Machine Translation

Cristina España-Bonet

UdS & DFKI, Saarbrücken, Germany

Artificial Intelligence with Deep Learning

2nd May 2019

Neural Machine Translation reaches historic milestone: human parity for Chinese to English translations

Microsoft Translator March 14, 2018

Achieving Human Parity on Automatic Chinese to English News Translation

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou

Microsoft AI & Research

Abstract

Machine translation has made rapid advances in recent years. Millions of people are using it today in online translation systems and mobile applications in order to communicate across language barriers. The question naturally arises whether such systems can approach or achieve parity with human translations. In this paper, we first address the problem of how to define and accurately measure human parity in translation. We then describe Microsoft's machine translation system and measure the quality of its translations on the widely used WMT 2017 news translation task from Chinese to English. We find that our latest neural machine translation system has reached a new state-of-the-art, and that the translation quality is at human parity when compared to professional human translations. We also find that it significantly exceeds the quality of crowd-sourced non-professional translations.

1 Introduction

Recent years have seen human performance levels reached or surpassed in tasks ranging from games such as Go [32] to classification of images in ImageNet [20] to conversational speech recognition on the Switchboard task [49].

In the area of machine translation, we have seen dramatic improvements in quality with the advent of attentional encoder-decoder neural networks [34, 3, 38]. However, translation quality continues to vary a great deal across language pairs, domains, and genres, more or less in direct

Motivation

Applications

SOCIETAT > **CASTELLERS** CIÈNCIA EDUCACIÓ MEDI AMBIENT TEMPS SANITAT SUCCESSOCE

Tot a punt per al concurs de castells de Tarragona

CASTELLERS

Els Castellers de Viladecans celebren el seu 5è aniversari

ENSURT SENSE CONSEQÜÈNCIES

Una enxaneta se salva miraculosament d'una caiguda a la plaça de Sant Jaume

CANVI DE JUNTA

Minyons demana girar full i treballar en cooperació amb la Coordinadora de Colles Castelleres

La colla egarenca advoca per iniciar una "nova etapa" arcada per la "lleialtat recíproca i...

DIADA CASTELLERA

Minyons, Capgrossos i Borinots han de renunciar les seves màximes fites en

SOCIEDAD > **CASTELLERS** CIENCIA EDUCACIÓN MEDIO AMBIENTE TIEMPO SANIDAD SUCE

Todo a punto para el concurso de castells de Tarragona

REIVINDICACIÓN

Los Castellers de Barcelona descargan una torre solo de mujeres

CASTELLERS

Los Castellers de Viladecans celebran su 5º aniversario

SUSTO EN LA PLAZA DE SANT JAUME

Una 'enxaneta' se salva tras una caída en Barcelona

CAMBIO DE JUNTA

Minyons pide pasar página y trabajar en cooperación con la Coordinadora de Colles Castelleres

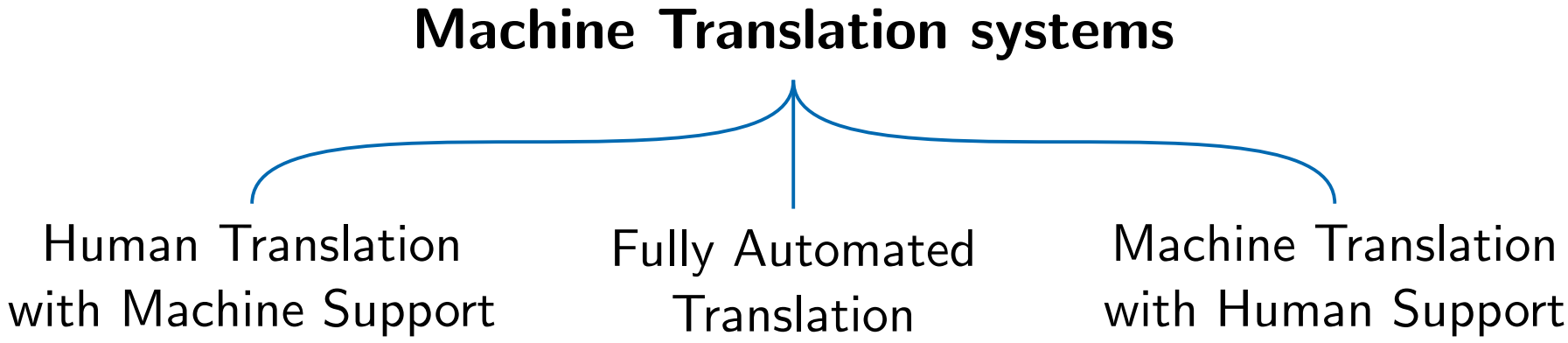
La colla egarense advoca por iniciar una "nueva etapa" arcada por la "lealtad recíproca y...

Outline

Introduction

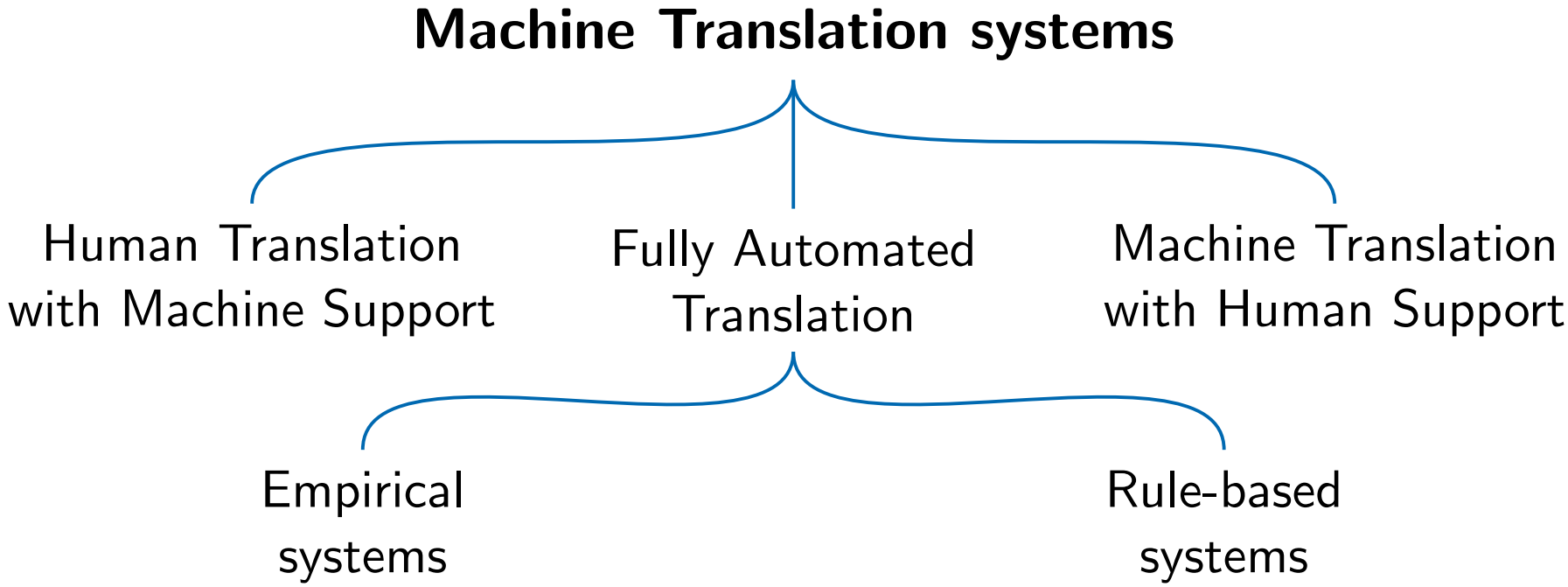
Introduction

Naïve Taxonomy



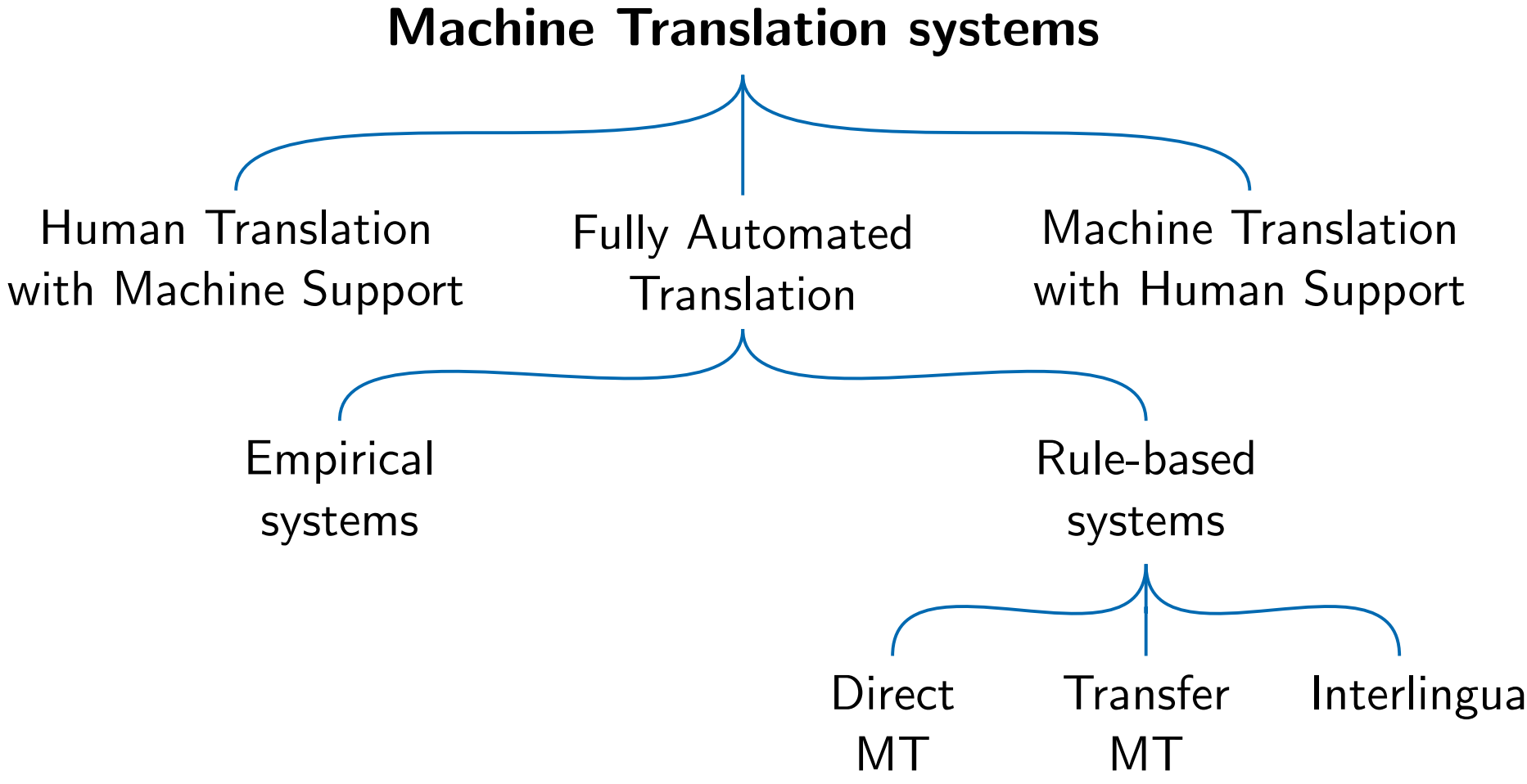
Introduction

Naïve Taxonomy



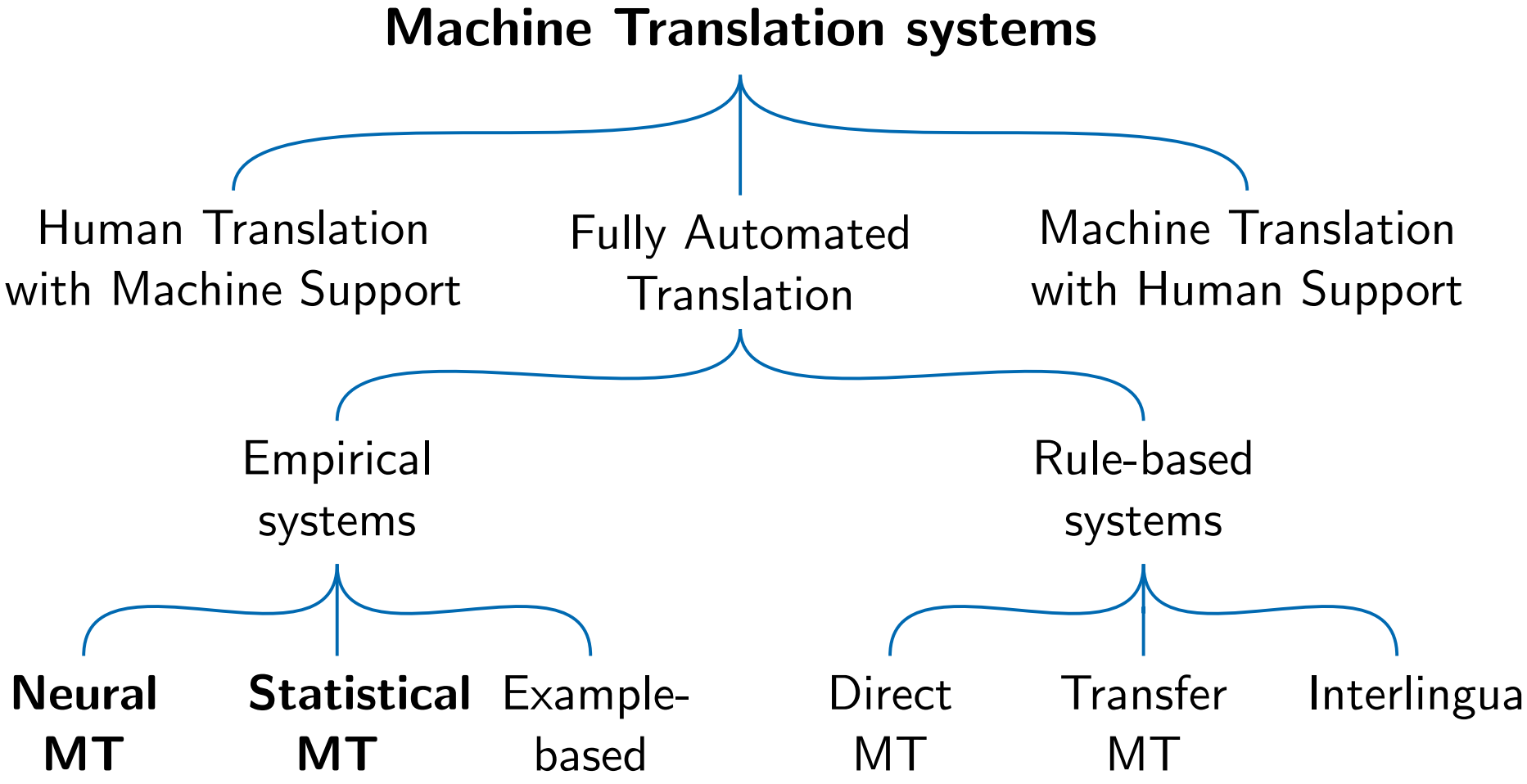
Introduction

Naïve Taxonomy



Introduction

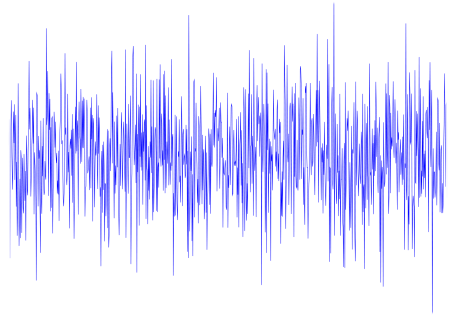
Naïve Taxonomy



Statistical Machine Translation

The Noisy Channel

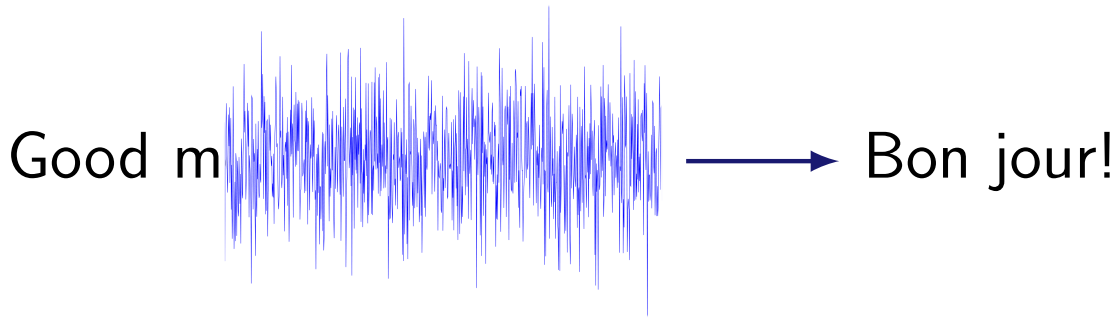
The Noisy Channel as a statistical approach to translation:

Good morning! → 

Statistical Machine Translation

The Noisy Channel

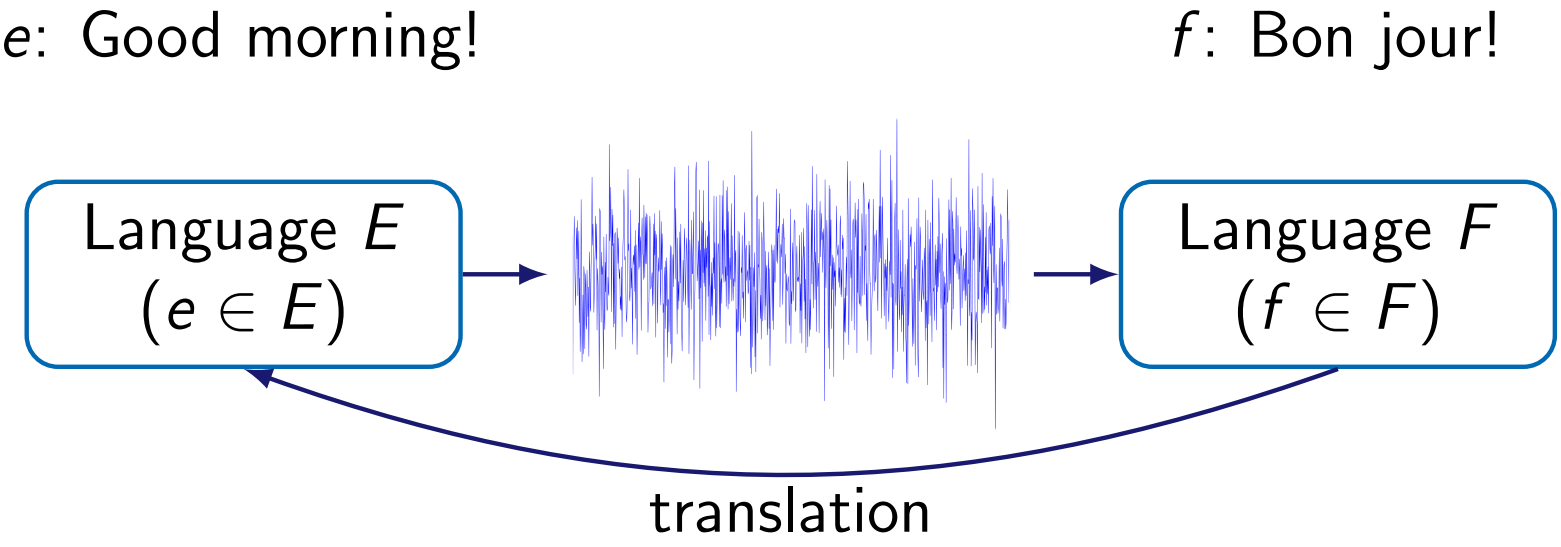
The Noisy Channel as a statistical approach to translation:



Statistical Machine Translation

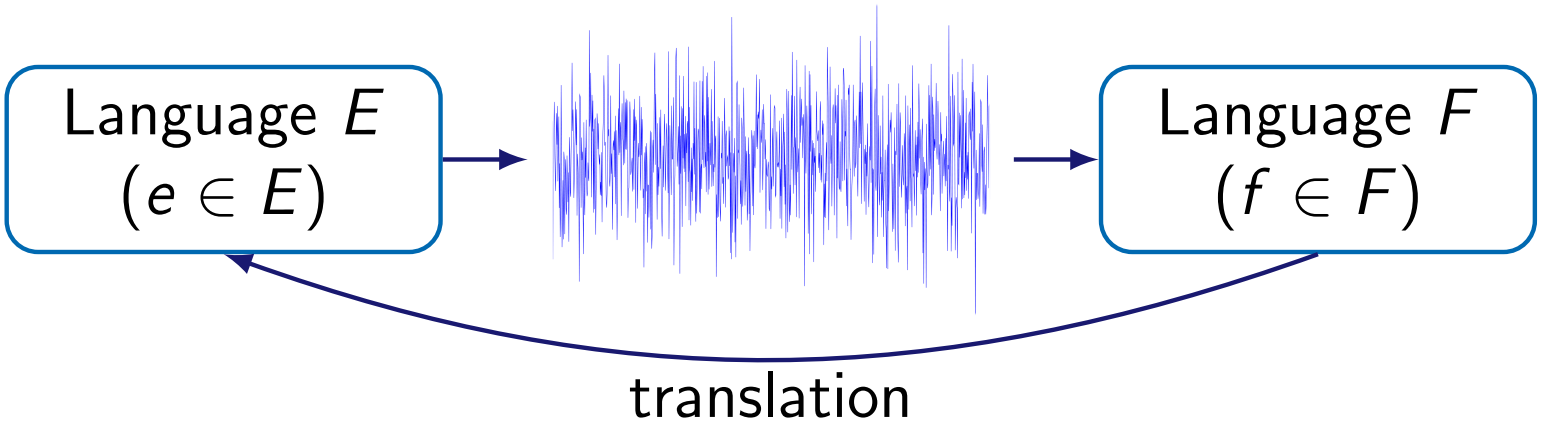
The Noisy Channel

The Noisy Channel as a statistical approach to translation:



Statistical Machine Translation

The Noisy Channel

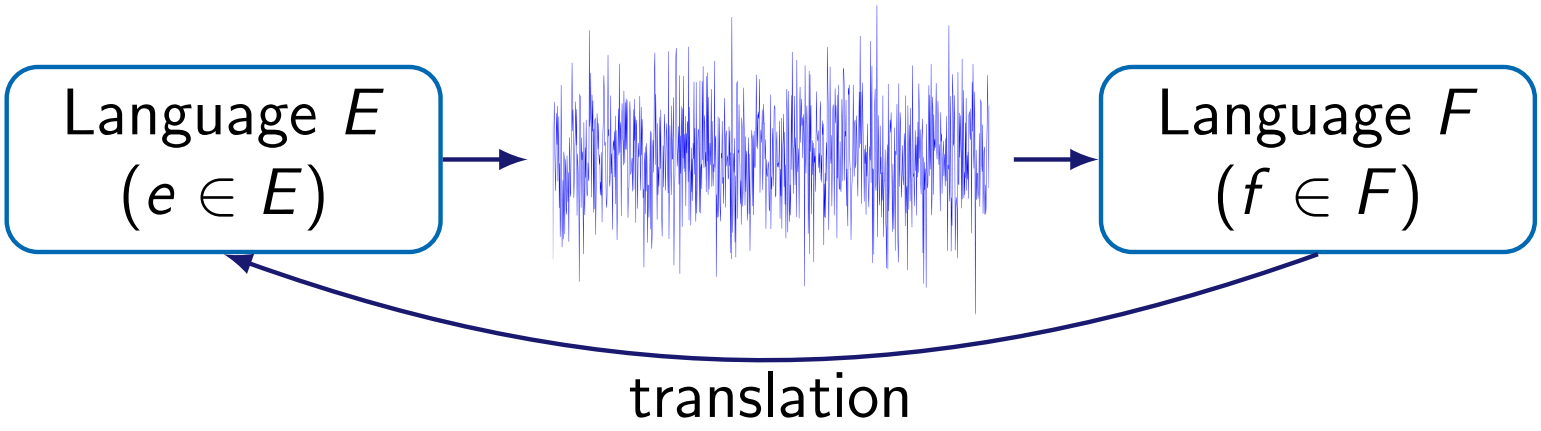


Mathematically:

$$P(e|f)$$

Statistical Machine Translation

The Noisy Channel



Mathematically:

$$P(e|f) = \frac{P(e)\,P(f|e)}{P(f)}$$

$$T(f) = \hat{e} = \text{argmax}_e\ P(e|f) = \text{argmax}_e\ P(e)\,P(f|e)$$

Statistical Machine Translation

Basic equation

$$T(f) = \hat{e} = \operatorname{argmax}_e \mathbf{P(e)}\, P(f|e)$$

Language Model

- Takes care of fluency in the target language
- Data: corpora in the target language

Statistical Machine Translation

Basic equation

$$T(f) = \hat{e} = \text{argmax}_e \; P(e) \; \mathbf{P(f|e)}$$

Language Model

- Takes care of fluency in the target language
- Data: corpora in the target language

Translation Model

- Lexical correspondence between languages
- Data: aligned corpora in source and target languages

Statistical Machine Translation

Basic equation

$$T(f) = \hat{e} = \textbf{argmax}_e \; P(e)\, P(f|e)$$

Language Model

- Takes care of fluency in the target language
- Data: corpora in the target language

Translation Model

- Lexical correspondence between languages
- Data: aligned corpora in source and target languages

argmax

- Search done by the *decoder*

Statistical Machine Translation

Remember from the LM lecture...

$$T(f) = \hat{e} = \text{argmax}_e \, \mathbf{P(e)} \, P(f|e)$$

The language model assigns a probability $P(e)$ to a sequence of words $e \Rightarrow \{w_1, \ldots, w_m\}$.

$$P(w_1, \ldots, w_m) = \prod_{i=1}^{m} P(w_i | w_{i-(n-1)}, \ldots, w_{i-1})$$

- Frequencies in a corpus
- *Softmaxed* output of a neural network

Statistical Machine Translation

The translation model $P(f|e)$

$$T(f) = \hat{e} = \text{argmax}_e \; P(e)\,\mathbf{P(f|e)}$$

Estimation of the lexical correspondence between languages.

Statistical Machine Translation

The translation model $P(f|e)$

$$T(f) = \hat{e} = \text{argmax}_e \; P(e)\,\mathbf{P(f|e)}$$

Estimation of the lexical correspondence between languages.

How can be $P(f|e)$ characterised?

```
NULL   Quan   tornes   a   casa   ?

When are you coming back  home   ?
```

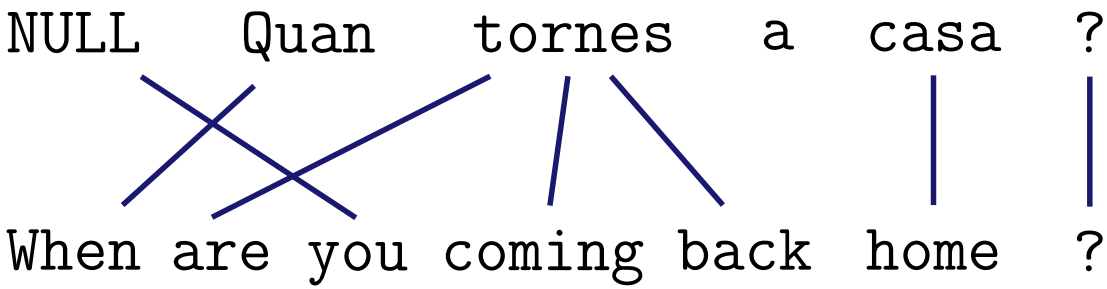
Statistical Machine Translation

The translation model $P(f|e)$

$$T(f) = \hat{e} = \text{argmax}_e \; P(e)\,\mathbf{P(f|e)}$$

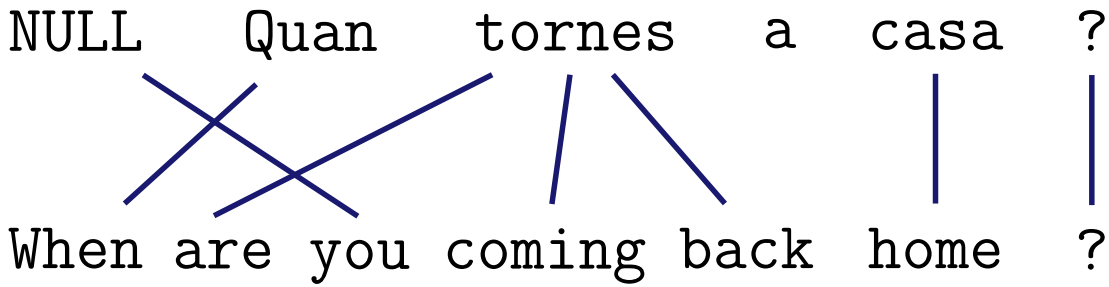
Estimation of the lexical correspondence between languages.

How can be $P(f|e)$ characterised?



Statistical Machine Translation

The translation model $P(f|e)$



One should at least model for *each word* in the source language:

- Its translation,
- the number of necessary words in the target language,
- the position of the translation within the sentence,
- and, besides, the number of words that need to be generated from scratch.

Statistical Machine Translation

The translation model $P(f|e)$

Word-based models: the IBM models

They characterise $P(f|e)$ with 4 parameters: t, n, d and p_1.

- Lexical probability t
  $t(\texttt{Quan}|\texttt{When})$: the prob. that `Quan` translates into `When`.

- Fertility n
  $n(3|\texttt{tornes})$: the prob. that `tornes` generates 3 words.

Word-based models: the IBM models

They characterise $P(f|e)$ with 4 parameters: t, n, d and p_1.

- Distortion d
  $d(j|i,m,n)$: the prob. that the word in the j position generates a word in the i position. m and n are the length of the source and target sentences.
- Probability p_1
  $p(\texttt{you}|\texttt{NULL})$: the prob. that the spurious word `you` is generated (from `NULL`).

Statistical Machine Translation

IBM models

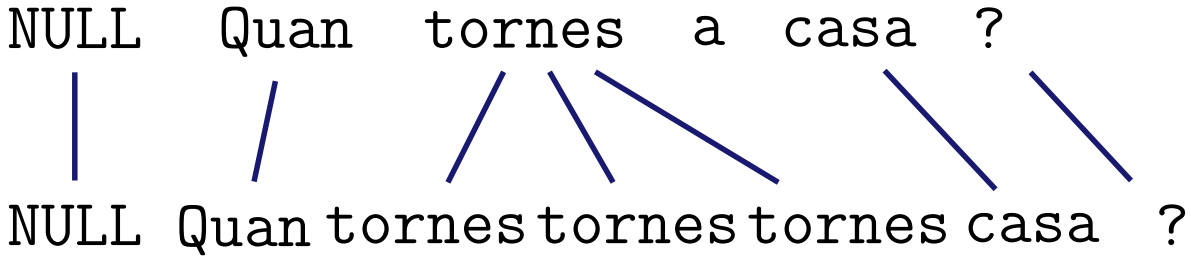
Back to the example:

```
NULL  Quan  tornes  a  casa  ?
```

Statistical Machine Translation

IBM models

Back to the example:

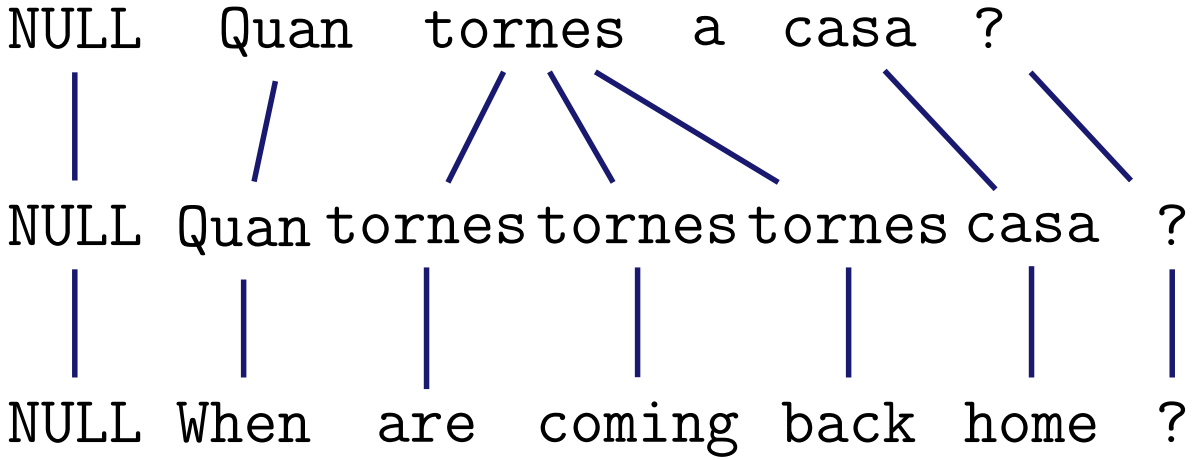


Fertility

Statistical Machine Translation

IBM models

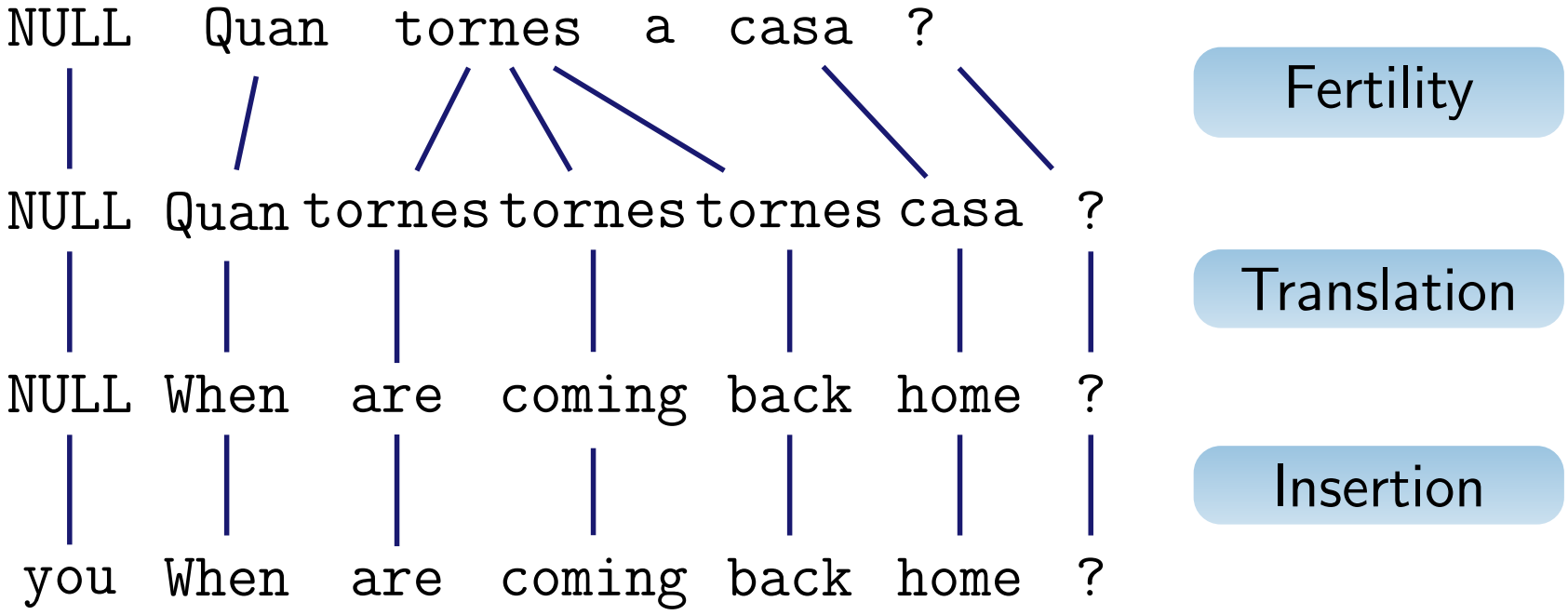
Back to the example:



Statistical Machine Translation

IBM models

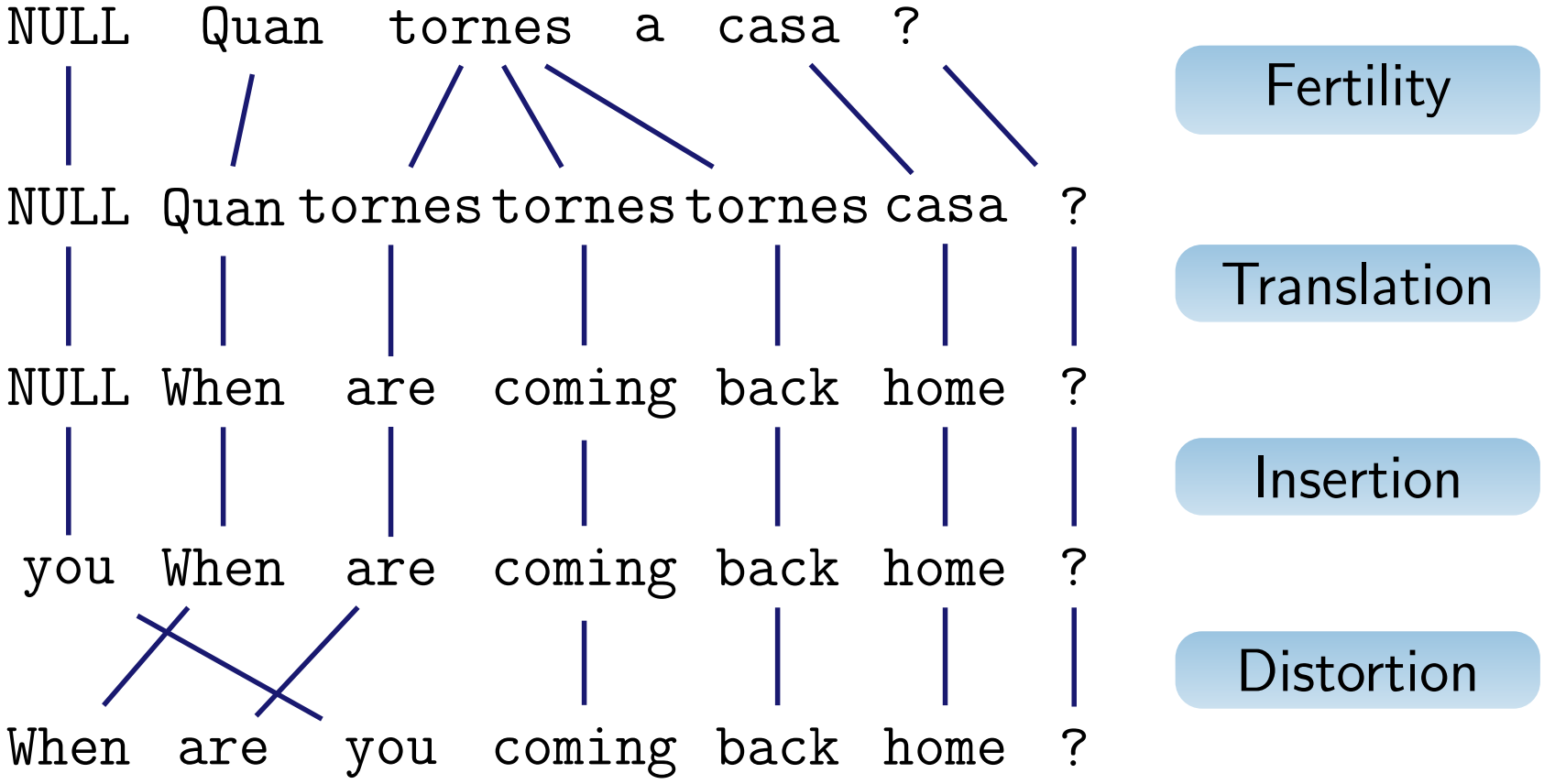
Back to the example:



Statistical Machine Translation

IBM models

Back to the example:



Statistical Machine Translation

Decoding

$$T(f) = \hat{e} = \textbf{argmax}_{\mathbf{e}}\ P(e)\,P(f|e)$$

Responsible for the search in the space of possible translations.

Given a model (LM+TM+...), the decoder constructs the possible translations and looks for the most probable one.

Statistical Machine Translation

Decoding

$$T(f) = \hat{e} = \mathbf{argmax_e}\ P(e)\,P(f|e)$$

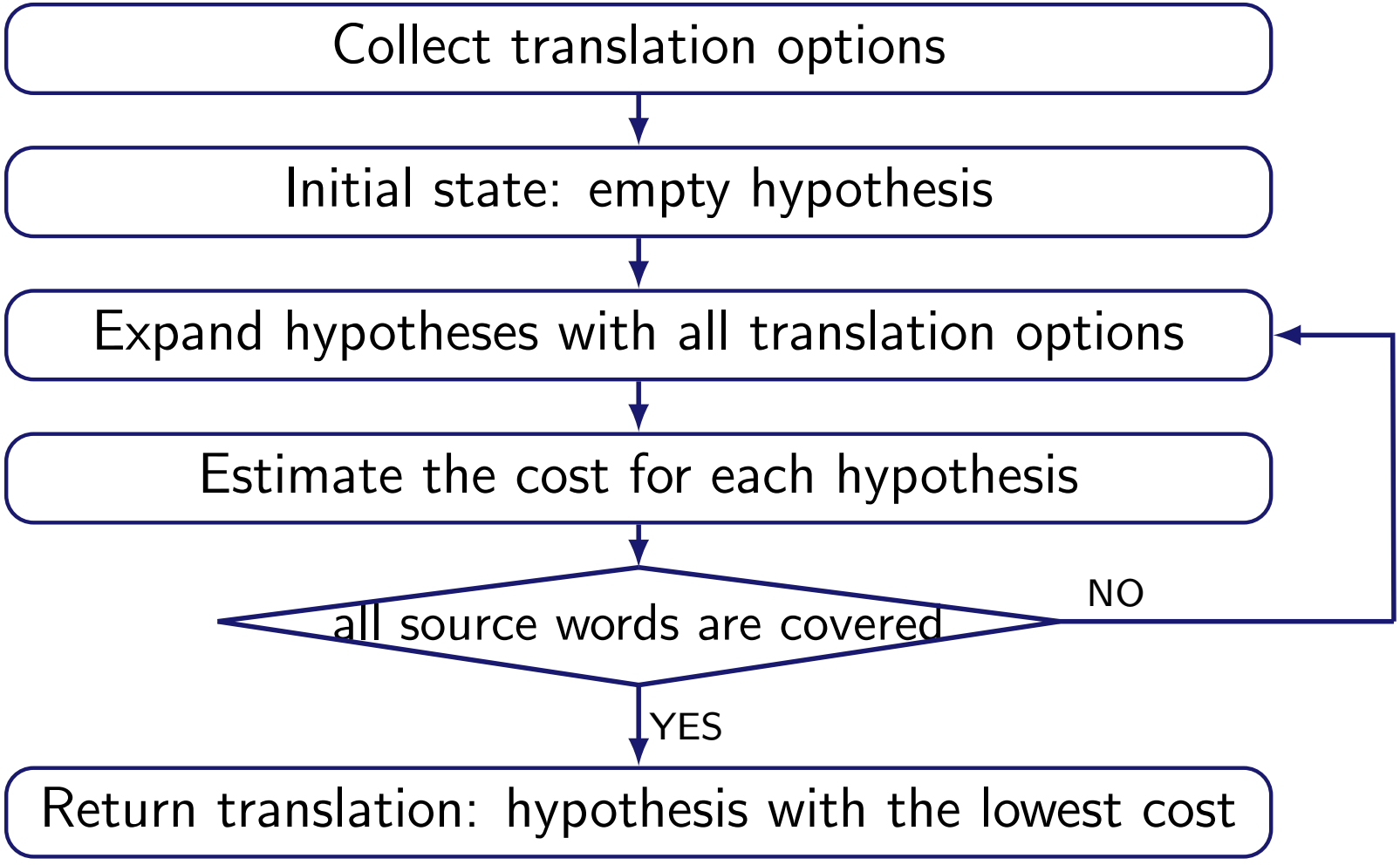
Responsible for the search in the space of possible translations.

Given a model (LM+TM+...), the decoder constructs the possible translations and looks for the most probable one.

In our context, one can find:

- Greedy decoders. Initial hypothesis (word by word translation) refined iteratively using hill-climbing heuristics.
- Beam search decoders.

Core algorithm



Statistical Machine Translation

State-of-the-art

- Log-linear models
- Phrase-based extension

$$\hat{e} = \text{argmax}_e \log P(e|f) = \text{argmax}_e \sum \lambda_m h_m(f, e)$$

Neural Machine Translation

Neural Machine Translation

Basic Model

The Encoder–Decoder Model (with attention)

1. encodes a sequence of word vectors into a fixed-sized context vector
2. decodes the fixed-sized vector back into a variable-length sequence

Neural Machine Translation

Basic Model

The Encoder–Decoder Model (with attention)

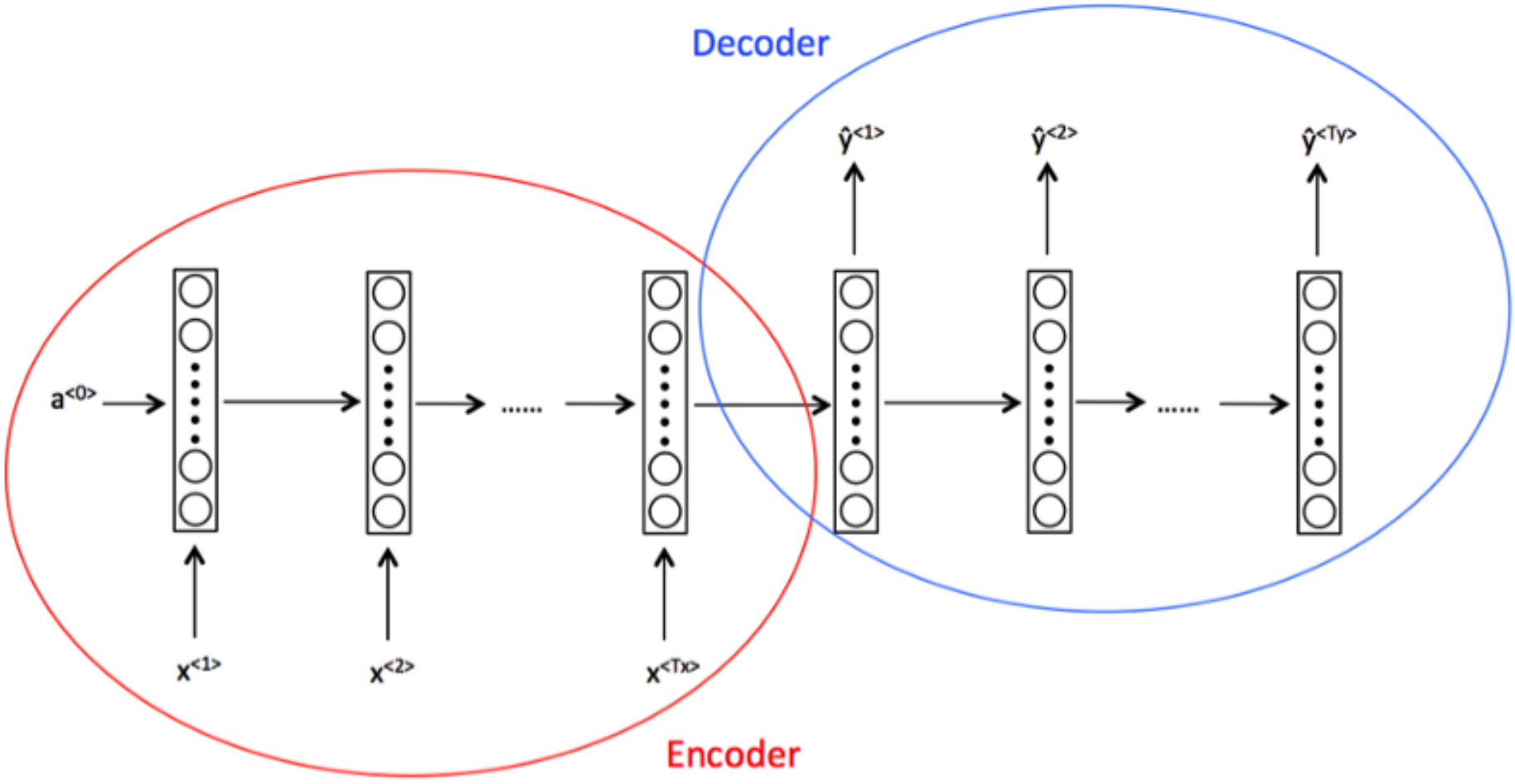
1. encodes a sequence of word vectors into a fixed-sized context vector
2. decodes the fixed-sized vector back into a variable-length sequence

Several NLP tasks use nowadays enc–dec architectures:

- Machine translation, but also...
- text summarisation, question answering, chatbots, speech recognition...

Neural Machine Translation

Remember from the Sequence Modelling lecture...



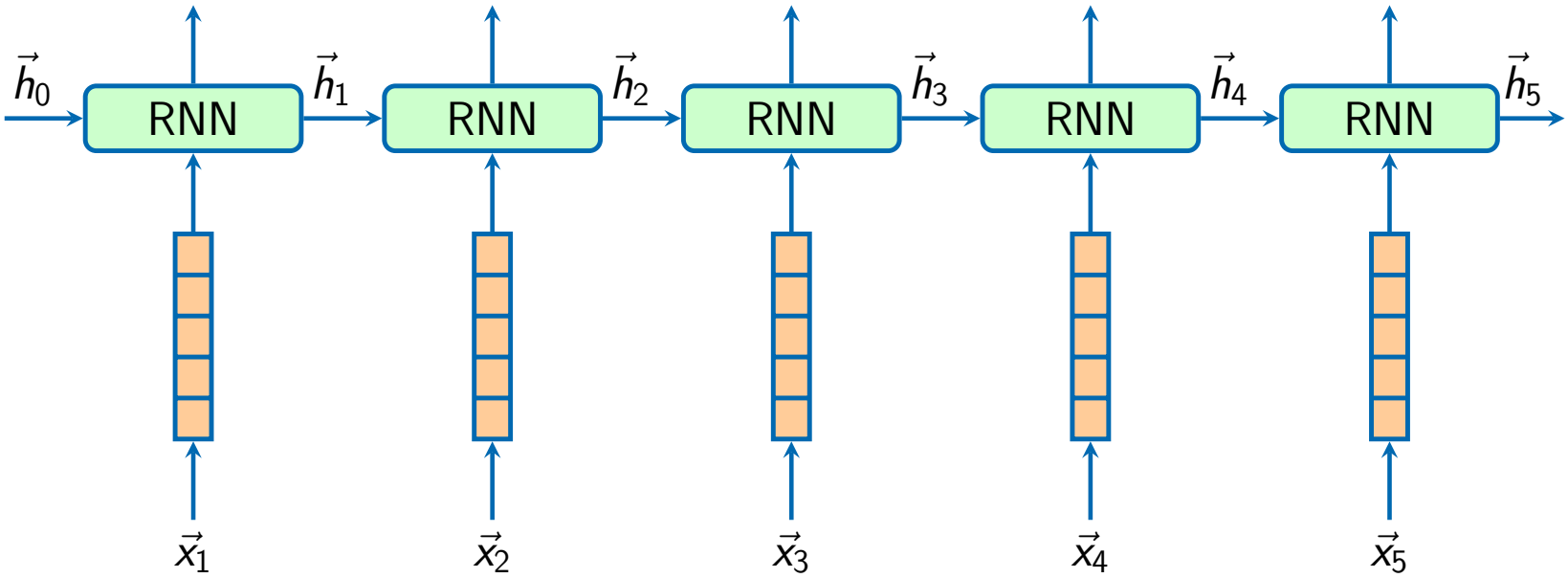
https://medium.com/machine-learning-bites/

/deeplearning-series-sequence-models-7855babeb586

Vanilla RNN Architecture

Encoder

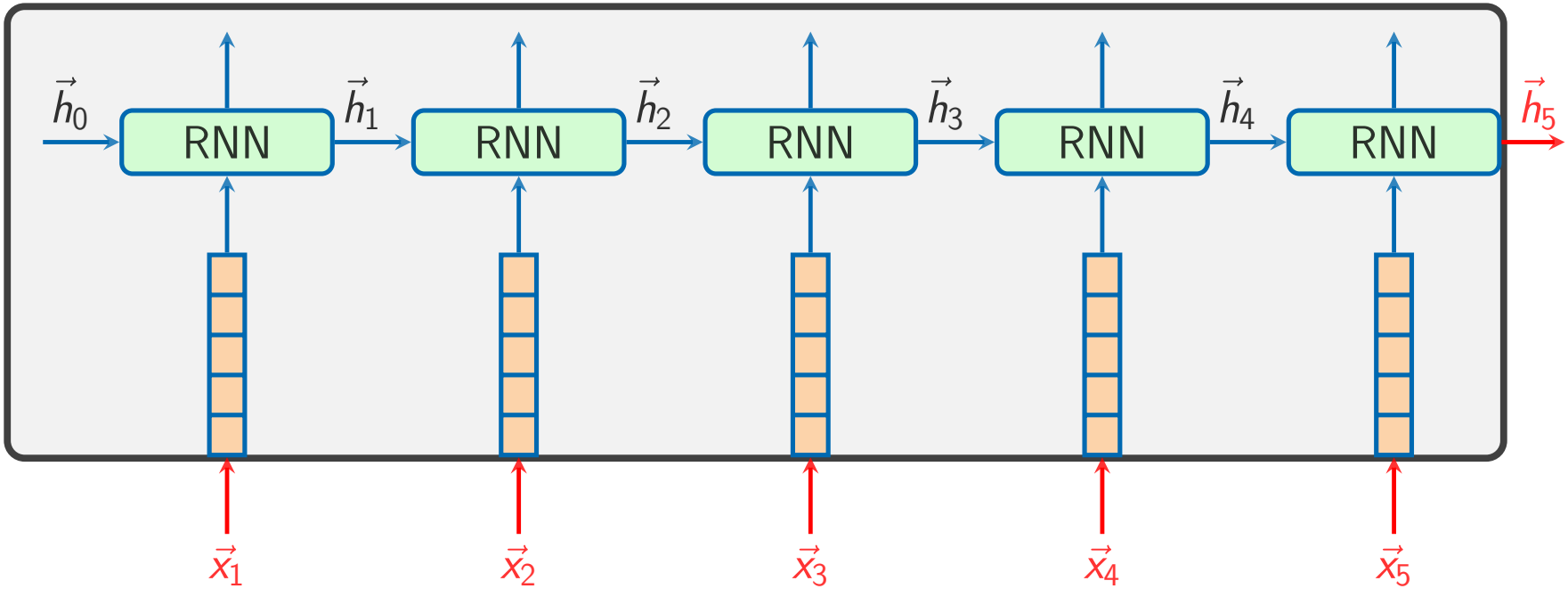
- Sequence encoded with RNNs
- Each word is a *time step*



Vanilla RNN Architecture

Encoder

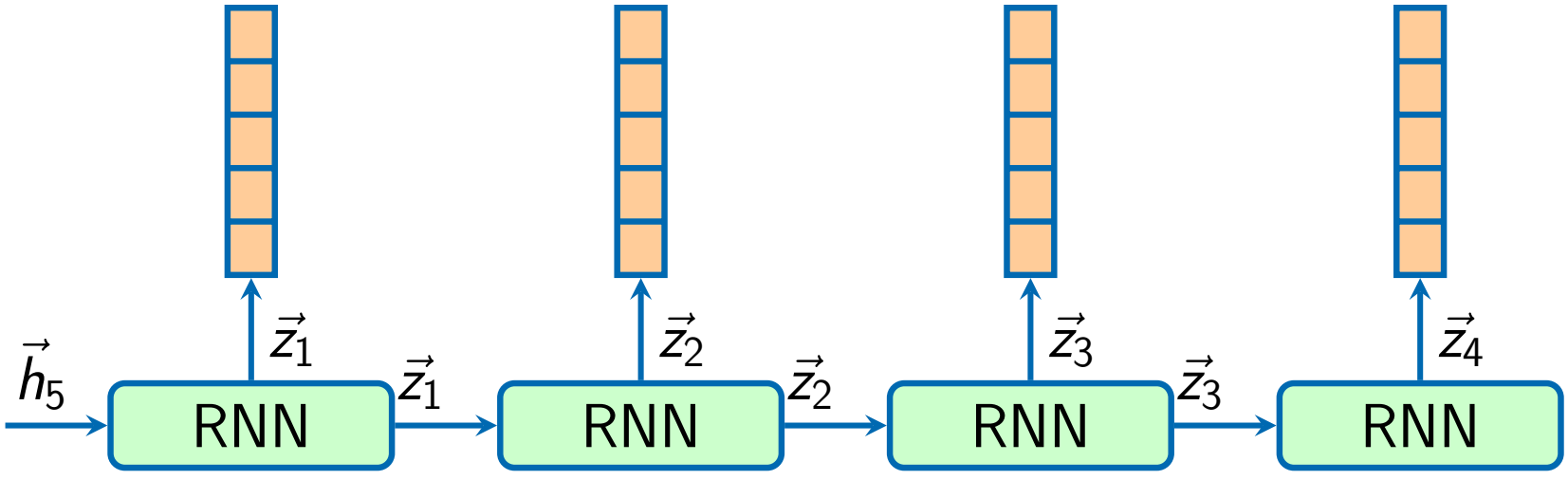
- Sequence encoded with RNNs
- Each word is a *time step*



Vanilla RNN Architecture

Decoder

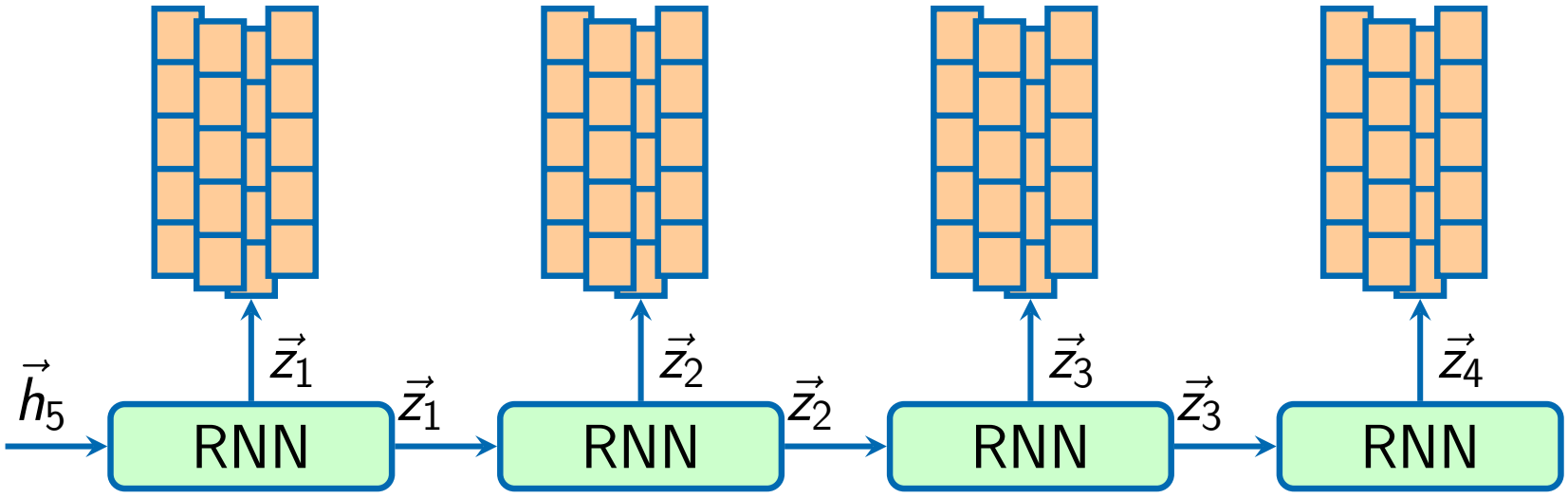
- *Inverted* encoder with special characteristics



Vanilla RNN Architecture

Decoder

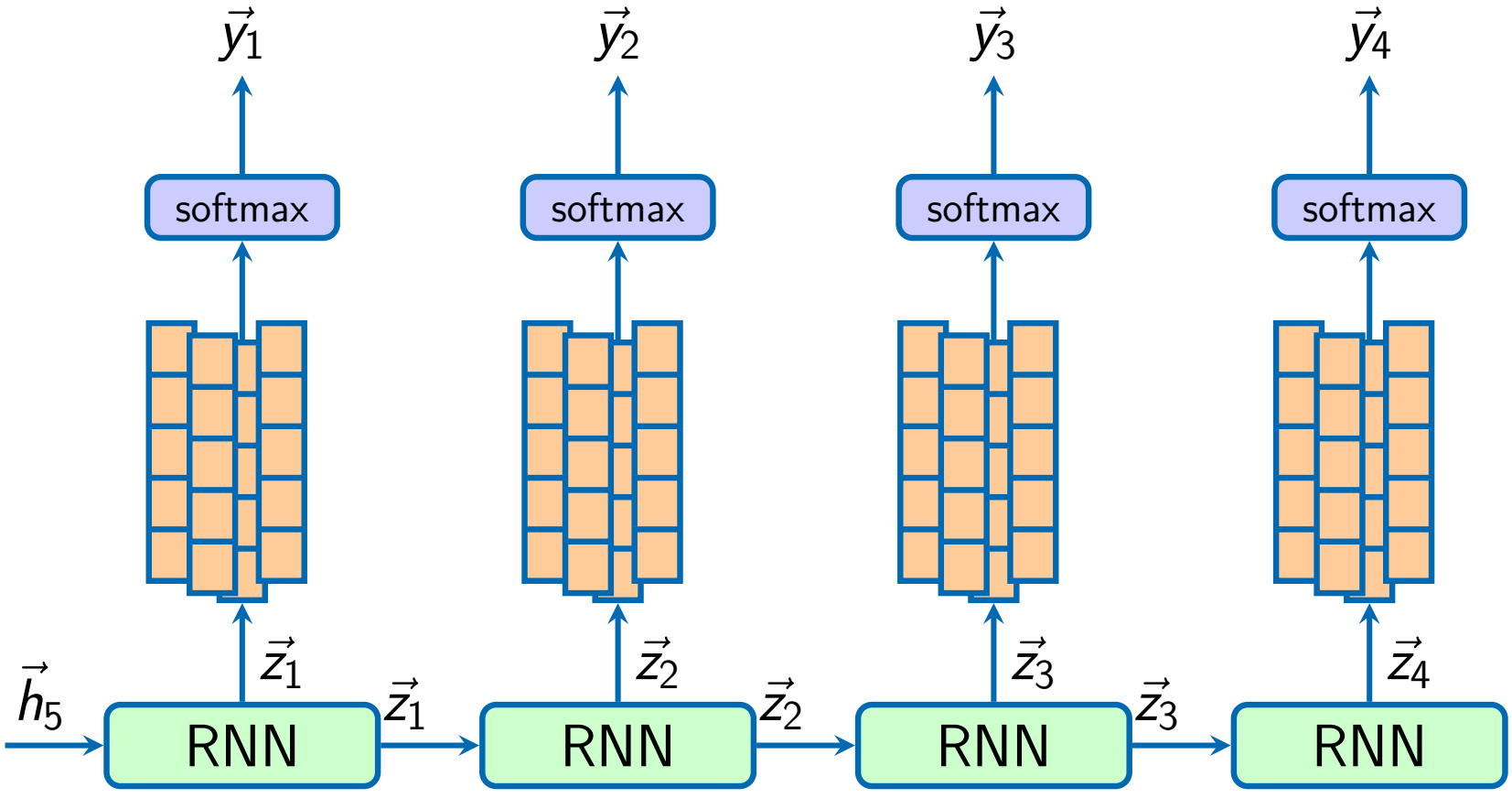
- *Inverted* encoder with special characteristics



Vanilla RNN Architecture

Decoder

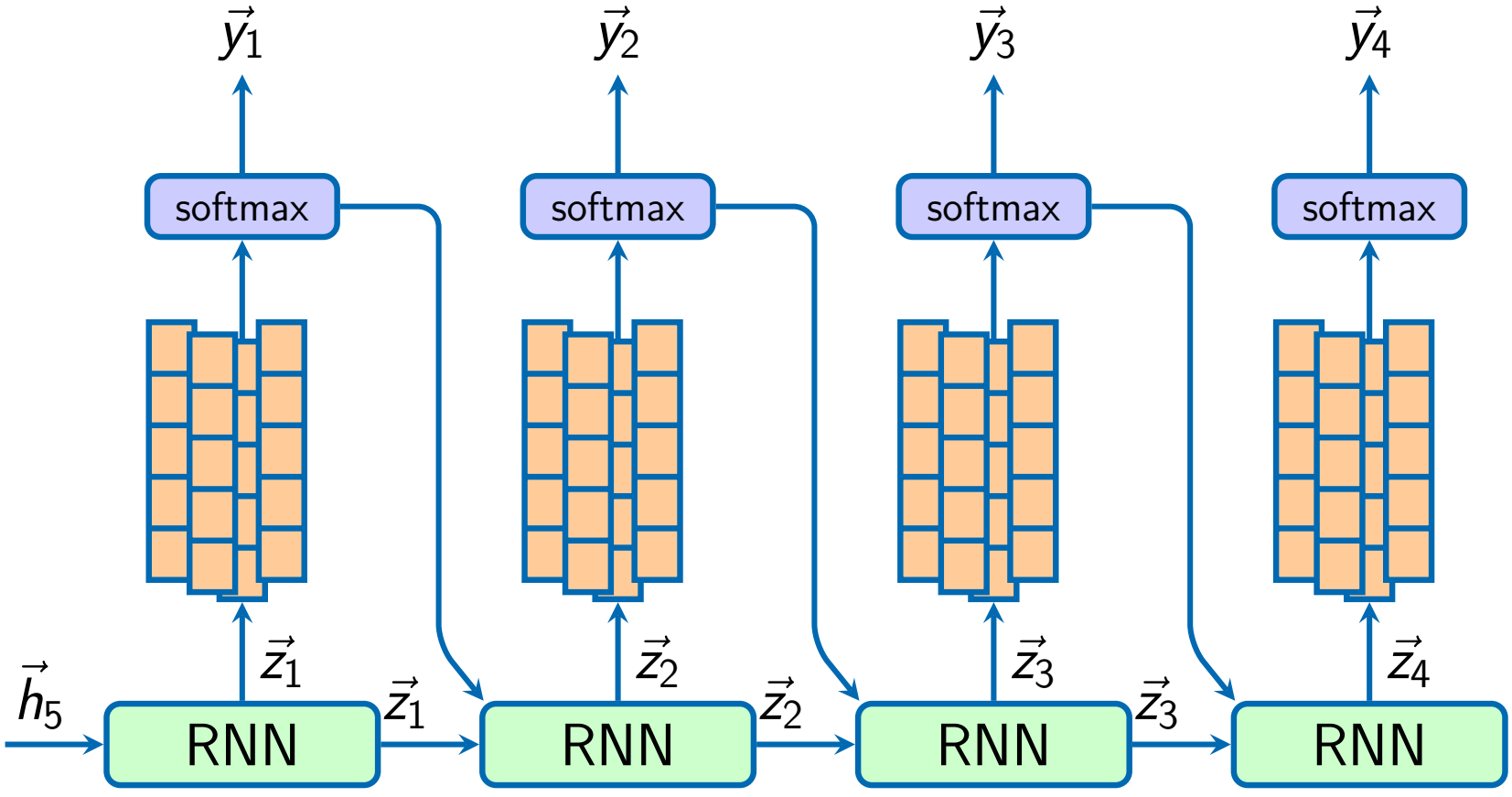
- *Inverted* encoder with special characteristics



Vanilla RNN Architecture

Decoder

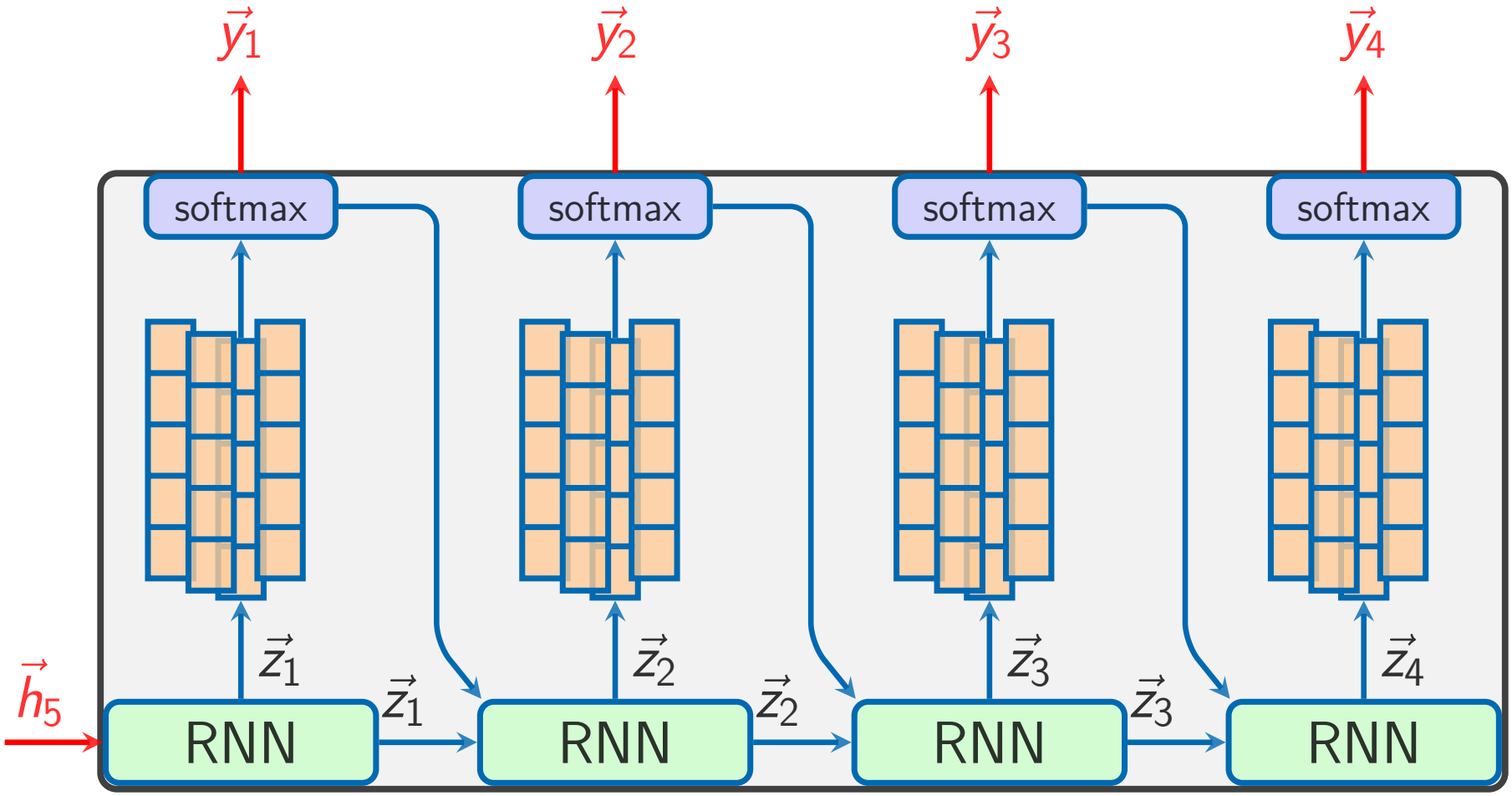
- *Inverted* encoder with special characteristics



Vanilla RNN Architecture

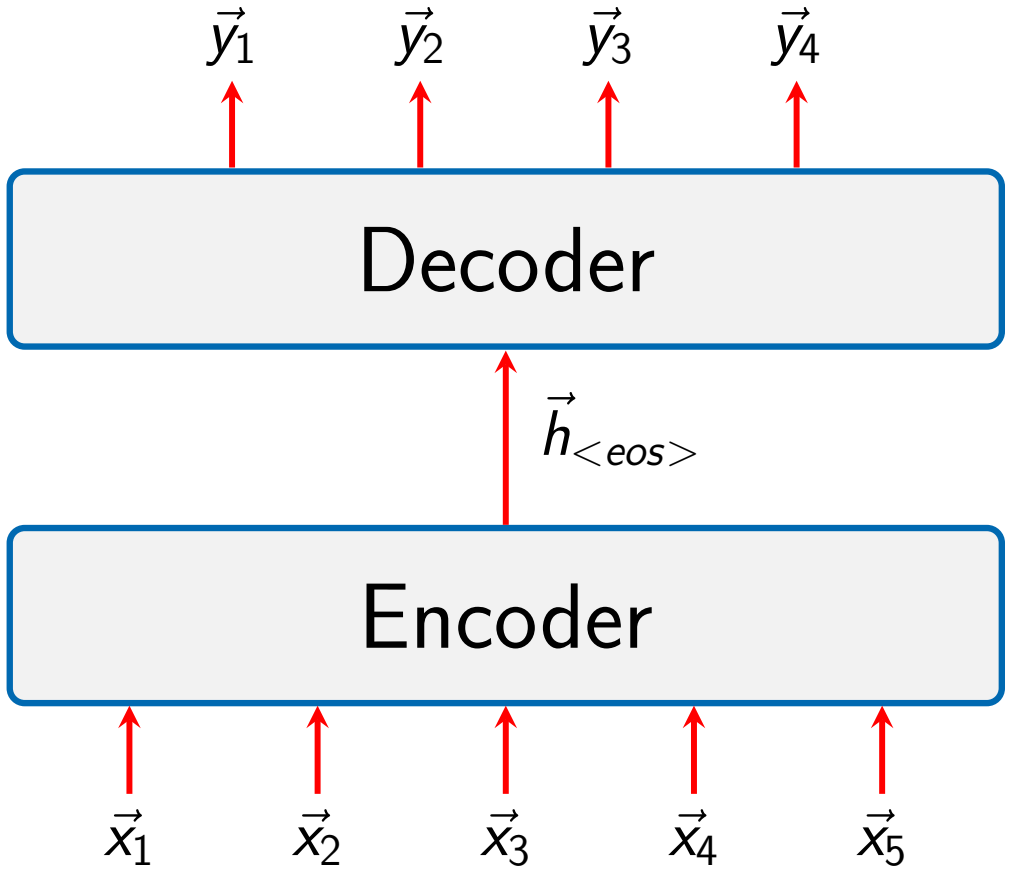
Decoder

- *Inverted* encoder with special characteristics



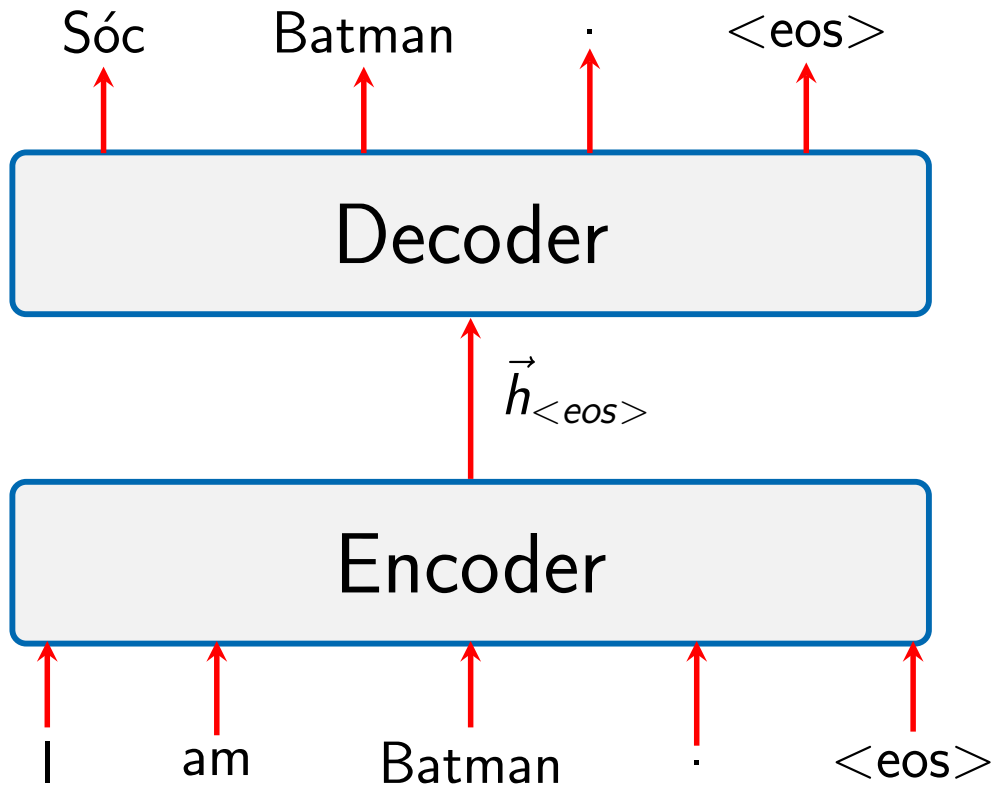
Vanilla RNN Architecture

Encoder–Decoder



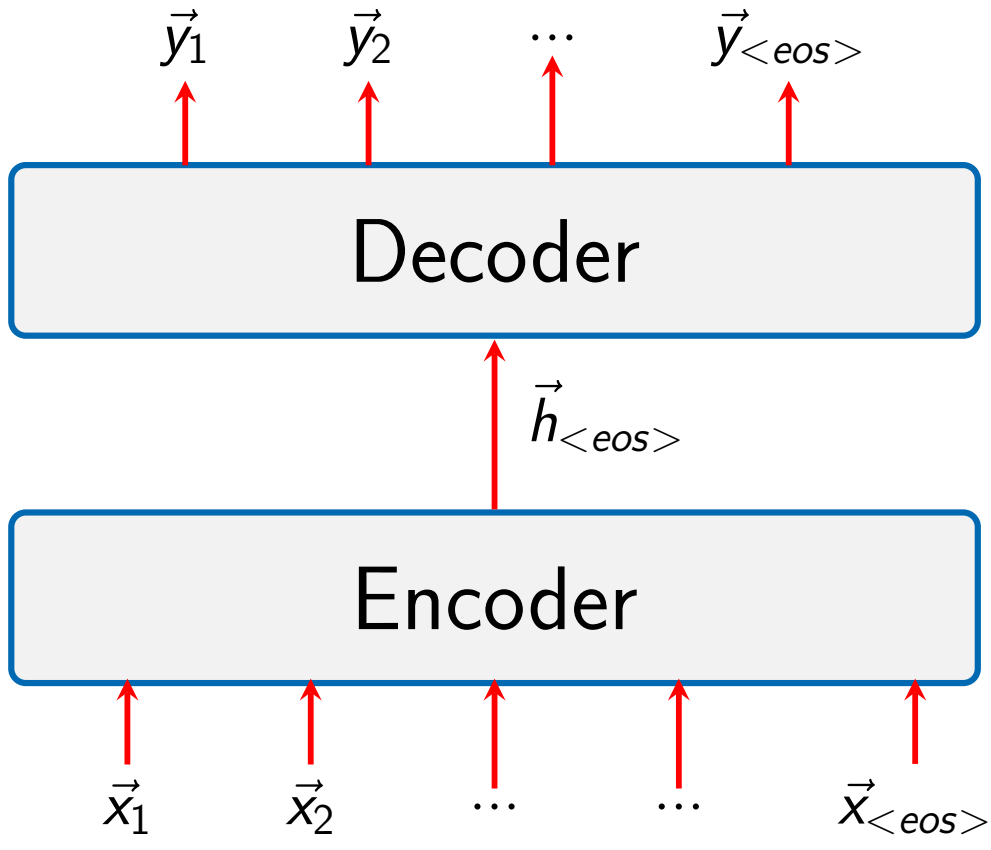
Vanilla RNN Architecture

Encoder–Decoder



Vanilla RNN Architecture

Encoder–Decoder

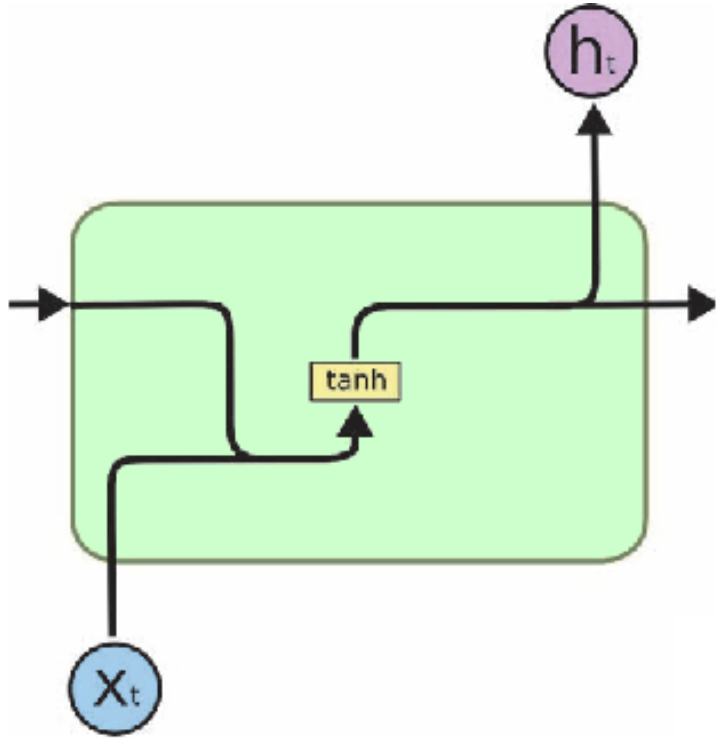


Vanilla RNN Architecture

RNN cells

- RNN cells can be any kind

tanh

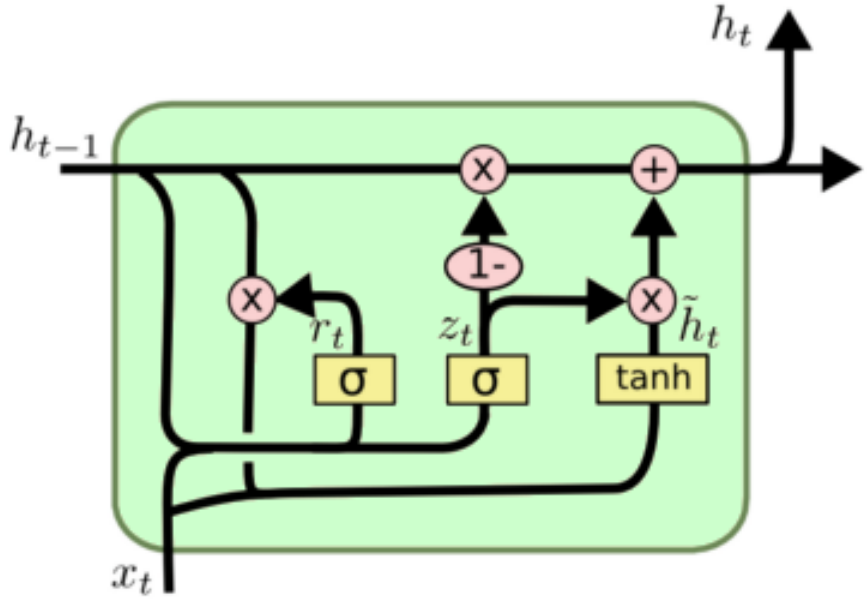


Vanilla RNN Architecture

RNN cells

- RNN cells can be any kind

  GRU, Gated Recurrent Unit

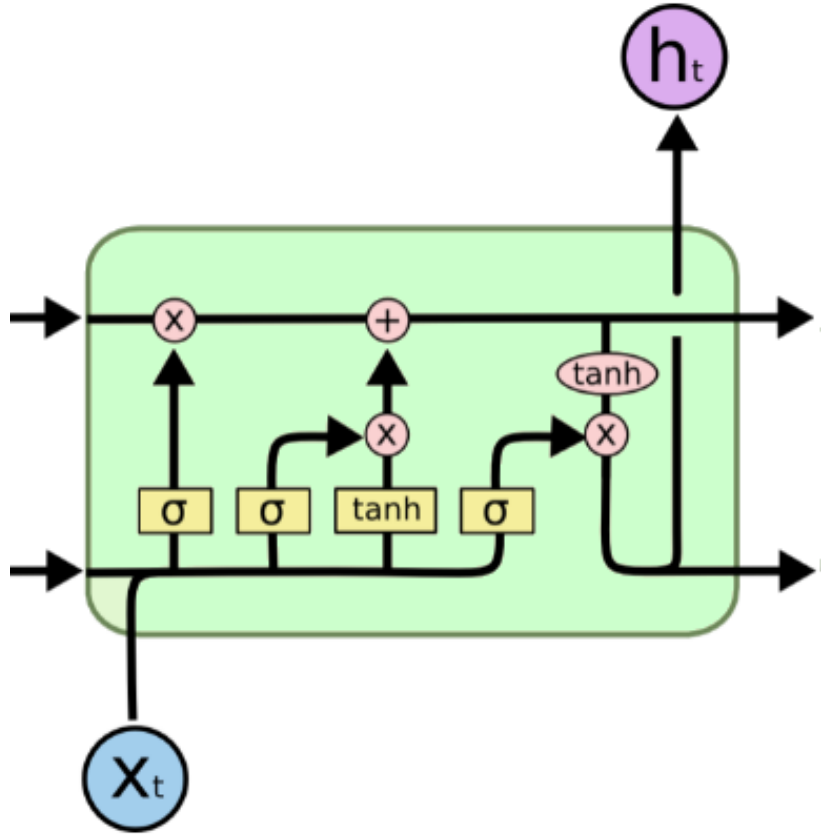


Vanilla RNN Architecture

RNN cells

- RNN cells can be any kind

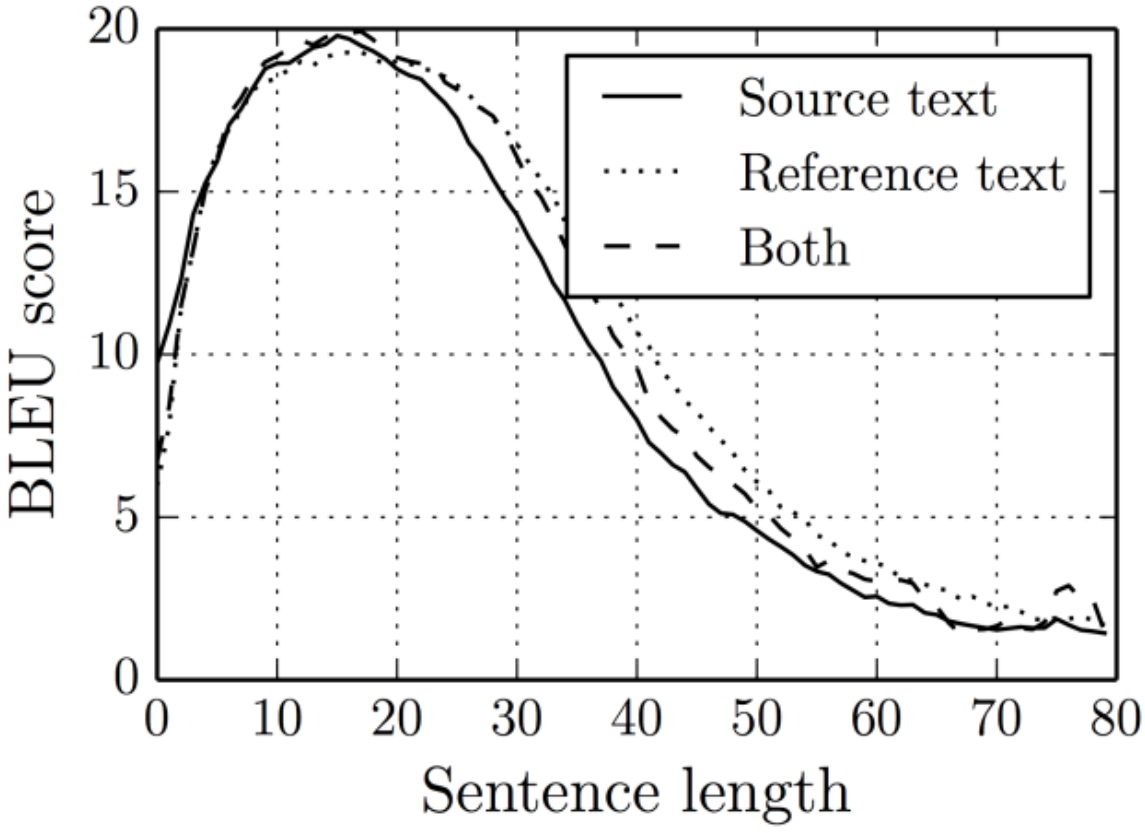
LSTM, Long Short Term Memory Networks



Vanilla RNN Architecture

Why Vanilla?

- Performance drops with long sentences

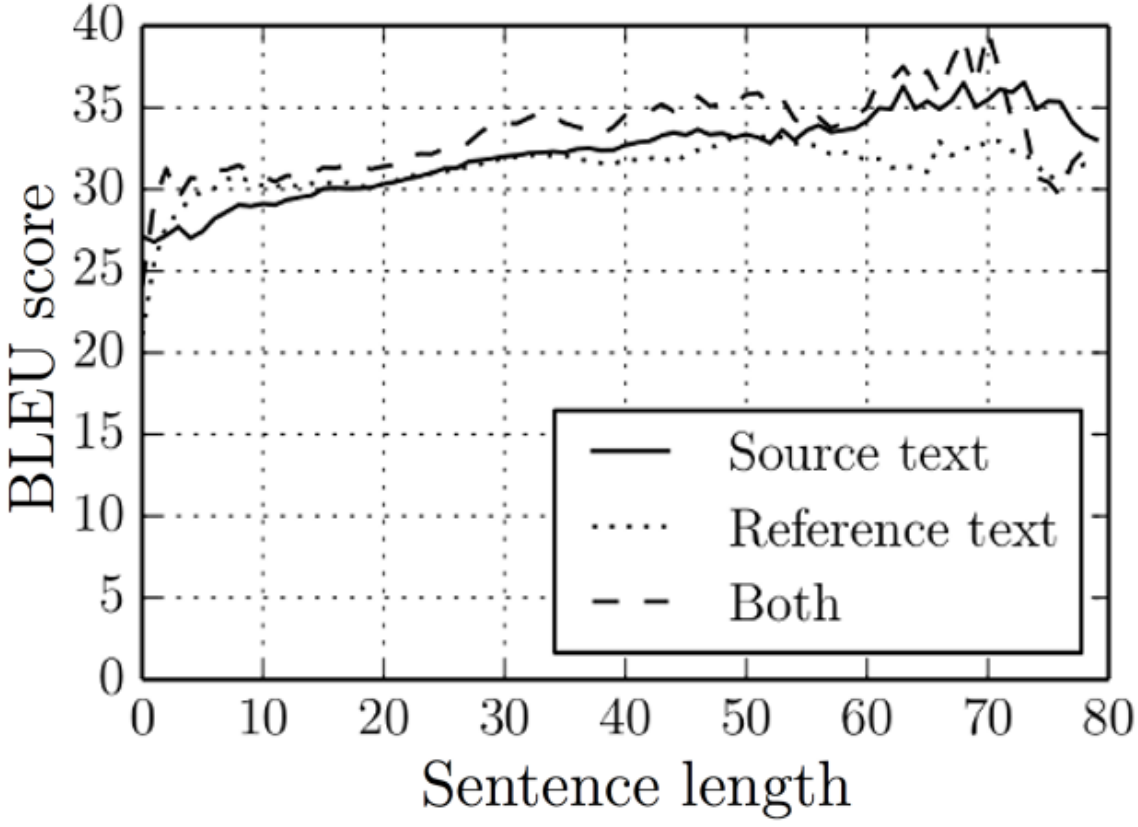


[Cho et al., 2014]

Vanilla RNN Architecture

Why Vanilla?

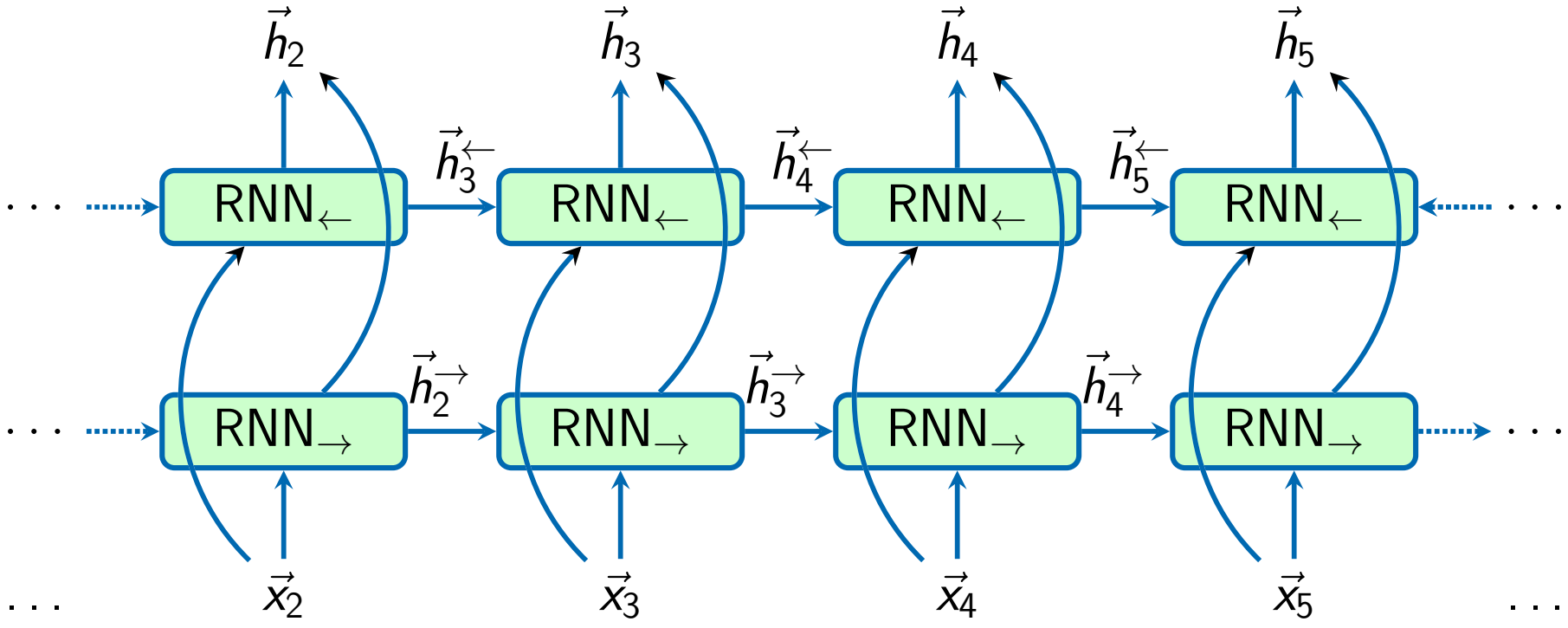
- SMT was better at that!



[Cho et al., 2014]

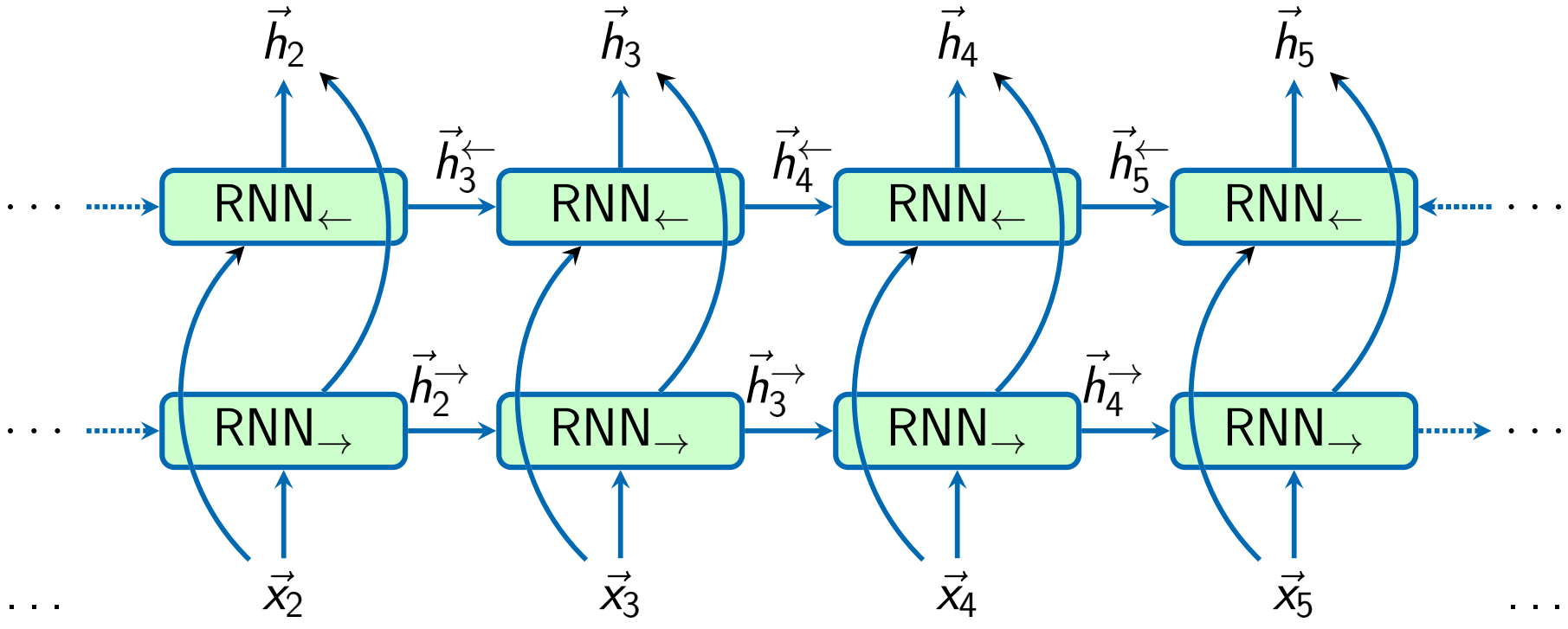
RNN Architecture

Bidirectional Encoder Hidden Layer



RNN Architecture

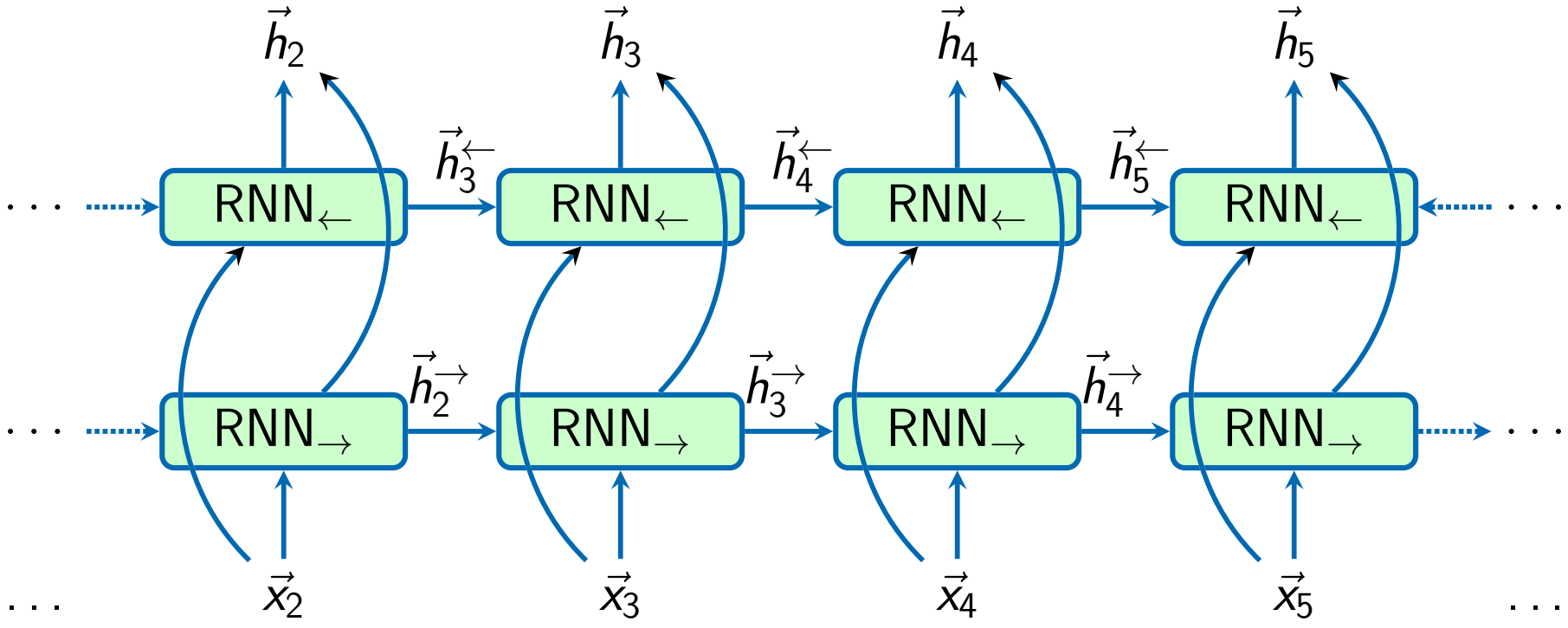
Bidirectional Encoder Hidden Layer



- Helps but not enough
- What else?

RNN Architecture

Bidirectional Encoder Hidden Layer

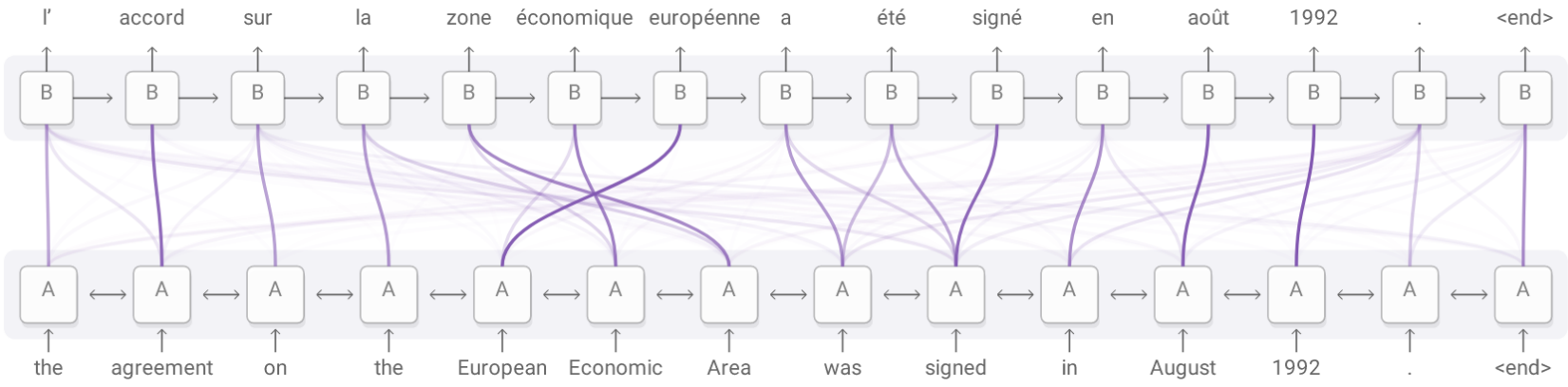


- Helps but not enough
- What else? **Attention!**

RNN Architecture

The Attention Mechanism

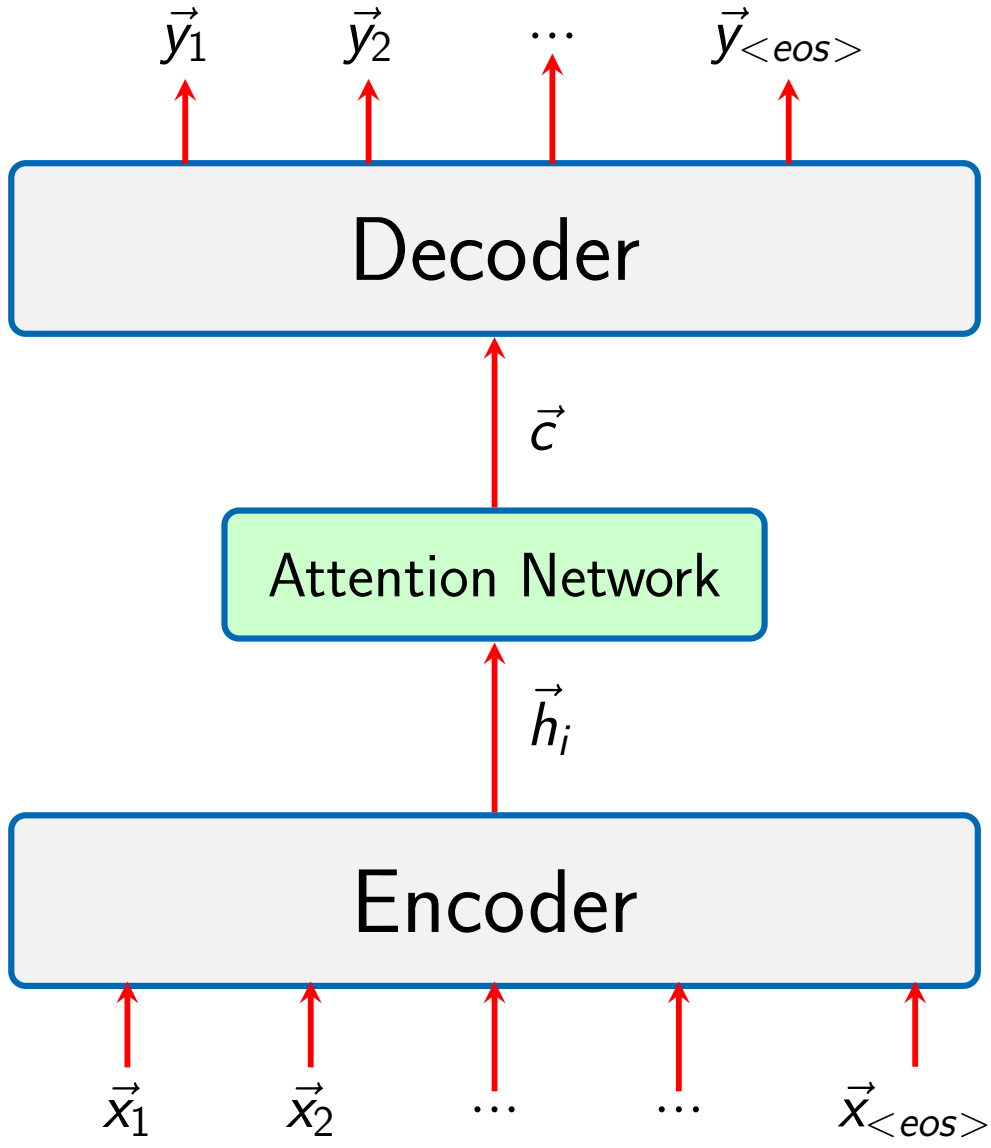
- Intuition: Not all the words contribute equally for a translation
- Let's weight! (weights, softmaxs, nns...)



`https://distill.pub/2016/augmented-rnns`

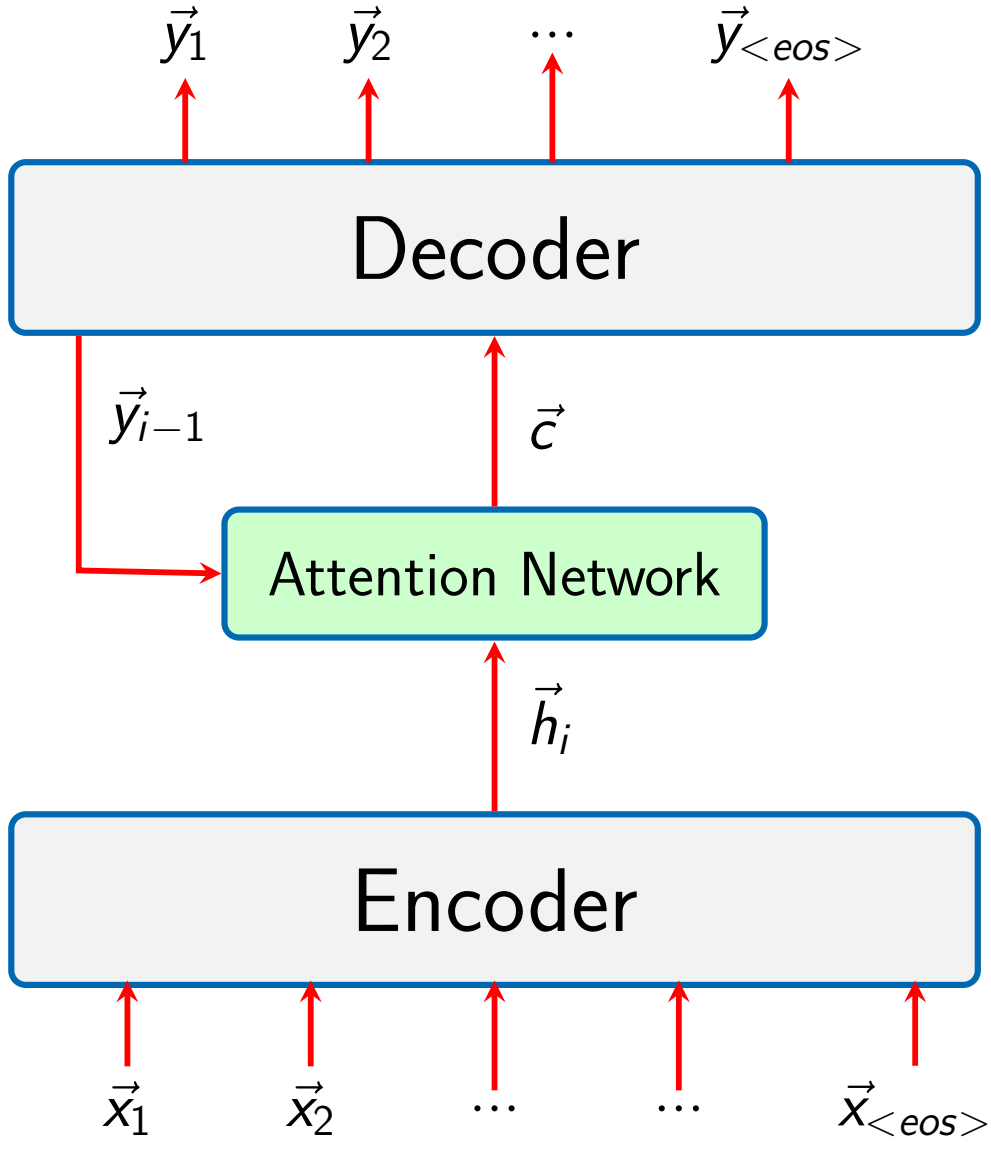
RNN Architecture

The Attention Mechanism



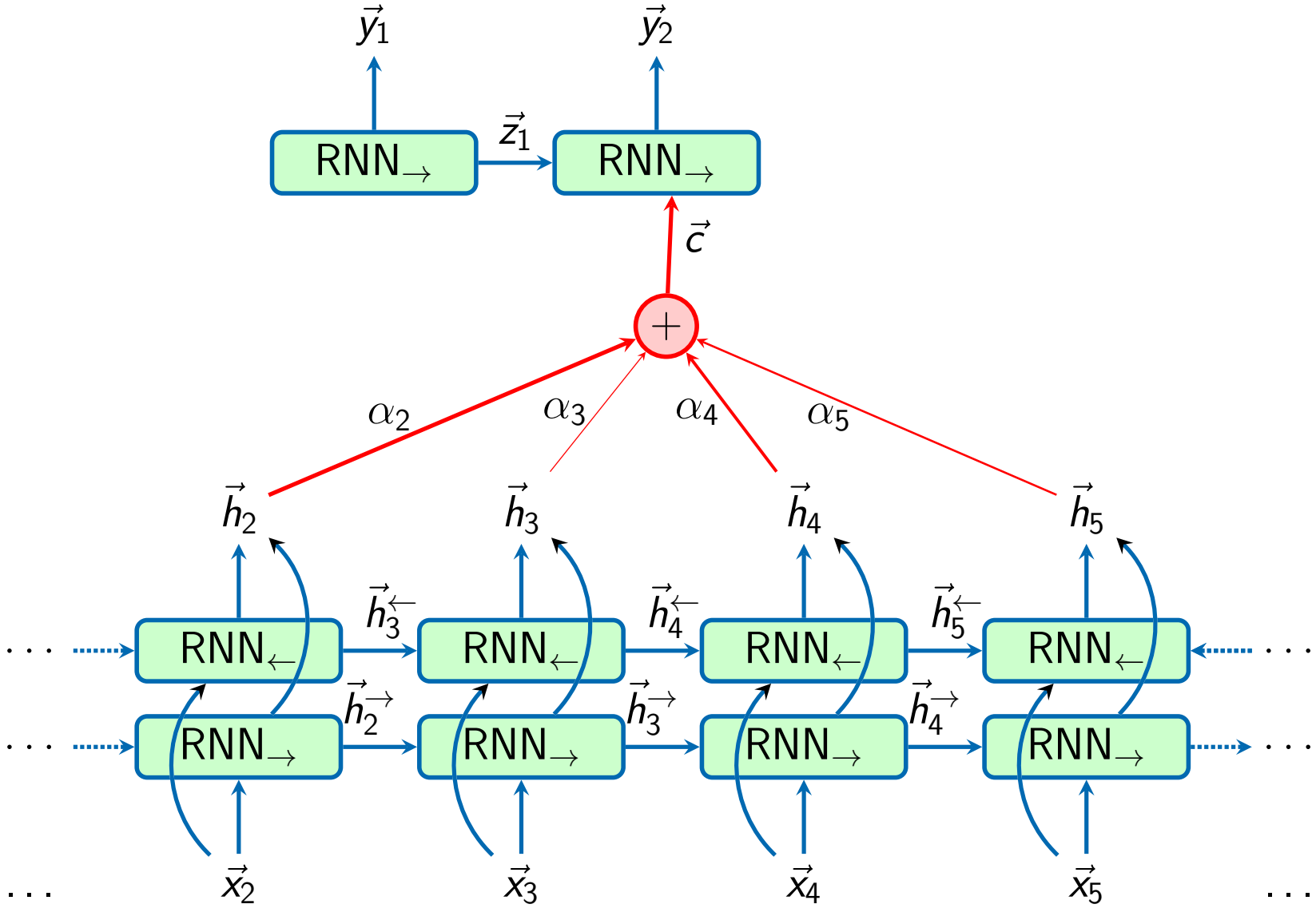
RNN Architecture

The Attention Mechanism



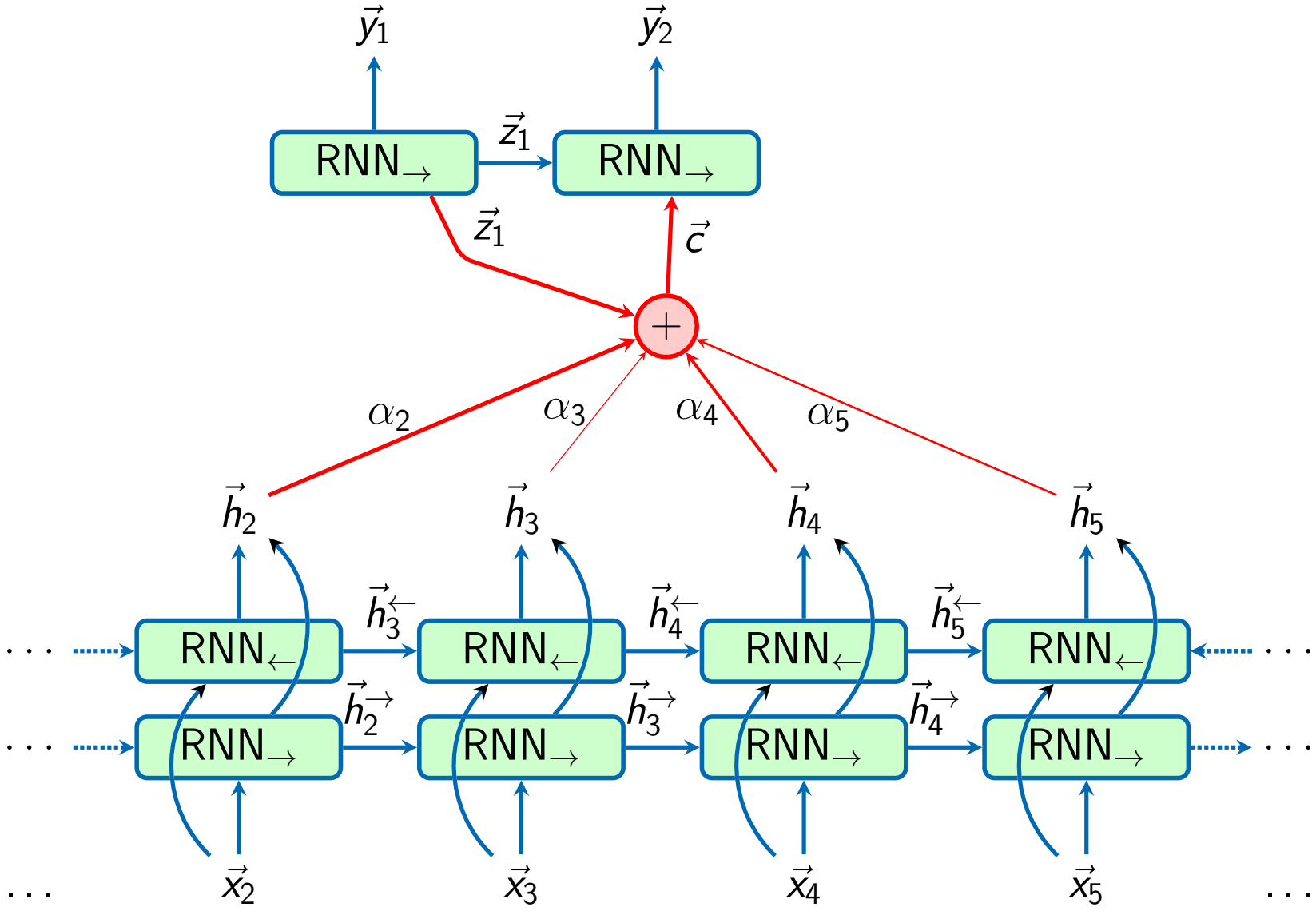
RNN Architecture

The Attention Mechanism



RNN Architecture

The Attention Mechanism



Encoder with bidirectional RNNs

Hidden vectors: concatenation of the forward ($\overrightarrow{\mathbf{h}}_i$) and backward ($\overleftarrow{\mathbf{h}}_i$) encoder RNN hidden states:

$$\begin{aligned}\mathbf{h}_i &= \left[\overleftarrow{\mathbf{h}}_i, \overrightarrow{\mathbf{h}}_i\right] \\ &= \left[f(\overleftarrow{\mathbf{h}}_{i-1}, \mathbf{r}_i), f(\overrightarrow{\mathbf{h}}_{i+1}, \mathbf{r}_i)\right],\end{aligned}$$

- f is a recurrent unit (LSTM, GRU, etc.)
- $\mathbf{r}_i$ is the embedding space representation of the source word at position i: $\mathbf{r}_i = \mathbf{W}_\mathbf{x} \cdot \mathbf{x}_i$.

RNN Architecture

The Attention Mechanism

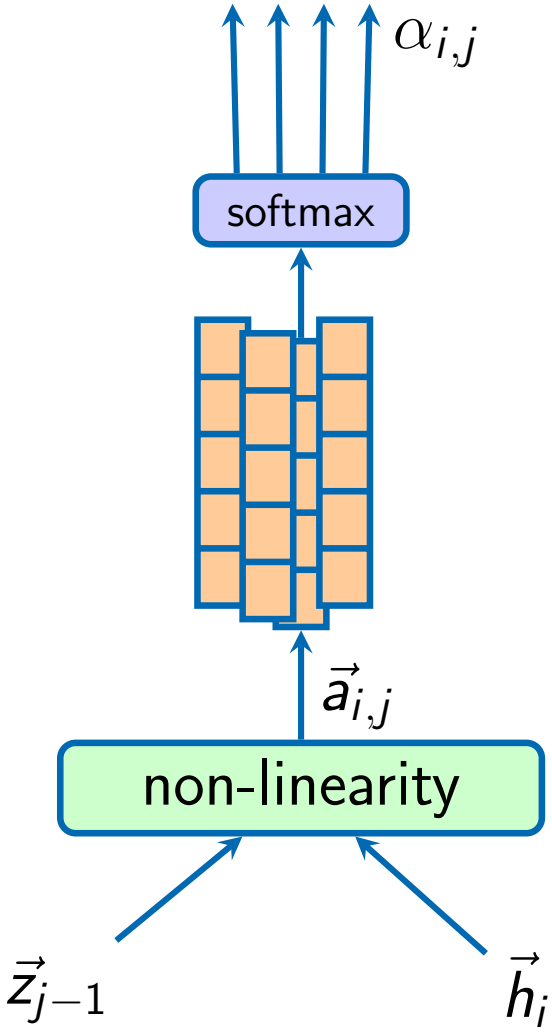
Decoder with RNNs

$$\mathbf{z}_j = g(\mathbf{z}_{j-1}, \mathbf{t}_{j-1}, \mathbf{q}_j)$$
$$\mathbf{t}_{j-1} = \mathbf{W}_\mathbf{y} \cdot \mathbf{y}_{j-1},$$

- $\mathbf{z}_j$ is the recurrent hidden state of the decoder
- $\mathbf{t}_j$ is the continuous representation of the target word
- g is a non-linear function
- $\mathbf{W}_\mathbf{y}$ is the matrix of the target embeddings
- The weighted context vector $\mathbf{q}_j$ is calculated by the *attention mechanism*

RNN Architecture

Attention Network



RNN Architecture

Attention Network

In equations:

$$a(\mathbf{z}_{j-1}, \mathbf{h}_i) = \mathbf{v}_a \cdot \tanh(\mathbf{W}_a \cdot \mathbf{z}_{j-1} + \mathbf{U}_a \cdot \mathbf{h}_i)$$

$$\alpha_{ij} = \mathrm{softmax}\left(\mathrm{a}(\mathbf{z}_{\mathrm{j}-1}, \mathbf{h}_{\mathrm{i}})\right)$$

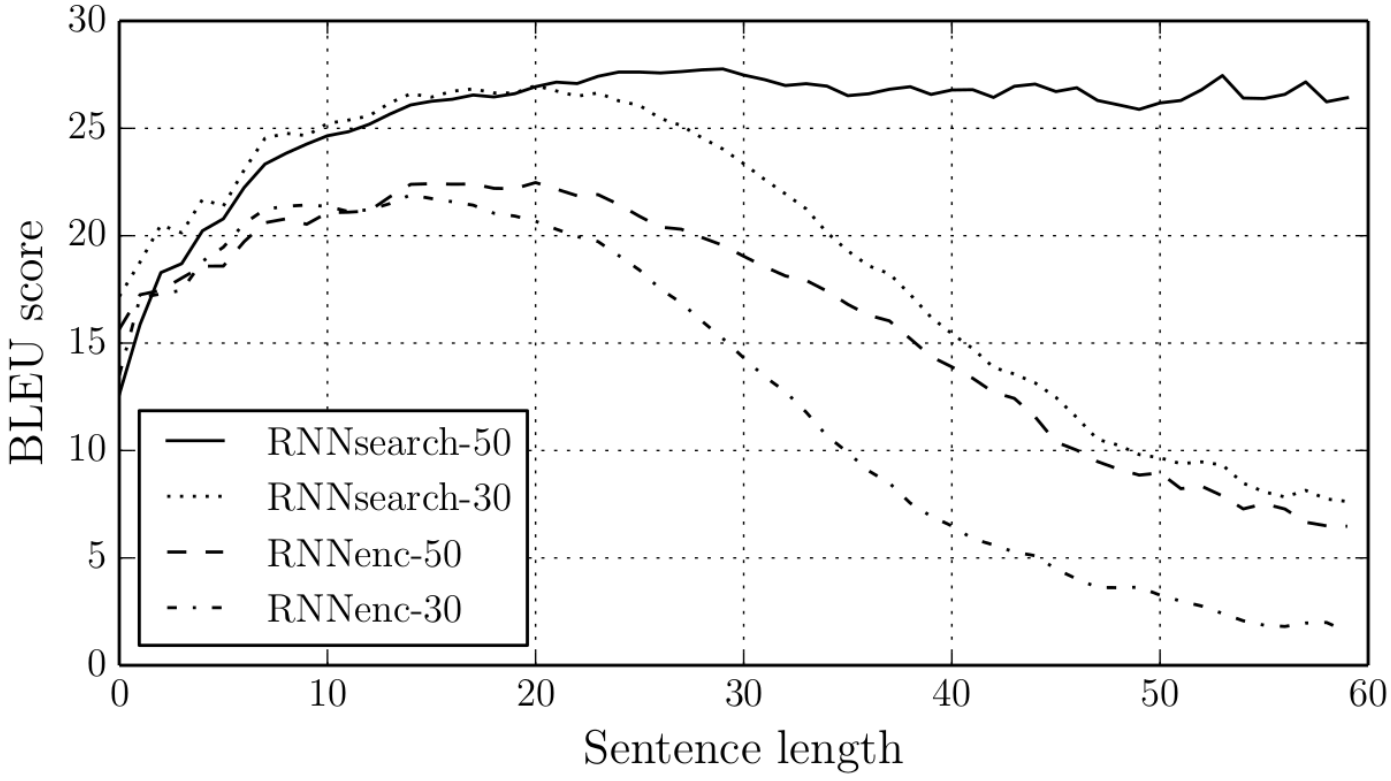
$$\mathbf{q}_j = \sum_i \alpha_{ij} \mathbf{h}_i$$

The attention mechanism takes the decoder's previous hidden state $\mathbf{z}_{j-1}$ and the context vector $\mathbf{h}_i$ as inputs and weighs them up with the trainable weight matrices $\mathbf{W}_a$ and $\mathbf{U}_a$, respectively.

RNN Architecture

The Attention Mechanism

- What achieves attention?



[Bahdanau et al., 2015]

Transformer Architecture

So, Attention is all you Need?!

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

Transformer Architecture

Remember the Language Modelling Lecture

We have already done the hard work:

- Remember self-attention?
- Remember multi-head attention?

`http://jalammar.github.io/illustrated-transformer/`

Transformer Architecture

Self-Attention

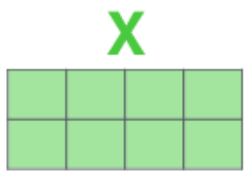
1) This is our input sentence

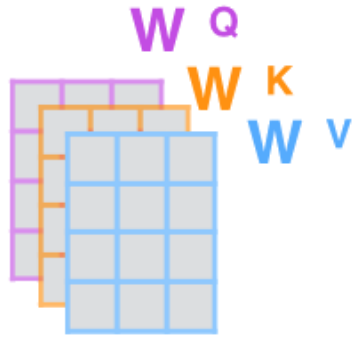
2) We embed each word

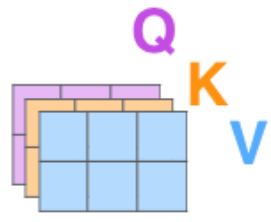
3) We multiply X with weight matrices

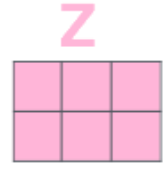
4) Calculate attention using the resulting Q/K/V matrices

Thinking
Machines









$$\text{softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) V = Z$$

Transformer Architecture

Multi-head Attention

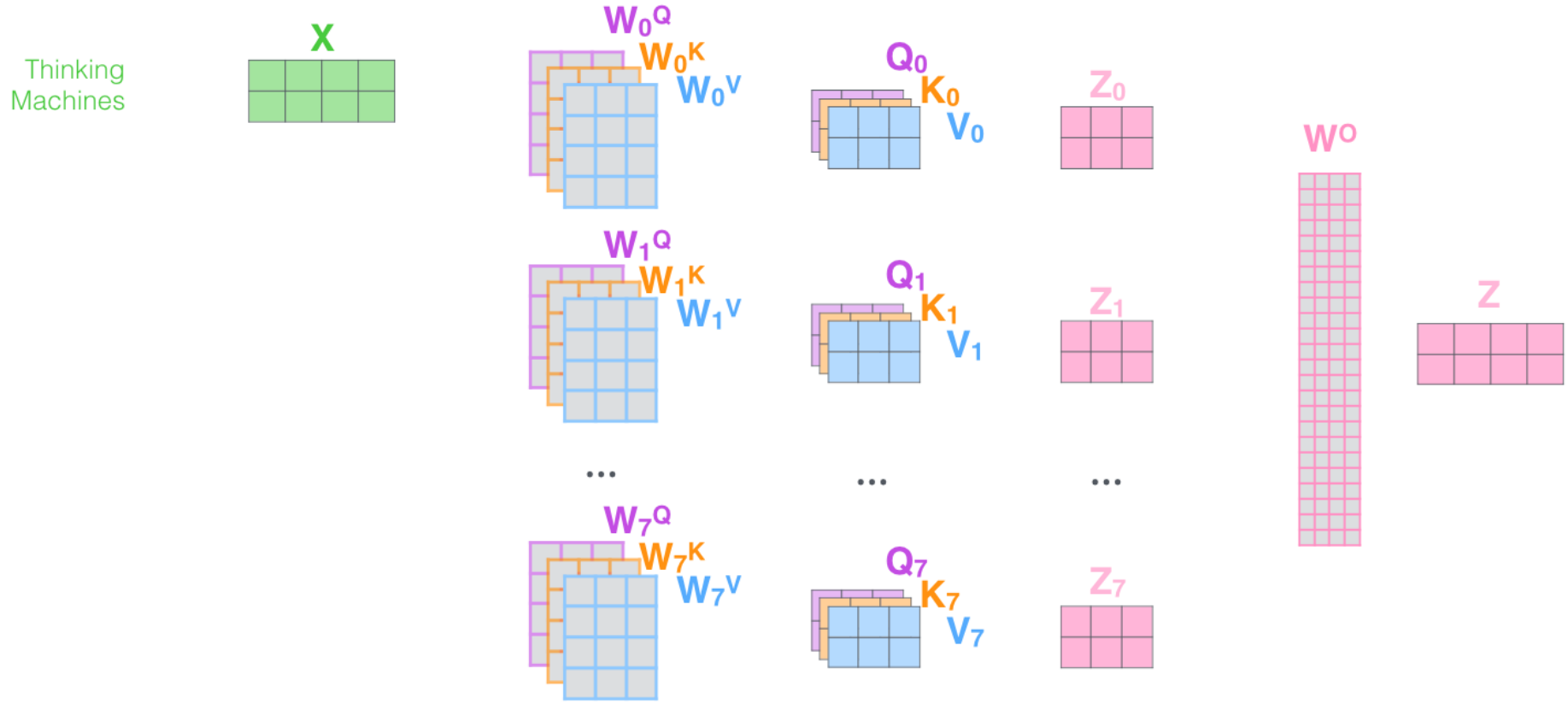
1) This is our input sentence

2) We embed each word

3) Split into 8 heads. We multiply X with weight matrices

4) Calculate attention using the resulting Q/K/V matrices

5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer



Transformer Architecture

Structure

- (Self)Attention has multiple heads
- Encoder has multiple encoder layers
- Decoder has multiple decoder layers
- (Inter)Attention behaves as usual (one per decoder layer)

Transformer Architecture

Structure

- (Self)Attention has multiple heads
- Encoder has multiple encoder layers
- Decoder has multiple decoder layers
- (Inter)Attention behaves as usual (one per decoder layer)

Why all this?

Transformer Architecture

Structure

- (Self)Attention has multiple heads
- Encoder has multiple encoder layers
- Decoder has multiple decoder layers
- (Inter)Attention behaves as usual (one per decoder layer)

Why all this? Because it works...

Transformer Architecture

Structure

Why all this? Because it works... **but also:**

Transformer Architecture

Structure

Why all this? Because it works... **but also:**

- Sequentiality in RNNs prohibits parallelization within instances
- CNNs are easy to parallelise but need $\log(n)$ layers to consider words at positions i and $i + n$

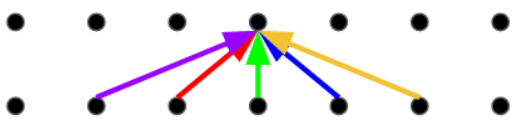
Transformer Architecture

Structure

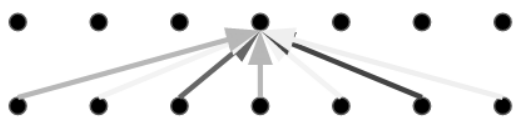
Why all this? Because it works... **but also:**

- Sequentiality in RNNs prohibits parallelization within instances
- CNNs are easy to parallelise but need $\log(n)$ layers to consider words at positions i and $i + n$

Convolution



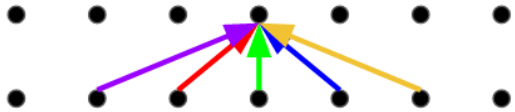
Self-Attention



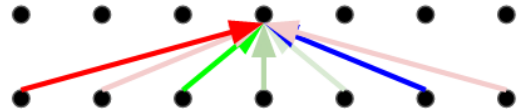
Transformer Architecture

Structure

Convolution



Multi-Head Attention

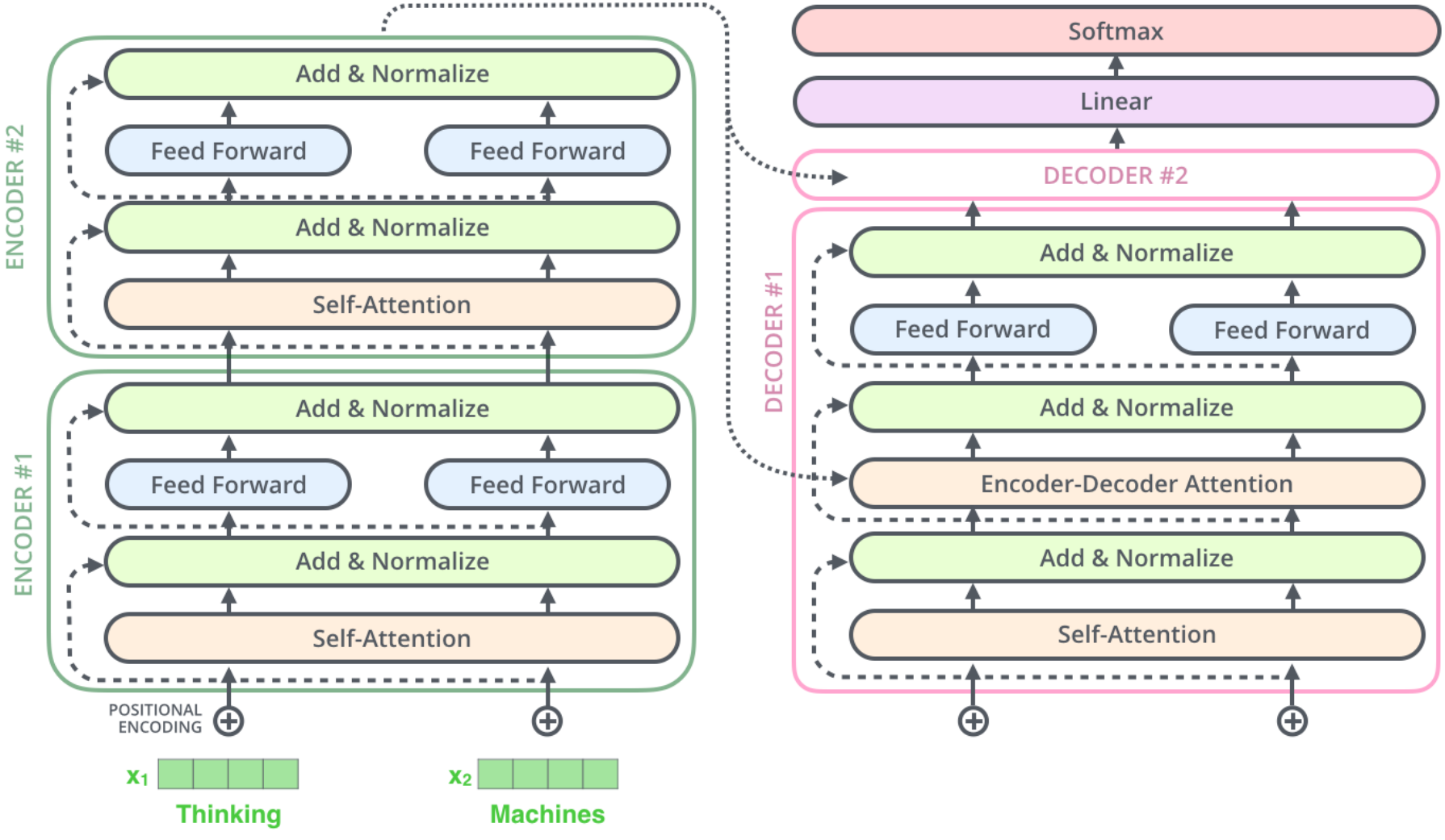


- Convolution: a different linear transformation for each relative position. Allows you to distinguish what information came from where.
- Multiple attention layers (head) in parallel. Each head uses different linear transformations which can learn different relationships.

(Lukasz Kaiser slides)

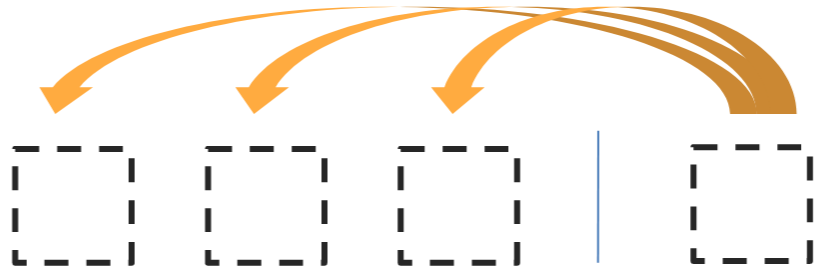
Transformer Architecture

The Complete (Reduced) Picture



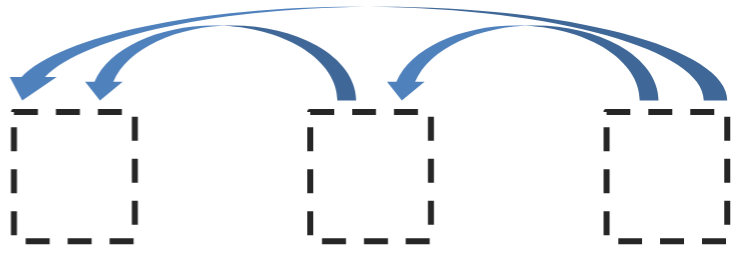
Transformer Architecture

Different Attentions



Encoder-Decoder Attention

Encoder Self-Attention

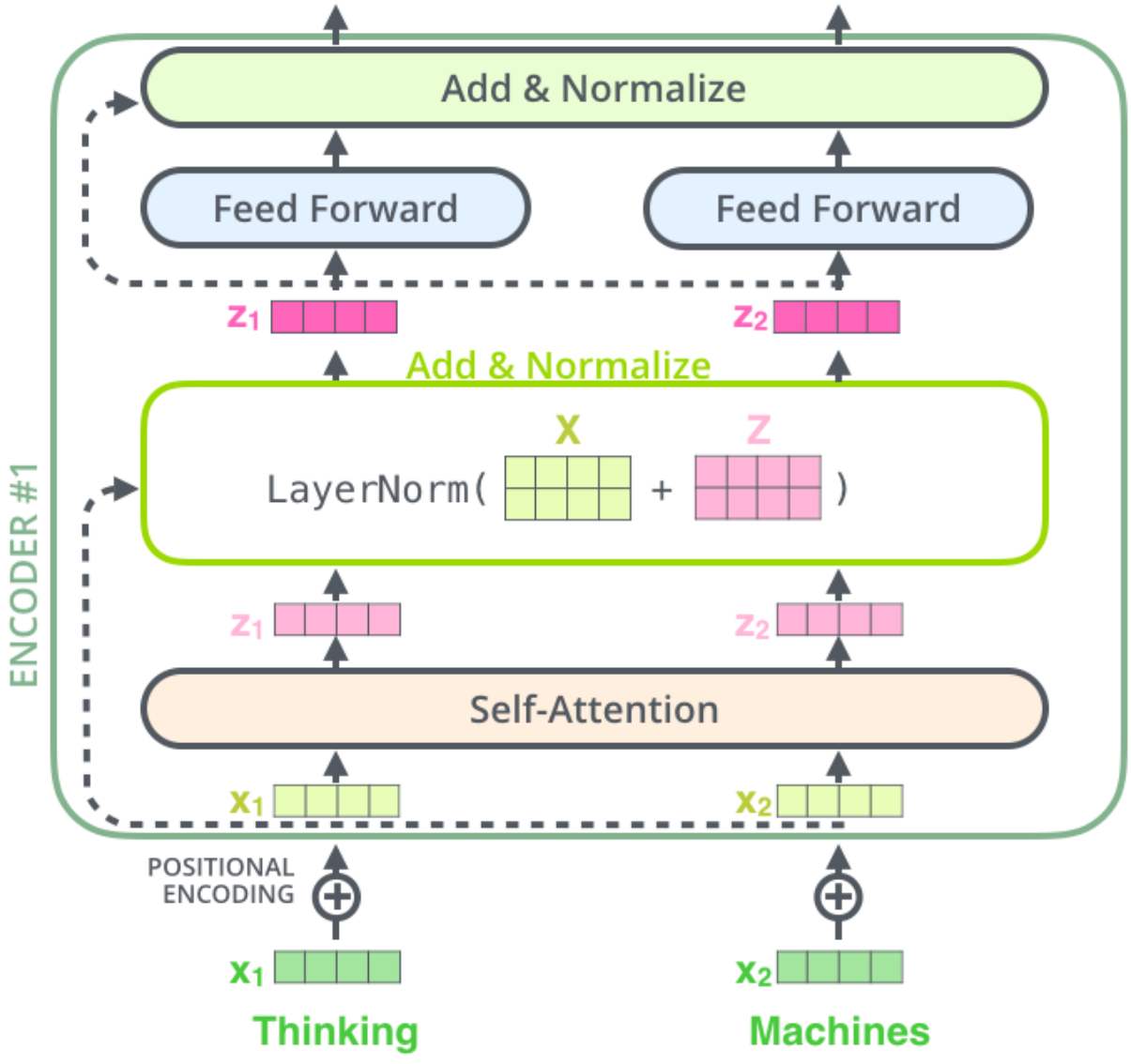


MaskedDecoder Self-Attention

(Lukasz Kaiser slides)

Transformer Architecture

An Encoder Layer



Transformer Architecture

Positional Embeddings

- Transformer is not sequential
- No information about the sentence structure, kind of BoWs
- What do we do? Encode position too!
- How? As sinusoidal waves (??)

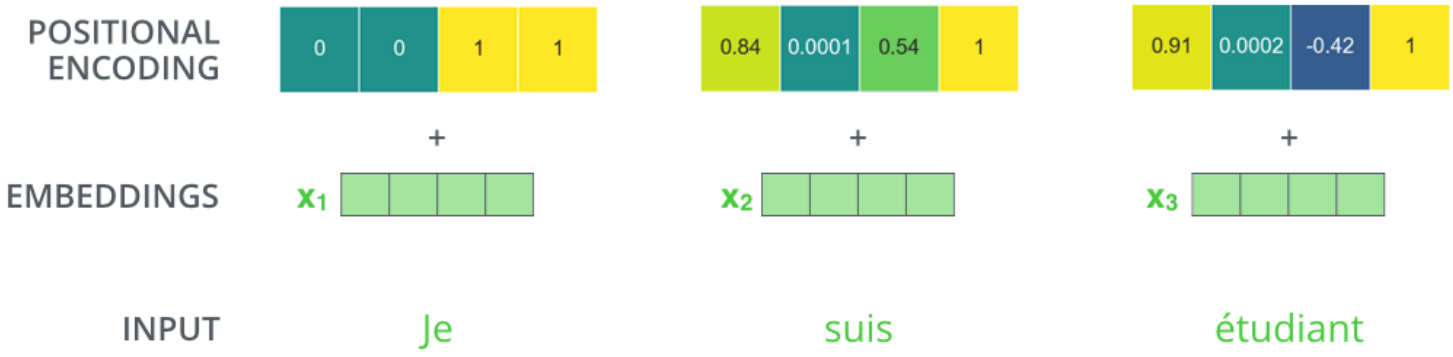
$$\text{PE}(pos, 2i) = \sin(pos/10000^{2i/d})$$

$$\text{PE}(pos, 2i+1) = \cos(pos/10000^{2i/d})$$

where pos is the position of the token, i is the dimension.

Transformer Architecture

Positional Embeddings



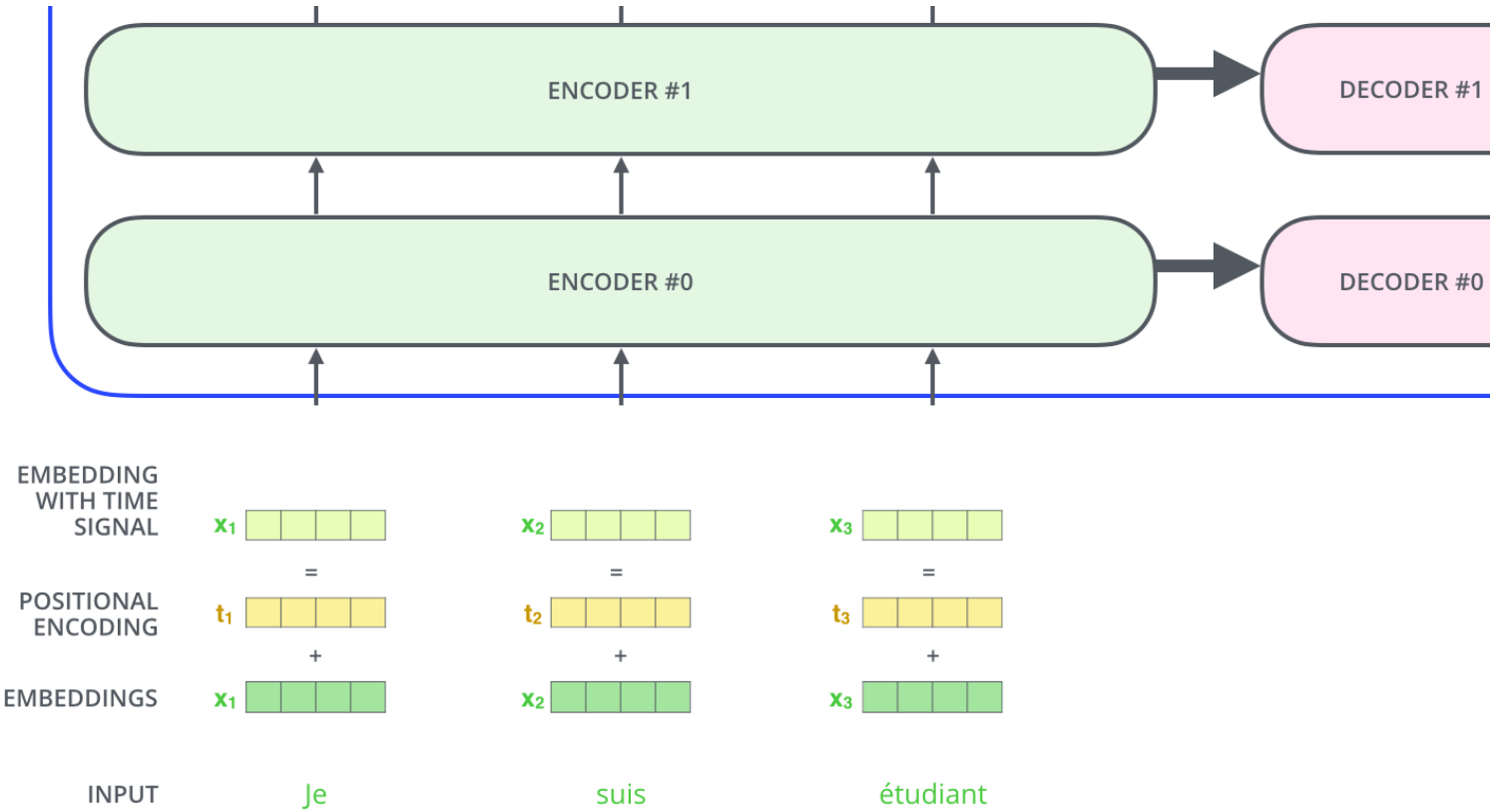
$$\text{PE}(pos, 2i) = \sin(pos/10000^{2i/d})$$

$$\text{PE}(pos, 2i+1) = \cos(pos/10000^{2i/d})$$

where pos is the position of the token, i is the dimension.

Transformer Architecture

Positional Embeddings



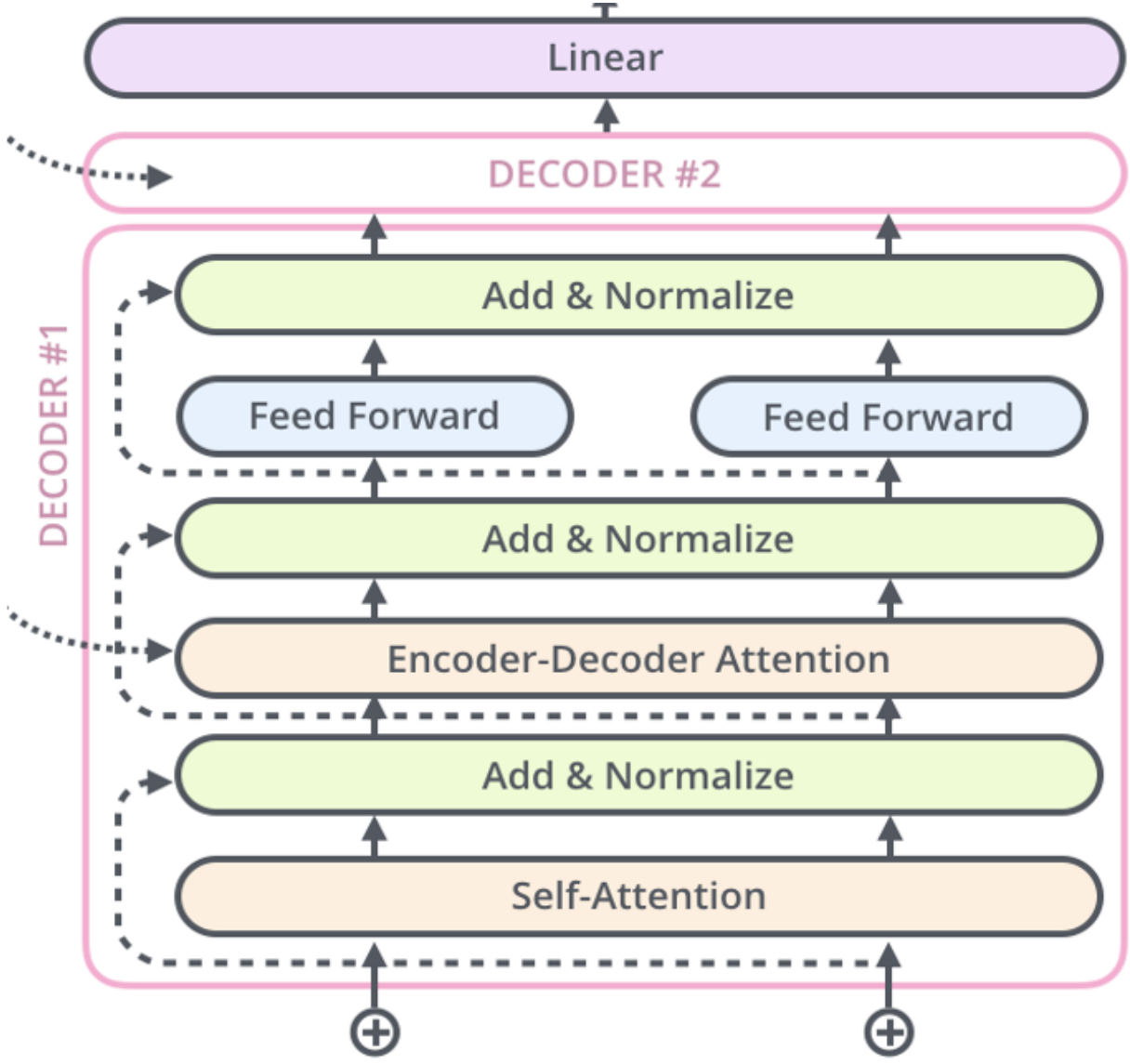
Transformer Architecture

Encoder vs. Decoder Layers

- **Masking**. The decoder cannot see *the future* when predicting the next word.
- How? $-\infty$ before self-attention softmax
- **Enc-Dec Attention**. Queries are taken from the layer below it, but keys and values from the final output of the encoder.
- The decoder adds an additional **linear and softmax** layer (just as RNNs NMT)

Transformer Architecture

A Decoder Layer



Transformer Architecture

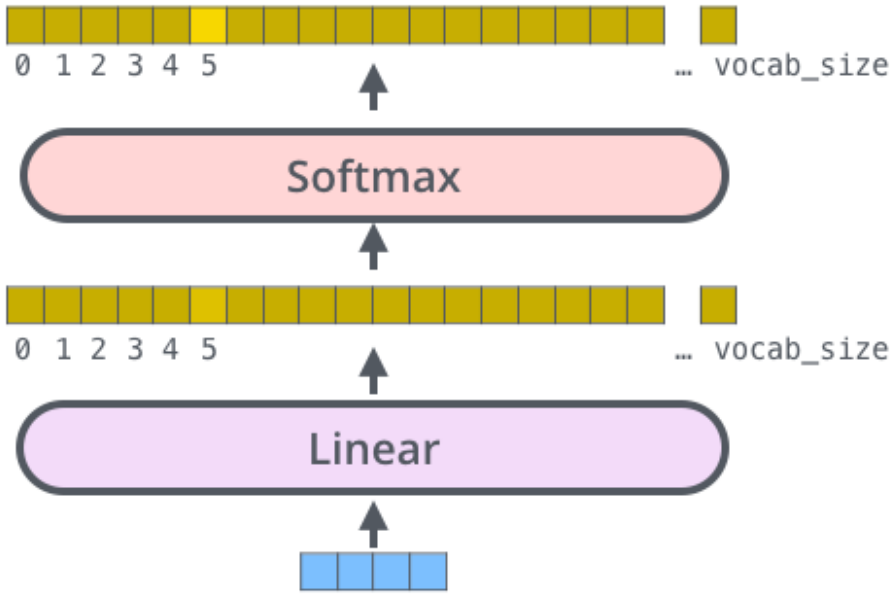
The Icing on the Cake

Which word in our vocabulary is associated with this index?

am

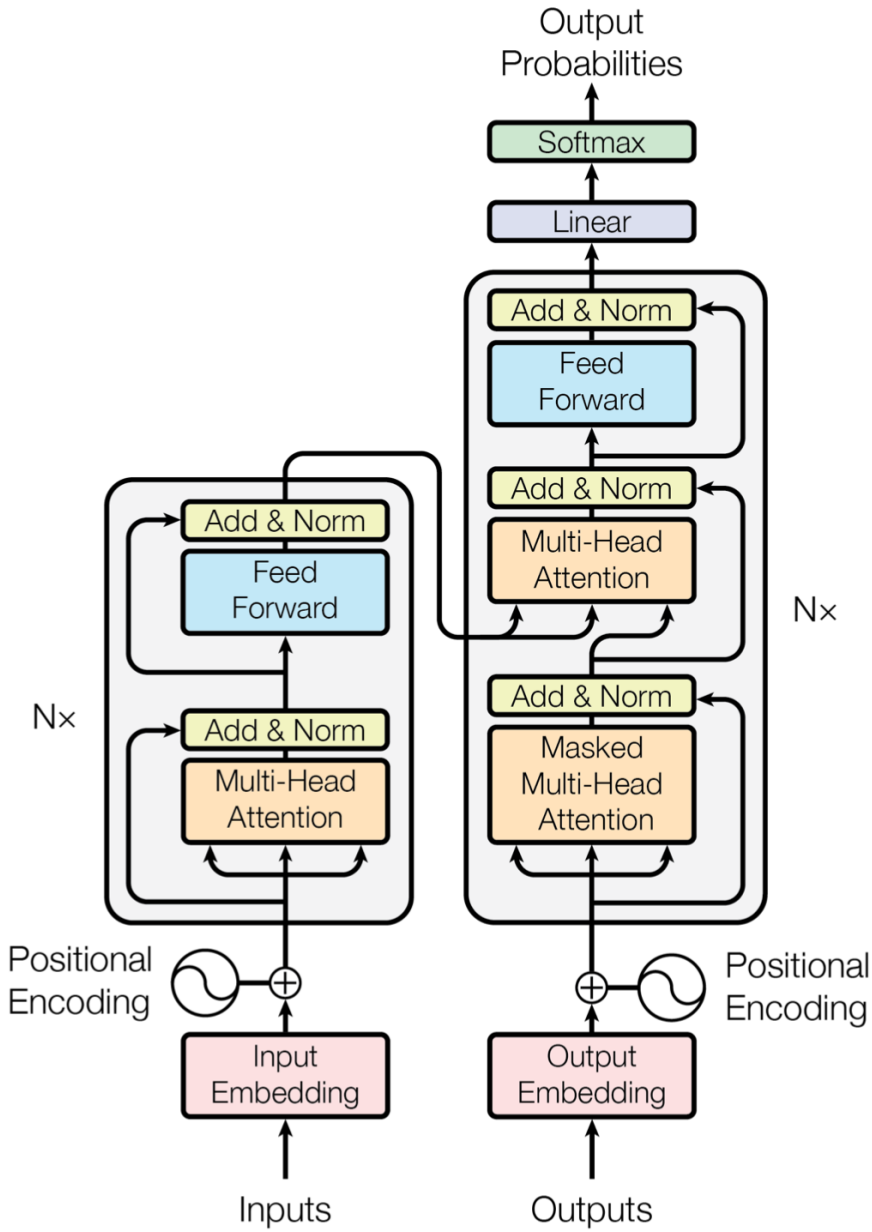
Get the index of the cell with the highest value **(argmax)**

5



Transformer Architecture

The Complete (Original) Picture



Transformer Architecture

Today's Icing on the Cake

Let's see some animations

`http://jalammar.github.io/illustrated-transformer/`

I love this blog!

Outline

Software & References

NMT Packages

- **Nematus**
  `https://github.com/EdinburghNLP/nematus`
- **Marian**
  `https://github.com/marian-nmt/marian`
- **OpenNMT**
  `http://opennmt.net/`
- **Facebook**
  `https://github.com/pytorch/fairseq`

Software & References

NMT & SMT Packages

- **Google**
  `https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/models/transformer.py`
- ...

- **Moses (SMT!)**
  `https://github.com/moses-smt/mosesdecoder`

Software & References

References Used

- Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra and Robert L.Mercer. 1993. **The mathematics of statistical machine translation: parameter estimation**. Computational Linguistics, 19(2). Pages 263–311.

- Philipp Koehn, Franz Josef Och and Daniel Marcu. **Statistical phrase-based translation**. 2015. Proceedings of the Human Language Technology and North AmericanAssociation for Computational Linguistics Conference (HLT/NAACL).

- D. Bahdanau, K. Cho and Y. Bengio. 2014. **Neural machine translation by jointly learning to align and translate**. Proceddings of the 3rd Int. Conf. Learn. Represent. Pages 1–15.

Software & References

References Used

- I. Sutskever, O. Vinyals and Q. V. Le. 2014. **Sequence to sequence learning with neural networks.** Proc. Adv. Neural Inf. Process. Syst. 27. Pages 3104–3112.

- Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. **Effective Approaches to Attention-based Neural Machine Translation**. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Pages 1412–1421.

- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin. 2017. **Attention Is All You Need.** 31st Conference on Neural Information Processing Systems (NIPS 2017).

Too much for a day?

Questions?

Machine Translation

Cristina España-Bonet

UdS & DFKI, Saarbrücken, Germany

Artificial Intelligence with Deep Learning

2nd May 2019