Pervasive Attention: 2D CNNs for Sequence-to-Sequence Prediction

Maha Elbayad[1,2] Laurent Besacier[1] Jakob Verbeek[2]

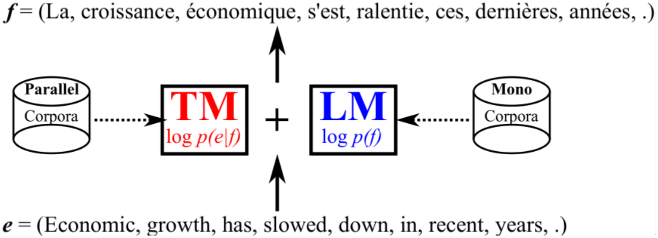
[1] Université Grenoble Alpes [2] INRIA Grenoble, France

Conference on Computational Natural Language Learning 2018



Machine translation

- Given pairs of aligned sentences (x, y) (source, target)
- Model the conditional distribution $p(y|x)$



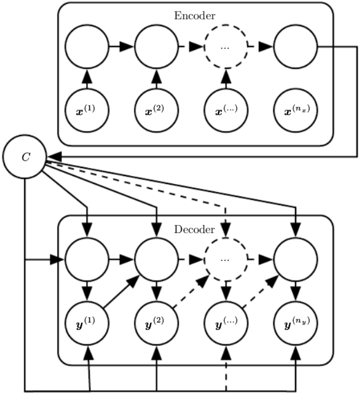
source: Kyunghyun Cho, NYU.

- Translation model: $p(y|x)$
- Language model: $p(y)$

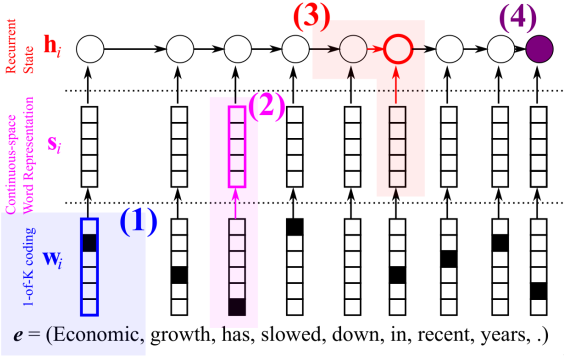
Neural machine translation

- RNN encoder-decoder models [Kalchbrenner and Blunsom, 2013, Cho et al., 2014, Sutskever et al., 2014]

$$p(y_{1:T}|x_{1:L}) = \prod_{t=1}^{T} p(y_t|y_{<t}, C(x_{1:L})) \tag{1}$$



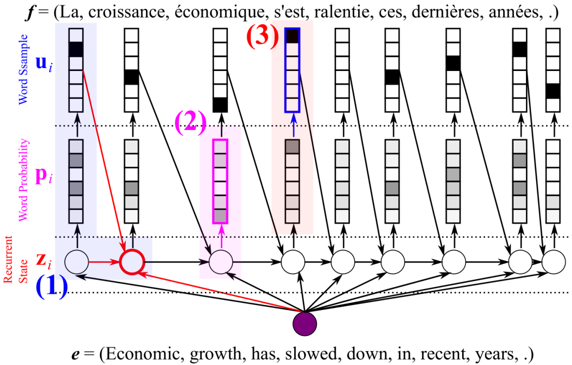
Recurrent neural encoder



Source: Kyunghyun Cho, NYU.

1. One-hot encoding: (sub)words tokens
2. Vector representation $s_t = Wx_t$, $W \in \mathbb{R}^{d \times V}$
3. Recursion: $h_t = f_\theta(h_{t-1}, s_t)$
4. Code: $C(x_{1:L})$

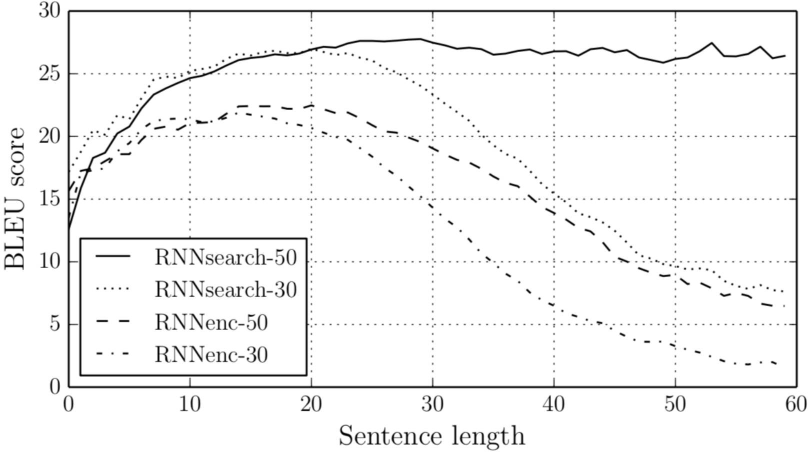
Recurrent neural decoder



Source: Kyunghyun Cho, NYU.

1. Recursion: $z_{t+1} = f_\theta(z_t, y_t, C(x_{1:L}))$
2. Emission prob.: $p(y_t|z_t) = \mathrm{SoftMax}(Ez_t)$
3. Generation: sampling, greedy, beam search

Performance vs. sentence length



"You can't cram the meaning of a whole %&!$ing sentence into a single $&!*ing vector!"

Ray Mooney @ ACL Workshop on Semantic Parsing, 2014

"You can't cram the meaning of a whole %&!$ing sentence into a single $&!*ing vector!"

Ray Mooney @ ACL Workshop on Semantic Parsing, 2014

- Ok, so how about cramming it into two vectors?!

"You can't cram the meaning of a whole %&!$ing sentence into a single $&!*ing vector!"

Ray Mooney @ ACL Workshop on Semantic Parsing, 2014

- Ok, so how about cramming it into two vectors?!

Bi-directional RNN encoder [Schuster and Paliwal, 1997]

Attention [Bahdanau et al., 2015]

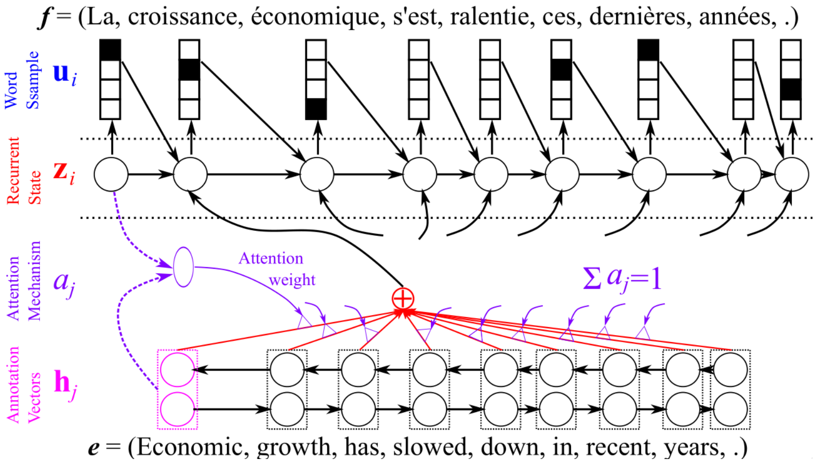
- Re-encode input given current decoder state z_t
- Use re-encoded input in decoder state update

$$z_{t+1} = f_\theta(z_t, y_t, C(x_{1:L}), A(x_{1:L}, y_{1:t})) \tag{2}$$

Attention [Bahdanau et al., 2015]

- Re-encode input given current decoder state z_t
- Use re-encoded input in decoder state update

$$z_{t+1} = f_\theta\left(z_t, y_t, C(x_{1:L}), A(x_{1:L}, y_{1:t})\right) \tag{2}$$



source: Kyunghyun Cho, NYU.

So far, so good...

So far, so good...

- Elements of state-of-the-art machine translation

So far, so good...

- Elements of state-of-the-art machine translation

1. Bi-directional RNN encoder

So far, so good...

- Elements of state-of-the-art machine translation

1. Bi-directional RNN encoder
2. RNN decoder with beam-search

So far, so good...

- Elements of state-of-the-art machine translation

1. Bi-directional RNN encoder
2. RNN decoder with beam-search
3. Attention mechanism

So far, so good...

- Elements of state-of-the-art machine translation

1. Bi-directional RNN encoder
2. RNN decoder with beam-search
3. Attention mechanism

Now let's try something else...

- No encoder

So far, so good...

- Elements of state-of-the-art machine translation

1. Bi-directional RNN encoder
2. RNN decoder with beam-search
3. Attention mechanism

Now let's try something else...

- No encoder
- No decoder

So far, so good...

- Elements of state-of-the-art machine translation

1. Bi-directional RNN encoder
2. RNN decoder with beam-search
3. Attention mechanism

Now let's try something else...

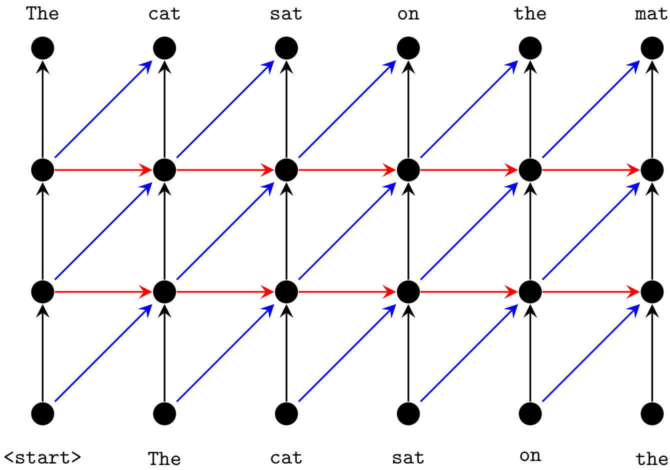
- No encoder
- No decoder
- No attention (?)

Trading depth for parallelism

- RNN: directed, shallow, unlimited receptive field with depth 1
- CNN: undirected, deep, receptive field grows by 1 each layer
  In NLP, eg. [Collobert and Weston, 2008, Kalchbrenner et al., 2014, Gehring et al., 2017b]

Trading depth for parallelism

- RNN: directed, shallow, unlimited receptive field with depth 1
- CNN: undirected, deep, receptive field grows by 1 each layer
  In NLP, eg. [Collobert and Weston, 2008, Kalchbrenner et al., 2014, Gehring et al., 2017b]

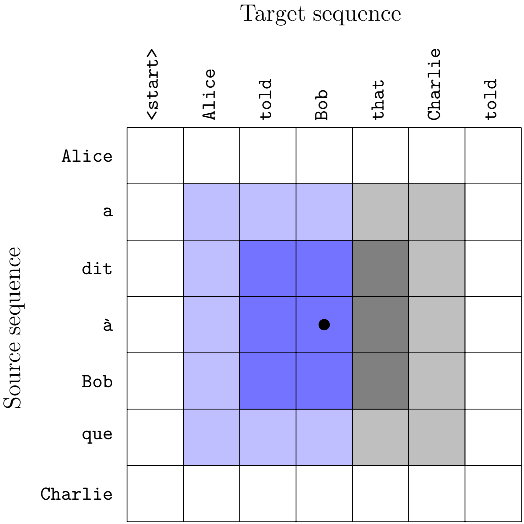


Stop cramming a sentence into a vector...

- Joint coding: input N-grams given last M output tokens

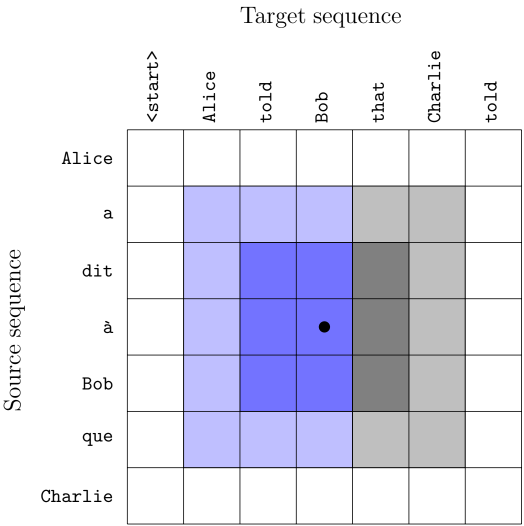
Stop cramming a sentence into a vector...

- Joint coding: input N-grams given last M output tokens



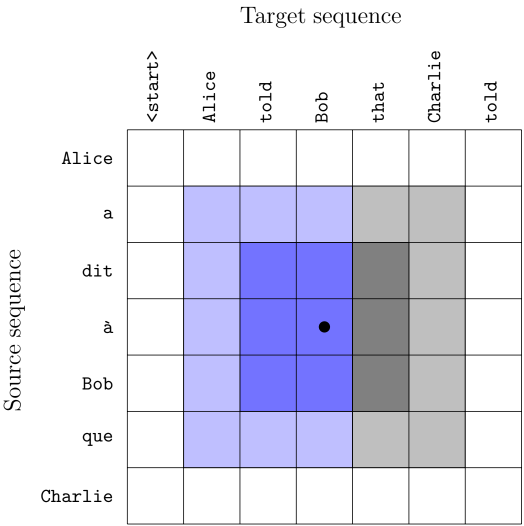
Stop cramming a sentence into a vector...

- Joint coding: input N-grams given last M output tokens
  - Receptive field: (N,M) = 1+ (2,1) × depth



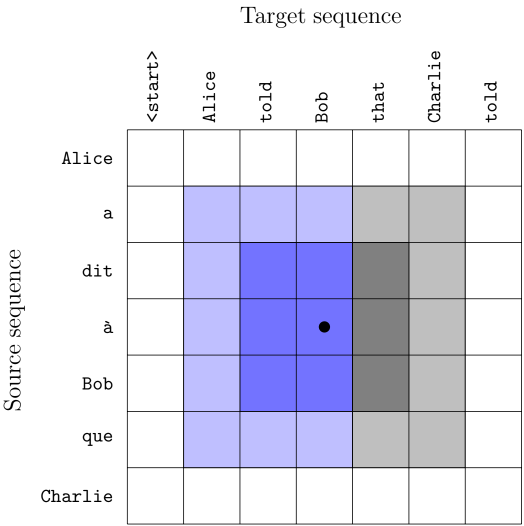
Stop cramming a sentence into a vector...

- Joint coding: input N-grams given last M output tokens
  - Receptive field: $(N,M) = 1 + (2,1) \times \text{depth}$
- Parrallel work in machine reading [Raison et al., 2018]



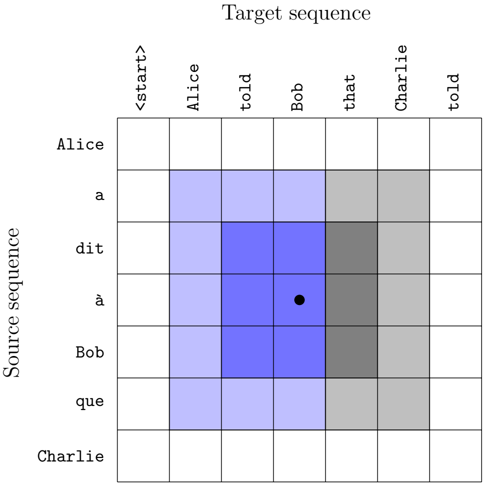
Pervasive attention

- Similar to “classic” attention: re-coding input given output
- Token-level interaction between source and target
- Present in every layer, rather than an “afterthought”



Network architecture

- Input tensor $X_{i,j} = [v_i, w_j]$ concatenates word embeddings
- 2D masked CNN layers, *e.g.* DenseNet [Huang et al., 2017]

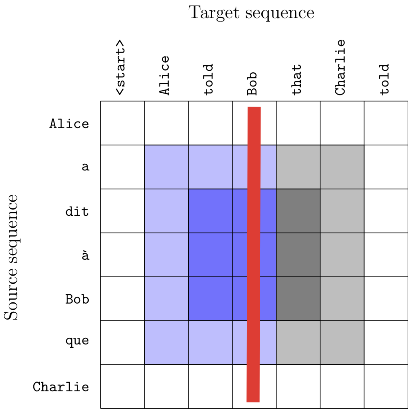


Collapsing source dimension

- Max-pool over variable-length source dimension
  - Generates one vector per target position

$$M_j = [\max_i X_{ij}^1, \ldots \max_i X_{ij}^D] \quad (3)$$

- Soft-max to predict next token at every target position



Experiments: IWSLT'14

- Translation of TED and TEDx talks
- 160k German-to-English train pairs
- Prediction at sub-word level (BPE)

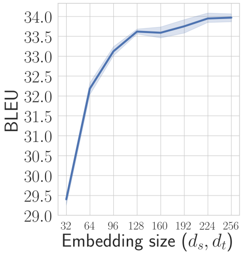
Experiments: IWSLT'14

- Translation of TED and TEDx talks
- 160k German-to-English train pairs
- Prediction at sub-word level (BPE)

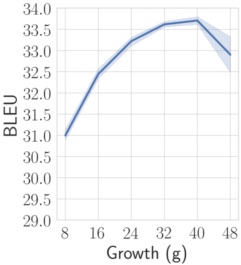
Model	BLEU	Flops$\times 10^5$	#params
Average	31.57 ± 0.11	3.63	7.18M
Max	33.70 ± 0.06	3.44	7.18M
Attention	32.09 ± 0.12	3.61	7.24M
[Max, Attn]	33.81 ± 0.03	3.51	7.24M

Our model with different pooling operators.
$(L = 24, g = 32, d_s = d_t = 128)$

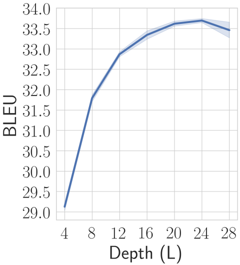
Embedding size, number of layers, and growth rate



$L = 20,\ g = 32$

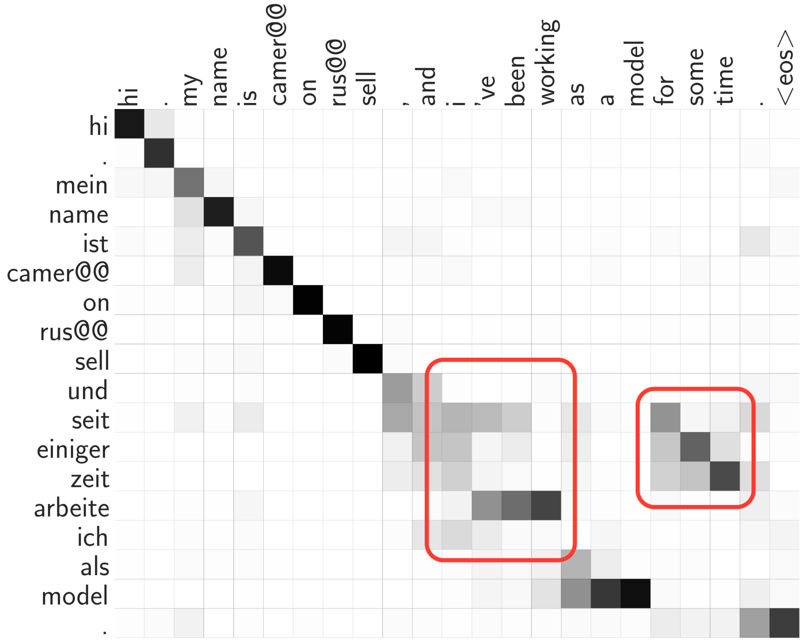


$L = 20,\ d = 128$

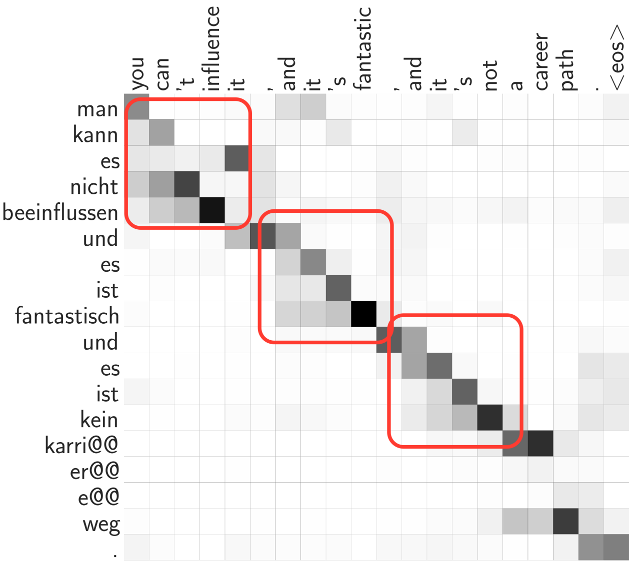


$d = 128,\ g = 32$

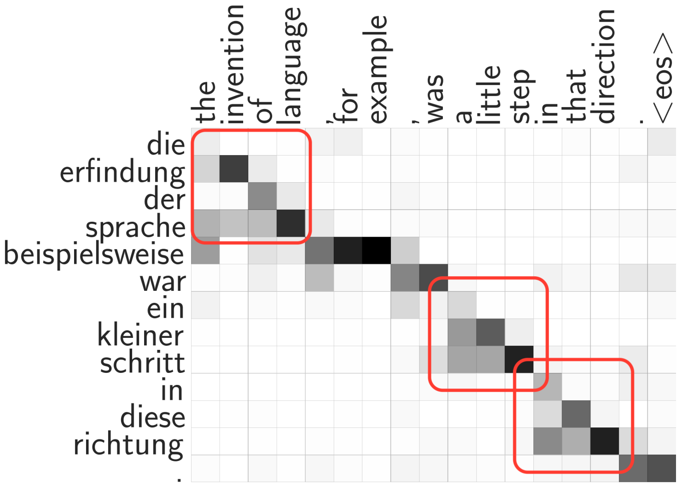
Token-level alignments from max-pooling



Token-level alignments from max-pooling



Token-level alignments from max-pooling



Comparison to the state of the art

Word-based	De-En	Flops ($\times 10^5$)	# prms	En-De	# prms
Conv-LSTM (MLE) [Bahdanau et al., 2017]	27.56				
Bi-GRU (MLE+SLE) [Bahdanau et al., 2017]	28.53				
Conv-LSTM (deep+pos) [Gehring et al., 2017a]	30.4				
NPMT + language model [Huang et al., 2018]	30.08			25.36	
BPE-based					
RNNsearch* [Bahdanau et al., 2015]	31.02	1.79	6M	25.92	7M
Varational attention [Deng et al., 2018]	33.10				
Transformer** [Vaswani et al., 2017]	32.83	3.53	59M	27.68	61M
ConvS2S** (MLE) [Gehring et al., 2017b]	32.31	1.35	21M	26.73	22M
ConvS2S (MLE+SLE) [Edunov et al., 2018]	32.84				
Pervasive Attention (this paper)	33.81± 0.03	3.51	7M	27.77± 0.1	7M

* Obtained using FairSeq.

** Obtained using author's code = FairSeq.

Conclusion

- Joint-coding approach, alternative to encoder-decoder
  - 2D CNN with masked filters

Conclusion

- Joint-coding approach, alternative to encoder-decoder
  - 2D CNN with masked filters
  - Source-target interactions pervasive in architecture

Conclusion

- Joint-coding approach, alternative to encoder-decoder
  - 2D CNN with masked filters
  - Source-target interactions pervasive in architecture
- Max-pooling generates implicit sentence alignment

Conclusion

- Joint-coding approach, alternative to encoder-decoder
  - 2D CNN with masked filters
  - Source-target interactions pervasive in architecture
- Max-pooling generates implicit sentence alignment
- Performance compares favorably to encoder-decoder models
  - Also in nr. of parameters and compute

Conclusion

- Joint-coding approach, alternative to encoder-decoder
  - 2D CNN with masked filters
  - Source-target interactions pervasive in architecture
- Max-pooling generates implicit sentence alignment
- Performance compares favorably to encoder-decoder models
  - Also in nr. of parameters and compute

- Future directions:
  - More efficient hybrid 1d-2d architectures
  - Architectures for multiple language pairs
  - Low-latency decoding

Thanks for your attention

References I

[Bahdanau et al., 2017] Bahdanau, D., Brakel, P., Xu, K., Goyal, A., Lowe, R., Pineau, J., Courville, A., and Bengio, Y. (2017).
An actor-critic algorithm for sequence prediction.
In *ICLR*.

[Bahdanau et al., 2015] Bahdanau, D., Cho, K., and Bengio, Y. (2015).
Neural machine translation by jointly learning to align and translate.
In *ICLR*.

[Cho et al., 2014] Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014).
Learning phrase representations using RNN encoder-decoder for statistical machine translation.
In *EMNLP*.

[Collobert and Weston, 2008] Collobert, R. and Weston, J. (2008).
A unified architecture for natural language processing: Deep neural networks with multitask learning.
In *ICML*.

[Deng et al., 2018] Deng, Y., Kim, Y., Chiu, J., Guo, D., and Rush, A. (2018).
Latent alignment and variational attention.
arXiv preprint arXiv:1807.03756.

[Edunov et al., 2018] Edunov, S., Ott, M., Auli, M., Grangier, D., and Ranzato, M. (2018).
Classical structured prediction losses for sequence to sequence learning.
In *NAACL*.

[Gehring et al., 2017a] Gehring, J., Auli, M., Grangier, D., and Dauphin, Y. (2017a).
A convolutional encoder model for neural machine translation.
In *ACL*.

[Gehring et al., 2017b] Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. (2017b).
Convolutional sequence to sequence learning.
In *ICML*.

References II

[Huang et al., 2017] Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. (2017).
Densely connected convolutional networks.
In *CVPR*.

[Huang et al., 2018] Huang, P., Wang, C., Huang, S., Zhou, D., and Deng, L. (2018).
Towards neural phrase-based machine translation.
In *ICLR*.

[Kalchbrenner and Blunsom, 2013] Kalchbrenner, N. and Blunsom, P. (2013).
Recurrent continuous translation models.
In *ACL*.

[Kalchbrenner et al., 2014] Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014).
A convolutional neural network for modelling sentences.
In *ACL*.

[Raison et al., 2018] Raison, M., Mazaré, P.-E., Das, R., and Bordes, A. (2018).
Weaver: Deep co-encoding of questions and documents for machine reading.
arXiv preprint arXiv:1807.03756.

[Schuster and Paliwal, 1997] Schuster, M. and Paliwal, K. (1997).
Bidirectional recurrent neural networks.
Signal Processing, 45(11):2673–2681.

[Sutskever et al., 2014] Sutskever, I., Vinyals, O., and Le, Q. (2014).
Sequence to sequence learning with neural networks.
In *NIPS*.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., and Polosukhin, I. (2017).
Attention is all you need.
In *NIPS*.