Jointly Predicting **Predicates** and **Arguments** in Neural Semantic Role Labeling

University of Washington

*Now at Google  +Now at Facebook AI Research
Semantic Role Labeling (SRL)

- Find out “who did what to whom” in text
- Capture predicate-argument structures
Semantic Role Labeling (SRL)

• Find out “**who did what to whom**” in text
• Capture **predicate-argument** structures
SRL as BIO Tagging

Input1

Many tourists visit Disney to meet their favorite cartoon characters

Output1

ARG0
B-A0
I-A0
ARG1
B-V
B-A1
AM-PRP
B-AM-PRP
I-AM-PRP
I-AM-PRP
I-AM-PRP
I-AM-PRP
I-AM-PRP

Needs target predicate as input!
(Prior works typically used gold predicates)

Collobert et al., 2011,
Zhou and Xu, 2015,
He et. al, 2017,
inter alia
SRL as BIO Tagging

Output1

ARG0  V  ARG1  AM-PRP
B-A0  B-V  B-A1  B-AM-PRP  B-AM-PRP  B-AM-PRP  B-AM-PRP  B-AM-PRP  B-AM-PRP

Input1

Many  tourists  visit  Disney  to  meet  their  favorite  cartoon  characters

Output2

ARG0  V  ARG1

Input2

Many  tourists  visit  Disney  to  meet  their  favorite  cartoon  characters

Needs to re-run the tagger for each predicate
Many tourists visit Disney to meet their favorite cartoon characters.
SRL as Predicting Word-Span Relations

**Advantages:**
* Jointly predict predicates
* Span-level features
  (similar to Punyakanok08, FitzGerald15, inter alia)

**Challenge:**
* Too many possible edges ($n^2$ argument spans x n predicates)
Our Model
Our Model: Overview

[Diagram showing the process of processing an input sentence through Word & Char Embeddings and Highway BiLSTMs.]

Input sentence: Many tourists visit Disney to meet their favorite cartoon characters

No predicate input!
Our Model: Overview

(1) Construct span representations for all $n^2$ spans!
Our Model: Overview

(1) Construct span representations for all $n^2$ spans!

(2) Local classifier over labels (including NULL) for all possible (predicate, argument) pairs

Many tourists visit Disney to meet their favorite cartoon characters
Many tourists visit Disney to meet their favorite cartoon characters.
Many tourists visit Disney to meet their favorite cartoon characters.

(Same as Lee et al., 2017)
(1) Span Representations

\[ [\text{BiLSTM}(w_1 : w_n)_{\text{START}}, \text{BiLSTM}(w_1 : w_n)_{\text{END}}] \]

(2) Local Label Classifiers

(3) Span Pruning

Input sentence: Many tourists visit Disney to meet their favorite cartoon characters

(Same as Lee et al., 2017)
Many tourists visit Disney to meet their favorite cartoon characters.

(Same as Lee et al., 2017)
Many tourists visit Disney to meet their favorite cartoon characters.
Many tourists visit Disney to meet their favorite cartoon characters.
Many tourists visit Disney to meet their favorite cartoon characters.

Disney

ARG0
ARG1
ARG2
...
AM-TMP
...
\( \epsilon \) (No Edge)
Many tourists visit Disney to meet their favorite cartoon characters.
\[ \phi(\text{pred}, \text{arg}, l) = \Phi_a(\text{arg}) + \Phi_p(\text{pred}) + \Phi_{\text{rel}}(\text{arg}, \text{pred}) \]
Many tourists meet

\[ \phi(p_{\text{pred}}, a_{\text{arg}}, l) = \Phi_a(a_{\text{arg}}) + \Phi_p(p_{\text{pred}}) + \Phi_{\text{rel}}^{(l)}(a_{\text{arg}}, p_{\text{pred}}) \]
\[ \phi(\text{pred, arg, } l) = \Phi_a(\text{arg}) + \Phi_p(\text{pred}) + \Phi_{\text{rel}}(\text{arg, pred}) \]

\[ \phi(\text{“Many tourists”, “meet”, } \epsilon) = 0 \]

\[ \Phi_{\text{rel}}(\text{ARG0})(\text{“Many tourists”, “meet”}) \]

\[ \Phi_{\text{rel}}(\text{ARG1})(\text{“Many tourists”, “meet”}) \]

\[ \Phi_a(\text{“Many tourists”}) \]

\[ \Phi_p(\text{“meet”}) \]

Many tourists

meet
(1) Span Representations

(2) Local Label Classifiers

(3) Span Pruning

O(n^2) arguments, O(n) predicates, → O(n^3) edges!

Many tourists visit Disney to meet their favorite cartoon characters
Many tourists visit Disney to meet their favorite cartoon characters

**Span Representation**

**Highway BiLSTMs**

**Word & Char Embeddings**

**Input sentence**

(1) Span Representations

(2) Local Label Classifiers

(3) Span Pruning

Only keep top $O(n)$ spans using their unary scores

$$\Phi_a(\text{“many tourists”}) = 2.5$$

$$\Phi_a(\text{“tourists visit Disney”}) = -0.8$$

...
Results & Analysis
End-to-End SRL Results

BIO-based, pipelined predicate ID

CoNL05 WSJ Test: He17 81.2, Ours 82.5, Ours+ELMo 79.8

CoNL05 Brown Test: He17 68.5, Ours 70.8

CoNLL2012 (OntoNotes): He17 (Ensemble) 76.8, Ours+ELMo 79.8
End-to-End SRL Results

With ELMo, over 3 points improvement over SotA ensemble!

*ELMo: Deep Contextualized Word Representations, Peters et al., 2018
## Span-based vs. BIO

<table>
<thead>
<tr>
<th></th>
<th>BIO (He17)</th>
<th>Span-based (this work)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Inputs</strong></td>
<td>(Sentence, Predicate)</td>
<td>Sentence</td>
</tr>
<tr>
<td><strong>Predicate Identification</strong></td>
<td>Pipelined</td>
<td>Joint</td>
</tr>
<tr>
<td><strong>Global Consistency</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Long-range Dependency</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Span-based vs. BIO

<table>
<thead>
<tr>
<th>Inputs</th>
<th>BIO (He17)</th>
<th>Span-based (this work)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sentence, Predicate</td>
<td>Sentence</td>
<td>Pipeline</td>
</tr>
<tr>
<td>Global Consistency</td>
<td>Joint</td>
<td></td>
</tr>
</tbody>
</table>

Due to the strong independence assumption we make.
Span-based vs. BIO

<table>
<thead>
<tr>
<th>Inputs</th>
<th>BIO (He17)</th>
<th>Span-based (this work)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sentence, Predicate</td>
<td>Sentence</td>
<td>Sentence</td>
</tr>
</tbody>
</table>

- **Due to the strong independence assumption we make.**

- **By allowing direct interaction between predicates and arguments**

Global Consistency

Long-range Dependency
Conclusion

• Joint prediction of predicates and arguments
Conclusion

- Joint prediction of predicates and arguments

- Our recipe:
  1. Contextualized span representations
  2. Local label classifiers
  3. Greedy span pruning
Conclusion

- Joint prediction of predicates and arguments

- Our recipe:
  1. Contextualized span representations
  2. Local label classifiers
  3. Greedy span pruning

- Future work: Improve global consistency, use span representations for downstream tasks, etc.
THANK YOU!

Code and pertained models: https://github.com/luheng/lsgn