Structured Attention Networks

Yoon Kim*  Carl Denton*  Luong Hoang  Alexander M. Rush

HarvardNLP
1. Deep Neural Networks for Text Processing and Generation

2. Attention Networks

3. Structured Attention Networks
   - Computational Challenges
   - Structured Attention In Practice

4. Conclusion and Future Work
1 Deep Neural Networks for Text Processing and Generation

2 Attention Networks

3 Structured Attention Networks
   - Computational Challenges
   - Structured Attention In Practice

4 Conclusion and Future Work
Pure Encoder-Decoder Network

Input (sentence, image, etc.)

Fixed-Size Encoder (MLP, RNN, CNN)

Encoder(input) \in \mathbb{R}^D

Decoder

Decoder(Encoder(input))
Pure Encoder-Decoder Network

Input (sentence, image, etc.)

\[ \text{Encoder}(\text{input}) \in \mathbb{R}^{D} \]

Decoder

\[ \text{Decoder}(\text{Encoder}(\text{input})) \]
Example: Neural Machine Translation (Sutskever et al., 2014)

Over the line!
Example: Neural Machine Translation (Sutskever et al., 2014)
Example: Neural Machine Translation (Sutskever et al., 2014)

Over
the
line
!
<s>
Example: Neural Machine Translation (Sutskever et al., 2014)
Example: Neural Machine Translation (Sutskever et al., 2014)

Over  the  line  !  <s>  Çizgiyi
Example: Neural Machine Translation (Sutskever et al., 2014)
Example: Neural Machine Translation (Sutskever et al., 2014)
Example: Neural Machine Translation (Sutskever et al., 2014)
Encoder-Decoder with Attention

- **Machine Translation** (Bahdanau et al., 2015; Luong et al., 2015)
- **Question Answering** (Hermann et al., 2015; Sukhbaatar et al., 2015)
- **Natural Language Inference** (Rocktäschel et al., 2016; Parikh et al., 2016)
- **Algorithm Learning** (Graves et al., 2014, 2016; Vinyals et al., 2015a)
- **Parsing** (Vinyals et al., 2015b)
- **Speech Recognition** (Chorowski et al., 2015; Chan et al., 2015)
- **Summarization** (Rush et al., 2015)
- **Caption Generation** (Xu et al., 2015)
- and more...
Neural Attention

Input (sentence, image, etc.)

Memory-Bank Encoder (MLP, RNN, CNN)

Encoder(input) = \( x_1, x_2, \ldots, x_T \)

Attention Distribution

"where"

Context Vector

"what"

Decoder
Neural Attention

Input (sentence, image, etc.)

Memory-Bank Encoder (MLP, RNN, CNN)

Encoder(input) = $x_1, x_2, \ldots, x_T$

Attention Distribution Context Vector

Attention Distribution

Decoder
Neural Attention

Input (sentence, image, etc.)

\[
\text{Memory-Bank Encoder (MLP, RNN, CNN)} \rightarrow \text{Encoder(input)} = x_1, x_2, \ldots, x_T
\]

Attention Distribution  
Context Vector

“where”  
“what”

Decoder
Attention-based Neural Machine Translation (Bahdanau et al., 2015)
Attention-based Neural Machine Translation (Bahdanau et al., 2015)
Attention-based Neural Machine Translation (Bahdanau et al., 2015)
Attention-based Neural Machine Translation (Bahdanau et al., 2015)
Attention-based Neural Machine Translation (Bahdanau et al., 2015)
Attention-based Neural Machine Translation (Bahdanau et al., 2015)
Attention-based Neural Machine Translation (Bahdanau et al., 2015)
Attention-based Neural Machine Translation (Bahdanau et al., 2015)
Attention-based Neural Machine Translation (Bahdanau et al., 2015)
Attention-based Neural Machine Translation (Bahdanau et al., 2015)
Question Answering (Sukhbaatar et al., 2015)

Greg is a frog
Brian is a rhino
Lily is a rhino
Greg is green
Brian is white
John is a frog
<table>
<thead>
<tr>
<th>Statement</th>
<th>Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greg is a frog</td>
<td><img src="image1.png" alt="Diagram" /></td>
</tr>
<tr>
<td>Brian is a rhino</td>
<td><img src="image2.png" alt="Diagram" /></td>
</tr>
<tr>
<td>Lily is a rhino</td>
<td><img src="image3.png" alt="Diagram" /></td>
</tr>
<tr>
<td>Greg is green</td>
<td><img src="image4.png" alt="Diagram" /></td>
</tr>
<tr>
<td>Brian is white</td>
<td><img src="image5.png" alt="Diagram" /></td>
</tr>
<tr>
<td>John is a frog</td>
<td><img src="image6.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>
Question Answering (Sukhbaatar et al., 2015)

What color is Lily?

- Greg is a frog
- Brian is a rhino
- Lily is a rhino
- Greg is green
- Brian is white
- John is a frog
Question Answering (Sukhbaatar et al., 2015)
Question Answering (Sukhbaatar et al., 2015)

What color is Lily?

- Greg is a frog
- Brian is a rhino
- Lily is a rhino
- Greg is green
- Brian is white
- John is a frog
Question Answering (Sukhbaatar et al., 2015)

What color is Lily?

- Greg is a frog
- Brian is a rhino
- Lily is a rhino
- Greg is green
- Brian is white
- John is a frog
Question Answering (Sukhbaatar et al., 2015)

What color is Lily?

Greg is a frog
Brian is a rhino
Lily is a rhino
Greg is green
Brian is white
John is a frog
Question Answering (Sukhbaatar et al., 2015)

What color is Lily?

Greg is a frog
Brian is a rhino
Lily is a rhino
Greg is green
Brian is white
John is a frog
Question Answering (Sukhbaatar et al., 2015)

What color is Lily?
Question Answering (Sukhbaatar et al., 2015)

What color is Lily?
Question Answering (Sukhbaatar et al., 2015)

What color is Lily?

Greg is a frog
Brian is a rhino
Lily is a rhino
Greg is green
Brian is white
John is a frog

white
Other Applications: Image Captioning (Xu et al., 2015)

(b) A woman is throwing a frisbee in a park.
Other Applications: Speech Recognition (Chan et al., 2015)
Applications From HarvardNLP: Summarization (Rush et al., 2015)

<table>
<thead>
<tr>
<th></th>
<th>&lt;s&gt;</th>
<th>russia</th>
<th>calls</th>
<th>for</th>
<th>joint</th>
<th>front</th>
<th>against</th>
<th>terrorism</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;s&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>russia</td>
<td>russian defense</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>russia</td>
<td>minister</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>called</td>
<td>ivanov</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>for</td>
<td>sunday</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>the</td>
<td>creation of</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>of</td>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>joint</td>
<td>front</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>front</td>
<td>for</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>combating</td>
<td>global</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>global</td>
<td>terrorism</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
\[ r = \frac{\sqrt{Q_3}}{l} \sin \left( \frac{l}{\sqrt{Q_3}} u \right), \]
Applications From HarvardNLP: OpenNMT

**Home**

OpenNMT is a industrial-strength, open-source (MIT) neural machine translation system utilizing the Torch/PyTorch mathematical toolkit.

OpenNMT is used as provided in production by major translation providers. The system is designed to be simple to use and easy to extend, while maintaining efficiency and state-of-the-art translation accuracy.
1 Deep Neural Networks for Text Processing and Generation

2 Attention Networks

3 Structured Attention Networks
   - Computational Challenges
   - Structured Attention In Practice

4 Conclusion and Future Work
## Attention Networks: Notation

\[ x_1, \ldots, x_T \quad \text{Memory bank} \]

\[ q \quad \text{Query} \]

\[ z \quad \text{Memory selection (random variable)} \]

\[ p(z \mid x, q; \theta) \quad \text{Attention distribution ("where")} \]

\[ f(x, z) \quad \text{Annotation function ("what")} \]

\[ c = \mathbb{E}_{z \mid x, q}[f(x, z)] \quad \text{Context Vector} \]

### End-to-End Requirements:

1. Need to compute attention \( p(z = i \mid x, q; \theta) \)
2. Need to backpropagate to learn parameters \( \theta \)
Attention Networks: Notation

\[ x_1, \ldots, x_T \quad \text{Memory bank} \]
\[ q \quad \text{Query} \]
\[ z \quad \text{Memory selection (random variable)} \]
\[ p(z \mid x, q; \theta) \quad \text{Attention distribution ("where")} \]
\[ f(x, z) \quad \text{Annotation function ("what")} \]
\[ c = \mathbb{E}_{z \mid x, q}[f(x, z)] \quad \text{Context Vector} \]

End-to-End Requirements:

1. Need to compute attention \( p(z = i \mid x, q; \theta) \)
2. Need to backpropagate to learn parameters \( \theta \)
Attention Networks: Machine Translation

\[ x_1, \ldots, x_T \quad \text{Memory bank} \quad \text{Source RNN hidden states} \]
\[ q \quad \text{Query} \quad \text{Decoder hidden state} \]
\[ z \quad \text{Memory selection} \quad \text{Source position} \{1, \ldots, T\} \]
\[ p(z = i | x, q; \theta) \quad \text{Attention distribution} \quad \text{softmax}(x_i^\top q) \]
\[ f(x, z) \quad \text{Annotation function} \quad \text{Memory at time } z, \text{ i.e. } x_z \]
\[ c = \mathbb{E}[f(x, z)] \quad \text{Context Vector} \]

End-to-End Requirements:

1. Need to compute attention \( p(z = i | x, q; \theta) \)
   \[ \implies \text{softmax function} \]
2. Need to backpropagate to learn parameters \( \theta \)
   \[ \implies \text{Backprop through softmax function} \]
Attention Networks: Machine Translation

\[
\begin{align*}
&x_1 & x_2 & x_3 & x_4 \\
&\text{Over} & \text{the} & \text{line} & !
\end{align*}
\]

\[
q \begin{array}{c}
<s>
\end{array}
\]
Attention Networks: Machine Translation

\[
p(z = i \mid x, q) = \text{softmax}(x_i^T q) = \frac{\exp(x_i^T q)}{\sum_{k=1}^{4} \exp(x_k^T q)}
\]

\[
p(z = 1 \mid x, q) \quad \ldots \quad p(z = 4 \mid x, q)
\]

\[
\begin{align*}
X_1 & \quad X_2 & \quad X_3 & \quad X_4 \\
\text{Over} & \quad \text{the} & \quad \text{line} & \quad ! \\
\end{align*}
\]

\[
q \quad <s>
\]
Attention Networks: Machine Translation

\[ c = \sum_{i=1}^{4} p(z = i \mid x, q) x_i = \mathbb{E}_{z \sim p(z \mid x, q)}[x_z] \]
Attention Networks: Machine Translation

\[ p(w_1 | x) = \text{softmax}(\text{MLP}([c; q])) \]
1 Deep Neural Networks for Text Processing and Generation

2 Attention Networks

3 **Structured Attention Networks**
   - Computational Challenges
   - Structured Attention In Practice

4 Conclusion and Future Work
Structured Attention Networks

- Replace simple attention with distribution over a combinatorial set of structures
- Attention distribution represented with graphical model over multiple latent variables
- Compute attention using embedded inference.

New Model

\[ p(z | x, q; \theta) \quad \text{Attention distribution over structures } z \]
Structured Attention Networks for Neural Machine Translation

$$\sum_{i=1}^{4} p(z = i \mid x, q) = 1$$

0.1
0.2
0.6
0.1

$x_1$ \rightarrow $x_2$ \rightarrow $x_3$ \rightarrow $x_4$ \rightarrow $q$

Over the line ! <s>
Structured Attention Networks

\[ p(z_1, z_2, z_3, z_4 \mid x, q) = \text{softmax}\{\theta(z_1, z_2, z_3, z_4)\} \]
\[ = \frac{1}{Z} \exp(\theta(z_1, z_2, z_3, z_4)) \]

\[ Z = \sum_{[z'_1, z'_2, z'_3, z'_4] \in \{0,1\}^4} \exp(\theta(z'_1, z'_2, z'_3, z'_4)) \]

\[ z_i = 1 \quad \text{and} \quad z_i = 0 \]

Diagram with nodes and arrows indicating connections between \( x_1, x_2, x_3, x_4 \) and \( q \) over the line.
Structured Attention Networks for Neural Machine Translation

\[ p(z_1 = 0, z_2 = 1, z_3 = 1, z_4 = 0 \mid x, q) \]
Structured Attention Networks for Neural Machine Translation

$p(z_1 = 0, z_2 = 0, z_3 = 1, z_4 = 0 | x, q)$

$z_i = 1$

$z_i = 0$

$x_1 \xrightarrow{} x_2 \xrightarrow{} x_3 \xrightarrow{} x_4$

$q$

Over the line ! <s>
Structured Attention Networks for Neural Machine Translation

\[ c = \sum_{z_1, z_2, z_3, z_4} p(z_1, z_2, z_3, z_4 \mid x, q) f(x, z) = \mathbb{E}_{z \sim p(z \mid x, q)}[f(x, z)] \]

\[ z_i = 1 \]

\[ z_i = 0 \]

\[ x_1 \quad x_2 \quad x_3 \quad x_4 \]

\[ q \]

Over the line!
Motivation: Structured Output Prediction

Modeling the structured output (i.e. graphical model on top of a neural net) has improved performance (LeCun et al., 1998; Lafferty et al., 2001; Collobert et al., 2011)

- Given a sequence \( x = x_1, \ldots, x_T \)
- Factored potentials \( \theta_{i,i+1}(z_i, z_{i+1}; x) \)

\[
p(z | x; \theta) = \text{softmax} \left( \sum_{i=1}^{T-1} \theta_{i,i+1}(z_i, z_{i+1}; x) \right) = \frac{1}{Z} \exp \left( \sum_{i=1}^{T-1} \theta_{i,i+1}(z_i, z_{i+1}; x) \right)
\]
Neural CRF for Sequence Tagging (Collobert et al., 2011)

Factored potentials $\theta$ come from neural network.
Inference in Linear-Chain CRF

Forward/backward: \( p(z_i \mid x) \) for all \( i \in [1, \ldots, T] \)

Fast algorithms for computing \( p(z_i \mid x) \)
1. Deep Neural Networks for Text Processing and Generation

2. Attention Networks

3. Structured Attention Networks
   - Computational Challenges
   - Structured Attention In Practice

4. Conclusion and Future Work
**Structured Attention Networks: Notation**

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1, \ldots, x_T$</td>
<td>Memory bank</td>
<td>-</td>
</tr>
<tr>
<td>$q$</td>
<td>Query</td>
<td>-</td>
</tr>
<tr>
<td>$z_1, \ldots, z_T$</td>
<td>Memory selection</td>
<td>Selection over structures</td>
</tr>
<tr>
<td>$p(z_i</td>
<td>x, q; \theta)$</td>
<td>Attention distribution</td>
</tr>
<tr>
<td>$f(x, z)$</td>
<td>Annotation function</td>
<td>Neural representation</td>
</tr>
</tbody>
</table>
Challenge: End-to-End Training

Requirements:

1. Compute attention distribution (marginals) $p(z_i | x, q; \theta)$
   $\implies$ Forward-backward algorithm

2. Gradients wrt attention distribution parameters $\theta$.
   $\implies$ Backpropagation through forward-backward algorithm
Challenge: End-to-End Training

Requirements:

1. **Compute attention distribution (marginals)** \( p(z_i | x, q; \theta) \)
   \[\implies \text{Forward-backward algorithm}\]

2. **Gradients wrt attention distribution parameters** \( \theta \).
   \[\implies \text{Backpropagation through forward-backward algorithm}\]
Challenge: End-to-End Training

Requirements:

1. Compute attention distribution (marginals) $p(z_i | x, q; \theta)$
   $\implies$ Forward-backward algorithm

2. Gradients wrt attention distribution parameters $\theta$.
   $\implies$ Backpropagation through forward-backward algorithm
Review: Forward-Backward Algorithm in Practice

\( \theta \): input potentials (e.g. from NN)

\( \alpha, \beta \): dynamic programming tables

**procedure** \textsc{StructAttention}(\( \theta \))

**Forward**

\textbf{for} \( i = 1, \ldots, n; z_i \) \textbf{do}

\[
\alpha[i, z_i] \leftarrow \sum_{z_{i-1}} \alpha[i - 1, z_{i-1}] \times \exp(\theta_{i-1,i}(z_{i-1}, z_i))
\]

**Backward**

\textbf{for} \( i = n, \ldots, 1; z_i \) \textbf{do}

\[
\beta[i, z_i] \leftarrow \sum_{z_{i+1}} \beta[i + 1, z_{i+1}] \times \exp(\theta_{i,i+1}(z_i, z_{i+1}))
\]
Forward-Backward Algorithm (Log-Space Semiring Trick)

$\theta$: input potentials (e.g. from MLP or parameters)

\[
x \oplus y = \log(\exp(x) + \exp(y))
\]

\[
x \otimes y = x + y
\]

**procedure** $\text{StructAttention}(\theta)$

**Forward**

for $i = 1, \ldots, n; z_i$ do

\[
\alpha[i, z_i] \leftarrow \bigoplus_{z_{i-1}} \alpha[i-1, y] \otimes \theta_{i-1,i}(z_{i-1}, z_i)
\]

**Backward**

for $i = n, \ldots, 1; z_i$ do

\[
\beta[i, z_i] \leftarrow \bigoplus_{z_{i+1}} \beta[i+1, z_{i+1}] \otimes \theta_{i,i+1}(z_i, z_{i+1})
\]
Structured Attention Networks for Neural Machine Translation

```
\rho(z_1 = 1|x, q) \quad \rho(z_2 = 1|x, q) \quad \rho(z_3 = 1|x, q) \quad \rho(z_4 = 1|x, q)
```

```
\text{ForwardBackward}(\theta)
```

```
z_i = 0 \quad z_i = 1
```

```
\theta
```

```
X_1 \quad X_2 \quad X_3 \quad X_4
```

```
Over \quad the \quad line \quad !
```

```
<s>
```

\( C \)
Backpropagating through Forward-Backward

$\nabla_L^p$: Gradient of arbitrary loss $L$ with respect to marginals $p$

**procedure** `BACKPROPSSTRUCTATTEN(θ, p, \nabla_L^α, \nabla_L^β)`

**Backprop Backward**

for $i = n, \ldots, 1; z_i$ do

$$
\hat{β}[i, z_i] \leftarrow \nabla_L^α[i, z_i] \oplus \bigoplus_{z_{i+1}} \theta_{i,i+1}(z_i, z_{i+1}) \otimes \hat{β}[i + 1, z_{i+1}]
$$

**Backprop Forward**

for $i = 1, \ldots, n; z_i$ do

$$
\hat{α}[i, z_i] \leftarrow \nabla_L^β[i, z_i] \oplus \bigoplus_{z_{i-1}} \theta_{i-1,i}(z_{i-1}, z_i) \otimes \hat{α}[i - 1, z_{i-1}]
$$

**Potential Gradients**

for $i = 1, \ldots, n; z_i, z_{i+1}$ do

$$
\nabla_L^\theta_{i-1,i}(z_i, z_{i+1}) \leftarrow \text{signexp} (\hat{α}[i, z_i] \otimes β[i + 1, z_{i+1}] \oplus α[i, z_i] \otimes \hat{β}[i + 1, z_{i+1}] \otimes −A)
$$
Interesting Issue: Negative Gradients Through Attention

- $\nabla_p^L$: Gradient could be negative, but working in log-space!
- Signed Log-space semifield Trick (Li and Eisner, 2009)
- Use tuples $(l_a, s_a)$ where $l_a = \log|a|$ and $s_a = \text{sign}(a)$

<table>
<thead>
<tr>
<th>$s_a$</th>
<th>$s_b$</th>
<th>$l_{a+b}$</th>
<th>$s_{a+b}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>+</td>
<td>$l_a + \log(1 + d)$</td>
<td>+</td>
</tr>
<tr>
<td>+</td>
<td>−</td>
<td>$l_a + \log(1 - d)$</td>
<td>+</td>
</tr>
<tr>
<td>−</td>
<td>+</td>
<td>$l_a + \log(1 - d)$</td>
<td>−</td>
</tr>
<tr>
<td>−</td>
<td>−</td>
<td>$l_a + \log(1 + d)$</td>
<td>−</td>
</tr>
</tbody>
</table>

(Similar rules for $\otimes$)
Structured Attention Networks for Neural Machine Translation

\[ \frac{\partial L}{\partial p_1} \quad \frac{\partial L}{\partial p_2} \quad \frac{\partial L}{\partial p_3} \quad \frac{\partial L}{\partial p_4} \quad \frac{\partial L}{\partial c} \]

BackpropForwardBackward \( \frac{\partial L}{\partial \theta} \)

\[ z_i = 1 \quad \text{Over} \quad \text{the} \quad \text{line} \quad ! \quad <s> \]
1 Deep Neural Networks for Text Processing and Generation

2 Attention Networks

3 Structured Attention Networks
   - Computational Challenges
   - Structured Attention In Practice

4 Conclusion and Future Work
Implementation
(http://github.com/harvardnlp/struct-attn))

- General-purpose structured attention unit.
- All dynamic programming is GPU optimized for speed.
- Additionally supports pairwise potentials and marginals.

NLP Experiments

- Machine Translation
- Question Answering
- Natural Language Inference
Use segmentation CRF for attention, i.e. binary vectors of length $n$

$p(z_1, \ldots, z_T \mid x, q)$ parameterized with a linear-chain CRF.

Neural “phrase-based” translation.

Unary potentials (Encoder RNN):

$$\theta_i(k) = \begin{cases} 
    x_i W q, & k = 1 \\
    0, & k = 0 
\end{cases}$$

Pairwise potentials (Simple Parameters):

4 additional binary parameters (i.e., $b_{0,0}, b_{0,1}, b_{1,0}, b_{1,1}$)
Neural Machine Translation Experiments

Data:

- Japanese → English (from WAT 2015)
- Traditionally, word segmentation as a preprocessing step
- Use structured attention learn an implicit segmentation model

Experiments:

- Japanese characters → English words
- Japanese words → English words
Neural Machine Translation Experiments

<table>
<thead>
<tr>
<th></th>
<th>Simple</th>
<th>Sigmoid</th>
<th>Structured</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CHAR → WORD</strong></td>
<td>12.6</td>
<td>13.1</td>
<td>14.6</td>
</tr>
<tr>
<td><strong>WORD → WORD</strong></td>
<td>14.1</td>
<td>13.8</td>
<td>14.3</td>
</tr>
</tbody>
</table>

BLEU scores on test set (higher is better).

Models:

- Simple softmax attention
- Sigmoid attention
- Structured attention
電気化問題の解決には二つの課題があった。
Attention Visualization: Simple Attention

There were two problems in the solution of the electrification problem.
Attention Visualization: Sigmoid Attention

There were two problems in the solution of the electrification problem.
**Attention Visualization: Structured Attention**

<table>
<thead>
<tr>
<th>帯電 問題 の 解 決 に は 二 つ の 課 項 が あ っ た。</th>
</tr>
</thead>
<tbody>
<tr>
<td>There were two problems in the solution of the electrification problem.</td>
</tr>
</tbody>
</table>
Simple Non-Factoid Question Answering

Simple attention: Greedy soft-selection of $K$ supporting facts

- Lily is a rhino → Brian is a rhino → Brian is white

- Greg is a frog
- Brian is a rhino
- Lily is a rhino
- Greg is green
- Brian is white
- John is a frog
Structured Attention Networks for Question Answering

Structured attention: Consider all possible sequences

- Lily is a rhino → Brian is a rhino → Brian is white
- Greg is a frog
- Brian is a rhino
- Lily is a rhino
- Greg is green
- Brian is white
- John is a frog
Structured Attention Networks for Question Answering

baBi tasks (Weston et al., 2015): 1k questions per task

<table>
<thead>
<tr>
<th>Task</th>
<th>$K$</th>
<th>Simple</th>
<th></th>
<th>Structured</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ans %</td>
<td>Fact %</td>
<td>Ans %</td>
<td>Fact %</td>
<td></td>
</tr>
<tr>
<td>TASK 02</td>
<td>2</td>
<td>87.3</td>
<td>46.8</td>
<td>84.7</td>
<td>81.8</td>
</tr>
<tr>
<td>TASK 03</td>
<td>3</td>
<td>52.6</td>
<td>1.4</td>
<td>40.5</td>
<td>0.1</td>
</tr>
<tr>
<td>TASK 11</td>
<td>2</td>
<td>97.8</td>
<td>38.2</td>
<td>97.7</td>
<td>80.8</td>
</tr>
<tr>
<td>TASK 13</td>
<td>2</td>
<td>95.6</td>
<td>14.8</td>
<td>97.0</td>
<td>36.4</td>
</tr>
<tr>
<td>TASK 14</td>
<td>2</td>
<td>99.9</td>
<td>77.6</td>
<td>99.7</td>
<td>98.2</td>
</tr>
<tr>
<td>TASK 15</td>
<td>2</td>
<td>100.0</td>
<td>59.3</td>
<td>100.0</td>
<td>89.5</td>
</tr>
<tr>
<td>TASK 16</td>
<td>3</td>
<td>97.1</td>
<td>91.0</td>
<td>97.9</td>
<td>85.6</td>
</tr>
<tr>
<td>TASK 17</td>
<td>2</td>
<td>61.1</td>
<td>23.9</td>
<td>60.6</td>
<td>49.6</td>
</tr>
<tr>
<td>TASK 18</td>
<td>2</td>
<td>86.4</td>
<td>3.3</td>
<td>92.2</td>
<td>3.9</td>
</tr>
<tr>
<td>TASK 19</td>
<td>2</td>
<td>21.3</td>
<td>10.2</td>
<td>24.4</td>
<td>11.5</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>–</td>
<td>81.4</td>
<td>39.6</td>
<td>81.0</td>
<td>53.7</td>
</tr>
</tbody>
</table>
Visualization of Structured Attention

Question: what color is bernhard? green
Correct Facts: 5, 6, 8
Natural Language Inference

Given a premise (P) and a hypothesis (H), predict the relationship: Entailment (E), Contradiction (C), Neutral (N)

<table>
<thead>
<tr>
<th>P</th>
<th>The boy is running through a grassy area.</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>The boy is in his room.</td>
</tr>
<tr>
<td></td>
<td>A boy is running outside.</td>
</tr>
<tr>
<td></td>
<td>The boy is in a park.</td>
</tr>
</tbody>
</table>

Many existing models run parsing as a preprocessing step and attend over parse trees.
Neural CRF Parsing (Durrett and Klein, 2015; Kipperwasser and Goldberg, 2016)

$z_{ij} = 1 \quad \rightarrow \quad i$-th word is parent of $j$-th word
Neural CRF Parsing (Durrett and Klein, 2015; Kipperwasser and Goldberg, 2016)

$\begin{align*}
    z_{1,3} &= 1 \\
    z_{3,2} &= 1 \\
    z_{3,5} &= 1 \\
    Z_{5,4} &= 1
\end{align*}$
.syntactic attention network

1. Attention distribution (probability of a parse tree)
   $\Rightarrow$ Inside/outside algorithm

2. Gradients wrt attention distribution parameters: $\frac{\partial L}{\partial \theta}$
   $\Rightarrow$ Backpropagation through inside/outside algorithm

Forward/backward pass on inside-outside version of Eisner’s algorithm (Eisner, 1996) takes $O(T^3)$ time.
Syntactic Attention Network

1. Attention distribution (probability of a parse tree) $\Rightarrow$ Inside/outside algorithm

2. Gradients wrt attention distribution parameters: $\frac{\partial L}{\partial \theta} \Rightarrow$ Backpropagation through inside/outside algorithm

Forward/backward pass on inside-outside version of Eisner’s algorithm (Eisner, 1996) takes $O(T^3)$ time.
Syntactic Attention Network

1. Attention distribution (probability of a parse tree)  
   \(\Rightarrow\) Inside/outside algorithm

2. Gradients wrt attention distribution parameters: \(\frac{\partial L}{\partial \theta}\)  
   \(\Rightarrow\) Backpropagation through inside/outside algorithm

Forward/backward pass on inside-outside version of Eisner’s algorithm (Eisner, 1996) takes \(O(T^3)\) time.
Syntactic Attention Network

1. Attention distribution (probability of a parse tree)  
   \[ \Rightarrow \text{ Inside/outside algorithm} \]

2. Gradients wrt attention distribution parameters:  
   \[ \frac{\partial L}{\partial \theta} \]  
   \[ \Rightarrow \text{ Backpropagation through inside/outside algorithm} \]

Forward/backward pass on inside-outside version of Eisner’s algorithm (Eisner, 1996) takes \( O(T^3) \) time.
Backpropagation through Inside-Outside Algorithm
Structured Attention Networks with a Parser ("Syntactic Attention")
Structured Attention Networks with a Parser ("Syntactic Attention")
Structured Attention Networks with a Parser ("Syntactic Attention")
Structured Attention Networks with a Parser ("Syntactic Attention")

\[ c_2 = \text{soft-parent}(\text{John}) = \sum_{i=1}^{5} p(z_{i,2} = 1 | x) x_i \]
Structured Attention Networks with a Parser ("Syntactic Attention")

$p(z_{i,3} = 1 | x)$

$c_3 = \text{soft-parent}(\text{hit})$

$= \sum_{i=1}^{5} p(z_{i,3} = 1 | x)x_i$
Structured Attention Networks with a Parser ("Syntactic Attention")

$c_4 = \text{soft-parent(}\text{the})$

$= \sum_{i=1}^{5} p(z_{i,4} = 1 | x) x_i$
Structured Attention Networks with a Parser ("Syntactic Attention")

\[ c_5 = \text{soft-parent}(\text{ball}) = \sum_{i=1}^{5} p(z_{i,5} = 1 | x) x_i \]
Structured Attention Networks for Natural Language Inference

Dataset: Stanford Natural Language Inference (Bowman et al., 2015)

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy %</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Attention</td>
<td>85.8</td>
</tr>
<tr>
<td>Hard parent</td>
<td>86.1</td>
</tr>
<tr>
<td>Simple Attention</td>
<td>86.2</td>
</tr>
<tr>
<td>Structured Attention</td>
<td>86.8</td>
</tr>
</tbody>
</table>

- No attention: word embeddings only
- “Hard” parent from a pipelined dependency parser
- Simple attention (simple softmax instead of syntactic attention)
- Structured attention (soft parents from syntactic attention)
Run Viterbi algorithm on the parsing layer to get the MAP parse:

$$\hat{z} = \arg \max_z p(z \mid x, q)$$

The men are fighting outside a deli.
1 Deep Neural Networks for Text Processing and Generation

2 Attention Networks

3 Structured Attention Networks
   - Computational Challenges
   - Structured Attention In Practice

4 Conclusion and Future Work
Structured Attention Networks

- Generalize attention to incorporate latent structure
- Exact inference through dynamic programming
- Training remains end-to-end.

Future work

- Approximate differentiable inference in neural networks
- Incorporate other probabilistic models into deep learning.
- Compare further to methods using EM or hard structures.
Other Work: **Lie-Access Neural Memory** (Yang and Rush, 2017)


References V


