Paper Reading

Xinwei Geng

2018-11-21

Posterior Attention Models for Sequence to Sequence Learning

ICLR 2019

Score: 7 8 9

Joint Distribution For Attention And Output Variables

- Pr(y|x) => conditional distribution of an output sequence y = y_1, . . . , y_n, given an input sequence x = x_1, . . . , x_m
- a_t => hidden variable, called the attention variable, denotes which part of x_1:m the output y_t depends on

$$\Pr(\mathbf{y}|\mathbf{x}_{1:m}) = \sum_{\mathbf{a}} \Pr(\mathbf{y}, \mathbf{a}|\mathbf{x}_{1:m}) = \sum_{a_1,\ldots,a_n} \Pr(y_1, \ldots, y_n, a_1, \ldots, a_n|\mathbf{x}_{1:m})$$

- Existing attention-based encoder-decoder model

$$\Pr(\mathbf{y}|\mathbf{x}_{1:m}) = \prod_{t=1}^{n} \Pr(y_t|\mathbf{s}_t, \mathbf{c}_t = \sum_a P_t(a)\mathbf{x}_a) \qquad \Pr(a|\mathbf{x}_{1:m}, \mathbf{s}_t) \propto e^{A_\theta(\mathbf{x}_a, \mathbf{s}_t)}$$

- Drawbacks
  - the interaction among multiple a_t is not expressed statistically
  - the influence of a_t on y_t by diffusing the inputs is less than satisfactory
- => present a statistically sounder model of the interaction of the various attention and output variables

Posterior Attention Models

- Decompose joint distribution with chain rule jointly on both a and y

$$P(\mathbf{y}) = \sum_{\mathbf{a}} P(\mathbf{y}, \mathbf{a}) = \sum_{\mathbf{a}_{<n}, a_n} P(y_n|\mathbf{y}_{<n}, \mathbf{a}_{<n}, a_n) P(a_n|\mathbf{y}_{<n}, \mathbf{a}_{<n}) P(\mathbf{y}_{<n}, \mathbf{a}_{<n})$$

  - Pr(y|x_1:m) = > P(y)
  - P(y_t|y_<t, a_<n, a_n) = P(y_t|y_<t, a_n) => assumption that the output y_t is dependent only on a_t and previous outputs y_<t and is independent of all other attention variables

$$\begin{aligned}
P(\mathbf{y}) &= \sum_{a_n} P(y_n|\mathbf{y}_{<n}, a_n) \sum_{\mathbf{a}_{<n}} P(a_n|\mathbf{y}_{<n}, \mathbf{a}_{<n}) P(\mathbf{y}_{<n}, \mathbf{a}_{<n}) \\
&= P(\mathbf{y}_{<n}) \sum_{a_n} P(y_n|\mathbf{y}_{<n}, a_n) \sum_{\mathbf{a}_{<n}} P(a_n|\mathbf{a}_{<n}, \mathbf{y}_{<n}) \frac{P(\mathbf{y}_{<n}, \mathbf{a}_{<n})}{P(\mathbf{y}_{<n})} \\
&= P(\mathbf{y}_{<n}) \sum_{a_n} P(y_n|\mathbf{y}_{<n}, a_n) \sum_{\mathbf{a}_{<n}} P(a_n|\mathbf{a}_{<n}, \mathbf{y}_{<n}) P(\mathbf{a}_{<n}|\mathbf{y}_{<n}) \\
&= \prod_{t=1}^{n} \sum_{a_t} P(y_t|\mathbf{y}_{<t}, a_t) \sum_{\mathbf{a}_{<t}} P(a_t|\mathbf{a}_{<t}, \mathbf{y}_{<t}) P(\mathbf{a}_{<t}|\mathbf{y}_{<t})
\end{aligned}$$

- Prior_t(a) => Prior attention

$$\sum_{\mathbf{a}_{<t}} P(a_t|\mathbf{a}_{<t}, \mathbf{y}_{<t}) P(\mathbf{a}_{<t}|\mathbf{y}_{<t}) = P(a_t|\mathbf{y}_{<t})$$

  - In existing models, this attention is computed independently at each step using the RNN state
  - propose to expand out the expressions and more carefully capture the dependencies among attention variables.

Computation Of Attention Distribution

- Use the decoder RNN to approximate the dependence on the attention and outputs before the immediate past

$$P(a_t|\mathbf{a}_{<t}, \mathbf{y}_{<t}) \approx P(a_t|\mathbf{s}_{t-1}, a_{t-1}, y_{t-1})$$

$$\text{Prior}(a_t) = \sum_{\mathbf{a}_{<t}} P(a_t|\mathbf{a}_{<t}, \mathbf{y}_{<t}) P(\mathbf{a}_{<t}|\mathbf{y}_{<t}) \approx \sum_{a_{t-1}} P(a_t|\mathbf{s}_{t-1}, a_{t-1}, y_{t-1}) P(a_{t-1}|\mathbf{y}_{<t})$$

  - the posterior attention Postr(a_t−1)=> the attention distribution after observing the output label at that step, and not just the previous steps as in prior attention

$$P(a_{t-1}|\mathbf{y}_{<t}) = P(a_{t-1}|\mathbf{y}_{<(t-1)}, y_{t-1})$$

$$\begin{aligned} \text{Postr}_t(a_t) &= P(a_t|\mathbf{y}_{<t}, y_t) = \frac{P(y_t|\mathbf{y}_{<t}, a_t) P(a_t|\mathbf{y}_{<t})}{P(y_t|\mathbf{y}_{<t})} = \frac{P(y_t|\mathbf{y}_{<t}, a_t) \text{Prior}_t(a_t)}{P(y_t|\mathbf{y}_{<t})} \\ \text{Prior}_t(a_t) &= \sum_{a_{t-1}} P(a_t|\mathbf{s}_{t-1}, a_{t-1}, y_{t-1}) \text{Postr}_{t-1}(a_{t-1}) \end{aligned}$$

  - Uses the same decoder RNN to absorb the posterior attention of the previous step

$$\text{Prior}_t(a_t) = \sum_{a'} P(a_t|\mathbf{s}_{t-1}, y_{t-1}, a') \text{Postr}_{t-1}(a') \approx P(a_t|\mathbf{s}_{t-1}, y_{t-1}, \sum_{a'} \text{Postr}_{t-1}(a') x_{a'}) \approx P(a_t|\mathbf{s}_t)$$

$$\mathbf{s}_t = \text{RNN}(\mathbf{s}_{t-1}, \textstyle\sum_{a'} \text{Postr}_{t-1}(a') x_{a'}, y_{t-1})$$

Couple Adjacent Attention

- Adjacent attention

$$logP(a_t|\mathbf{s}_{t-1}, a_{t-1}) = k(a_t, a_{t-1}) + A_\theta(x_{a_t}, \mathbf{s}_{t-1})$$

  - A is the attention logit computed from the previous RNN step
  - k(a_t, a_t-1) is is the attention coupling energy

- Two strategies
  - Proximity biased coupling => natural bias towards attending on inputs where attention has focused recently

$$\mathbb{I}(|a_t - a_{t-1}| < 3)\delta_{a_t - a_{t-1}}$$

  - Monotonicity biased coupling => This model biases attention towards a monotonic attention which keeps moving ahead

$$\mathbb{I}(a_t > a_{t-1})\delta^{a_t - a_{t-1} - 1}$$

Put it all together

$$\Pr(\mathbf{y}|\mathbf{x}_{1:m}) = \prod_{t=1}^{n} \Pr(y_t|\mathbf{s}_t, \sum_{a=1}^{m} P_t(a)\mathbf{x}_a) \quad (8)$$

$$\mathbf{s}_{t+1} = \text{RNN}(\mathbf{s}_t, y_t, \sum_{a} P_t(a)\mathbf{x}_a) \quad (9)$$

$$P_t(a) = \frac{e^{A_\theta(\mathbf{x}_a, \mathbf{s}_t)}}{\sum_{r=1}^{m} e^{A_\theta(\mathbf{x}_r, \mathbf{s}_t)}} \quad (10)$$

$$\Pr(\mathbf{y}|\mathbf{x}_{1:m}) = \prod_{t=1}^{n} \sum_{a=1}^{m} P(y_t|\mathbf{s}_t, \mathbf{x}_j)\text{Prior}_t(a) \quad (11)$$

$$\mathbf{s}_{t+1} = \text{RNN}(\mathbf{s}_t, y_t, \sum_{a} \text{Postr}_t(a)\mathbf{x}_a) \quad (12)$$

$$\text{Postr}_t(a) = \frac{P(y_t|\mathbf{s}_t, x_a)\text{Prior}_t(a)}{\sum_{a'} P(y_t|\mathbf{s}_t, x_{a'})\text{Prior}_t(a')} \quad (13)$$

$$\text{Prior}_t(a_t) = \sum_{a'} P(a_t|\mathbf{s}_{t-1}, a')\text{Postr}_{t-1}(a') \quad (14)$$

$$P(a_t|\mathbf{s}_{t-1}, a') = \text{See Section 2.2.1} \quad (15)$$

Machine Translation

Soft: This is the standard soft attention mechanism with Luong attention.

Sparse: This is the sparse-attention model presented in Niculae & Blondel (2017).

Postr-Joint: This is our default posterior attention network as described in 2.2.1 . We try two more variants of PAMbased on explicit coupling called

Mono-Postr-Joint: which refers to the monotonic biased model in 2.2.1

Prox-Postr-Joint: which refers to the explicitly coupled proximity biased model described in 2.2.1

Prior-Joint: This is our model minus the posterior attention. That is, we use joint attention output distribution as in Eq 11 but prior and RNN updates are as in soft attention.

- Overall comparsion
  - all Postr-Joint variants and Prior-Joint outperform soft attention and sparse-attention by large margins
  - models with posterior attention show improvement over those which use prior attention
- Comparing Attention Coupling
  - gains over Postr-Joint by explicitly modeling attention coupling
  - For language-pairs with a natural monotonic alignment like German-English, Mono-Postr-Joint slightly outperforms other model
  - English-Vietnamese is a more non-monotonic pair and as expected we do not find gains by incorporating a monotonic bias.

Dataset	Attention	PPL	BLEU	
			B=4	B=10
IWSLT14 DE-EN	Soft	9.61	28.6	28.5
	Sparse	9.85	28.4	28.0
	Prior-Joint	8.47	29.7	29.6
	Postr-Joint	8.51	29.8	29.7
	Mono-Postr-Joint	**8.23**	**30**	29.9
	Prox-Postr-Joint	8.26	29.8	29.7
IWSLT14 EN-DE	Soft	10.68	24.2	24.2
	Sparse	10.89	23.4	23.3
	Prior-Joint	8.72	25.4	25.3
	Postr-Joint	8.6	25.6	25.4
	Mono-Postr-Joint	**8.45**	**25.7**	25.6
	Prox-Postr-Joint	8.52	25.6	25.5
IWSLT15 EN-VI	Soft	10.27	26.6	26.4
	Sparse	10.13	26.6	26.1
	Prior-Joint	9.67	27.4	27.3
	Postr-Joint	**9.11**	**27.6**	27.4
	Mono-Postr-Joint	9.52	**27.6**	27.3
	Prox-Postr-Joint	9.59	27.5	27.3
IWSLT14 VI-EN	Soft	8.30	24.7	24.6
	Sparse	8.48	24.2	23.9
	Prior-Joint	7.57	25.7	25.6
	Postr-Joint	7.34	**25.9**	25.8
	Mono-Postr-Joint	**7.14**	**25.9**	25.6
	Prox-Postr-Joint	7.26	**25.9**	25.9
WAT17 JA-EN	Soft	12.46	18.9	18.5
	Sparse	14.18	17.5	16.8
	Prior-Joint	10.00	20.6	20.2
	Postr-Joint	9.96	20.5	20.3
	Mono-Postr-Joint	9.98	20.7	20.5
	Prox-Postr-Joint	**9.78**	**20.9**	20.5

Anecdotal Examples

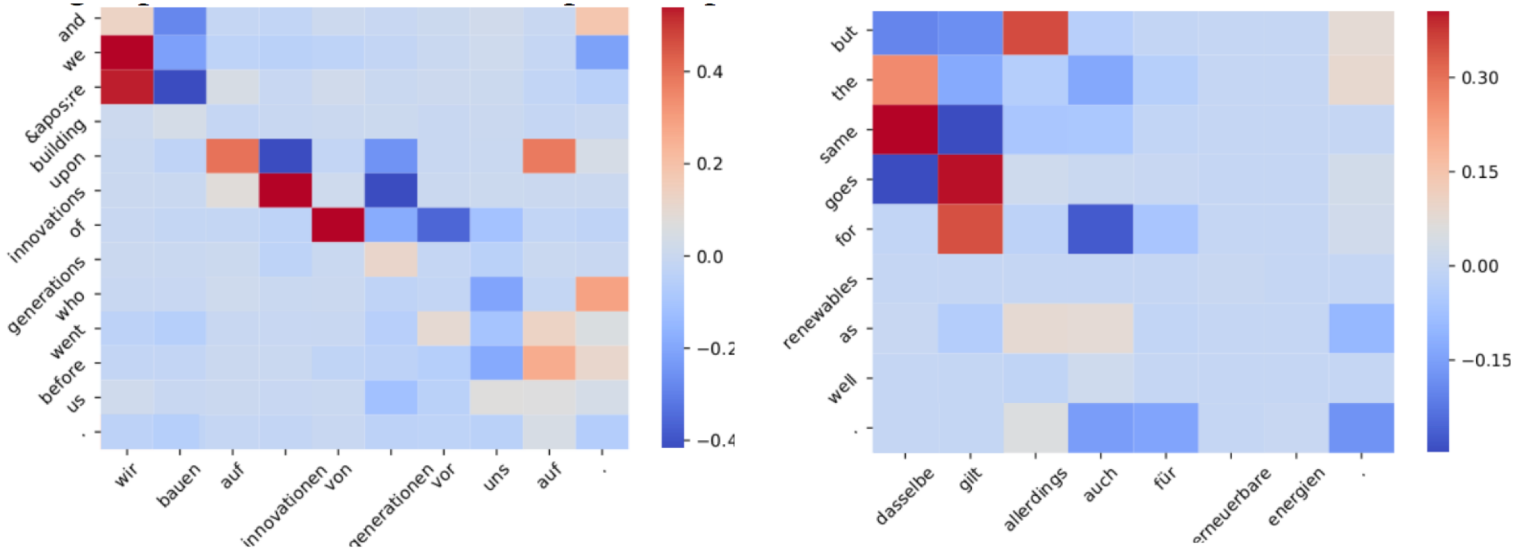
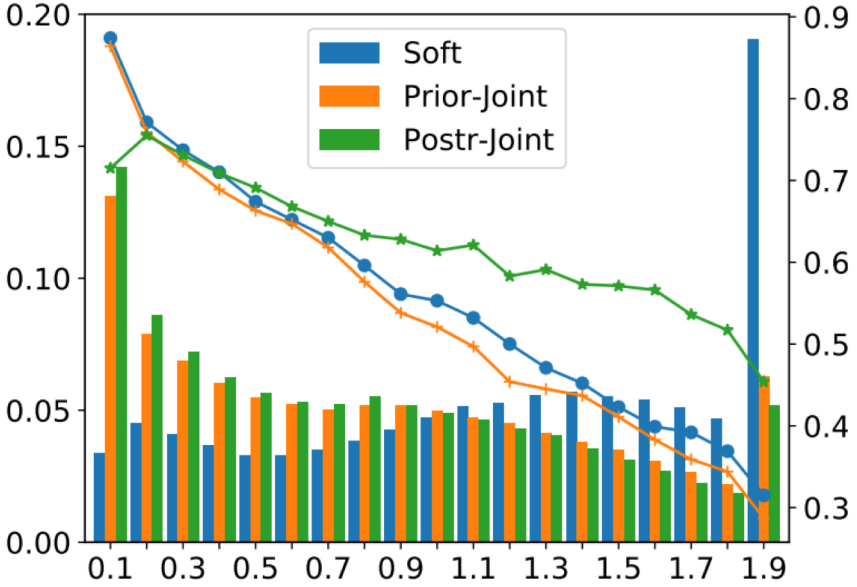


Figure 2: Heatmap of differences between Posterior-Attention (Red) and Soft-Attention (Blue). Mark the corrected red alignments for 'innovation' and 'but the same'

Attention Entropy Vs Accuracy



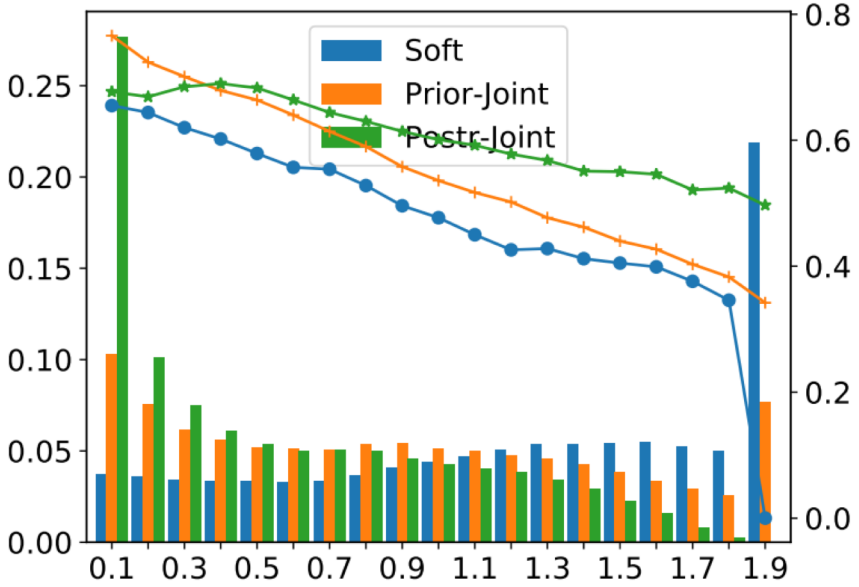
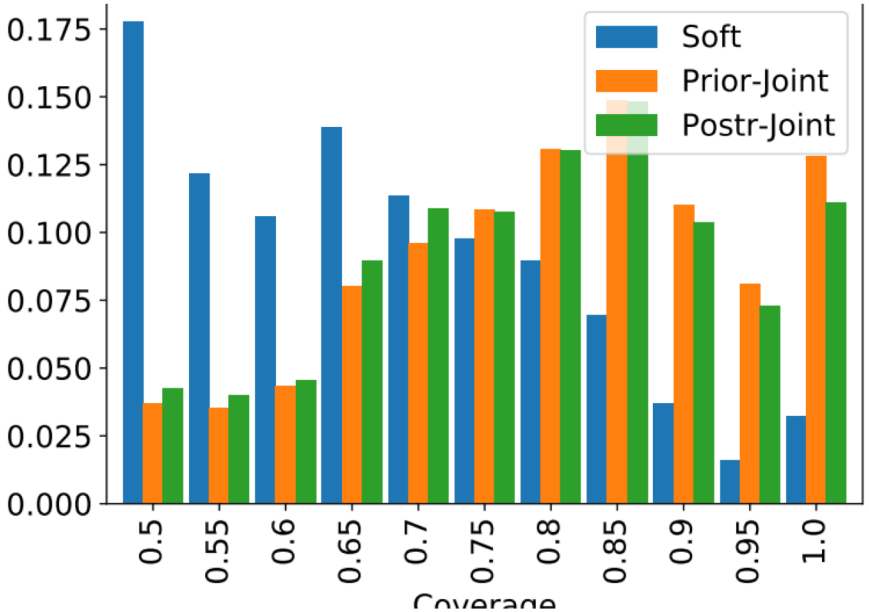


Figure 3: Variation of accuracy and histogram of attention entropy on De-En (left) and En-De (right) . Note the smoother accuracy decay in Postr-Joint and the entropy distibution for Sot-Attention

Fraction of covered tokens



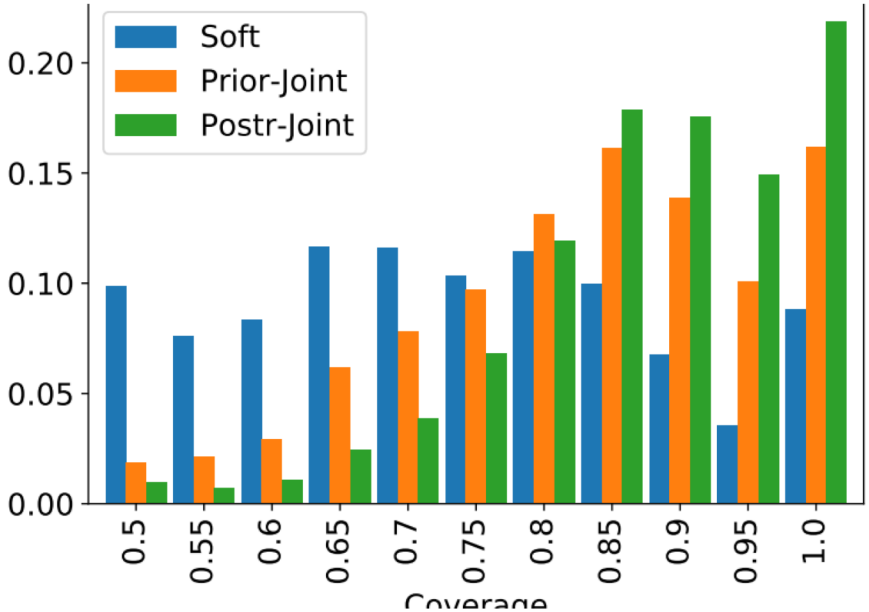


Figure 4: Coverage for different attention models on the En-De (left) and De-En(right) tasks

Alignment accuracy

- Used the RWTH German-English dataset which provides alignment information manually tagged by experts

Attention	AER
Soft	0.449
Prior-Joint	0.502
Postr-Joint	**0.583**

Thanks & QA