Sequence to Sequence (seq2seq) + Attention

LING 574 Deep Learning for NLP

Shane Steinert-Threlkeld

Announcements

- Patas back! HW3 due tonight; HW4 no late penalty
- Unit tests and grading
- Edugrad, numpy, etc:
  - Numpy *only* inside of backward/forward methods of an Operation
  - @tensor_op: your Operations becomes methods that take Tensor arguments
  - With Tensors: must use these methods, not numpy
  - These operations build the graph, the plumbing for backprop, etc
- Broadcasting/shapes in edugrad (lack thereof :))
  - Remember: annotate shapes!! (hw3 ref will have some examples)
- Adagrad:
  - param._grad_hist: this is $G_{t,i}$
  - Order of operations: first update $G_{t,i}$, then apply update rule

Today's Plan

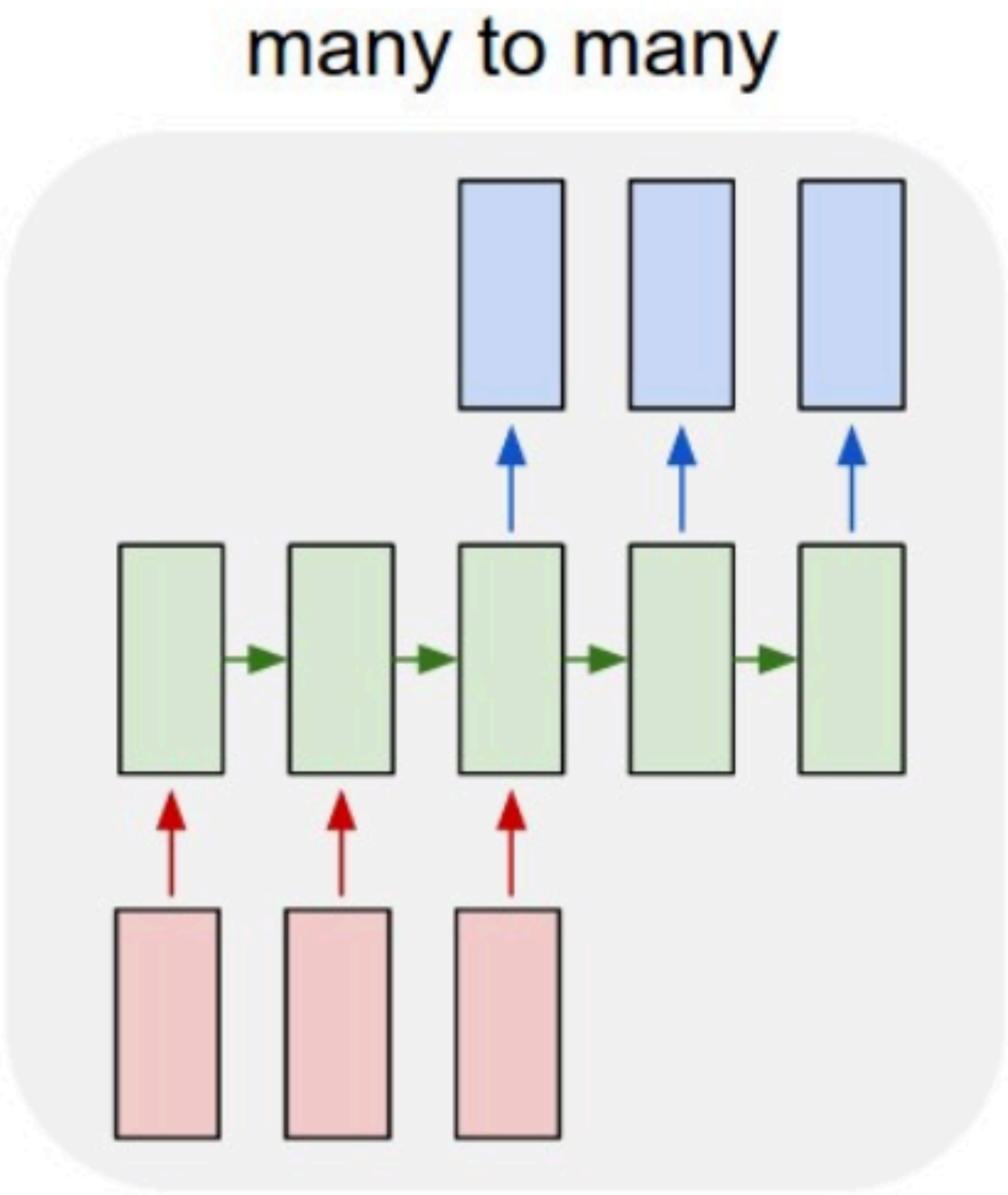
- Last time: RNNs for sequence processing
  - Motivation (long-distance dependencies), Vanilla/Elman, stacked/bidirectional, classification and LM
- Today:
  - Vanishing gradient problem for vanilla RNNs
  - *Gating* mechanisms / fancier RNNs to overcome this (LSTM, GRU)
  - Sequence-to-sequence tasks/models
    - Attention mechanism

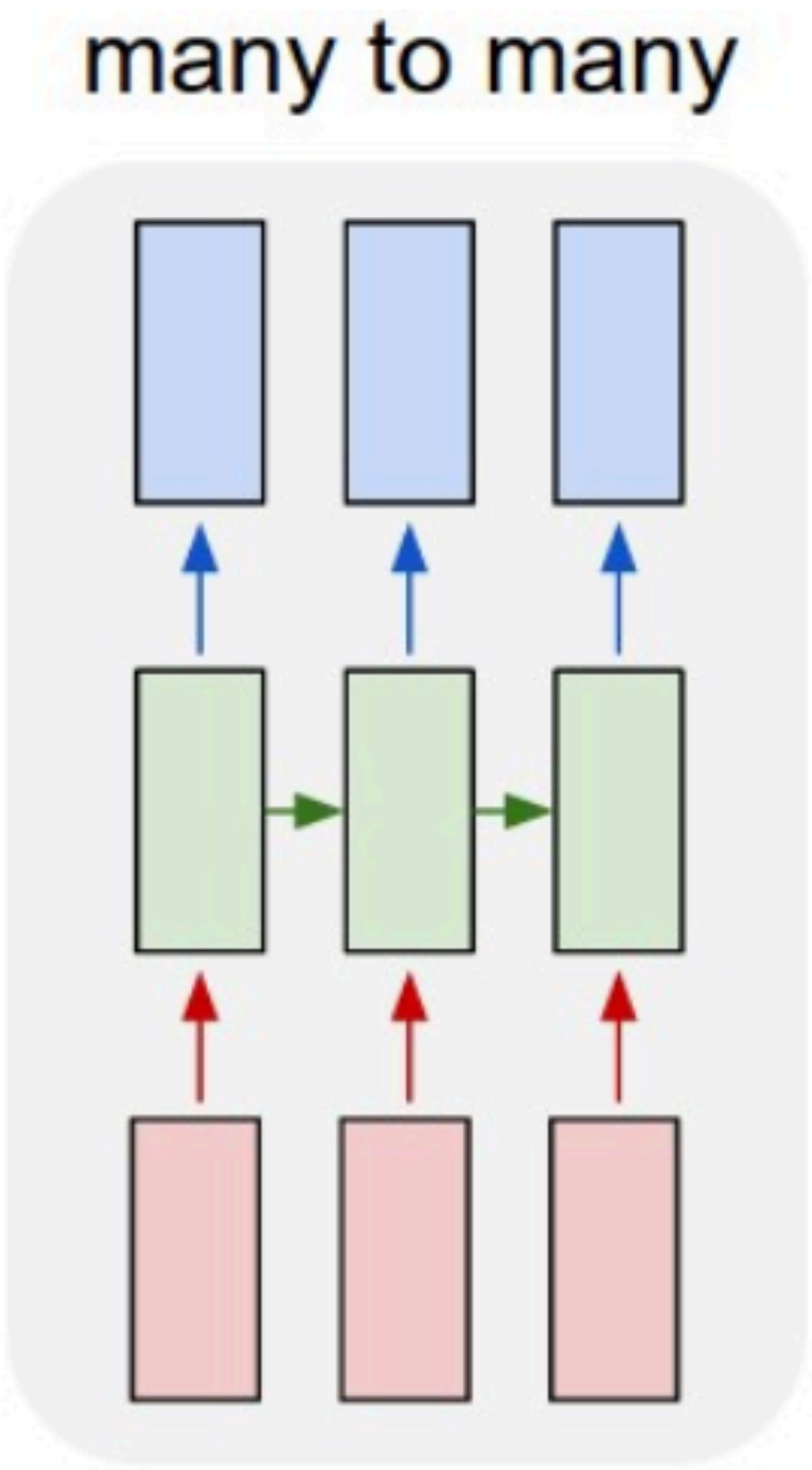
seq2seq: Overview

Sequence to sequence problems

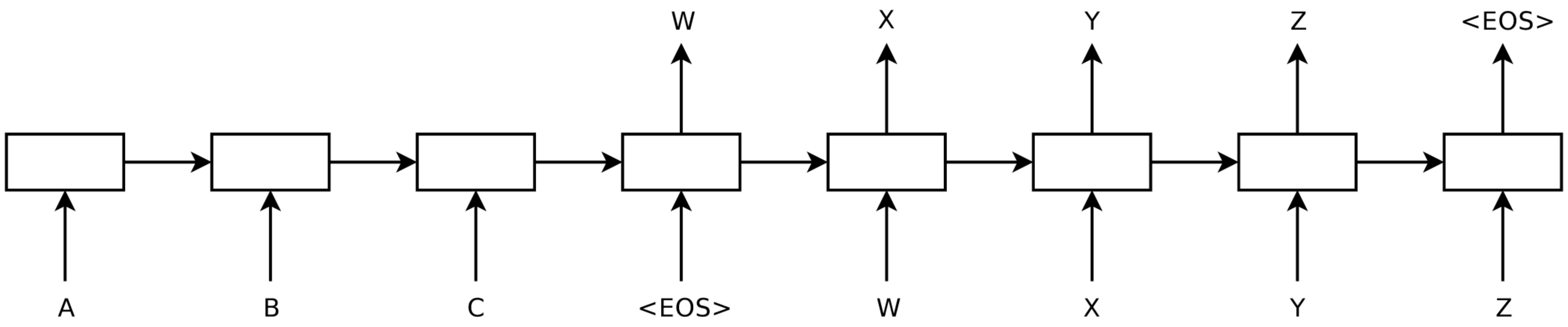
- Many NLP tasks can be construed as *sequence-to-sequence* problems
  - Machine translations: sequence of source lang tokens to sequence of target lang tokens
  - Parsing: “Shane talks.” —> “(S (NP (N Shane)) (VP V talks))”
    - Incl semantic parsing (“Shane talks.” -> “$\exists e$(talking$(e) \wedge$ Agent$(e,$ S$)$)”)
  - Summarization
  - …
- NB: not the same as *tagging*, which assigns a label to each position in a given sequence (POS tagging, language modeling)

Seq2seq vs Tagging



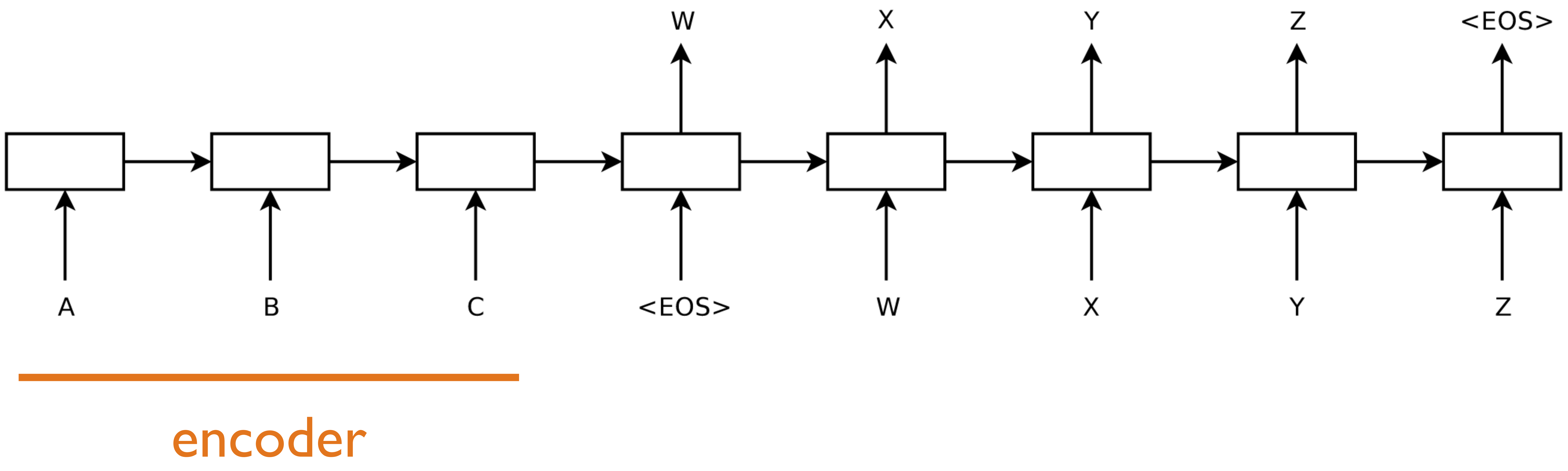


seq2seq architecture [e.g. NMT]



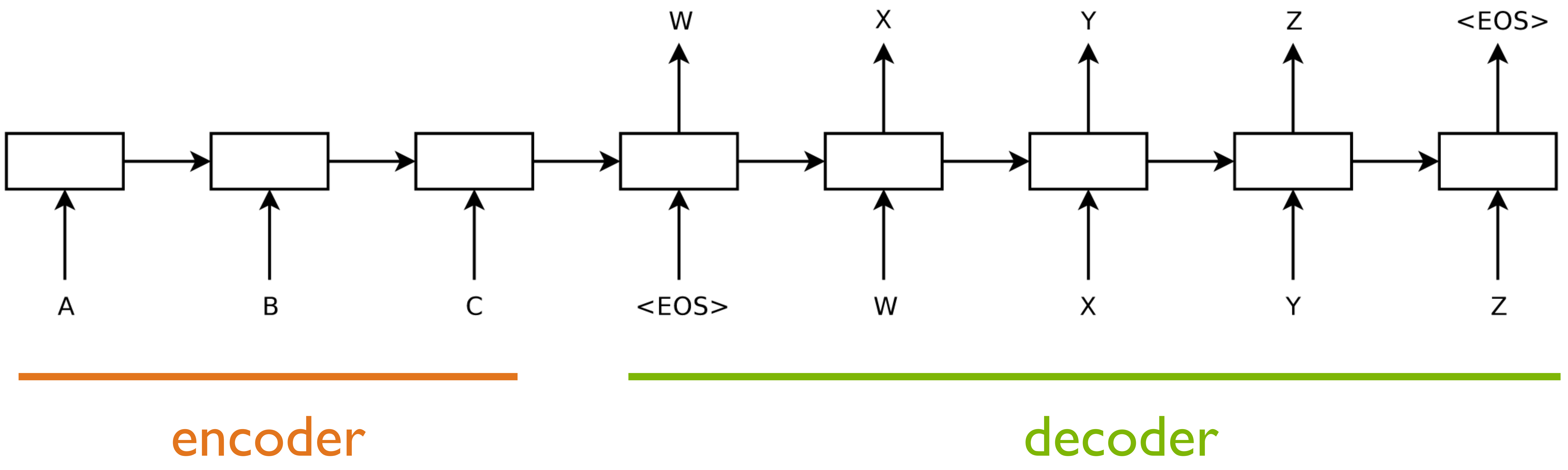
Sutskever et al 2013

seq2seq architecture [e.g. NMT]



Sutskever et al 2013

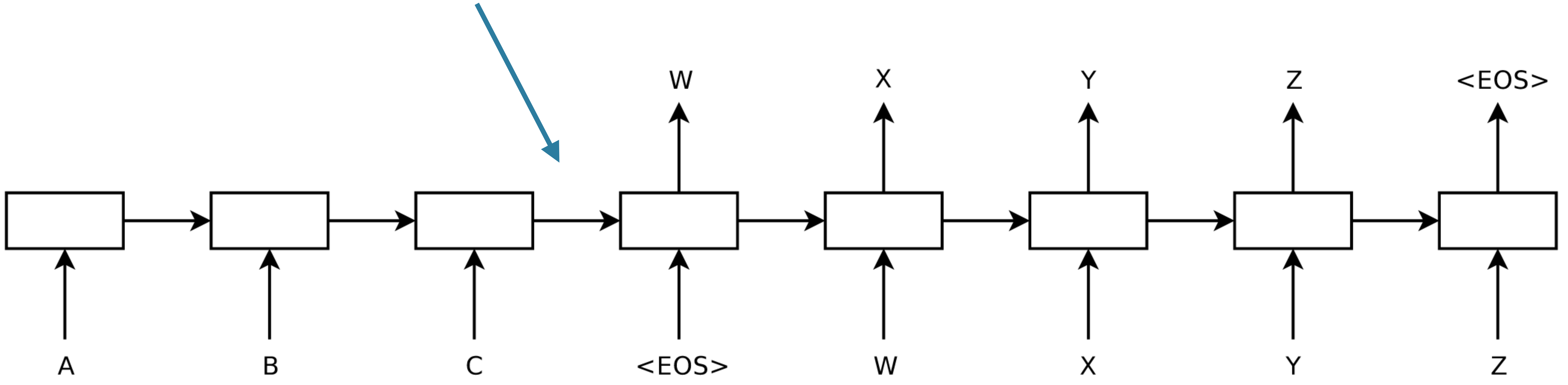
seq2seq architecture [e.g. NMT]



Sutskever et al 2013

seq2seq architecture [e.g. NMT]

Initial hidden state of decoder =
final hidden state of encoder



encoder

decoder

Sutskever et al 2013

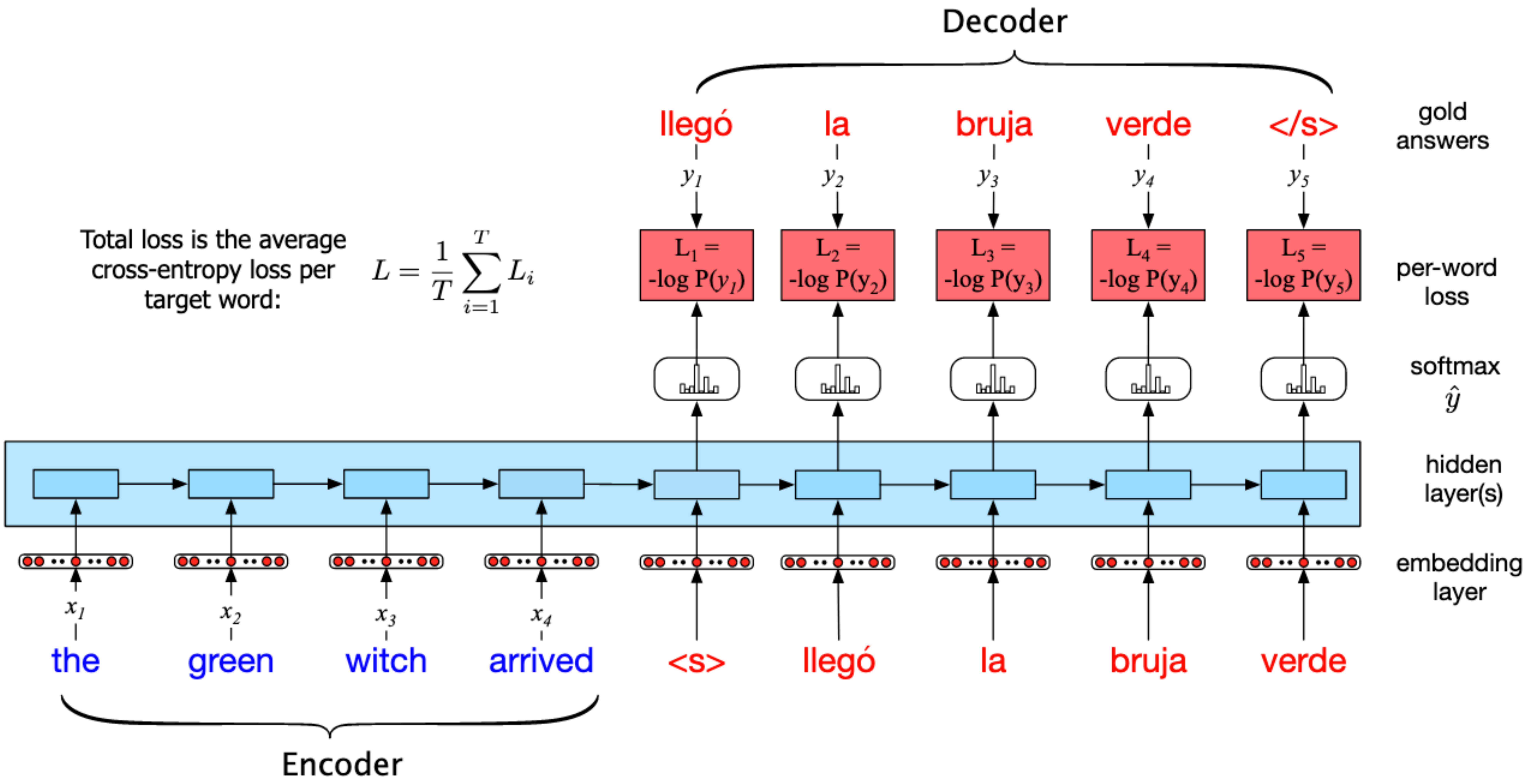
seq2seq architecture

- Two components:
  - Encoder
    - Input sequence —> vector representation (“context” vector)
  - Decoder
    - Vector (“context” vector) —> Output sequence

- High-level “API”: encoder/decoder can be different architectures (LSTM, GRU, Transformer, convolutional, …)

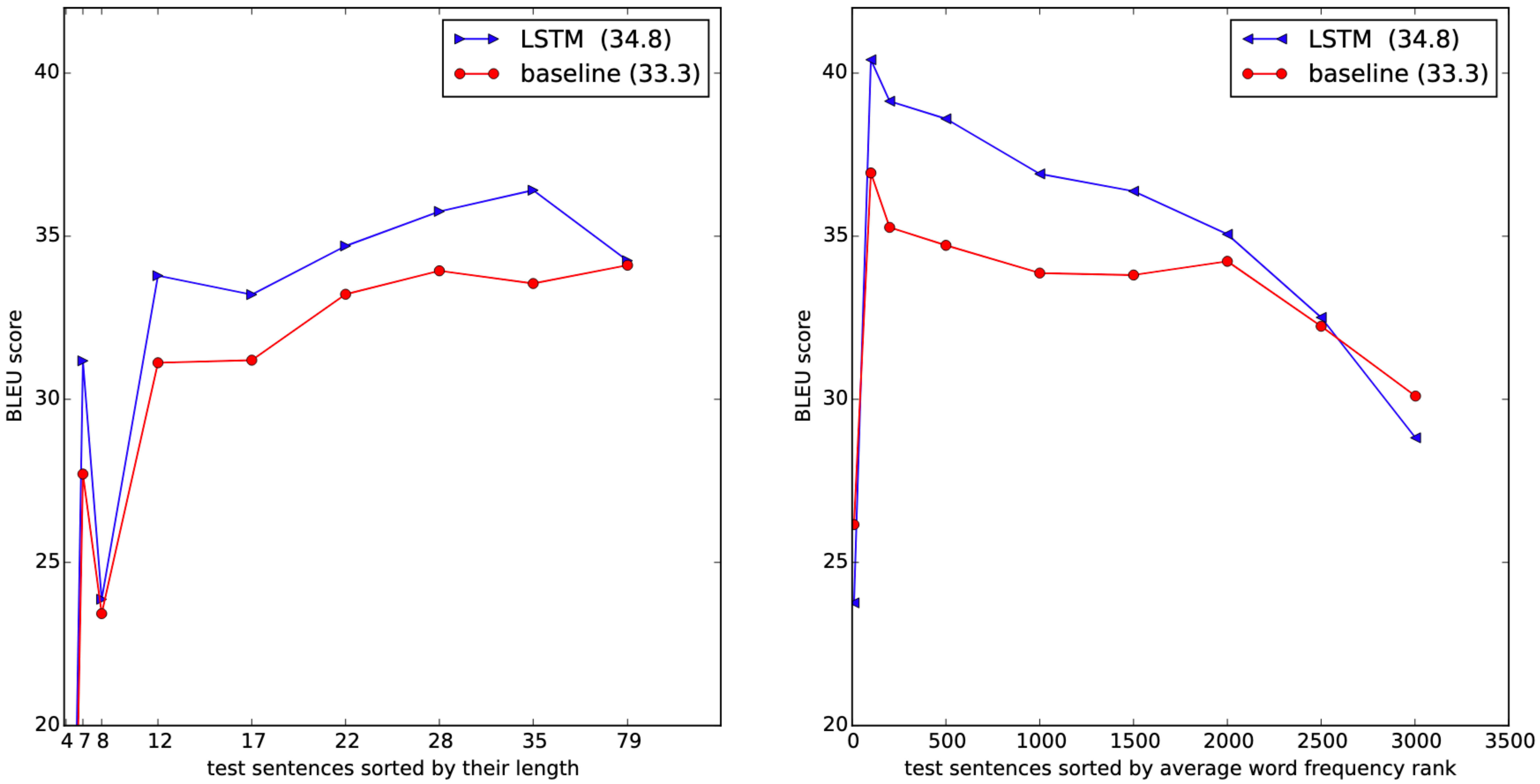
Training an encoder-decoder RNN

Total loss is the average cross-entropy loss per target word: $L = \frac{1}{T}\sum_{i=1}^{T} L_i$

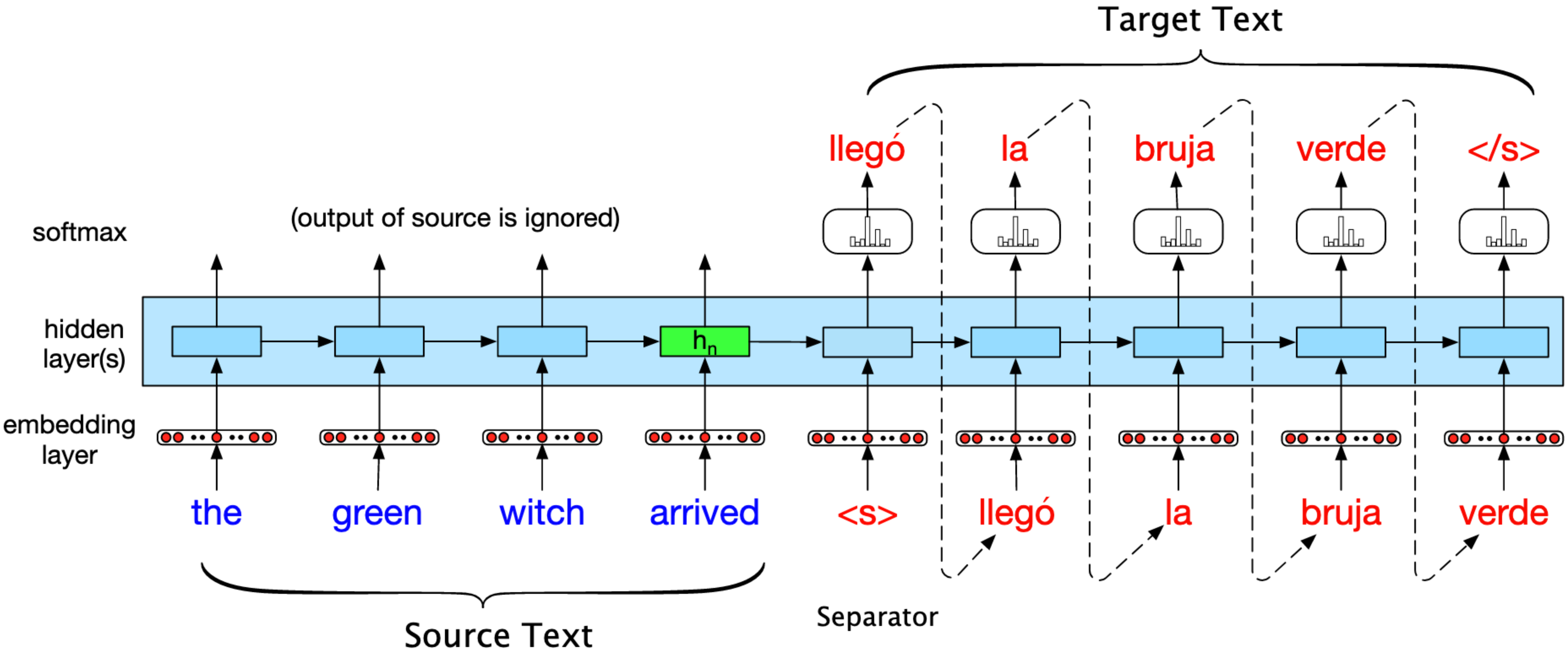


JM 11.3.1

seq2seq initial results



Inference / Generation



JM S11.3

Seq2seq interim summary

- Effectively, a seq2seq model is a *conditional* language model: the same kind of language model that we have seen, but conditioned on the context of the input sequence

$$P(y \mid x) = \prod_{i=1}^{|y|} P(y_i \mid x, y_{<i})$$

NMT Evaluation

- "Ideal": human evaluation (fluency, adequacy, ranking)
- BLEU (BiLingual Evaluation Understudy): roughly, n-gram overlap between reference translations and machine translations
  - Penalizes synonymous translations
    - METEOR, BERTScore attempt to alleviate
  - Low correlation with human ratings
- chrF++
  - Refinement of *character* n-gram F1 score
  - Seems to have better correlations
- In general: still no perfect solution

Source

la verdad, cuya madre es la historia, émula del tiempo, depósito de las acciones, testigo de lo pasado, ejemplo y aviso de lo presente, advertencia de lo por venir.

Reference

truth, whose mother is history, rival of time, storehouse of deeds, witness for the past, example and counsel for the present, and warning for the future.

Candidate 1

truth, whose mother is history, voice of time, deposit of actions, witness for the past, example and warning for the present, and warning for the future

Candidate 2

the truth, which mother is the history, émula of the time, deposition of the shares, witness of the past, example and notice of the present, warning of it for coming

JM S11.8

Outstanding Issues in NMT

- Evaluation: automated metrics are all flawed
  - "Tangled Up in BLEU"
- Low-resource / unsupervised MT
  - Can we build good translation models in the absence of huge amounts of parallel text?
    - Common technique: *backtranslation*
  - http://www.statmt.org/wmt20/unsup_and_very_low_res/
  - http://turing.iimas.unam.mx/americasnlp/st.html
  - https://www.aclweb.org/anthology/2020.acl-main.560/

Statistical Machine Translation: Alignment

Statistical Machine Translation (90s-2010s)

- Goal: find best translation *y* (e.g. English) of source sentence *x* (e.g. French)

$$\arg\max_y P(y \mid x)$$

- Use Bayes to decompose into two components:

$$\arg\max_y P(x \mid y)P(y)$$

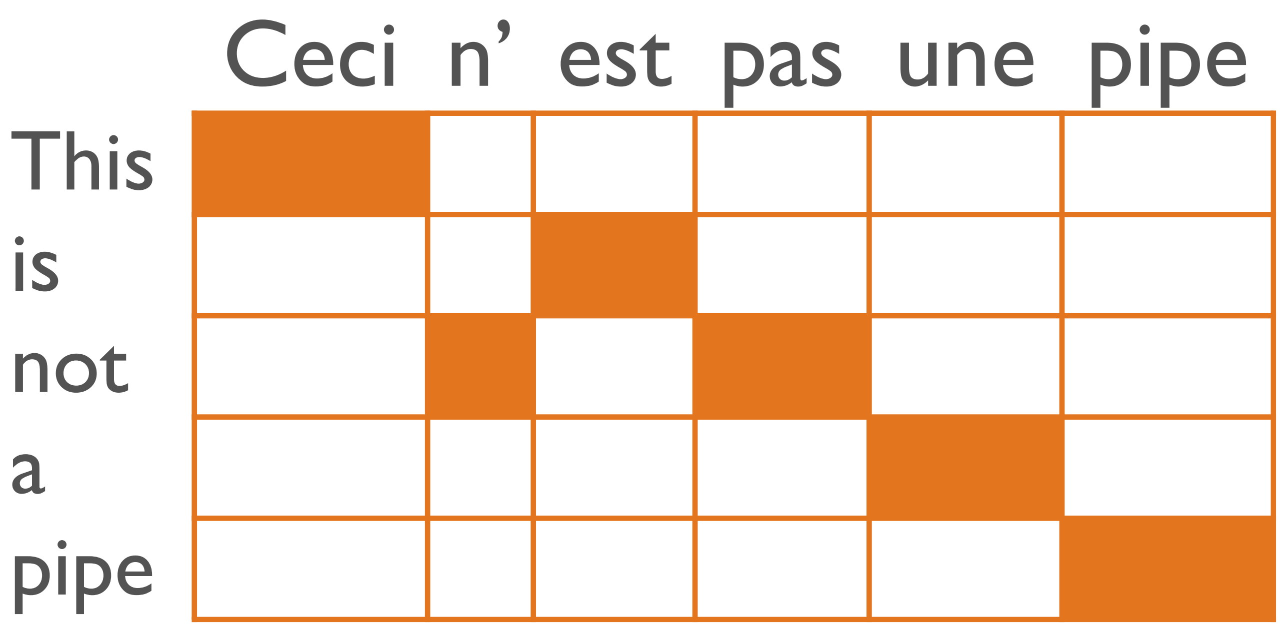
  - Core translation model P(xly)
  - Language model P(y): produce good / fluent target language text (e.g. English)

Alignment

- Most SMT systems factored through an *alignment*
  - Correspondence between words/phrases in source and target sentence
  - Typologically different languages have, e.g., very different word order (see JM 11.1 for more examples)
- Add alignment as a latent variable:

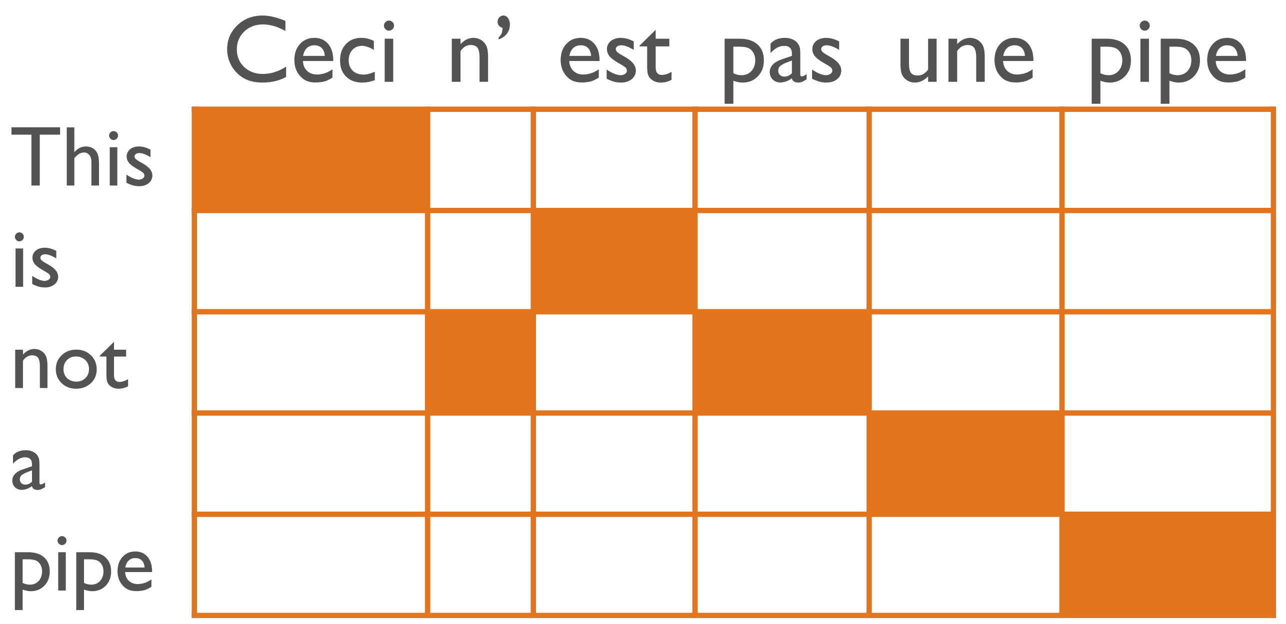
$$P(x, a \mid y)$$

Alignment, example



Alignment, example

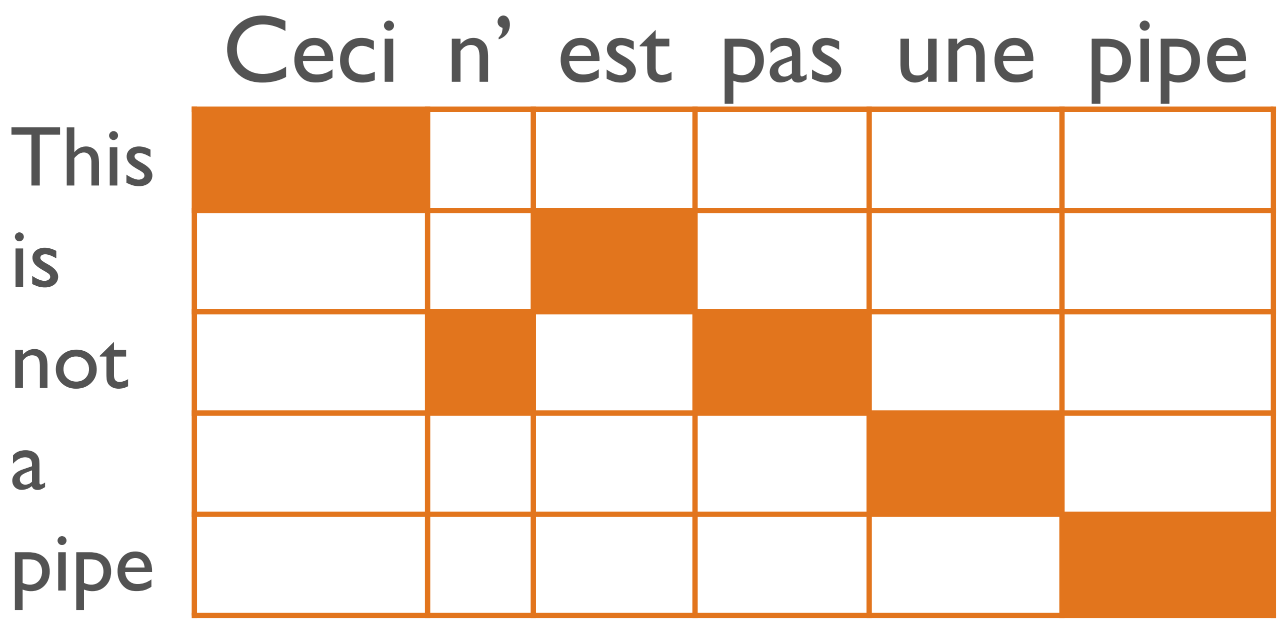
Ceci n’ est pas une pipe



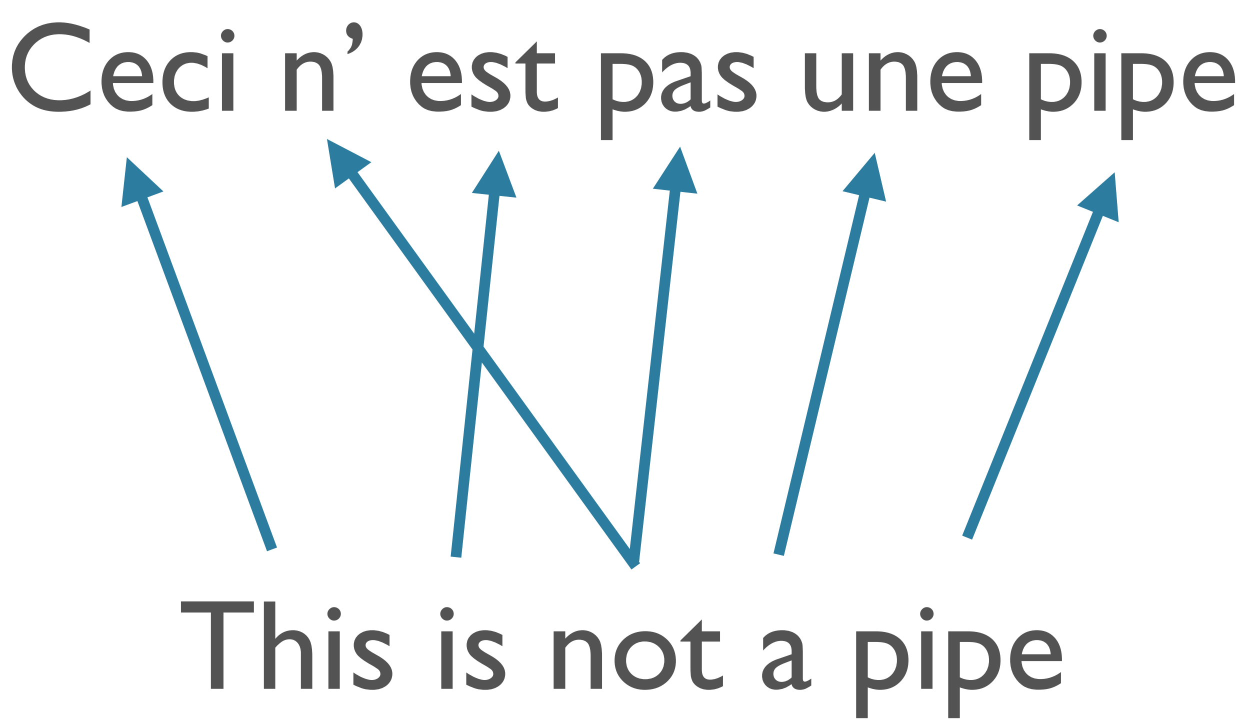
Alignment, example

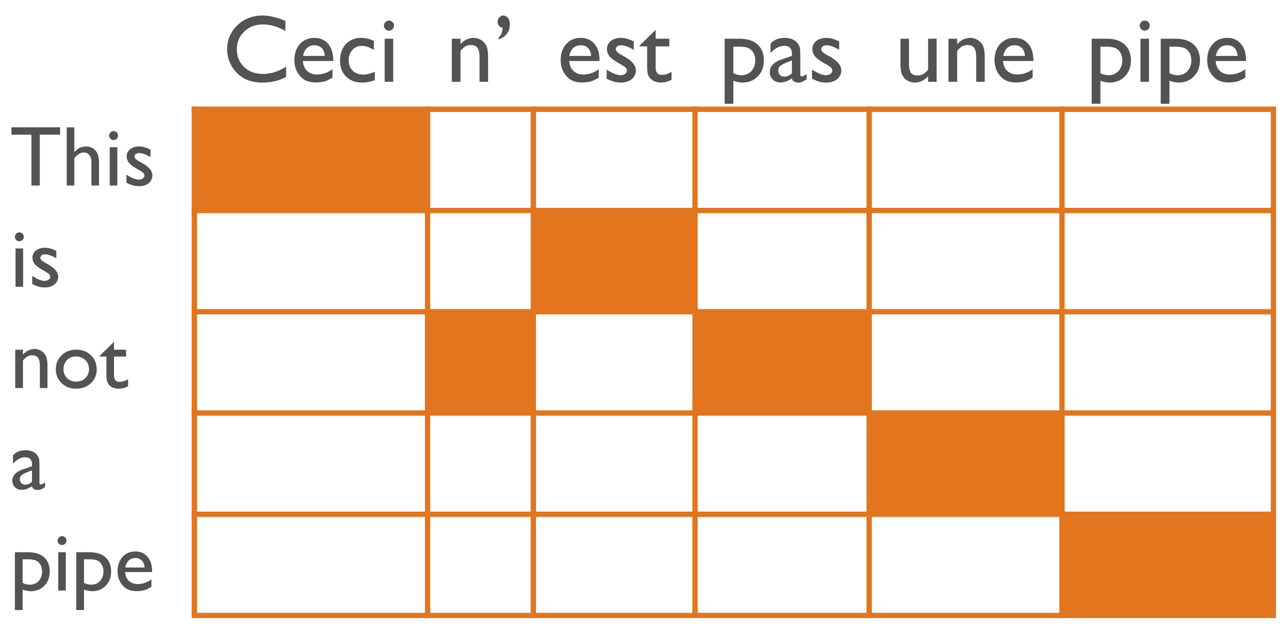
Ceci n’ est pas une pipe

This is not a pipe



Alignment, example



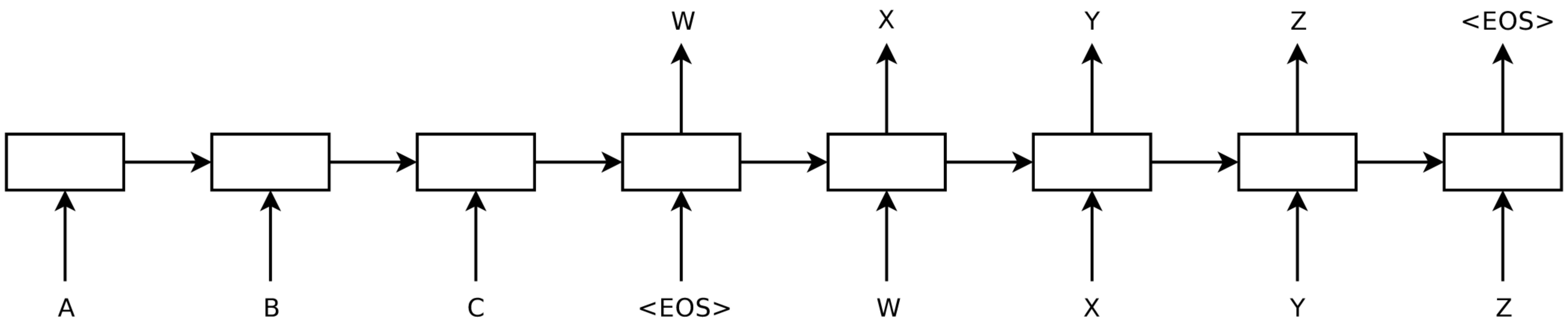


SMT Difficulties

- Features for alignment:
  - Probability of particular pairs aligning (lexicon / bilingual dictionary)
  - Probability of a word aligning to a phrase (in general)
- More generally:
  - Huge amounts of feature engineering
  - Reliance on human curated resources like dictionaries
  - Most of the above are *language-pair-specific*, have to be repeated
- NMT was one of the first major success stories of neural methods in NLP:
  - End-to-end systems, “language-agnostic” models, equal/better performance

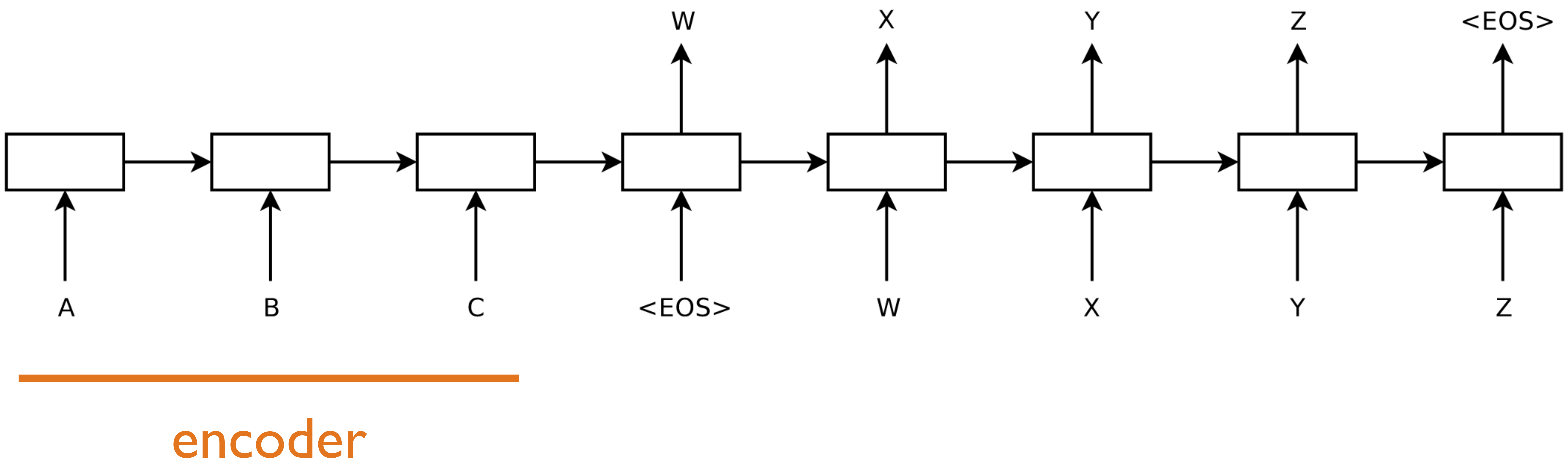
Attention

seq2seq architecture: problem



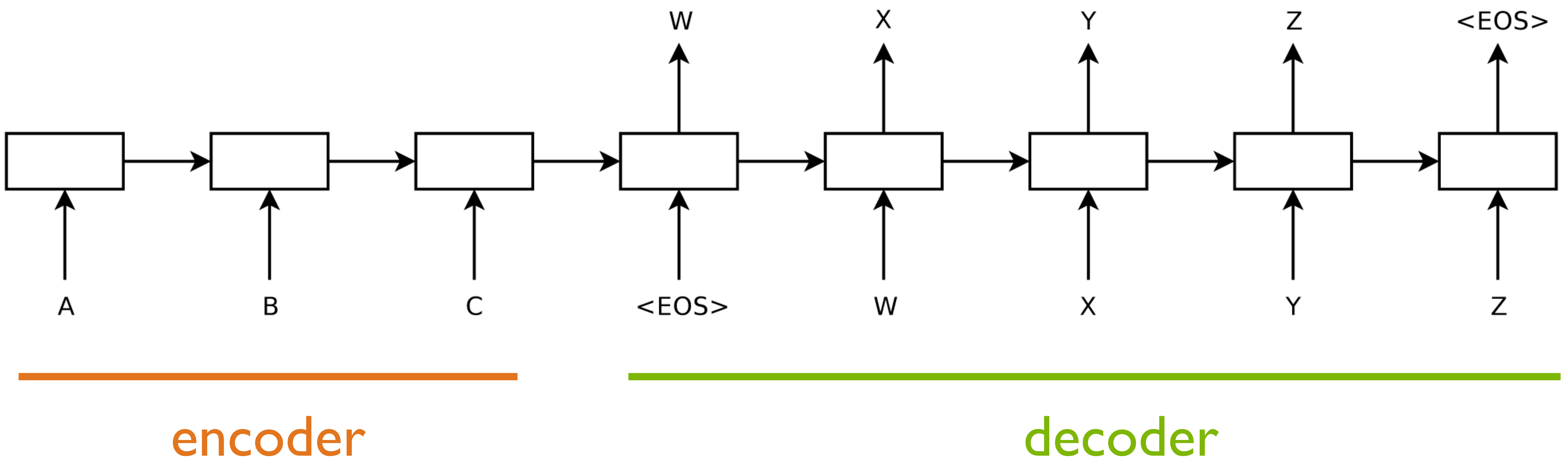
[Sutskever et al 2013]

seq2seq architecture: problem



Sutskever et al 2013

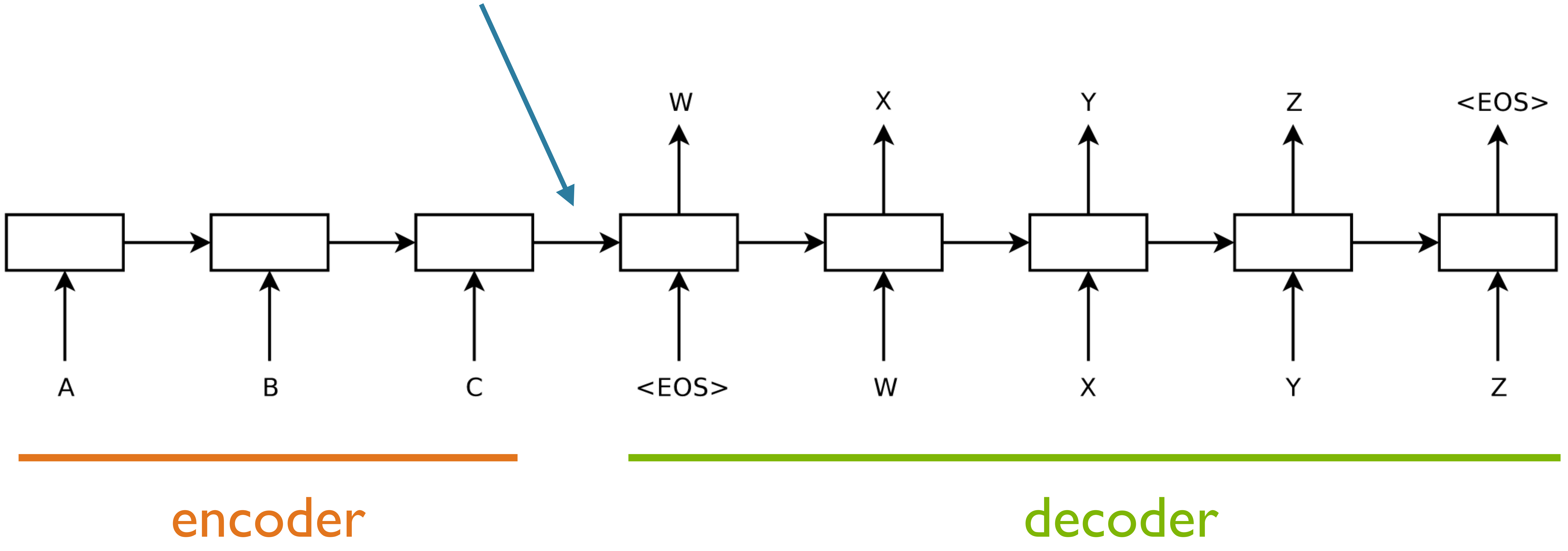
seq2seq architecture: problem



Sutskever et al 2013

seq2seq architecture: problem

Decoder can only see info in this one vector
all info about source must be "crammed" into here

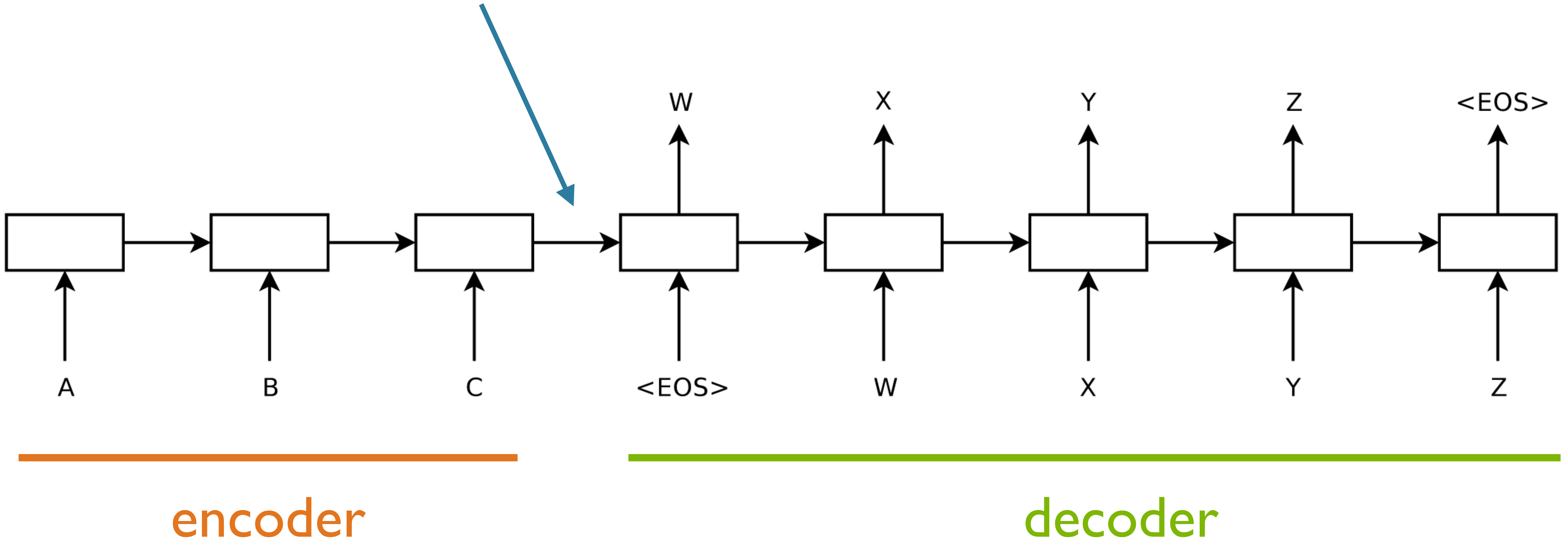


Sutskever et al 2013

seq2seq architecture: problem

Decoder can only see info in this one vector
all info about source must be "crammed" into here

Mooney 2014: "You can't cram the meaning of a whole %&!$# sentence into a single $&!#* vector!"



Sutskever et al 2013

Neural Machine Translation by Jointly Learning to Align and Translate

Dzmitry Bahdanau
Jacobs University Bremen, Germany

KyungHyun Cho **Yoshua Bengio***
Université de Montréal

Abstract

Neural machine translation is a recently proposed approach to machine translation. Unlike the traditional statistical machine translation, the neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance. The models proposed recently for neural machine translation often belong to a family of encoder–decoders and encode a source sentence into a fixed-length vector from which a decoder generates a translation. In this paper, we conjecture that the use of a fixed-length vector is a bottleneck in improving the performance of this basic encoder–decoder architecture, and propose to extend this by allowing a model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word, without having to form these parts as a hard segment explicitly. With this new approach, we achieve a translation performance comparable to the existing state-of-the-art phrase-based system on the task of English-to-French translation. Furthermore, qualitative analysis reveals that the (soft-)alignments found by the model agree well with our intuition.

source

Neural Machine Translation by Jointly Learning to Align and Translate

Dzmitry Bahdanau
Jacobs University Bremen, Germany

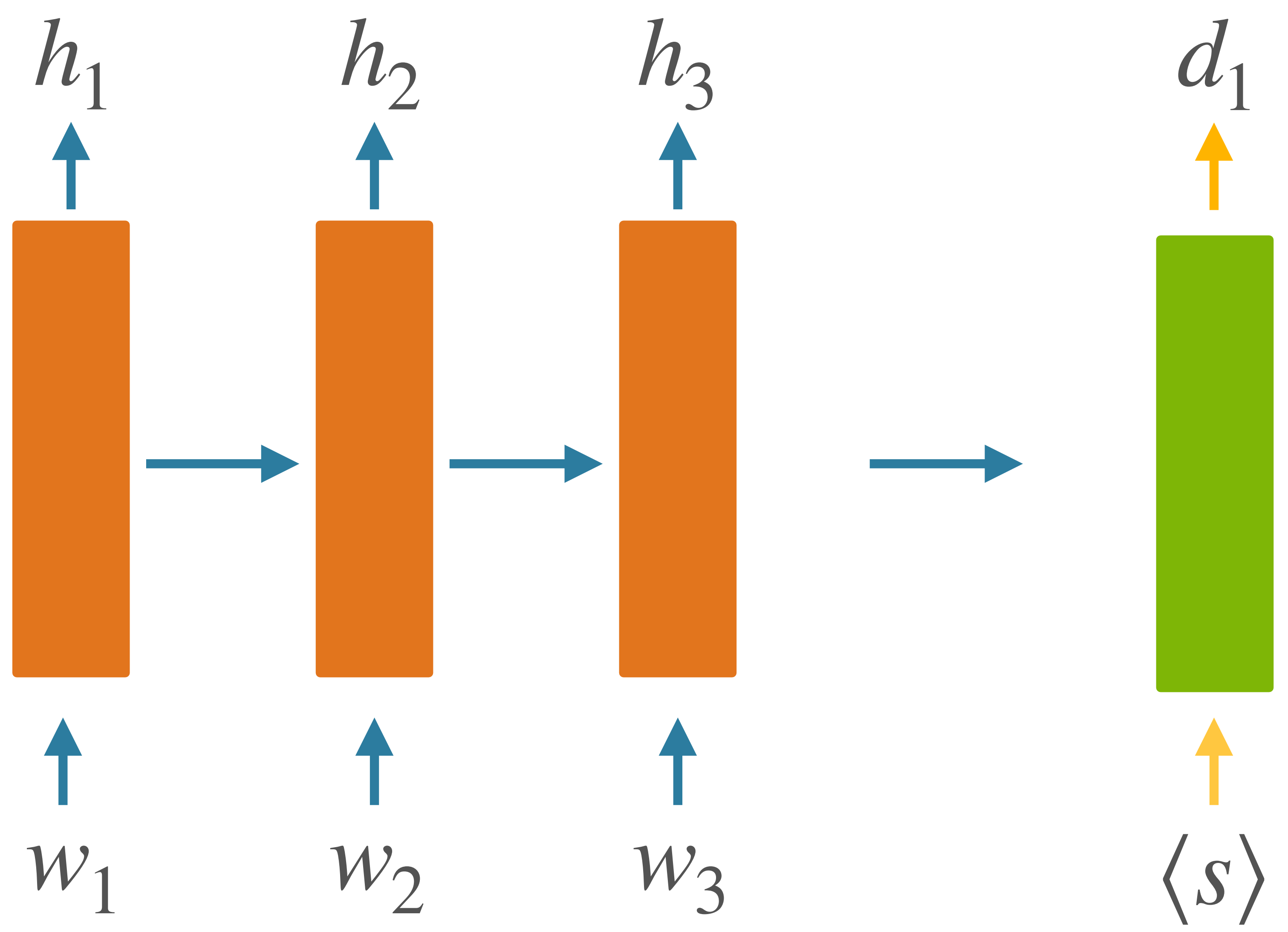
KyungHyun Cho **Yoshua Bengio***
Université de Montréal

Abstract

Neural machine translation is a recently proposed approach to machine translation. Unlike the traditional statistical machine translation, the neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance. The models proposed recently for neural machine translation often belong to a family of encoder–decoders and encode a source sentence into a fixed-length vector from which a decoder generates a translation. In this paper, we conjecture that the use of a fixed-length vector is a bottleneck in improving the performance of this basic encoder–decoder architecture, and propose to extend this by allowing a model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word, without having to form these parts as a hard segment explicitly. With this new approach, we achieve a translation performance comparable to the existing state-of-the-art phrase-based system on the task of English-to-French translation. Furthermore, qualitative analysis reveals that the (soft-)alignments found by the model agree well with our intuition.

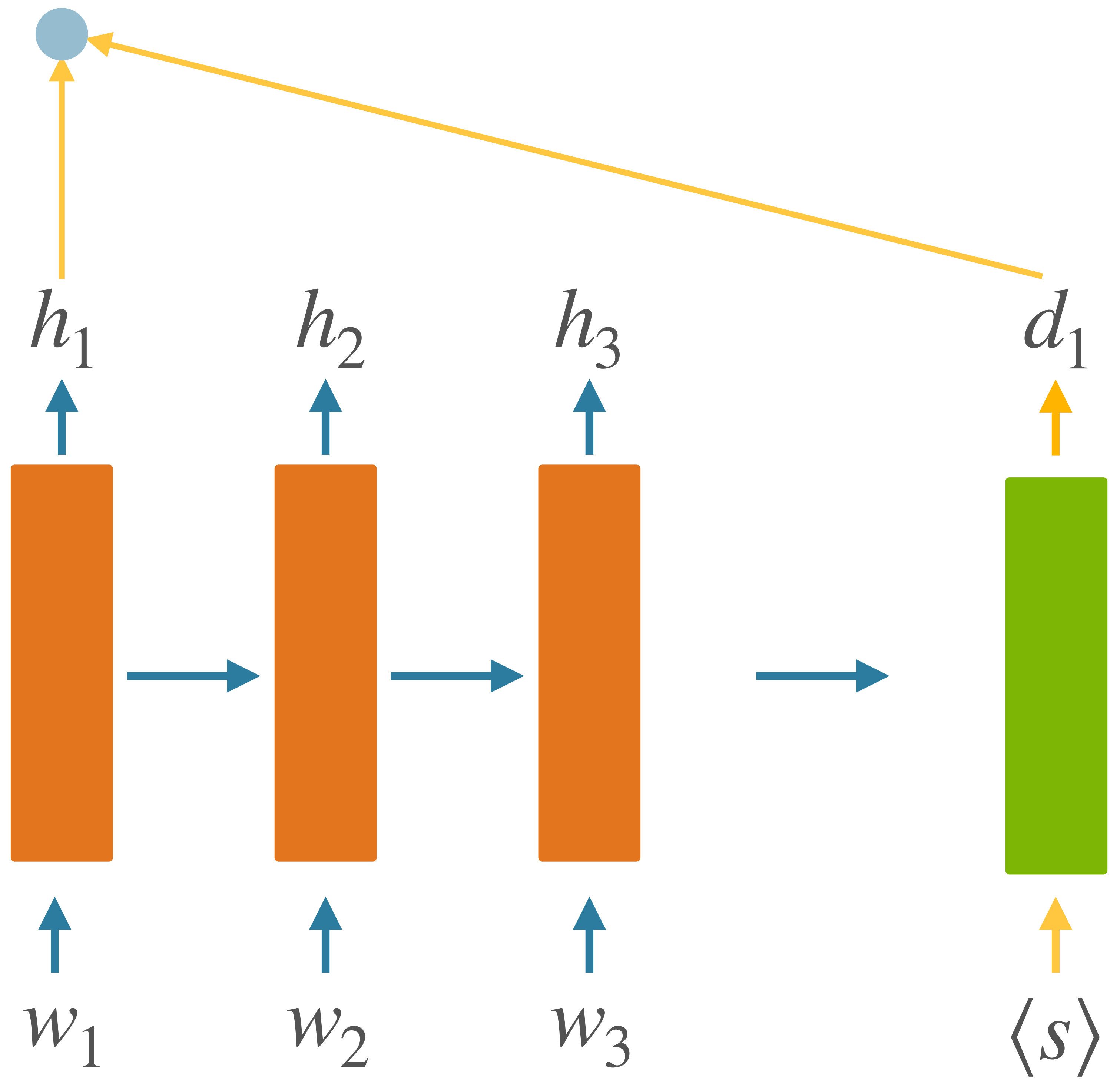
source

Adding Attention



Badhanau et al 2014
Luong et al 2015

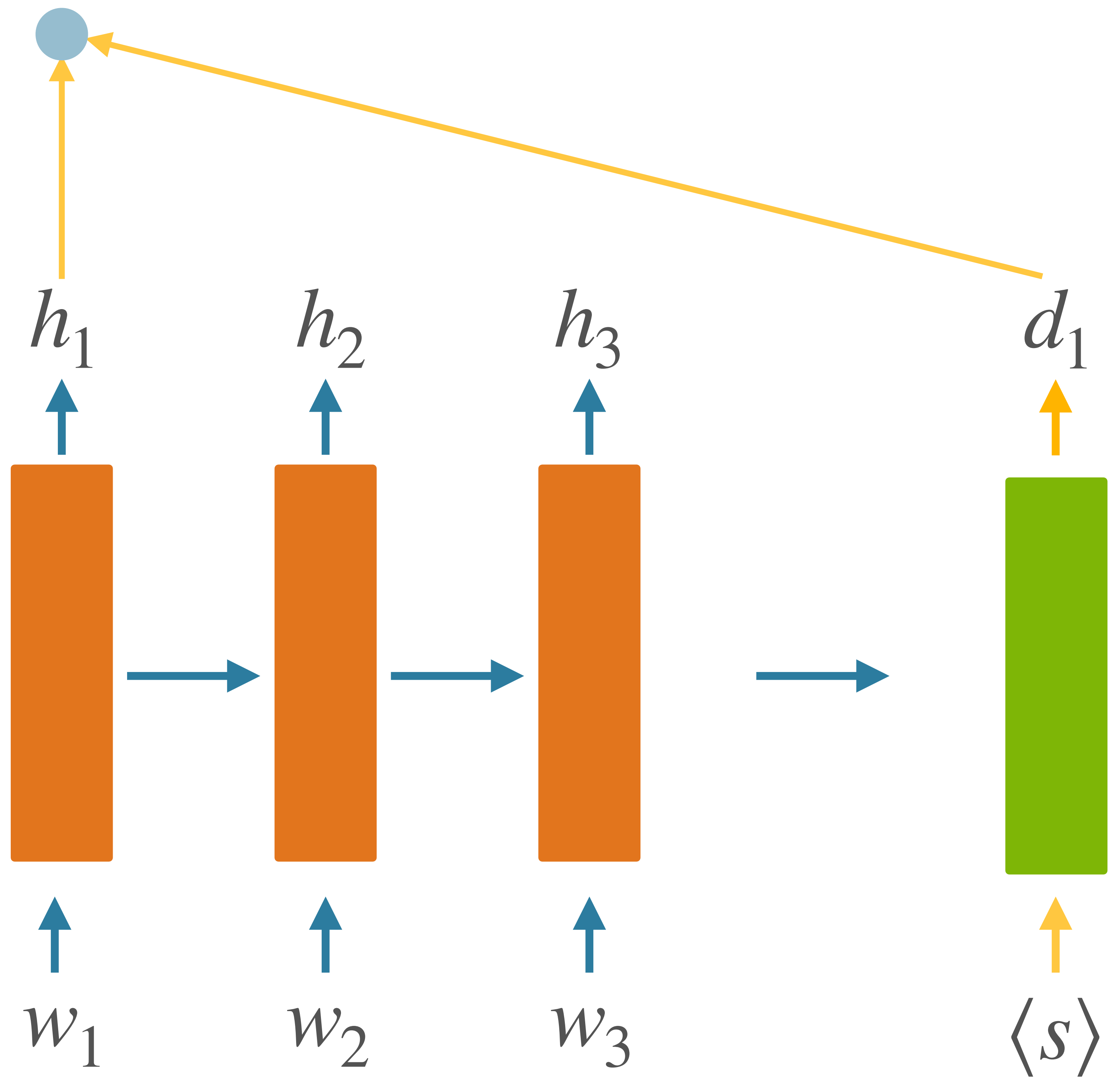
Adding Attention



Badhanau et al 2014
Luong et al 2015

Adding Attention

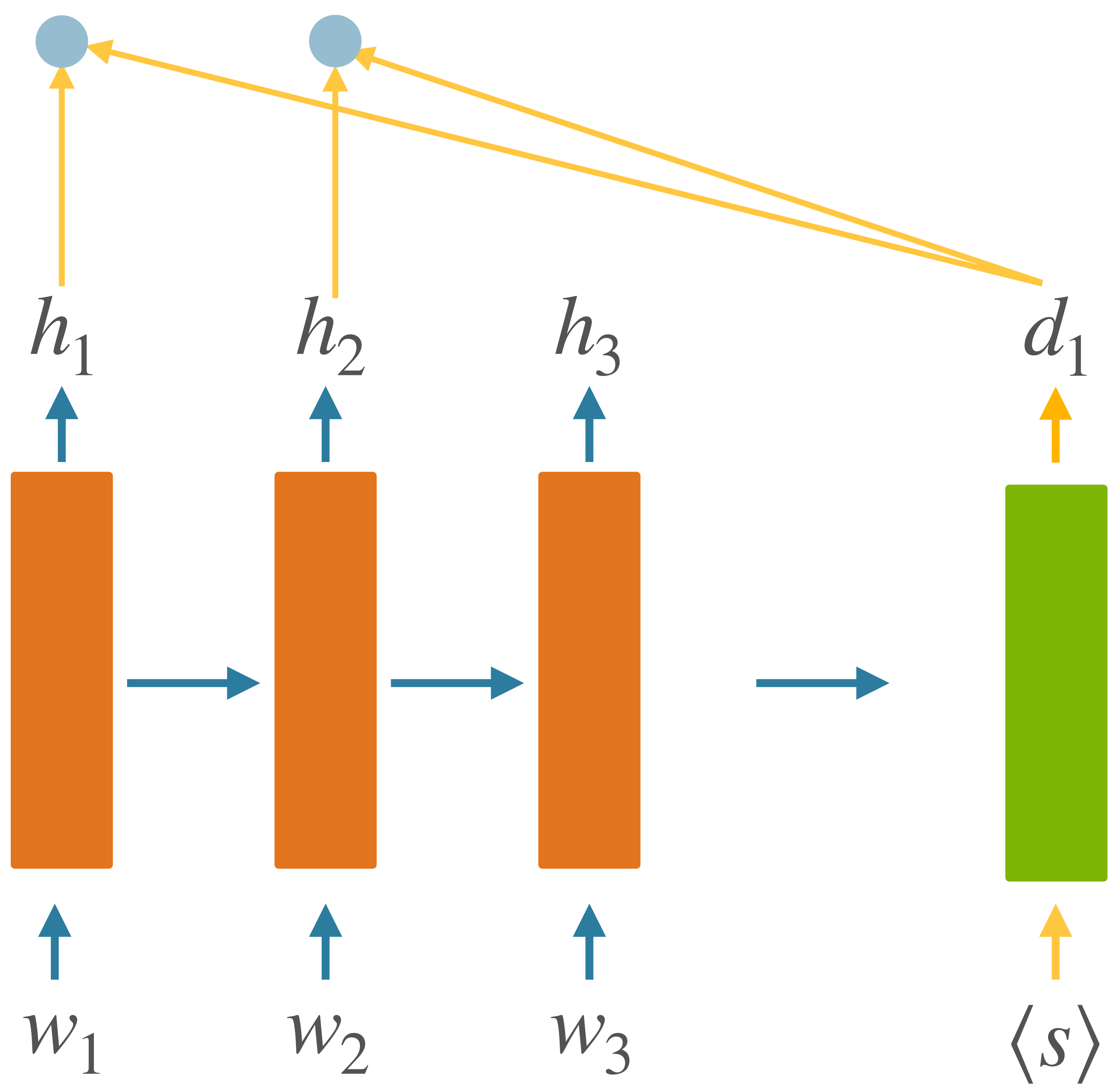
$\alpha_{ij} = a(h_j, d_i)$
(dot product usually)



Badhanau et al 2014
Luong et al 2015

Adding Attention

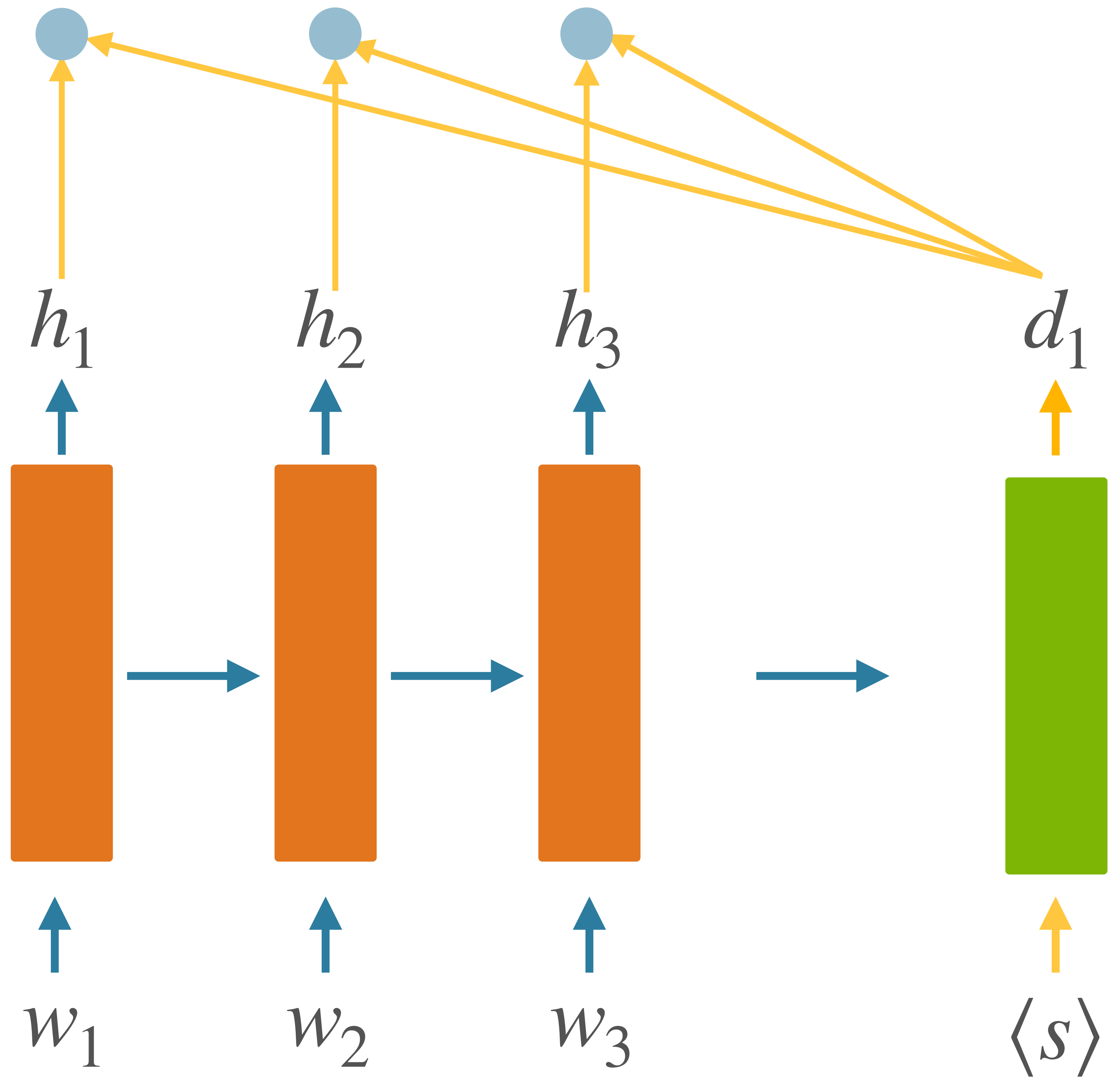
$\alpha_{ij} = a(h_j, d_i)$
(dot product usually)



Badhanau et al 2014
Luong et al 2015

Adding Attention

$\alpha_{ij} = a(h_j, d_i)$
(dot product usually)

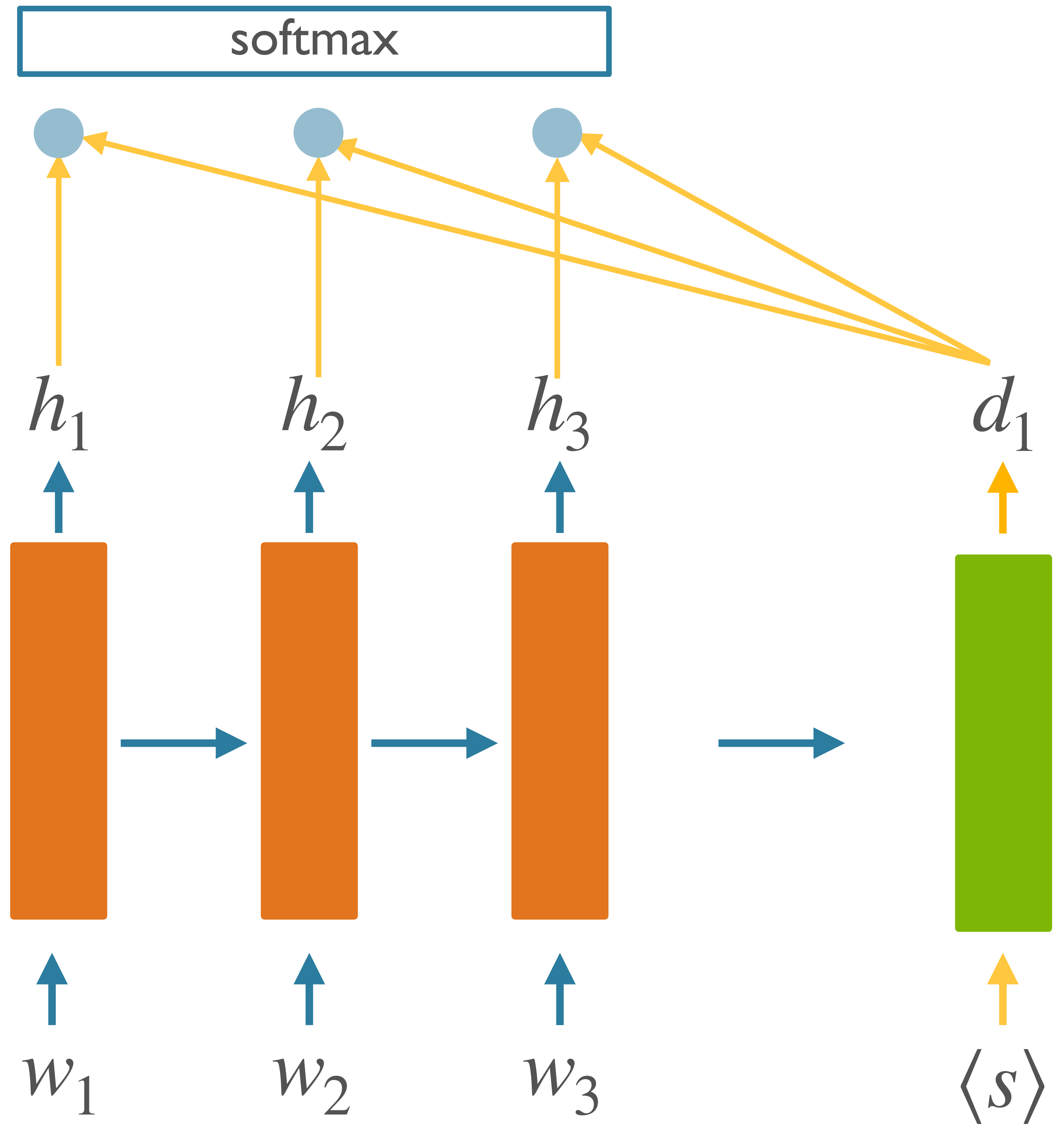


[Badhanau et al 2014](#)
[Luong et al 2015](#)

Adding Attention

$e_{ij} = \text{softmax}(\alpha)_j$

$\alpha_{ij} = a(h_j, d_i)$
(dot product usually)



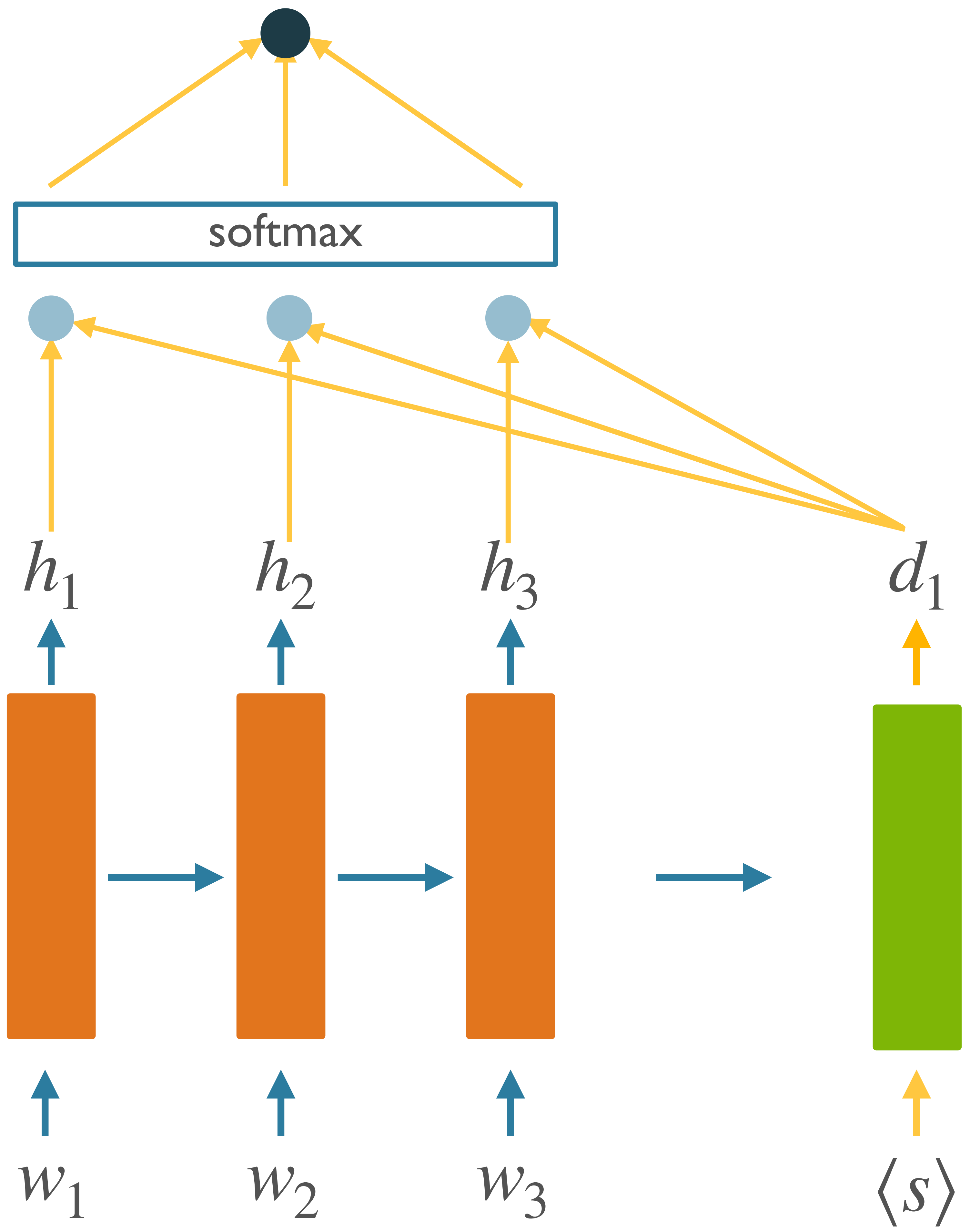
[Badhanau et al 2014](#)
[Luong et al 2015](#)

Adding Attention

$c_i = \Sigma_j e_{ij} h_j$

$e_{ij} = \text{softmax}(\alpha)_j$

$\alpha_{ij} = a(h_j, d_i)$
(dot product usually)



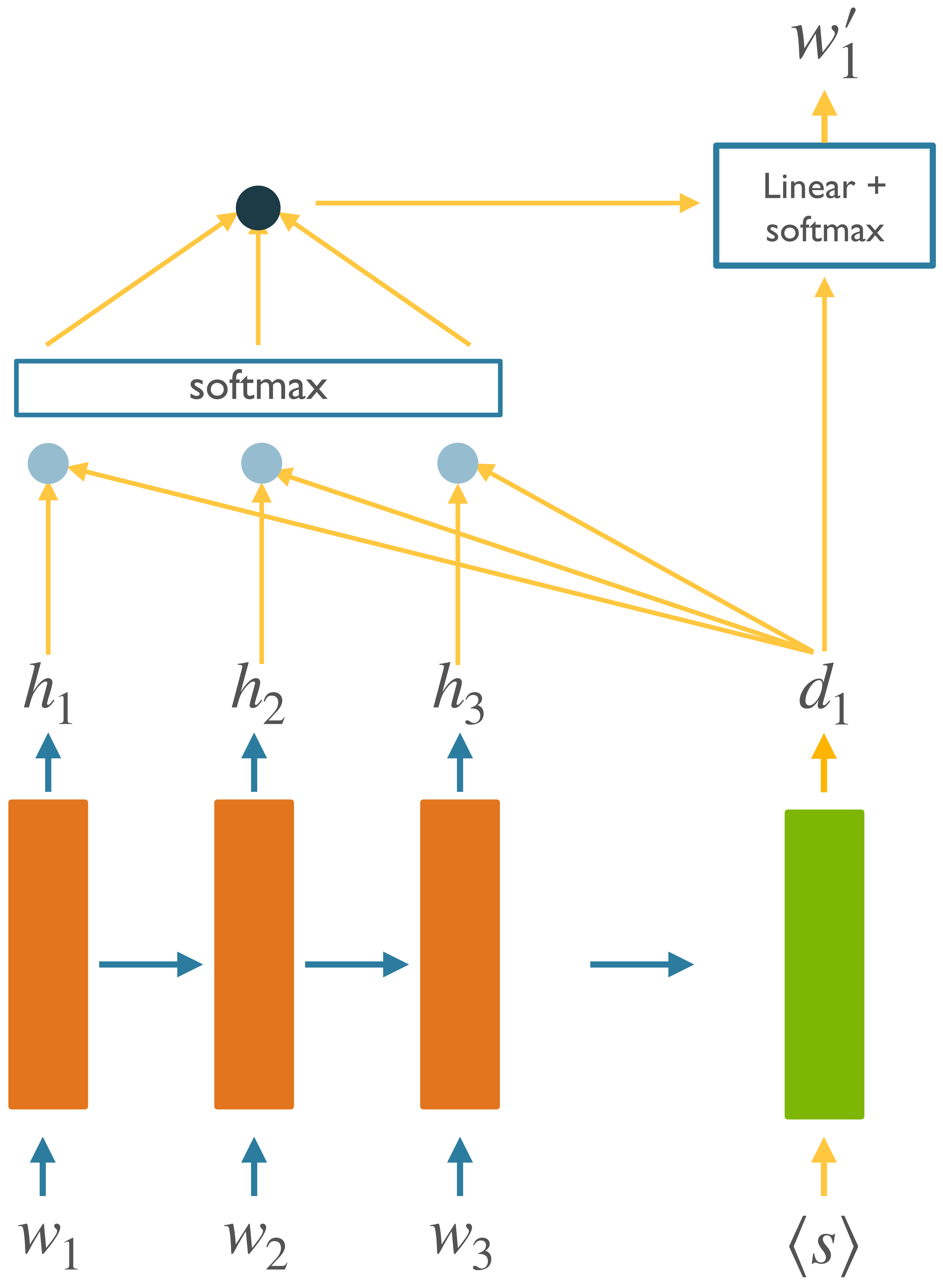
Badhanau et al 2014
Luong et al 2015

Adding Attention

$c_i = \Sigma_j e_{ij} h_j$

$e_{ij} = \text{softmax}(\alpha)_j$

$\alpha_{ij} = a(h_j, d_i)$
(dot product usually)



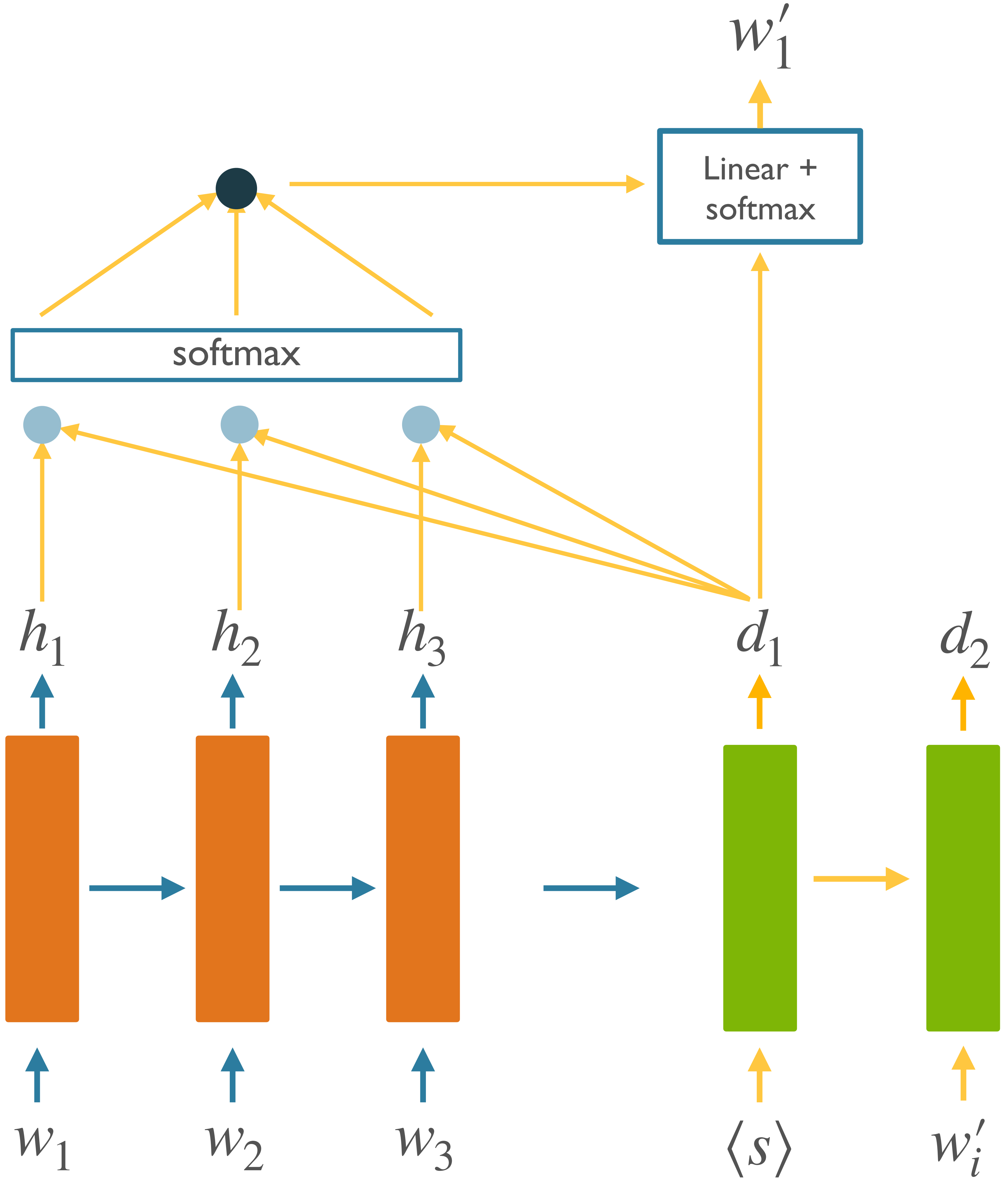
Badhanau et al 2014
Luong et al 2015

Adding Attention

$c_i = \Sigma_j e_{ij} h_j$

$e_{ij} = \text{softmax}(\alpha)_j$

$\alpha_{ij} = a(h_j, d_i)$
(dot product usually)



Badhanau et al 2014
Luong et al 2015

Attention, Generally

Attention, Generally

- A *query* q pays attention to some *values* $\{v_k\}$ based on similarity with some *keys* $\{k_v\}$.

Attention, Generally

- A *query* q pays attention to some *values* $\{v_k\}$ based on similarity with some *keys* $\{k_v\}$.
- Dot-product attention:

$$\alpha_j = q \cdot k_j$$

$$e_j = e^{\alpha_j} / \Sigma_j e^{\alpha_j}$$

$$c = \Sigma_j e_j v_j$$

Attention, Generally

- A *query* q pays attention to some *values* $\{v_k\}$ based on similarity with some *keys* $\{k_v\}$.
- Dot-product attention:

$$\alpha_j = q \cdot k_j$$
$$e_j = e^{\alpha_j} / \Sigma_j e^{\alpha_j}$$
$$c = \Sigma_j e_j v_j$$

- In the previous example: encoder hidden states played *both* the keys and the values roles.

Why attention?

Why attention?

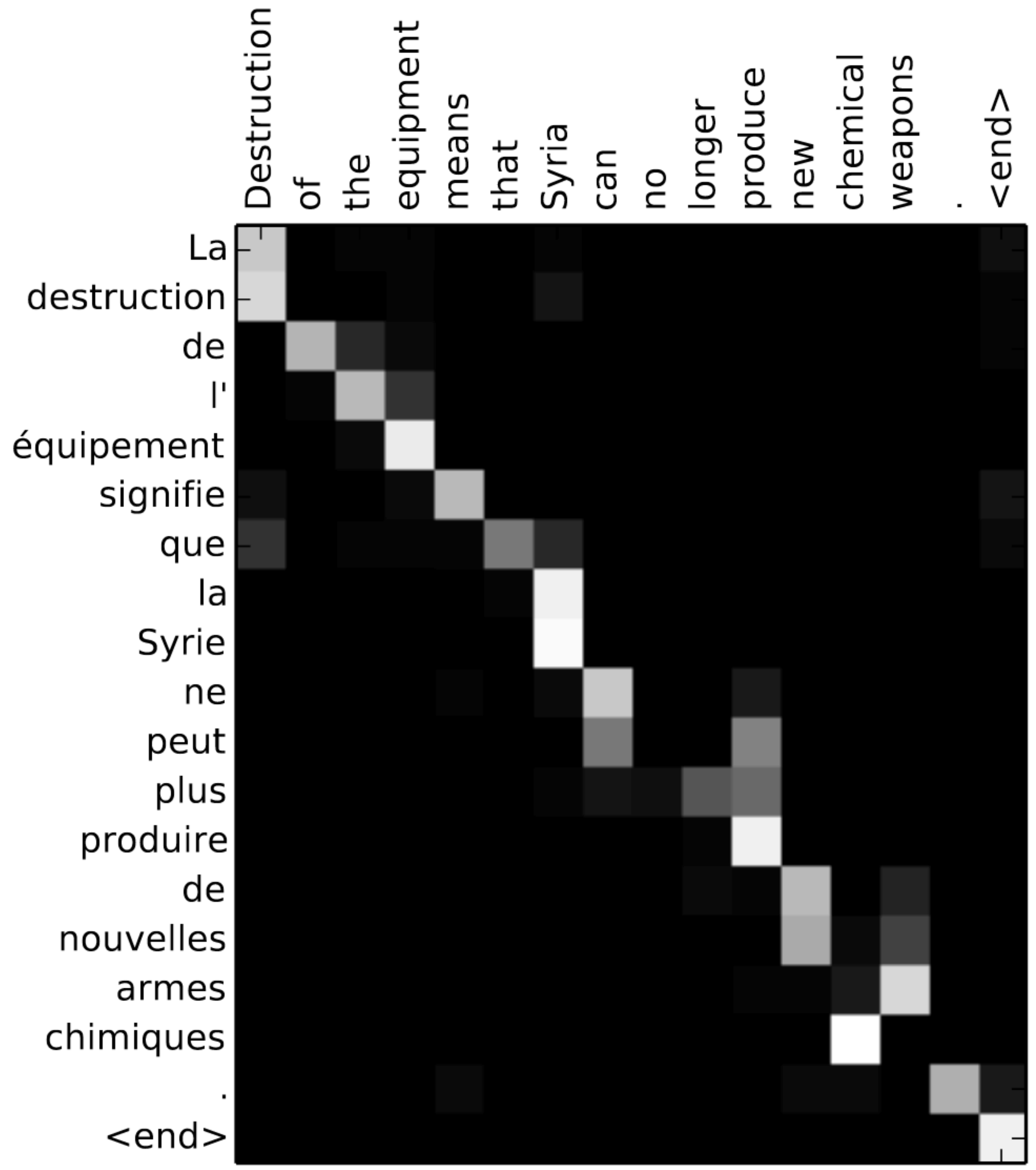
- Incredibly useful (for performance)
  - By “solving” the bottleneck issue

Why attention?

- Incredibly useful (for performance)
  - By “solving” the bottleneck issue
- Aids interpretability (maybe)

Why attention?

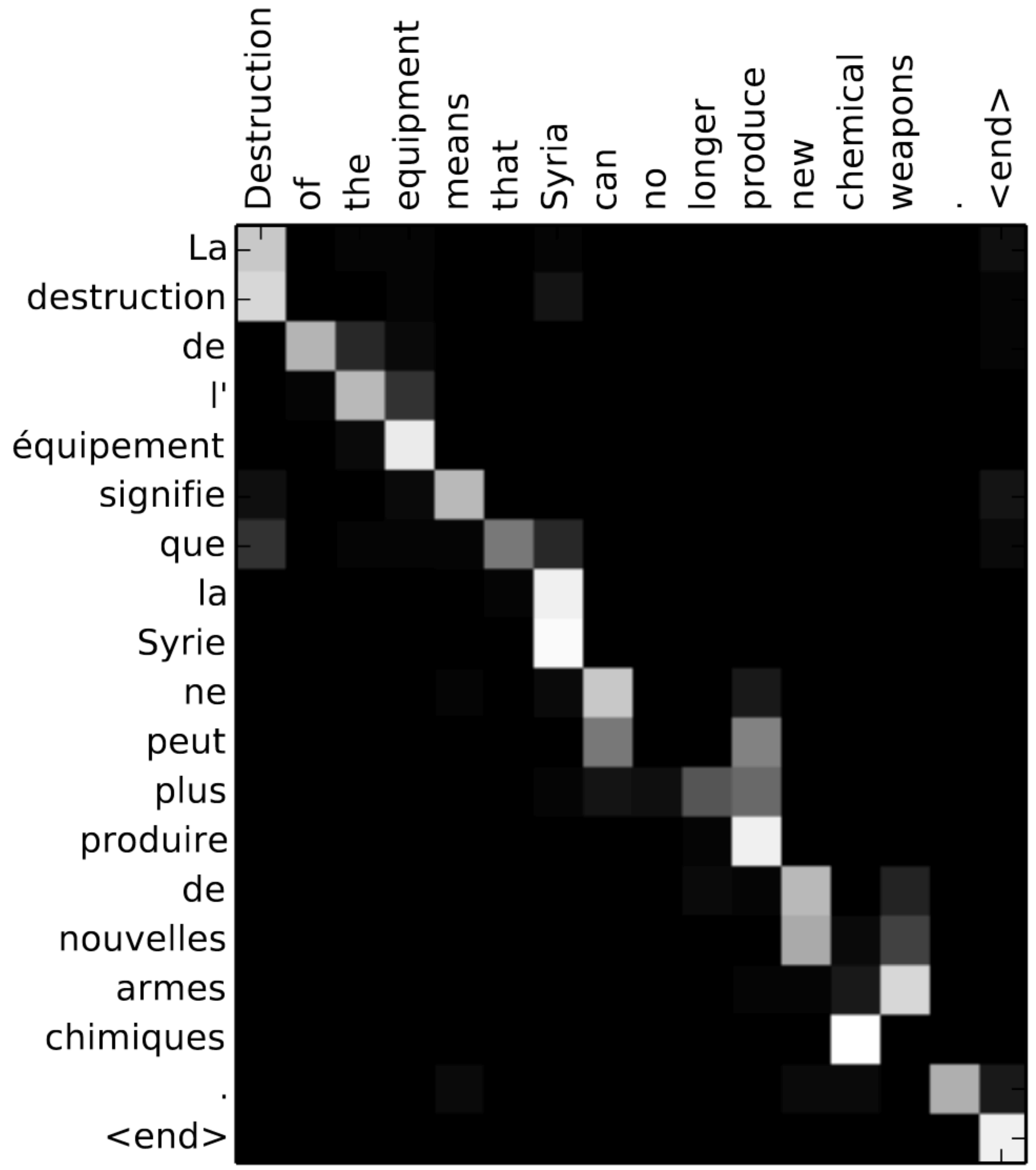
- Incredibly useful (for performance)
  - By “solving” the bottleneck issue
- Aids interpretability (maybe)



Badhanau et al 2014

Why attention?

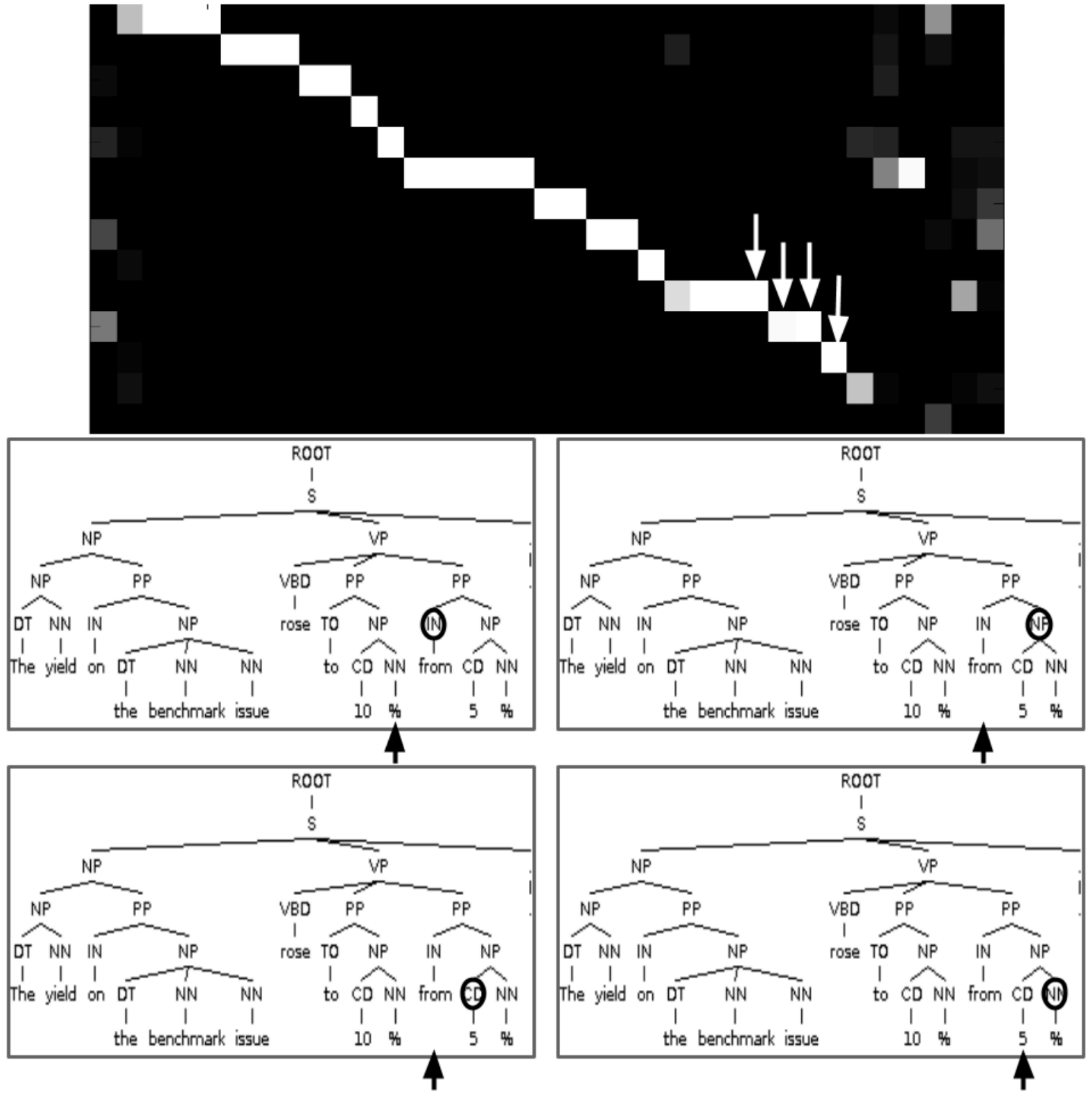
- Incredibly useful (for performance)
  - By “solving” the bottleneck issue
- Aids interpretability (maybe)
- A general technique for combining representations, applications in:
  - NMT, parsing, image/video captioning, …, everything



Badhanau et al 2014

Why attention?

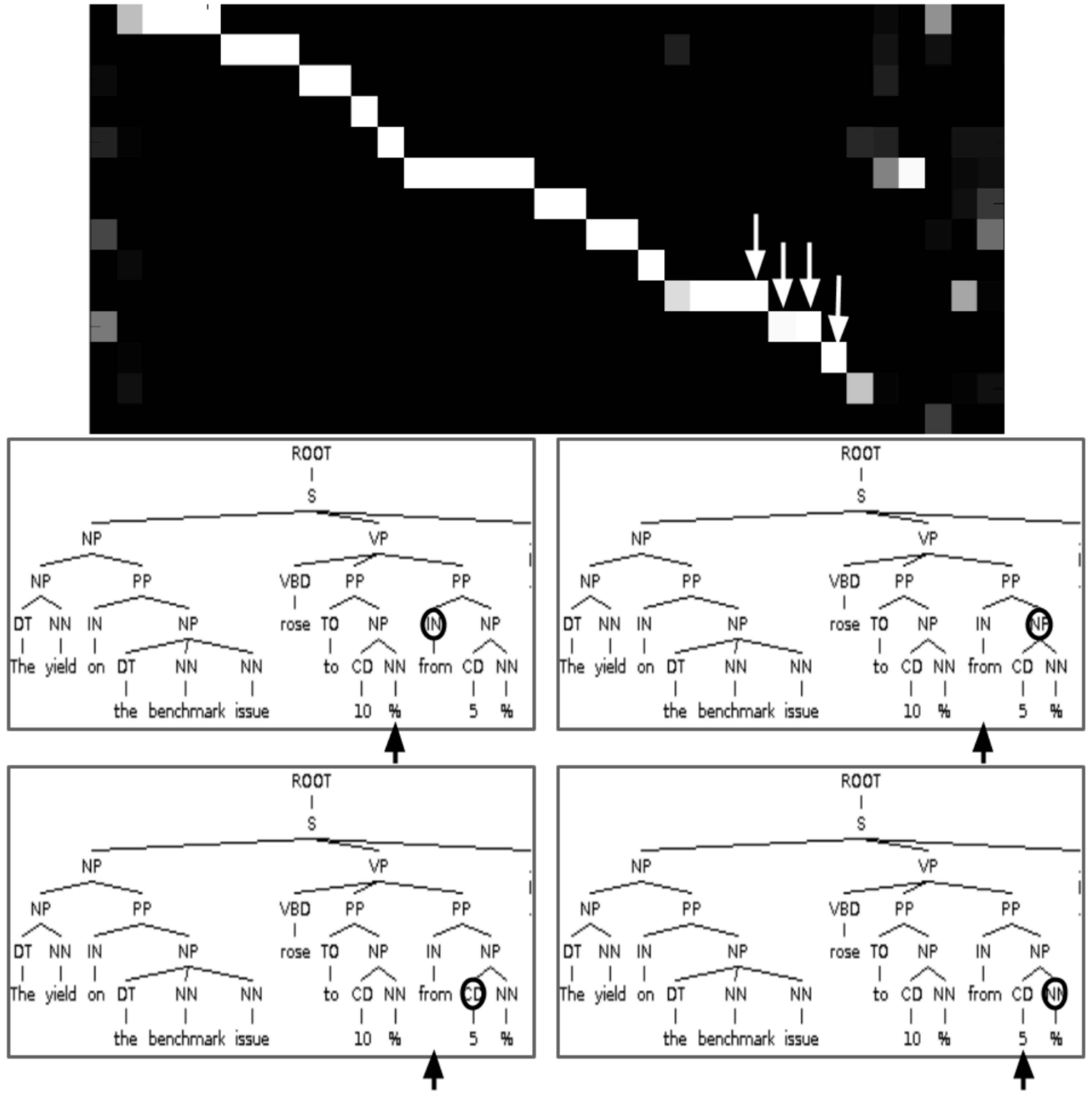
- Incredibly useful (for performance)
  - By “solving” the bottleneck issue
- Aids interpretability (maybe)
- A general technique for combining representations, applications in:
  - NMT, parsing, image/video captioning, …, everything



[Vinyals et al 2015](#)

Why attention?

- Incredibly useful (for performance)
  - By “solving” the bottleneck issue
- Aids interpretability (maybe)
- A general technique for combining representations, applications in:
  - NMT, parsing, image/video captioning, …, everything
- Conceptually, let the model *learn to align* representations
  - “Soft” alignment, just like gates = “soft” masks



[Vinyals et al 2015](#)

Next Time

- Introduction to the *Transformer* architecture
  - Hint:

Next Time

- Introduction to the *Transformer* architecture
  - Hint:

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser*
Google Brain
lukaszkaiser@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly