Joint Extraction of Entities and Relations Based on a Novel Decomposition Strategy

Bowen Yu[1,2] and **Zhenyu Zhang**[1,2] and **Xiaobo Shu**[1,2] and **Tingwen Liu**[1*]
Yubin Wang[1,2] and **Bin Wang**[3] and **Sujian Li**[4]

Main Work

- First decompose the joint extraction task into two **interrelated subtasks**, namely **HE extraction** and **TER extraction**
- Next,these two subtasks are further deconstructed into several sequence labeling problems based on our proposed span-based tagging scheme.
- The sequence labeling problems are conveniently solved by a **hierarchical boundary tagger(HBT) and a multi-span decoding algorithm**
- Experimental results show that our method outperforms previous work by 5.2%, 5.9% and 21.5% (F1 score), achieving a new state-of-the-art on three public datasets.

Tagging

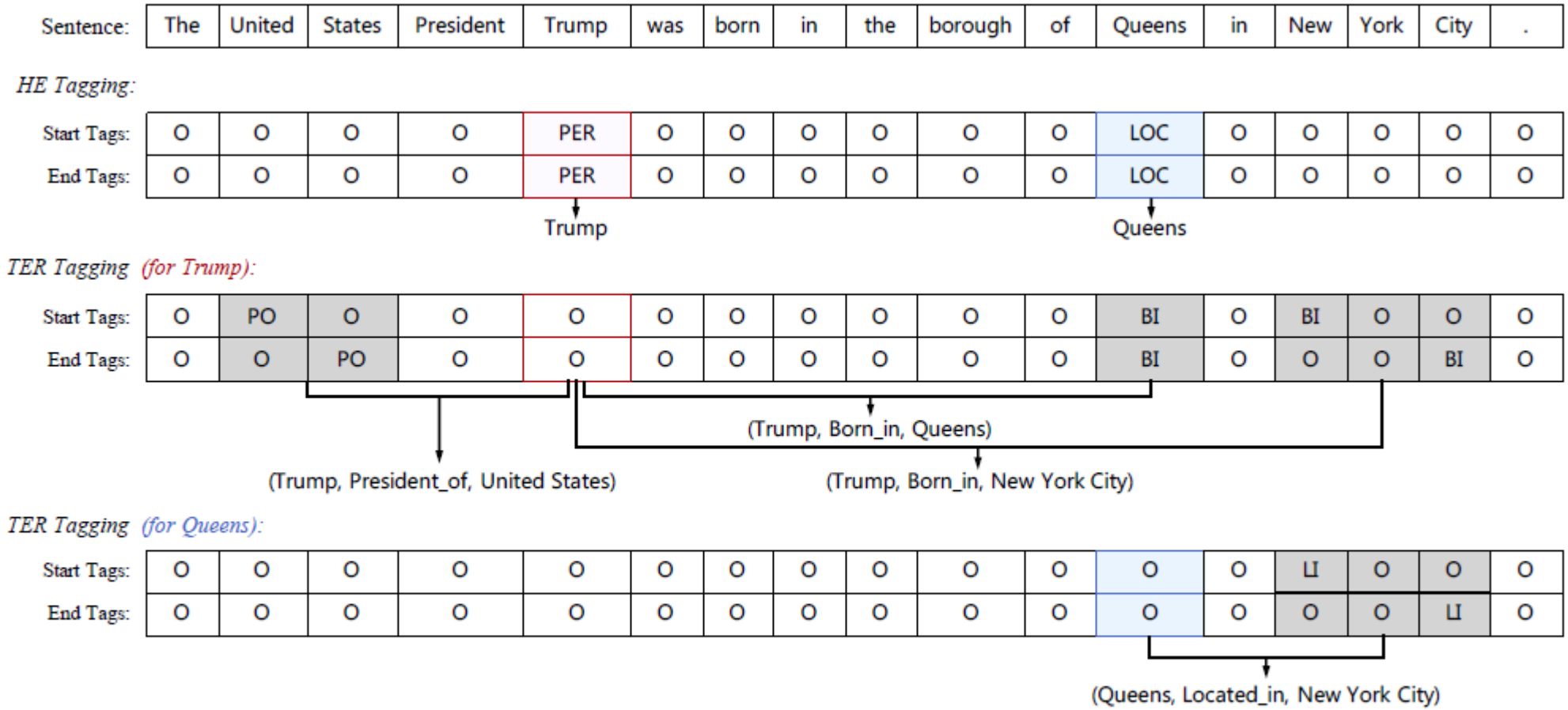


Figure 1. An example of our tagging scheme. *PER* is short for entity type *PERSON*, *LOC* is short for *LOCATION*, *PO* is short for relation type *President_of*, *BI* is short for *Born_in*, and *LI* is short for *Located_in*.

Model

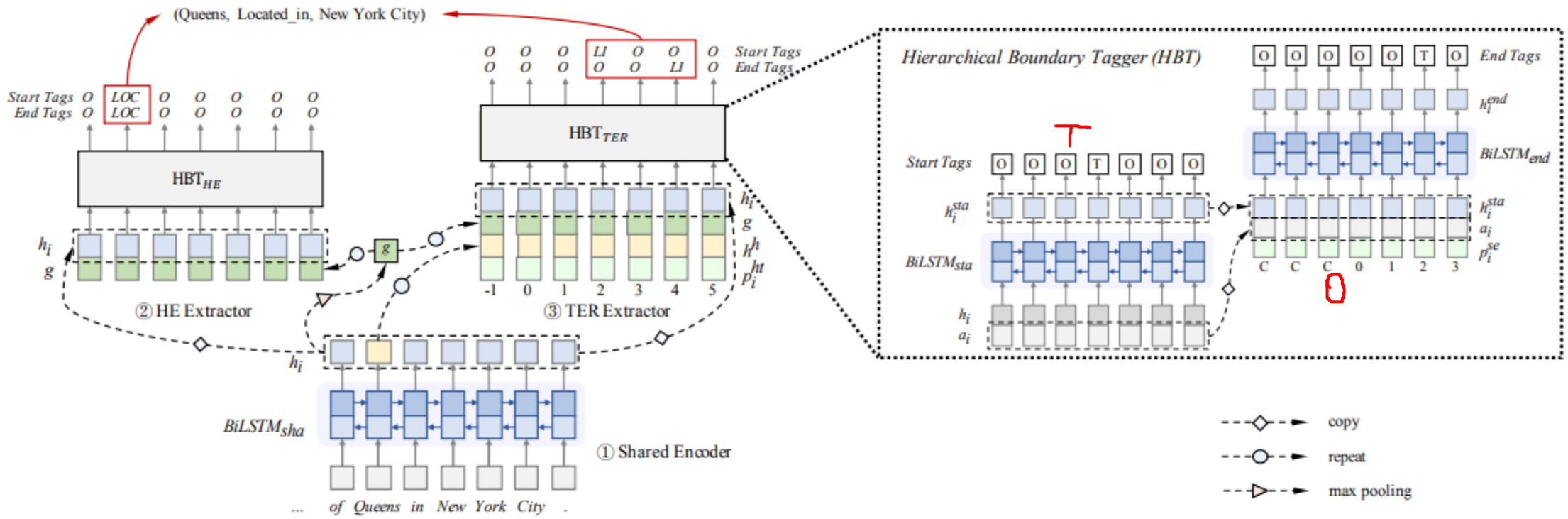


Figure 2. An illustration of our model. The left panel is an overview of our joint extraction system, and the right panel shows the detailed structure of our sequence tagger HBT. Here, "*Queens*" is extracted by the HE extractor, then its hidden state in the shared encoder is marked as the yellow box and entered into the TER extractor as prior knowledge.

Hierarchical Boundary Tagger

- START Tagger

$$\mathbf{h}_i^{sta} = \text{BiLSTM}_{sta}([\mathbf{h}_i; \mathbf{a}_i]) \quad (2)$$

$$P(y_i^{sta}) = \text{Softmax}(\mathbf{W}^{sta} \cdot \mathbf{h}_i^{sta} + \mathbf{b}^{sta}) \quad (3)$$

$$\text{sta\_tag}(x_i) = \arg\max_k P(y_i^{sta} = k) \quad (4)$$

- End Tagger

$$\mathbf{h}_i^{end} = \text{BiLSTM}_{end}([\mathbf{h}_i^{sta}; \mathbf{a}_i; \mathbf{p}_i^{se}]) \quad (5)$$

$$P(y_i^{end}) = \text{Softmax}(\mathbf{W}^{end} \cdot \mathbf{h}_i^{end} + \mathbf{b}^{end}) \quad (6)$$

$$\text{end\_tag}(x_i) = \arg\max_k P(y_i^{end} = k) \quad (7)$$

- Position Encoding

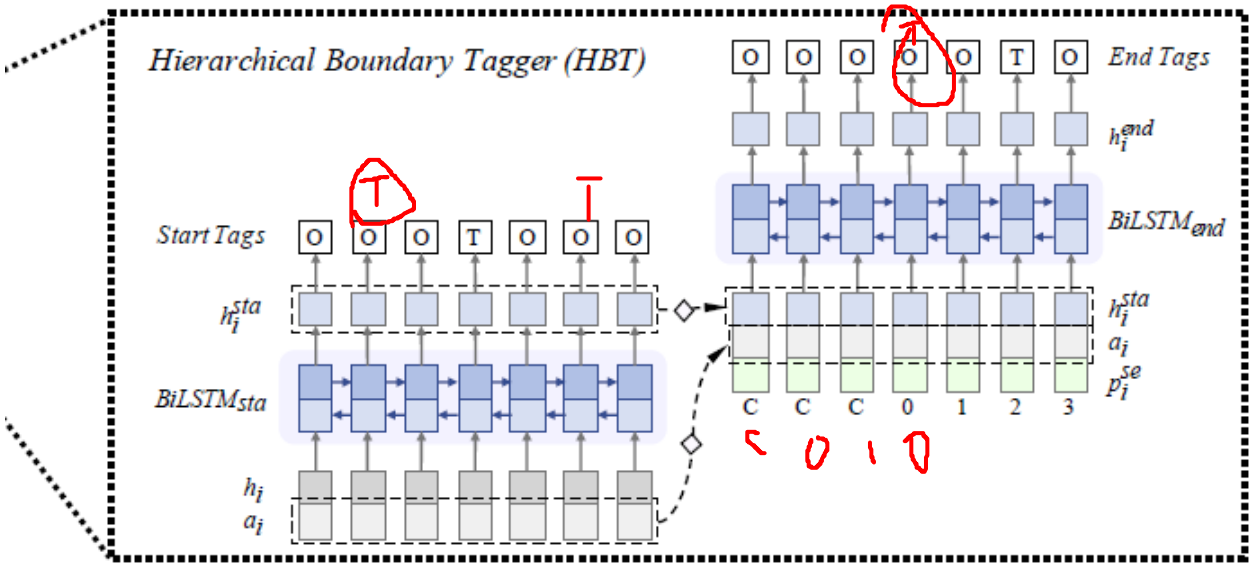
$$p_i^{se} = \begin{cases} i - s^*, & \text{if } s^* \text{ exists} \\ C, & \text{otherwise} \end{cases}$$

Multi-span decoding algorithm

Algorithm 1 Multi-span decoding

Input:
S, C
S denotes the input sentence
C is a predefined distance constant
Output:
$\{(e_j, tag_j)\}_{j=1}^{m}$,
e_j denotes the j-th extracted target and tag_j is the type tag

Define $n \leftarrow$ Sentence Length
Initialize $\mathbf{R} \leftarrow \{\}$
Initialize $s^* \leftarrow 0$
Initialize p^{se} as a list of length n with default value C
Obtain sta_tag(S) by Eq. 4
for $idx \leftarrow 1$ to n **do**
    if sta_tag $(S)[idx] \neq$ "O" **then**
        $s^* \leftarrow idx$
    if $s^* > 0$ **then**
        $p^{se}[idx] \leftarrow idx - s^*$
Obtain $\mathbf{p}^{se}$ by transforming p^{se} into matrix
Obtain end_tag(S) by Eq. 7
for $idx_s \leftarrow 1$ to n **do**
    if sta_tag$(S)[idx_s] \neq$ "O" **then**
        for $idx_e \leftarrow idx_s$ to n **do**
            if end_tag$(S)[idx_e]$ = sta_tag$(S)[idx_s]$ **then**
                $e \leftarrow S[idx_s : idx_e]$
                $tag \leftarrow$end_tag$(S)[idx_e]$
                $\mathbf{R} \leftarrow \mathbf{R} \cup \{(e, tag)\}$
                Break
return $\mathbf{R}$



Result

Table 3. Comparison of test-time speed. Bat/s refers to the number of batches can be processed per second.

Model	NYT-single	NYT-multi	WebNLG
ETL-BIES	10.9 Bat/s	11.2 Bat/s	6.3 Bat/s
ETL-Span	26.1 Bat/s	25.6 Bat/s	23.5 Bat/s

Table 4. An ablation study of ETL-Span on the NYT-multi dev set.

Model	Precision	Recall	F1
ETL-Span	**86.5%**	**73.5%**	**79.5%**
– Char embedding	83.1%	71.2%	76.7%
– Position embedding $\mathbf{p}^{ht}$	81.9%	70.3%	75.7%
– Hierarchical tagging	84.6%	70.7%	77.0%
– Head-entity type tagging	85.8%	72.2%	78.4%
– Joint learning	80.4%	68.9%	74.2%

Result

Table 3. Comparison of test-time speed. Bat/s refers to the number of batches can be processed per second.

Model	NYT-single	NYT-multi	WebNLG
ETL-BIES	10.9 Bat/s	11.2 Bat/s	6.3 Bat/s
ETL-Span	26.1 Bat/s	25.6 Bat/s	23.5 Bat/s

Table 4. An ablation study of ETL-Span on the NYT-multi dev set.

Model	Precision	Recall	F1
ETL-Span	**86.5%**	**73.5%**	**79.5%**
– Char embedding	83.1%	71.2%	76.7%
– Position embedding $\mathbf{p}^{ht}$	81.9%	70.3%	75.7%
– Hierarchical tagging	84.6%	70.7%	77.0%
– Head-entity type tagging	85.8%	72.2%	78.4%
– Joint learning	80.4%	68.9%	74.2%

Result

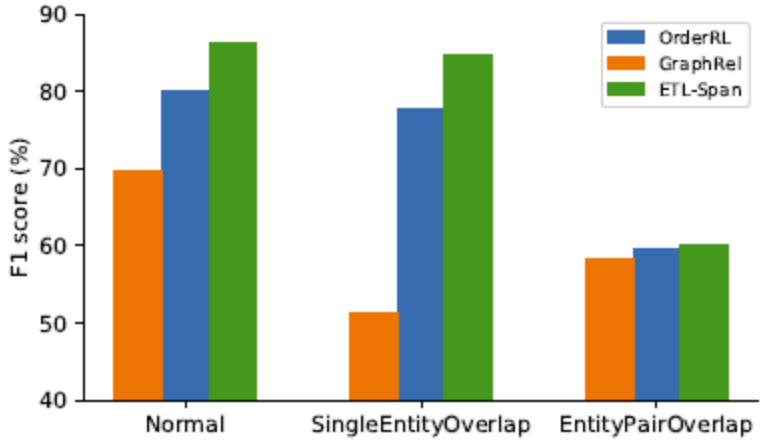


Figure 3. F1 score by overlapping category on the NYT-multi test set.

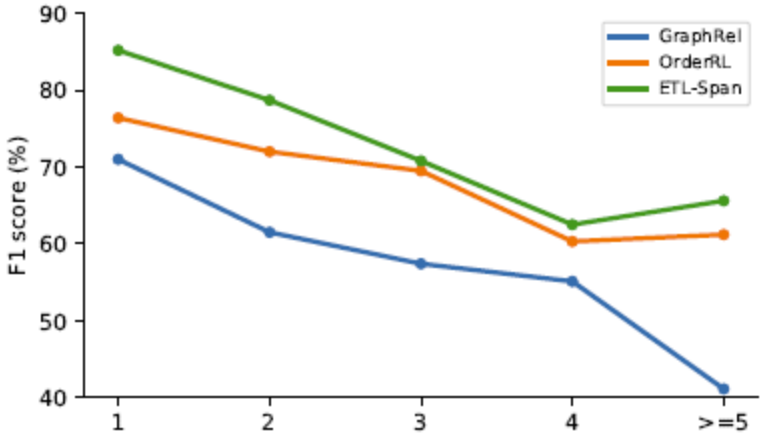


Figure 4. F1 score by sentence triplet count on the NYT-multi test set.