

3.3 序列标注

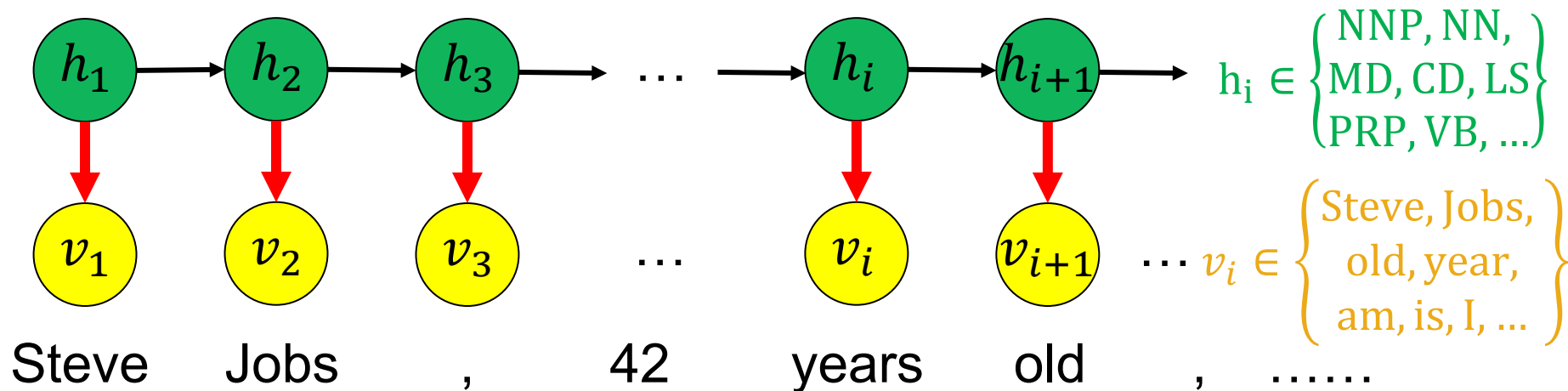
林洲汉

上海交大计算机系

- ▶ **序列标注 (Sequence Tagging)** 是NLP中最基础的任务之一，应用十分广泛，如：
 - ▶ 分词
 - ▶ 词性标注 (POS tagging)
 - ▶ 命名实体识别 (Named Entity Recognition, NER)
 - ▶ 关键词抽取
 - ▶ 语义角色标注 (Semantic Role Labeling)
 - ▶ 槽位抽取 (Slot Filling)
 - ▶

- ▶ 输入序列 $\mathbf{X}=\{x_1, x_2, x_3, \dots, x_n\}$, 输出标注序列 $\mathbf{Y}=\{y_1, y_2, y_3, \dots, y_n\}$, 其中 $|y|=m$ (y 的种类数一共有 m 类)
- ▶ Y一般标注法(以NER为例)为:
 - ▶ B (begin): signifies beginning of an NE
 - ▶ I (inside): signifies that the word is inside an NE
 - ▶ E (end): signifies that the word is the end of an NE
 - ▶ S (singleton): signifies that the single word is an NE
 - ▶ O (outside): signifies that the word is just a regular word outside of an NE

序列标注：隐马尔科夫模型 (HMM)



$$\operatorname{argmax}_h P(h | v) = \operatorname{argmax}_h P(h, v) = \operatorname{argmax}_h P(v | h) \cdot P(h)$$

$$= \operatorname{argmax}_h \left(\prod_{j=1}^N P(v_j | h_j) P(h_1) \prod_{i=1}^{N-1} P(h_{i+1} | h_i) \right)$$

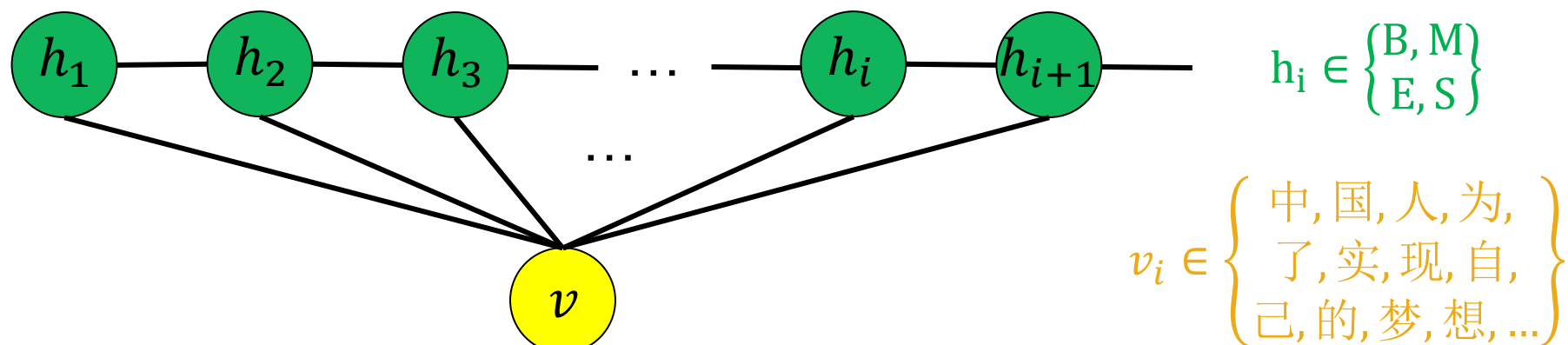
参数:

	am	is	are	a	an	the	old	help	price	sugar
NNP	■	■	■	■	■	■	■	■	■	■
NN	■	■	■	■	■	■	■	■	■	■
MD	■	■	■	■	■	■	■	■	■	■
PRP	■	■	■	■	■	■	■	■	■	■
VB	■	■	■	■	■	■	■	■	■	■

NNP	■	■	■	■	■	■	■	■	■	■
NN	■	■	■	■	■	■	■	■	■	■
MD	■	■	■	■	■	■	■	■	■	■
PRP	■	■	■	■	■	■	■	■	■	■
VB	■	■	■	■	■	■	■	■	■	■

...

序列标注：条件随机场 (CRF)



$$P(h|v) = \frac{\text{score}(h, v)}{\sum_h \text{score}(h, v)} \quad (\text{s.t.} \quad \text{score}(h, v) \geq 0)$$

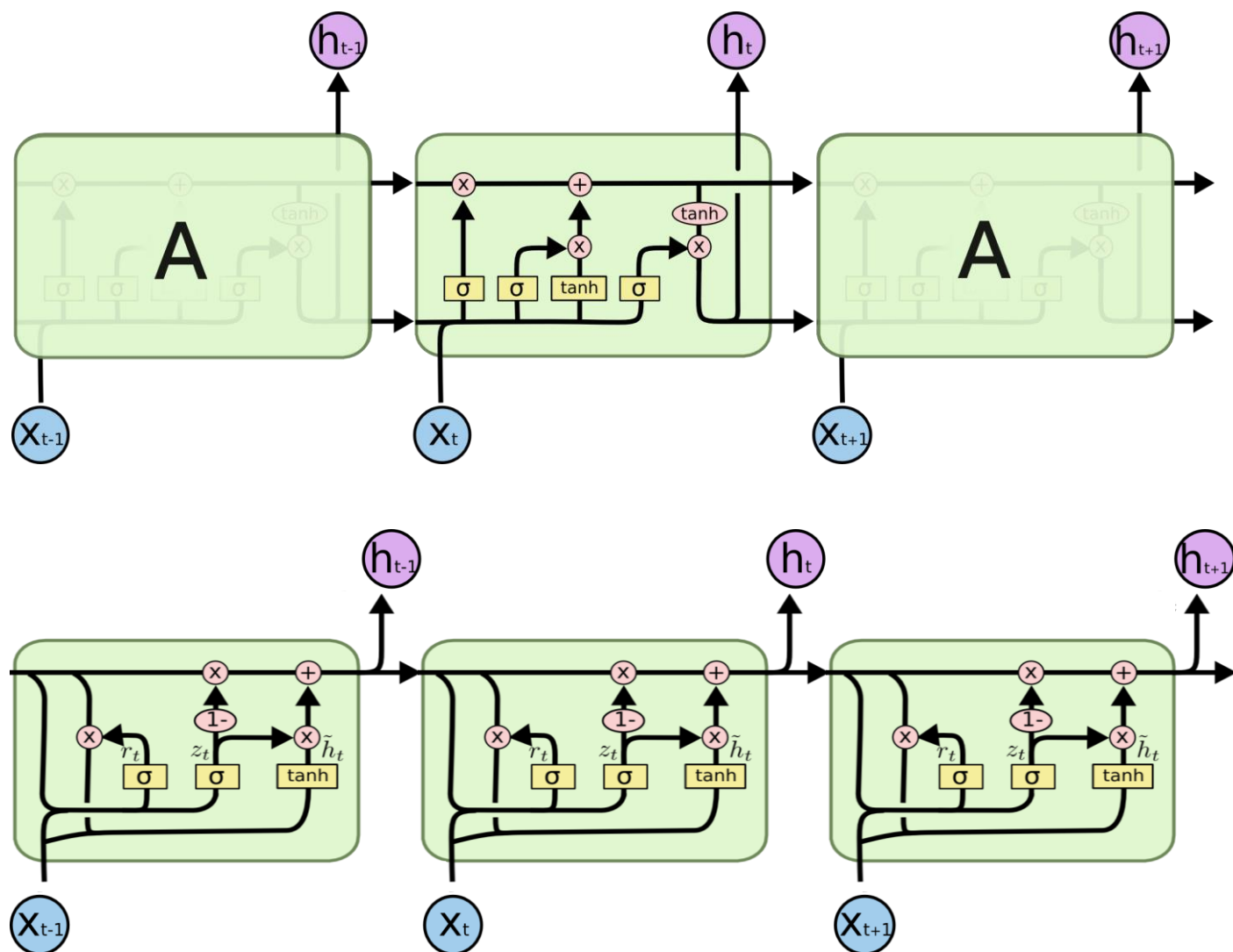
$$\text{score}(h, v) = \exp \left(\sum_{k=1}^K w_k F_k(h, v) \right)$$

$$F_k(h, v) = F_k(h_1, h_2, \dots, h_i, \dots, h_N, v_1, v_2, \dots, v_i, \dots, v_N)$$

$$= \sum_{i=2}^N \left[f_k(h_i, h_{i-1}, v, i) \right]$$

与HMM不同，CRF中允许用户指定具体feature。这使得CRF可以引入各种各样的基于规则的feature

序列标注：LSTM与GRU用于序列标注



序列标注：LSTM与GRU用于序列标注

- **单向模型**

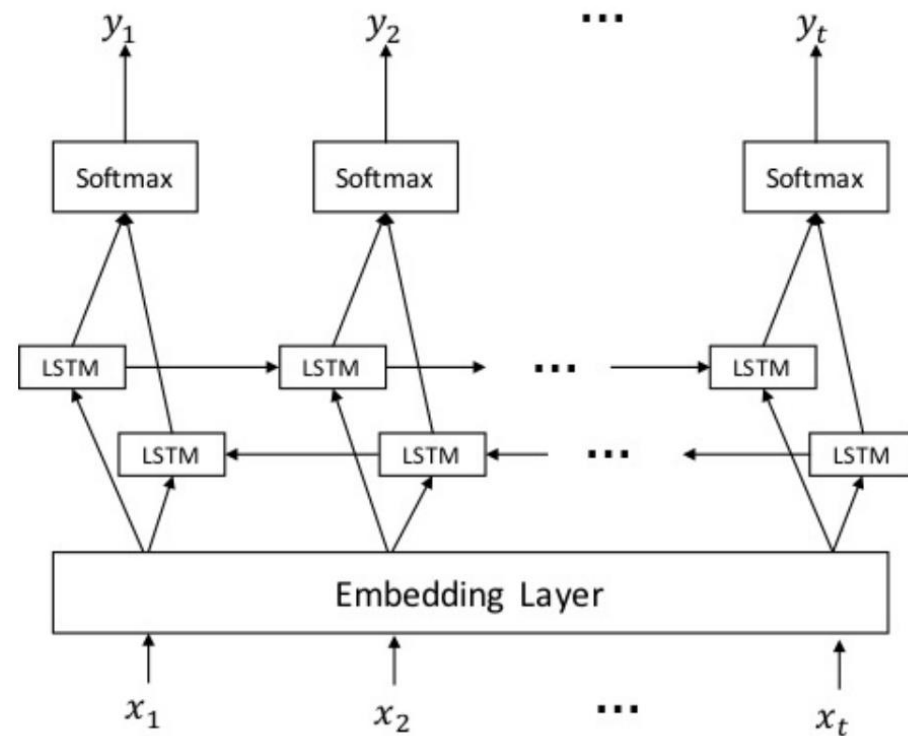
- 前面我们展示了使用LSTM和GRU进行序列标注的模型。模型自左向右读取文本，并且在每一步都生成一个隐层表征 h 。这使得模型可以以“流”的方式读取文本，并且边读取边生成标注。
- 这样的模型在运行时可以实现得**具有较强的实时性**。在语音识别、实时翻译等场景下较为适用。
- 然而这种形式的模型架构使得模型**无法根据后文来反过来修改前文的标注**。比如对这个句子做POS tagging：

I have got a raspberry pi device .

序列标注：LSTM与GRU的变种架构

• 双向模型

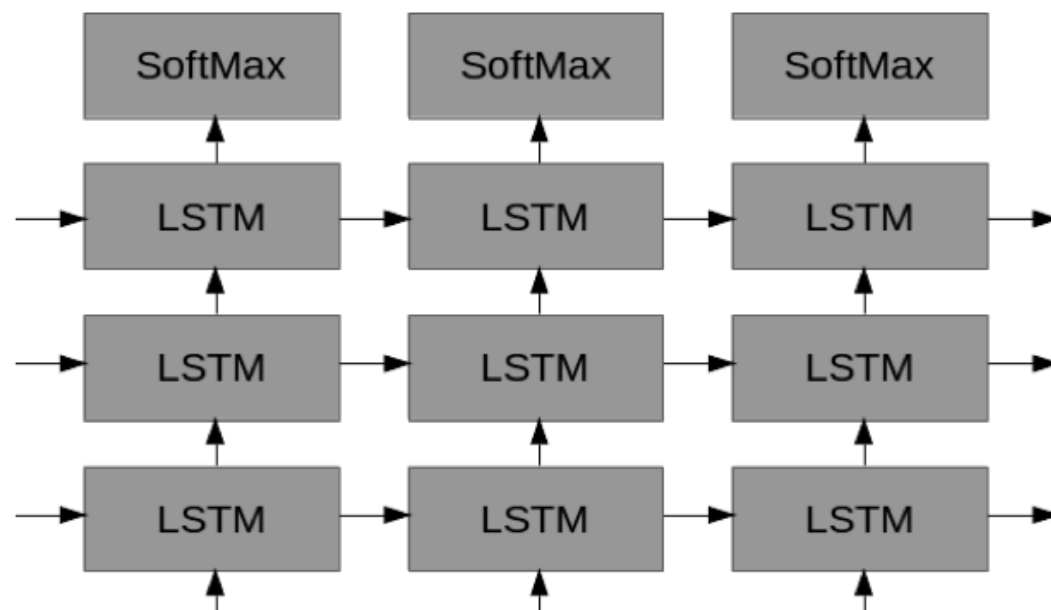
- 对于那些对实时性要求不高的序列标注任务，事实上序列的标注可以在观察到整个序列之后才开始。所以我们就可以设计双向的模型。
- 由于学习到了序列在前后两个方向上的依赖关系，双向模型会比单向模型有更好的performance。
- 双向模型需要观察到整个序列之后才能开始对第一个元素的标注。因此只适用于对实时性要求不高的序列标注任务。幸运的是，大部分任务（POS tagging、NER.....）在大多数场景下都不强调很高的实时性。



I have got a raspberry pi device .

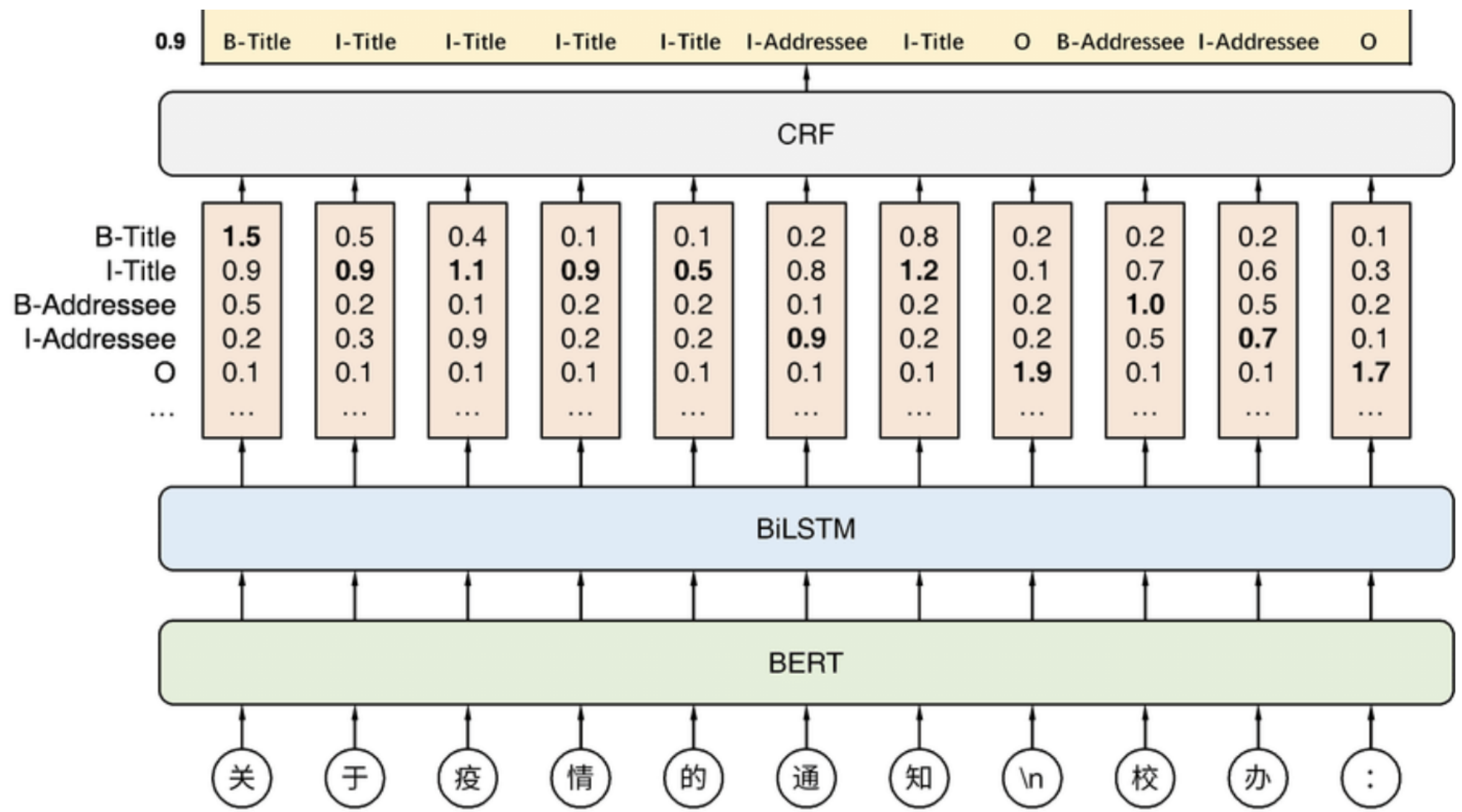
- **多层叠加的LSTM或GRU**

- 事实上，就像堆叠前向传播网络一样，如果我们不将LSTM (GRU)的输出给到softmax，而是将其作为新的序列，然后再次使用一个LSTM (GRU)来读取这个序列，形成新的输出，我们就得到了**多层叠加的LSTM (GRU)**。（这一叠加的过程可以不断重复，直到你想停止为止。）
- 对于双向模型，我们可以把两个方向上的隐层表征粘合（concatenate）到一起，再**整体**作为下一个双向层的输入。
- 同样，也像堆叠前向传播网络一样，多层叠加的LSTM (GRU)会比单层的有更好的performance。在某些情况下，模型表现的差距会非常大。



I have got a raspberry pi device .

序列标注：深度学习框架与传统统计模型的融合（LSTM+CRF）



不同架构的落地表现

Table 2: Comparison of tagging performance on POS, chunking and NER tasks for various models.

		POS	CoNLL2000	CoNLL2003
Random	Conv-CRF (Collobert et al., 2011)	96.37	90.33	81.47
	LSTM	97.10	92.88	79.82
	BI-LSTM	97.30	93.64	81.11
	CRF	97.30	93.69	83.02
	LSTM-CRF	97.45	93.80	84.10
	BI-LSTM-CRF	97.43	94.13	84.26
Senna	Conv-CRF (Collobert et al., 2011)	97.29	94.32	88.67 (89.59)
	LSTM	97.29	92.99	83.74
	BI-LSTM	97.40	93.92	85.17
	CRF	97.45	93.83	86.13
	LSTM-CRF	97.54	94.27	88.36
	BI-LSTM-CRF	97.55	94.46	88.83 (90.10)

[Huang et al., 2015]