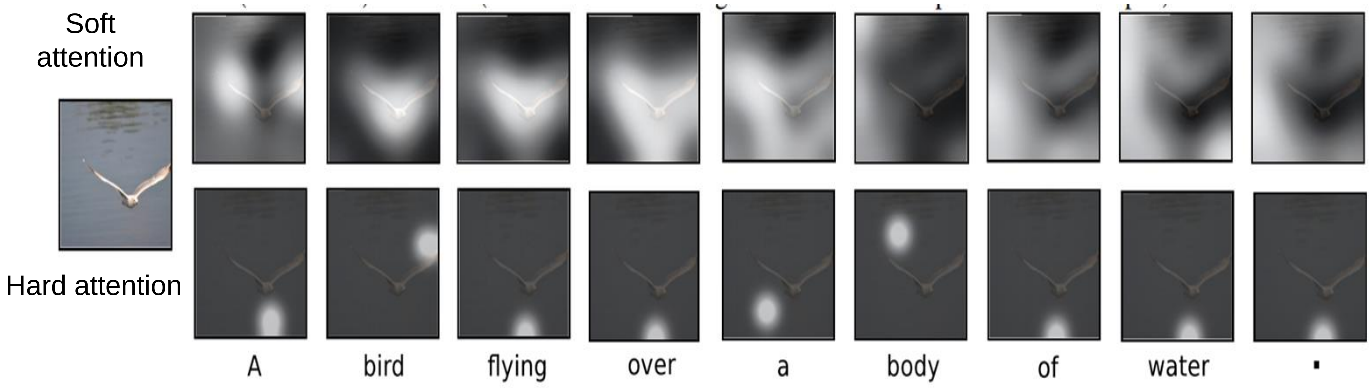
# CS194/294-129: Designing, Visualizing and Understanding Deep Neural Networks

**John Canny**

Spring 2018

Lecture 14: Translation

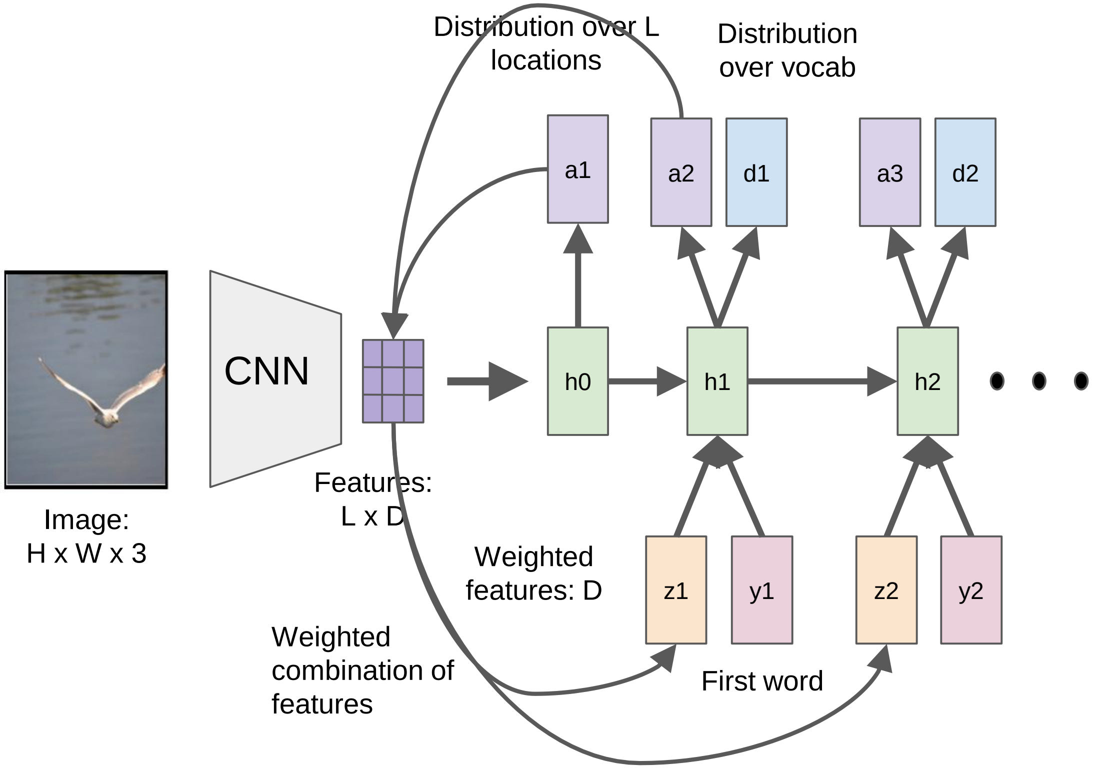
# Last Time: Soft vs Hard Attention




**Hard attention:** Attend to a single input location, can't use gradient descent, Need **reinforcement learning.**

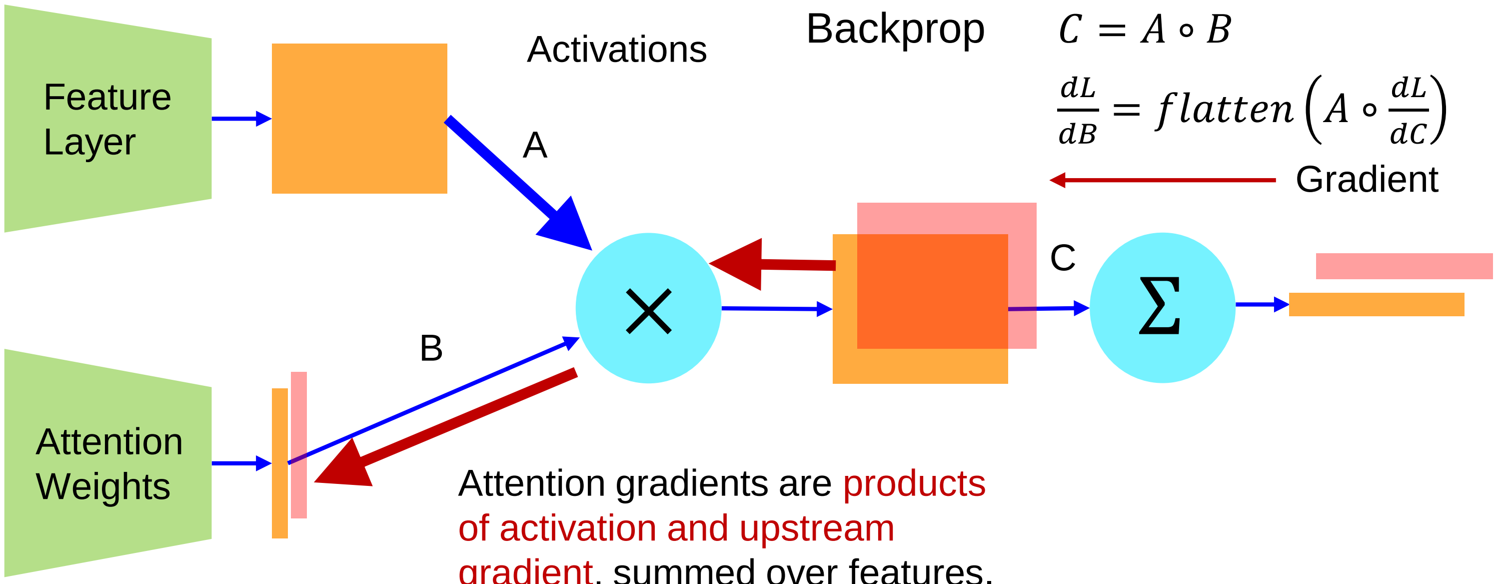
**Soft attention:** Compute a weighted combination (attention) over some inputs using an attention network. Can use backpropagation to train end-to-end.

# Last Time: Recurrent Attention for Captioning



# Last Time: Attention Mechanics: Salience

During training, the attention layer receives gradients which are the product of the upstream gradient and the feature layer activations (salience).



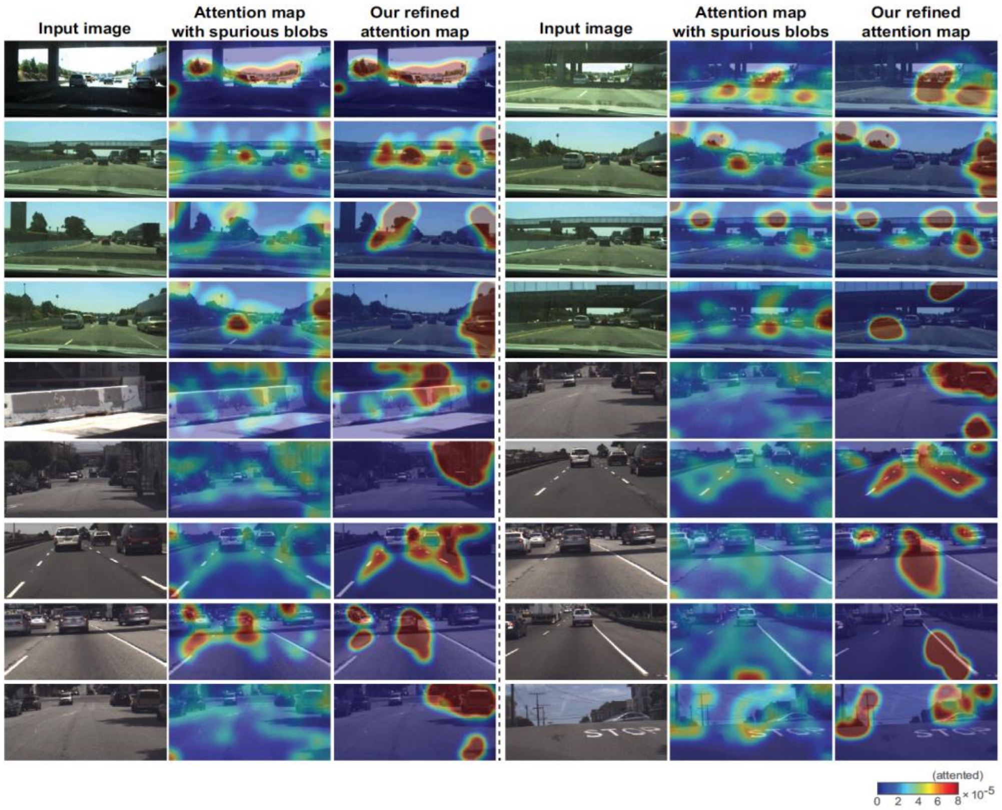

Attention gradients are products of activation and upstream gradient, summed over features.
This is a ***salience*** (importance) signal.

# Last Time: Attention and Interpretability

Attention models learn to predict salient (important) inputs.

Attention visualizations help users understand the causes of the network's behavior.

Not every attended region is actually important, but post-processing can remove regions that aren't.

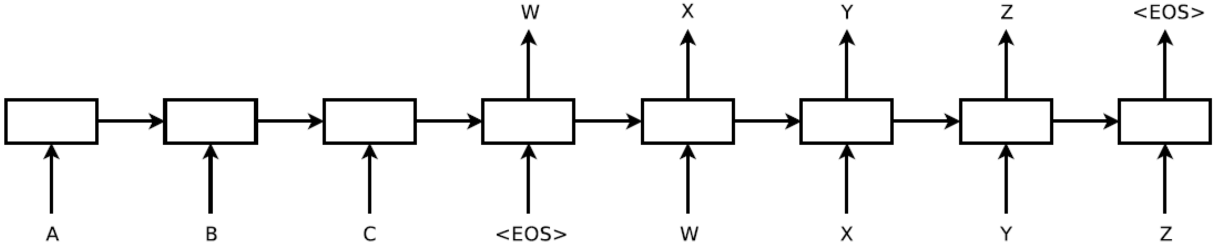


# This Time: Translation

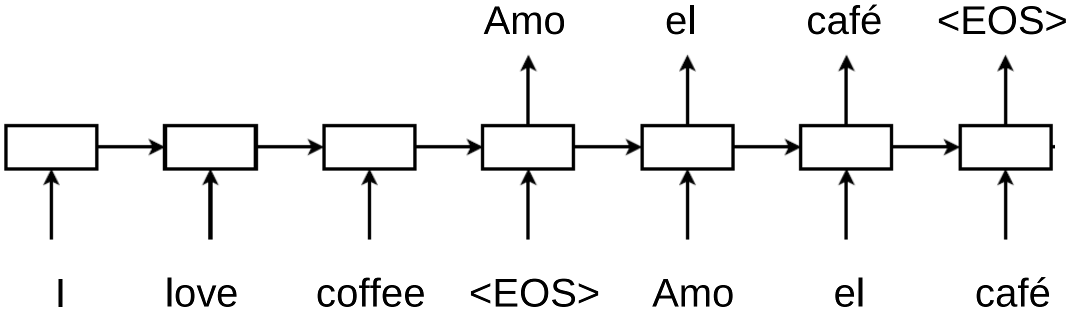
- Sequence-to-sequence translation
- Adding Attention
- Parsing as translation
- Attention only models
- English-to-English translation ?!

# Sequence-To-Sequence RNNs

An input sequence is fed to the left array, output sentence to the right array for training:

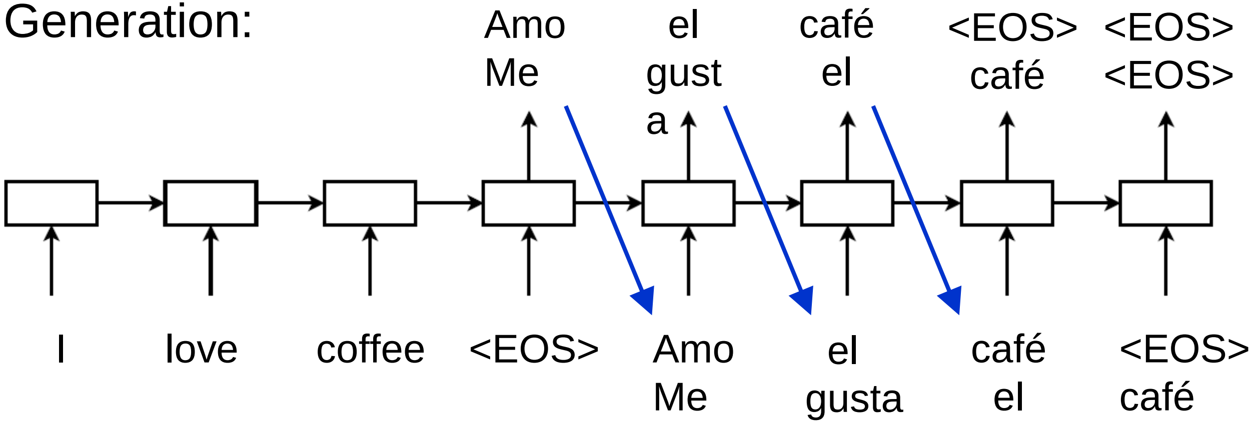



For translation:



# Sequence-To-Sequence RNNs

Generation:




Keep an n-best list of partial sentences, along with their partial softmax scores.

# Bleu scores for Translation

The goal of bleu scores is to compare machine translations against human-generated translations, allowing for variation.

Consider these translations for a Chinese sentence:

**Candidate 1:** It is a guide to action which ensures that the military always obeys the commands of the party.

**Candidate 2:** It is to insure the troops forever hearing the activity guidebook that party direct.

# Bleu Scores for Translation

**Candidate 1:** It is a guide to action which ensures that the military always obeys the commands of the party.

**Candidate 2:** It is to insure the troops forever hearing the activity guidebook that party direct.

**Reference 1:** It is a guide to action that ensures that the military will forever heed Party commands.

**Reference 2:** It is the guiding principle which guarantees the military forces always being under the command of the Party.

**Reference 3:** It is the practical guide for the army always to heed the directions of the party.

# Bleu Scores for Translation

**Candidate 1:** It is a guide to action which ensures that the military always obeys the commands of the party.

**Candidate 2:** It is to insure the troops forever hearing the activity guidebook that party direct.

**Reference 1:** It is a guide to action that ensures that the military will forever heed Party commands.

**Reference 2:** It is the guiding principle which guarantees the military forces always being under the command of the Party.

**Reference 3:** It is the practical guide for the army always to heed the directions of the party.

# Bleu Scores for Translation

**Candidate 1:** It is a guide to action which ensures that the military always obeys the commands of the party.

**Candidate 2:** It is to insure the troops forever hearing the activity guidebook that party direct.

**Reference 1:** It is a guide to action that ensures that the military will forever heed Party commands.

**Reference 2:** It is the guiding principle which guarantees the military forces always being under the command of the Party.

**Reference 3:** It is the practical guide for the army always to heed the directions of the party.

# Bleu Scores for Translation

Unigram precision:

$$\frac{\text{correct unigrams occuring in reference sentence}}{\text{unigrams occuring in test sentence}}$$

Modified unigram precision: clip counts by maximum occurrence in any reference sentence:

Candidate: the the the the the the the.
Reference 1: The cat is on the mat.
Reference 2: There is a cat on the mat.

Modified precision is 2/7.

# Bleu Scores for Translation

**Candidate 1:** It is a guide to action which ensures that the military always obeys the commands of the party. **unigram precision 17/18**

**Candidate 2:** It is to insure the troops forever hearing the activity guidebook that party direct. **unigram precision 8/14**

**Reference 1:** It is a guide to action that ensures that the military will forever heed Party commands.

**Reference 2:** It is the guiding principle which guarantees the military forces always being under the command of the Party.

**Reference 3:** It is the practical guide for the army always to heed the directions of the party.

# Bleu Scores for Translation

N-gram precision is defined similarly:

$$\frac{\text{correct ngrams occuring in reference sentence}}{\text{ngrams occuring in test sentence}}$$

Modified ngram precision: clip counts by maximum occurrence in any reference sentence.

Unigram scores tend to capture ***adequacy***

Ngram scores tend to capture ***fluency***

# Bleu Scores for Translation

**Candidate 1:** It is a guide to action which ensures that the military always obeys the commands of the party. **bigram precision 10/17**

**Candidate 2:** It is to insure the troops forever hearing the activity guidebook that party direct. **bigram precision 1/13**

**Reference 1:** It is a guide to action that ensures that the military will forever heed Party commands.

**Reference 2:** It is the guiding principle which guarantees the military forces always being under the command of the Party.

**Reference 3:** It is the practical guide for the army always to heed the directions of the party.

# Bleu Scores for Translation

How to combine scores for different n-grams?
Averaging sounds good, but precisions are very different for different n (unigrams have much higher scores).

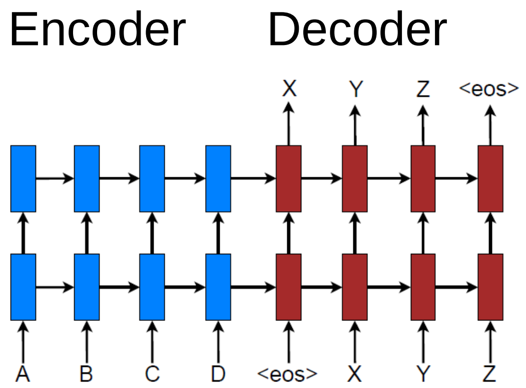
**BLEU Score:** Take a weighted geometric mean of the logs of n-gram precisions up to some length (usually 4). Add a penalty for too-short predictions.

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

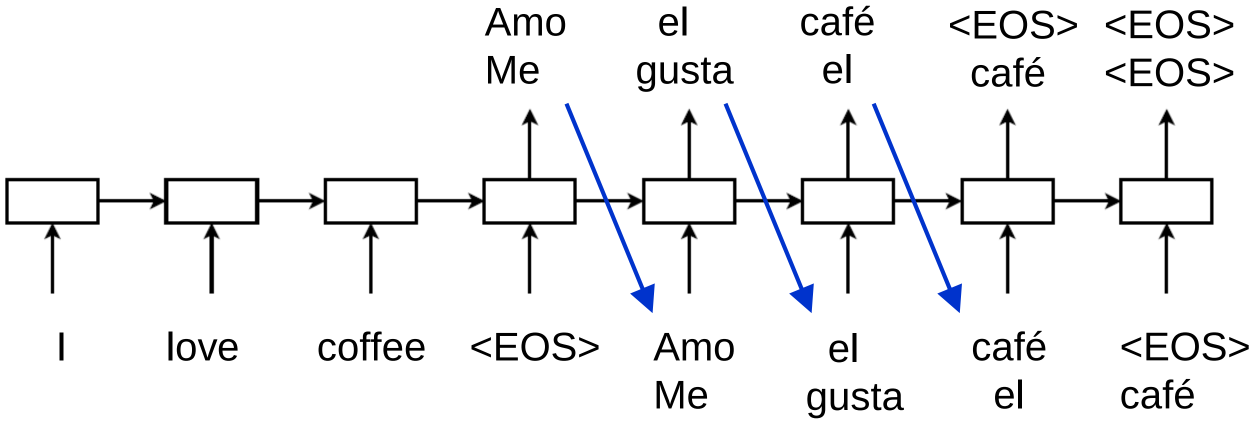
$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

Candidate length c shorter than reference r translation

# Sequence-To-Sequence Model Translation




A depth-2 model




Beam search width = 2

# Sequence-To-Sequence Model Translation

## Raw scores for French-English Translation, depth = 4

| Method | test BLEU score (ntst14) |
|---|---|
| Bahdanau et al. [2] | 28.45 |
| Baseline System [29] | 33.30 |
| Single forward LSTM, beam size 12 | 26.17 |
| Single reversed LSTM, beam size 12 | 30.59 |
| Ensemble of 5 reversed LSTMs, beam size 1 | 33.00 |
| Ensemble of 2 reversed LSTMs, beam size 12 | 33.27 |
| Ensemble of 5 reversed LSTMs, beam size 2 | 34.50 |
| Ensemble of 5 reversed LSTMs, beam size 12 | **34.81** |

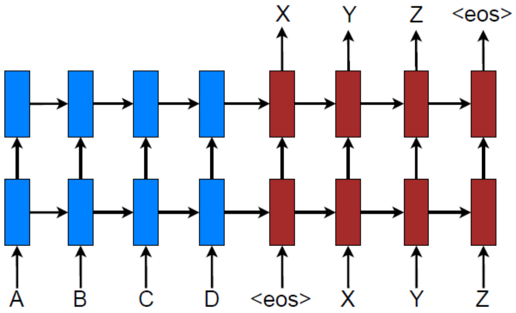
Reversed = reverse the order of the input sentence.

Intuition: the first part of the sentence is the most important, and reversal eases the long-term dependencies from output to input sentence.

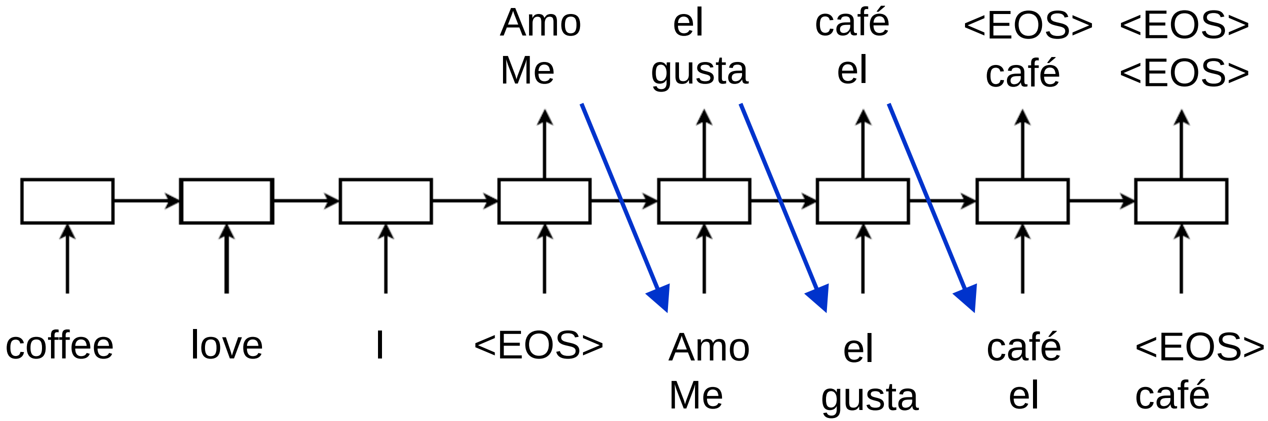
From Sutskeyver et al. “Sequence to Sequence Learning with Neural Networks” 2014.

# Sequence-To-Sequence Model Translation

## Input sequence reversal




A depth-2 model




Beam search width = 2

# Sequence-To-Sequence Model Translation

Raw scores for French-English Translation, depth = 4

| Method | test BLEU score (ntst14) |
|---|---|
| Bahdanau et al. [2] | 28.45 |
| Baseline System [29] | 33.30 |
| Single forward LSTM, beam size 12 | 26.17 |
| Single reversed LSTM, beam size 12 | 30.59 |
| Ensemble of 5 reversed LSTMs, beam size 1 | 33.00 |
| Ensemble of 2 reversed LSTMs, beam size 12 | 33.27 |
| Ensemble of 5 reversed LSTMs, beam size 2 | 34.50 |
| Ensemble of 5 reversed LSTMs, beam size 12 | **34.81** |

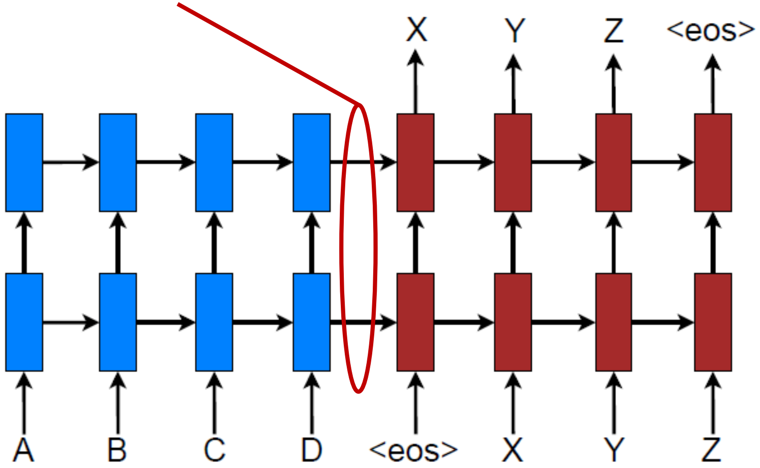
Beam sizes are tiny!!

The model produces state-of-the-art translations with almost no search.

From Sutskeyver et al. “Sequence to Sequence Learning with Neural Networks” 2014.

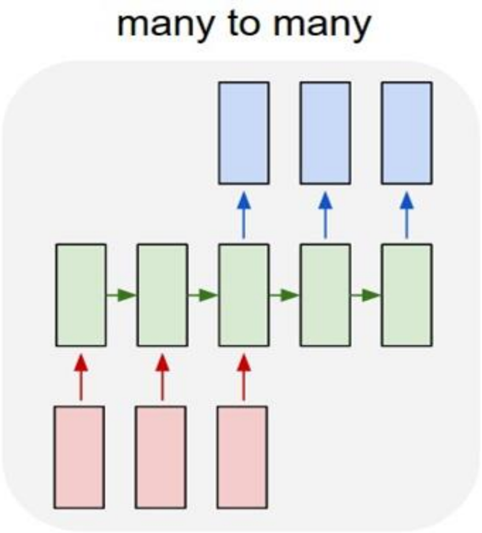
# Sequence-To-Sequence Criticisms

All the information from the source sentence has to pass through the bottleneck at the last unit(s) of the encoder.




Sentence length varies, but the encoding always has a fixed size.

# Soft Attention for Translation

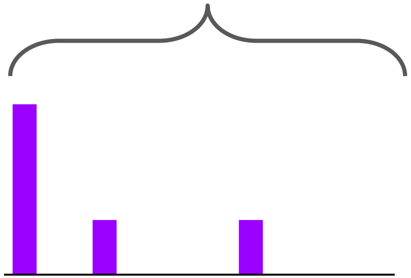



“I love coffee” -> “Me gusta el café”

Bahdanau et al, “Neural Machine Translation by Jointly Learning to Align and Translate”, ICLR 2015

# Soft Attention for Translation

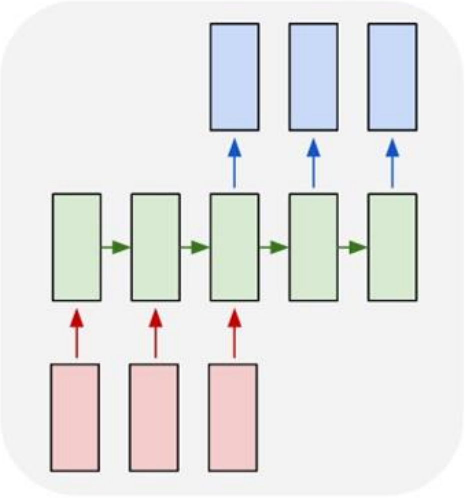
Distribution over input words



"I love coffee" -> "Me gusta el café"

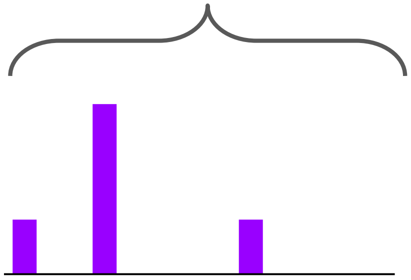
Bahdanau et al, "Neural Machine Translation by Jointly Learning to Align and Translate", ICLR 2015

many to many



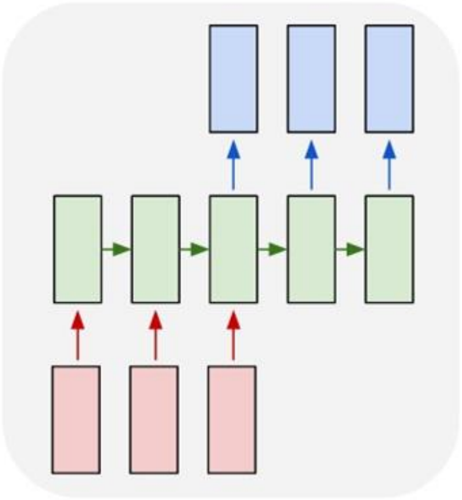
# Soft Attention for Translation

Distribution over input words



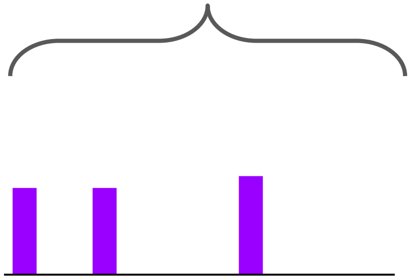
"I love coffee” -> “Me gusta el café”

Bahdanau et al, “Neural Machine Translation by Jointly Learning to Align and Translate”, ICLR 2015



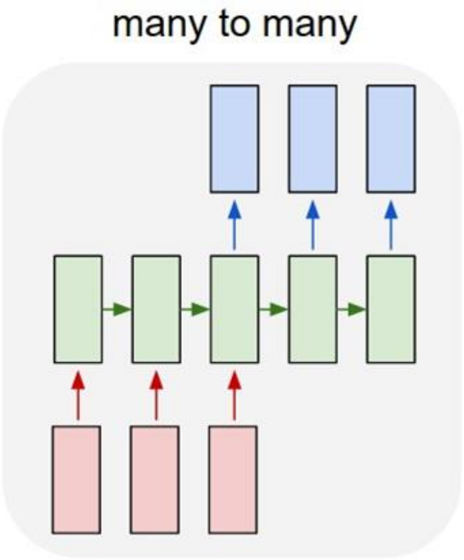
# Soft Attention for Translation

Distribution over input words

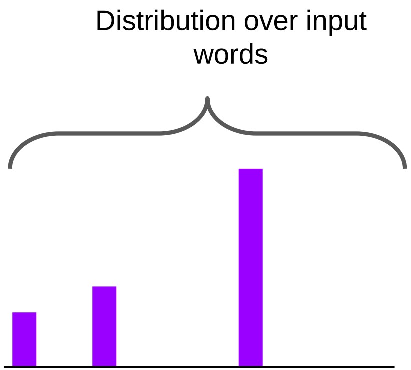


"I love coffee” -> “Me gusta el café”

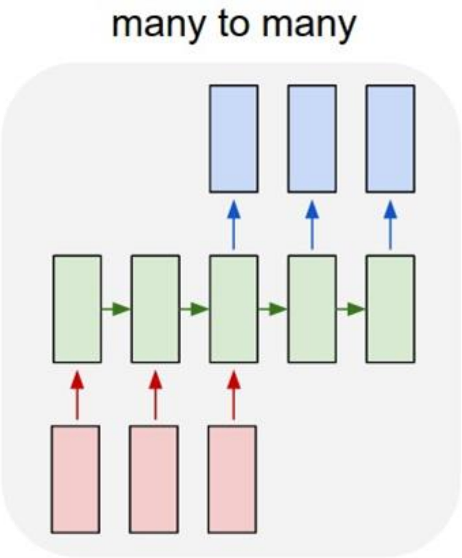
Bahdanau et al, “Neural Machine Translation by Jointly Learning to Align and Translate”, ICLR 2015



# Soft Attention for Translation



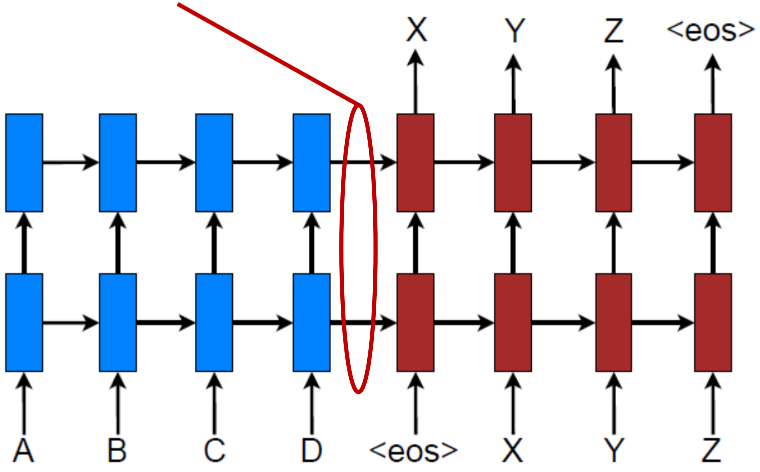

"I love coffee" -> "Me gusta el café"




Bahdanau et al, "Neural Machine Translation by Jointly Learning to Align and Translate", ICLR 2015

# Sequence-To-Sequence Criticisms

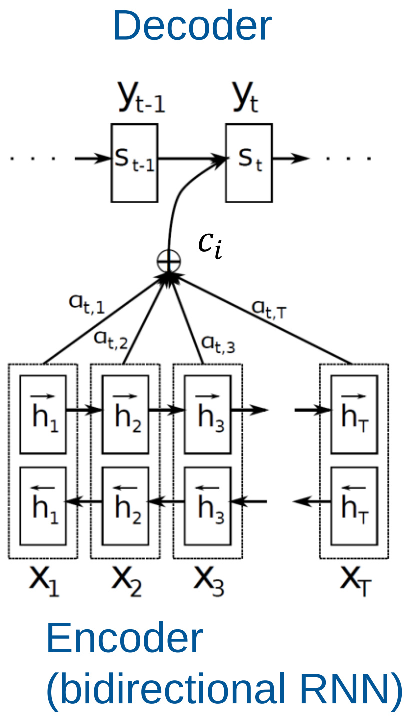
All the information from the source sentence has to pass through the bottleneck at the last unit(s) of the encoder.




Sentence length varies, but the encoding always has a fixed size.

# Soft Attention for Translation – Bahdanau et al. model

For each output word, focus attention on a subset of all input words.



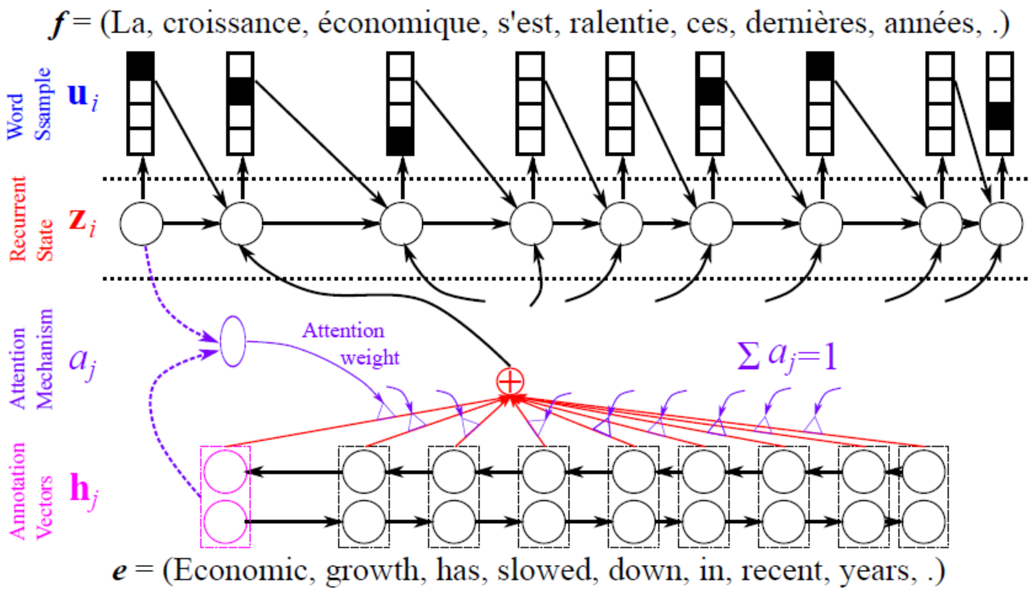

Context vector (input to decoder): $$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$

Mixture weights: $$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$$

Alignment score (how well do input words near j match output words at position i): $$e_{ij} = a(s_{i-1}, h_j)$$

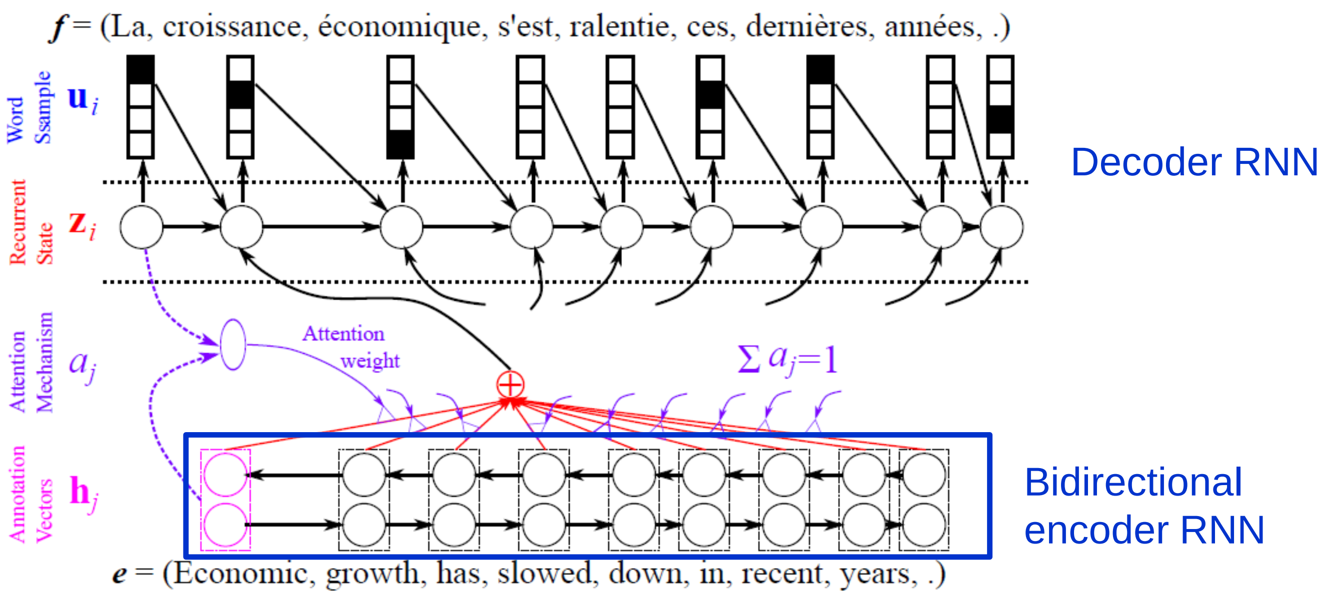
Bahdanau et al, “Neural Machine Translation by Jointly Learning to Align and Translate”, ICLR 2015

# Soft Attention for Translation



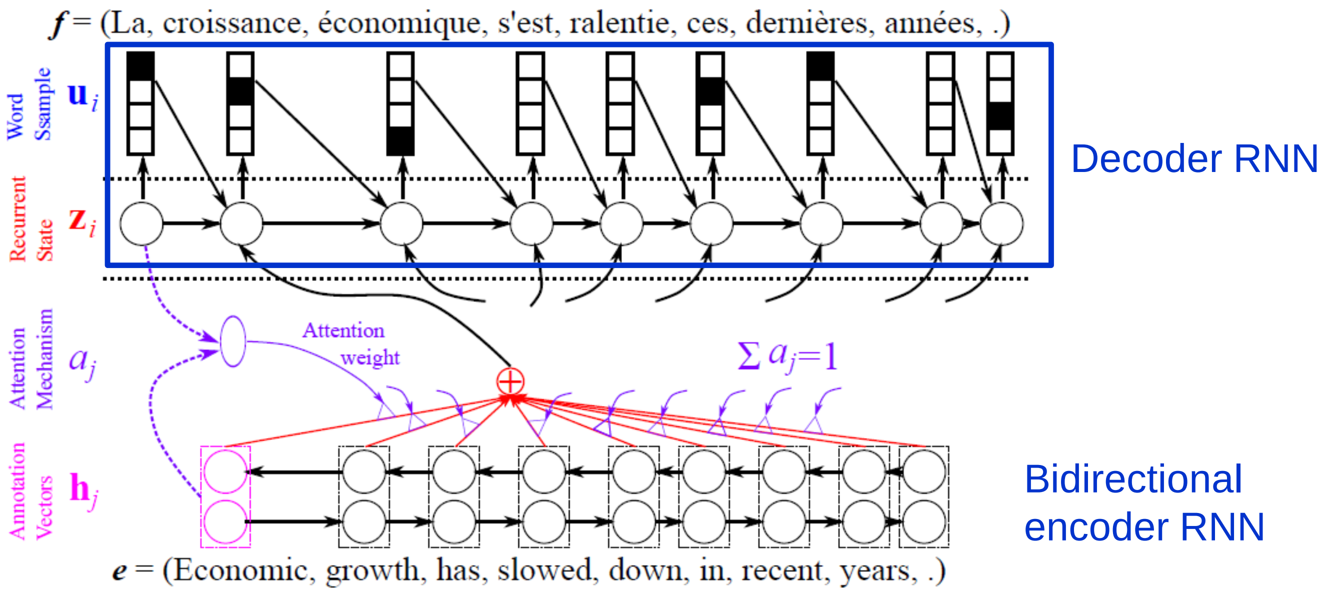

From Y. Bengio CVPR 2015 Tutorial

# Soft Attention for Translation



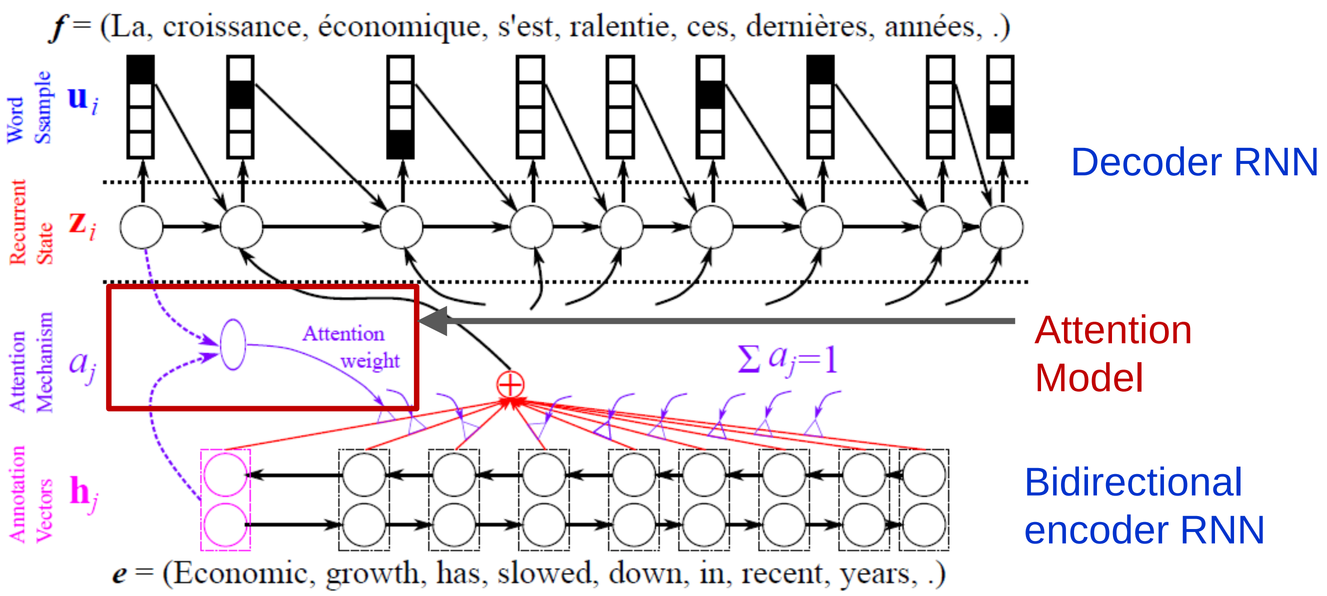

From Y. Bengio CVPR 2015 Tutorial

# Soft Attention for Translation



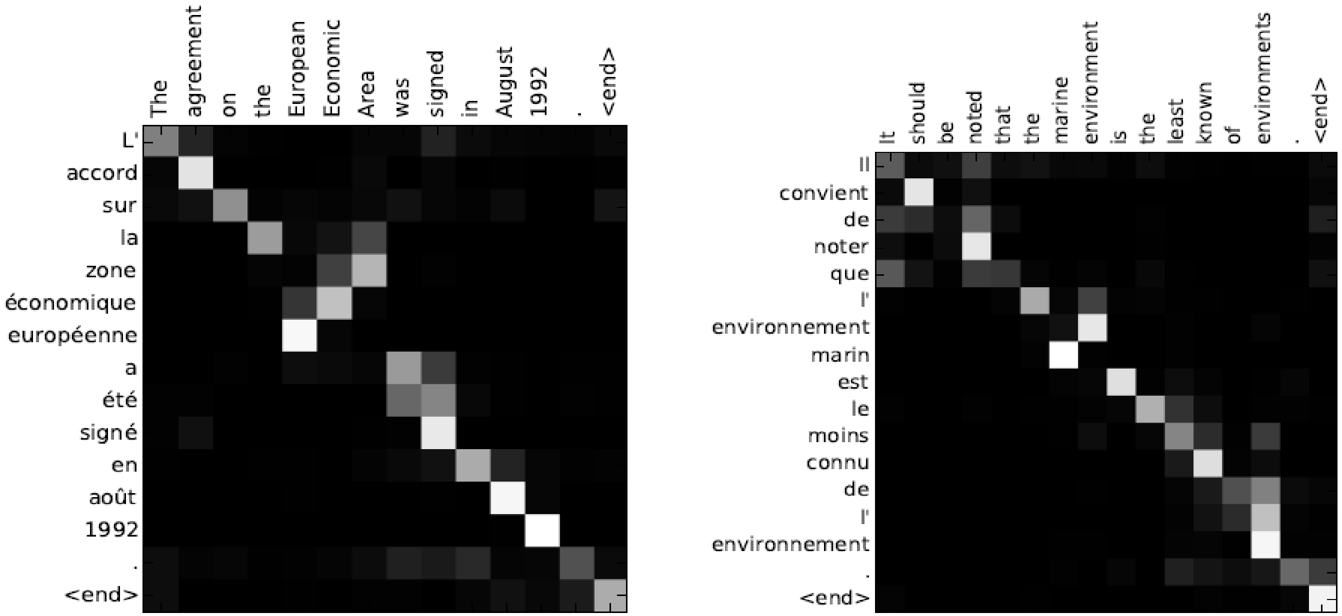

From Y. Bengio CVPR 2015 Tutorial

# Soft Attention for Translation




From Y. Bengio CVPR 2015 Tutorial

# Soft Attention for Translation




(a) (b)

Bahdanau et al, “Neural Machine Translation by Jointly Learning to Align and Translate”, ICLR 2015

# Soft Attention for Translation

Reached State of the art in one year:

(a) **English→French (WMT-14)**

| | NMT(A) | Google | P-SMT |
|---|---|---|---|
| NMT | 32.68 | 30.6* | **37.03•** |
| +Cand | 33.28 | – | |
| +UNK | 33.99 | 32.7° | |
| +Ens | 36.71 | **36.9°** | |

(b) **English→German (WMT-15)**

| Model | Note |
|---|---|
| 24.8 | Neural MT |
| 24.0 | U.Edinburgh, Syntactic SMT |
| 23.6 | LIMSI/KIT |
| 22.8 | U.Edinburgh, Phrase SMT |
| 22.7 | KIT, Phrase SMT |

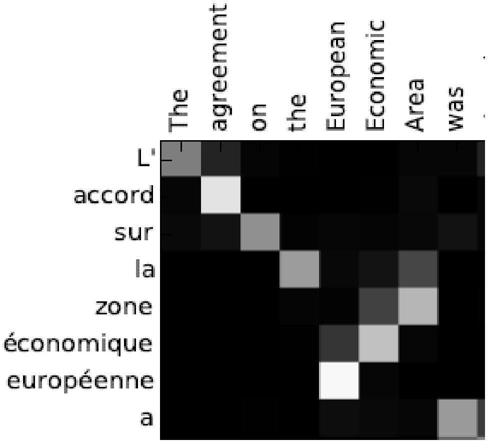
(c) **English→Czech (WMT-15)**

| Model | Note |
|---|---|
| 18.3 | Neural MT |
| 18.2 | JHU, SMT+LM+OSM+Sparse |
| 17.6 | CU, Phrase SMT |
| 17.4 | U.Edinburgh, Phrase SMT |
| 16.1 | U.Edinburgh, Syntactic SMT |

Yoshua Bengio, NIPS RAM workshop 2015

# Criticism of Badanau et al.

The attention function $a\left(s_{i-1}, h_j\right)$ is rather complex, yet the attention often seems to be a simple heat map on word similarity:




The data path in Badanau is quite complicated: the attention path is another recurrent path between output states.

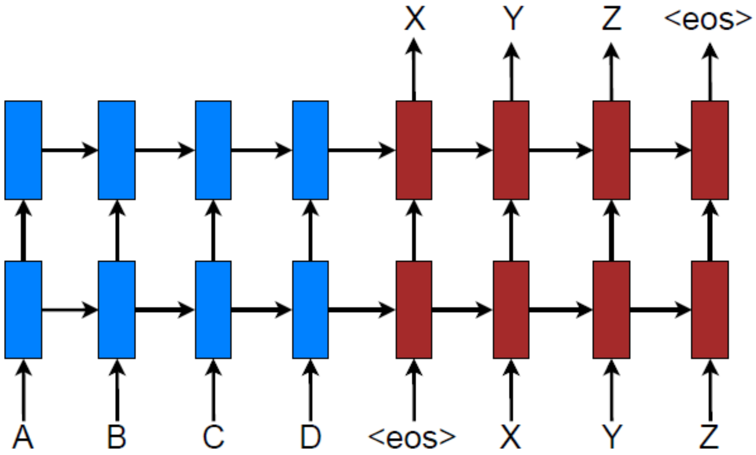
Doesn't generalize to deeper networks (shown to be Important by Sutskeyver et al.).

Luong and Manning added several architectural improvements.

Effective Approaches to Attention-based Neural Machine Translation
Minh-Thang Luong, Hieu Pham, Christopher D. Manning, EMNLP 15

# Luong, Pham and Manning 2015

Stacked LSTM with arbitrary depth (c.f. bidirectional flat encoder in Bahdanau et al):

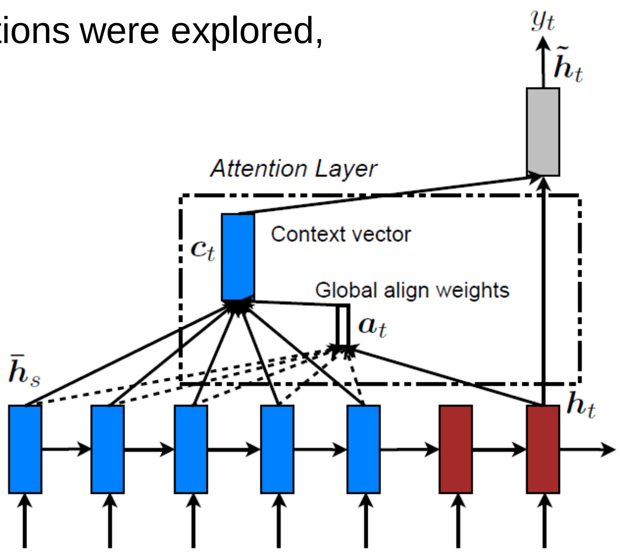



Effective Approaches to Attention-based Neural Machine Translation
Minh-Thang Luong, Hieu Pham, Christopher D. Manning, EMNLP 15

# Global Attention Model

Global attention model is similar but simpler than Badanau's. It sits above the encoder/decoder and is not itself recurrent.

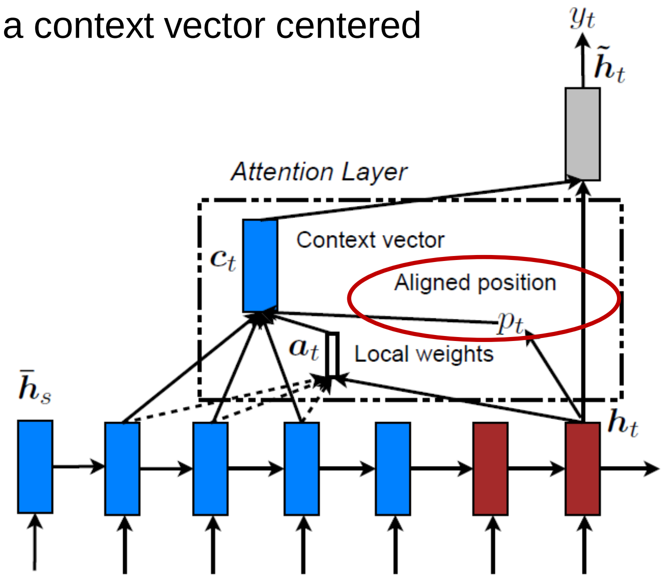
Different word matching functions were explored, some yielding better results.

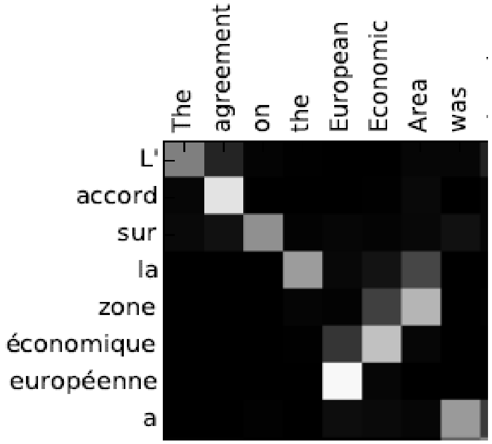



Effective Approaches to Attention-based Neural Machine Translation
Minh-Thang Luong Hieu Pham Christopher D. Manning, EMNLP 15

# Local Attention Model

- Compute a best aligned position $p_t$ first
- Then compute a context vector centered at that position




Effective Approaches to Attention-based Neural Machine Translation
Minh-Thang Luong Hieu Pham Christopher D. Manning, EMNLP 15

# Luong, Pham and Manning's Translation System (2015):

| System | BLEU |
|---|---|
| Top – *NMT + 5-gram rerank* (Montreal) | 24.9 |
| Our ensemble 8 models + unk replace | **25.9** |

Table 2: **WMT'15 English-German results** – *NIST* BLEU scores of the winning entry in WMT'15 and our best one on newstest2015.

| System | Ppl. | BLEU |
|---|---|---|
| *WMT'15 systems* | | |
| SOTA – *phrase-based* (Edinburgh) | | **29.2** |
| NMT + 5-gram rerank (MILA) | | 27.6 |
| *Our NMT systems* | | |
| Base (reverse) | 14.3 | 16.9 |
| + global (*location*) | 12.7 | 19.1 (+*2.2*) |
| + global (*location*) + feed | 10.9 | 20.1 (+*1.0*) |
| + global (*dot*) + drop + feed | 9.7 | 22.8 (+*2.7*) |
| + global (*dot*) + drop + feed + unk | | 24.9 (+*2.1*) |

Table 3: **WMT'15 German-English results** –

# Parsing

Recall (Lecture 10) RNNs ability to generate Latex, C code:

*Proof.* Omitted. □

**Lemma 0.1.** *Let $\mathcal{C}$ be a set of the construction.*

*Let $\mathcal{C}$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that*

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

.

*Proof.* This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules. □

**Lemma 0.2.** *This is an integer $\mathcal{Z}$ is injective.*

*Proof.* See Spaces, Lemma ??. □

**Lemma 0.3.** *Let $S$ be a scheme. Let $X$ be a scheme and $X$ is an affine open covering. Let $\mathcal{U} \subset \mathcal{X}$ be a canonical and locally of finite type. Let $X$ be a scheme. Let $X$ be a scheme which is equal to the formal complex.*

*The following to the construction of the lemma follows.*

*Let $X$ be a scheme. Let $X$ be a scheme covering. Let*

$$b : X \to Y' \to Y \to Y \to Y' \times_X Y \to X.$$

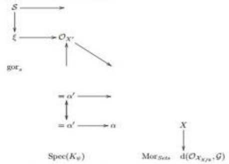
*be a morphism of algebraic spaces over $S$ and $Y$.*

*Proof.* Let $X$ be a nonzero scheme of $X$. Let $X$ be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

(1) $\mathcal{F}$ is an algebraic space over $S$.
(2) If $X$ is an affine open covering.

Consider a common structure on $X$ and $X$ the functor $\mathcal{O}_X(U)$ which is locally of finite type. □

This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram




is a limit. Then $\mathcal{G}$ is a finite type and assume $S$ is a flat and $\mathcal{F}$ and $\mathcal{G}$ is a fi type $f_*$. This is of finite type diagrams, and

- the composition of $\mathcal{G}$ is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

*Proof.* We have see that $X = \mathrm{Spec}(R)$ and $\mathcal{F}$ is a finite type representabl algebraic space. The property $\mathcal{F}$ is a finite morphism of algebraic stacks. Then cohomology of $X$ is an open neighbourhood of $U$.

*Proof.* This is clear that $\mathcal{G}$ is a finite presentation, see Lemmas ??.
A *reduced above* we conclude that $U$ is an open covering of $\mathcal{C}$. The functor $\mathcal{F}$ "field

$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_{\bar{x}} \ \ {-1}(\mathcal{O}_{X_{\acute{e}tale}}) \longrightarrow \mathcal{O}_{X_\ell}^{-1}\mathcal{O}_{X_\lambda}(\mathcal{O}_{X_\eta}^{\overline{v}})$$

is an isomorphism of covering of $\mathcal{O}_{X_i}$. If $\mathcal{F}$ is the unique element of $\mathcal{F}$ such tha is an isomorphism.
The property $\mathcal{F}$ is a disjoint union of Proposition ?? and we can filtered se presentations of a scheme $\mathcal{O}_X$-algebra with $\mathcal{F}$ are opens of finite type over $S$.
If $\mathcal{F}$ is a scheme theoretic image points.

If $\mathcal{F}$ is a finite direct sum $\mathcal{O}_{X_\lambda}$ is a closed immersion, see Lemma ??. This sequence of $\mathcal{F}$ is a similar morphism.

```c
static void do_command(struct seq_file *m, void *v)
{
  int column = 32 << (cmd[2] & 0x80);
  if (state)
    cmd = (int)(int_state ^ (in_8(&ch->ch_flags) & Cmd) ? 2 : 1);
  else
    seq = 1;
  for (i = 0; i < 16; i++) {
    if (k & (1 << 1))
      pipe = (in_use & UMXTHREAD_UNCCA) +
        ((count & 0x00000000fffffff8) & 0x000000f) << 8;
    if (count == 0)
      sub(pid, ppc_md.kexec_handle, 0x20000000);
    pipe_set_bytes(i, 0);
  }
  /* Free our user pages pointer to place camera if all dash */
  subsystem_info = &of_changes[PAGE_SIZE];
  rek_controls(offset, idx, &soffset);
  /* Now we want to deliberately put it to device */
  control_check_polarity(&context, val, 0);
  for (i = 0; i < COUNTER; i++)
    seq_puts(s, "policy ");
}
```

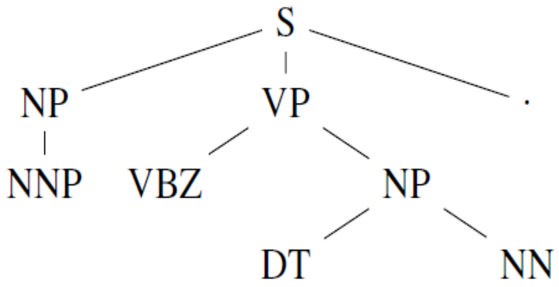
They seem to do well with tree-structured data.

What about natural language parsing?

# Parsing

Sequence models generate linear structures, but these can easily encode trees by “closing parens” (prefix tree notation):

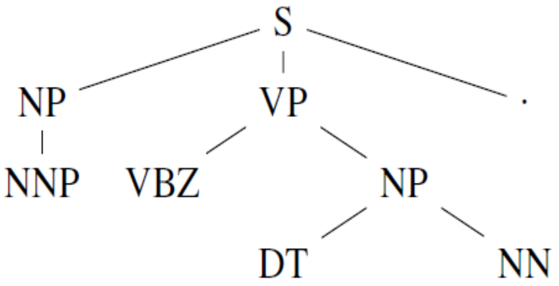
John has a dog . $\rightarrow$




John has a dog . $\rightarrow$ (S (NP NNP $)_{NP}$ (VP VBZ (NP DT NN $)_{NP}$ $)_{VP}$ . $)_{S}$

# Parsing Cheat Sheet

John has a dog . $\rightarrow$




John has a dog . $\rightarrow$ $(\mathrm{S}\ (\mathrm{NP}\ \mathrm{NNP}\ )_{\mathrm{NP}}\ (\mathrm{VP}\ \mathrm{VBZ}\ (\mathrm{NP}\ \mathrm{DT}\ \mathrm{NN}\ )_{\mathrm{NP}}\ )_{\mathrm{VP}}\ .\ )_{\mathrm{S}}$

S = Sentence

NP = Noun Phrase

VP = Verb Phrase

NNP = Proper Noun (“John”)

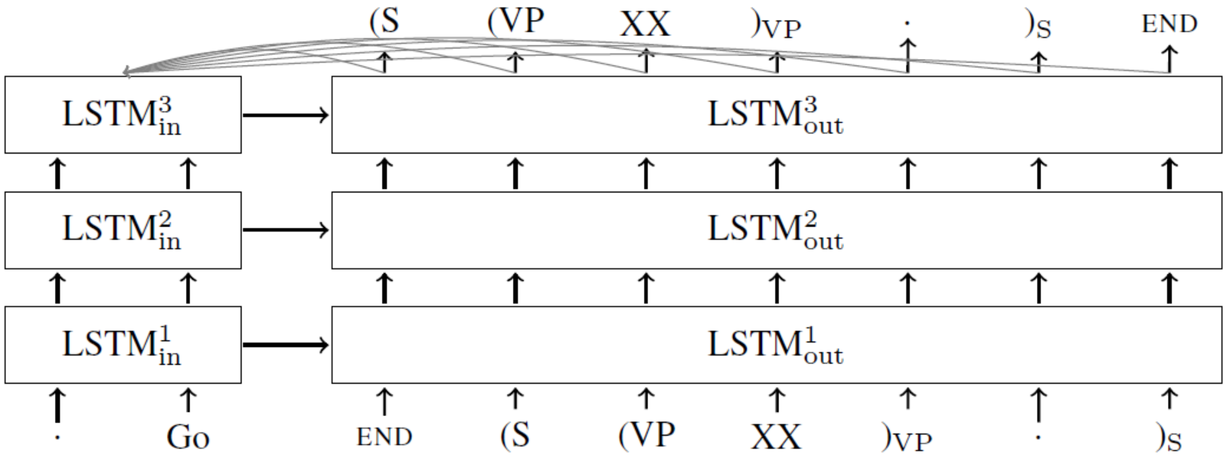
VBZ = Verb, 3$^{rd}$ person, singular (“has”)

DT = Determiner (“a”)

NN = Noun, singular (“dog”)

# A Sequence-To-Sequence Parser

The model is a depth-3 sequence-to-sequence predictor, augmented with the attention model of Bahdanau 2014.




Grammar as a Foreign Language Oriol Vinyals, Google, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, Geoffrey Hinton, NIPS 2015

“Neural machine translation by jointly learning to align and translate.” Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. arXiv 2014.

# A Sequence-To-Sequence Parser

Chronology:

- First tried training a basic sequence-to-sequence model on human-annotated training treebanks. **Poor results.**
- Then training on parse trees **generated by the Berkeley Parser**, achieved similar performance (90.5 F1 score) to it.
- Next added the attention model, trained **on human treebank data**, also achieved 90.5 F1.
- Finally, created a synthetic dataset of **high-confidence parse trees** (agreed on by two parsers). Achieved a new state-of-the-art of 92.5 F1 score (WSJ dataset).

F1 is a widely-used accuracy measure that combines precision and recall

# A Sequence-To-Sequence Parser

Quick Training Details:

- Depth = 3, layer dimension = 256.
- **Dropout** between layers 1 and 2, and 2 and 3.
- **No POS tags!!** Improved by F1 1 point by leaving them out.
- Input reversing.

# Attention-only Translation Models

Problems with recurrent networks:

- Sequential training and inference: time grows in proportion to sentence length. Hard to parallelize.
- Long-range dependencies have to be remembered across many single time steps.
- Tricky to learn hierarchical structures (“car”, “blue car”, “into the blue car”…)

Alternative:

- Convolution – but has other limitations.

# Self-Attention

Information flows from within the same subnetwork (either encoder or decoder).
Convolution applies fixed transform weights. Self-attention applies variable weights (but typically not transformations):


image from Lukas Kaiser, Stanford NLP seminar

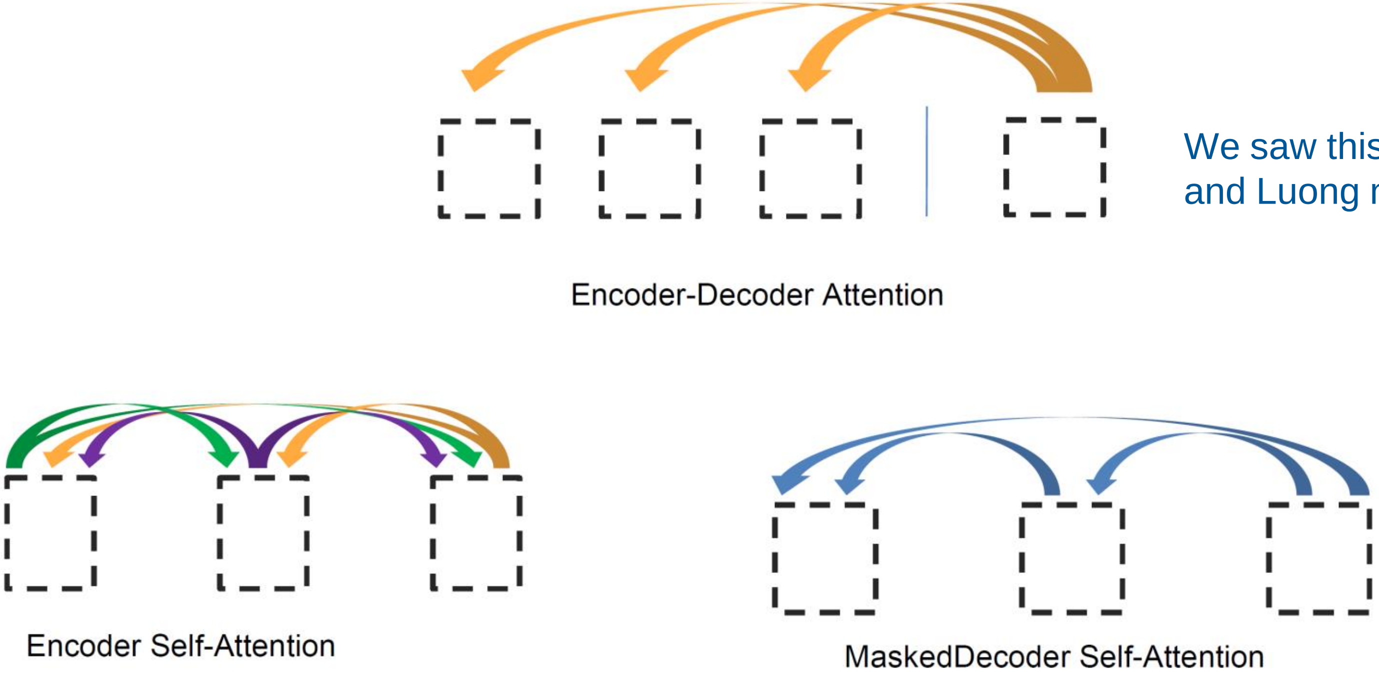
# Self-Attention “Transformers” (not spatial transformers)

- Constant path length between any two positions.
- Variable receptive field (or the whole input sequence).
- Supports hierarchical information flow by stacking self-attention layers.
- Trivial to parallelize.
- Attention weighting controls information propagation.

- **Can replace word-based recurrence entirely.**

Vaswani et al. “Attention is all you need”, arXiv 2017

# Attention in Transformer Networks




We saw this in Bahdanau and Luong models

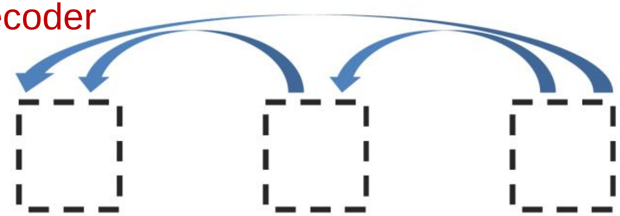
image from Lukas Kaiser, Stanford NLP seminar

# Attention in Transformer Networks

Encoder-Decoder Attention

Replaces word recurrence in
encoder and decoder

Encoder Self-Attention



MaskedDecoder Self-Attention

image from Lukas Kaiser, Stanford NLP seminar

# Attention in Transformer Networks

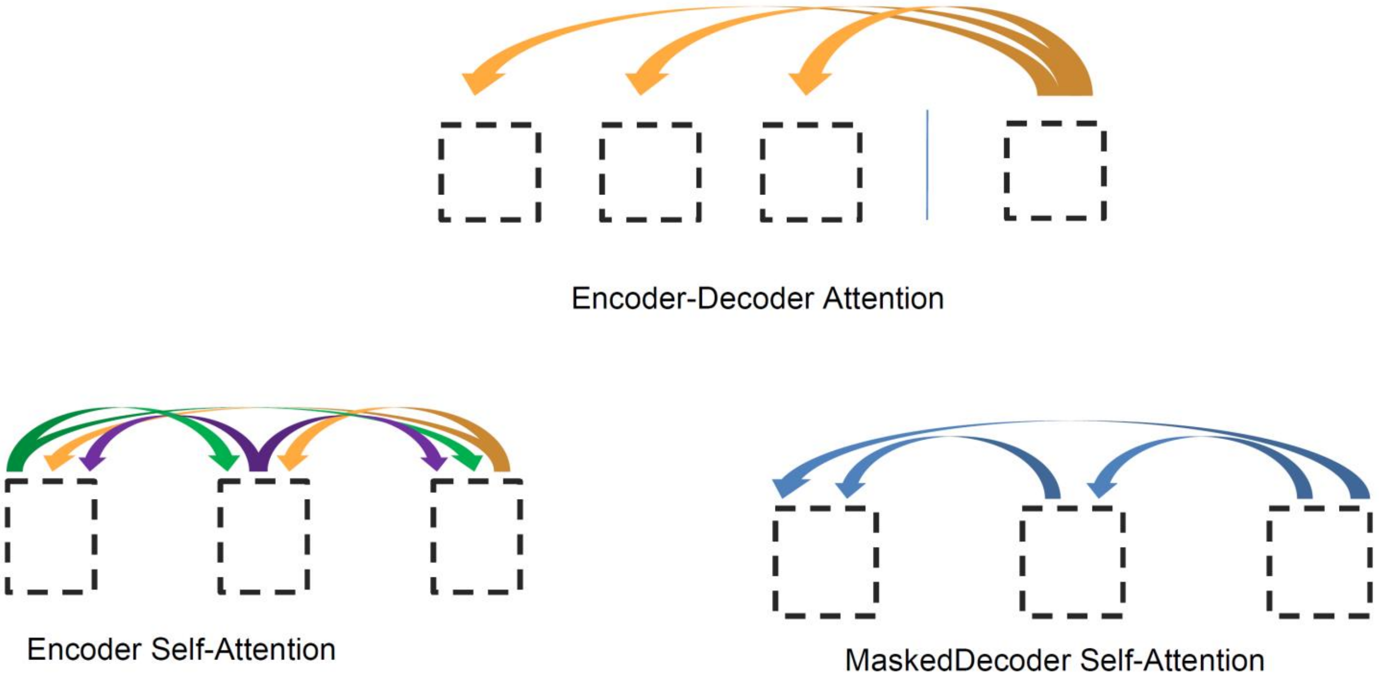
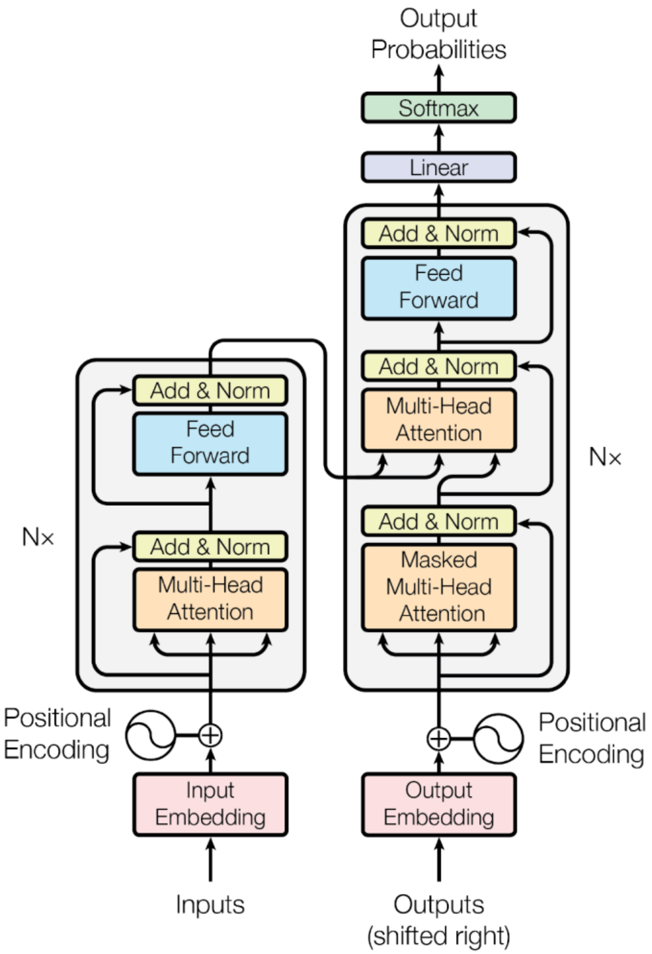



image from Lukas Kaiser, Stanford NLP seminar

Masking limits attention to earlier units:
$y_i$ depends only on $y_j$ for $j < i$.

# The Transformer

- Basic unit shown at right.
- In experiments, stacked with N=6.
- Output words fed back as input, shifted right. Can use beam search as before.
- Inputs and outputs are embedded in vector spaces of fixed dimension.
- Positional encoding: when words are combined through attention, their location is lost. Positional encoding adds it back.



# Attention Implementation

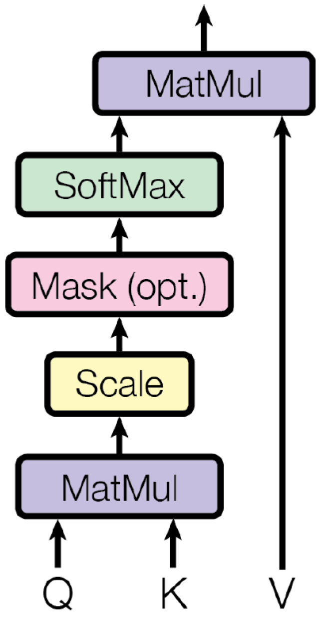
- Attention is modeled as a key-value store:
  Q = query vector
  K = key
  V = value

Encoder-decoder layer: the queries come from the previous decoder layer, and the memory keys and values come from the output of the encoder. (Similar to Bahdanau).

Self-attention layer: all of the keys, values and queries come from the output of the previous layer in the encoder.

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

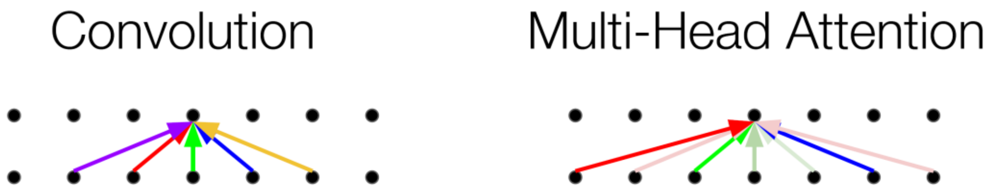
Scaled Dot-Product Attention

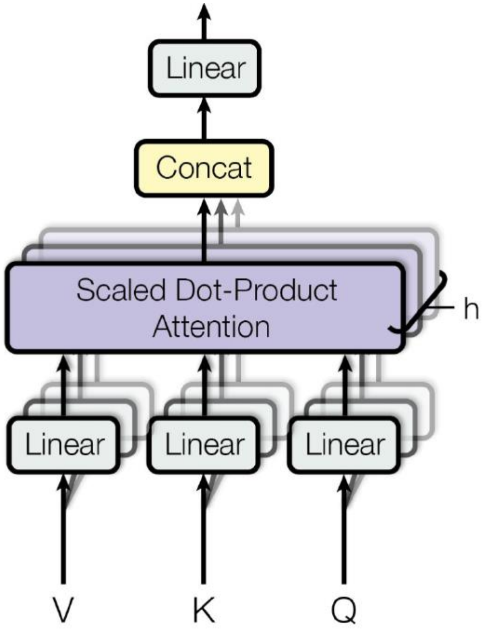


# Multi-Headed Attention

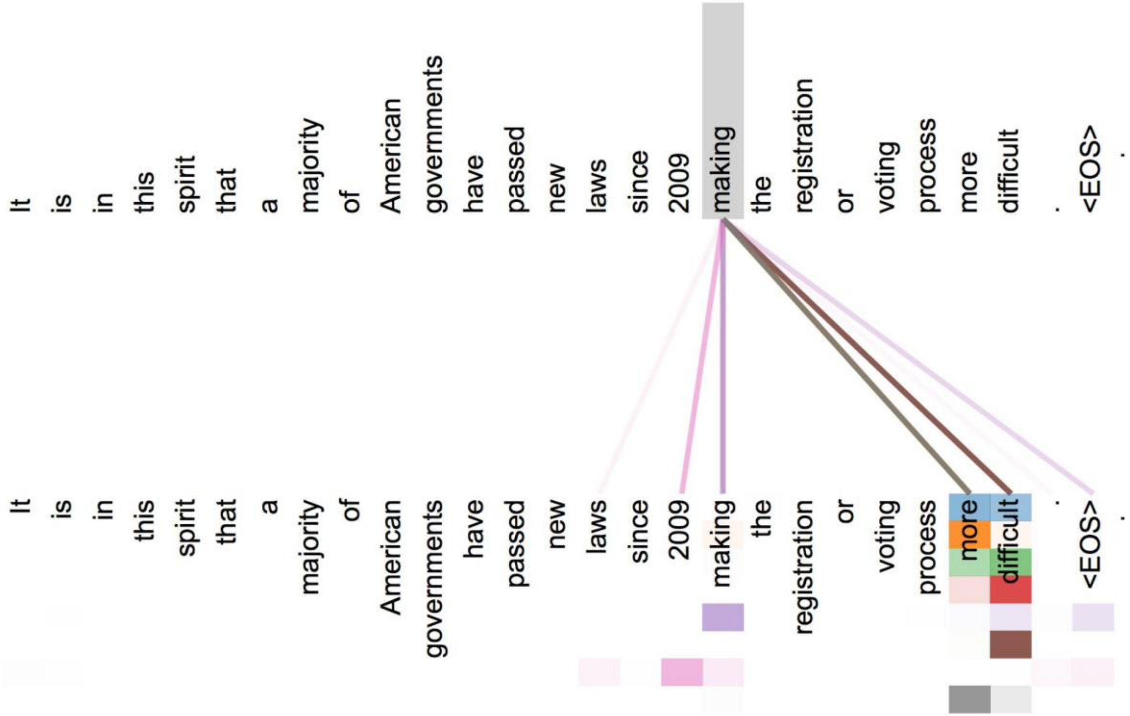
- Simple attention blends the results of all the attended-to inputs. It doesn't allow a per-input transformation, as convolution does.

- The solution is to use "multi-headed attention":

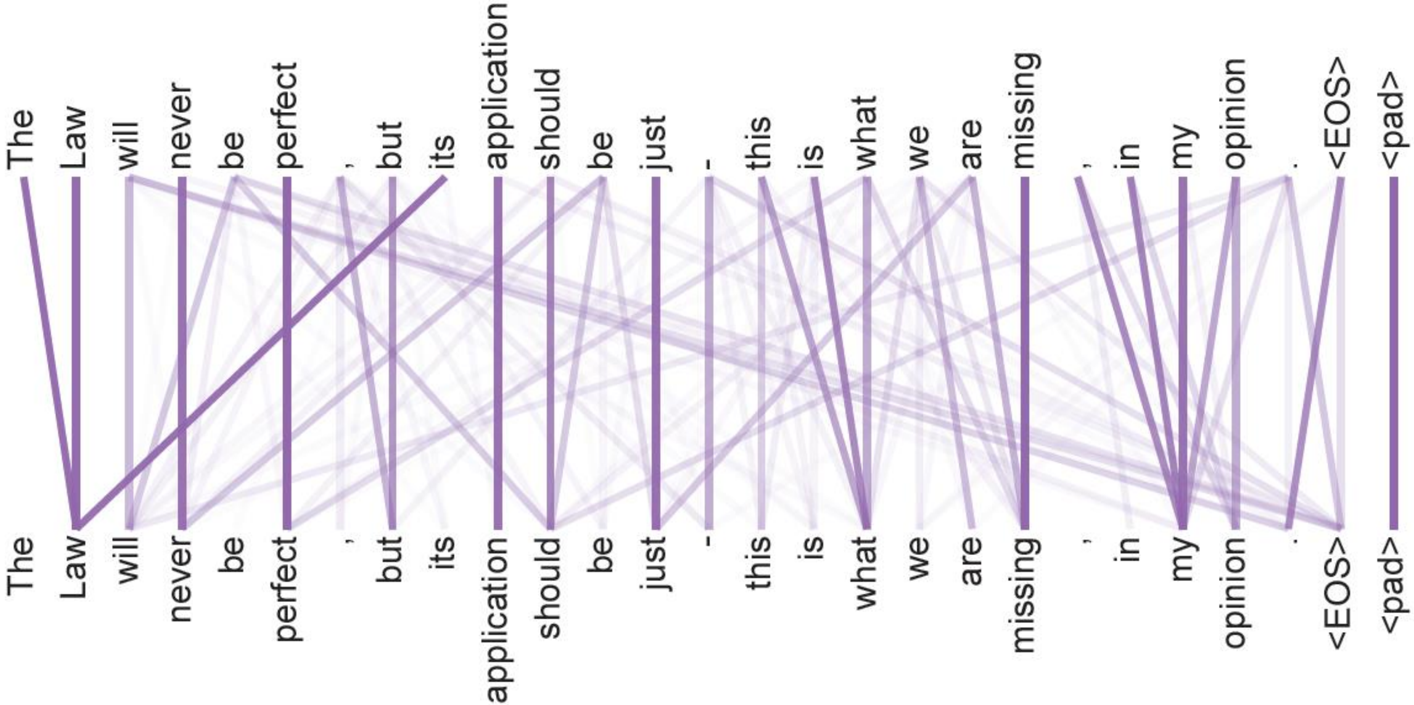


# Multi-Headed Attention

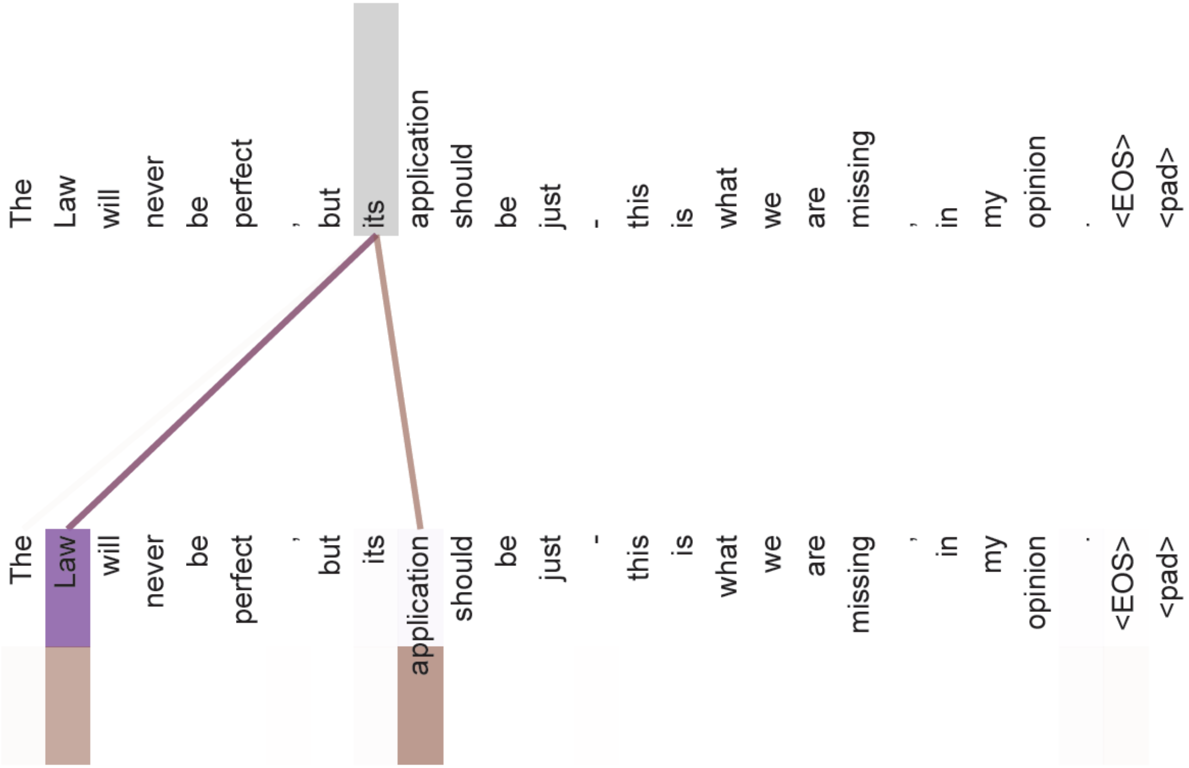


# Multi-Headed Attention




Anaphora (pronoun or article) resolution

# Multi-Headed Attention




Anaphora (pronoun or article) resolution

# Transformer Results

## Machine Translation Results: WMT-14

| Model | BLEU | | Training Cost (FLOPs) | |
|---|---|---|---|---|
| | EN-DE | EN-FR | EN-DE | EN-FR |
| ByteNet [17] | 23.75 | | | |
| Deep-Att + PosUnk [37] | | 39.2 | | $1.0 \cdot 10^{20}$ |
| GNMT + RL [36] | 24.6 | 39.92 | $2.3 \cdot 10^{19}$ | $1.4 \cdot 10^{20}$ |
| ConvS2S [9] | 25.16 | 40.46 | $9.6 \cdot 10^{18}$ | $1.5 \cdot 10^{20}$ |
| MoE [31] | 26.03 | 40.56 | $2.0 \cdot 10^{19}$ | $1.2 \cdot 10^{20}$ |
| Deep-Att + PosUnk Ensemble [37] | | 40.4 | | $8.0 \cdot 10^{20}$ |
| GNMT + RL Ensemble [36] | 26.30 | 41.16 | $1.8 \cdot 10^{20}$ | $1.1 \cdot 10^{21}$ |
| ConvS2S Ensemble [9] | 26.36 | **41.29** | $7.7 \cdot 10^{19}$ | $1.2 \cdot 10^{21}$ |
| Transformer (base model) | 27.3 | 38.1 | $\mathbf{3.3 \cdot 10^{18}}$ | |
| Transformer (big) | **28.4** | **41.0** | $2.3 \cdot 10^{19}$ | |

# English-to-English Translation ?!

Yes, it does make sense. a.k.a. summarization.

Liu et al, “GENERATING WIKIPEDIA BY SUMMARIZING LONG SEQUENCES” arXiv 2018

M = input length, N = output length

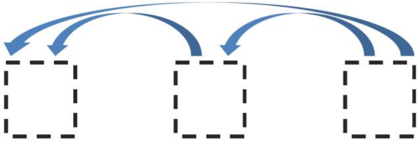
Summarization: M >> N

$O(MN)$

Encoder-Decoder Attention

$O(M^2)$

Encoder Self-Attention



$O(N^2)$

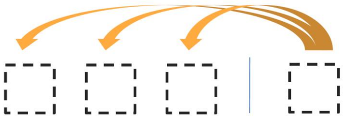
MaskedDecoder Self-Attention

# Large-scale Summarization (Wikipedia)

Like translation, but we completely remove the encoder.

Source data (large!):

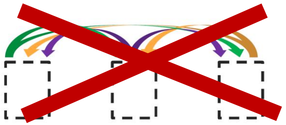
- The references for a Wikipedia article.
- Web search using article section titles, ~ 10 web pages per query.



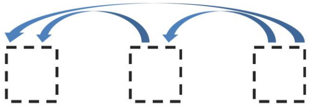
$O(MN)$

Encoder-Decoder Attention

$O(M^2)$

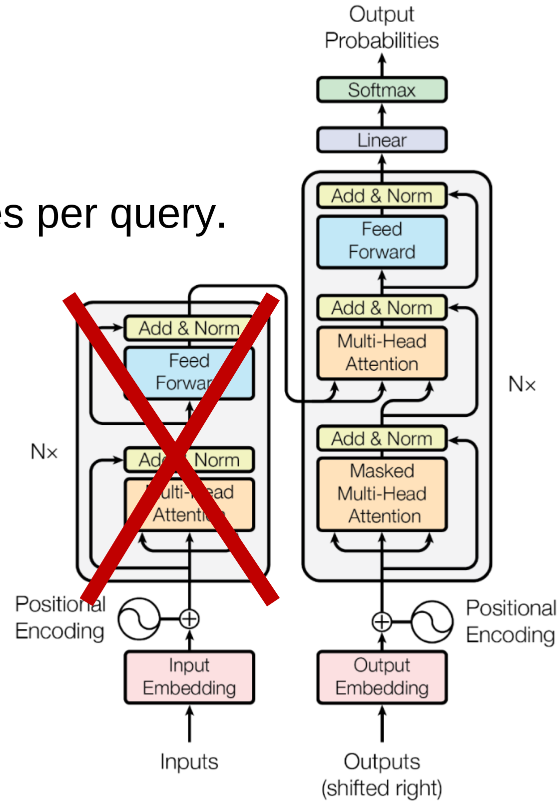


Encoder Self-Attention



$O(N^2)$

MaskedDecoder Self-Attention



# Large-scale Summarization

Results:

| Model | Test perplexity | ROUGE-L |
|---|---|---|
| *seq2seq-attention*, $L = 500$ | 5.04952 | 12.7 |
| *Transformer-ED*, $L = 500$ | 2.46645 | 34.2 |
| *Transformer-D*, $L = 4000$ | 2.22216 | 33.6 |
| *Transformer-DMCA, no MoE-layer*, $L = 11000$ | 2.05159 | 36.2 |
| *Transformer-DMCA, MoE-128*, $L = 11000$ | 1.92871 | 37.9 |
| *Transformer-DMCA, MoE-256*, $L = 7500$ | 1.90325 | 38.8 |

L = input window length.
ED = encoder-decoder.
D = decoder only.
DMCA = a memory compression technique (strided convolution).
MoE = mixture of experts layer.

Liu et al, “GENERATING WIKIPEDIA BY SUMMARIZING LONG SEQUENCES” arXiv 2018

# Translation Takeaways


- Sequence-to-sequence translation

  - Input reversal
  - Narrow beam search

- Adding Attention

  - Compare latent states of encoder/decoder (Bahdanau).
  - Simplify and avoid more recurrence (Luong).

# Translation Takeaways


- Parsing as translation:
  - Translation models can solve many “transduction” tasks.

- Attention only models:
  - Self-attention replaces recurrence, improves performance.
  - Use depth to model hierarchical structure.
  - Multi-headed attention allows interpretation of inputs.