Simplify the Usage of Lexicon in Chinese NER

Ruotian Ma[1*], **Minlong Peng**[1*], **Qi Zhang**[1,3], **Zhongyu Wei**[2,3], **Xuanjing Huang**[1]
[1]Shanghai Key Laboratory of Intelligent Information Processing,
School of Computer Science, Fudan University
[2]School of Data Science, Fudan University
[3]Research Institute of Intelligent and Complex Systems, Fudan University
{rtma19,mlpeng16,qz,zywei,xjhuang}@fudan.edu.cn

Introduction

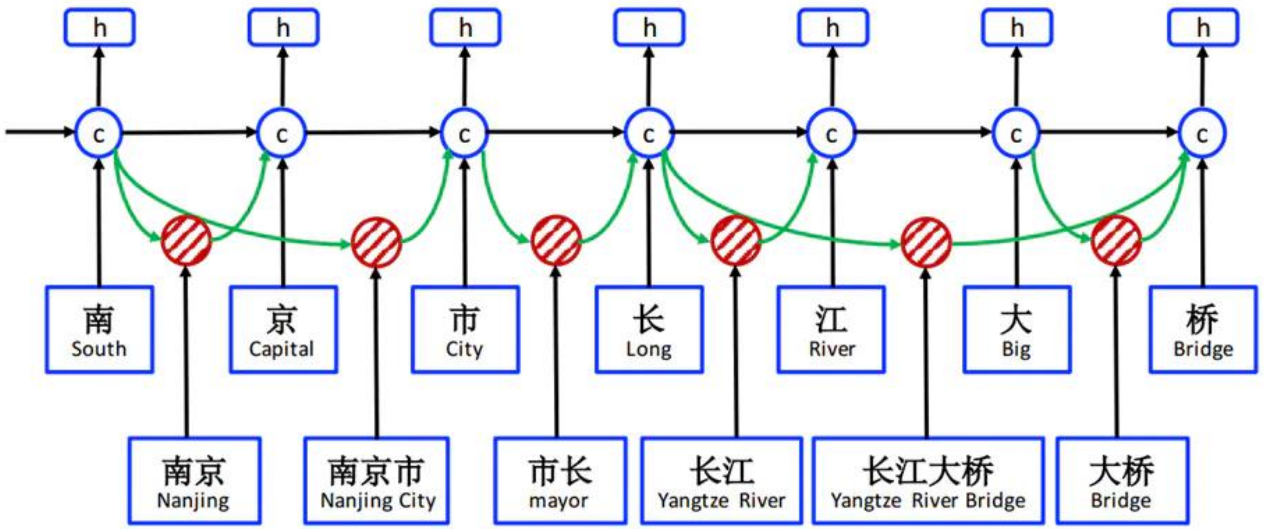
- Chinese NER is more difficult since sentences in Chinese are not naturally segmented.

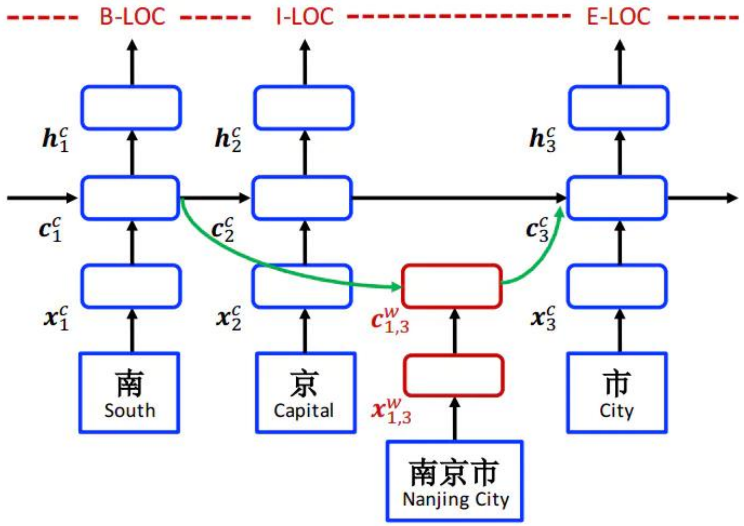
✓First perform word segmentation(result in errors in the detection of entity boundary and the prediction of entity category in NER)

✓Perform Chinese NER directly at the **character level**(word information is not fully exploited)

✓**Lattice-LSTM** for incorporating word lexicons into the character-based NER model(complicated, slow its training and inference speeds.)

Lattice-LSTM

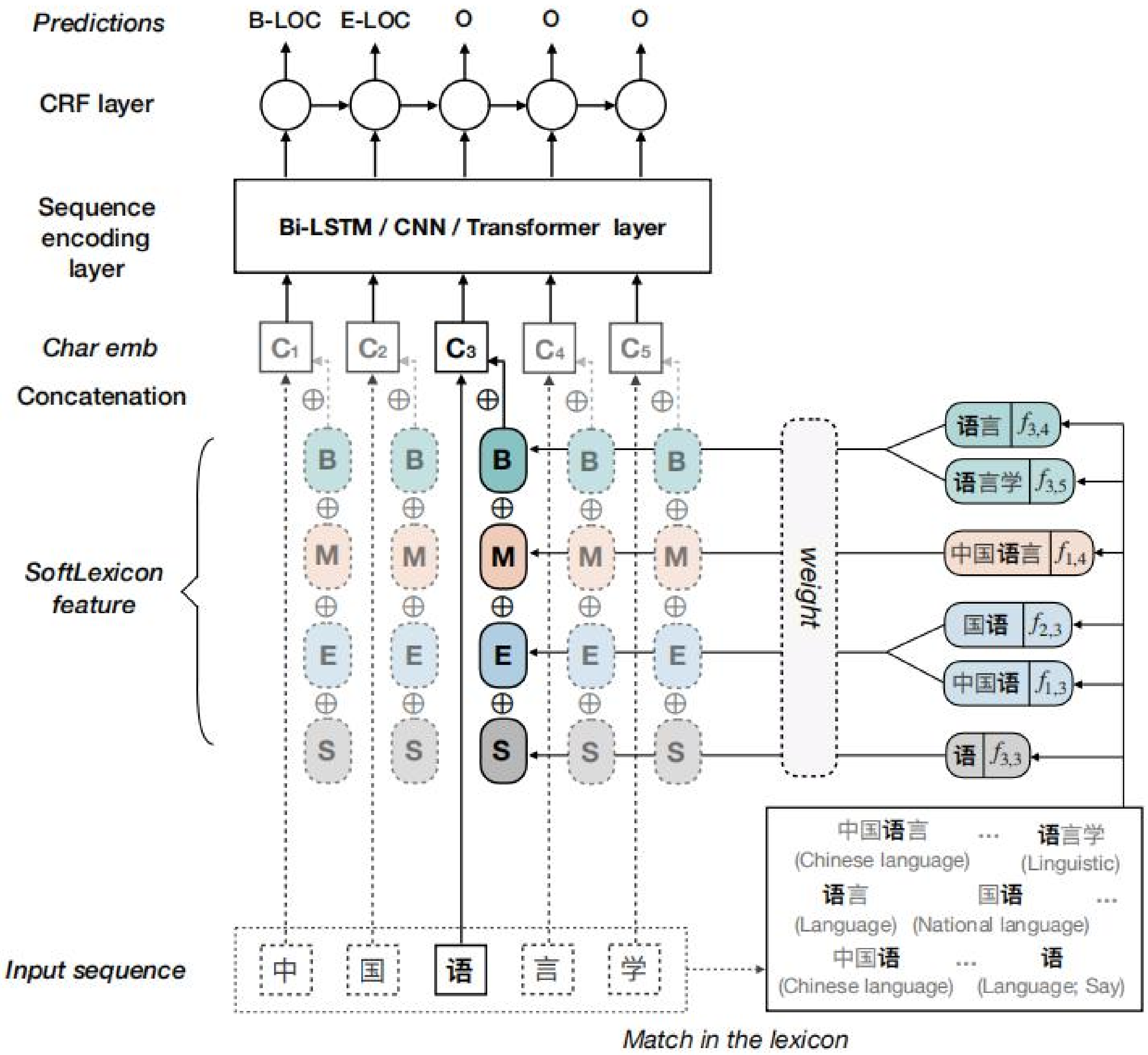




- **subsequence** $\{ci, \cdots, cj\}$ $h_j, c_j = f(h_{j-1}, c_{j-1}, x_j^c, s_{<*,j>}, h_{<*,j>}, c_{<*,j>})$
- 优点：所有词语的信息均被保留，而不是启发式选取某一个词语，可能导致的错误累积；使用了预训练的word embeddings。
- 缺点：效率问题：每一个时间步的update需要额外传入 s<*,j>, h<*,j>,c<*,j> Lattice-LSTM设计得太复杂，模型不容易实现；导致无法平行处理多个样本（因此在发布的模型中，batch_size设置为1。

Summary

- Propose a simple but effective method for **incorporating word lexicons** into the character representations for Chinese NER
- The proposed method is **transferable to different sequence-labeling architectures** and can be easily incorporated with pre-trained models like BERT



Predictions
B-LOC
E-LOC
O
O
O
CRF layer
Sequence encoding layer
Bi-LSTM / CNN / Transformer layer
Char emb
C1
C2
C3
C4
C5
Concatenation
SoftLexicon feature
B
M
E
S
weight
语言 $f_{3,4}$
语言学 $f_{3,5}$
中国语言 $f_{1,4}$
国语 $f_{2,3}$
中国语 $f_{1,3}$
语 $f_{3,3}$
中国语言 (Chinese language)
语言学 (Linguistic)
语言 (Language)
国语 (National language)
中国语 (Chinese language)
语 (Language; Say)
Input sequence
中
国
语
言
学
Match in the lexicon

Character Representation Layer

s = {c1, c2, · · · , cn} **Char + bichar**: $x_i^c = [e^c(c_i); e^b(c_i, c_{i+1})]$,

Incorporating Lexicon Information

Softword Feature: The Softword technique was originally used for incorporating **word segmentation information** into downstream tasks.**{B, M, E, S, O}**

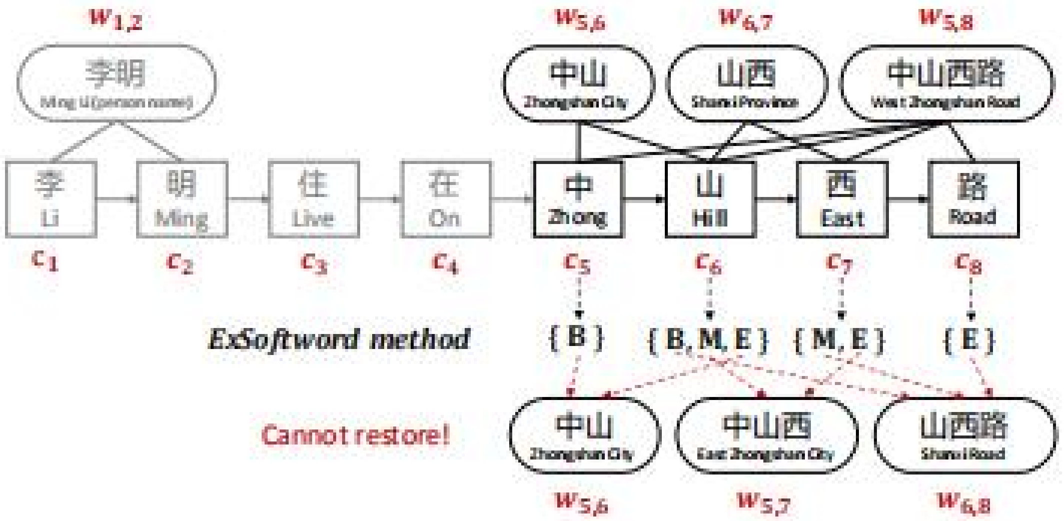


Figure 2: The ExSoftword method.

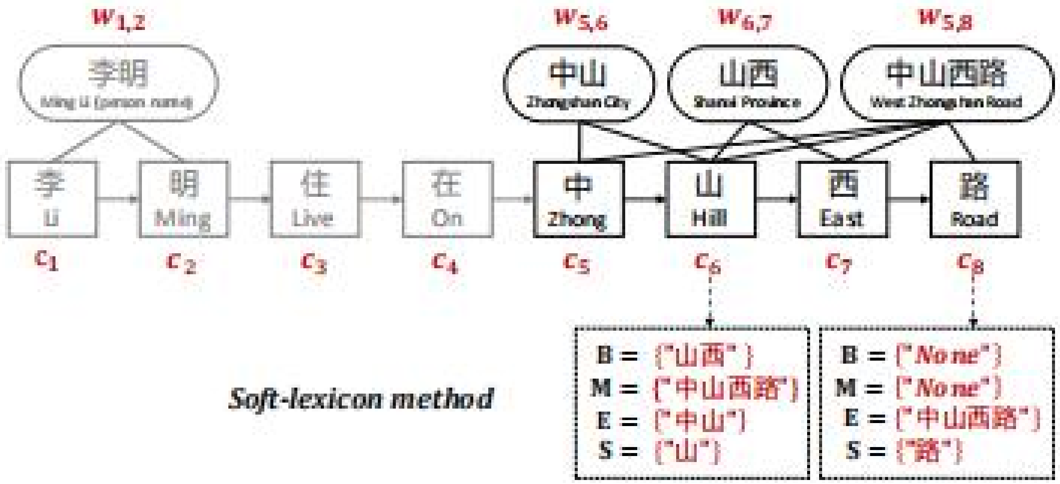


Figure 3: The SoftLexicon method.

Categorizing the matched words.

$$
\begin{aligned}
\mathrm{B}(\mathrm{c_i}) &= \{\mathrm{w_{i,k}}, \forall \mathrm{w_{i,k}} \in \mathrm{L}, \mathrm{i} < \mathrm{k} \leq \mathrm{n}\},\\
\mathrm{M}(\mathrm{c_i}) &= \{\mathrm{w_{j,k}}, \forall \mathrm{w_{j,k}} \in \mathrm{L}, 1 \leq \mathrm{j} < \mathrm{i} < \mathrm{k} \leq \mathrm{n}\},\\
\mathrm{E}(\mathrm{c_i}) &= \{\mathrm{w_{j,i}}, \forall \mathrm{w_{j,i}} \in \mathrm{L}, 1 \leq \mathrm{j} < \mathrm{i}\},\\
\mathrm{S}(\mathrm{c_i}) &= \{\mathrm{c_i}, \exists \mathrm{c_i} \in \mathrm{L}\}.
\end{aligned}
$$

$$
\boldsymbol{v}^s(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{w \in \mathcal{S}} \boldsymbol{e}^w(w).
$$

$$
\boldsymbol{v}^s(S) = \frac{4}{Z} \sum_{w \in S} z(w) \boldsymbol{e}^w(w),
$$

$$
Z = \sum_{w \in \mathrm{B} \cup \mathrm{M} \cup \mathrm{E} \cup \mathrm{S}} z(w).
$$

$$
\begin{aligned}
\boldsymbol{e}^s(\mathrm{B}, \mathrm{M}, \mathrm{E}, \mathrm{S}) &= [\boldsymbol{v}^s(\mathrm{B}); \boldsymbol{v}^s(\mathrm{M}); \boldsymbol{v}^s(\mathrm{E}); \boldsymbol{v}^s(\mathrm{S})],\\
x^c &\leftarrow [x^c; \boldsymbol{e}^s(\mathrm{B}, \mathrm{M}, \mathrm{E}, \mathrm{S})].
\end{aligned}
$$

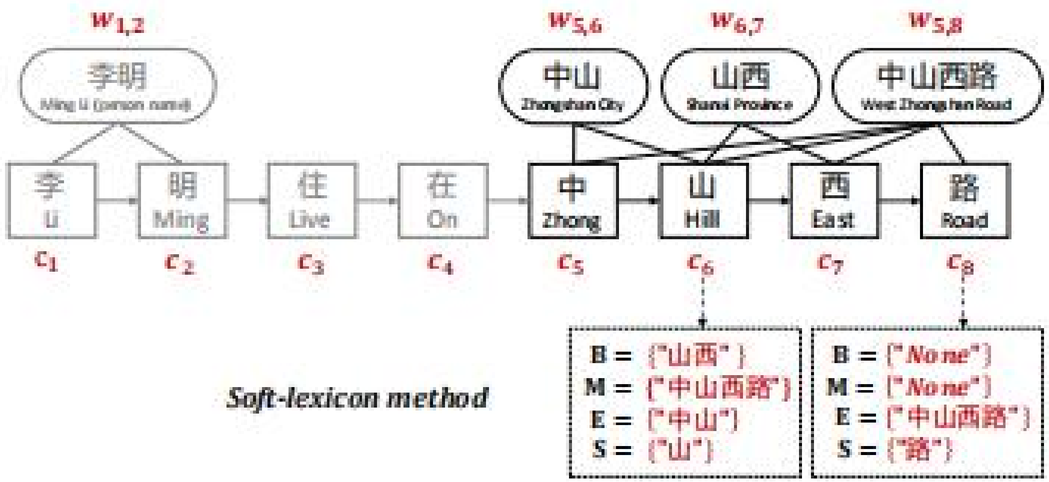


Figure 3: The SoftLexicon method.

Sequence Modeling Layer： BiLSTM、CNN、Transformer

Label Inference Layer： CRF

Datasets	**Type**	**Train**	**Dev**	**Test**
OntoNotes	Sentence	15.7k	4.3k	4.3k
	Char	491.9k	200.5k	208.1k
MSRA	Sentence	46.4k	-	4.4k
	Char	2169.9k	-	172.6k
Weibo	Sentence	1.4k	0.27k	0.27k
	Char	73.8k	14.5	14.8k
Resume	Sentence	3.8k	0.46	0.48k
	Char	124.1k	13.9k	15.1k

Table 1: Statistics of datasets.

Experiments

Models	OntoNotes	MSRA	Weibo	Resume
Lattice-LSTM	1×	1×	1×	1×
LR-CNN (Gui et al., 2019)	2.23×	1.57×	2.41×	1.44×
BERT-tagger	2.56×	2.55×	4.45×	3.12×
BERT + LSTM + CRF	2.77×	2.32×	2.84×	2.38×
SoftLexicon (LSTM)	6.15×	5.78×	6.10×	6.13×
SoftLexicon (LSTM) + bichar	6.08×	5.95×	5.91×	6.45×
SoftLexicon (LSTM) + BERT	2.74×	2.33×	2.85×	2.32×

Table 2: Inference speed (average sentences per second, the larger the better) of our method with LSTM layer compared with Lattice-LSTM, LR-CNN and BERT.

Input	Models	P	R	F1
Gold seg	Yang et al., 2016	65.59	71.84	68.57
	Yang et al., 2016*†	72.98	**80.15**	**76.40**
	Che et al., 2013*	77.71	72.51	75.02
	Wang et al., 2013*	76.43	72.32	74.32
	Word-based (LSTM)	76.66	63.60	69.52
	+ char + bichar	**78.62**	73.13	75.77
Auto seg	Word-based (LSTM)	72.84	59.72	65.63
	+ char + bichar	73.36	70.12	71.70
No seg	Char-based (LSTM)	68.79	60.35	64.30
	+ bichar + softword	74.36	69.43	71.89
	+ ExSoftword	69.90	66.46	68.13
	+ bichar + ExSoftword	73.80	71.05	72.40
	Lattice-LSTM	76.35	71.56	73.88
	LR-CNN (Gui et al., 2019)	76.40	72.60	74.45
	SoftLexicon (LSTM)	77.28	74.07	75.64
	SoftLexicon (LSTM) + bichar	77.13	**75.22**	**76.16**
	BERT-Tagger	76.01	79.96	77.93
	BERT + LSTM + CRF	81.99	81.65	81.82
	SoftLexicon (LSTM) + BERT	**83.41**	**82.21**	**82.81**

Table 3: Performance on OntoNotes. A model followed by (LSTM) (e.g., Proposed (LSTM)) indicates that its sequence modeling layer is LSTM-based.

Models	P	R	F1
Chen et al., 2006	91.22	81.71	86.20
Zhang et al. 2006*	92.20	90.18	91.18
Zhou et al. 2013	91.86	88.75	90.28
Lu et al. 2016	-	-	87.94
Dong et al. 2016	91.28	90.62	90.95
Char-based (LSTM)	90.74	86.96	88.81
+ bichar+softword	92.97	90.80	91.87
+ ExSoftword	90.77	87.23	88.97
+ bichar+ExSoftword	93.21	91.57	92.38
Lattice-LSTM	93.57	92.79	93.18
LR-CNN (Gui et al., 2019)	94.50	92.93	93.71
SoftLexicon (LSTM)	94.63	92.70	93.66
SoftLexicon (LSTM) + bichar	**94.73**	**93.40**	**94.06**
BERT-Tagger	93.40	94.12	93.76
BERT + LSTM + CRF	95.06	94.61	94.83
SoftLexicon (LSTM) + BERT	**95.75**	**95.10**	**95.42**

Table 4: Performance on MSRA.

Models	NE	NM	Overall
Peng and Dredze, 2015	51.96	61.05	56.05
Peng and Dredze, 2016*	**55.28**	**62.97**	**58.99**
He and Sun, 2017a	50.60	59.32	54.82
He and Sun, 2017b*	54.50	62.17	58.23
Char-based (LSTM)	46.11	55.29	52.77
+ bichar+softword	50.55	60.11	56.75
+ ExSoftword	44.65	55.19	52.42
+ bichar+ExSoftword	58.93	53.38	56.02
Lattice-LSTM	53.04	62.25	58.79
LR-CNN (Gui et al., 2019)	57.14	**66.67**	59.92
SoftLexicon (LSTM)	**59.08**	62.22	**61.42**
SoftLexicon (LSTM) + bichar	58.12	64.20	59.81
BERT-Tagger	65.77	62.05	63.80
BERT + LSTM + CRF	69.65	64.62	67.33
SoftLexicon (LSTM) + BERT	**70.94**	**67.02**	**70.50**

Table 5: Performance on Weibo. NE, NM and Overall denote F1 scores for named entities, nominal entities (excluding named entities) and both, respectively.

Models	P	R	F1
Word-based (LSTM)	93.72	93.44	93.58
+char+bichar	94.07	94.42	94.24
Char-based (LSTM)	93.66	93.31	93.48
+ bichar+softword	94.53	94.29	94.41
+ ExSoftword	95.29	94.42	94.85
+ bichar+ExSoftword	**96.14**	94.72	95.43
Lattice-LSTM	94.81	94.11	94.46
LR-CNN (Gui et al., 2019)	95.37	94.84	95.11
SoftLexicon (LSTM)	95.30	95.77	95.53
SoftLexicon (LSTM) + bichar	95.71	95.77	**95.74**
BERT-Tagger	94.87	**96.50**	95.68
BERT + LSTM + CRF	95.75	95.28	95.51
SoftLexicon (LSTM) + BERT	**96.08**	96.13	**96.11**

Table 6: Performance on Resume.

Models	OntoNotes	MSRA	Weibo	Resume
SoftLexicon (LSTM)	75.64	93.66	61.42	95.53
ExSoftword (CNN)	68.11	90.02	53.93	94.49
SoftLexicon (CNN)	**74.08**	**92.19**	**59.65**	**95.02**
ExSoftword (Transformer)	64.29	86.29	52.86	93.78
SoftLexicon (Transformer)	**71.21**	**90.48**	**61.04**	**94.59**

Table 7: F1 score with different implementations of the sequence modeling layer. ExSoftword is the shorthand of Char-based+bichar+ExSoftword.

Models	OntoNotes	MSRA	Weibo	Resume
SoftLexicon (LSTM)	75.64	93.66	61.42	95.53
- "M" group	75.06	93.09	58.13	94.72
- Distinction	70.29	92.08	54.85	94.30
- Weighted pooling	72.57	92.76	57.72	95.33
- Overall weighting	74.28	93.16	59.55	94.92

Table 8: An ablation study of the proposed model.