Latent Variable Models for Text

March 20, 2018

1. Language as a Latent Variable

2. Our Proposal: Attention as a Latent Variable

1. Language as a Latent Variable

2. Our Proposal: Attention as a Latent Variable
   - Attention Mechanism
   - Attention as a Latent Variable

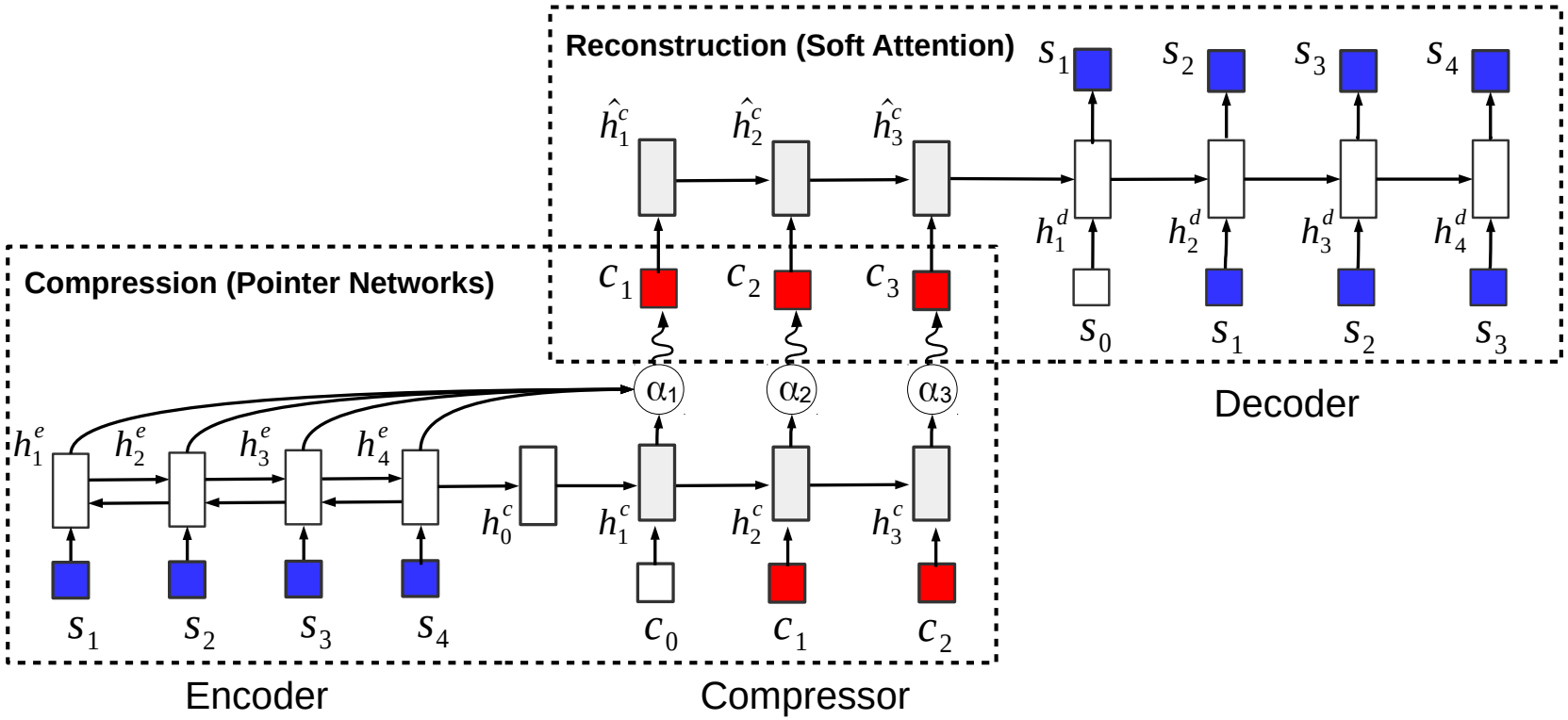
Overview of Paper

- Semi-supervised summarization
- Builds on Kingma et al. 2014's M2 (the semisupervised version)
- But uses a sequence of words as the latent representation

The Idea for Extractive Summarization

- Start with source sentence $\mathbf{x}$ = 'I wish I could love dogs but I just hate them'
- Sample a summary $\mathbf{y}$ = 'I love dogs' by picking words from the source
- Reconstruct the source sentence given the summary using an attentive decoder
- Use the probability of the source sentence under the reconstruction decoder (and a couple other terms) as signal for how good the summary was

Auto-Encoding Sentence Compression



Note: Some of the parameters of the inference network are used as the decoder network's encoder, but the shared parameters are not updated using gradients from the decoder

Auto-Encoding Sentence Compression

- The inference network $q_\lambda(\mathbf{y} \mid \mathbf{x})$ uses hard attention at every timestep to select a source token
- Use a bidirectional source encoder on $\mathbf{x}$ to get source embedding matrix $H^e = \mathrm{BRNN}(\mathbf{x})$, whose ith element is the vector $\mathbf{h}_i^e$.
- Select source token $\mathbf{x}_i$ with a pointer network 'compressor'

$$\mathbf{h}_j^c = \mathrm{RNN}(\mathbf{h}_{j-1}^c, \mathbf{y}_{j-1}) \tag{1}$$

$$\boldsymbol{\alpha}_j = \mathrm{attention}(\mathbf{h}_j^c, H^e) \tag{2}$$

$$\mathbf{y}_j \sim \mathrm{Cat}(\boldsymbol{\alpha}_j) \tag{3}$$

ASC Continued

- Let H^c be the concatenation of all the compressor hidden states
- The decoder $p_\theta(\mathbf{x} \mid \mathbf{y})$ is a conditional language model that attends over the hidden states H^c to reconstruct the original source sentence $\mathbf{x}$

$$\mathbf{h}_k^d = \mathrm{RNN}(\mathbf{h}_{k-1}^d, \mathbf{x}_{k-1}) \tag{4}$$

$$\mathbf{v}_k = \mathrm{attention}(\mathbf{h}_k^d, H^c) \tag{5}$$

$$\mathbf{d}_k = \mathbf{v}_k^T H^c \tag{6}$$

$$p_\theta(\mathbf{x}_k \mid \mathbf{x}_{<k}, \mathbf{y}) = \mathrm{softmax}(W\mathbf{d}_k) \tag{7}$$

Details and Recap

- The attention formulation from Vinyals, Fortunato, and Jaitly 2015, which is pretty close to the 'general' attention in Luong, Pham, and Manning 2015

$$\text{attention}(q, C) = \text{softmax}(\mathbf{v}^T \tanh(Wq + VC)) \tag{8}$$

- $p_\theta(\mathbf{x} \mid \mathbf{y})$ is the reconstructive attention based decoder
- $q_\lambda(\mathbf{y} \mid \mathbf{x})$ is the summarizing pointer network, and we omit the conditioning on $\mathbf{x}$ when convenient (randomly)
- $p(\mathbf{y})$ is a language model prior

Marginal Likelihood

- The inference network's parameters will be denoted by λ and the decoder network's by θ
- As usual, the marginal likelihood is intractable

$$\log p(\mathbf{x}) = \log \sum_{\mathbf{y}} p_\theta(\mathbf{x}, \mathbf{y}) \tag{9}$$

since we cannot enumerate all possible summaries, even if they are extractive

Objective

- So we lower bound it with Jensen's inequality and maximize the ELBO $\mathcal{L}$

$$\log \sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y}) = \log \sum_{\mathbf{y}} q_\lambda(\mathbf{y}) \frac{p_\theta(\mathbf{x}, \mathbf{y})}{q_\lambda(\mathbf{y})} \tag{10}$$

$$= \log \mathbb{E}_{\mathbf{y} \sim q_\lambda(\mathbf{y})} \left[\frac{p_\theta(\mathbf{x}, \mathbf{y})}{q_\lambda(\mathbf{y})} \right] \tag{11}$$

$$\geq \mathbb{E}_{\mathbf{y} \sim q_\lambda(\mathbf{y})} \left[\log \frac{p_\theta(\mathbf{x}, \mathbf{y})}{q_\lambda(\mathbf{y})} \right] \tag{12}$$

$$= \mathbb{E}_{\mathbf{y} \sim q_\lambda(\mathbf{y})} \left[\log p_\theta(\mathbf{x} \mid \mathbf{y}) \right] \tag{13}$$

$$- D_{KL}(q_\lambda(\mathbf{y}) \| p(\mathbf{y}))$$

$$= \mathcal{L}$$

Training Details

- The gradient of the ELBO with respect to the reconstructive decoder only depends on $p_\theta(\mathbf{x} \mid \mathbf{y})$

$$\mathcal{L} = \underbrace{\mathbb{E}_{\mathbf{y} \sim q_\lambda(\mathbf{y})} \left[\log p_\theta(\mathbf{x} \mid \mathbf{y}) \right]}_{\text{1. Reconstruction}} - \underbrace{KL[q_\lambda(\mathbf{z}) \| p(\mathbf{z})]}_{\text{2. Regularization towards prior}}$$

- It's given by term 1

$$\nabla_\theta \mathcal{L} = \nabla_\theta \mathbb{E}_{\mathbf{y} \sim q_\lambda(\mathbf{y})} \left[\log p_\theta(\mathbf{x} \mid \mathbf{y}) \right] \tag{14}$$

$$\approx \frac{1}{M} \sum_m \nabla_\theta \log p_\theta(\mathbf{x} \mid \mathbf{y}^{(m)}) \tag{15}$$

where M sample summaries are generated through ancestral sampling

Training Details

- The gradient with respect to the inference network requires REINFORCE
- We rewrite the ELBO

$$\mathcal{L} = \mathbb{E}_{\mathbf{y} \sim q_\lambda(\mathbf{y})} \left[\log \frac{p_\theta(\mathbf{x}, \mathbf{y})}{q_\lambda(\mathbf{y})} \right] \tag{16}$$

$$= \mathbb{E}_{\mathbf{y} \sim q_\lambda(\mathbf{y})} \left[\log p_\theta(\mathbf{x} \mid \mathbf{y}) + \log p(\mathbf{y}) - \log q_\lambda(\mathbf{y}) \right] \tag{17}$$

$$= \mathbb{E}_{\mathbf{y} \sim q_\lambda(\mathbf{y})} \left[l(\mathbf{x}, \mathbf{y}) \right] \tag{18}$$

- So we have $l(\mathbf{x}, \mathbf{y}) = \log p_\theta(\mathbf{x} \mid \mathbf{y}) + \log p(\mathbf{y}) - \log q_\lambda(\mathbf{y})$

REINFORCE

- Recall the score function gradient estimator

$$p(\mathbf{x})\nabla \log p(\mathbf{x}) = \nabla p(\mathbf{x}) \tag{19}$$

- We use this to find an approximation of

$$\begin{aligned}
\nabla_\lambda \mathcal{L} &= \nabla_\lambda \mathbb{E}_{\mathbf{y}\sim q_\lambda(\mathbf{y})}\left[l(\mathbf{x},\mathbf{y})\right] && (20)\\
&= \sum_{\mathbf{y}} l(\mathbf{x},\mathbf{y})\nabla_\lambda q_\lambda(\mathbf{y}) && (21)\\
&= \sum_{\mathbf{y}} q_\lambda(\mathbf{y}) l(\mathbf{x},\mathbf{y})\nabla_\lambda \log q_\lambda(\mathbf{y})) && (22)\\
&= \mathbb{E}_{\mathbf{y}\sim q_\lambda(\mathbf{y})}\left[l(\mathbf{x},\mathbf{y})\nabla_\lambda \log q_\lambda(\mathbf{y})\right] && (23)
\end{aligned}$$

Details

- They also train a baseline to predict $l(\mathbf{x}, \mathbf{y})$ for variance reduction (control variate)
- They use a variant of KL annealing and augment the loss as follows

$$l(\mathbf{x}, \mathbf{y}) = \log p_\theta(\mathbf{x} \mid \mathbf{y}) + \lambda(\log p(\mathbf{y}) - \log q_\lambda(\mathbf{y}))$$

with $\lambda = 0.1$ without justification, but note that increasing λ results in shorter summaries $\mathbf{y}$ since the prior $p(\mathbf{y})$ prefers shorter sequences

Questions

- Why don't we use

$$\log \sum_{\mathbf{y}} p(\mathbf{y}) p_\theta(\mathbf{x} \mid \mathbf{y}) \geq \sum_{\mathbf{y}} \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y})} \left[\log p_\theta(\mathbf{x} \mid \mathbf{y}) \right]$$

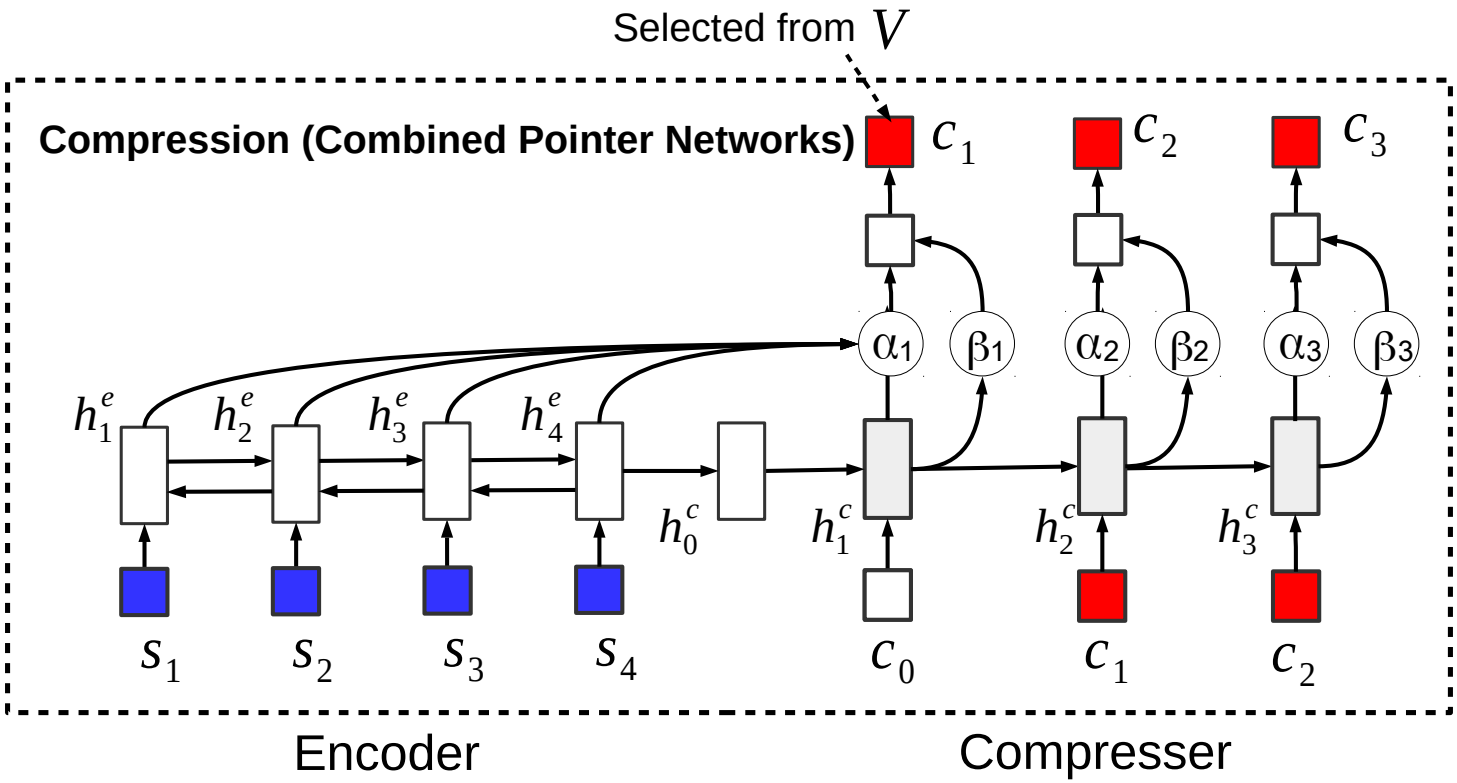
  instead of introducing $q_\lambda(\mathbf{y})$ if we're using REINFORCE anyway?

- Which parts of the reward

$$l(\mathbf{x}, \mathbf{y}) = \log p_\theta(\mathbf{x} \mid \mathbf{y}) + \log p(\mathbf{y}) - \log q_\lambda(\mathbf{y} \mid \mathbf{x})$$

  can we decompose to try to lower variance a bit more?

Forced Sentence Compression



Forced Sentence Compression

- The FSC generates from a mixture of the pointer network and a distribution over the full vocabulary
- Recall the pointer network attentions are $\boldsymbol{\alpha}_j$ at time step j
- Let the distribution over the vocabulary be $\boldsymbol{\beta}_j = \mathrm{softmax}(W\mathbf{h}_j^c)$
- They use the weighted context $\boldsymbol{\eta}_j = \boldsymbol{\alpha}_j^T H^e$ as well as the current hidden state $\mathbf{h}_j^c$ as a copy gate $\mathbf{t}_j = \sigma(\eta_j^T M \mathbf{h}_j^c)$
- The probability of a word in the summary is then

$$p(\mathbf{y}_j|\mathbf{y}_{<j},\mathbf{x}) = \begin{cases} \mathbf{t}_j\alpha_j(i) + (1-\mathbf{t}_j)\beta_j(\mathbf{y}_j), & \mathbf{y}_j = \mathbf{x}_i \\ (1-\mathbf{t}_j)\beta_j(\mathbf{y}_j), & \mathbf{y}_j \notin \mathbf{x}_i \end{cases} \tag{24}$$

Semi-Supervised Training

- Add the conditional likelihood of supervised summaries to the objective
- They do not train $p_\theta(\mathbf{x} \mid \mathbf{y})$ on the supervised data. What's an argument for and against this?

Results

- Semi-supervised saw some benefit over supervised for extractive summarization when very using very little supervision
- Semi-supervised provided more consistently some benefit over purely supervised for abstractive summarization
- Why?

1. Language as a Latent Variable

2. Our Proposal: Attention as a Latent Variable
   - Attention Mechanism
   - Attention as a Latent Variable

1. Language as a Latent Variable

2. Our Proposal: Attention as a Latent Variable
   - Attention Mechanism
   - Attention as a Latent Variable

Attention Mechanism: Basics

- Source Input: $\mathbf{x} = \{x_1, x_2, \cdots, x_S\}$
- Target Output: $\mathbf{y} = \{y_1, y_2, \cdots, y_T\}$
- Encoded Source Features: $\text{enc}(\mathbf{x}) = \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_S\}$

Attention Mechanism: Basics

- Source Input: $\mathbf{x} = \{x_1, x_2, \cdots, x_S\}$
- Target Output: $\mathbf{y} = \{y_1, y_2, \cdots, y_T\}$
- Encoded Source Features: $\text{enc}(\mathbf{x}) = \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_S\}$

$$P(\mathbf{y}|\mathbf{x}) = \prod_{j=1}^{T} P(y_j|y_{<i}, \mathbf{x})$$

Attention Mechanism: Basics

- Source Input: $\mathbf{x} = \{x_1, x_2, \cdots, x_S\}$
- Target Output: $\mathbf{y} = \{y_1, y_2, \cdots, y_T\}$
- Encoded Source Features: $\text{enc}(\mathbf{x}) = \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_S\}$

$$
\begin{aligned}
P(\mathbf{y}|\mathbf{x}) &= \prod_{j=1}^{T} P(y_j|y_{<i}, \mathbf{x}) \\
&= \prod_{j=1}^{T} P(y_j|y_{<j}, \text{enc}(\mathbf{x})) \\
&= \prod_{j=1}^{T} P(y_j|y_{<j}, \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_S\})
\end{aligned}
$$

Attention Mechanism: Basics

- Recall that

$$P(\mathbf{y}|\mathbf{x}) = \prod_{j=1}^{T} P(y_j|y_{<j}, \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_S\})$$

Attention Mechanism: Basics

- Recall that

$$P(\mathbf{y}|\mathbf{x}) = \prod_{j=1}^{T} P(y_j|y_{<j}, \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_S\})$$

- $\{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_S\}$ is of varying lengths, in order to get fixed length vector, use weighted sum:

$$P(y_j|y_{<j}, \{\mathbf{h}_1, \cdots, \mathbf{h}_S\}) = P(y_j|y_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i)$$

Attention Mechanism: Basics

- Recall that

$$P(\mathbf{y}|\mathbf{x}) = \prod_{j=1}^{T} P(y_j|y_{<j}, \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_S\})$$

- $\{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_S\}$ is of varying lengths, in order to get fixed length vector, use weighted sum:

$$P(y_j|y_{<j}, \{\mathbf{h}_1, \cdots, \mathbf{h}_S\}) = P(y_j|y_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i)$$

- The attention weights $\mathbf{a}_j = \{a_{j1}, a_{j2}, \cdots, a_{jS}\}$ lay on a simplex

Attention Mechanism: Basics

- Recall that

$$P(\mathbf{y}|\mathbf{x}) = \prod_{j=1}^{T} P(y_j|y_{<j}, \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_S\})$$

- $\{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_S\}$ is of varying lengths, in order to get fixed length vector, use weighted sum:

$$P(y_j|y_{<j}, \{\mathbf{h}_1, \cdots, \mathbf{h}_S\}) = P(y_j|y_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i)$$

- The attention weights $\mathbf{a}_j = \{a_{j1}, a_{j2}, \cdots, a_{jS}\}$ lay on a simplex
- $\mathbf{a}_j$ are modeled (discriminatively) via a neural network f:

$$\mathbf{a}_j = f(y_{<i}, \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_S\})$$

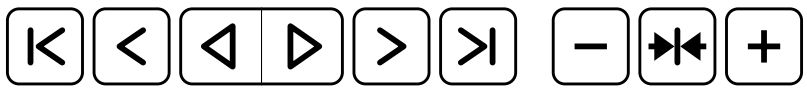
Attention Mechanism: Interpretation as Alignments

- Recall that

$$P(y_j|y_{<j}, \{\mathbf{h}_1, \cdots, \mathbf{h}_S\}) = P(y_j|y_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i)$$

- $P(y_j|y_{<j}, \mathbf{x})$ is related to $\mathbf{x}$ only through $\sum_{i=1}^{S} a_{ji}\mathbf{h}_i$
- The larger a_{ji}, the closer $\mathbf{h}_i$ to $\sum_{i=1}^{S} a_{ji}\mathbf{h}_i$
- $\mathbf{a}_j$ can be interpreted as how target word j is aligned to source words $\{x_1, \cdots, x_S\}$

$$V = i \begin{bmatrix} B & 2\overline{A} \\ -2A & -B \end{bmatrix}$$



1. Language as a Latent Variable

2. Our Proposal: Attention as a Latent Variable
   - Attention Mechanism
   - Attention as a Latent Variable

Attention as a Latent Variable

- Formally treat attention (alignments) as a latent variable
- Introduce prior over attention

Attention as a Latent Variable

- Formally treat attention (alignments) as a latent variable
- Introduce prior over attention
- Still feed decoder with $\sum_{i=1}^{S} a_{ji}\mathbf{h}_i$

Attention as a Latent Variable

- Formally treat attention (alignments) as a latent variable
- Introduce prior over attention
- Still feed decoder with $\sum_{i=1}^{S} a_{ji}\mathbf{h}_i$

$$P(y_j, \mathbf{a}_j | y_{<j}, \mathbf{a}_{<j}, \mathbf{x})$$

Attention as a Latent Variable

- Formally treat attention (alignments) as a latent variable
- Introduce prior over attention
- Still feed decoder with $\sum_{i=1}^{S} a_{ji}\mathbf{h}_i$

$$
\begin{aligned}
&P(y_j, \mathbf{a}_j | y_{<j}, \mathbf{a}_{<j}, \mathbf{x}) \\
&= P(y_j | y_{<j}, \mathbf{a}_{<j}, \mathbf{x}, \mathbf{a}_j) P(\mathbf{a}_j | y_{<j}, \mathbf{a}_{<j}, \mathbf{x})
\end{aligned}
$$

Attention as a Latent Variable

- Formally treat attention (alignments) as a latent variable
- Introduce prior over attention
- Still feed decoder with $\sum_{i=1}^{S} a_{ji}\mathbf{h}_i$

$$
\begin{aligned}
&P(y_j, \mathbf{a}_j | y_{<j}, \mathbf{a}_{<j}, \mathbf{x}) \\
&= P(y_j | y_{<j}, \mathbf{a}_{<j}, \mathbf{x}, \mathbf{a}_j) P(\mathbf{a}_j | y_{<j}, \mathbf{a}_{<j}, \mathbf{x}) \\
&= P(y_j | y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i) P(\mathbf{a}_j | y_{<j}, \mathbf{a}_{<j}, \mathbf{x})
\end{aligned}
$$

Attention as a Latent Variable

- Recall that

$$
\begin{aligned}
& P(y_j, \mathbf{a}_j | y_{<j}, \mathbf{a}_{<j}, \mathbf{x}) \\
& = P(y_j | y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji} \mathbf{h}_i) P(\mathbf{a}_j | y_{<j}, \mathbf{a}_{<j}, \mathbf{x})
\end{aligned}
$$

Attention as a Latent Variable

- Recall that

$$P(y_j, \mathbf{a}_j | y_{<j}, \mathbf{a}_{<j}, \mathbf{x})$$
$$= P(y_j | y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji} \mathbf{h}_i) P(\mathbf{a}_j | y_{<j}, \mathbf{a}_{<j}, \mathbf{x})$$

- But we do not observe attentions $\mathbf{a}$

Attention as a Latent Variable

- Recall that

$$P(y_j, \mathbf{a}_j | y_{<j}, \mathbf{a}_{<j}, \mathbf{x})$$
$$= P(y_j | y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji} \mathbf{h}_i) P(\mathbf{a}_j | y_{<j}, \mathbf{a}_{<j}, \mathbf{x})$$

- But we do not observe attentions $\mathbf{a}$
- We need to marginalize $\mathbf{a}$:

$$\log P(\mathbf{y}|\mathbf{x}) = \log \int_{\mathbf{a}} \prod_{j=1}^{T} P(y_j | y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji} \mathbf{h}_i) P(\mathbf{a}_j | y_{<j}, \mathbf{a}_{<j}, \mathbf{x})$$

Attention as a Latent Variable

- Recall that

$$P(y_j, \mathbf{a}_j | y_{<j}, \mathbf{a}_{<j}, \mathbf{x})$$
$$= P(y_j | y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i) P(\mathbf{a}_j | y_{<j}, \mathbf{a}_{<j}, \mathbf{x})$$

- But we do not observe attentions $\mathbf{a}$
- We need to marginalize $\mathbf{a}$:

$$\log P(\mathbf{y}|\mathbf{x}) = \log \int_{\mathbf{a}} \prod_{j=1}^{T} P(y_j | y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i) P(\mathbf{a}_j | y_{<j}, \mathbf{a}_{<j}, \mathbf{x})$$

- Normal attention is a special case if we set:

$$P(\mathbf{a}_j | y_{<j}, \mathbf{a}_{<j}, \mathbf{x})$$
$$= \delta(f(y_{<i}, \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_S\}))$$

Attention as a Latent Variable: Training

- Recall that

$$\log P(\mathbf{y}|\mathbf{x}) = \log \int_{\mathbf{a}} \prod_{j=1}^{T} P(y_j|y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i) P(\mathbf{a}_j|y_{<j}, \mathbf{a}_{<j}, \mathbf{x})$$

Attention as a Latent Variable: Training

- Recall that

$$\log P(\mathbf{y}|\mathbf{x}) = \log \int_{\mathbf{a}} \prod_{j=1}^{T} P(y_j|y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i) P(\mathbf{a}_j|y_{<j}, \mathbf{a}_{<j}, \mathbf{x})$$

- Generally intractable due to the integral

Attention as a Latent Variable: Training

- Recall that

$$\log P(\mathbf{y}|\mathbf{x}) = \log \int_{\mathbf{a}} \prod_{j=1}^{T} P(y_j|y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i) P(\mathbf{a}_j|y_{<j}, \mathbf{a}_{<j}, \mathbf{x})$$

- Generally intractable due to the integral
- Use VAE by introducing an approximate posterior $Q(\mathbf{a}|\mathbf{y}, \mathbf{x})$:

Attention as a Latent Variable: Training

- Recall that

$$\log P(\mathbf{y}|\mathbf{x}) = \log \int_{\mathbf{a}} \prod_{j=1}^{T} P(y_j|y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i) P(\mathbf{a}_j|y_{<j}, \mathbf{a}_{<j}, \mathbf{x})$$

- Generally intractable due to the integral
- Use VAE by introducing an approximate posterior $Q(\mathbf{a}|\mathbf{y}, \mathbf{x})$:

$\log P(\mathbf{y}|\mathbf{x})$

Attention as a Latent Variable: Training

- Recall that

$$\log P(\mathbf{y}|\mathbf{x}) = \log \int_{\mathbf{a}} \prod_{j=1}^{T} P(y_j|y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i) P(\mathbf{a}_j|y_{<j}, \mathbf{a}_{<j}, \mathbf{x})$$

- Generally intractable due to the integral
- Use VAE by introducing an approximate posterior $Q(\mathbf{a}|\mathbf{y}, \mathbf{x})$:

$$\log P(\mathbf{y}|\mathbf{x})$$

$$= \log \int_{\mathbf{a}} Q(\mathbf{a}) \frac{1}{Q(\mathbf{a})} \prod_{j=1}^{T} P(y_j|y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i) P(\mathbf{a}_j|y_{<j}, \mathbf{a}_{<j}, \mathbf{x})$$

Attention as a Latent Variable: Training

- Recall that

$$\log P(\mathbf{y}|\mathbf{x}) = \log \int_{\mathbf{a}} \prod_{j=1}^{T} P(y_j|y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i) P(\mathbf{a}_j|y_{<j}, \mathbf{a}_{<j}, \mathbf{x})$$

- Generally intractable due to the integral
- Use VAE by introducing an approximate posterior $Q(\mathbf{a}|\mathbf{y}, \mathbf{x})$:

$$\log P(\mathbf{y}|\mathbf{x})$$

$$= \log \int_{\mathbf{a}} Q(\mathbf{a}) \frac{1}{Q(\mathbf{a})} \prod_{j=1}^{T} P(y_j|y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i) P(\mathbf{a}_j|y_{<j}, \mathbf{a}_{<j}, \mathbf{x})$$

$$\geq \int_{\mathbf{a}} Q(\mathbf{a}) \log \frac{1}{Q(\mathbf{a})} \prod_{j=1}^{T} P(y_j|y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i) P(\mathbf{a}_j|y_{<j}, \mathbf{a}_{<j}, \mathbf{x})$$

Attention as a Latent Variable: Training

- Recall that

$$\log P(\mathbf{y}|\mathbf{x}) = \log \int_{\mathbf{a}} \prod_{j=1}^{T} P(y_j|y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i) P(\mathbf{a}_j|y_{<j}, \mathbf{a}_{<j}, \mathbf{x})$$

- Generally intractable due to the integral
- Use VAE by introducing an approximate posterior $Q(\mathbf{a}|\mathbf{y}, \mathbf{x})$:

$$\begin{aligned}
&\log P(\mathbf{y}|\mathbf{x}) \\
&= \log \int_{\mathbf{a}} Q(\mathbf{a}) \frac{1}{Q(\mathbf{a})} \prod_{j=1}^{T} P(y_j|y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i) P(\mathbf{a}_j|y_{<j}, \mathbf{a}_{<j}, \mathbf{x}) \\
&\geq \int_{\mathbf{a}} Q(\mathbf{a}) \log \frac{1}{Q(\mathbf{a})} \prod_{j=1}^{T} P(y_j|y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i) P(\mathbf{a}_j|y_{<j}, \mathbf{a}_{<j}, \mathbf{x}) \\
&= \mathop{\mathbb{E}}_{\mathbf{a}\sim Q}[-\log Q(\mathbf{a}) + \sum_{j=1}^{T} \log P(y_j|y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i) + \log P(\mathbf{a}_j|y_{<j}, \mathbf{a}_{<j}, \mathbf{x})]
\end{aligned}$$

Attention as a Latent Variable: Training

- Recall that

$$\log P(\mathbf{y}|\mathbf{x})$$

$$= \mathop{\mathbb{E}}_{\mathbf{a}\sim Q}[-\log Q(\mathbf{a}) + \sum_{j=1}^{T} \log P(y_j|y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i) + \log P(\mathbf{a}_j|y_{<j}, \mathbf{a}_{<j}, \mathbf{x})]$$

$$= \underbrace{\mathop{\mathbb{E}}_{\mathbf{a}\sim Q}[\sum_{j=1}^{T} \log P(y_j|y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i)]}_{\text{Fit data}} + \underbrace{KL(Q(\mathbf{a})||P(\mathbf{a}_j|y_{<j}, \mathbf{a}_{<j}, \mathbf{x}))}_{Q(a) \text{ be close to prior } P(\mathbf{a}_j|y_{<j},\mathbf{a}_{<j},\mathbf{x})}$$

- Apply reparamterization trick by specifying reparamterizable $Q(\mathbf{a}|\mathbf{y}, \mathbf{x})$: Dirichlet or Logistic Normal

Attention as a Latent Variable: Training

- Dirichlet:

$$\mathbf{a}_j \sim \text{Dir}(\alpha_1, \cdots, \alpha_S)$$

  - Mean: $\frac{\alpha_1}{\alpha_0}, \cdots, \frac{\alpha_S}{\sum_i \alpha_0}$ where $\alpha_0 = \sum_i \alpha_i$
  - Variance: $\frac{\alpha_1(\alpha_0 - \alpha_1)}{\alpha_0^2(\alpha_0+1)}, \cdots, \frac{\alpha_S(\alpha_0 - \alpha_S)}{\alpha_0^2(\alpha_0+1)}$
  - Use inference network to generate the mean $\frac{\alpha_1}{\alpha_0}, \cdots, \frac{\alpha_S}{\sum_i \alpha_0}$ and α_0.

Attention as a Latent Variable: Training

- Dirichlet:

$$\mathbf{a}_j \sim \text{Dir}(\alpha_1, \cdots, \alpha_S)$$

  - Mean: $\frac{\alpha_1}{\alpha_0}, \cdots, \frac{\alpha_S}{\sum_i \alpha_0}$ where $\alpha_0 = \sum_i \alpha_i$
  - Variance: $\frac{\alpha_1(\alpha_0 - \alpha_1)}{\alpha_0^2(\alpha_0+1)}, \cdots, \frac{\alpha_S(\alpha_0 - \alpha_S)}{\alpha_0^2(\alpha_0+1)}$
  - Use inference network to generate the mean $\frac{\alpha_1}{\alpha_0}, \cdots, \frac{\alpha_S}{\sum_i \alpha_0}$ and α_0.

- Logistic Normal

$$\mathbf{r}_j \sim \mathcal{N}(\mu, \mathbf{\Sigma})$$

$$\mathbf{a}_j = \text{softmax}(\mathbf{r}_j)$$

  - Use inference network to generate μ and $\mathbf{\Sigma}$

Attention as a Latent Variable: Training

- Recall that

$$\log P(\mathbf{y}|\mathbf{x})$$

$$= \mathop{\mathbb{E}}_{\mathbf{a}\sim Q}[-\log Q(\mathbf{a}) + \sum_{j=1}^{T} \log P(y_j|y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i) + \log P(\mathbf{a}_j|y_{<j}, \mathbf{a}_{<j}, \mathbf{x})]$$

Attention as a Latent Variable: Training

- Recall that

$$\log P(\mathbf{y}|\mathbf{x})$$

$$= \mathop{\mathbb{E}}_{\mathbf{a}\sim Q}[-\log Q(\mathbf{a}) + \sum_{j=1}^{T} \log P(y_j|y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i) + \log P(\mathbf{a}_j|y_{<j}, \mathbf{a}_{<j}, \mathbf{x})]$$

- Sampling from Q is equivalent to sampling from a simple distribution $\mathcal{U}$ followed by transformation g_ϕ:

$$\epsilon_j \sim \mathcal{U}$$

$$\mathbf{a}_j = g_\phi(\epsilon_j)$$

Attention as a Latent Variable: Training

- Recall that

$$\log P(\mathbf{y}|\mathbf{x})$$

$$= \mathop{\mathbb{E}}_{\mathbf{a}\sim Q}[-\log Q(\mathbf{a}) + \sum_{j=1}^{T} \log P(y_j|y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i) + \log P(\mathbf{a}_j|y_{<j}, \mathbf{a}_{<j}, \mathbf{x})]$$

- Sampling from Q is equivalent to sampling from a simple distribution $\mathcal{U}$ followed by transformation g_ϕ:

$$\epsilon_j \sim \mathcal{U}$$

$$\mathbf{a}_j = g_\phi(\epsilon_j)$$

- The objective can be written as

$$\mathop{\mathbb{E}}_{\epsilon_1,\cdots,\epsilon_T\sim\mathcal{U}} \sum_{j=1}^{T} \log P(y_j|y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S}(g_\phi(\epsilon_j))_i\mathbf{h}_i) + \log P(g_\phi(\epsilon_j)|y_{<j}, \mathbf{a}_{<j}, \mathbf{x}) + \mathcal{H}(Q)$$

Attention as a Latent Variable: Inference

- Intractable to decode directly due to the integral w.r.t. $\mathbf{a}$
- First, we can find the joint argmax of $\mathbf{a}$ and $\mathbf{y}$ using beam search (stepwise) and back-propagation for optimizing w.r.t. $\mathbf{a}$
- Alternatively, we can use the model for rescoring

Attention as a Latent Variable: Inference

- Intractable to decode directly due to the integral w.r.t. $\mathbf{a}$
- First, we can find the joint argmax of $\mathbf{a}$ and $\mathbf{y}$ using beam search (stepwise) and back-propagation for optimizing w.r.t. $\mathbf{a}$
- Alternatively, we can use the model for rescoring

Attention as a Latent Variable: Network Structure

- $P(y_j|y_{<j}, \mathbf{a}_{<j}, \sum_{i=1}^{S} a_{ji}\mathbf{h}_i)$: vanilla decoder
- $P(\mathbf{a}_j|y_{<j}, \mathbf{a}_{<j}, \mathbf{x})$: vanilla attention network: the mean of $\mathbf{a}_j$, MLP with pooling over time: parameter controlling variance (σ^2 in logistic normal or α_0 for Dirichlet).
- Inference network Q: normalized dot-products of embeddings: mean, MLP with pooling: variance

Attention as a Latent Variable: Conclusion

The proposed framework enjoys three-fold benefits:

- alleviate decoder's burden to learn alignment model with a proper approximate attention posterior
- the attention network gets better supervision with the KL term
- more flexible compared to vanilla attention

References I

- Kingma, Diederik P. et al. (2014). "Semi-Supervised Learning with Deep Generative Models". In: *CoRR* abs/1406.5298. arXiv: 1406.5298. URL: http://arxiv.org/abs/1406.5298.
- Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning (2015). "Effective Approaches to Attention-based Neural Machine Translation". In: *Proceedings of EMNLP*.
- Vinyals, Oriol, Meire Fortunato, and Navdeep Jaitly (2015). "Pointer Networks". In: *Proceedings of NIPS*.