Structured Attention Networks

Yoon Kim Carl Denton Luong Hoang Alexander M. Rush

HarvardNLP

ICLR, 2017

Presenter: Chao Jiang

Outline

Pure Encoder-Decoder Network

Input (sentence, image, etc.)

Fixed-Size Encoder (MLP, RNN, CNN)

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

Decoder

$\text{Decoder}(\text{Encoder}(\text{input}))$

Pure Encoder-Decoder Network

Input (sentence, image, etc.)

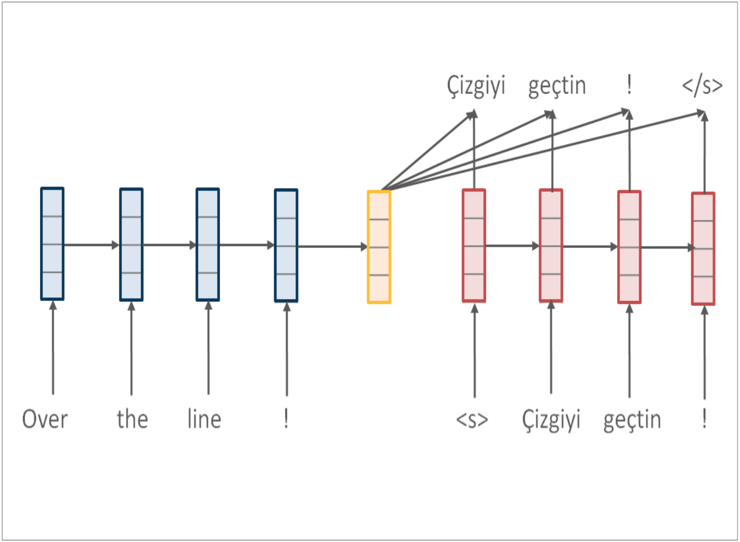
Fixed-Size Encoder (MLP, RNN, CNN)

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

Decoder

$\text{Decoder}(\text{Encoder}(\text{input}))$

Pure Encoder-Decoder Network



Encoder-Decoder with Attention

- Machine Translation
- Question Answering
- Natural Language Inference
- Algorithm Learning
- Parsing
- Speech Recognition
- Summarization
- Caption Generation
- and more $\cdots$

Attention Networks

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

End-to-End Requirements:

1. Need to compute attention $p(z = i \mid x, q; \theta)$

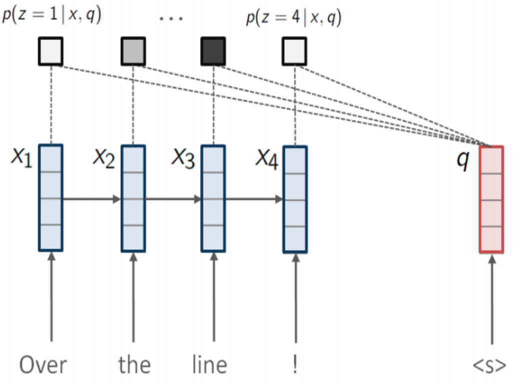
   $\Longrightarrow$ softmax function

2. Need to backpropagate to learn parameters θ

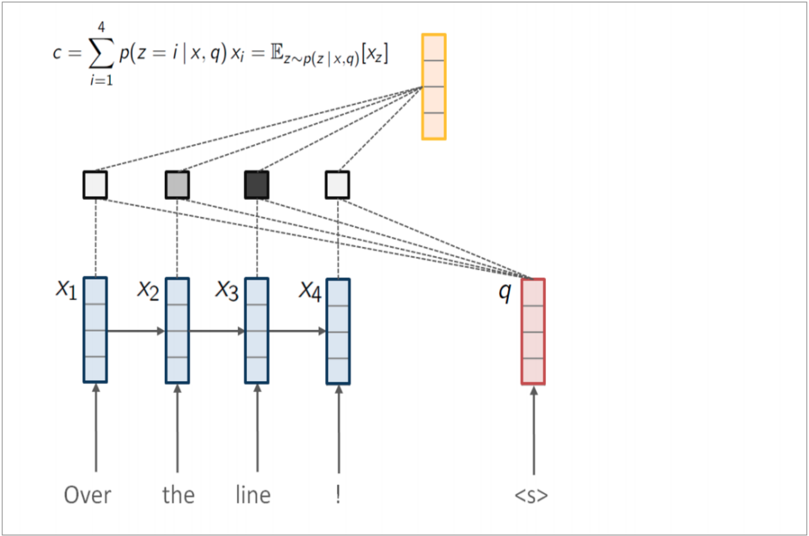
   $\Longrightarrow$ Backprop through softmax function

Attention Networks

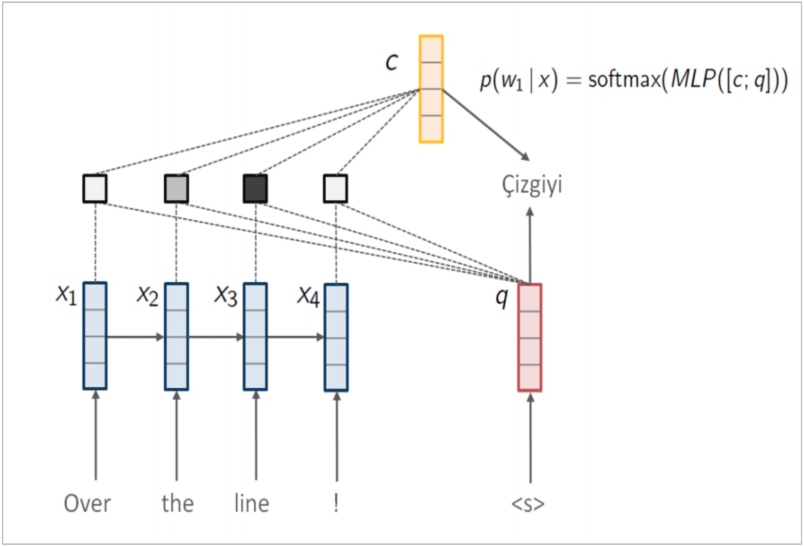
$$p(z = i \mid x, q) = \mathrm{softmax}(x_i^\top q) = \frac{\exp(x_i^\top q)}{\sum_{k=1}^{4} \exp(x_k^\top q)}$$



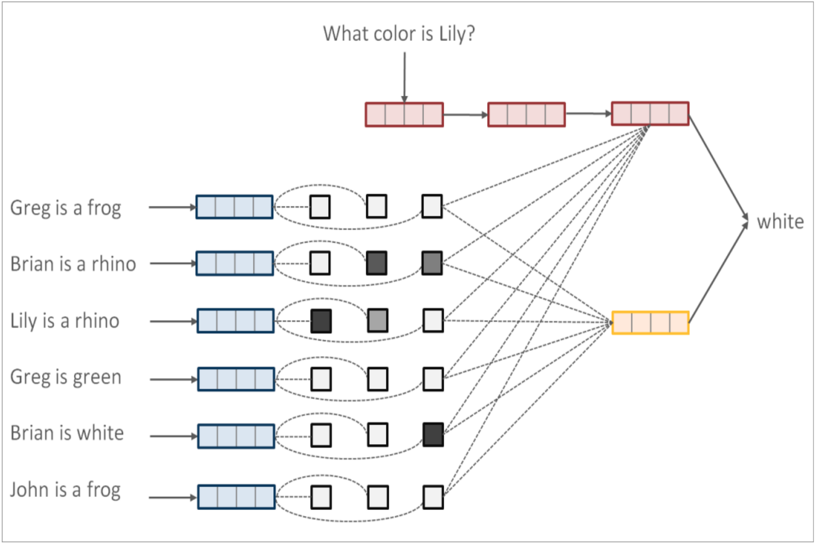
Attention Networks



Attention Networks



Attention Networks



Outline

Overview

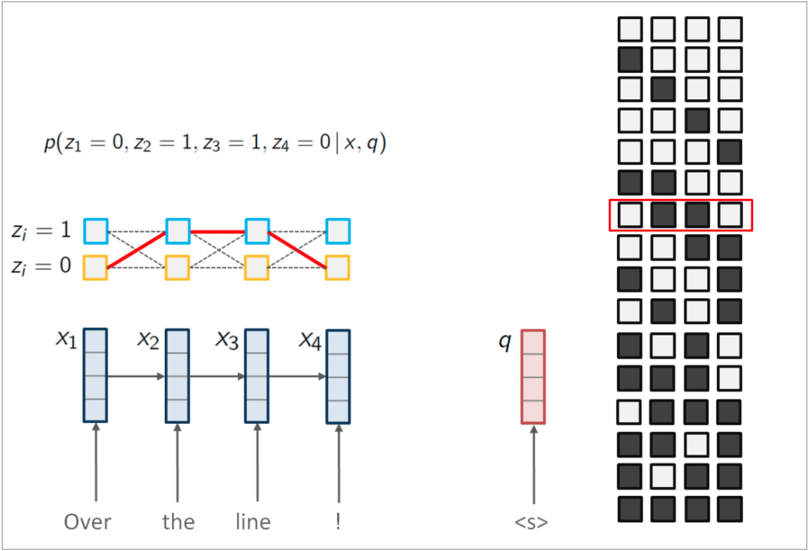
Key difference:

- Replace simple attention with distribution over a combinatorial set of structures
- Attention distribution represented with graph model over multiple latent variables
- Compute attention using embedding infoerence

New Model:

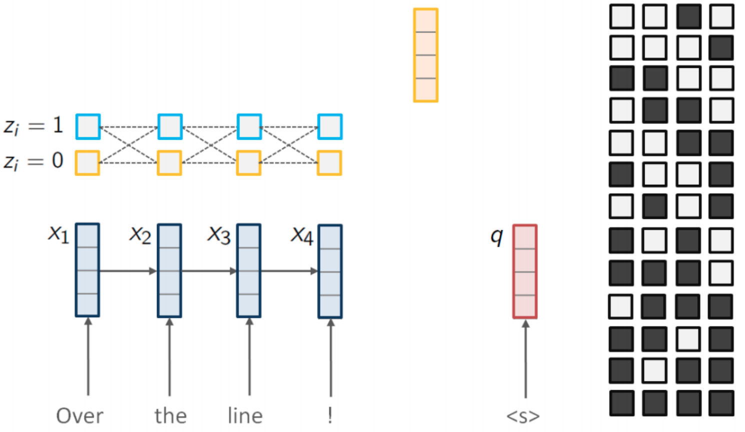
- $P(z|x, q:\theta)$ Attention distribution over structures z

Structured Attention Networks

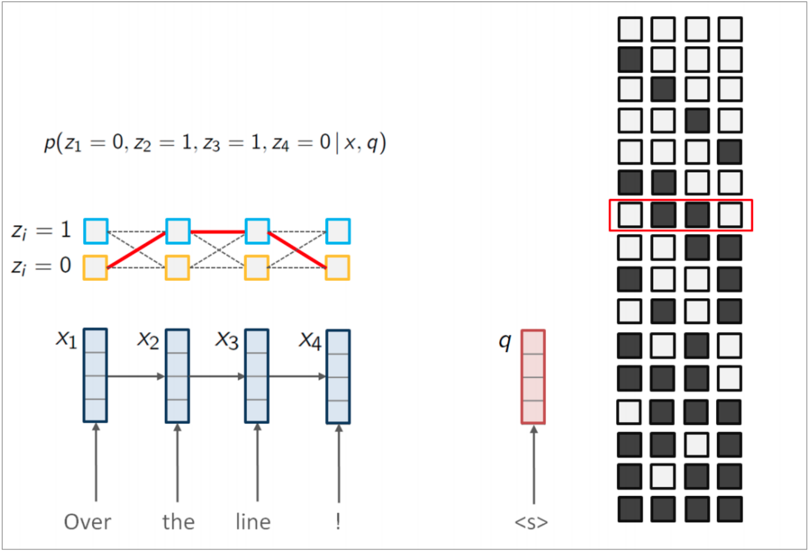


Structured Attention Networks

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



Structured Attention Networks



Motivation: Structured Output Prediction

Modeling the structured **output** (i.e. graphical model in top of a neural net) has improved performance

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

$$p(z|x;\theta) = softmax(\sum_{i=1}^{T-1} \theta_{i,i+1}(z_i, z_{i+1}; x)) = \frac{1}{Z} exp(\sum_{i=1}^{T-1} \theta_{i,i+1}(z_i, z_{i+1}; x))$$

Outline

Structured Attention Networks: Notation

$x_1, \ldots, x_T$	Memory bank	-
q	Query	-
$z_1, \ldots, z_T$	Memory selection	Selection over structures
$p(z_i \mid x, q; \theta)$	Attention distribution	Marginal distributions
$f(x, z)$	Annotation function	Neural representation

Challenge: End-to-End Training

Requirements:

1. Compute attention distribution (marginals) $p(z_i \mid x, q; \theta)$

   $\Longrightarrow$ Forward-backward algorithm

2. Gradients wrt attention distribution parameters θ.

   $\Longrightarrow$ Backpropagation through forward-backward algorithm

Forward-Backward Algorithms

θ: input potentials (e.g. from NN)

α, β: dynamic programming tables

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

Forward

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

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

$$\beta[i, z_i] \leftarrow \sum_{z_{i+1}1} \beta[i+1, z_{i+1}] \times \exp(\theta_{i,i+1}(z_i, z_{i+1}))$$

Forward-Backward Algorithms (Log-Space)

θ: input potentials (e.g. from MLP or parameters)

$x \oplus y = \log(\exp(x) + \exp(y))$

$x \otimes y = x + y$

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

Forward

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

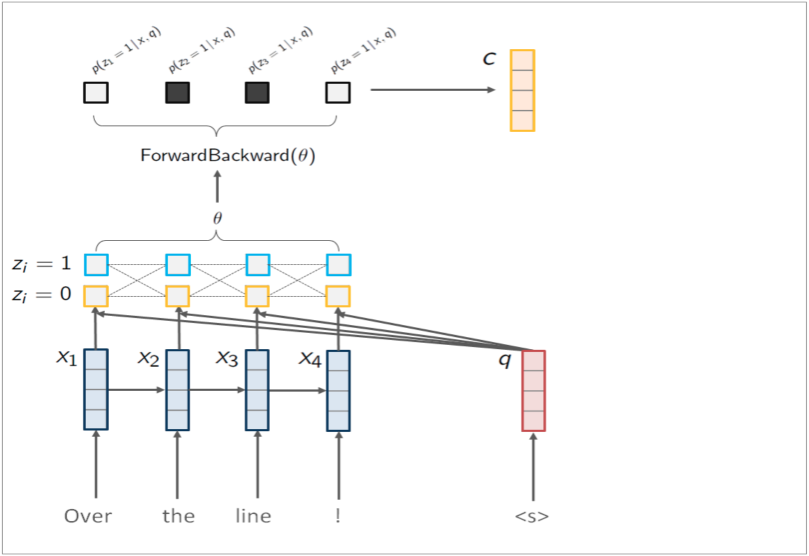
$$\alpha[i, z_i] \leftarrow \bigoplus_{z_{i-1}} \alpha[i-1, y] \otimes \theta_{i-1,i}(z_{i-1}, z_i)$$

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 NMT



Backpropagating through Forward-Backward

$\nabla_p^{\mathcal{L}}$: Gradient of arbitrary loss $\mathcal{L}$ with respect to marginals p

procedure $\textsc{BackpropStructAtten}(\theta, p, \nabla_\alpha^{\mathcal{L}}, \nabla_\beta^{\mathcal{L}})$

Backprop Backward

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

$$\hat{\beta}[i, z_i] \leftarrow \nabla_\alpha^{\mathcal{L}}[i, z_i] \oplus \bigoplus_{z_{i+1}} \theta_{i,i+1}(z_i, z_{i+1}) \otimes \hat{\beta}[i+1, z_{i+1}]$$

Backprop Forward

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

$$\hat{\alpha}[i, z_i] \leftarrow \nabla_\beta^{\mathcal{L}}[i, z_i] \oplus \bigoplus_{z_{i-1}} \theta_{i-1,i}(z_{i-1}, z_i) \otimes \hat{\alpha}[i-1, z_{i-1}]$$

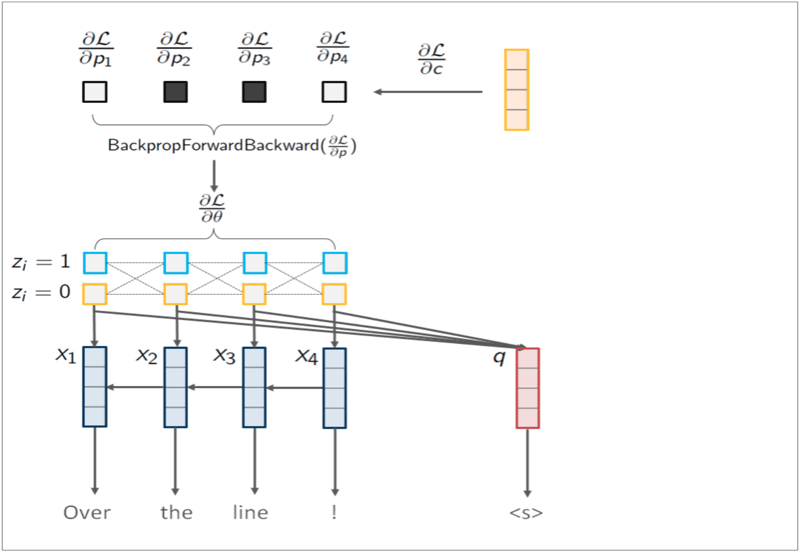
Potential Gradients

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

$$\nabla_{\theta_{i-1,i}(z_i, z_{i+1})}^{\mathcal{L}} \leftarrow \mathrm{signexp}(\hat{\alpha}[i, z_i] \otimes \beta[i+1, z_{i+1}] \oplus \alpha[i, z_i] \otimes \hat{\beta}[i+1, z_{i+1}] \oplus \alpha[i, z_i] \otimes \beta[i+1, z_{i+1}] \otimes -A)$$

Outline

Structured Attention Networks for NMT



Neural Machine Translation Experiments

Data

- Dataset is from WAT 2015)
- Japanese characters to English characters
- Japanese words to English words

Neural Machine Translation Experiments

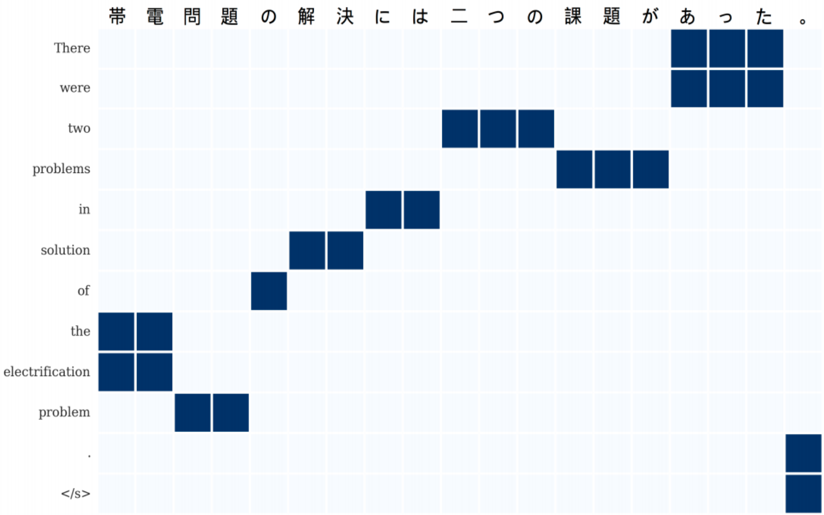
	Simple	Sigmoid	Structured
Char → Word	12.6	13.1	14.6
Word → Word	14.1	13.8	14.3

BLEU scores on test set (higher is better).

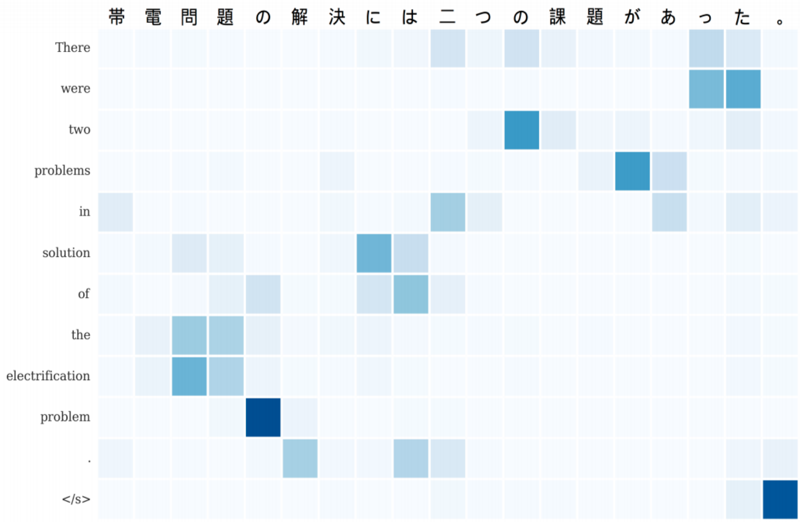
Models:

- Simple softmax attention
- Sigmoid attention
- Structured attention

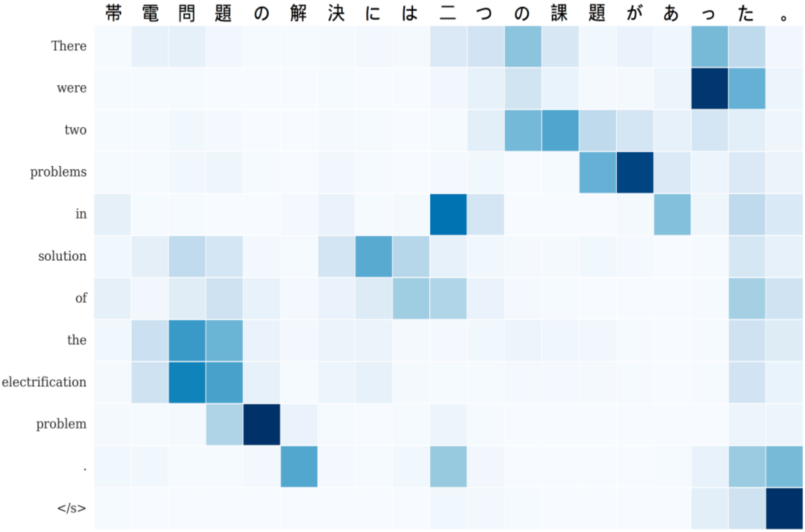
Attention Visualization: Ground Truth



Attention Visualization: Simple Attention



Attention Visualization: Structured Attention



Structured Attention Networks for Question Answering

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

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

Structured Attention Networks for Natural Language Inference

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

Model	Accuracy %
No Attention	85.8
Hard parent	86.1
Simple Attention	86.2
Structured Attention	86.8

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