Nested Named Entity Recognition with Partially Observed TreeCRFs

Yao Fu¹, Chuanqi Tan²*, Mosha Chen², Songfang Huang², Fei Huang²

*Equal Contribution

¹University of Edinburgh ²Alibaba Group
US officials publicly declined to confirm any of North Korea’s announcements with US State Department spokesman Richard Boucher…

Formulation: constituency parsing with partially-observed trees
A model that properly tackles nested NER

- Jointly model the observed and the latent
- Good performance

A simple and efficient model

- Avoid re-inventing neural architectures
- Batchified and tensorized computation
Approach

TreeCRFs with partially marginalization
• But inefficient, a major drawback in previous literature

No clear batchification:
Difference sentences, different tree structures

O(n^3) complexity for each sentence

This work focuses on these two problems
Our contribution

No clear batchification:
Difference sentences, different tree structures

→ Batchfied likelihood evaluation

$O(n^3)$ complexity for each sentence

$O(n \log n)$ complexity

We propose efficient partial marginalization with MASKED INSIDE algorithm
Model

\[ e_1, \ldots, e_n = \text{FF}(\text{Enc}(x)) \]

\[ s_{ijk} = e_i^T U_k^{(1)} e_j + (e_i + e_j)^T U_k^{(2)} + b_k \]

Standard Biaffine Scoring (Dozat and Manning 16)

\[
\log p(T|x) = s(T) - \log Z \\
s(T) = \log \sum_{\tilde{T} \in \tilde{T}} \exp(s(\tilde{T}))
\]

Training maximize the likelihood for a partial tree …

… by summing over all compatible full trees
This summation can be done with an Inside-styled DP.

But no clear batchification for sentences with different partial trees because DP graph is different.
To perform partial marginalization.

Left: Observed partial tree
Right: An example full tree realized from left (other possible full trees exist)
Example Observed
Example
Observed
Rejected because of overlapped spans
Rejected because of overlapped spans
Latent nodes can be realized in a full tree
Tagging for Latent

Latent nodes cannot have observed tags: PER, LOC, ORG …
Latent nodes can only be labeled as: LATENT_1, LATENT_2 …

During training we marginalize all possible latent tags
During Inference we drop entities with latent tags to get partial trees
Why we care about nodes with difference types?

Different operations for different nodes in this DP summation

\[ s(T) = \log \sum_{\tilde{T} \in \tilde{T}} \exp(s(\tilde{T})) \]
Likelihood Evaluation for Observed

\[ \beta_{ijk} = \exp(s_{ijk}) \ldots \]
Rejection for
Rejected

\[ \beta_{ijk} = 0 \]
Marginalization for the Latent

\[ \beta_{ijk} = \sum_{k \in \mathcal{L}_l} \exp(s_{ijk}) \cdots \]
Unify Operations with Mask

\[ \beta_{ijk} = \exp(s_{ijk}) \ldots \]

Likelihood Evaluation

\[ \beta_{ijk} = 0 \]

Rejectio

n

Partial Marginalization

\[ \beta_{ijk} = \sum_{k \in \mathcal{L}} \exp(s_{ijk}) \ldots \]

A uniform masked summation

\[ m_{ijk} = 1, m_{ijk'} = 0 \]

\( k \) observed tag, \( k' \) all other tags

Likelihood Evaluation

\[ \forall k, m_{ijk} = 0 \]

\( k \) all tags

Rejectio

n

Partial Marginalization

\[ \forall k_1, m_{ijk_1} = 1, \forall k_2, m_{ijk_2} = 0 \]

\( k_1 \) latent tags, \( k_2 \) observed tags
Compared with the original

Inside

\[ \beta_{ijk} = \sum_k \exp(s_{ijk}) \ldots \]

Sum over all possible trees

Only difference is the mask term

\[ \beta_{ijk} = \sum_k m_{ijk} \cdot \exp(s_{ijk}) \ldots \]

Sum over all possible full trees compatible with a partial tree
One single line change of the original Inside algorithm
Unify the DP graph for sentences with different partial trees

Reuse recent efficient bachification and tensorization works for the original Inside. We use Torch-Struct
At the end of the day, turn a conceptually complicated, practically inefficient partial marginalization algorithm into a simple and efficient Masked Inside

Algorithm 2 INSIDE FOR PARTIAL MARGINALIZATION

1: Input: Scores $s$, partial tree $T$ and its corresponding $\bar{T}$
2: for $i \leftarrow 1$ to $n$ do
3:   if $\bar{T}[i, i] = \bullet$ then \texttt{// Observed leaf}
4:   $\exists k \in L_0, T_{i,k} = 1, \beta[i, i, k] = \exp(s_{i,k})$
5:   $\forall m \neq k, \beta[i, i, m] = 0$
6:   else if $\bar{T}[i, i] = \square$ then \texttt{// Latent leaf}
7:   $\forall k \in L_0, \beta[i, i, k] = 0$
8:   $\forall k \in L_1, \beta[i, i, k] = \exp(s_{i,k})$
9: for $d \leftarrow 1$ to $n - 1$ do
10: for $i \leftarrow 1$ to $n - d$ do
11:   $j = i + d$
12:   if $\bar{T}[i, j] = \bullet$ then \texttt{// Observed}
13:   $\exists k \in L_0, T_{i,j} = 1$
14:   $\beta[i, j, k] = \exp(s_{i,j})$
15:   $\sum_{l=1}^{i-1} \sum_{k_1, k_2 \in L} \beta[i, l, k_1] \beta[l + 1, j, k_2]$
16:   $\forall m \neq k, \beta[i, j, m] = 0$
17:   else if $\bar{T}[i, j] = \square$ then \texttt{// Latent}
18:   $\forall k \in L_0, \beta[i, j, k] = 0$
19:   $\forall k \in L_1, \beta[i, j, k] = \exp(s_{i,j})$
20:   $\sum_{l=1}^{i-1} \sum_{k_1, k_2 \in L} \beta[i, l, k_1] \beta[l + 1, j, k_2]$
21:   else if $\bar{T}[i, j] = \circ$ then \texttt{// Rejected}
22:   $\forall k \in L_0, \beta[i, j, k] = 0$
23: if $\bar{T}[1, n] = \bullet$ then \texttt{// Observed root}
24: $\exists k \in L_0, T_{1,k} = 1$. Return $s(T) = \beta[1, n, k]$
25: else if $\bar{T}[1, n] = \square$ then \texttt{// Latent root}
26: Return $s(T) = \log(\sum_{k \in \mathcal{L}_1} \beta[1, n, k])$

$s(T) = \text{MASKEDINSIDE}(s, M)$

$= \text{INSIDE}(\log M + s)$
<table>
<thead>
<tr>
<th>Model</th>
<th>ACE2004 P</th>
<th>ACE2004 R</th>
<th>ACE2004 F1</th>
<th>ACE2005 P</th>
<th>ACE2005 R</th>
<th>ACE2005 F1</th>
<th>GENIA F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM-CRF (Lample et al. 2016)</td>
<td>71.3</td>
<td>50.5</td>
<td>58.3</td>
<td>70.3</td>
<td>55.7</td>
<td>62.2</td>
<td>75.2</td>
</tr>
<tr>
<td>FOFE(c=6) (Xu et al. 2017)</td>
<td>68.2</td>
<td>54.3</td>
<td>60.5</td>
<td>76.5</td>
<td>66.3</td>
<td>71.0</td>
<td>75.4</td>
</tr>
<tr>
<td>Transition (Wang et al. 2018)</td>
<td>74.9</td>
<td>71.8</td>
<td>73.3</td>
<td>74.5</td>
<td>71.5</td>
<td>73.0</td>
<td>78.0</td>
</tr>
<tr>
<td>Cascaded-CRF (Ju et al. 2018)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>74.2</td>
<td>70.3</td>
<td>72.2</td>
<td>78.5</td>
</tr>
<tr>
<td>SH(c=n) (Wang and Lu 2018)</td>
<td>77.7</td>
<td>72.1</td>
<td>74.5</td>
<td>76.8</td>
<td>72.3</td>
<td>74.5</td>
<td>77.0</td>
</tr>
<tr>
<td>ML (Fisher and Vlachos 2019)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>75.1</td>
<td>74.1</td>
<td>74.6</td>
<td>-</td>
</tr>
<tr>
<td>BENSNC (Tan et al. 2020)</td>
<td>78.1</td>
<td>72.8</td>
<td>75.3</td>
<td>77.1</td>
<td>74.2</td>
<td>75.6</td>
<td>78.9</td>
</tr>
<tr>
<td>Pyramid (Jue et al. 2020)</td>
<td>81.1</td>
<td>79.4</td>
<td>80.3</td>
<td>80.0</td>
<td>78.9</td>
<td>79.4</td>
<td>78.6</td>
</tr>
<tr>
<td>with Pretrained LM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MGNER (ELMo) (Xia et al. 2019)</td>
<td>81.7</td>
<td>77.4</td>
<td>79.5</td>
<td>79.0</td>
<td>77.3</td>
<td>78.2</td>
<td>-</td>
</tr>
<tr>
<td>ML (ELMo) (Fisher and Vlachos 2019)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>79.7</td>
<td>78.0</td>
<td>78.9</td>
<td>-</td>
</tr>
<tr>
<td>ML (BERT) (Fisher and Vlachos 2019)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>82.7</td>
<td>82.1</td>
<td>82.4</td>
<td>-</td>
</tr>
<tr>
<td>Seq2seq (Straková, Straka, and Hajic 2019)</td>
<td>-</td>
<td>-</td>
<td>84.3</td>
<td>-</td>
<td>83.4</td>
<td>-</td>
<td>78.2</td>
</tr>
<tr>
<td>BENSNC (BERT) (Tan et al. 2020)</td>
<td>85.8</td>
<td>84.8</td>
<td>85.3</td>
<td>83.8</td>
<td>83.9</td>
<td>83.9</td>
<td>79.2</td>
</tr>
<tr>
<td>Pyramid (BERT) (Jue et al. 2020)</td>
<td>86.1</td>
<td>86.5</td>
<td>86.3</td>
<td>84.0</td>
<td>85.4</td>
<td>84.7</td>
<td>79.5</td>
</tr>
<tr>
<td>with Additional Supervision</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DYGIE (Luan et al. 2019)</td>
<td>-</td>
<td>-</td>
<td>84.7</td>
<td>-</td>
<td>82.9</td>
<td>-</td>
<td>76.2</td>
</tr>
<tr>
<td>Yu, Bohnet, and Poesio (2020)</td>
<td>87.3</td>
<td>86.0</td>
<td>86.7</td>
<td>85.2</td>
<td>85.6</td>
<td>85.4</td>
<td>81.8</td>
</tr>
<tr>
<td>BERT-MRC (Li et al. 2020)</td>
<td>85.0</td>
<td>86.3</td>
<td>86.0</td>
<td>87.2</td>
<td>86.6</td>
<td>86.9</td>
<td>85.2</td>
</tr>
<tr>
<td>PO-TreeCRFs (ours)</td>
<td>86.7</td>
<td>86.5</td>
<td>86.6</td>
<td>84.5</td>
<td>86.4</td>
<td>85.4</td>
<td>78.2</td>
</tr>
<tr>
<td>PO-TreeCRFs Ablation Study</td>
<td>±0.4</td>
<td>±0.4</td>
<td>±0.3</td>
<td>±0.4</td>
<td>±0.2</td>
<td>±0.1</td>
<td>±0.7</td>
</tr>
</tbody>
</table>

Table 2: Main results and ablation studies on three datasets. We report the average scores of 5 runs for main results.
<table>
<thead>
<tr>
<th>Method</th>
<th>Inside (Vanilla)</th>
<th>MASKED INSIDE</th>
<th>Biaffine</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU Time</td>
<td>14m58s</td>
<td>3m20s</td>
<td>2m27s</td>
</tr>
<tr>
<td>CPU Time</td>
<td>2h5m</td>
<td>24m</td>
<td>22m10s</td>
</tr>
<tr>
<td>Complexity</td>
<td>$O(n^3)$</td>
<td>$O(n \log n)$</td>
<td>$O(1)$</td>
</tr>
</tbody>
</table>
A method using partially-observed TreeCRFs for nested NER

Key contribution is about efficient inference
- Construct masks to unify different inference operations
- Replace original partial marginalization algorithm with Masked Inside algorithm
Conclusions

Code: https://github.com/FranxYao/Partially-Observed-TreeCRFs

Any questions, please contact:
chuanqi.tcq@alibaba-inc.com
chenmosha.cms@alibaba-inc.com

Thanks!