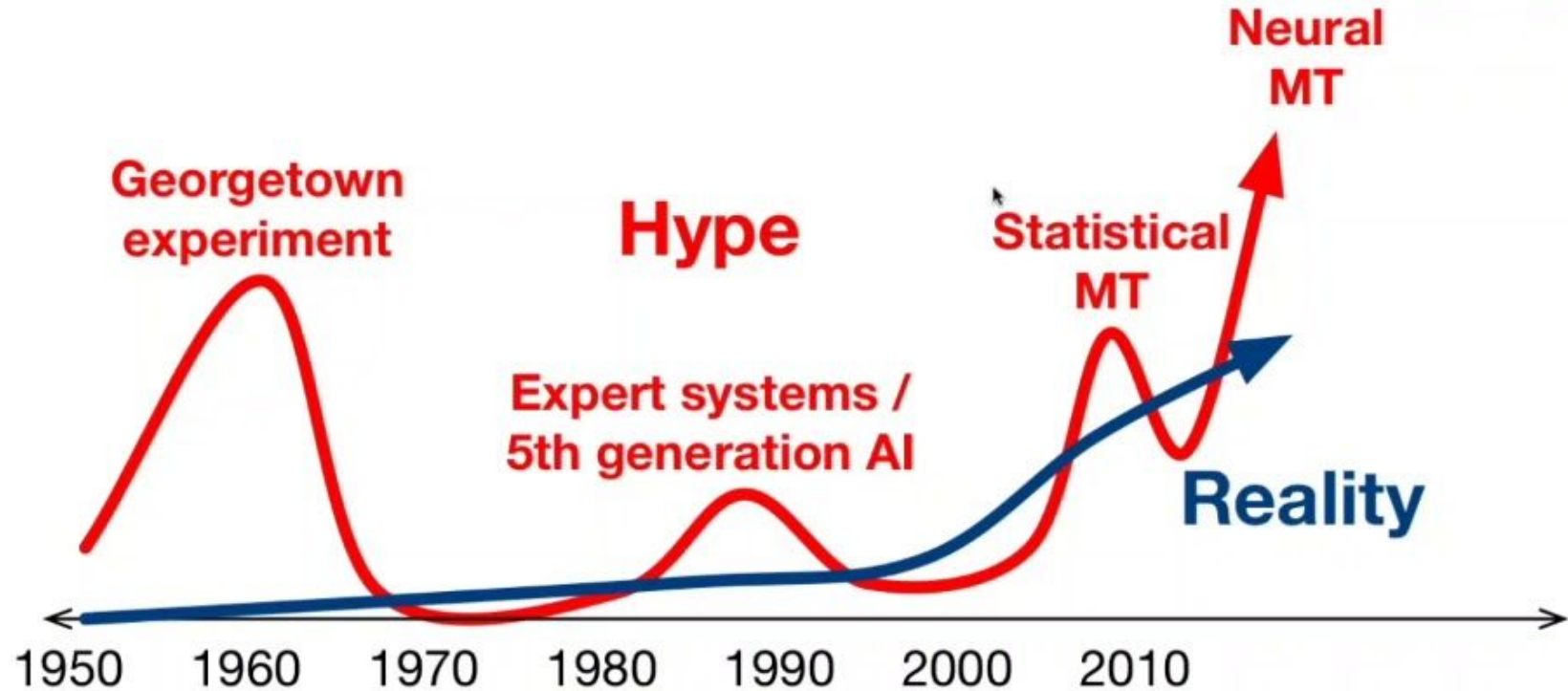


# Machine Translation, Attention Mechanism

**Radoslav Neychev**

- Machine Translation historical overview
  - Statistical Machine Translation
  - Word alignments
- Neural Machine Translation (NMT)
  - Seq2Seq
  - Beam Search
- Attention mechanism

# Historical overview



# Before Deep Learning

# 1950s: first Machine Translation

- Georgetown experiment (7 Jan 1954)
  - Automatic Russian-English translation of 60 sentences
  - 250 vocabulary articles
  - 6 grammar rules
  - Calculated on Mainframe IBM 701
- The same experiment in the USSR (1954 too)
  - Rule-based translation
  - Calculated on BESM

# 1990-2010: Statistical Machine Translation

We want to find best English sentence  $y$ , given French sentence  $x$

Let's use Bayes Rule to break this down into two components:

$$\begin{aligned} & \operatorname{argmax}_y P(y|x) \\ &= \operatorname{argmax}_y \underbrace{P(x|y)}_{\text{Translation Model}} \underbrace{P(y)}_{\text{Language Model}} \end{aligned}$$

## Translation Model

Models how words and phrases  
should be translated (*fidelity*).  
Learnt from parallel data.

## Language Model

Models how to write  
good English (*fluency*).  
Learnt from monolingual data.

# 1990-2010: Statistical Machine Translation

How to learn translation model from the parallel corpus?

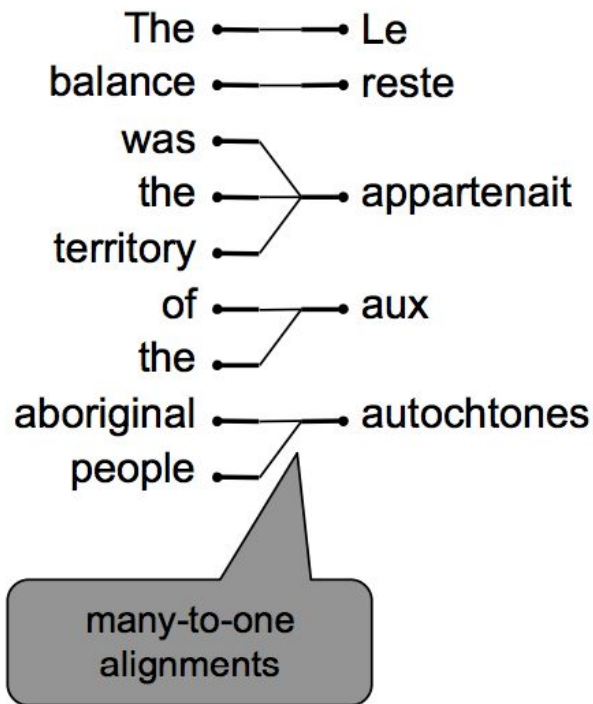
Let's calculate

$$P(x, a|y)$$

Where **a** is an **alignment** (word-level correspondence between French sentence  $x$  and English sentence  $y$ )

# 1990-2010: Statistical Machine Translation

Alignment can be: **many-to-one**

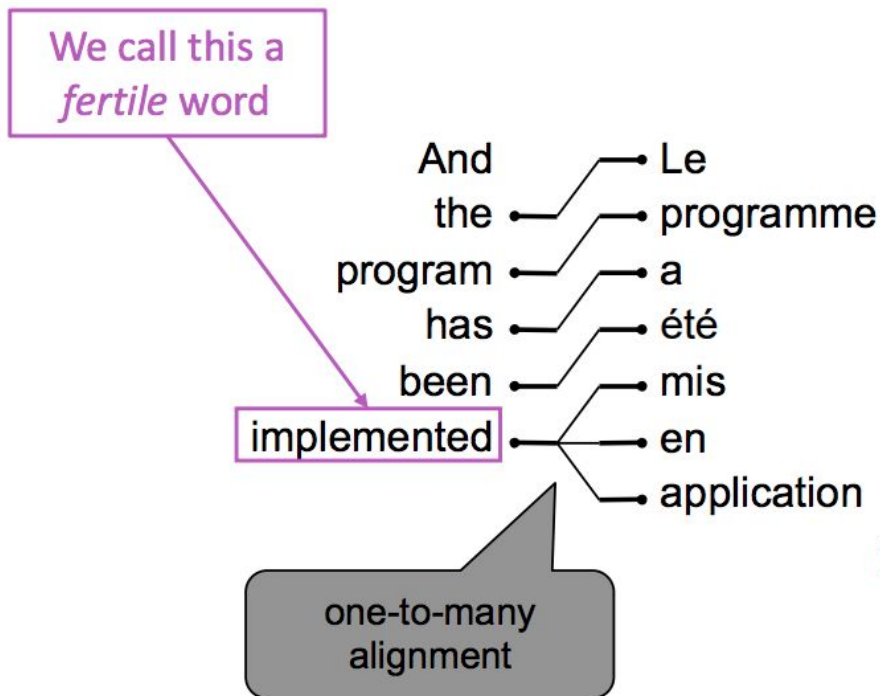


	Le	reste	appartenait	aux	autochtones
The	■				
balance		■			
was			■		
the			■		
territory			■		
of				■	
the				■	
aboriginal					■
people					■



# 1990-2010: Statistical Machine Translation

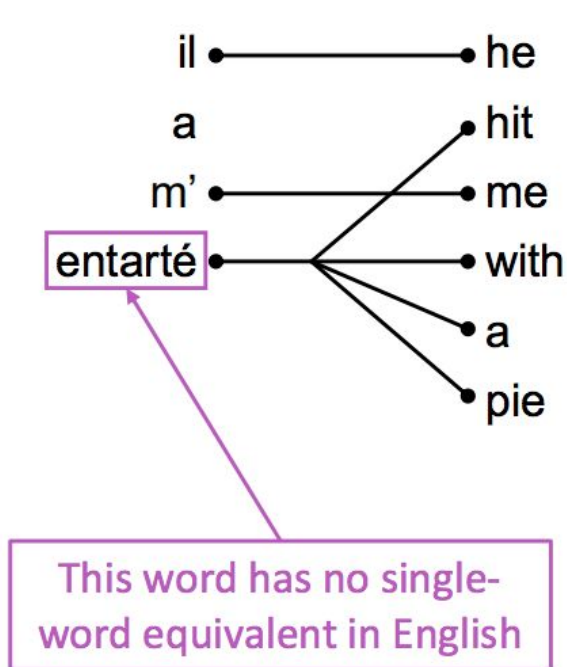
Alignment can be: **one-to-many**



	Le	programme	a	été	mis	en	application
And							
the							
program							
has							
been							
implemented							

# 1990-2010: Statistical Machine Translation

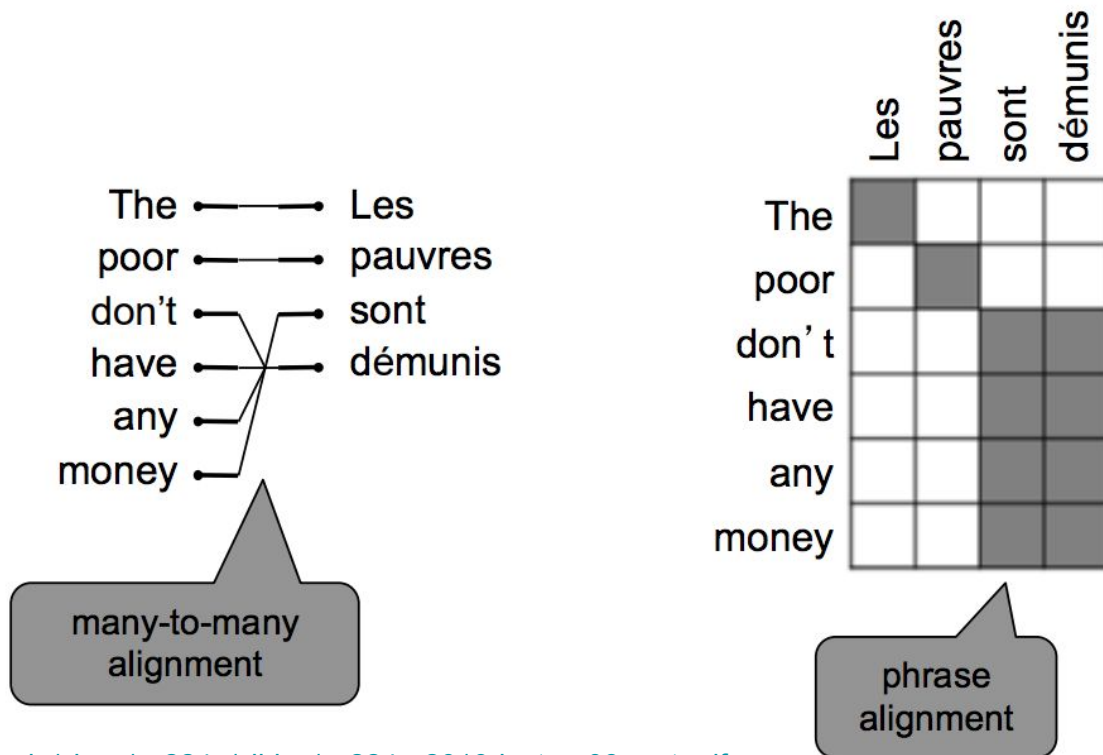
Some words are very fertile!



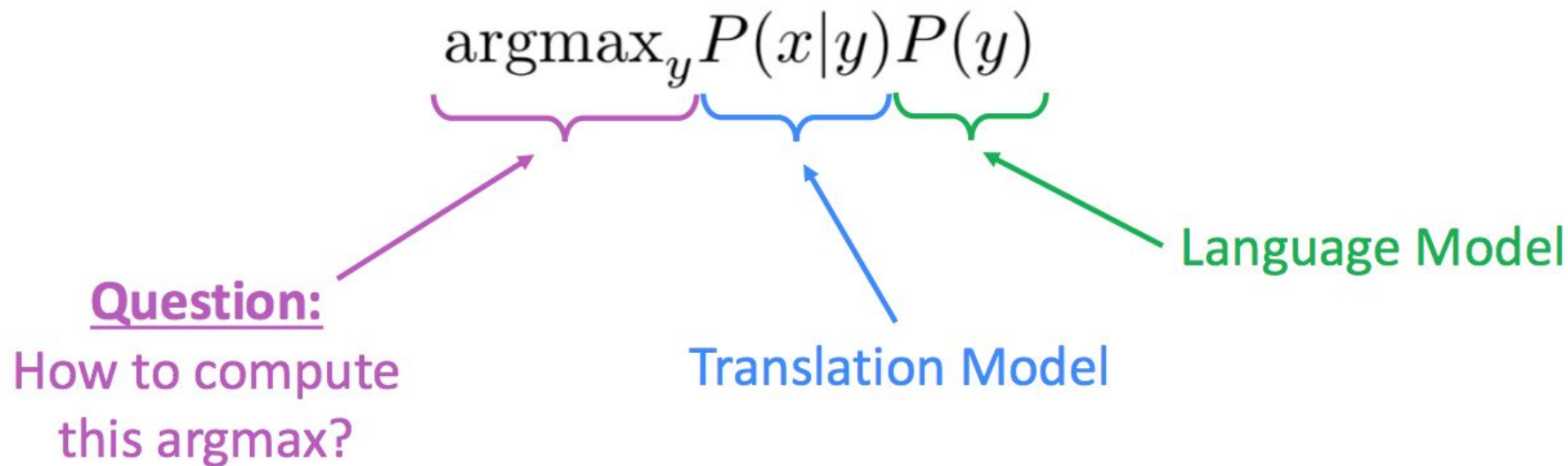
	he	hit	me	with	a	pie
il						
a						
m'						
entarté						

# 1990-2010: Statistical Machine Translation

Alignment can be: **many-to-many**



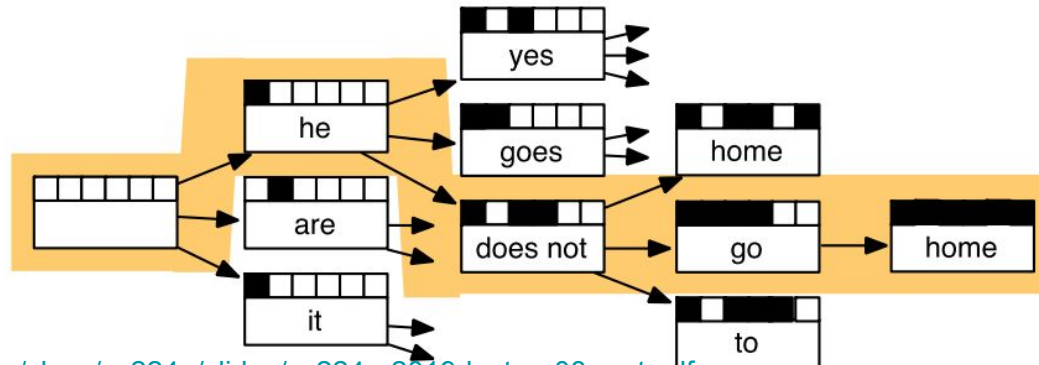
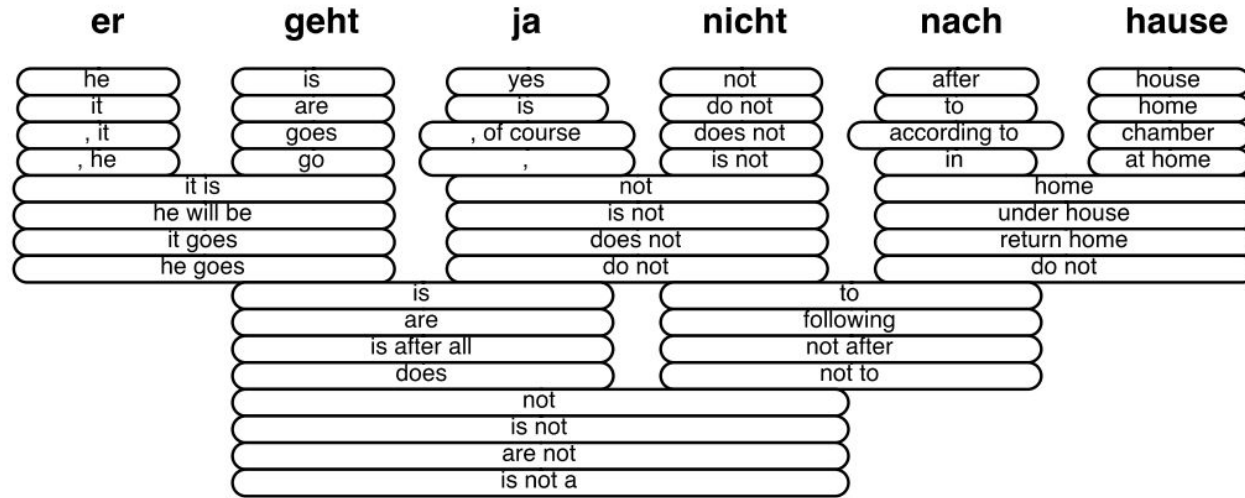
# 1990-2010: Statistical Machine Translation



Enumerate every possible  $y$  and calculate the probability? No!

Use a heuristic search algorithm to search for the best translation, discarding hypotheses that are too low-probability

# 1990-2010: Statistical Machine Translation



# 1990-2010: Statistical Machine Translation

- Systems had many separately-designed subcomponents
- Lots of feature engineering
- Need to design features to capture particular language phenomena
- Require compiling and maintaining extra resources (tables of equivalent phrases)
- Lots of human effort to maintain
- Repeated effort for each language pair!

# Neural Machine Translation

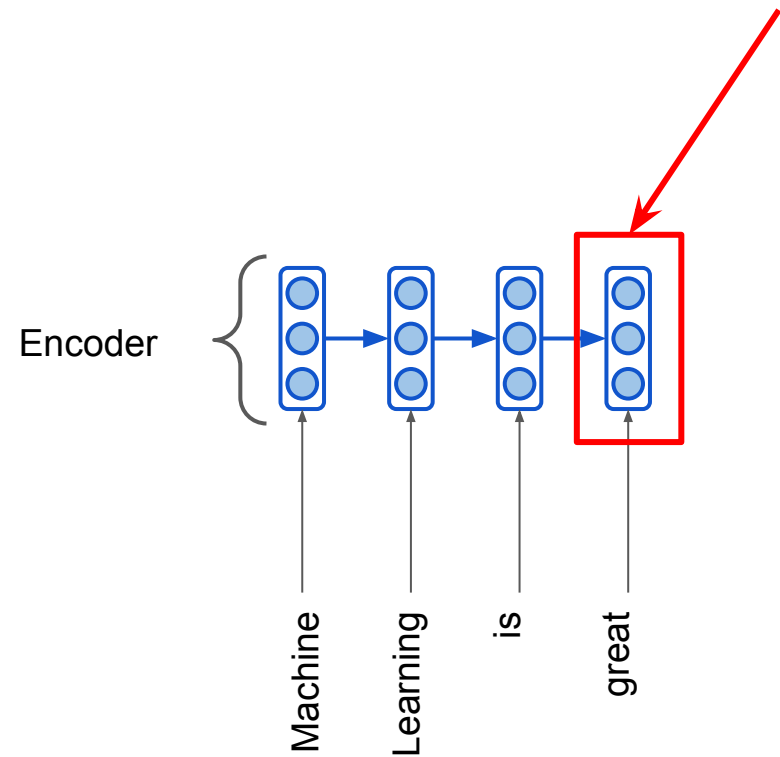
# What is Neural Machine Translation?

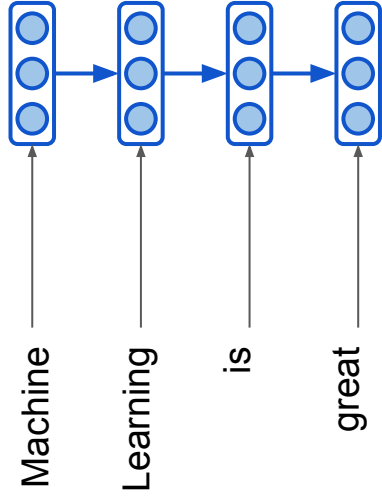
- Neural Machine Translation (NMT) is a way to do Machine Translation with a single neural network
- The neural network architecture is called sequence-to-sequence (aka **seq2seq**), it involves two **RNNs**



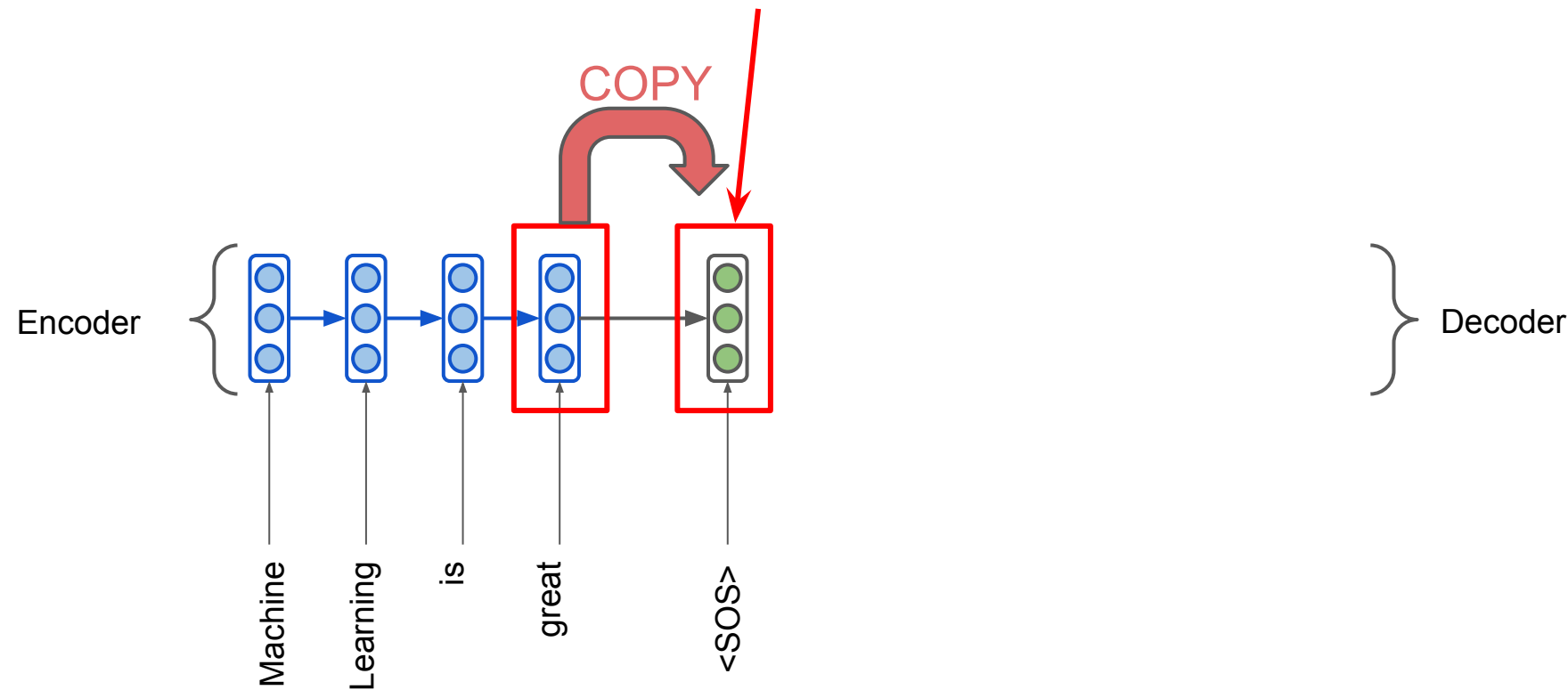
# Seq2seq NMT

This state encodes  
the whole sentence

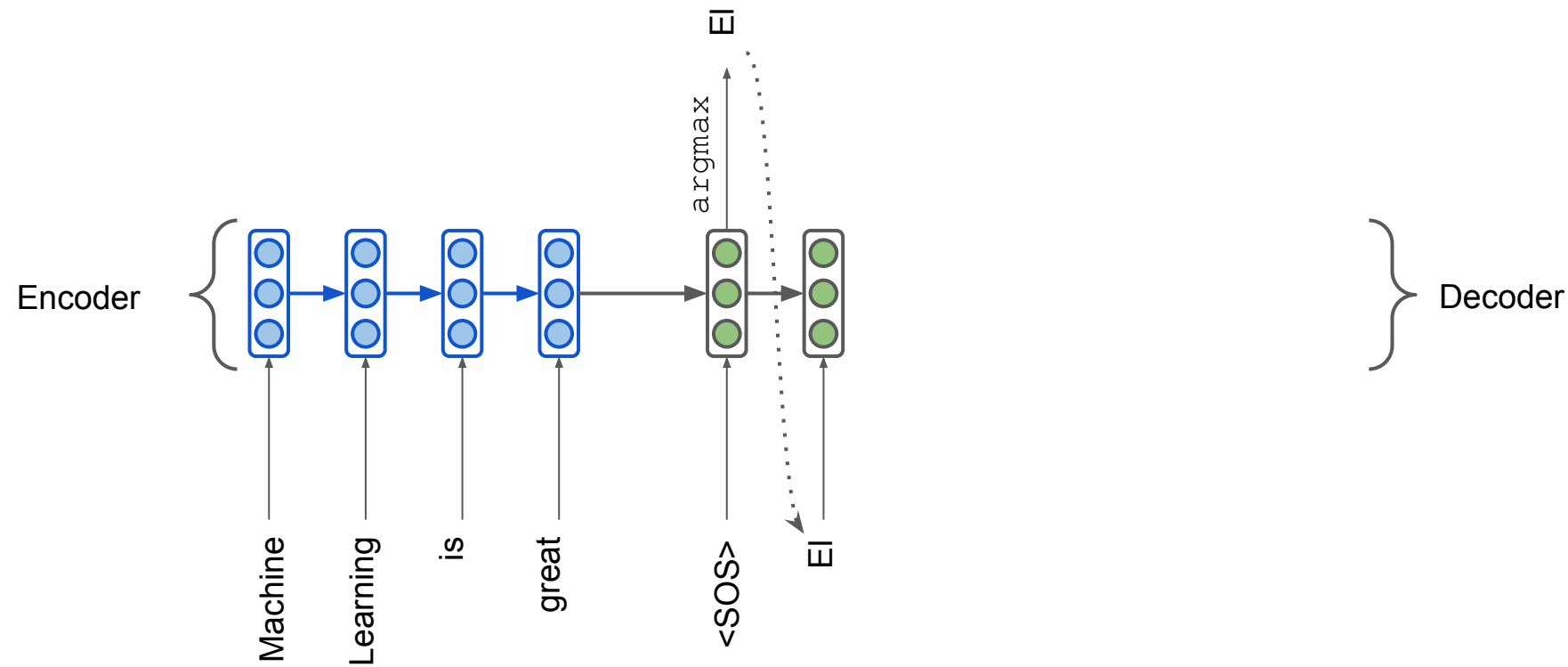




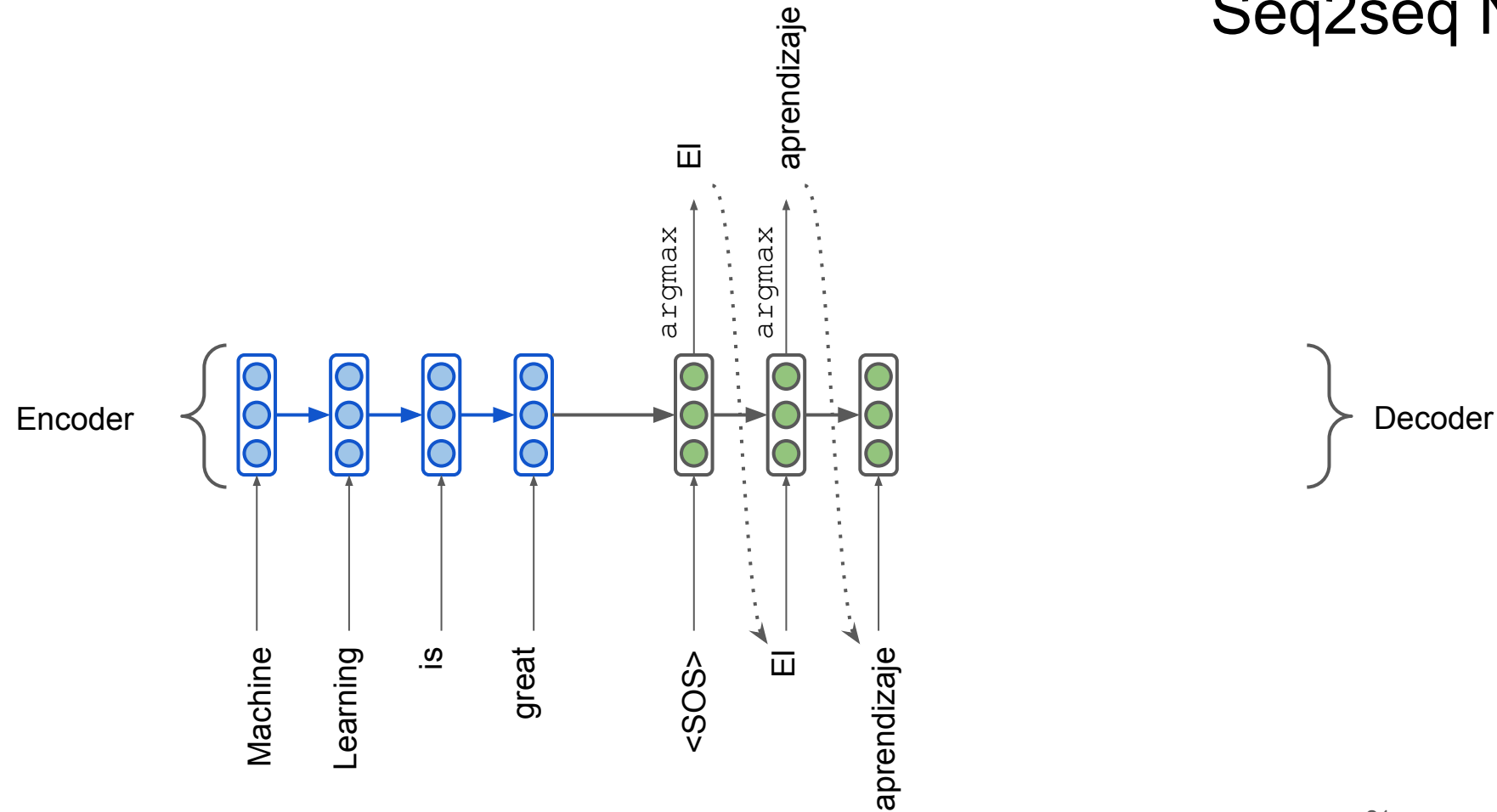
Forwarded as initial  
hidden state to decoder



# Seq2seq NMT

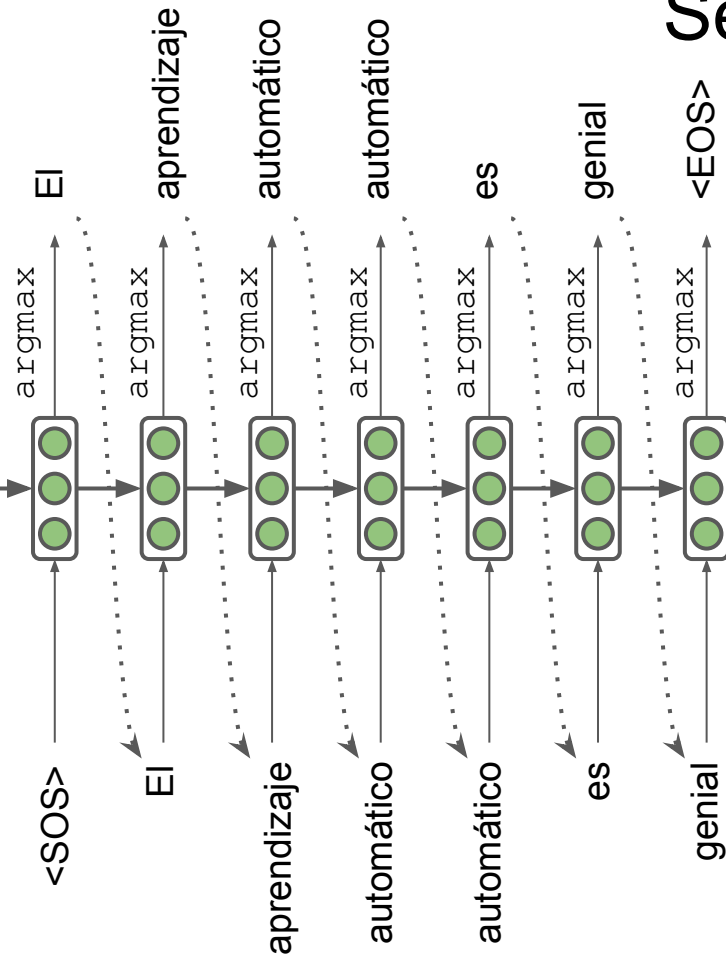
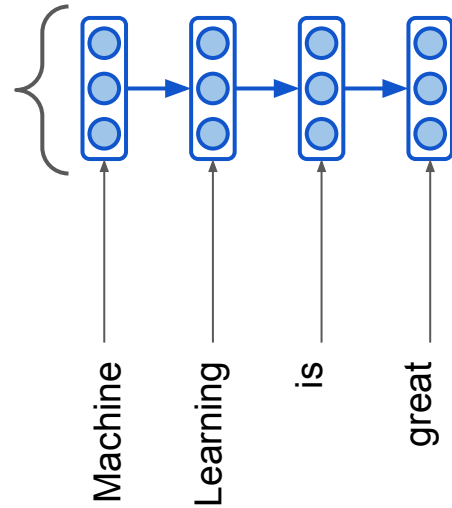


# Seq2seq NMT



# Seq2seq NMT

Encoder



Decoder

# NMT: how does it work?

- NMT directly calculates  $P(y|x)$ 
  - $y$  – target sentence,  $x$  – source sentence

$$P(y|x) = P(y_2|y_1, x)P(y_3|y_1, y_2, x) \dots \underbrace{P(y_T|y_1, y_2, \dots, x)}$$

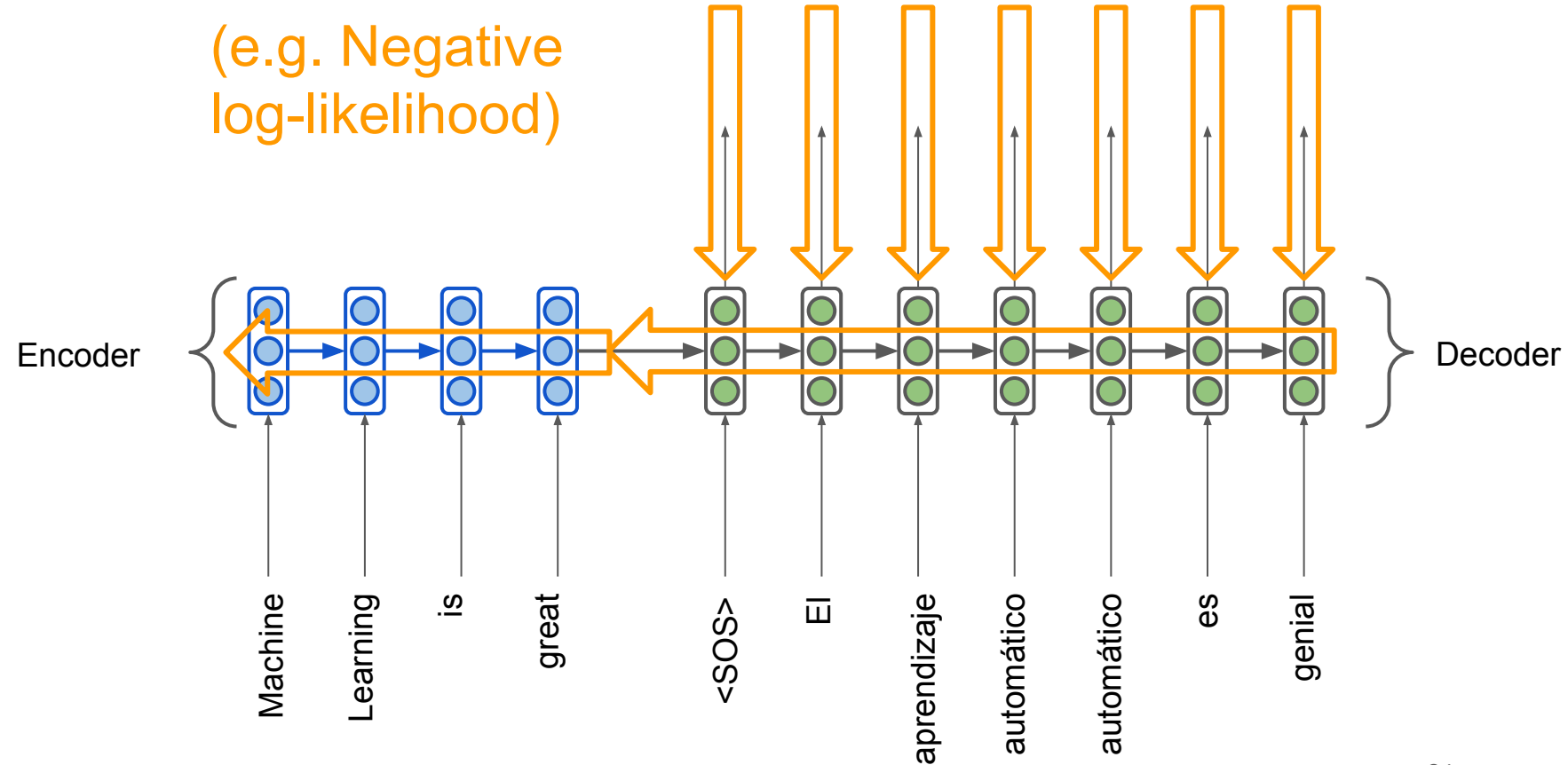
Probability of next word  
in target language



- To train it we need a huge parallel corpus.

# Seq2seq is trained end-to-end

Loss  
(e.g. Negative  
log-likelihood)



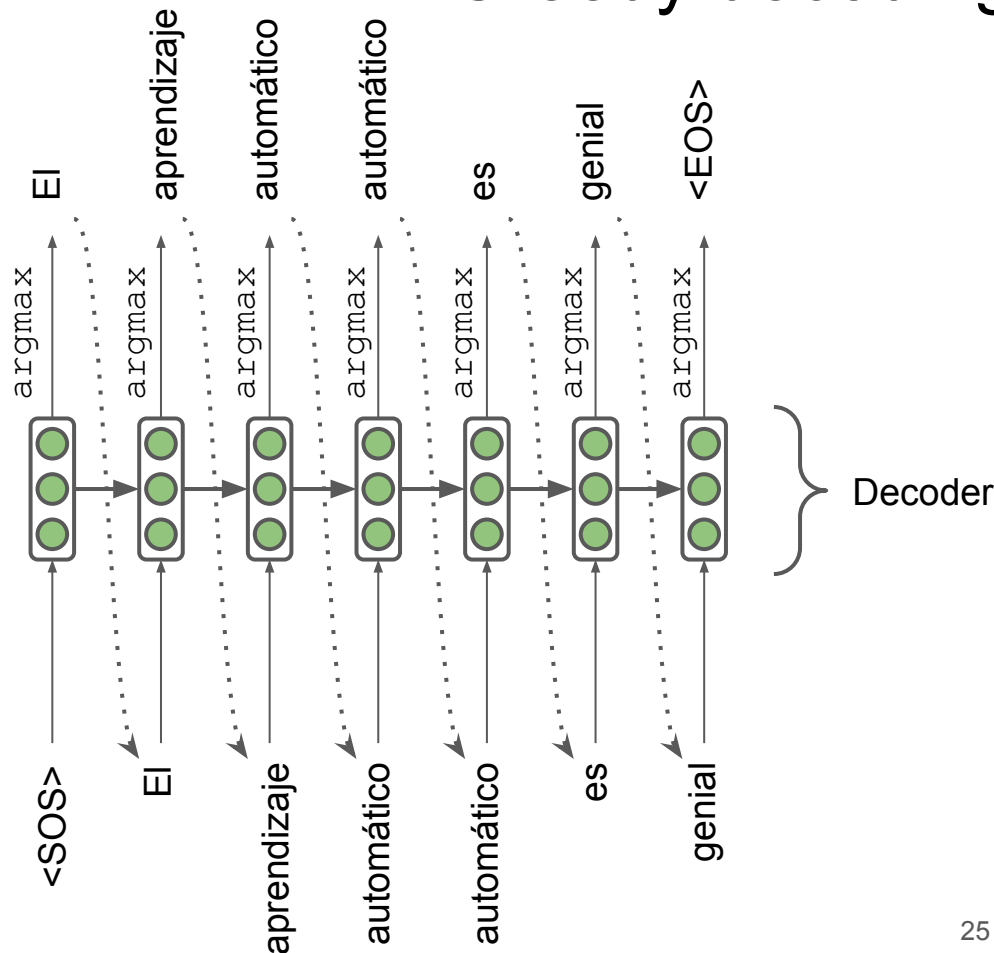


- Decoder predicts the most probable token (argmax) on each step
- The approach is **greedy**

**Any problems with it?**

**Any mistake is treated as input on the next step!**

## Greedy decoding



# Exhaustive search

- We want the translation that maximizes the likelihood:

$$P(y|x) = P(y_1|x) \prod_{t=2}^T P(y_t|y_1, \dots, y_{t-1}, x)$$

- We cannot compute all the possible sequences (exponential complexity)

# Beam search

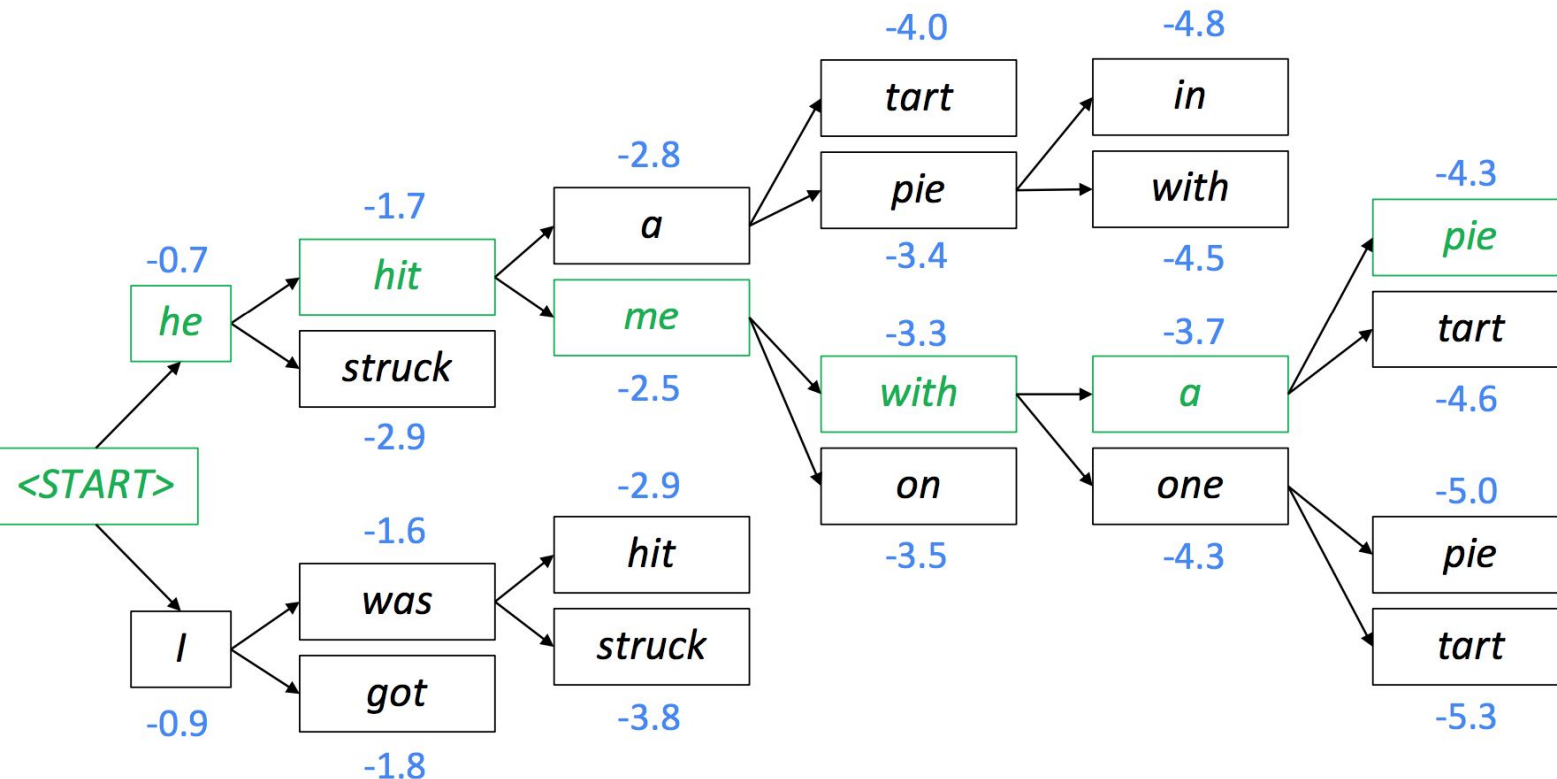
- On each step of decoder, keep track of the k most probable partial translations (which we call hypotheses)
- k is the beam size (in practice around 5 to 10)
- A hypothesis has a score which is its log probability:

$$\text{score}(y_1, \dots, y_t) = \log P_{\text{LM}}(y_1, \dots, y_t | x) = \sum_{i=1}^t \log P_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x)$$

- We search for high-scoring hypotheses, tracking top k on each step
- Beam search does not guarantee finding optimal solution

# Beam search decoding: example

Beam size =  $k = 2$ . Blue numbers =  $\text{score}(y_1, \dots, y_t) = \sum_{i=1}^t \log P_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x)$



## Beam search decoding: stopping criterion

- In **greedy decoding**, usually we decode until the model produces  $\langle \text{EOS} \rangle$  token
- In **beam search decoding**, different hypotheses may produce  $\langle \text{EOS} \rangle$  tokens on different timesteps
  - When a hypothesis produces  $\langle \text{EOS} \rangle$ , that hypothesis is complete.
  - Place it aside and continue exploring other hypotheses via beam search.
- Usually we continue beam search until:
  - We reach pre-defined timestep  $T$
  - We have at least  $n$  completed hypotheses

# Beam search decoding: finishing up

- How to select top one with highest score?
- Each hypothesis on our list has a score:

$$\text{score}(y_1, \dots, y_t) = \log P_{\text{LM}}(y_1, \dots, y_t | x) = \sum_{i=1}^t \log P_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x)$$

- **Problems?**

Longer hypotheses have lower scores

- **Fix:** Normalize by length. Use this to select top one instead:

$$\frac{1}{t} \sum_{i=1}^t \log P_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x)$$

# NMT: Quality Evaluation

BLEU (Bilingual Evaluation Understudy) compares the machine-written translation to human-written translation, and computes a similarity score based on:

- n-gram precision
- penalty for too-short system translations (brevity penalty)

$$BLEU = \text{brevity penalty} \cdot \left( \prod_{i=1}^n \text{precision}_i \right)^{1/n} \cdot 100\%$$

$$\text{brevity penalty} = \min \left( 1, \frac{\text{output length}}{\text{reference length}} \right)$$



BLEU (Bilingual Evaluation Understudy) compares the machine-written translation to human-written translation, and computes a similarity score based on:

- n-gram precision
- brevity penalty

SYSTEM A: Israeli officials responsibility of airport safety  
2-GRAM MATCH                      1-GRAM MATCH

REFERENCE: Israeli officials are responsible for airport security

SYSTEM B: airport security Israeli officials are responsible  
2-GRAM MATCH                      4-GRAM MATCH

Metric	System A	System B
precision (1gram)	3/6	6/6
precision (2gram)	1/5	4/5
precision (3gram)	0/4	2/4
precision (4gram)	0/3	1/3
brevity penalty	6/7	6/7
BLEU	0%	52%

$$BLEU = \text{brevity penalty} \cdot \left( \prod_{i=1}^n \text{precision}_i \right)^{1/n} \cdot 100\%$$

BLEU is imperfect:

- There are many valid ways to translate a sentence
- So a good translation may get a poor BLEU score just because of low n-gram overlap with the human translation

## Other ways to estimate translation quality

- **ROUGE** (Recall-Oriented Understudy for Gisting Evaluation)
- **METEOR** (Metric for Evaluation of Translation with Explicit ORdering)
  - Uses synonyms from WordNet
- **NIST** (of US National Institute of Standards and Technology)
  - More weight to rare n-grams, less punishment for short texts
- **TER**
  - Uses the number of changes that should be made to get to the reference translation

- Better performance
  - More fluent
  - Better use of context
  - Better use of phrase similarities
- A single neural network to be optimized end-to-end
  - No subcomponents to be individually optimized
- Requires much less human engineering effort
  - No feature engineering
  - Same method for all language pairs

# NMT: disadvantages

- NMT is less interpretable
  - Hard to debug
- NMT is difficult to control
  - For example, can't easily specify rules or guidelines for translation
  - Safety concerns!

# NMT: disadvantages



*Feedback*



# NMT: disadvantages



АНГЛИЙСКИЙ

Instead I spent friday evening cleaning the kitchen.

52 / 10000

РУССКИЙ



Вместо этого я провела вечер пятницы, убирая кухню.



АНГЛИЙСКИЙ

Instead I spent friday evening drinking with friends.

53 / 10000

РУССКИЙ



Вместо этого я провел вечер пятницы, выпивая с друзьями.

## NMT: disadvantages

Somali ▾  
Translate from Irish

ag ag ag ag ag ag ag ag ag ag ag ag  
ag ag ag ag ag ag ag ag ag ag ag ag  
ag Edit

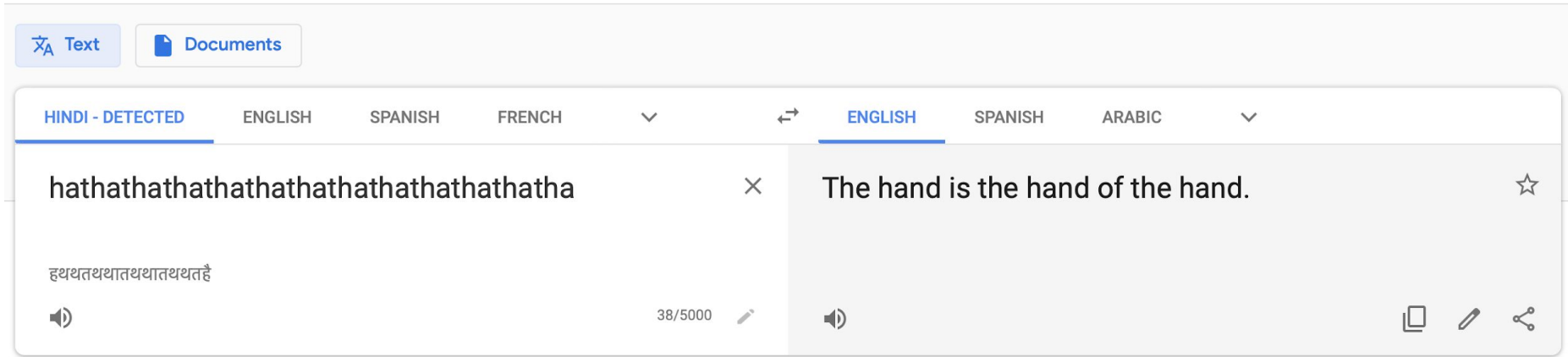
English ▾

As the name of the LORD was written  
in the Hebrew language, it was written  
in the language of the Hebrew Nation

### Feedback



# NMT: disadvantages



# Is Machine Translation solved?

- Many difficulties remain:
  - Out-of-vocabulary words
  - Domain mismatch between train and test data
  - Maintaining context over long texts
  - Low-resource language pairs (no big parallel corpora)

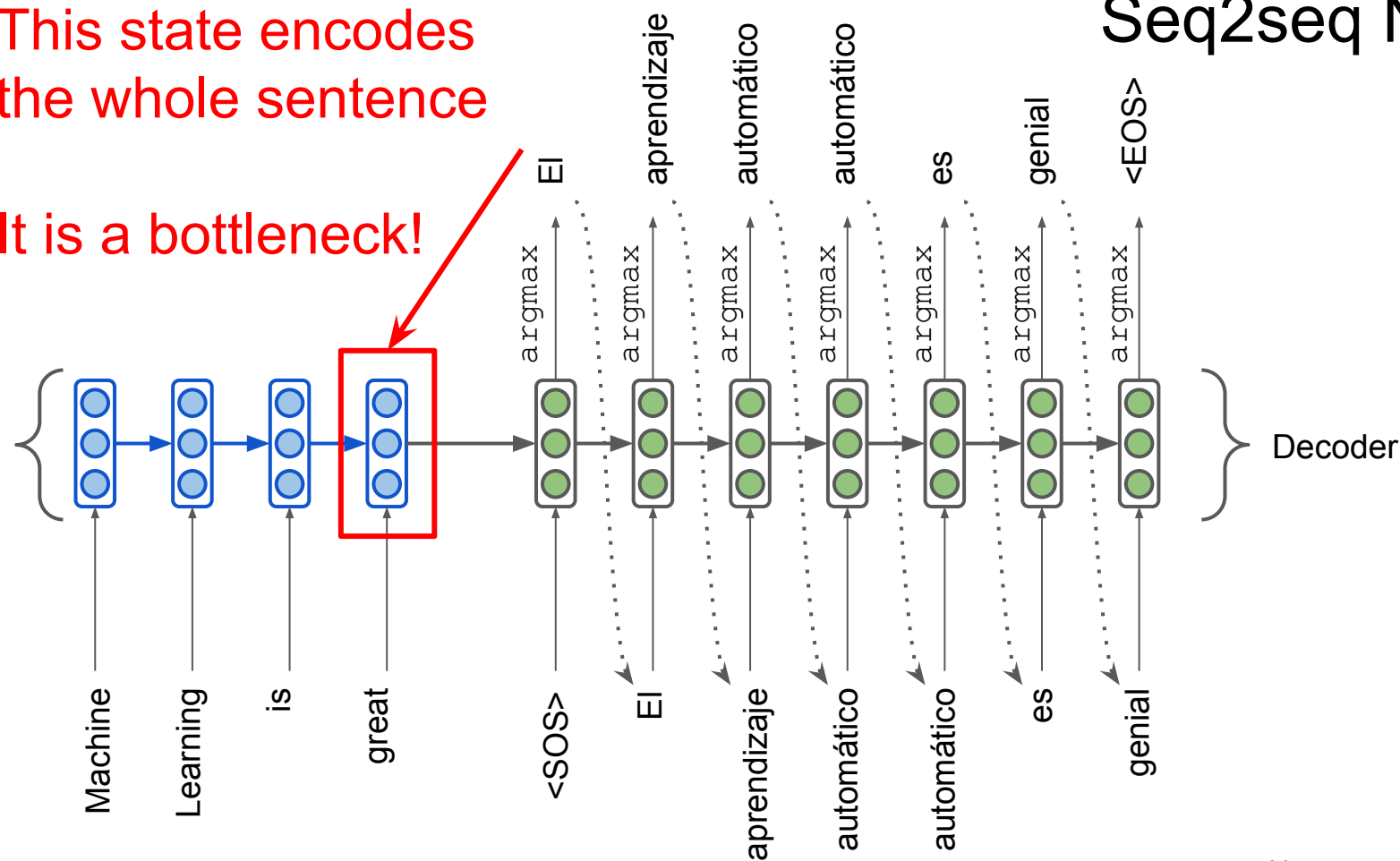
# Attention

# Seq2seq NMT

This state encodes the whole sentence

It is a bottleneck!

Encoder

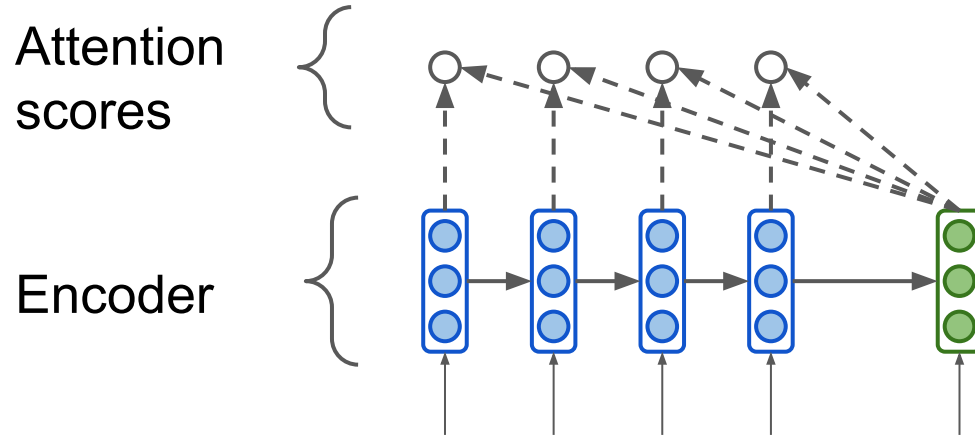


## Main idea:

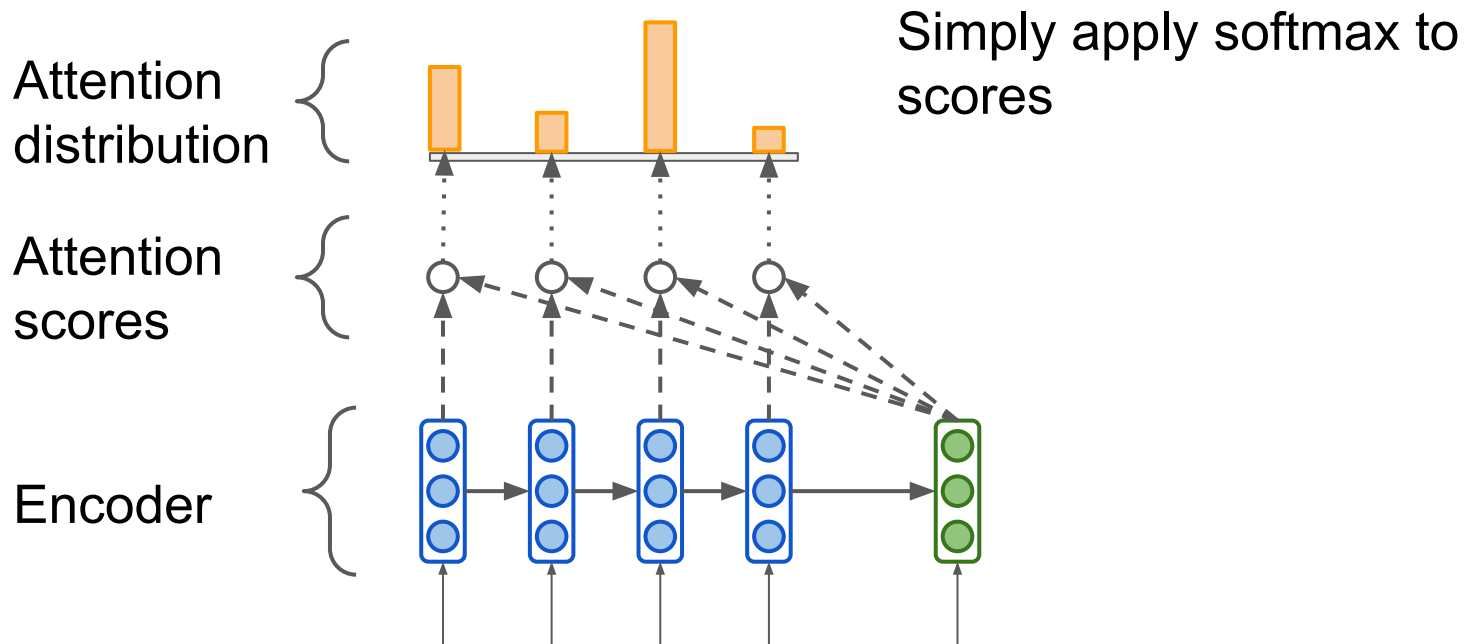
on each step of the **decoder**, use **direct connection to the encoder** to focus on a particular part of the source sequence



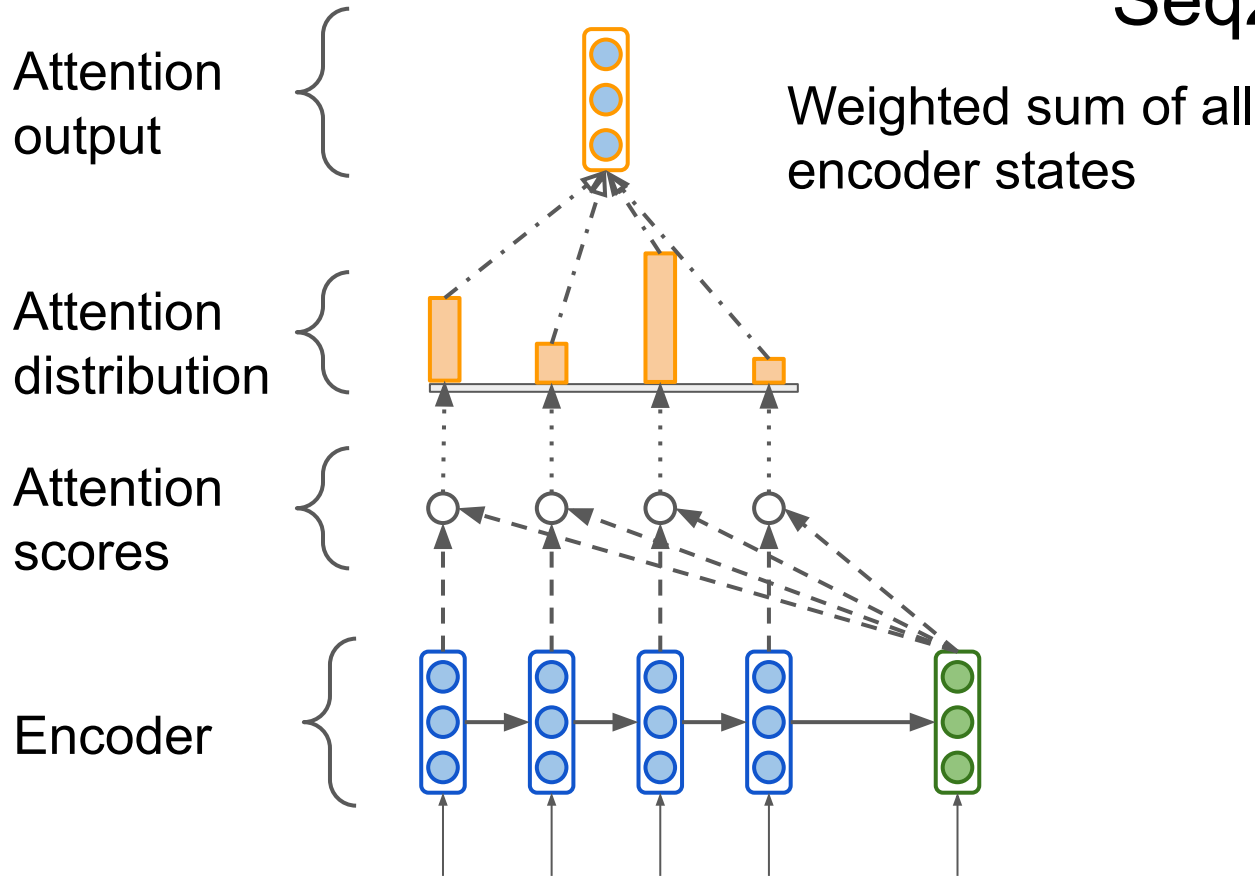
# Seq2seq with attention



# Seq2seq with attention

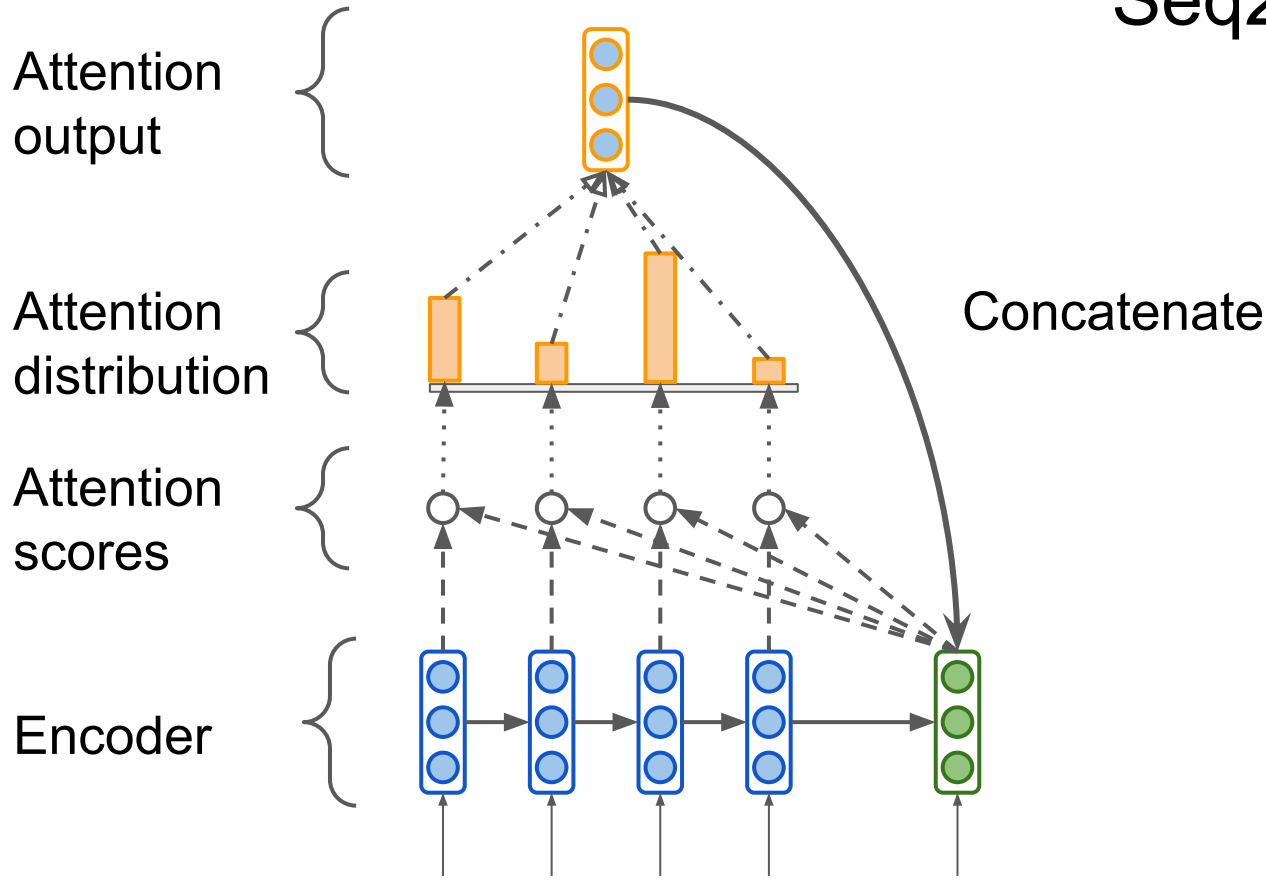


# Seq2seq with attention

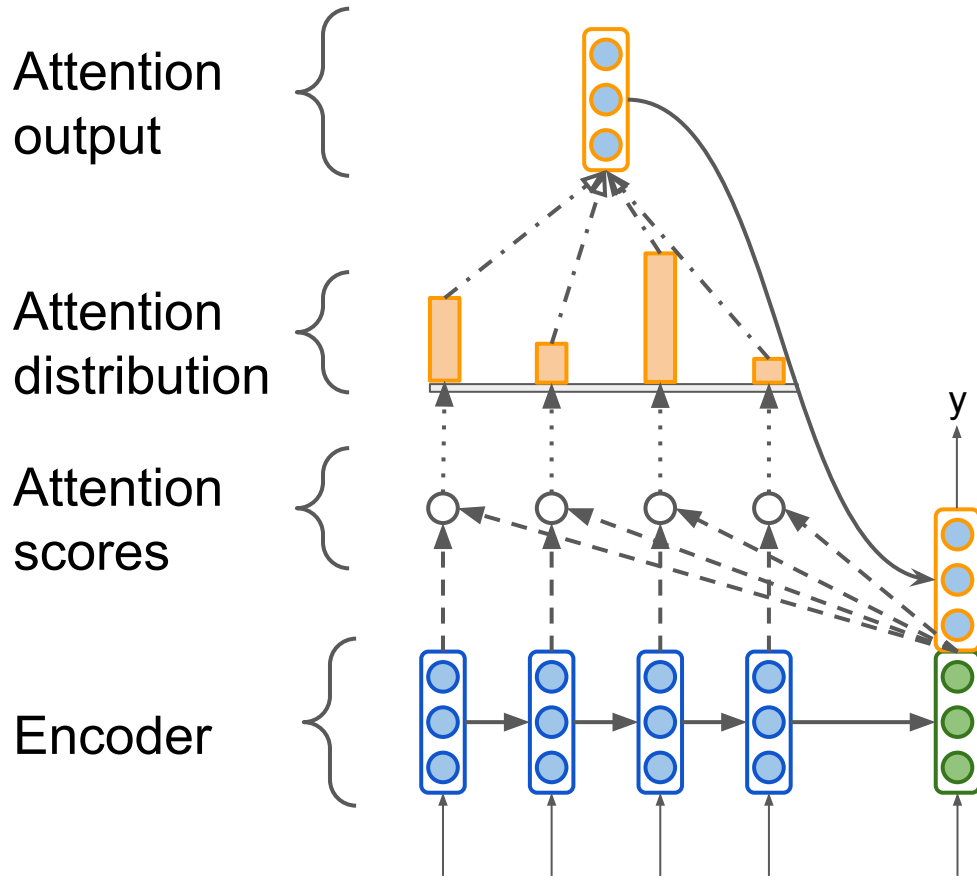




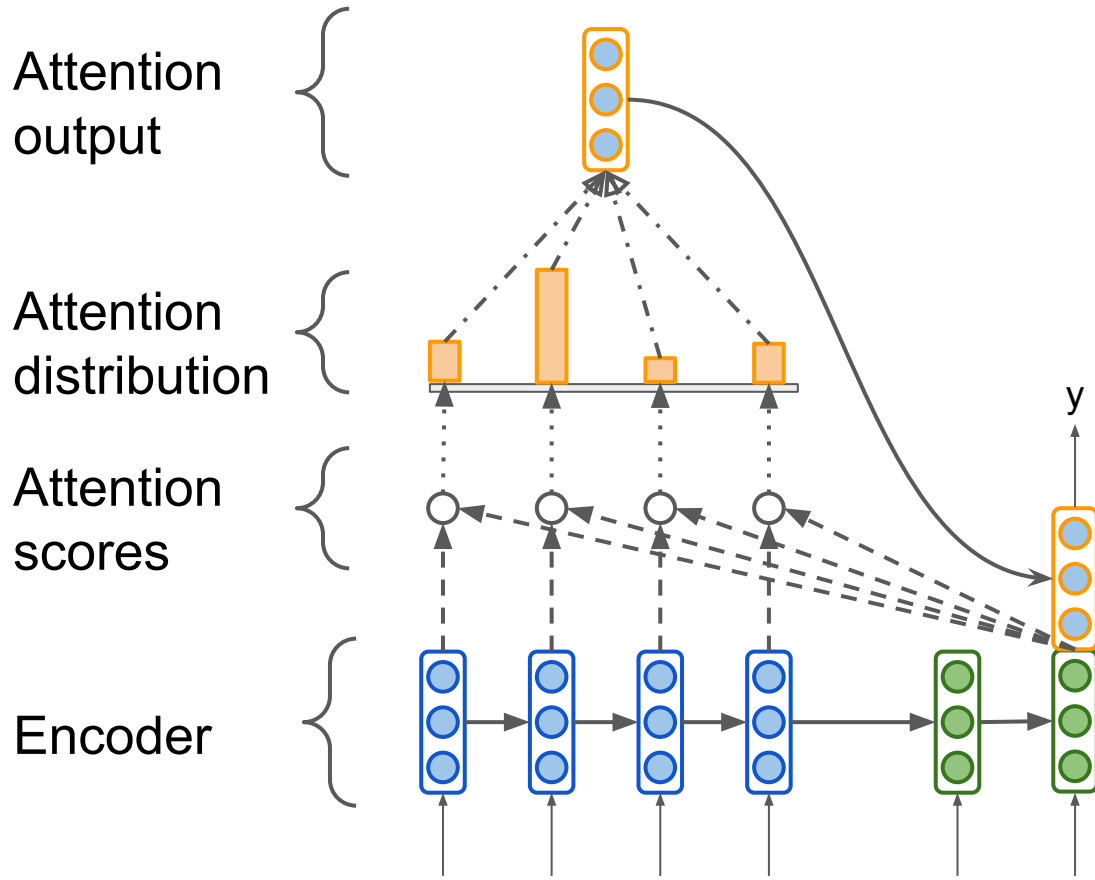
# Seq2seq with attention



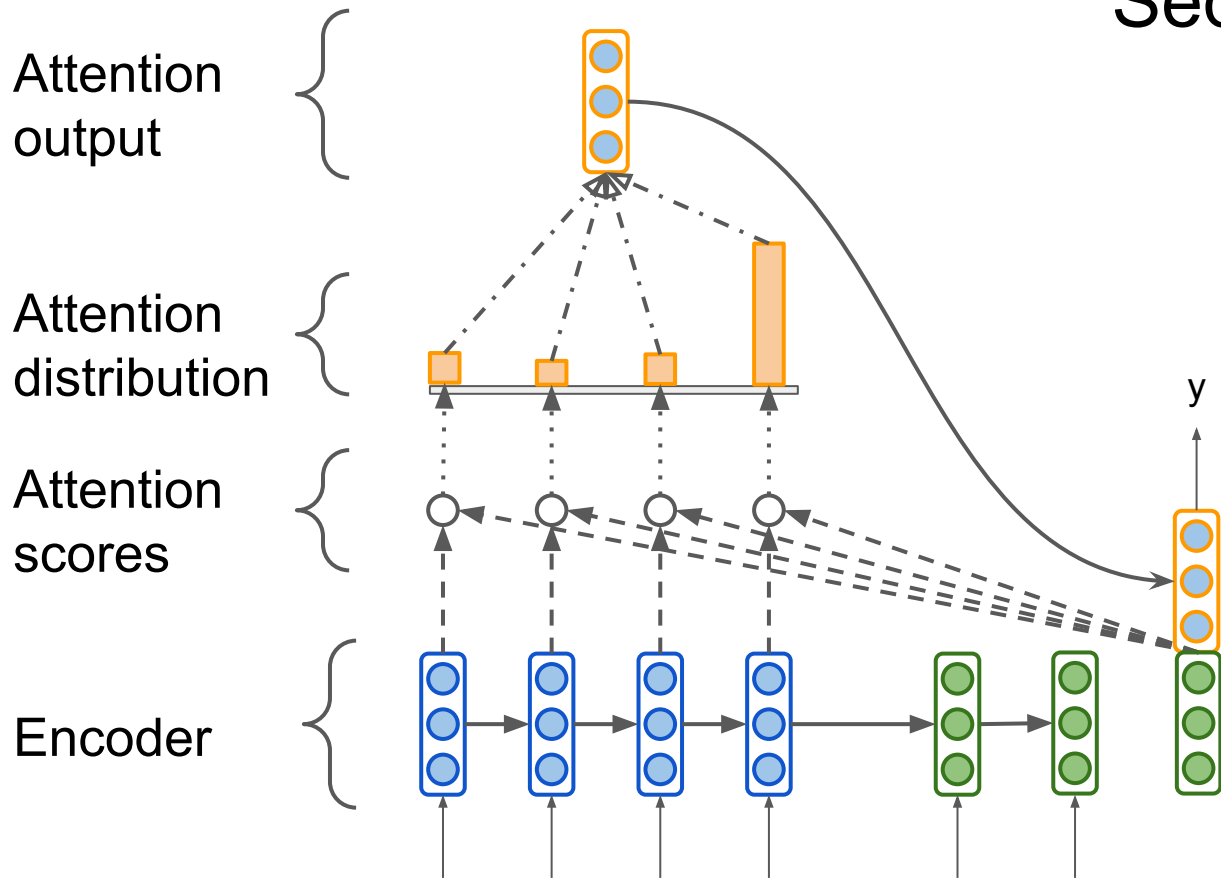
# Seq2seq with attention



# Seq2seq with attention



# Seq2seq with attention



## Attention in equations

Denote encoder hidden states  $\mathbf{h}_1, \dots, \mathbf{h}_N \in \mathbb{R}^k$   
and decoder hidden state at time step  $t$   $\mathbf{s}_t \in \mathbb{R}^k$

The attention scores  $\mathbf{e}^t$  can be computed as dot product

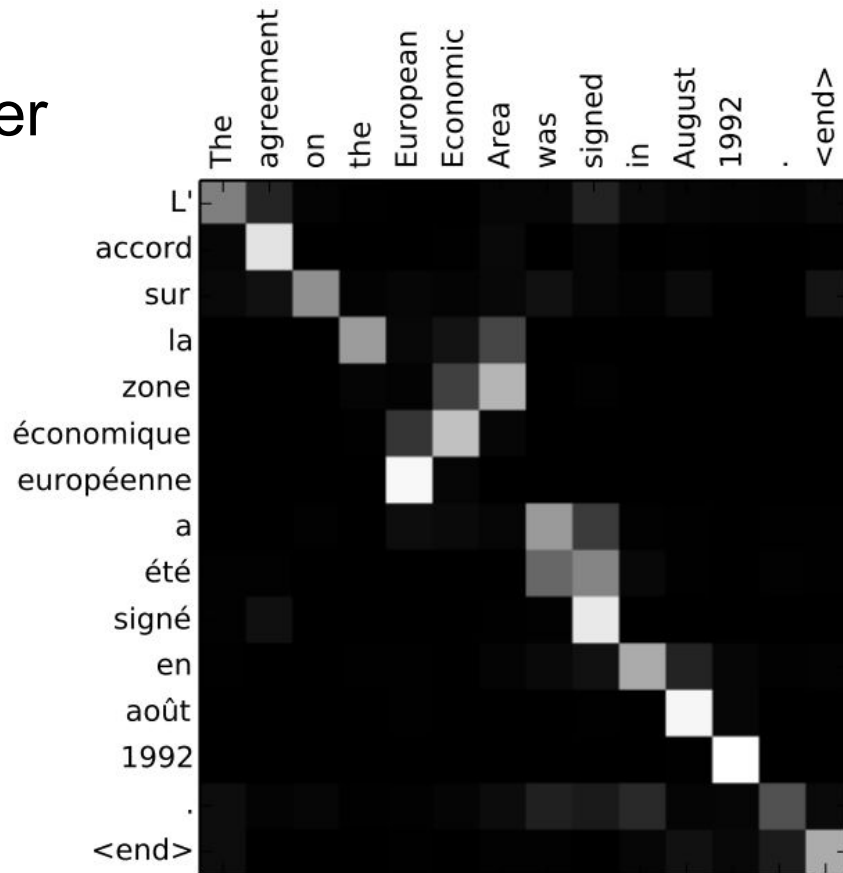
$$\mathbf{e}^t = [\mathbf{s}^T \mathbf{h}_1, \dots, \mathbf{s}^T \mathbf{h}_N]$$

Then the attention vector is a linear combination of encoder states

$$\mathbf{a}_t = \sum_{i=1}^N \alpha_i^t \mathbf{h}_i \in \mathbb{R}^k, \text{ where } \boldsymbol{\alpha}_t = \text{softmax}(\mathbf{e}_t)$$

# Attention provides interpretability

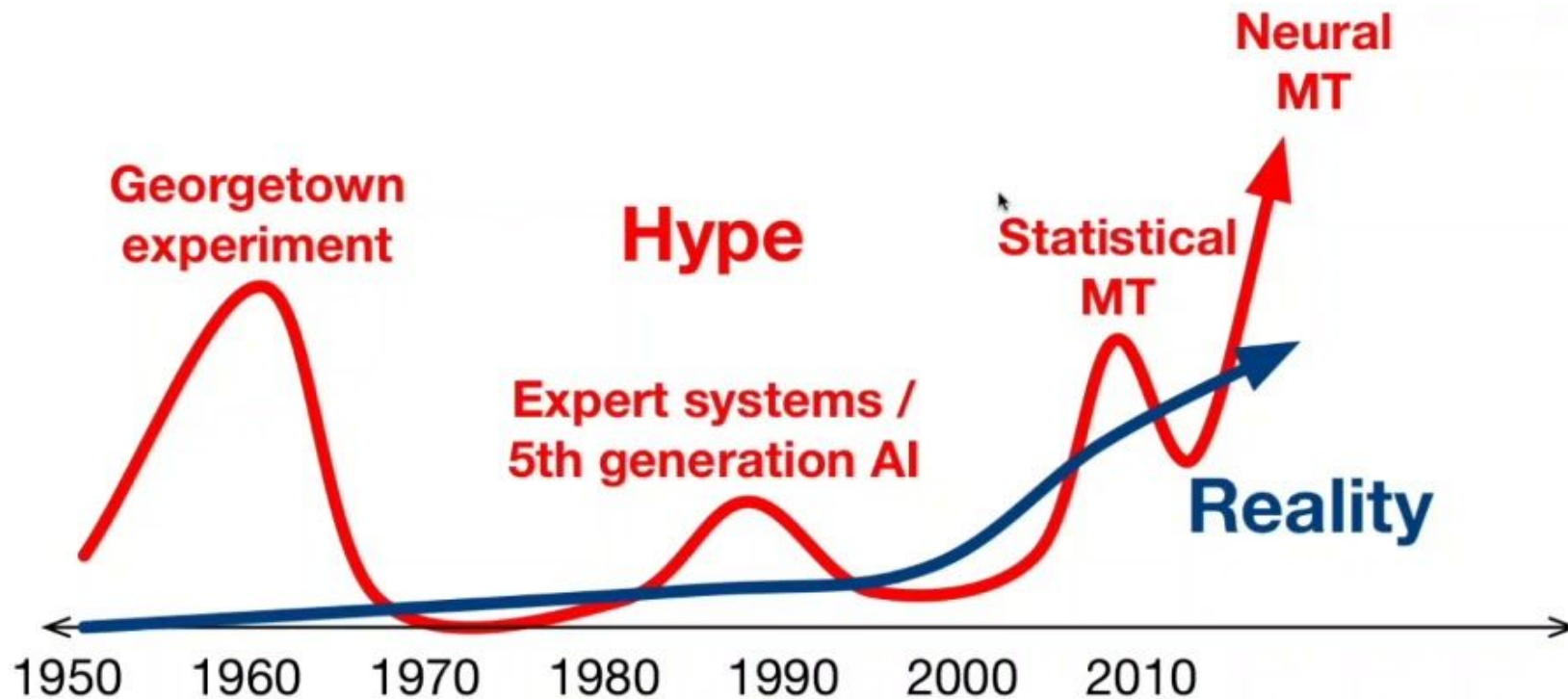
- We may see what the decoder was focusing on
- We get word alignment for free!



# Attention variants

- Basic dot-product (the one discussed before):  $e_i = \mathbf{s}^T \mathbf{h}_i \in \mathbb{R}$
- Multiplicative attention:  $e_i = \mathbf{s}^T \mathbf{W} \mathbf{h}_i \in \mathbb{R}$ 
  - $\mathbf{W} \in \mathbb{R}^{d_2 \times d_1}$  - weight matrix
- Additive attention:  $e_i = \mathbf{v}^T \tanh(\mathbf{W}_1 \mathbf{h}_i + \mathbf{W}_2 \mathbf{s}) \in \mathbb{R}$ 
  - $\mathbf{W}_1 \in \mathbb{R}^{d_3 \times d_1}, \mathbf{W}_2 \in \mathbb{R}^{d_3 \times d_2}$  - weight matrices
  - $\mathbf{v} \in \mathbb{R}^{d_3}$  - weight vector

# Summary



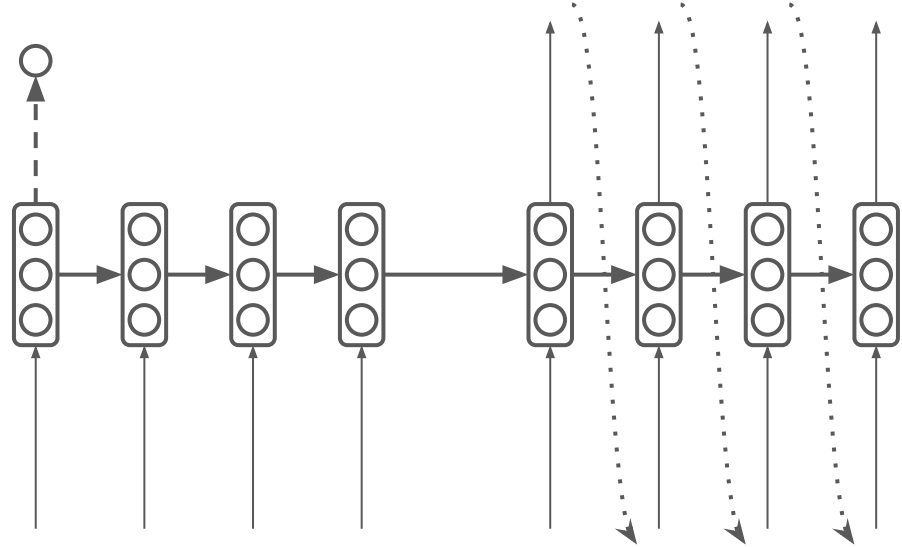


# Summary

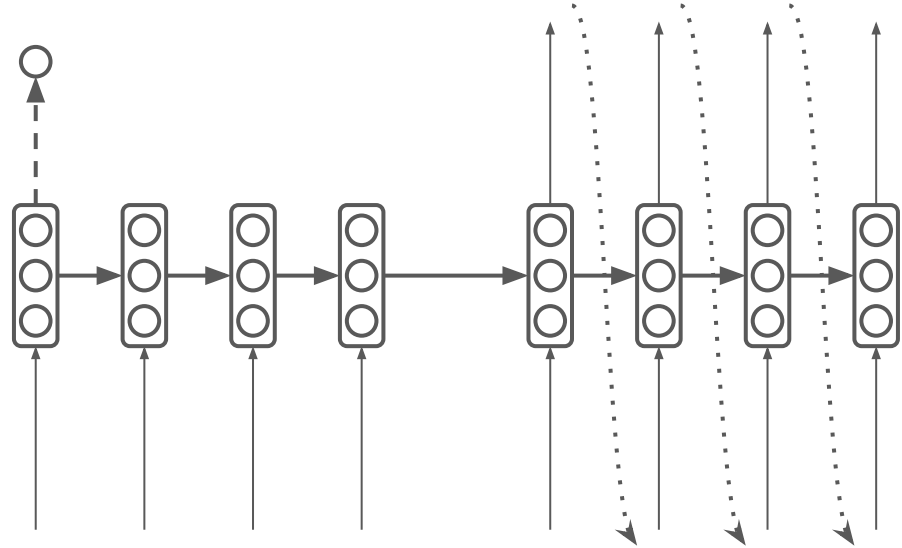
- Seq2seq is an architecture for NMT (2 RNNs)
- Attention is a way to focus on particular parts of the input



# Seq2seq with attention



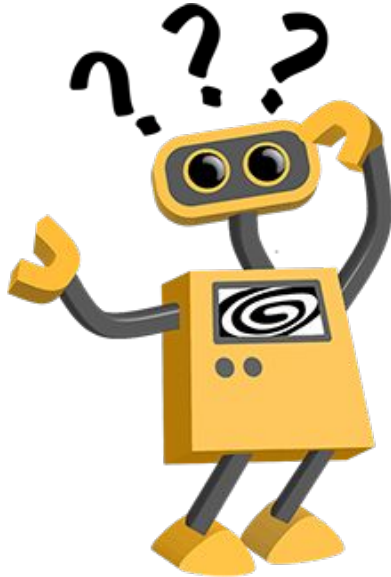
# Seq2seq with attention



# Machine Translation

I'M GOING TO THE THEATER = ICH GEHE INS THEATER

I'M GOING TO THE CINEMA <sup>???</sup> = ICH GEHE INS KINO



KINO

# Quality evaluation: Perplexity

$$PP(W) = P(w_1, w_2, \dots, w_N)^{-\frac{1}{N}} = \sqrt[N]{\frac{1}{P(w_1, w_2, \dots, w_N)}} = \sqrt[N]{\frac{1}{\prod_{i=1}^N P(w_i | w_1, \dots, w_{i-1})}}$$

# WER (Word Error Rate)

$$WER = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C}$$

- S is the number of substitutions,
- D is the number of deletions,
- I is the number of insertions,
- C is the number of correct words,
- N is the number of words in the reference ( $N = S + D + C$ )

- **ROUGE** Recall-Oriented Understudy for Gisting Evaluation
- Recall in the context of ROUGE means how much of the reference summary is the system summary recovering or capturing
- **BLEU** is focusing on **precision**:
  - $\text{overlapping\_words} / \text{total\_words\_in\_system\_summary}$
- **ROUGE** is focusing on **recall**:
  - $\text{overlapping\_words} / \text{total\_words\_in\_reference\_summary}$

## **ROUGE** - Recall-Oriented Understudy for Gisting Evaluation

- **ROUGE-N**: Overlap of N-grams between the system and reference summaries.
- **ROUGE-L**: Longest Common Subsequence (LCS) based statistics. Longest common subsequence problem takes into account sentence level structure similarity naturally and identifies longest co-occurring in sequence n-grams automatically.
- **ROUGE-W**: Weighted LCS-based statistics
- etc.