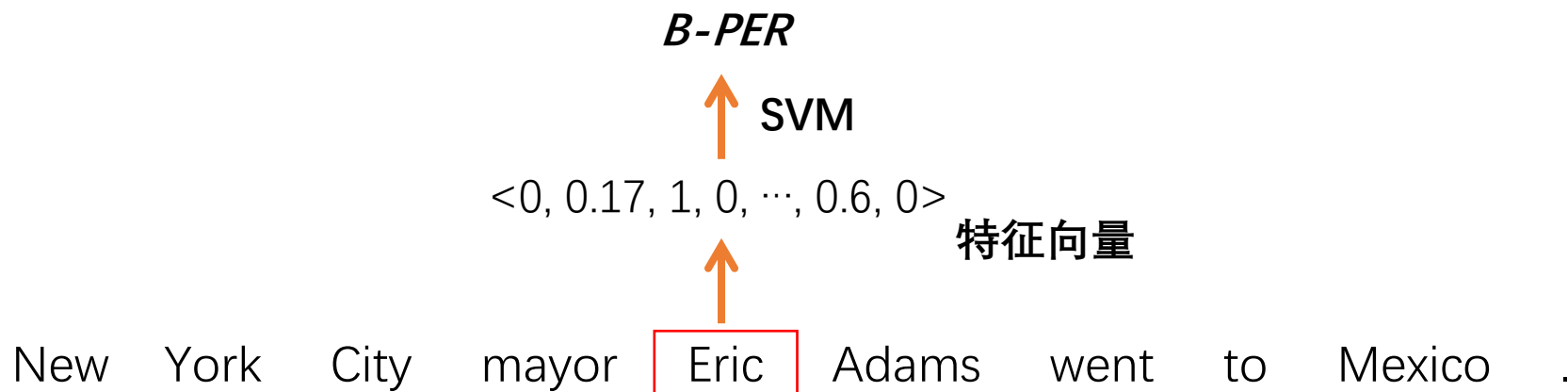
- 多分类策略

- one-vs-all
 - 如有K个类别则学习K个分类器，每个区分其中一个类别和剩下所有类别
 - one-vs-one
 - 为每两个类别学习一个分类器，共 $K*(K-1)/2$ 个分类器



基于支持向量机 (SVM) 的NER

- 为每个词得到一个特征向量
 - 手动设计特征
- 使用SVM基于特征向量分类，从而为每个词得到相应的序列标注标签



基于支持向量机的NER

- 特征

- 人工设计各类特征，如：前一个词、当前词、当前词的词法标签、当前词的词形等。可用one-hot encoding将词转变为向量。

例： $\mathbf{x} = \{w_{-1,1}, w_{-1,2}, \dots, w_{-1,|\mathcal{V}|}, w_{0,1}, \dots, w_{0,|\mathcal{V}|},$
 $pre_{-1,1}, \dots, pre_{0,|\mathcal{P}|}, pc_{-1,1}, \dots, pc_{-1,K}\}$

$$w_{k,i} = \begin{cases} 1 & \text{if a word at } k, W_k, \text{ is the } i\text{th word} \\ & \text{in the vocabulary } \mathcal{V} \\ 0 & \text{otherwise (word feature)} \end{cases}$$

$$pre_{k,i} = \begin{cases} 1 & \text{if } W_k \text{ starts with the } i\text{th prefix} \\ & \text{in the prefix list } \mathcal{P} \\ 0 & \text{otherwise (prefix feature)} \end{cases}$$

$$pc_{k,i} = \begin{cases} 1 & \text{if } W_k (k < 0) \text{ was assigned } i\text{th class} \\ 0 & \text{otherwise (preceding class feature)} \end{cases}$$

基于支持向量机的NER

- 特征

- 例：词形特征

Feature description	Example text	Feature description	Example text
1-digit number	3	Number contains alpha and slash	1/10 th
2-digit number	30	All capital word	NCU
4-digit number	2004	Capital period (only one)	M.
Year decade	2000s	Capital periods (more than one)	I.B.M.
Only digits	1234	Alpha contains money	US\$
Number contains one slash	3/4	Alpha and periods	Mr.
Number contains two slash	2004/8/10	Capital word	Taiwan
Number contains money	\$199	Number and alpha	F-16
Number contains percent	100%	Initial capitalization	Mr., Jason
Number contains hyphen	1-2	Inner capitalization	WordNet
Number contains comma	19,999	All lower case	am, is, are
Number contains period	3.141	Others	3V4
Number contains colon	08:00		

基于该表可将词形表示为向量。如“42”表示为向量<0, 1, 0, 0, 1, 0, 0, ..., 0>

基于条件随机场 (CRF) 的NER

- 也是基于特征预测标签
- CRF是一种判别式 (discriminative) 无向图模型 (undirected graphical model)
 - 根据以随机变量为节点的无向图对 $p(\mathbf{y}|\mathbf{x})$ 建模
- 在NER中一般使用Linear-chain CRF
- 可以有效建模相邻词标签间的相关关系，对标签序列作整体预测
 - SVM一个一个地预测标签，而非整体预测标签序列

Linear-chain CRF

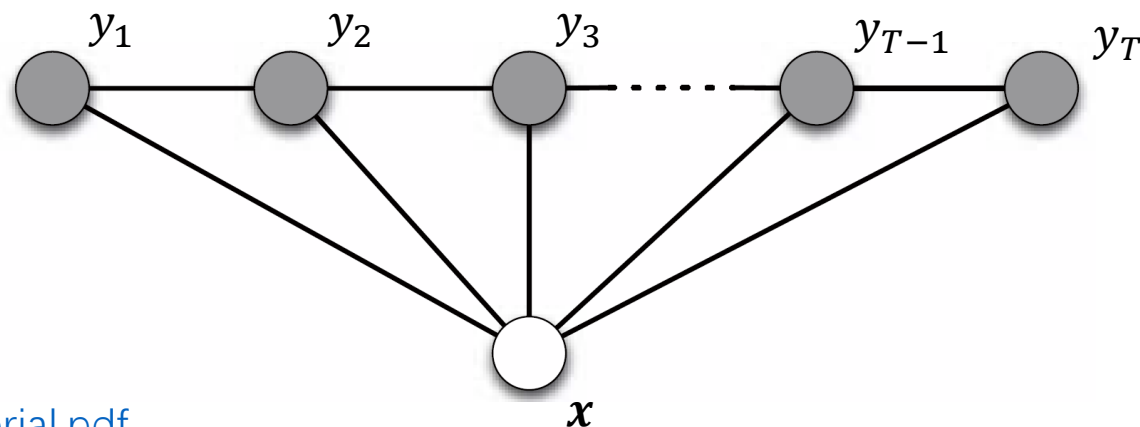
- Linear-chain CRF将 $p(\mathbf{y}|\mathbf{x})$ 建模为:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^T \Psi_t(y_t, y_{t-1}, \mathbf{x}_t)$$

- 引入特征, 定义为:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^T \exp \left\{ \sum_{k=1}^K \theta_k f_k(y_t, y_{t-1}, \mathbf{x}_t) \right\} \quad \text{其中 } f_k \text{ 为特征函数, } \theta_k \text{ 为可训练参数}$$

\mathbf{x}_t : 表示全局输入序列中用于计算第 t 步特征的部分



Linear-chain CRF

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^T \exp \left\{ \sum_{k=1}^K \theta_k f_k(y_t, y_{t-1}, \mathbf{x}_t) \right\}$$

特征例: $f_{ij}^{\text{LL}}(y_t, y_{t-1}, \mathbf{x}_t) = \mathbf{1}_{\{y_t=i\}} \mathbf{1}_{\{y_{t-1}=j\}} \forall i, j \in \mathcal{Y}$

$$f_{ib}^{\text{LO}}(y_t, y_{t-1}, \mathbf{x}_t) = \mathbf{1}_{\{y_t=i\}} q_b(\mathbf{x}_t) \quad \forall i \in \mathcal{Y}$$

其中 q_b 为observation function, 每个根据 \mathbf{x}_t 得到一个特征值

Linear-chain CRF

特征例: $f_{ib}^{\text{LO}}(y_t, y_{t-1}, \mathbf{x}_t) = \mathbf{1}_{\{y_t=i\}} q_b(\mathbf{x}_t) \quad \forall i \in \mathcal{Y}$

q_b 为 observation function

W= v	$w_t = v$	$\forall v \in \mathcal{V}$
T= j	part-of-speech tag for w_t is j (as determined by an automatic tagger)	$\forall \text{POS tags } j$
P=I- j	w_t is part of a phrase with syntactic type j (as determined by an automatic chunker)	
Capitalized	w_t matches $[A-Z][a-z]^+$	
Allcaps	w_t matches $[A-Z][A-Z]^+$	
EndsInDot	w_t matches $[\^\.]+.*\.$	
	w_t contains a dash	
	w_t matches $[A-Z]^+[a-z]^+[A-Z]^+[a-z]^+$	
Acro	w_t matches $[A-Z][A-Z\\\.]*\\. [A-Z\\\.]^*$	
Stopword	w_t appears in a hand-built list of stop words	
CountryCapital	w_t appears in list of capitals of countries	
:	many other lexicons and regular expressions	

实际使用Linear-chain CRF时，与SVM类似，直接设计特征即可

CRF训练

- 目标函数
 - Log likelihood

$$\ell(\theta) = \sum_{i=1}^N \log p(\mathbf{y}^{(i)} | \mathbf{x}^{(i)}; \theta)$$

$$\ell(\theta) = \sum_{i=1}^N \sum_{t=1}^T \sum_{k=1}^K \theta_k f_k(y_t^{(i)}, y_{t-1}^{(i)}, \mathbf{x}_t^{(i)}) - \sum_{i=1}^N \log Z(\mathbf{x}^{(i)})$$

可用随机梯度下降（SGD）训练

CRF Inference

- CRF推理 (inference) 的目标是找到使 $p(\mathbf{y}|\mathbf{x})$ 最大的 \mathbf{y}

$$\mathbf{y}^* = \operatorname{argmax}_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}; \theta)$$

其中
$$p(\mathbf{y}|\mathbf{x}; \theta) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^T \exp \left\{ \sum_{k=1}^K \theta_k f_k(y_t, y_{t-1}, \mathbf{x}_t) \right\}$$

- 序列中一个词的标签既受它前面词的标签影响，又受它后面词的标签影响，因此不能一个一个词地单独预测

CRF Inference

- CRF推理 (inference) 的目标是找到使 $p(\mathbf{y}|\mathbf{x})$ 最大的 \mathbf{y}

$$\mathbf{y}^* = \operatorname{argmax}_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}; \theta)$$

其中
$$p(\mathbf{y}|\mathbf{x}; \theta) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^T \exp \left\{ \sum_{k=1}^K \theta_k f_k(y_t, y_{t-1}, \mathbf{x}_t) \right\}$$

令
$$F(y_t, y_{t-1}, \mathbf{x}_t; \theta) = \sum_{k=1}^K \theta_k f_k(y_t, y_{t-1}, \mathbf{x}_t)$$

对 $p(\mathbf{y}|\mathbf{x}; \theta)$ 取log, 忽略 $Z(\mathbf{x})$ 则
$$\mathbf{y}^* = \operatorname{argmax}_{\mathbf{y}} \sum_{t=1}^T F(y_t, y_{t-1}, \mathbf{x}_t; \theta)$$

Viterbi算法

$$\mathbf{y}^* = \operatorname{argmax}_{\mathbf{y}} \sum_{t=1}^T F(y_t, y_{t-1}, \mathbf{x}_t; \theta)$$

- Viterbi算法是一个动态规划算法
- 令 $d(t, y)$ 为：在只考虑前 t 个标签的情况下，以 $y_t = y$ 结尾的标签序列能达到的最大分数
 - $d(t + 1, y) = \max_{y'} (d(t, y') + F(y, y', \mathbf{x}_t; \theta))$

Viterbi算法 – 例

$$d(t, y) = \max_{y'} (d(t-1, y') + F(y, y', \mathbf{x}_t; \theta))$$

$t = 1$

$d(0, y'_0)$	0
--------------	---

$F(A, y'_0, x_1; \theta)$	2
---------------------------	---

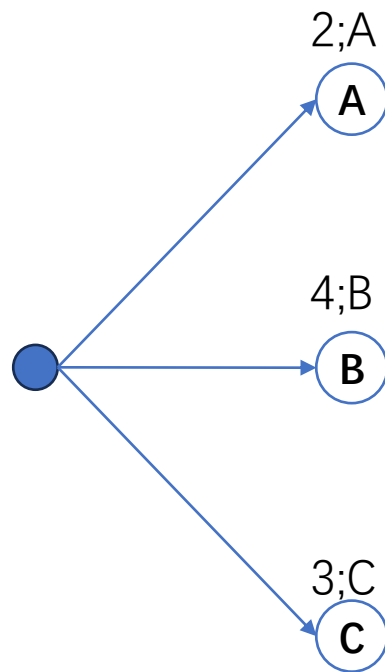
$F(B, y'_0, x_1; \theta)$	4
---------------------------	---

$F(C, y'_0, x_1; \theta)$	3
---------------------------	---

$d(1, A)$	2
-----------	---

$d(1, B)$	4
-----------	---

$d(1, C)$	3
-----------	---

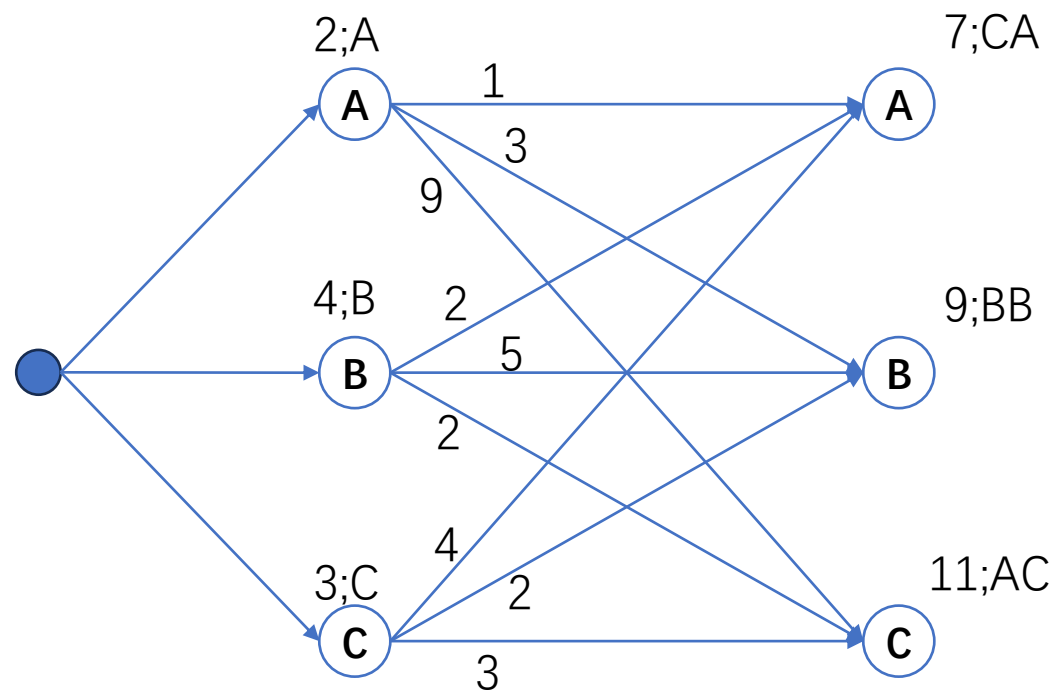


Viterbi算法 – 例

$$d(t, y) = \max_{y'} (d(t-1, y') + F(y, y', \mathbf{x}_t; \theta))$$

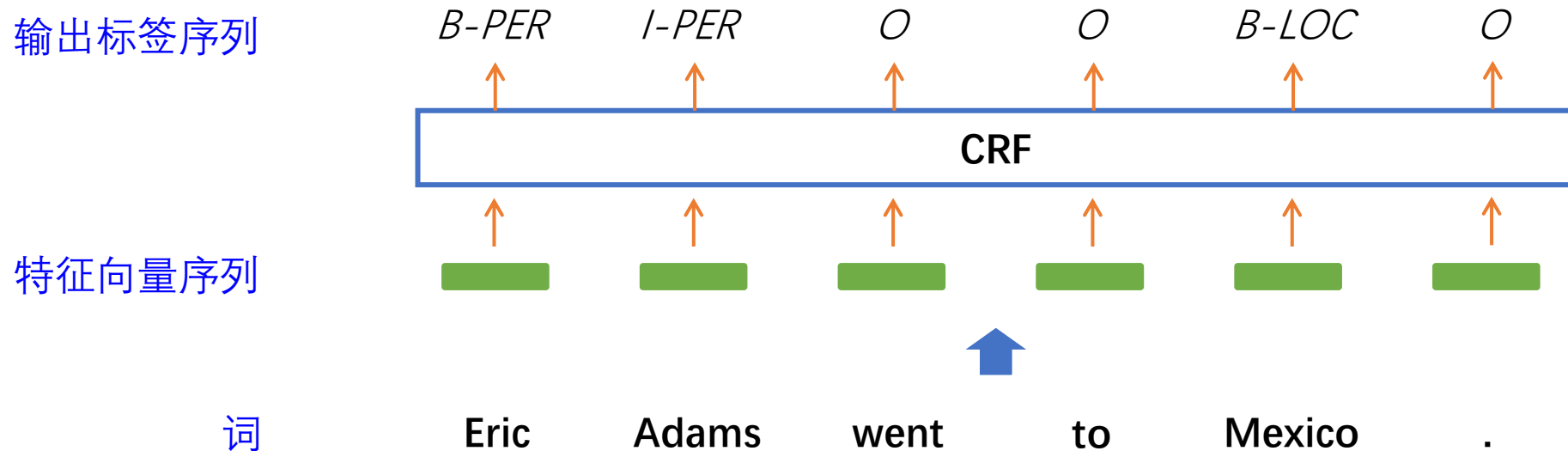
$t = 2$

$F(A, A, x_2; \theta)$	1
$F(B, A, x_2; \theta)$	3
$F(C, A, x_2; \theta)$	9
$F(A, B, x_2; \theta)$	2
$F(B, B, x_2; \theta)$	5
$F(C, B, x_2; \theta)$	2
$F(A, C, x_2; \theta)$	4
$F(B, C, x_2; \theta)$	2
$F(C, C, x_2; \theta)$	3



基于CRF的NER

- 实际应用：设计好词序列中每个词对应的特征向量；定义好要预测的类别



基于条件随机场（CRF）的NER

- CoNLL 2003数据集上的效果

English	Development			Test		
	Prec	Recall	F1	Prec	Recall	F1
LOC	93.82	91.78	92.79	87.23	87.65	87.44
MISC	83.99	78.52	81.17	74.44	71.37	72.87
ORG	84.23	82.03	83.11	79.52	78.33	78.92
PER	92.64	93.65	93.14	91.05	89.98	90.51
Overall	89.84	88.10	88.96	84.52	83.55	84.04

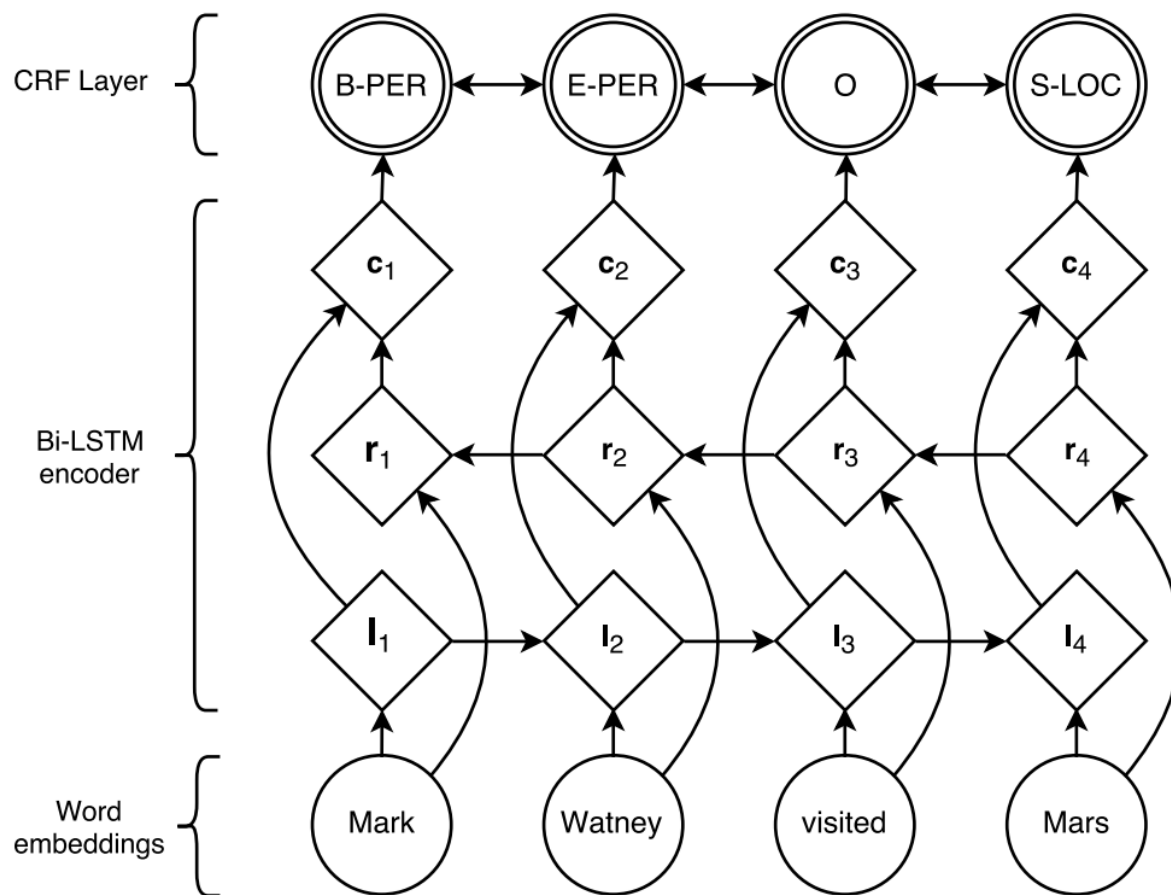
- Precision = 预测对的实体数 / 预测实体数
- Recall = 预测对的实体数 / 真实实体数
- $F1 = 2 * Precision * Recall / (Precision + Recall)$

基于传统机器学习的方法

- 手工设计特征的一些不足之处：
 - 手工设计特征需要专家知识
 - 泛化能力和鲁棒性难以得到保证
 - 不易捕捉一些语义相关信息

基于神经网络的方法

- LSTM-CRF模型



基于神经网络的方法

- 词向量

- 最基本的方法：one-hot编码

$$w^{aardvark} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, w^a = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, w^{at} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \dots w^{zebra} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

- 问题：只表示了是哪个词，没有词的特征（语法、语义）
 - 有些词可能出现次数非常少，甚至测试集可能出现训练集未出现过的词

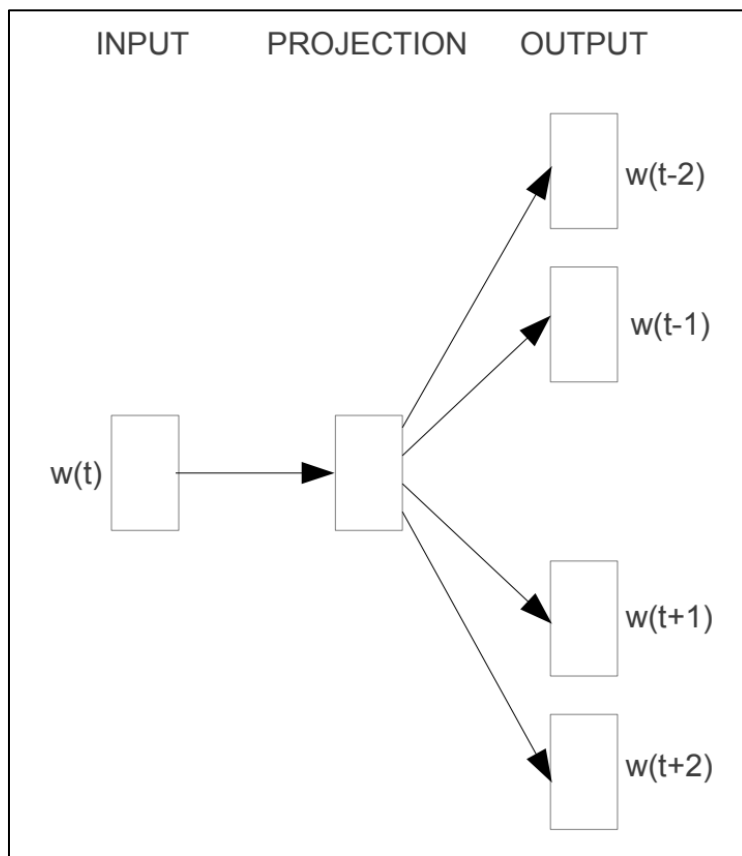
基于神经网络的方法

- 词向量
 - 将词表示为稠密 (dense) 的实向量
 - 维度一般小于词典大小 (如300维)
 - 如: word2vec, GloVe

word2vec

- 包含两种模型：Skip-gram和CBOW

Skip-gram:



word2vec为每个词 w 设置两个向量:

- Input vector v_w
- Output vector v'_w

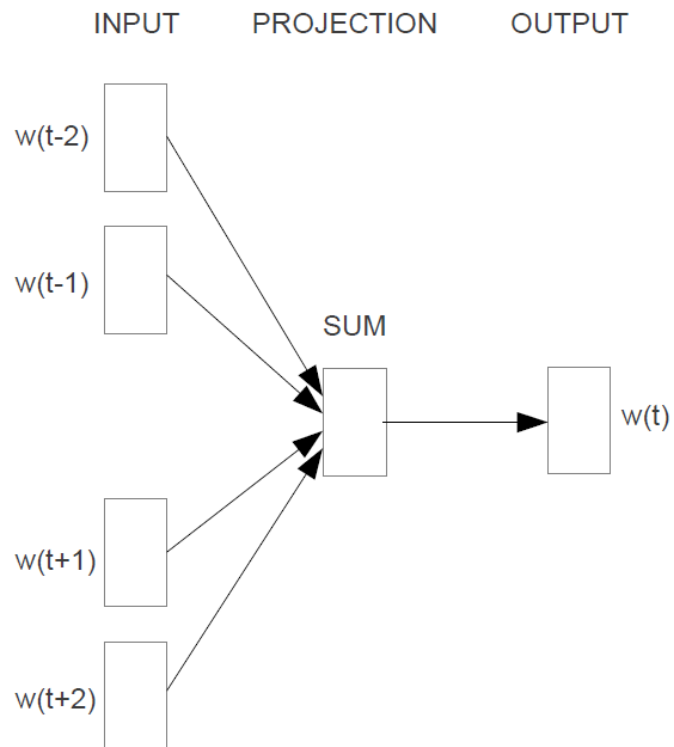
Skip-gram训练思路: 根据当前词预测它周围的词

$$p(w_O|w_I) = \frac{\exp(v'_{w_O} \top v_{w_I})}{\sum_{w=1}^W \exp(v'_w \top v_{w_I})}$$

word2vec

- 包含两种模型：Skip-gram和CBOW

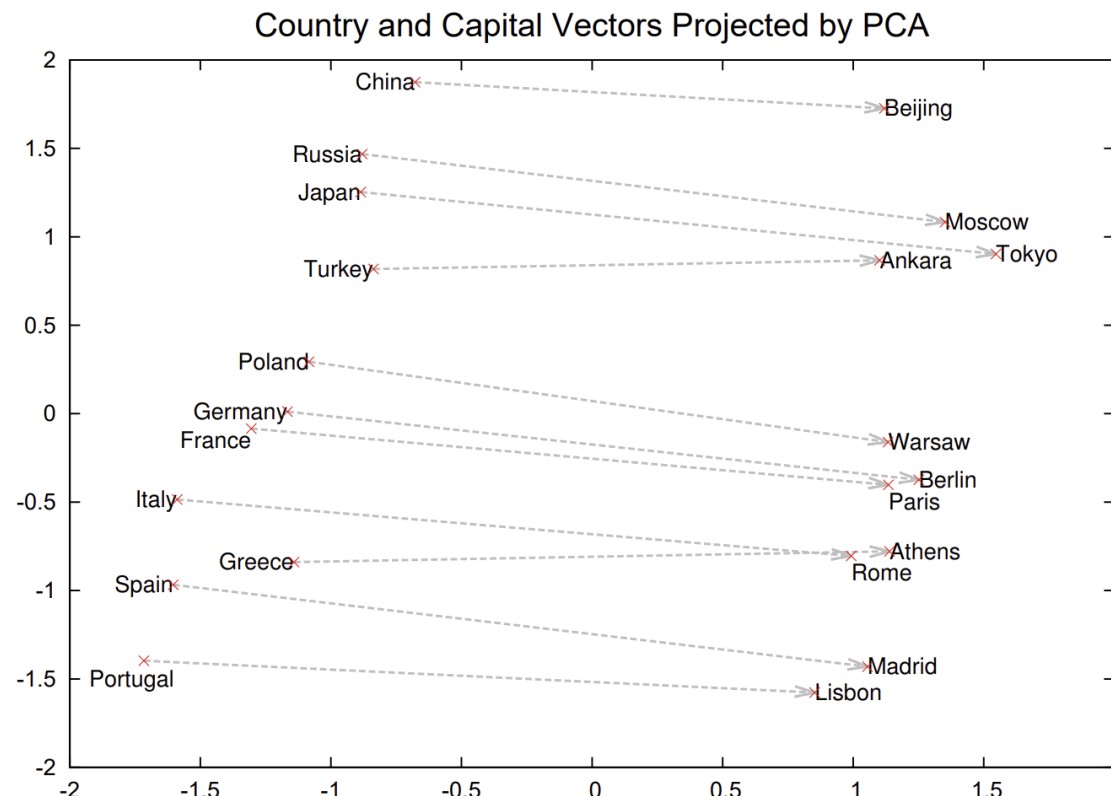
CBOW:



思路：根据周围词预测当前词

word2vec

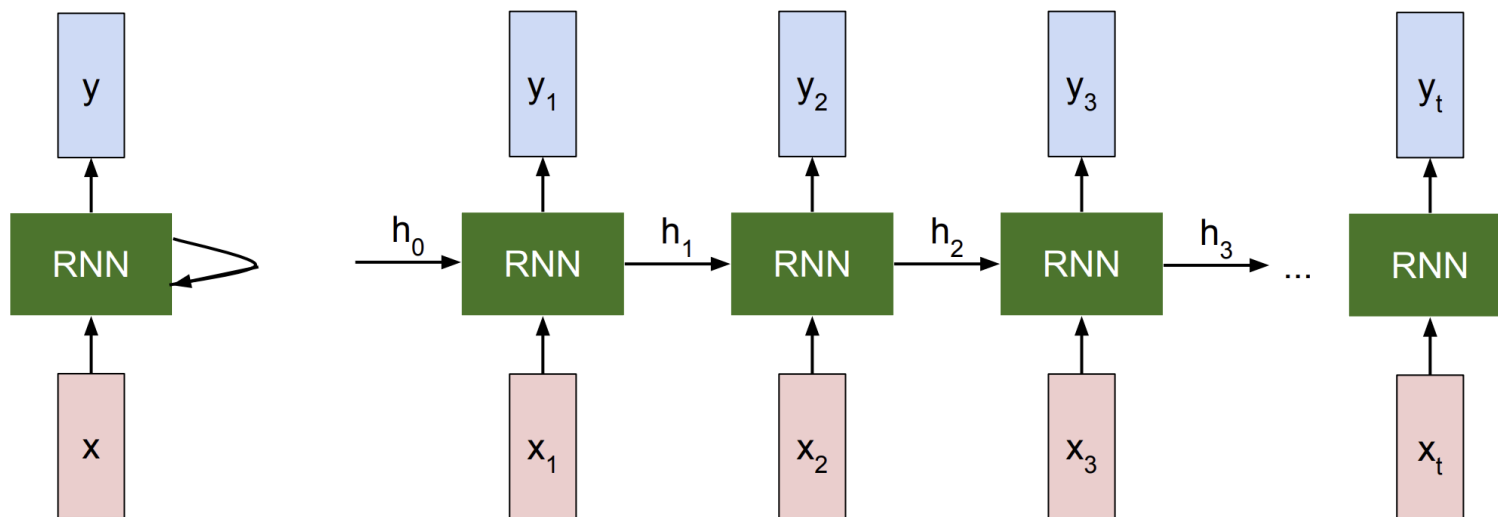
- word2vec的效果
 - 捕捉到了词的一些语法及语义特征



- 训练好的word vector可以用在为各种不同任务设计的不同模型中
 - 原作训练好的word2vec依然可下载: <https://code.google.com/archive/p/word2vec/>

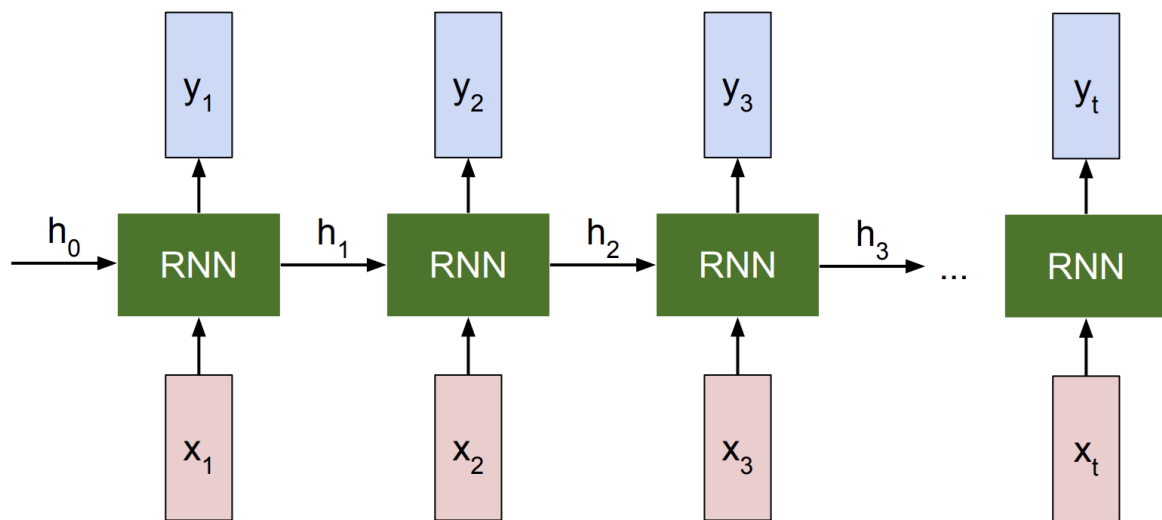
LSTM-CRF模型

- LSTM是一种RNN (Recurrent Neural Network) 模型
- RNN模型可以处理序列数据
 - 对序列中某处的输出会依赖于将模型本身之前的输出作为输入



LSTM-CRF模型

- LSTM是一种RNN (Recurrent Neural Network) 模型
- RNN模型可以处理序列数据
 - 对序列中某处的输出会依赖于将模型本身之前的输出作为输入



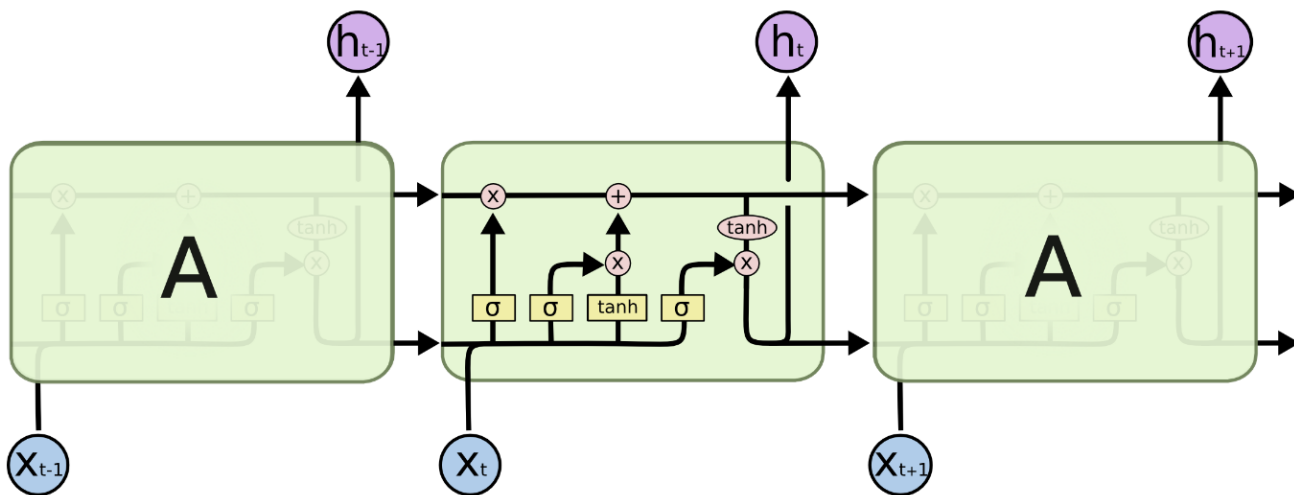
Elman network:

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h)$$

$$y_t = \sigma_y(W_y h_t + b_y)$$

LSTM-CRF模型

- LSTM模型
 - 缓解训练时的梯度消失问题
 - 梯度消失问题会导致难以学习到长距离依赖



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

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$$

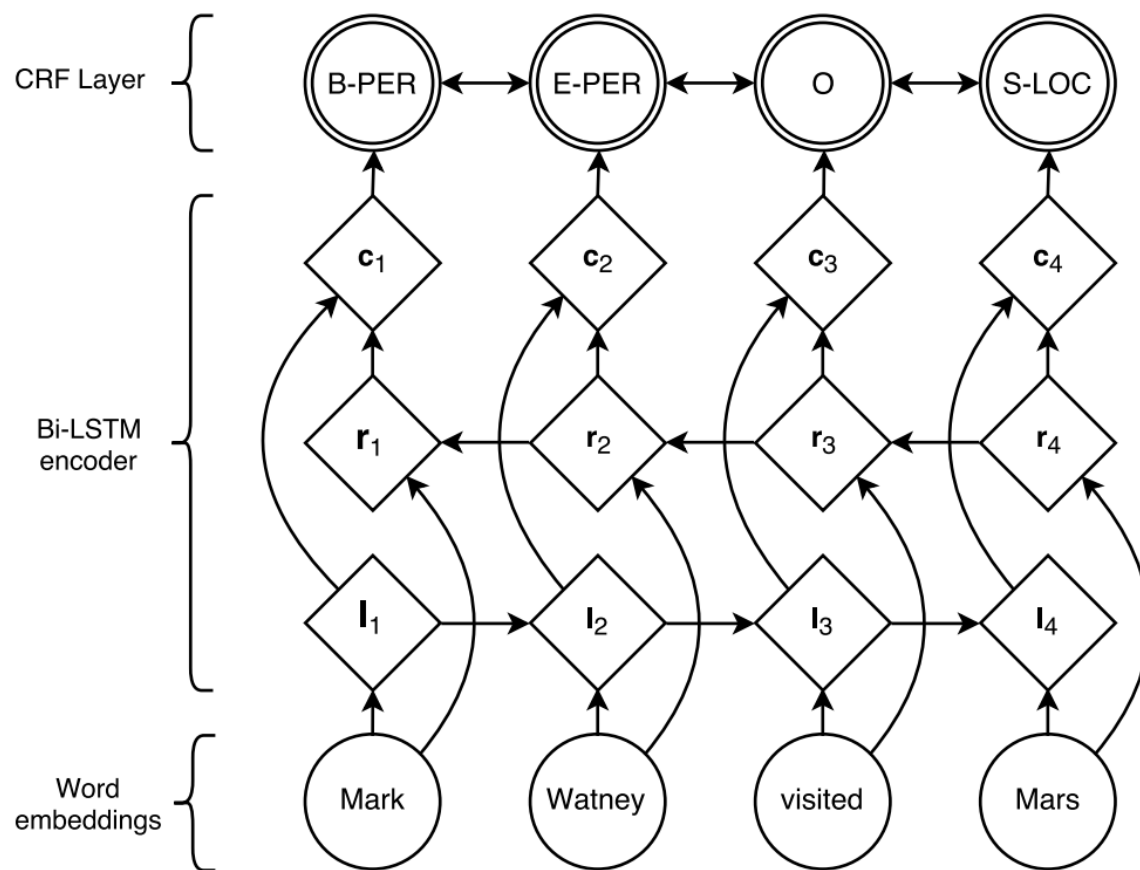
$$h_t = o_t * \tanh(C_t)$$

其中 C_t 向量叫做cell state，代表了模型记忆的信息

基于神经网络的方法

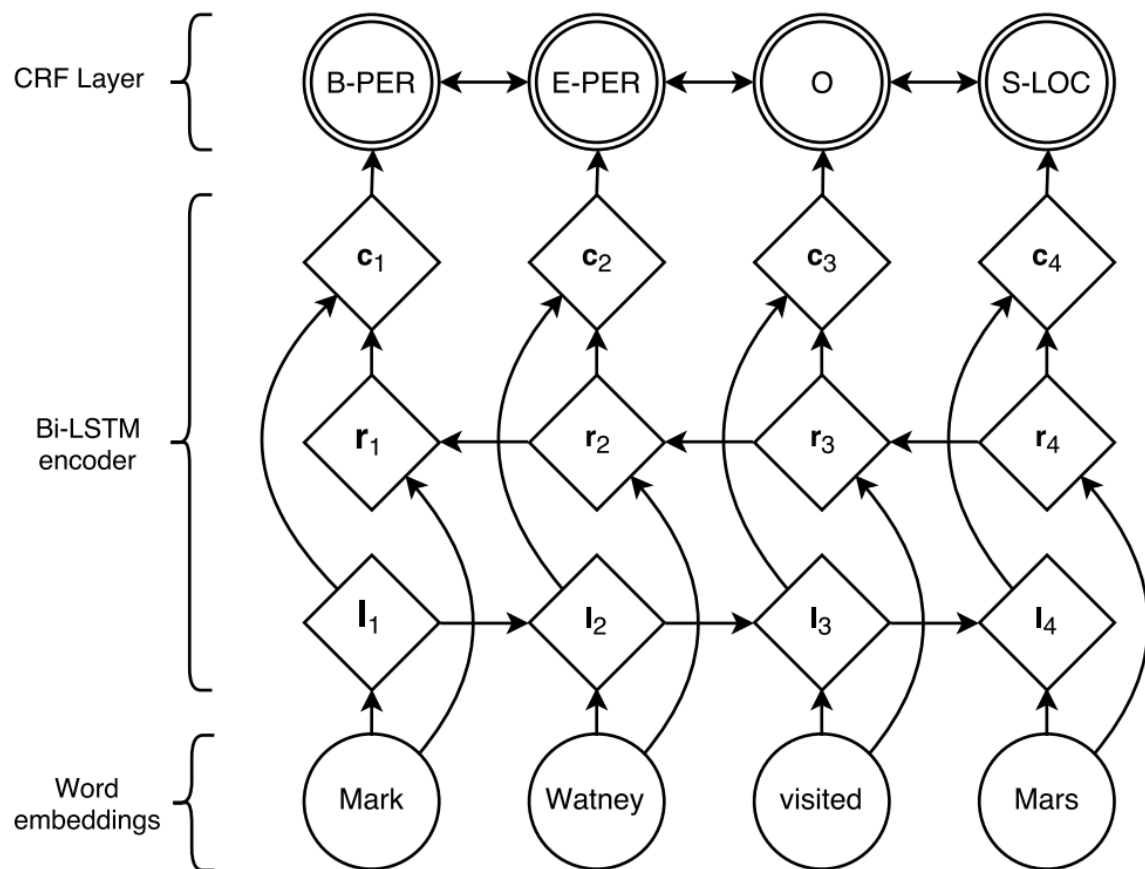
- LSTM-CRF模型

- Word embeddings: 每个词的向量表示
- Bi-LSTM: 为每个词得到一个与上下文相关的向量表示, 作为标签预测的特征
- CRF: 基于Bi-LSTM得到的向量序列预测标签序列

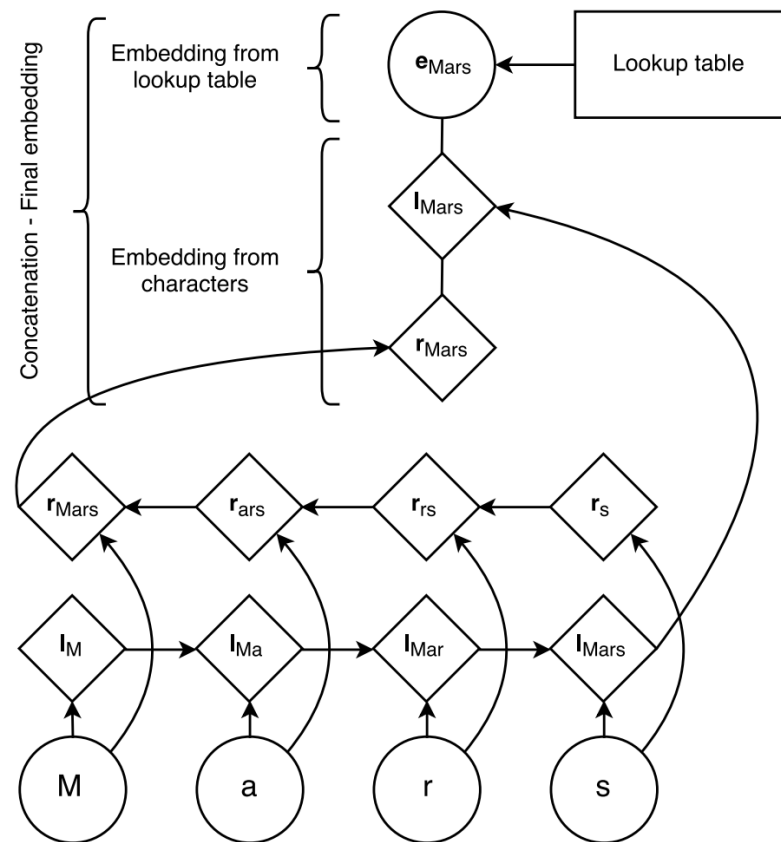


基于神经网络的方法

• LSTM-CRF模型



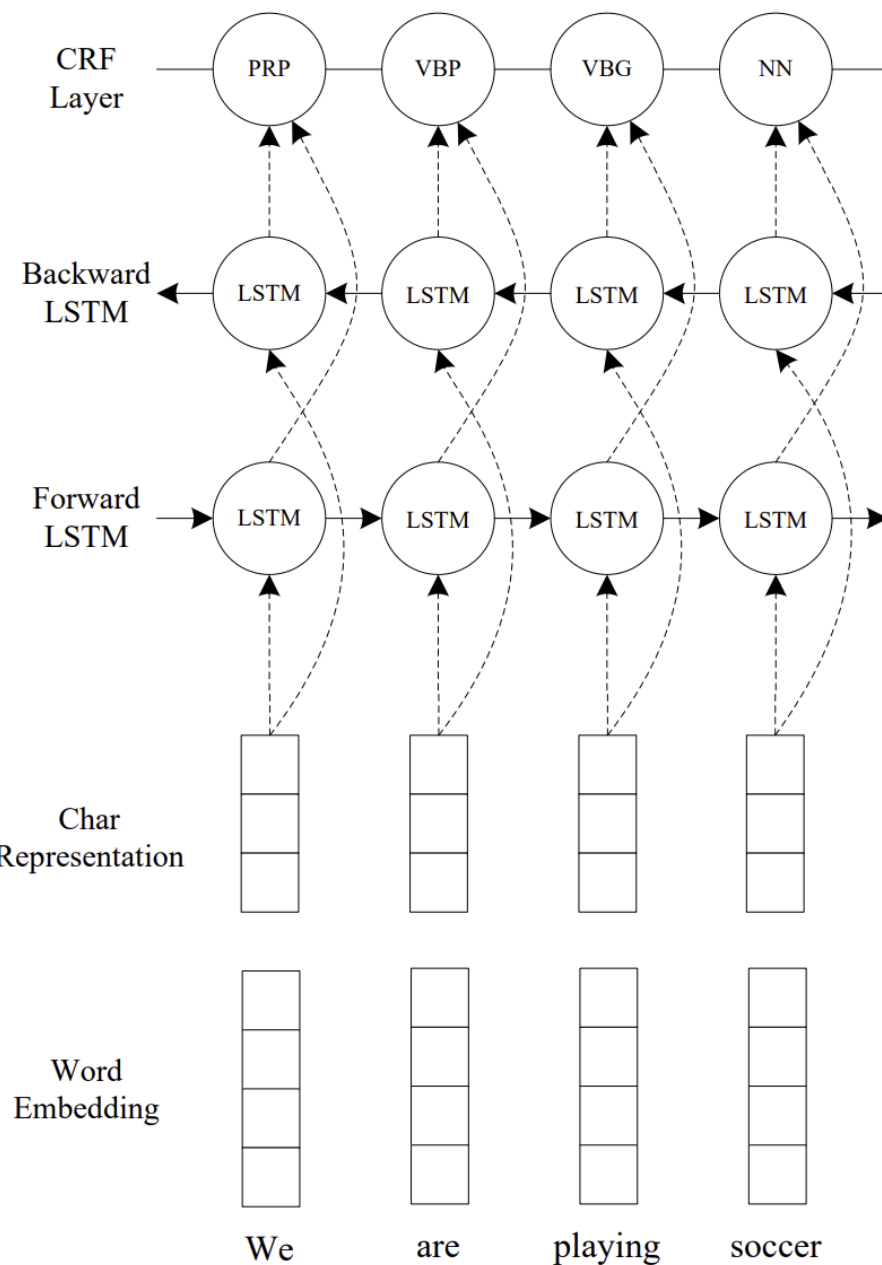
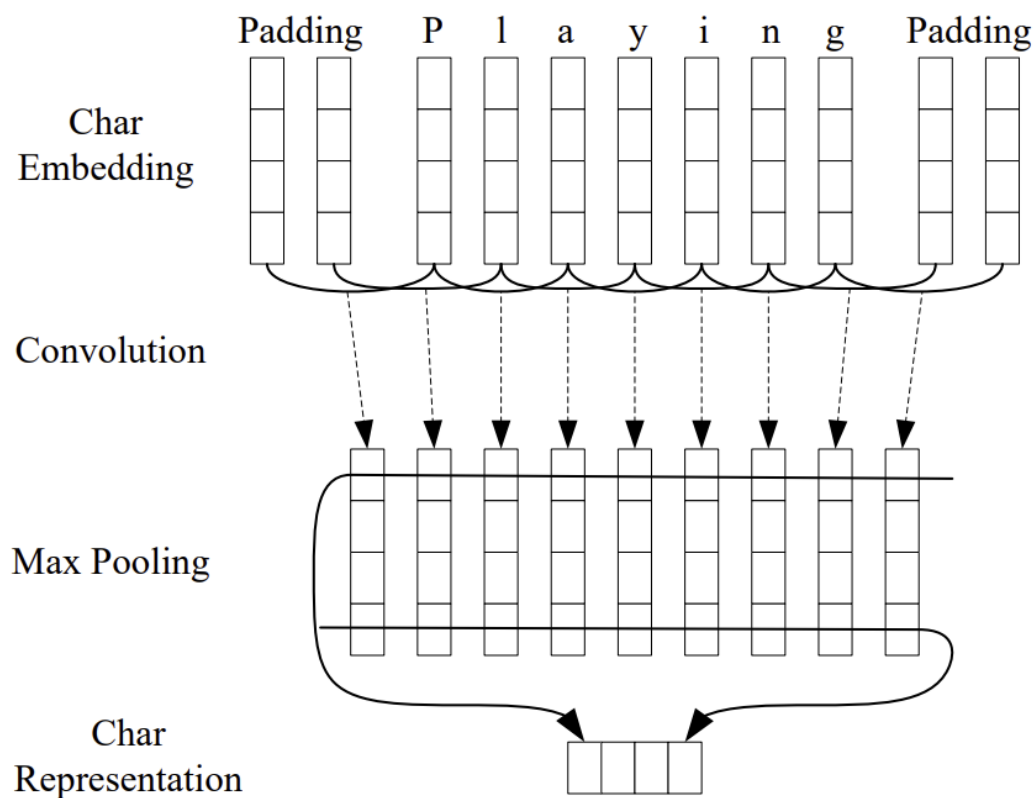
字符级的词向量表示获取:



使用随机初始化的字符向量

基于神经网络的方法

- CNN-LSTM-CRF (Ma and Hovy, 2016)



基于神经网络的方法

- CNN-LSTM-CRF (Ma and Hovy, 2016)
 - CoNLL 2003数据集上的效果

Model	F1
Chieu and Ng (2002)	88.31
Florian et al. (2003)	88.76
Ando and Zhang (2005)	89.31
Collobert et al. (2011) [‡]	89.59
Huang et al. (2015) [‡]	90.10
Chiu and Nichols (2015) [‡]	90.77
Ratinov and Roth (2009)	90.80
Lin and Wu (2009)	90.90
Passos et al. (2014)	90.90
Lample et al. (2016) [‡]	90.94
Luo et al. (2015)	91.20
This paper	91.21

基于迁移学习的NER

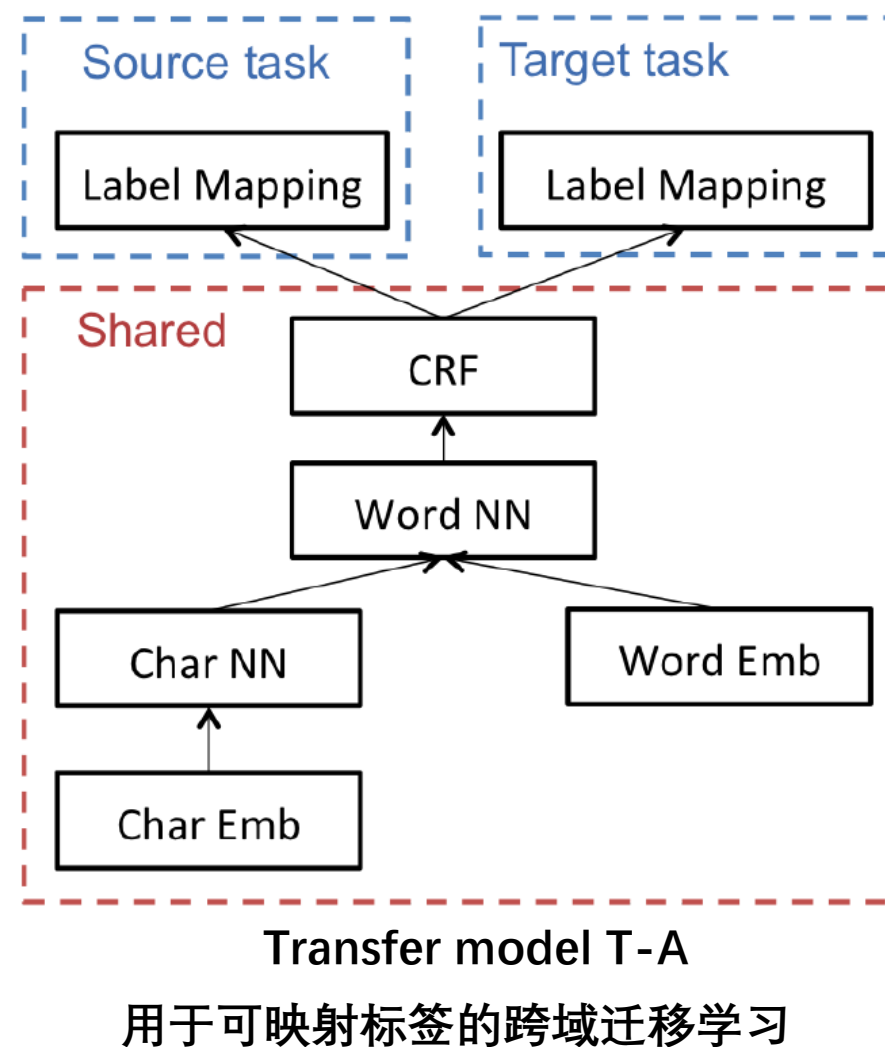
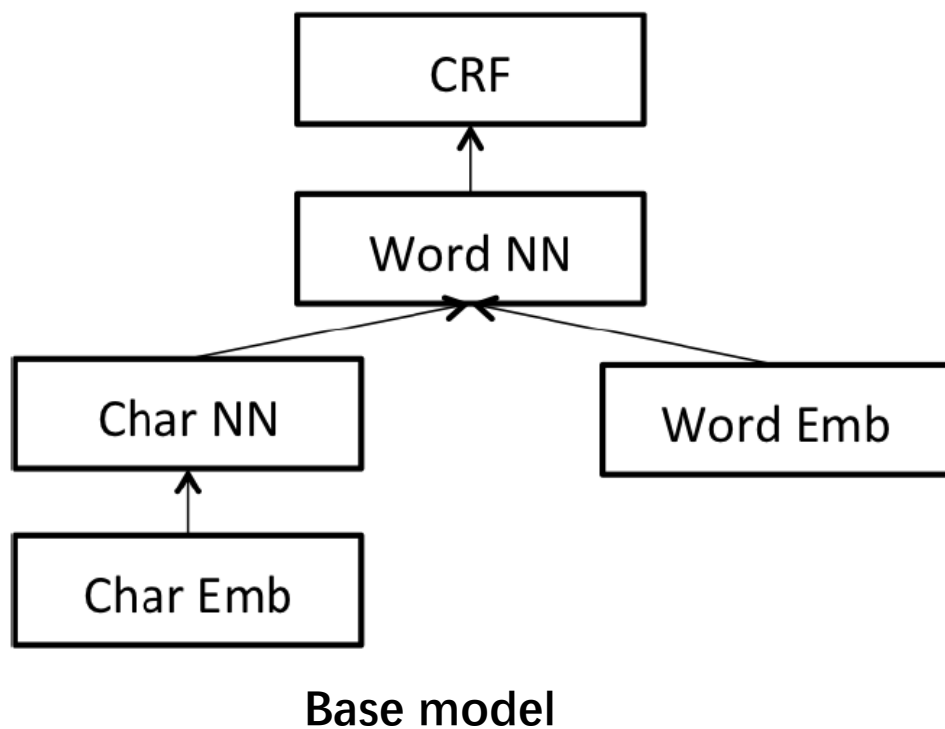
- 迁移学习（Transfer Learning）是属于机器学习的一个子研究领域，该研究领域的目标在于利用数据、任务、或模型之间的相似性，将在旧领域学习过的知识，迁移应用于新领域中。
- 适合在目标任务训练数据量较少的情况下使用

基于迁移学习的NER

- 以 (Yang et al., 2016)为例，考虑三种迁移学习模式：
 - 跨域 (Cross-Domain)
 - 应用相同，文本风格或领域不同，如从新闻文本到微博文本
 - 跨应用 (Cross-Application)
 - 如从词性标注 (POS tagging) 到NER
 - 跨语言 (Cross-Lingual)
 - 如从西班牙语到英语

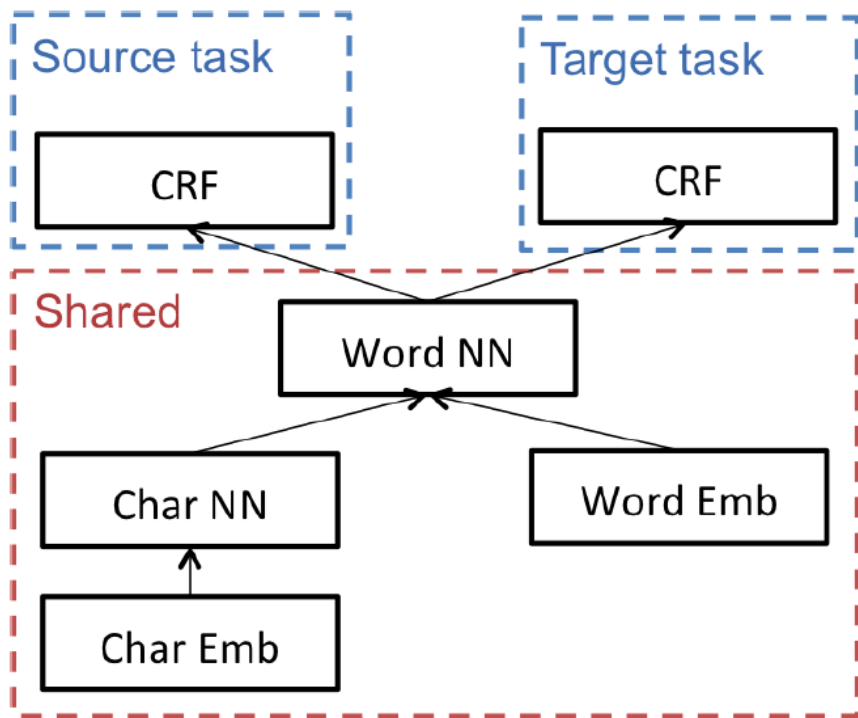
基于迁移学习的NER

- (Yang et al., 2017)中的模型



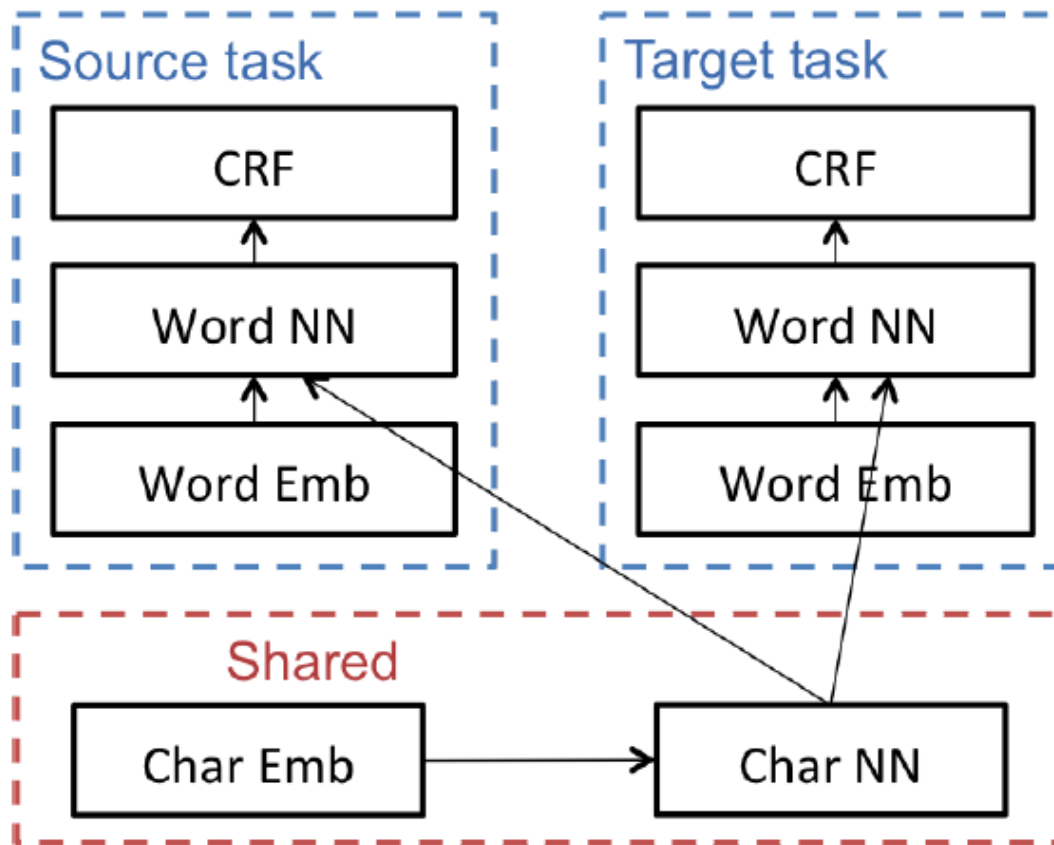
基于迁移学习的NER

- (Yang et al., 2017)中的模型



Transfer model T-B

用于无法做标签映射的跨域迁移学习，
以及跨应用迁移学习



Transfer model T-C
用于跨语言迁移学习

基于迁移学习的NER

- (Yang et al., 2017)的模型训练
 - 每次采样一个任务 (source task或target task)
 - 采样该任务一个训练样本batch
 - 更新参数
 - 该任务独有的参数
 - 两个任务共享的参数

基于迁移学习的NER

- (Yang et al., 2017)中的迁移学习效果

Source	Target	Model	Setting	Transfer	No Transfer	Delta
PTB	Twitter/0.1	T-A	dom	83.65	74.80	8.85
CoNLL03	Twitter/0.1	T-A	dom	43.24	34.65	8.59
PTB	CoNLL03/0.01	T-B	app	74.92	68.64	6.28
PTB	CoNLL00/0.01	T-B	app	86.73	83.49	3.24
CoNLL03	PTB/0.001	T-B	app	87.47	84.16	3.31
Spanish	CoNLL03/0.01	T-C	ling	72.61	68.64	3.97
CoNLL03	Spanish/0.01	T-C	ling	60.43	59.84	0.59
PTB	Genia/0.001	T-A	dom	92.62	83.26	9.36
CoNLL03	Genia/0.001	T-B	dom&app	87.47	83.26	4.21
Spanish	Genia/0.001	T-C	dom&app&ling	84.39	83.26	1.13
PTB	Genia/0.001	T-B	dom	89.77	83.26	6.51
PTB	Genia/0.001	T-C	dom	84.65	83.26	1.39

基于迁移学习的NER

- (Yang et al., 2017)中的迁移学习效果
- 使用完整数据

Table 3: Comparison with state-of-the-art results (%).

Model	CoNLL 2000	CoNLL 2003	Spanish	Dutch	PTB 2003
Collobert et al. (2011)	94.32	89.59	—	—	97.29
Passos et al. (2014)	—	90.90	—	—	—
Luo et al. (2015)	—	91.2	—	—	—
Huang et al. (2015)	94.46	90.10	—	—	97.55
Gillick et al. (2015)	—	86.50	82.95	82.84	—
Ling et al. (2015)	—	—	—	—	97.78
Lample et al. (2016)	—	90.94	85.75	81.74	—
Ma & Hovy (2016)	—	91.21	—	—	97.55
Ours w/o transfer	94.66	91.20	84.69	85.00	97.55
Ours w/ transfer	95.41	91.26	85.77	85.19	97.55

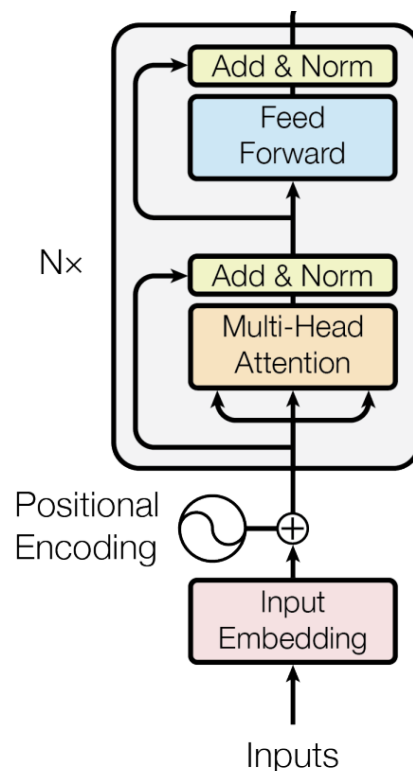
基于预训练模型的NER

- 预训练语言模型（Pre-trained language models, PLMs）是用大规模语料以自监督方式训练得到的语言模型。
 - 如：BERT、RoBERTa、T5、GPT-3等

基于预训练模型的NER

- BERT
 - 其模型结构基于Transformer Encoder
 - 采用Masked language model (MLM) 和Next sentence prediction (NSP) 两项任务预训练

Transformer Encoder:

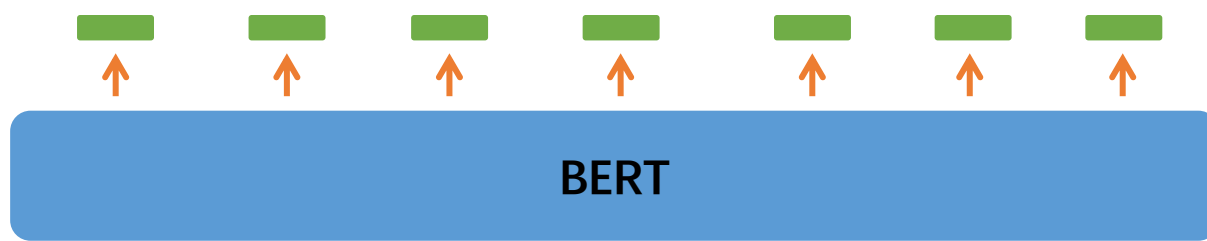


基于预训练模型的NER - BERT

- BERT先对文本进行tokenization，将文本变成token序列
- 每个token对应一个embedding向量，向量序列成为BERT模型的输入
- BERT模型最终为每个token输出一个向量表示

例：“He is eating.” tokenization后得到 “He”, “is”, “eat”, “##ing”, “.” 五个token，在它们前后分别加上[CLS]、[SEP]两个特殊token后形成BERT的输入token序列

向量表示序列：



输入Token序列：

[CLS] He is eat ##ing . [SEP]

基于预训练模型的NER - BERT

- 对于一个token t , 设BERT为其得到的向量为 \mathbf{h}_t
- 在 \mathbf{h}_t 基础上加上一个分类头, 即可实现对 t 按BIO标注体系分类

如:
$$p(y_t = \hat{y}) = \text{softmax}(\mathbf{W}\mathbf{h}_t + \mathbf{b})_{\hat{y}}$$

其中 $\hat{y} \in \{O, B-PER, I-PER, B-LOC, I-LOC, \dots\}$

基于预训练模型的NER - BERT

- 基于BERT的NER效果

System	Dev F1	Test F1
ELMo (Peters et al., 2018a)	95.7	92.2
CVT (Clark et al., 2018)	-	92.6
CSE (Akbik et al., 2018)	-	93.1
Fine-tuning approach		
BERT _{LARGE}	96.6	92.8
BERT _{BASE}	96.4	92.4

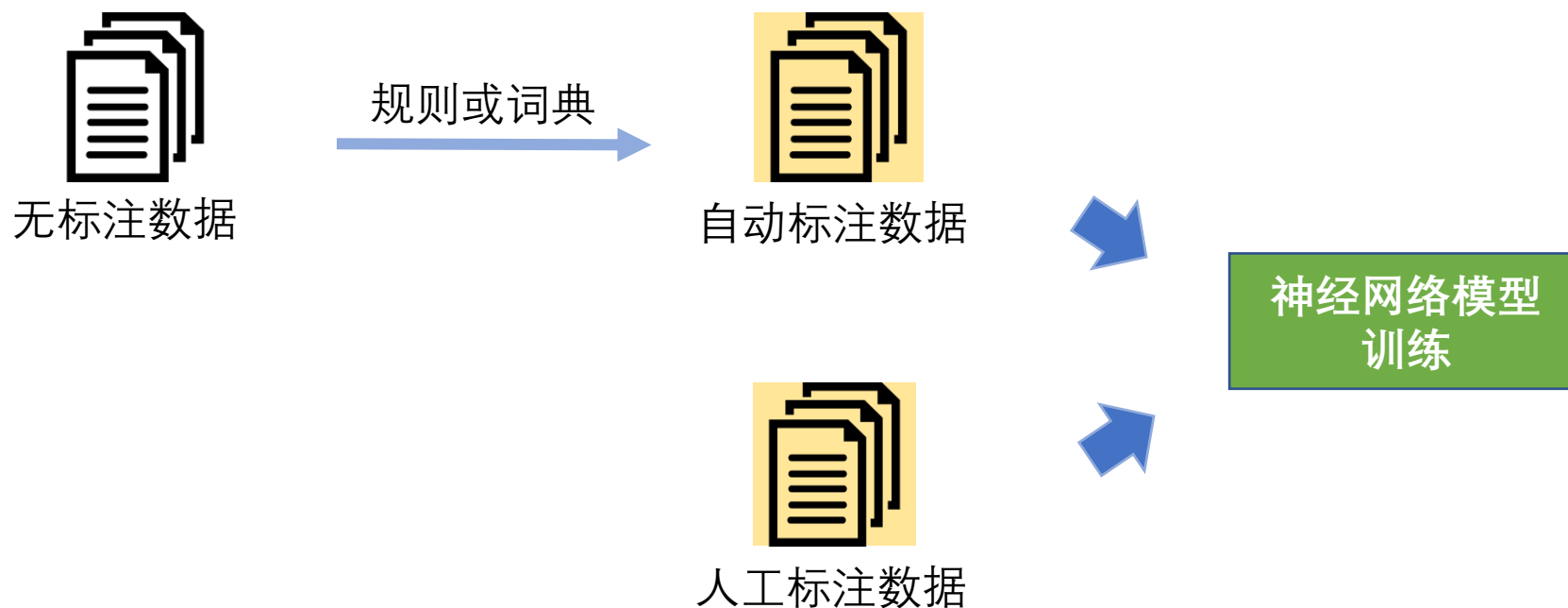
CoNLL-2003 Named Entity Recognition results

基于预训练模型 vs 直接基于LSTM

- 基于预训练模型 (PLM, 如BERT)
 - PLM的参数是预训练过的, 在未使用NER数据fine-tune前, 通过PLM获得的token向量表示就已经包含了有意义的特征, 且包含的语义特征是上下文相关的
- 直接基于LSTM
 - LSTM中的参数是随机初始化的, 在未使用NER数据训练前, 通过LSTM获得的词向量表示尚未包含有意义的特征

结合规则与神经网络模型的方法

- 使用规则或词典自动标注大量数据后，结合人工标注数据训练神经网络模型
 - 一般可先用自动标注数据训练模型，再用人工标注数据继续训练
 - 适合人工标注数据量少的情况



基于大模型的方法

- Zero-shot设定

Given entity label set: ['Person', 'Organization', 'Location', 'Facility', 'Weapon', 'Vehicle', 'Geo-Political Entity'].

Please recognize the named entities in the given text. Based on the given entity label set, provide answer in the following JSON format: [{ 'Entity Name': 'Entity Label' }]. If there is no entity in the text, return the following empty list: [].

Text: New York City mayor Eric Adams went to Mexico.

Answer:

ChatGPT

HL

You

Given entity label set: ['Person', 'Organization', 'Location', 'Facility', 'Weapon', 'Vehicle', 'Geo-Political Entity'].

Please recognize the named entities in the given text. Based on the given entity label set, provide answer in the following JSON format: [{ 'Entity Name': 'Entity Label' }]. If there is no entity in the text, return the following empty list: [].


Text: New York City mayor Eric Adams went to Mexico.

Answer:



ChatGPT

json

 Copy code

```
[{"New York City": "Location"}, {"Eric Adams": "Person"}, {"Mexico":
```

基于大模型的方法

- Few-shot设定; 使用In-context learning (ICL)
- 即给大模型提供一些样例

Given entity label set: ['Person', 'Organization', 'Location', 'Facility', 'Weapon', 'Vehicle', 'Geo-Political Entity'].
Please recognize the named entities in the given text. Based on the given entity label set, provide answer in the following JSON format: [{'Entity Name': 'Entity Label'}]. If there is no entity in the text, return the following empty list: [].

Text: right now we 're also waiting to hear from the president at the white house .

Answer: [{'white house': 'Location'}, {'president': 'Person'}]

Text: At the Pentagon , Barbara Starr reports officials say today begins a new strategy in the skies over Baghdad .

Answer: [{'Barbara Starr': 'Person'}, {'Pentagon': 'Facility'}, {'officials': 'Person'}, {'skies': 'Location'}, {'Baghdad': 'Geo-Political Entity'}]

Text: John Irvine , ITV News , Baghdad .

Answer: [{'John Irvine': 'Person'}, {'ITV News': 'Organization'}, {'Baghdad': 'Geo-Political Entity'}]

... ..

Text: New York City mayor Eric Adams went to Mexico.

Answer:

基于大模型的方法

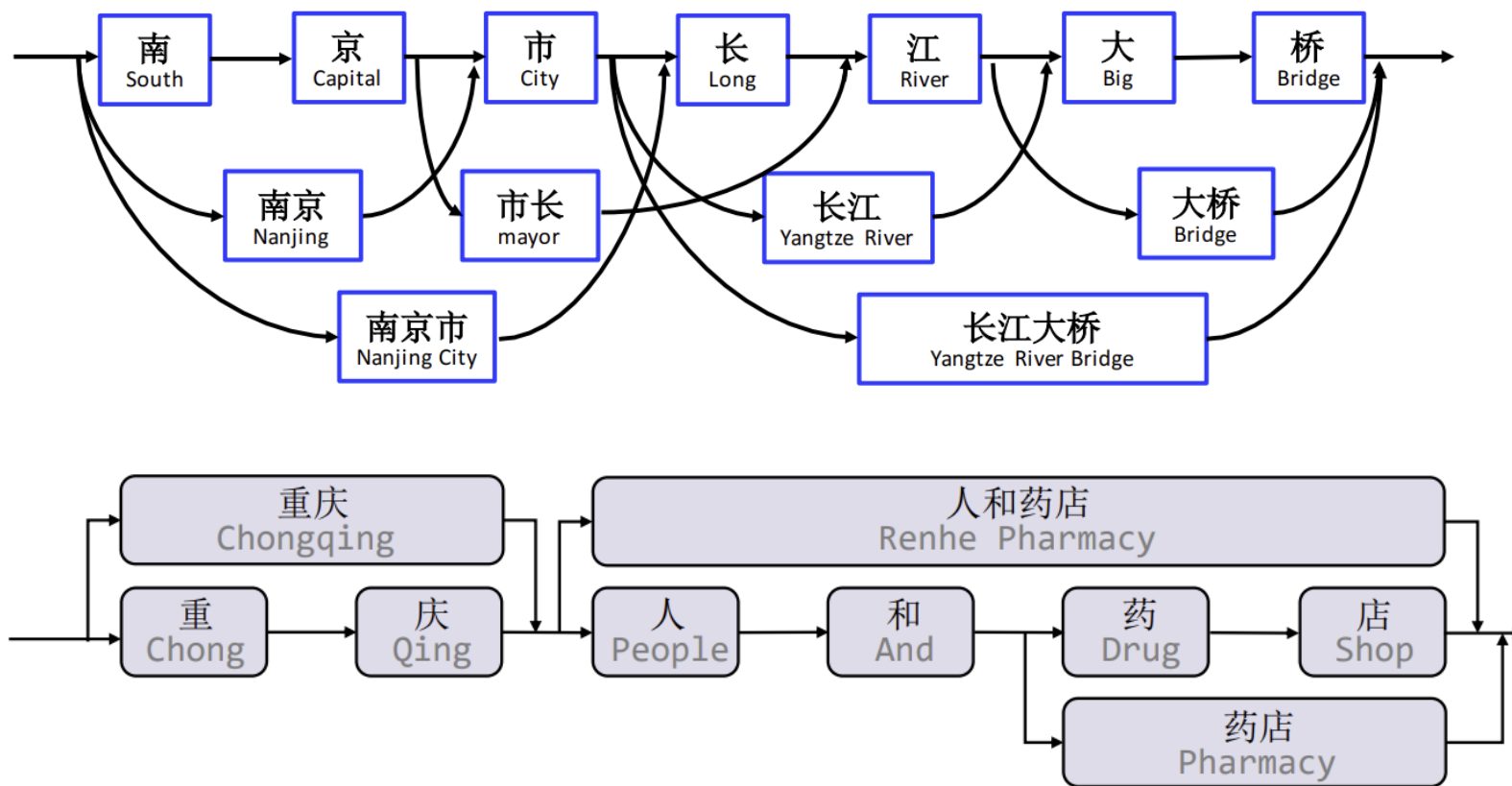
- 使用GPT-4在CoNLL 2003数据集上F1可达0.8+

中文NER方法

- 基于Lattice的方法
- 处理中文时的一个难点：
 - 如直接给模型提供字级别的输入，可能导致模型学习难度大
 - 如分词后再提供词级别的输入，又可能因分词错误导致error propagation
- 应对方法：使用Lattice将有可能（潜在）的词都提供给模型

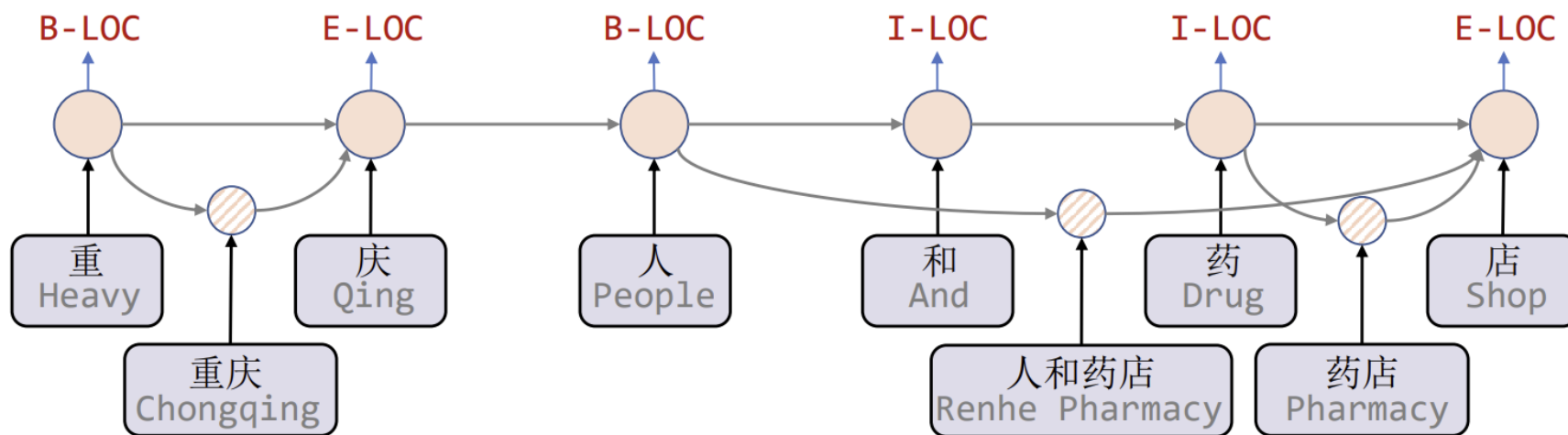
基于Lattice的中文NER

Lattice例:



基于Lattice的中文NER

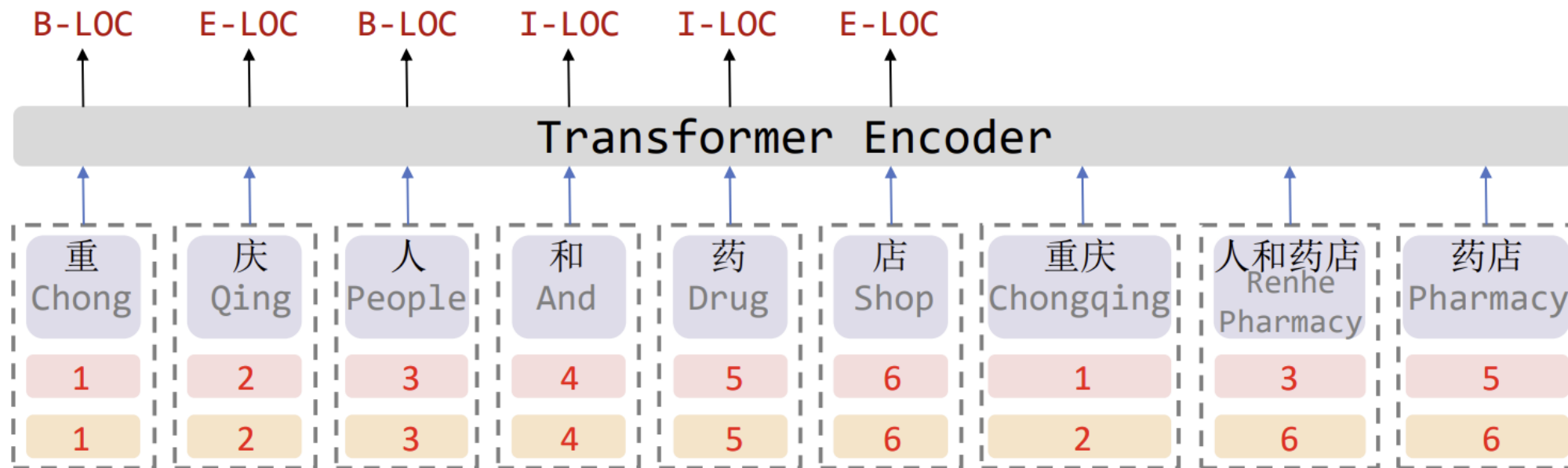
- Lattice LSTM



基于Lattice的中文NER

- Flat-Lattice Transformer

- 将所有的词也输入Transformer，用起始和结束位置的encoding提供它们在原文本中的位置信息
- 预测时，仅为字符序列预测标签



嵌套命名实体识别 (Nested NER)

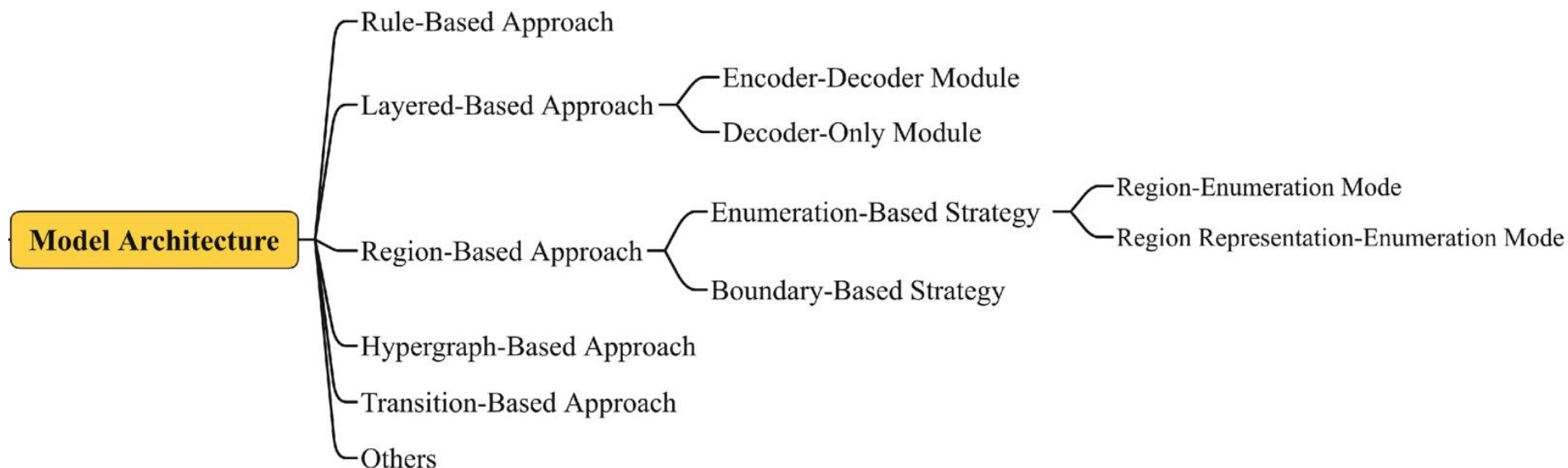
- 考虑嵌套在其他实体中的实体

LOC

南京航空航天大学食堂真不错

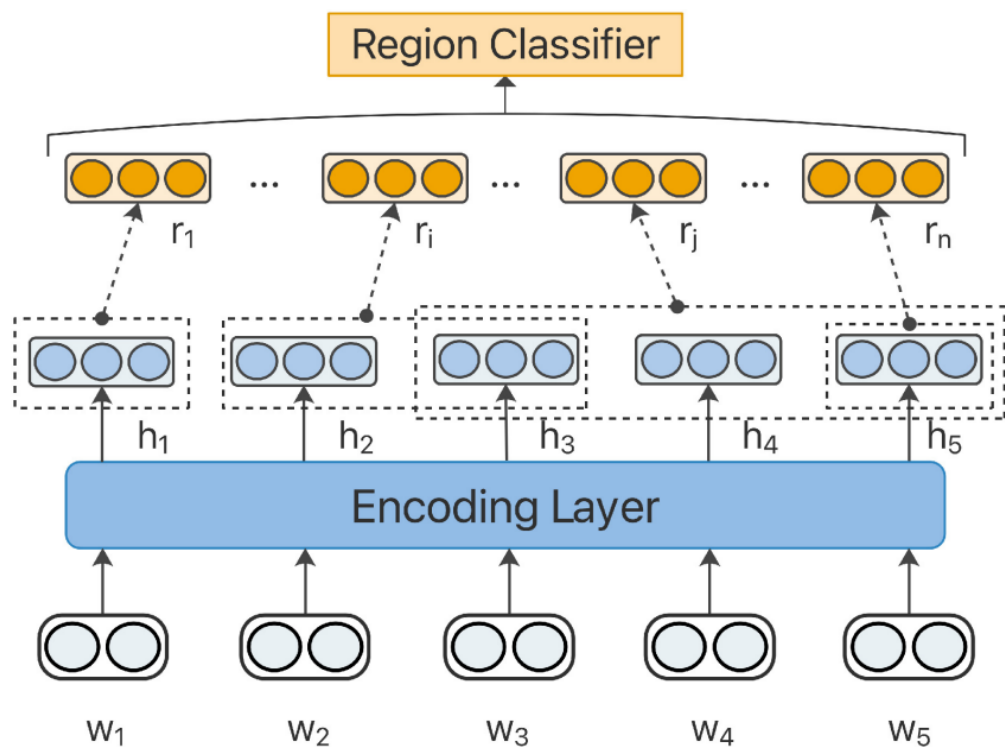
ORG

- 使用序列标注的方法难以应对嵌套实体



嵌套命名实体识别 (Nested NER)

- Region-based / Span-based方法
- 基本思路：考虑所有可能的span，看每个span是否是实体
 - 一般会对span长度进行限制



如，将起始位置为 i ，结束位置为 j 的span的类别分数

$$r_{ij} = W(h_i \oplus h_j) + b$$

其中 $h_i, h_j \in R^d$ 分别为第 i, j 个词对应的模型输出向量表示
 $W \in R^{c \times 2d}$ 是可训练矩阵
 c 是实体类别标签（含NA，即表示不是实体）数

从文本中抽取知识

- 命名实体识别
- **细粒度实体分类**
- 关系抽取
- 事件抽取