Attention-based No-Seg Chinese NER

2018-12-27

1. Background

About named entity recognition…

Background - NER example (character-based 字)

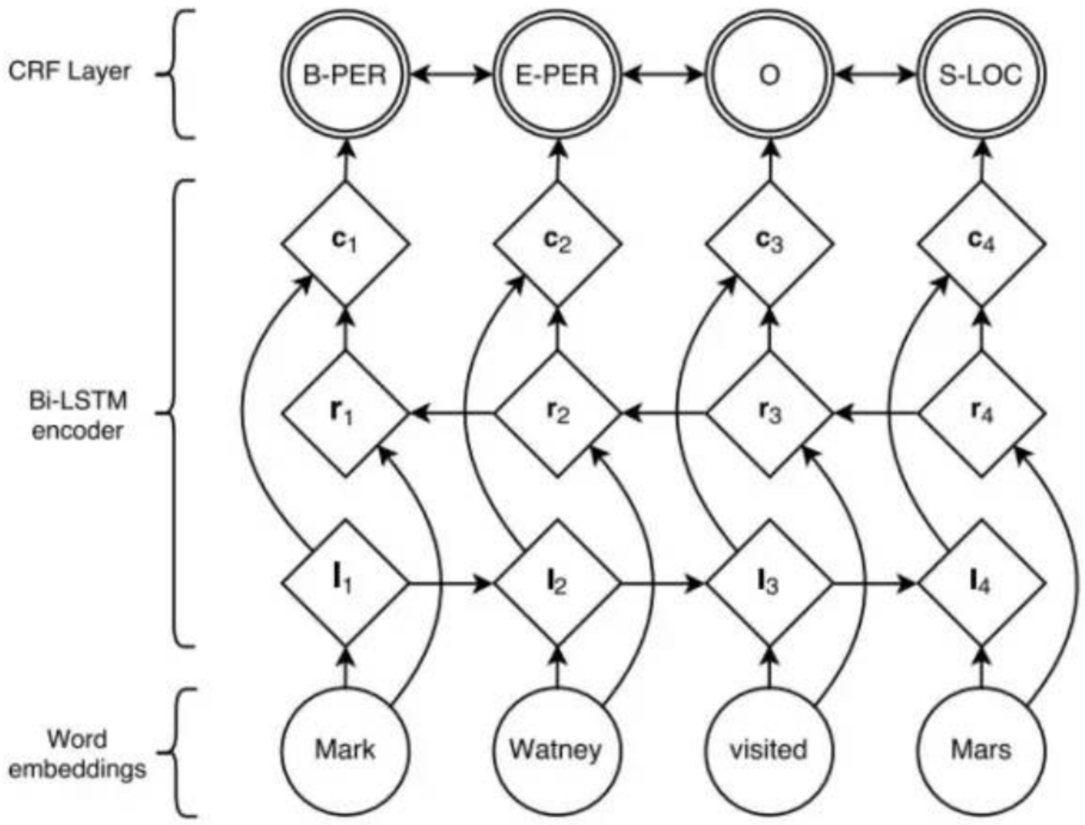
调	查	范	围	涉	及	故	宫	、	古	研	所	、	北	大	清	华
O	O	O	O	O	O	B-LOC	E-LOC	O	B-ORG	I-ORG	E-ORG	O	B-LOC	I-LOC	I-LOC	I-LOC

图	书	馆	、	日	伪	资	料	库	等	二	十	几	家	。
I-LOC	I-LOC	E-LOC	O	S-LOC	O	O	O	O	O	O	O	O	O	O

- Using BIOES style, b-begin, i-inside, e-end, o-outside, s-single
- Sequence labeling

Background – basic deep learning model (word-based)

- Bi-LSTM: encoding information at each time stamp.
- CRF(conditional random field): learning rules between tags in a sequence. (B-xx should be followed by I-xx or E-xx)

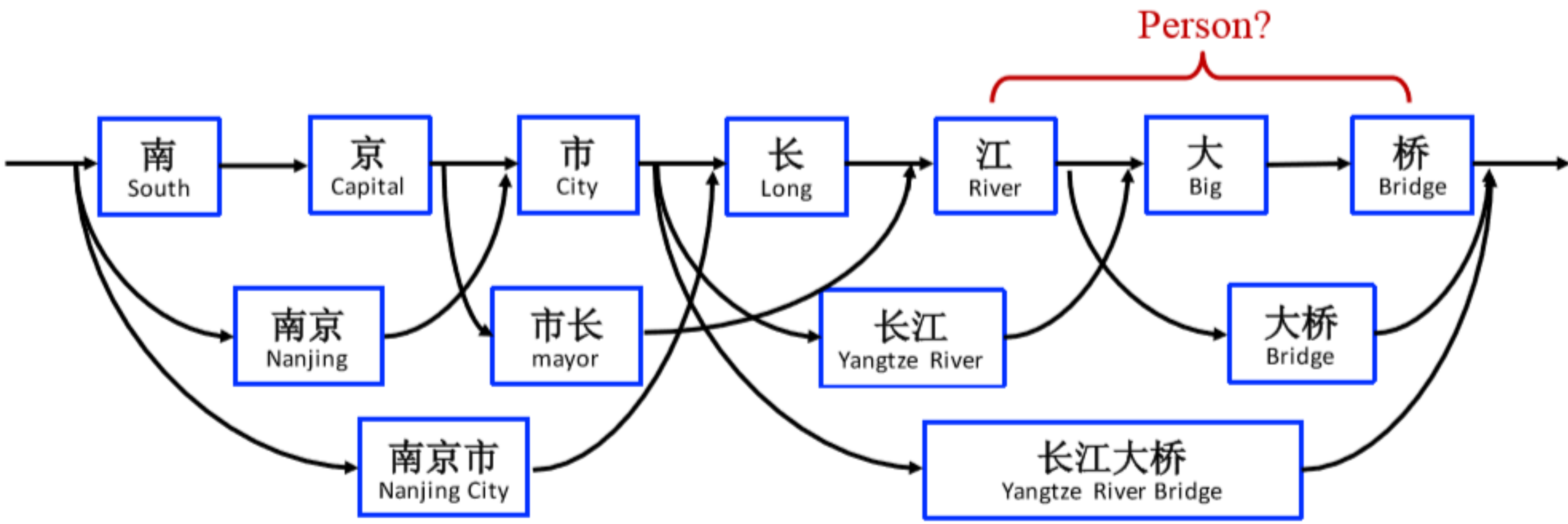


B- begin
I – inside
E- end
O- outside
S- single

2. Challenges

With Chinese NER…

Challenges



- **No natural word boundaries.**

南京/市长/江大桥 or 南京市/长江大桥

- If use vanilla character-based bi-LSTM (feed character embeddings):

  --lack word semantic information

- If use word-based bi-LSTM (feed word embeddings)

  --word segmentation error propagation, resolve entities

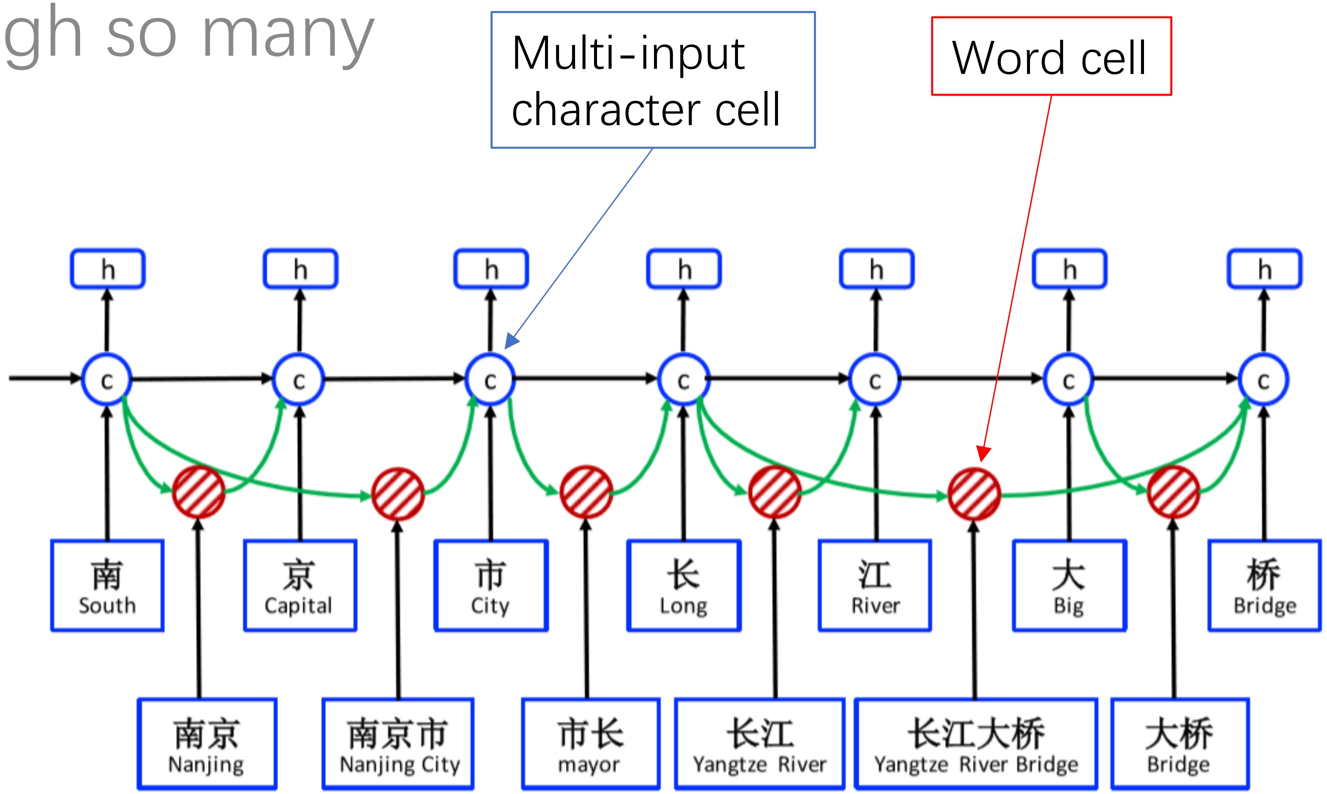
Goal

Better integrate word-level information and character-level information.

Lattice LSTM (SOTA)

- Redundant and ineffective
  LSTM cells are costly,
  Forgetting and updating through so many word cells and character cells.

- Loss of info:
  Information of a word only arrives at the ending character of this word, without influence on characters before it.



3. Proposed Model

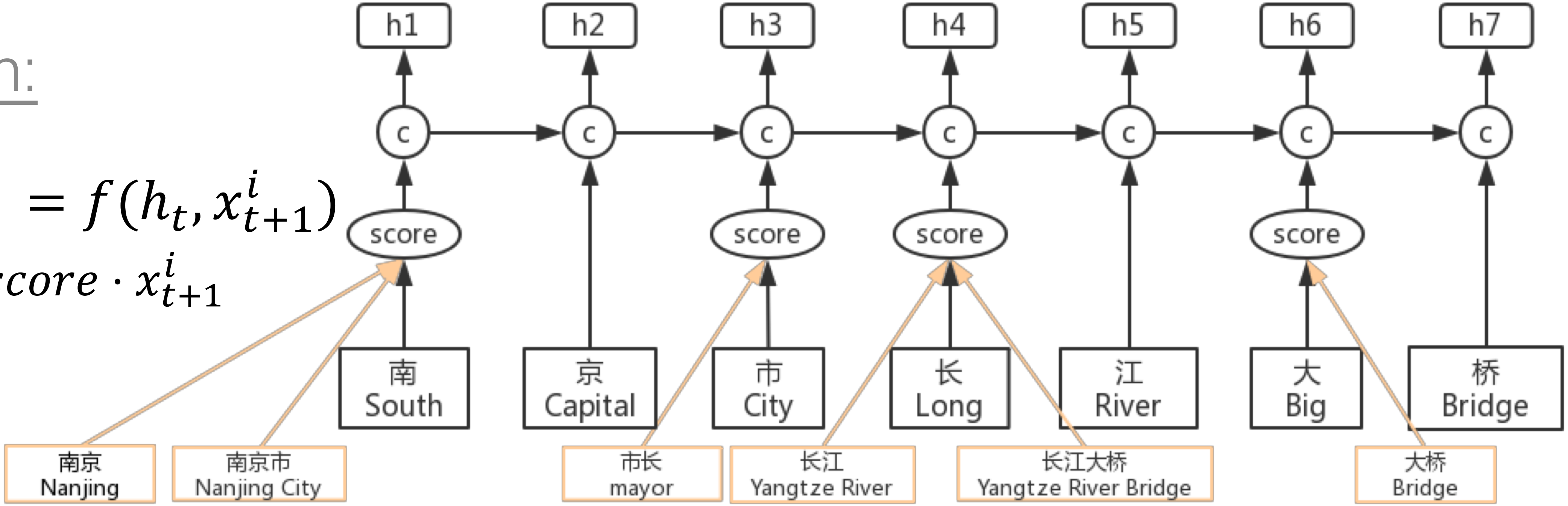
Inspired by *Chinese NER Using Lattice LSTM, 2018 ACL*

Proposed Model 1

attention:

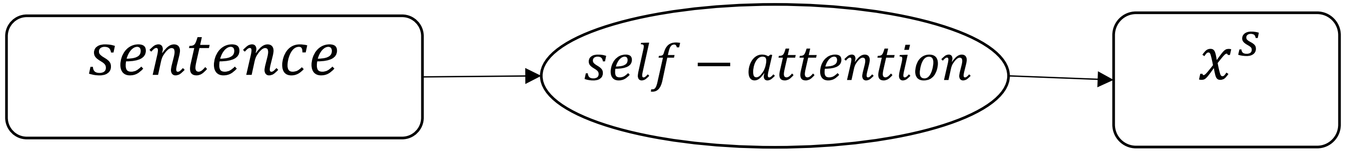
$$score^{\mathrm{i}}_{t+1} = f(h_t, x^i_{t+1})$$

$$x'_{t+1} = \sum score \cdot x^i_{t+1}$$



*the words come from auto segmentation on a huge corpus

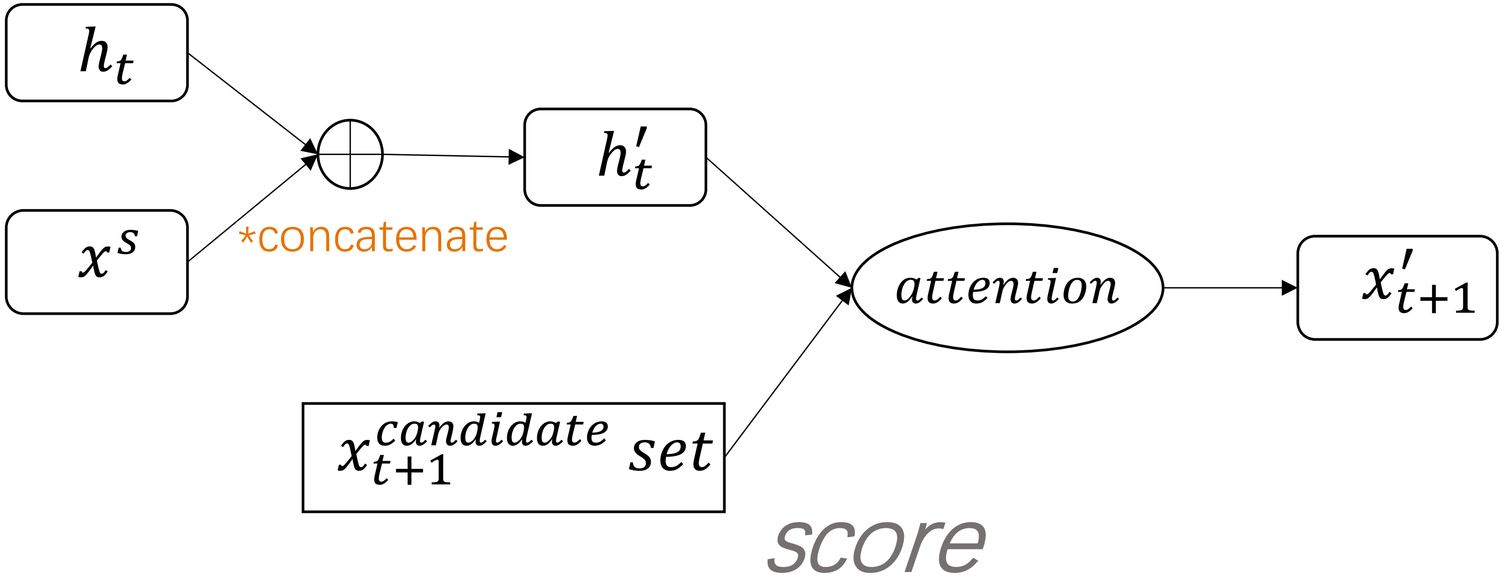
Proposed Model 2



attention:

$$score^{i}_{t+1} = f(h'_t, x^i_{t+1})$$

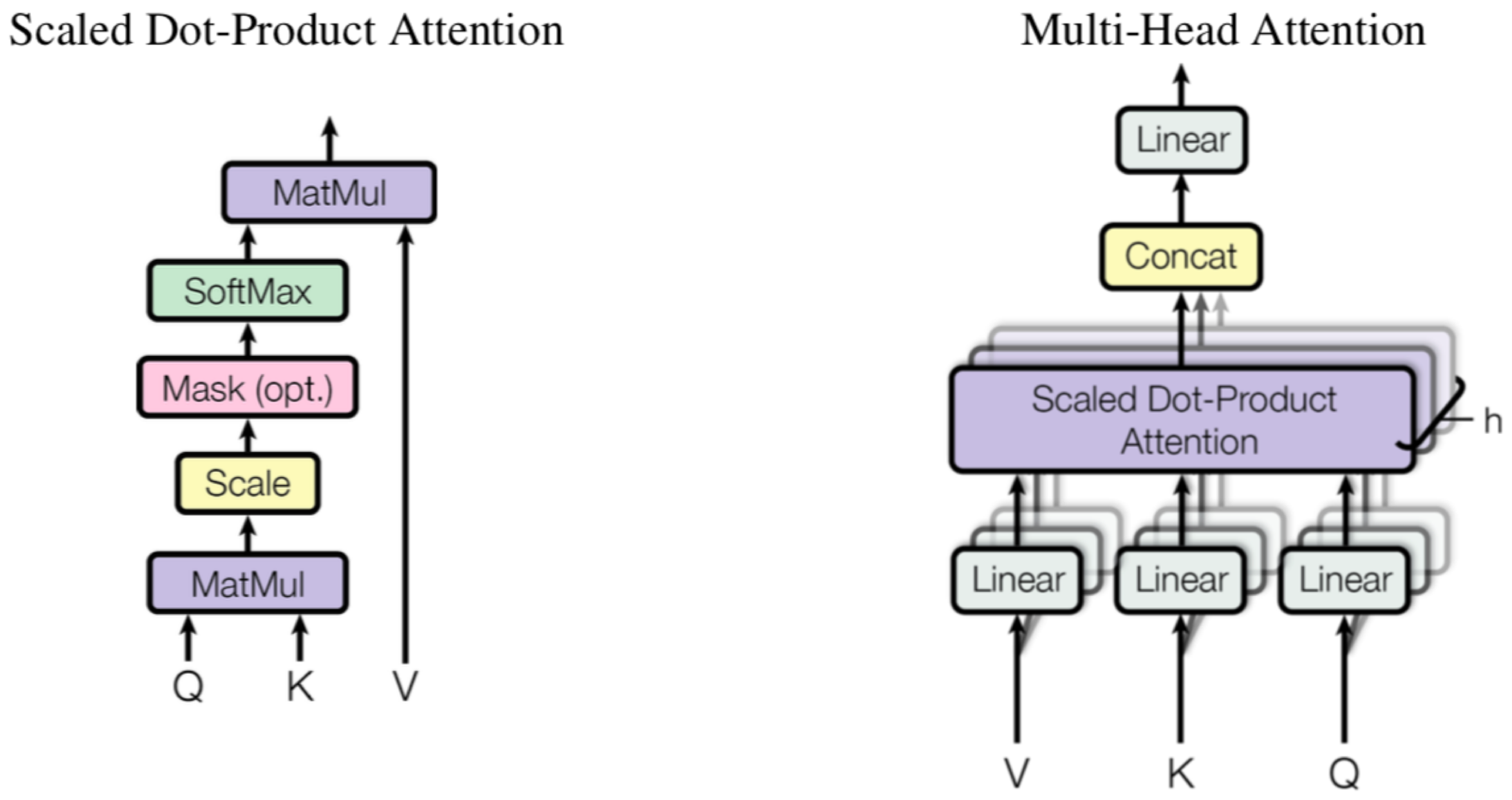
$$x'_{t+1} = \sum score \cdot x^i_{t+1}$$



Proposed Model 2

Self-attention(multi-head):

$Q = K = V$



$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Possible Attempts

self-attention

vanilla character-based bi-LSTM

Sentence representation extraction

supervise

attention scoring of candidate words

Possible Attempts

1. Transformation from word embedding space to character embedding space (failed)

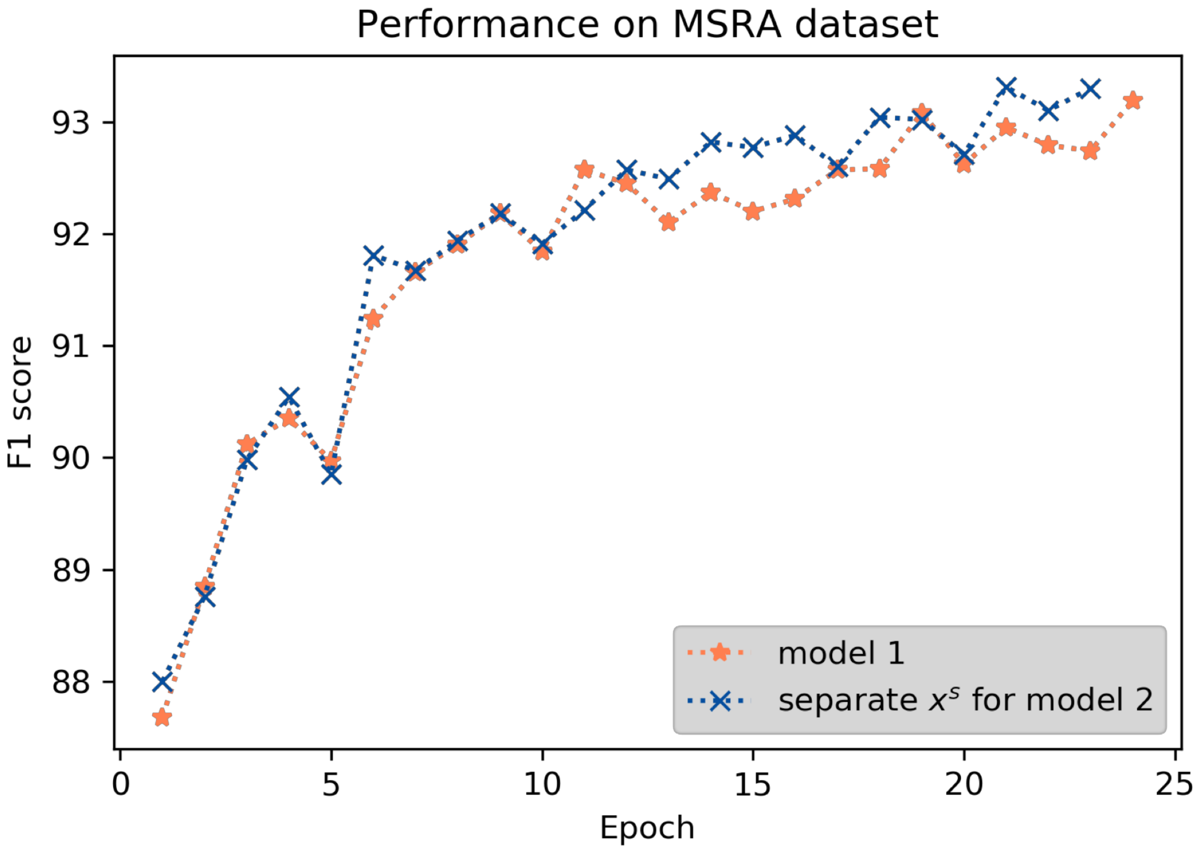
$$score^{c} \cdot x_{t+1}^{c} + \sum_{i} score^{w_i} \ W_{w \to c} \cdot x_{t+1}^{w_i}$$

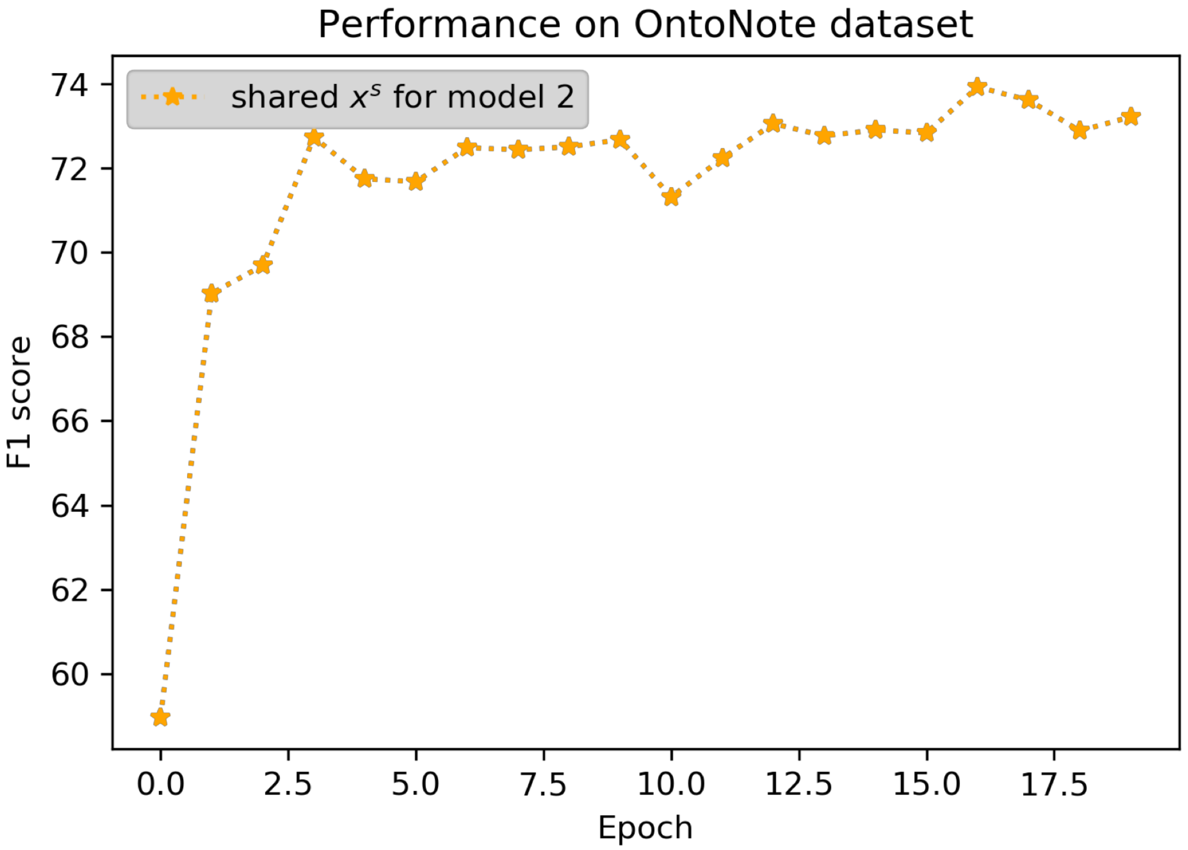
2. Sentence embedding: shared or separate

3. Hidden states from char lstm as sentence embeddings, attention over hidden states.

4. Different ways to combine x^{s} and h_t : concat, sum, mul, …

Results so far...





Results so far…

This part is deprecated.
The current best F1 for MSRA is:
Model 1 - 93.59
Model 2 – 93.67

Datasets	**Models**	**P**	**R**	**F1**
MSRA	Lattice	93.57	**92.79**	93.18
	Model 1	93.95	92.44	93.20
	Model 2	**94.51**	92.15	**93.33**
OntoNotes	Lattice	**76.35**	71.56	73.88
	Model 2	74.56	**73.30**	**73.93**

Thanks.