Bidirectional LSTM-CRF Models for Sequence Tagging

ADVISOR: JIA-LING, KOH

SOURCE: CORR 2015

SPEAKER: SHAO-WEI, HUANG

DATE: 2020/01/15

OUTLINE

- Introduction
- Method
- Experiment
- Conclusion

INTRODUCTION

- Sequence tagging：Tag each character(part) in the sentence(sequence).
  - POS tagging

    (Ex): She lives in Taiwan.

    (PRO) (V) (Prep) (N)
  - Chunking

    (Ex): [Np He] [VP estimates] [Np the current account deficit] [VP will shrink] [PP to] [NP just 1.8 billion].

INTRODUCTION

- Sequence tagging：Tag each character in the sentence(sequence).
  - Name entity recognization：

    (Ex): EU rejects German call to boycott British lamb .

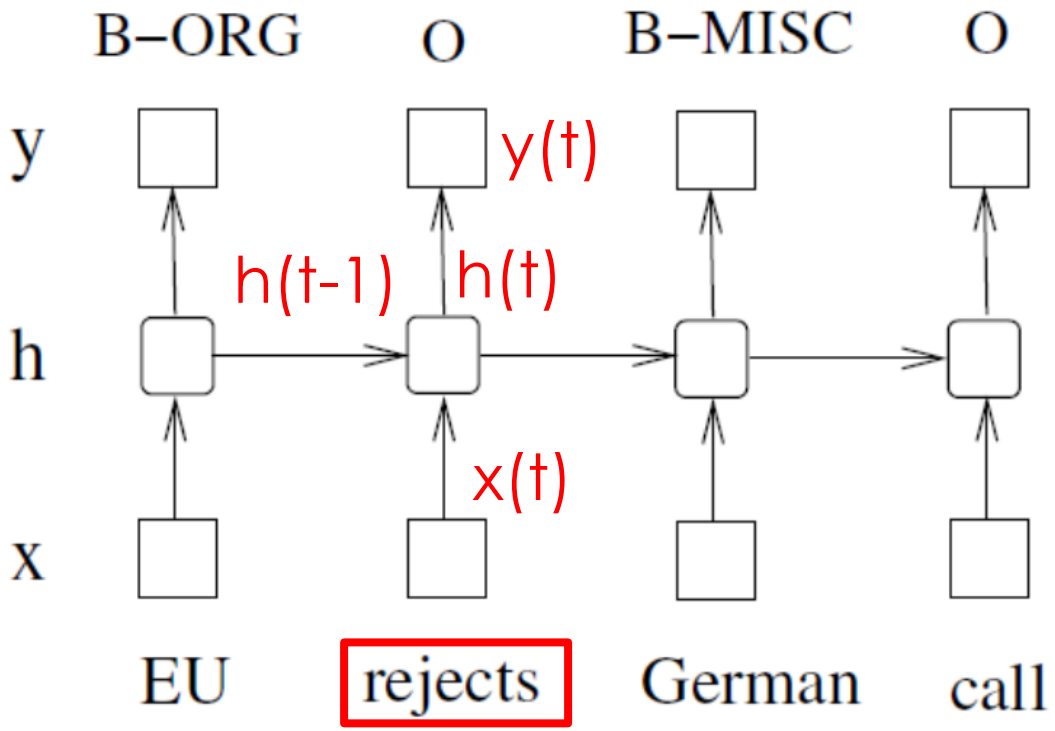
    (B-ORG) (O) (B-MISC) (O) (O) (O) (B-MISC) (O)

OUTLINE

- Introduction
- Method
- Experiment
- Conclusion

METHOD Simple RNN model

➢**Simple RNN：**



$$\mathbf{h}(t) = f(\mathbf{U}\mathbf{x}(t) + \mathbf{W}\mathbf{h}(t-1)),$$
$$\mathbf{y}(t) = g(\mathbf{V}\mathbf{h}(t)),$$

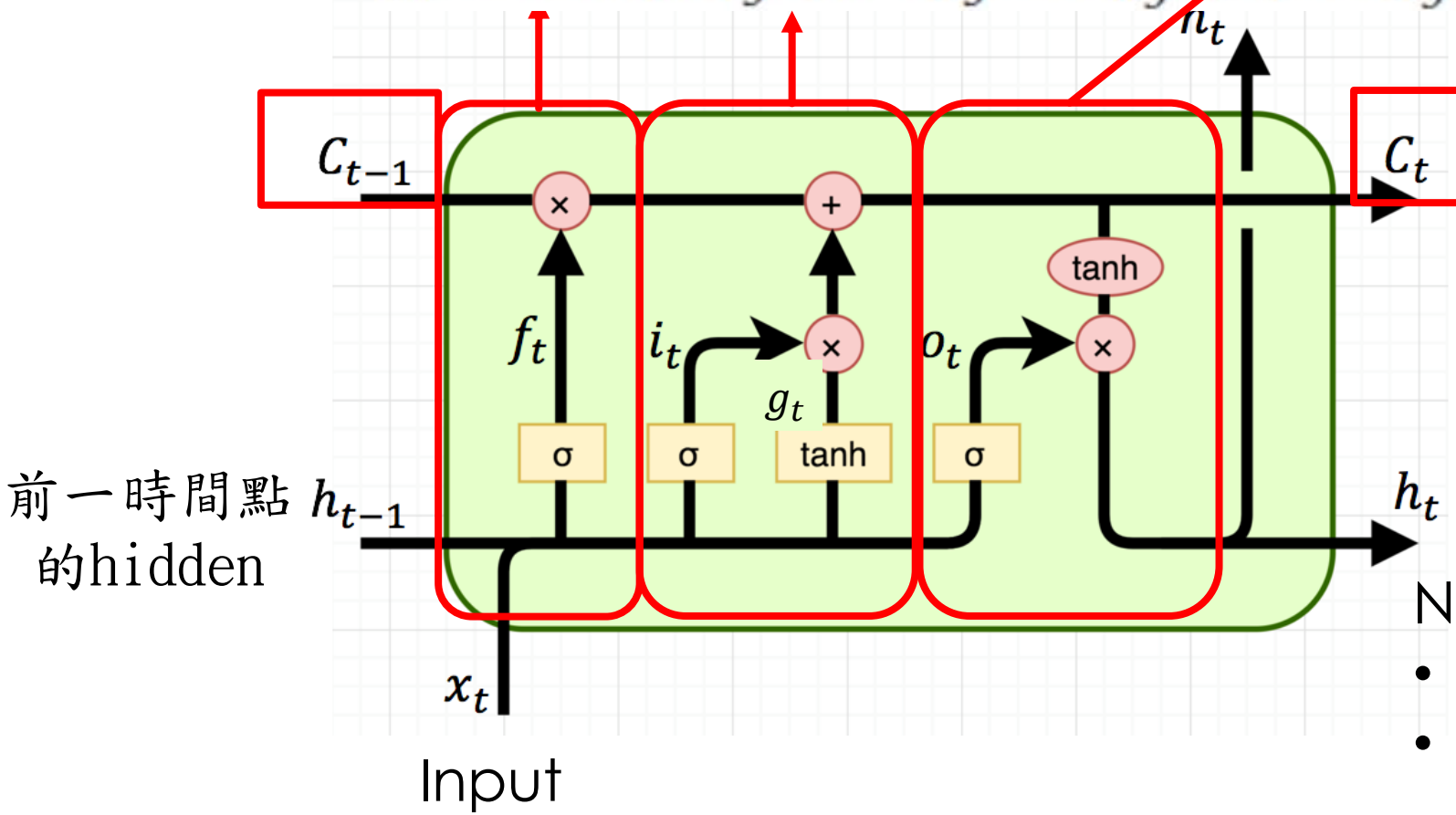
$$o_t = \sigma(W_{xo}x_t + b_{xo} + W_{ho}h_{t-1} + b_{ho})$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$
$$h_t = o_t \odot tanh(c_t)$$

METHOD LSTM model

$$i_t = \sigma(W_{xi}x_t + b_{xi} + W_{hi}h_{t-1} + b_{hi})$$
$$g_t = tanh(W_{xg}x_t + b_{xg} + W_{hg}h_{t-1} + b_{hg})$$

➢**LSTM**：

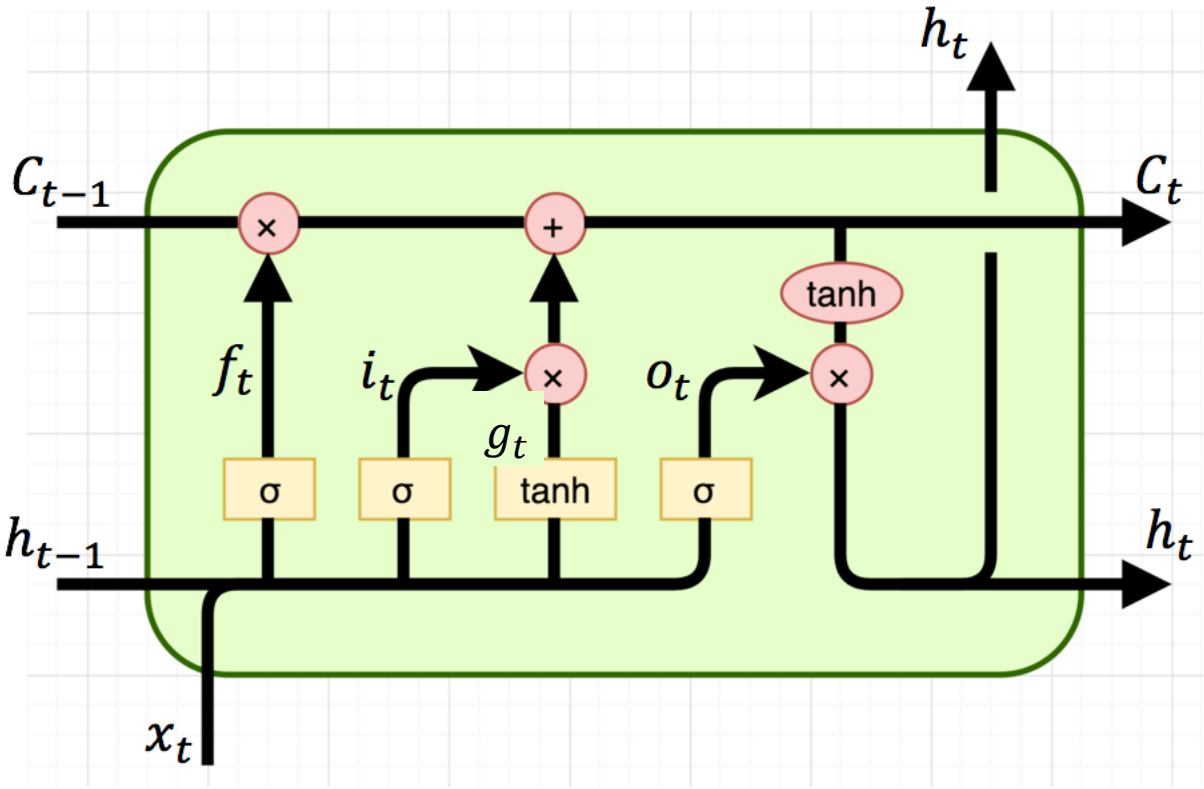
- **Forget gate**：根據上一時間點的輸出與本時間點的輸入，選擇需要在細胞中遺忘多少。
- **Output gate**：根據上一時間點的輸出與本時間點的輸入，選擇需要在細胞中遺忘多少。



Note:
- σ is the sigmoid function.
- ⊙ = ⊗ is the element-wise product.

METHOD LSTM model

➢**LSTM：**

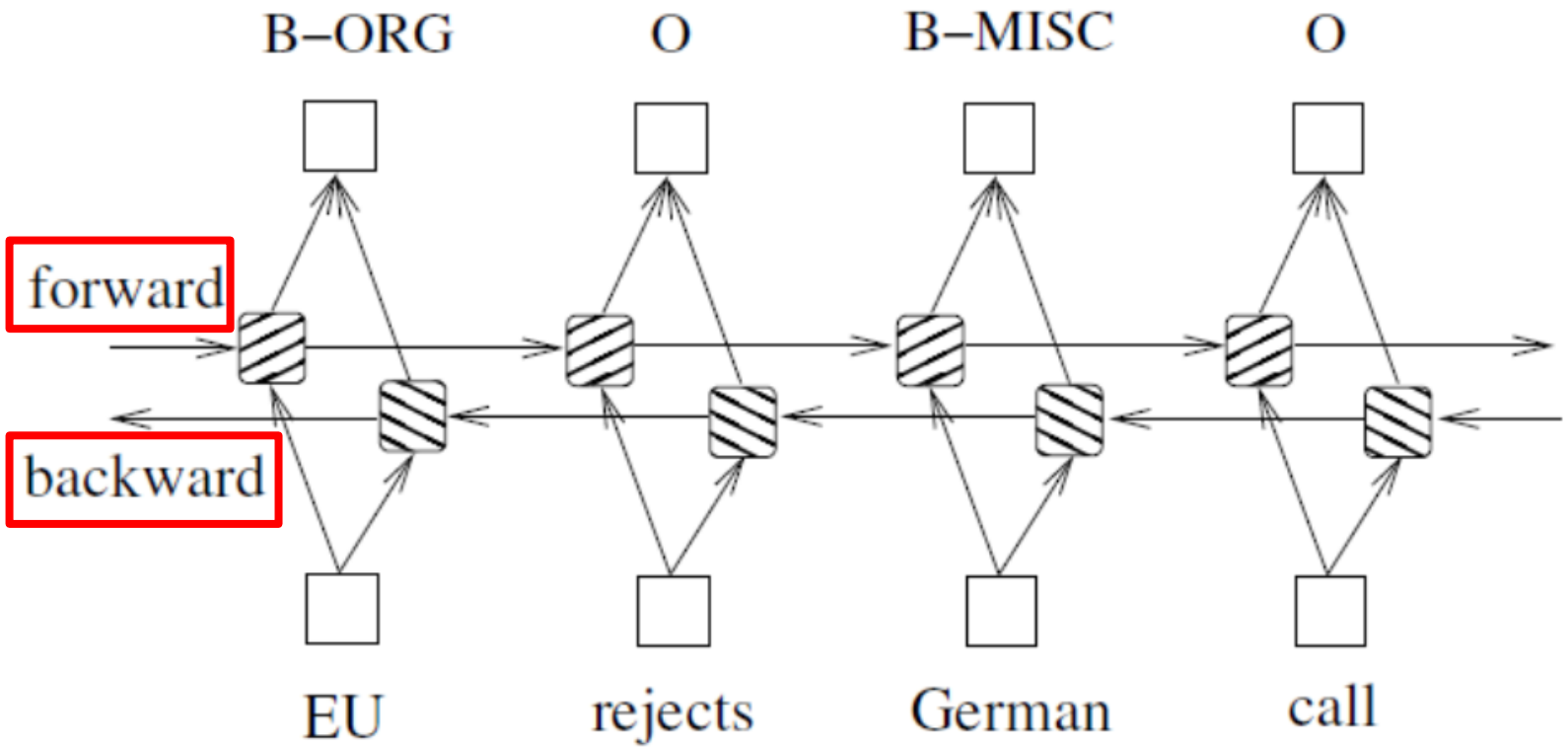


$$i_t = \sigma(W_{xi}x_t + b_{xi} + W_{hi}h_{t-1} + b_{hi})$$
$$f_t = \sigma(W_{xf}x_t + b_{xf} + W_{hf}h_{t-1} + b_{hf})$$
$$g_t = tanh(W_{xg}x_t + b_{xg} + W_{hg}h_{t-1} + b_{hg})$$
$$o_t = \sigma(W_{xo}x_t + b_{xo} + W_{ho}h_{t-1} + b_{ho})$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$
$$h_t = o_t \odot tanh(c_t)$$

https://www.itread01.com/content/1545027542.html

METHOD Bi-LSTM model

➢**Bi-LSTM：**



METHOD CRF model

- **CRF**：Instead of modeling tagging decisions independently, CRF model them jointy.
- X = $(x_1, x_2, \ldots, x_n)$ → an input sentence.
  y = $(y_1, y_2, \ldots, y_n)$ → a sequence of predictions(tags).

A：

	tag1	tag2	tag3	tag4
tag1	0.6	0.2	0.1	0.1
tag2	0.1	0.1	0.1	0.7
tag3	0.1	0.7	0.1	0.1
tag4	0.5	0.1	0.1	0.3

- Score：

$$s(\mathbf{X}, \mathbf{y}) = \sum_{i=0}^{n} \boxed{A_{y_i, y_{i+1}}} + \sum_{i=1}^{n} \boxed{P_{i, y_i}}$$

P：

	tag1	tag2	tag3	tag4
W1	0.7	0.1	0.1	0.1
W2	0.1	0.1	0.1	0.7
W3	0.1	0.7	0.1	0.1

A matrix of transition scores, A_{y_i, y_i+1} represents the score of a transition from the tag y_i to tag y_{i+1}.

The score of the $y_i{}^{th}$ tag of the ith word in a sentence(independently).

METHOD CRF model

- Normalization :

$$p(\mathbf{y}|\mathbf{X}) = \frac{e^{s(\mathbf{X},\mathbf{y})}}{\sum_{\tilde{\mathbf{y}}\in\mathbf{Y}_{\mathbf{X}}} e^{s(\mathbf{X},\tilde{\mathbf{y}})}}.$$

- Loss function :

$$L_{ner} = -\log(\boxed{p(y|X)}) \longrightarrow \text{Max}$$

METHOD CRF model、LSTM-CRF model、Bi-LSTM-CRF model

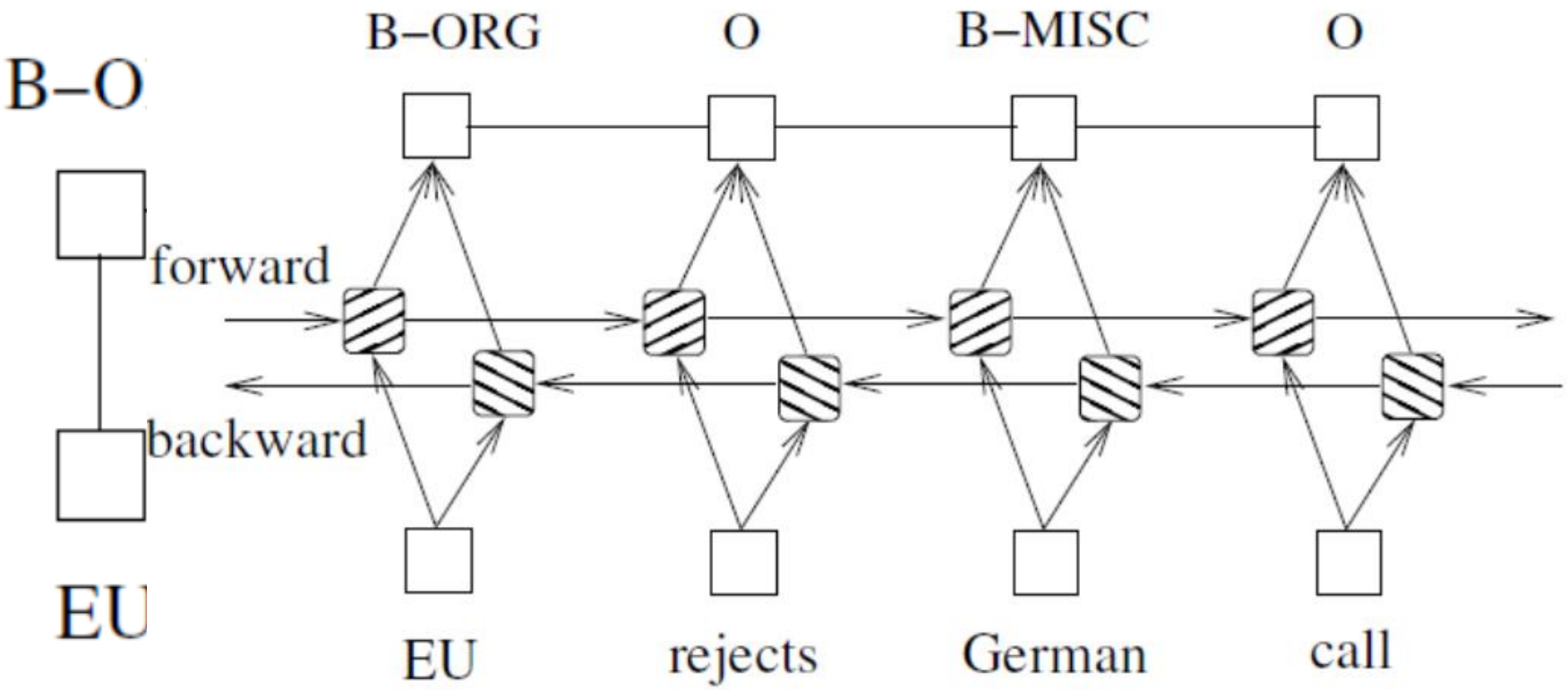


Figure 7: A BI-LSTM-CRF model.

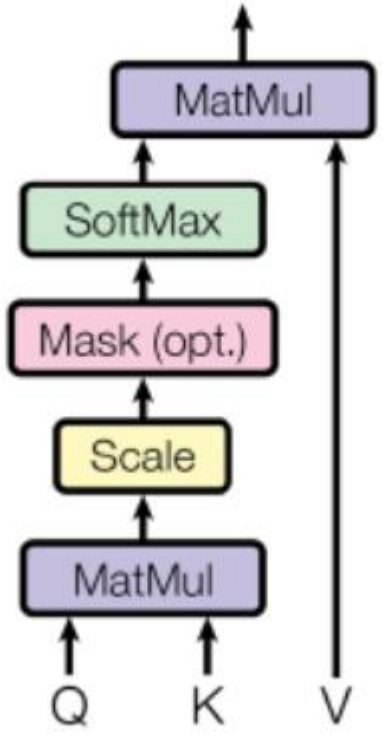
https://blog.csdn.net/jiaowoshouzi/article/details/89073944

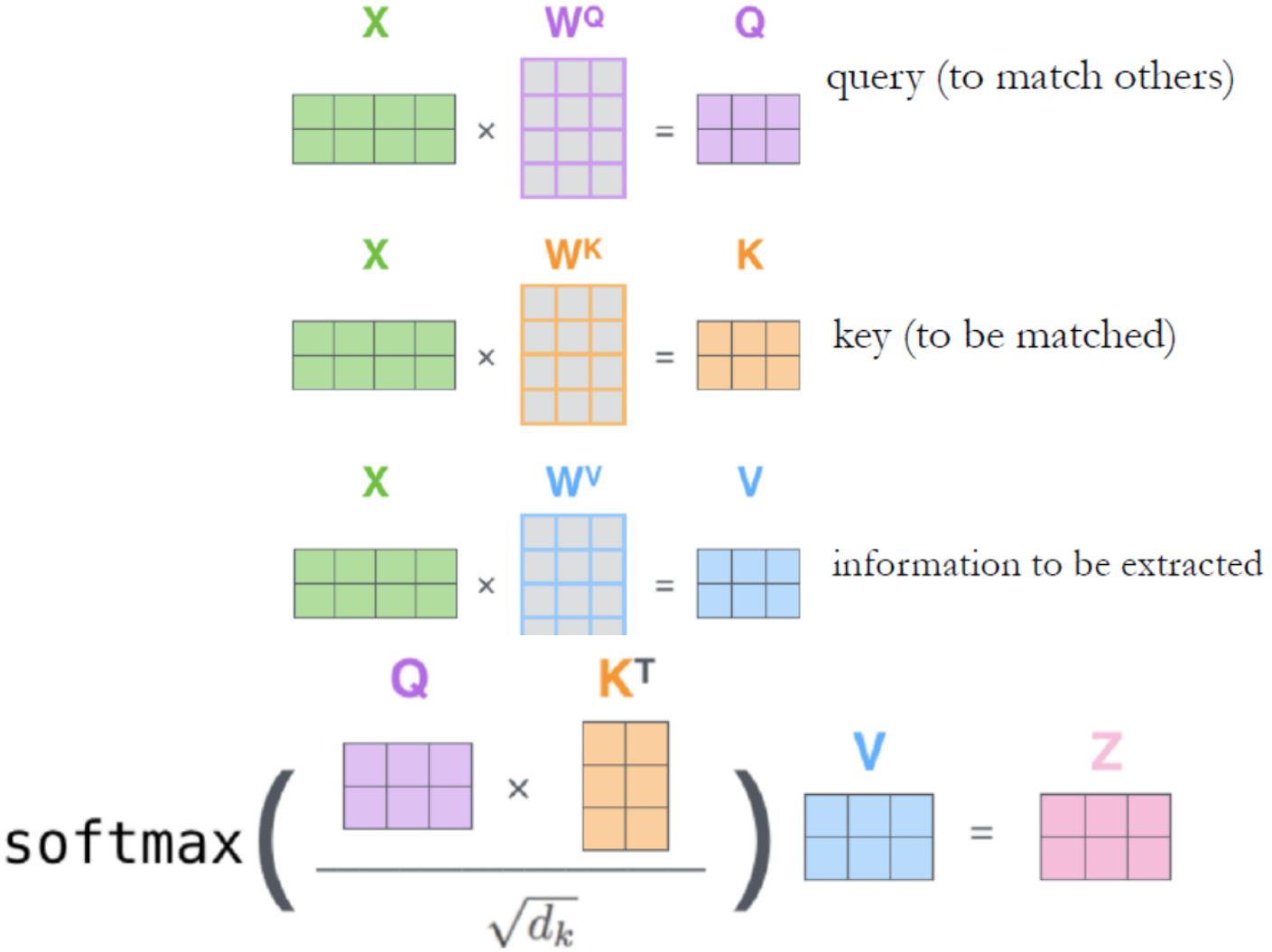
Reference：*Attention is all you need*

METHOD **BERT_Transformers**

➢Self Attention ：

Scaled Dot-Product Attention

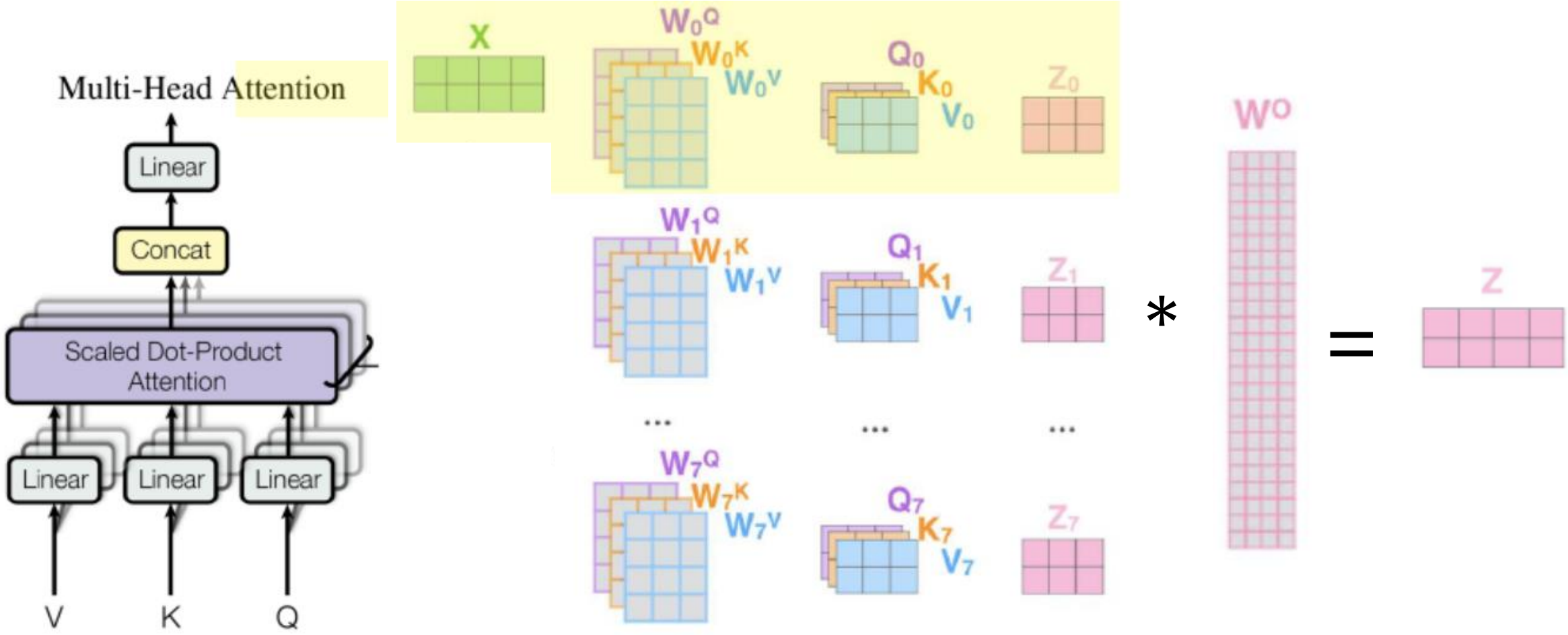




Reference：*Attention is all you need*

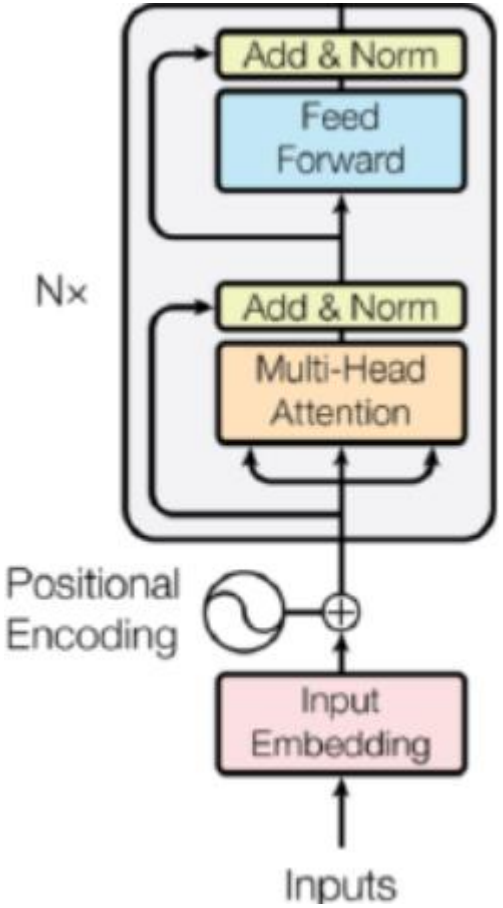
METHOD **BERT_Transformers**

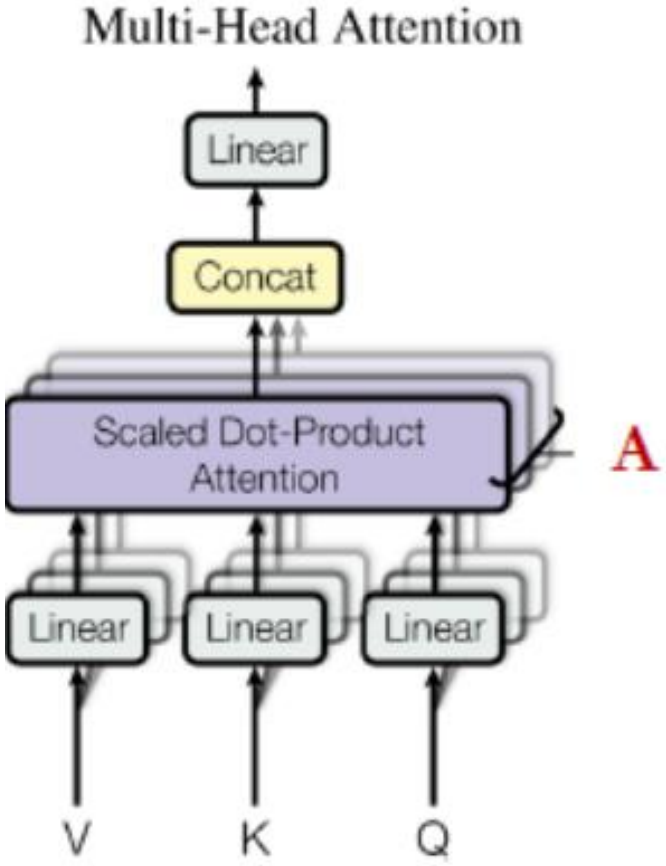
➢Multi-Head Attention ：



METHOD BERT model

➢**BERT model：**





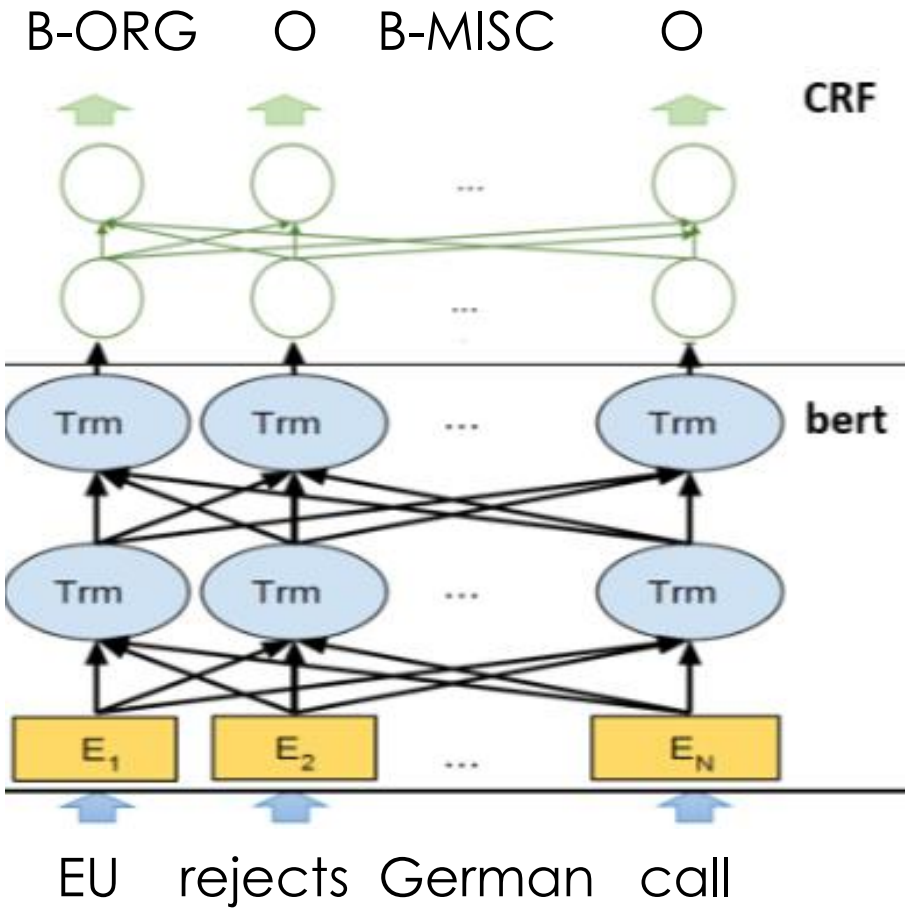
$BERT_{BASE}$
(L=12, H=768, A=12, Parameters=110M)

$BERT_{LARGE}$
(L=24, H=1024, A=16, Parameters=340M)

Reference：*Transfer learning for scientific data chain extraction in small chemical corpus with BERT-CRF model*

METHOD **BERT-CRF model**

➢**BERT-CRF model**：Connect CRF layer behind BERT's hidden layer.



OUTLINE

- Introduction
- Method
- Experiment
- Conclusion

EXPERIMENT

Dataset

- **Penn TreeBank (PTB)** ：POS tagging
- **CoNLL 2000**： chunking
- **CoNLL 2003** ：named entity tagging

		POS	CoNLL2000	CoNLL2003
training	sentence #	39831	8936	14987
	token #	950011	211727	204567
validation	sentence #	1699	N/A	3466
	token #	40068	N/A	51578
test	sentences #	2415	2012	3684
	token #	56671	47377	46666
	label #	45	22	9

EXPERIMENT

Features

- **Spelling features**
- **Context features**：uni-gram、bi-gram、tri-gram
- **Word embedding**：Senna word enbedding（each word corresponds to a 50-dimensional embedding vector.）

EXPERIMENT

- Comparison with other networks :

Accuracy F1 F1

		POS	CoNLL2000	CoNLL2003
Random				
Senna				

EXPERIMENT

- Performance with only word feature：

		POS (Accuracy)	CoNLL2000 (F1)	CoNLL2003 (F1)
Senna	LSTM	94.63 (-2.66)	90.11 (-2.88)	75.31 (-8.43)
	BI-LSTM	96.04 (-1.36)	93.80 (-0.12)	83.52 (-1.65)
	CRF	94.23 (-3.22)	85.34 (-8.49)	77.41 (-8.72)
	LSTM-CRF	95.62 (-1.92)	93.13 (-1.14)	81.45 (-6.91)
	BI-LSTM-CRF	**96.11** (-1.44)	**94.40** (-0.06)	**84.74** (-4.09)

OUTLINE

- Introduction
- Method
- Experiment
- Conclusion

CONCLUSION

- Systematically compare the performance of aforementioned models.
- The first to apply a bidirectional LSTM CRF model to NLP benchmark sequence tagging data sets.
- Show that BI-LSTMCRF model is robust and it has less dependence on word embeddin.
- BERT+CRF model(proposed in another paper).