Surprisingly Easy Hard-Attention for Sequence to Sequence Learning

Shiv Shankar[*]
IIT Bombay
shiv_shankar@iitb.ac.in

Siddhant Garg[*]
IIT Bombay
sidgarg@cs.wisc.edu

Sunita Sarawagi
IIT Bombay
sunita@iitb.ac.in

Abstract

In this paper we show that a simple beam approximation of the joint distribution between attention and output is an easy, accurate, and efficient attention mechanism for sequence to sequence learning. The method combines the advantage of sharp focus in hard attention and the implementation ease of soft attention. On five translation and two morphological inflection tasks we show effortless and consistent gains in BLEU compared to existing attention mechanisms.

1 Introduction

In structured input-output models as used in tasks like translation and image captioning, the attention variable decides which part of the input aligns to the current output. Many attention mechanisms have been proposed (Xu et al., 2015; Bahdanau et al., 2014; Luong et al., 2015; Martins and Astudillo, 2016) but the de facto standard is a soft attention mechanism that first assigns attention weights to input encoder states, then computes an attention weighted 'soft' aligned input state, which finally derives the output distribution. This method is end to end differentiable and easy to implement.

Another less popular variant is hard attention that aligns each output to exactly one input state but requires intricate training to teach the network to choose that state. When successfully trained, hard attention is often found to be more accurate (Xu et al., 2015; Zaremba and Sutskever, 2015). In NLP, a recent success has been in a monotonic hard attention setting in morphological inflection tasks (Yu et al., 2016; Aharoni and Goldberg, 2017). For general seq2seq learning, methods like SparseMax (Martins and Astudillo, 2016) and local attention (Luong et al., 2015) were proposed to bridge the gap between soft and hard attention.

In this paper we propose a surprisingly simpler alternative based on the original joint distribution between output and attention, of which existing soft and hard attention mechanisms are approximations. The joint model couples input states individually to the output like in hard attention, but it combines the advantage of end-to-end trainability of soft attention. When the number of input states is large, we propose to use a simple approximation of the full joint distribution called Beam-joint. This approximation is also easily trainable and does not suffer from the high variance of Monte-Carlo sampling gradients of hard attention.

We evaluated our model on five translation tasks and increased BLEU by 0.8 to 1.7 over soft attention, which in turn was better than hard and the recent Sparsemax (Martins and Astudillo, 2016) attention. More importantly, the training process was as easy as soft attention. For further support, we also evaluate on two morphological inflection tasks and got gains over soft and hard attention.

2 Background and Related Work

For sequence to sequence (seq2seq) learning the encoder-decoder model is the standard and we review it here. We then review related work on attention mechanisms on these models.

2.1 Attention-based Encoder Decoder Model

Let $x_1, \ldots, x_m$ denote the tokens in the input sequence that have been transformed by an encoder network to state vectors $\mathbf{x}_1, \ldots, \mathbf{x}_m$, which we jointly denote as $\mathbf{x}_{1\ldots m}$. Let $y_1, \ldots, y_n$ denote the output tokens in the target sequence. The Encoder-Decoder (ED) network factorizes $\Pr(y_1, \ldots, y_n|\mathbf{x}_{1\ldots m})$ as $\prod_{t=1}^{n} \Pr(y_t|\mathbf{x}_{1\ldots m}, \mathbf{s}_t)$ where $\mathbf{s}_t$ is a decoder state summarizing $y_1, \ldots y_{t-1}$. For each t, a hidden attention variable a_t is used to denote which part of $\mathbf{x}_{1\ldots m}$ aligns with y_t. Let $P(a_t = j|\mathbf{x}_{1\ldots m}, \mathbf{s}_t)$ denote the

[*]Both authors contributed equally to this work

probability that encoder state $\mathbf{x}_j$ is relevant for output y_t. Typically this is estimated using a softmax function over attention scores computed from $\mathbf{x}_j$ and decoder state $\mathbf{s}_t$ as follows.

$$P(a_t = j|\mathbf{x}_{1...m}, \mathbf{s}_t) = \frac{e^{A_\theta(\mathbf{x}_j, \mathbf{s}_t)}}{\sum_{r=1}^{m} e^{A_\theta(\mathbf{x}_r, \mathbf{s}_t)}} \quad (1)$$

where $A_\theta(.,.)$ is the attention unit that scores each input state $\mathbf{x}_j$ as per the decoder state $\mathbf{s}_t$. Thereafter, in the popular soft-attention mechanism, the attention weighted sum of the input states is used to model log likelihood for each y_t as

$$\log \Pr(y_t|\mathbf{x}_{1...m}) = \log \Pr(y_t| \sum_a P_t(a)\mathbf{x}_a) \quad (2)$$

where $P_t(a_t = j)$ is the short form for $P(a_t = j|\mathbf{x}_{1...m}, \mathbf{s}_t)$. Also, here and in the rest of the paper we drop $\mathbf{s}_t$ from $P(y_t)$ and $P_t(a)$ for ease of notation. The weighted sum $\sum_a P_t(a)\mathbf{x}_a$ is called an input context $\mathbf{c}_t$ which is fed to the decoder RNN along with y_t for computing the next state $\mathbf{s}_{t+1}$.

2.2 Related Work

We next review existing attention types.

Soft Attention is the attention method described in the previous section and is the current standard for seq2seq learning (Xu Chen, 2018; Koehn, 2017). It was proposed for translation in (Bahdanau et al., 2014) and refined further in (Luong et al., 2015). As shown in Eq 2, here each output is derived from an attention averaged input. This diffuses the coupling between the input and output. The advantage of soft attention is end to end differentiability, and fast training and inference.

Hard Attention was proposed in its current form in (Xu et al., 2015) and attends to exactly one input state for an output[1]. During training, log-likelihood is an expectation over sampled attentions:

$$\log P_t(y_t|\mathbf{x}_{1...m}) = \sum_{l=1}^{M} \log P_t(y_t|\mathbf{x}_{\tilde{a_l}}) \quad (3)$$

where $\tilde{a_1}, \ldots, \tilde{a_M}$ are sampled from the multinomial $P_t(a)$. Because of the sampling, the gradient has to be computed by Monte Carlo gradient/REINFORCE (Williams, 1992) and is subject to high variance. Many tricks are required to train hard attention and there is little standardization across implementations. Xu et al (2015) use a combination of REINFORCE and soft attention. Zaremba et al(2015) uses curriculum learning that starts as soft-attention and gradually becomes discrete. Ling& Rush (2017) aggregates multiple samples during training, and a single sampled attention while testing. However, once trained well the sharp focus on memory provided by hard-attention has been found to yield superior performance (Xu et al., 2015; Shankar and Sarawagi, 2018).

Sparse/Local Attention Many attempts have been made to bridge the gap between soft and hard attention. Luong et al (2015) proposes local attention that averages a window of input. This has been refined later to include syntax (Chen et al., 2017; Sennrich and Haddow, 2016; Chen et al., 2018). Another idea is to replace the softmax in soft attention with sparsity inducing operators (Martins and Astudillo, 2016; Niculae and Blondel, 2017). However, all sparse/local attention methods continue to compute $P(y)$ from an attention weighted sum of inputs (Eq: 2) unlike hard attention.

3 Joint Attention-Output Models

We start from an explicit joint representation of the uncertainty of the attention and output variables.

$$\log P_t(y_t|\mathbf{x}_{1...m}) = \log \sum_a P_t(a)P_t(y_t|\mathbf{x}_a) \quad (4)$$

The joint model directly couples individual input states to the output, and thus is a type of hard attention. Also, by taking an expectation, instead of a single hard attention, it enjoys differentiability as in soft-attention. We call this the **full-joint** method.

Unfortunately, either when the vocabulary or the number of encoder states (m) is large, full-joint is not practical. Existing hard and soft attentions can be viewed as its approximations that either marginalize early or hard select attention. We show a surprisingly simple alternative approximation that provides hard attention without its training complexity. Our method called **Beam-joint** deterministically selects the top-k highest attention values and approximates the full joint log probability as

$$\log P_t(y_t|\mathbf{x}_{1...m}) \approx \log \sum_{a \in \text{TopK}(P_t(a))} P_t(a)P_t(y_t|\mathbf{x}_a) \quad (5)$$

Thus, in beam-joint, we first compute the multinomial attention distribution in $O(m)$ time using

[1]Note, attention on a single input encoder state does not imply attention on a single input token because RNNs or self-attention capture the context around the token.

Eq 1, select the Top-K input positions from the multinomial, next with hard attention on each position compute K output softmax, and finally compute the attention weighted output mixture distribution. The number of output softmax is K times in normal soft-attention but the actual running time overhead is only 20–30% for translation tasks. We used the default pass-through TopK operator (which is not differentiable) and optimize the beam-approximation directly. We also experimented with a version which smoothly shifts from soft-attention to beam-attention, but found that training the beam-approximation directly leads to best results.

We show empirically that this very simple scheme is surprisingly effective compared to existing hard and soft attention over several translation tasks. Unlike sampling and variational methods that require careful tuning and exotic tricks during training, this simple scheme trains as easily as soft-attention, without significant increase in training time because even $K = 6$ works well enough.

Another reason why our 'sum of probabilities' form performs better could be the softmax barrier effect highlighted in (Yang et al., 2018). The authors argue that the richness of natural language cannot be captured in normal softmax due to the low rank constraint it imposes on input-to-output matrix. They improve performance using a Mixture of Softmax model. Our beam-joint also is a mixture of softmax and possibly achieves higher rank than a single softmax. However their mixture requires learning multiple softmax matrices, whereas ours are due to varying attention and we do not learn any extra parameters than soft attention.

4 Experiments

We compare attention models on two NLP tasks: machine translation and morphological inflection.

4.1 Machine translation

We experiment on five language pairs from three datasets: **IWSLT15 English↔Vietnamese** (Cettolo et al., 2015) which contains 133k train, 1.5k validation(tst2012) and 1.2k test(tst2013) sentence pairs respectively; **IWSLT14 German↔English** (Cettolo et al., 2014) which contains 160k train, 7.2k validation and 6.7k test sentence pairs respectively ; **Workshop on Asian Translation 2017 Japanese→English** (Nakazawa et al., 2016) which contains 2M train, 1.8k validation and 1.8k test sentence pairs respectively. We use a 2 layer bi-directional encoder and a 2 layer unidirectional decoder with 512 hidden LSTM units and 0.2 dropout rate with vanilla SGD optimizer. We base our implementation[2] on the NMT code[3] in Tensorflow. We did no special hyper-parameter tuning and used standard-softmax tuned parameters on a batch size of 64.

Comparing attention models We compare beam-joint (default $K = 6$) with standard soft and hard attention. To further dissect the reasons behind beam-joint's gains, we compare beam-joint with a sampling based approximation of full-joint called Sample-Joint that replaces the TopK in Eq 5 with K attention weighted samples. We train sample-joint as well as hard-attention with REINFORCE with 6-samples. Also to ascertain that our gains are not explained by sparsity alone, we compare with Sparsemax (Martins and Astudillo, 2016).

In Table 1 we show perplexity and BLEU with three beam sizes (B). Beam-joint significantly outperforms all other variants, including the standard soft attention by 0.8 to 1.7 BLEU points. The perplexity shows even a more impressive drop in all five datasets. Also we observe training times for beam-joint to be only 20–30% higher than soft-attention, establishing that beam-joint is both practical and more accurate.

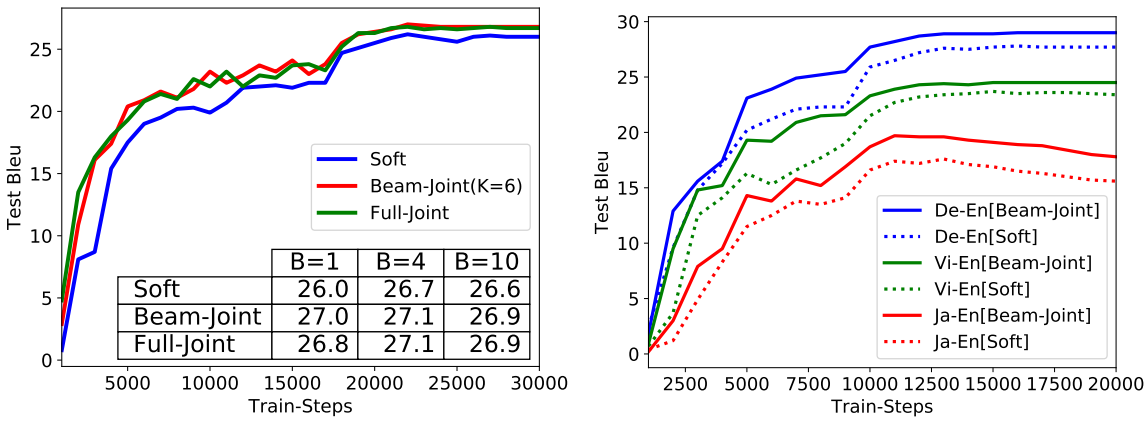
Sample-joint is much worse than beam-joint. Apart from the problem of high variance of gradients in the reinforce step, another problem is that sampling repeats states whereas TopK in beam-joint gets distinct states. Hard attention too faces training issues and performs worse than soft attention, explaining why it is not commonly used in NMT. Sample-joint is better than Hard attention, further highlighting the merits of the joint distribution. Sparsemax is competitive but marginally worse than soft attention. This is concordant with the recent experiments of (Niculae and Blondel, 2017).

Comparison with Full Joint Next we evaluate the impact of our beam-joint approximation against full-joint and soft attention. Full-joint cannot scale to large vocabularies, therefore we only compare on En-Vi with a batch size of 32. Figure 1a shows final BLEU of these methods as well as BLEU against increasing training steps. Beam-joint both converges faster and to a higher score than soft-

[2]https://github.com/sid7954/beam-joint-attention
[3]https://github.com/tensorflow/nmt

Dataset	Attention	PPL	BLEU		
			B=1	B=4	B=10
IWSLT14 DE-EN	Soft	9.61	27.7	28.6	28.5
	Hard	9.50	25.3	25.6	25.5
	Sparse	9.85	27.2	28.4	28.0
	Sample-Joint	9.96	26.3	27.8	27.8
	Beam-Joint	**8.47**	29.0	**29.7**	29.6
IWSLT14 EN-DE	Soft	10.68	23.1	24.2	24.2
	Hard	10.15	21.4	21.8	21.7
	Sparse	10.89	22.5	23.4	23.3
	Sample-Joint	10.05	22.8	23.8	23.6
	Beam-Joint	**8.72**	24.7	**25.4**	25.3
IWSLT15 EN-VI	Soft	10.27	26.0	26.6	26.4
	Hard	10.53	24.1	24.3	24.0
	Sparse	10.13	25.9	26.6	26.1
	Sample-Joint	11.00	25.8	26.3	25.9
	Beam-Joint	**9.67**	27.0	**27.4**	27.3
IWSLT14 VI-EN	Soft	8.30	23.6	24.7	24.6
	Hard	8.28	21.1	21.9	21.5
	Sparse	8.48	22.8	24.2	23.9
	Sample-Joint	8.28	22.7	24.0	23.9
	Beam-Joint	**7.57**	24.5	**25.8**	25.7
WAT17 JA-EN	Soft	12.46	17.6	18.9	18.5
	Hard	12.78	13.2	13.1	12.7
	Sparse	14.18	16.7	17.5	16.8
	Sample-Joint	13.21	16.2	18.1	17.9
	Beam-Joint	**10.00**	19.6	**20.6**	20.2

Table 1: Perplexity and test BLEU with three inference beam widths (B) on five translation tasks. Beam-joint consistently and substantially outperforms soft-attention.



(a) BLEU Vs Training steps on En-Vi

(b) BLEU Vs Training steps (Beam=1)

Figure 1: Test BLEU in various settings (Beam=1). Best viewed in color

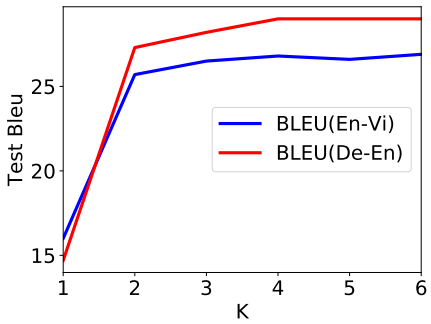


Figure 2: BLEU for Beam-Joint with increasing K. BLEU increases with K and saturates at K=5

attention. For example by 10000 steps (5 epochs), beam-joint has surpassed soft-attention by almost 2 BLEU points (20 vs 22). Moreover beam-joint tracks full-joint well, and both converge finally to similar BLEUs near 27 against 26 for soft attention. This shows that an attention-beam of size 6 suffices to approximate full joint almost perfectly.

Next, in Figure 1b, we compare beam-joint (solid lines) and soft attention (dotted lines) for convergence rates on three other datasets. For each dataset beam-joint trains faster with a consistent improvement of more than 1 BLEU.

Effect of K in Beam-joint We show the effect of K used in TopK of beam-joint in Figure 2 on the En-Vi and De-En tasks. On En-Vi BLEU increases from 16.0 to 25.7 to 26.5 as K increases from 1 to 2 to 3; and then saturates quickly. Similar behavior is observed in the other dataset. This shows that small K values like 6 suffice for translation.

We further evaluate whether the performance gain of beam-joint is due to the softmax barrier alone in Table 2. We used our models trained with K=6, and deployed them for test-time greedy decoding with K set to 1. Since the output now has only a single softmax component, this model faces the same bottleneck as soft-attention. One can observe that as expected these results are worse than beam-joint with K=6, however they still exceed soft-attention by a significant margin, demonstrating that the performance gain is not solely due to the effect of ensembling or softmax-barrier.

4.2 Morphological Inflection

To demonstrate the use of this approach beyond translation, we next consider two morphological inflection tasks. We use (Durrett and DeNero, 2013)'s dataset containing 8 inflection forms for German Nouns (de-N) and 27 forms for German Verbs (de-V). The number of training words is 2364 and 1627 respectively while the validation and test words are 200 each. We train a one layer encoder and decoder with 128 hidden LSTM units each with a dropout rate of 0.2 using Adam(Kingma and Ba, 2014) and measure 0/1 accuracy for soft, hard and full-joint attention models. Due to limited input length and vocabulary, we were able to run directly the full-joint model. We also ran the 100 units wide two layer LSTM with hard-monotonic attention provided by (Aharoni and Goldberg, 2017) labeled Hard-Mono[4]. The table below shows that even for this task full-joint scores over existing attention models[5]. The generic full-joint attention provides slight gains even over the task specific hard-monotonic attention.

Dataset	Soft	Hard	Hard-Mono	Full-Joint
de-N	85.50	85.13	85.65	**85.81**
de-V	94.91	95.04	95.31	**95.52**

Dataset	Soft	Beam-Joint (K=1)	Beam-Joint (K=6)
En-De	23.1	24.5	**24.7**
De-En	27.7	28.4	**29.0**
En-Vi	26.0	26.5	**27.0**

Table 2: Comparing soft attention with Beam-Joint using different values of K during inference. During training $K = 6$ for both Beam-Joint models.

Conclusion

In this paper we showed a simple yet effective approximation of the joint attention-output distribution in sequence to sequence learning. Our joint model consistently provides higher accuracy without significant running time overheads in five translation and two morphological inflection tasks. An interesting direction for future work is to extend beam-joint to multi-head attention architectures as in (Vaswani et al., 2017; Xu Chen, 2018).

Acknowledgements We thank NVIDIA Corporation for supporting this research by the donation of Titan X GPU.

[4]https://github.com/roeeaharoni/morphological-reinflection

[5]Our numbers are lower than earlier reported because ours use a single model whereas (Aharoni and Goldberg, 2017) and others report from an ensemble of five models.

References

Roee Aharoni and Yoav Goldberg. 2017. Morphological inflection generation with hard monotonic attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 2004–2015.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, Roldano Cattoni, and Marcello Federico. 2015. The iwslt 2015 evaluation campaign. In *IWSLT 2015, International Workshop on Spoken Language Translation*.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th iwslt evaluation campaign, iwslt 2014.

Huadong Chen, Shujian Huang, David Chiang, and Jiajun Chen. 2017. Improved neural machine translation with a syntax-aware encoder and decoder. In *ACL*.

Kehai Chen, Rui Wang, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. 2018. Syntax-directed attention for neural machine translation. *CoRR*, abs/1711.04231.

Greg Durrett and John DeNero. 2013. Supervised learning of complete morphological paradigms. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Philipp Koehn. 2017. Neural machine translation. *CoRR*, abs/1709.07809.

Jeffrey Ling and Alexander M. Rush. 2017. Coarse-to-fine attention models for document summarization. In *Proceedings of the Workshop on New Frontiers in Summarization, NFiS at EMNLP 2017, Copenhagen, Denmark*.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. *EMNLP*.

André F. T. Martins and Ramón Fernández Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *ICML*.

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. Aspec: Asian scientific paper excerpt corpus. In *LREC*.

Vlad Niculae and Mathieu Blondel. 2017. A regularized framework for sparse and structured neural attention. In *NIPS*.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *WMT*.

Shiv Shankar and Sunita Sarawagi. 2018. Labeled memory networks for online model adaptation. In *AAAI*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*.

Mia et al Xu Chen. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *ACL*.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. 2018. Breaking the softmax bottleneck: A high-rank RNN language model. In *International Conference on Learning Representations*.

Lei Yu, Jan Buys, and Phil Blunsom. 2016. Online segment to segment neural transduction. In *EMNLP*, pages 1307–1316.

Wojciech Zaremba and Ilya Sutskever. 2015. Reinforcement learning neural turing machines. *CoRR*, abs/1505.00521.