



# Deep Learning for Vision & Language

Natural Language Processing: Single-Head and Multi-Head Attention



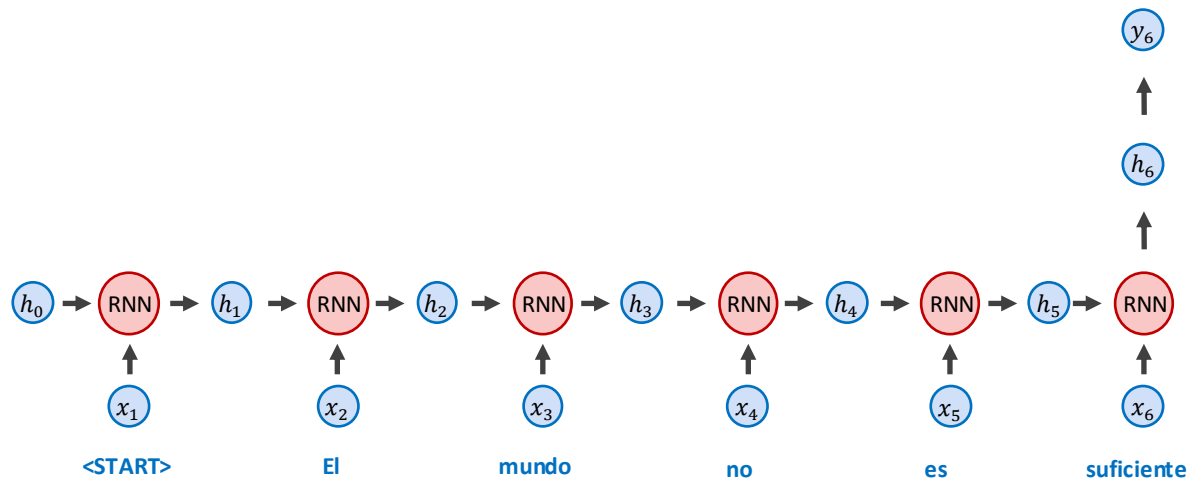
RICE UNIVERSITY



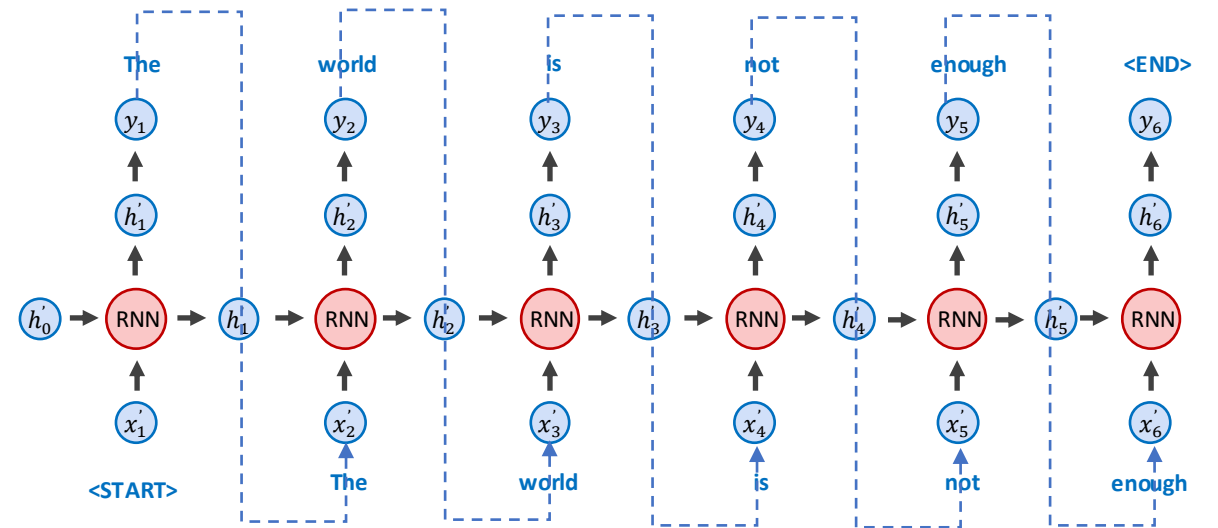
# RNNs – Sequence to score prediction

Classify

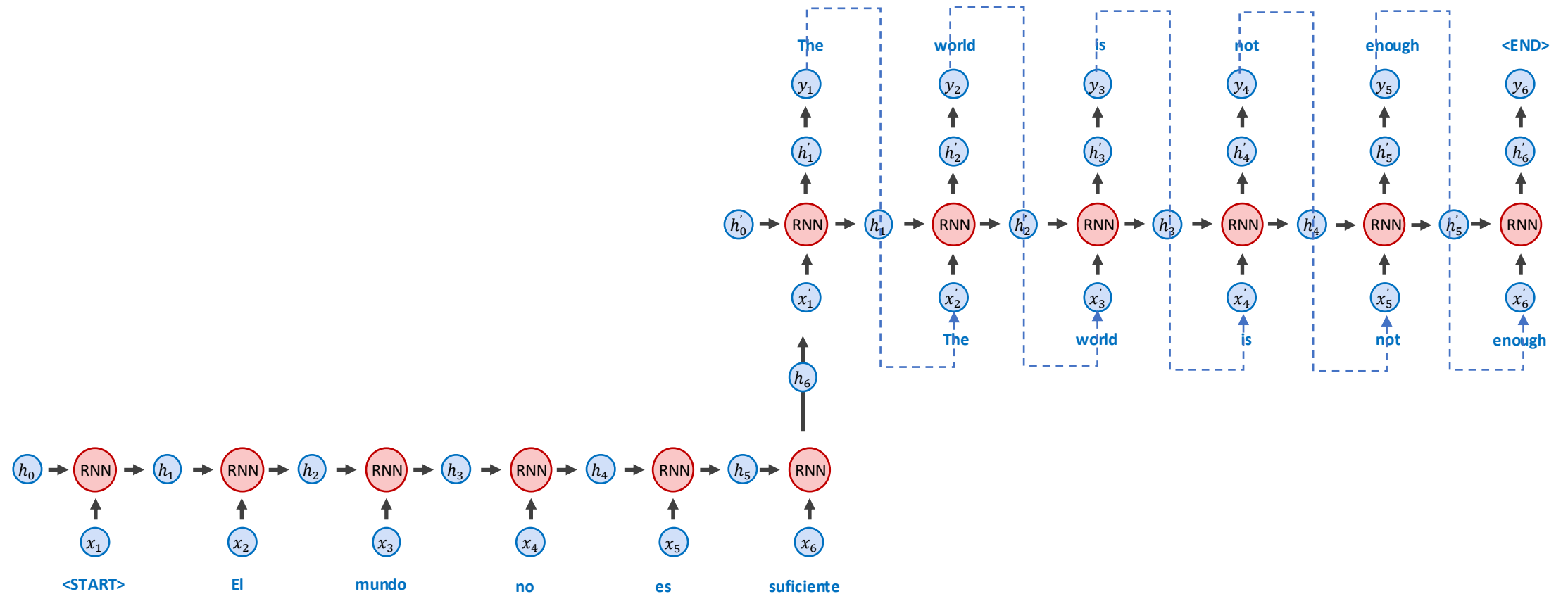
[English, German, Swiss German, Gaelic, Dutch, Afrikaans, Luxembourgish, Limburgish, other]



# RNNs for Text Generation (Auto-regressive)

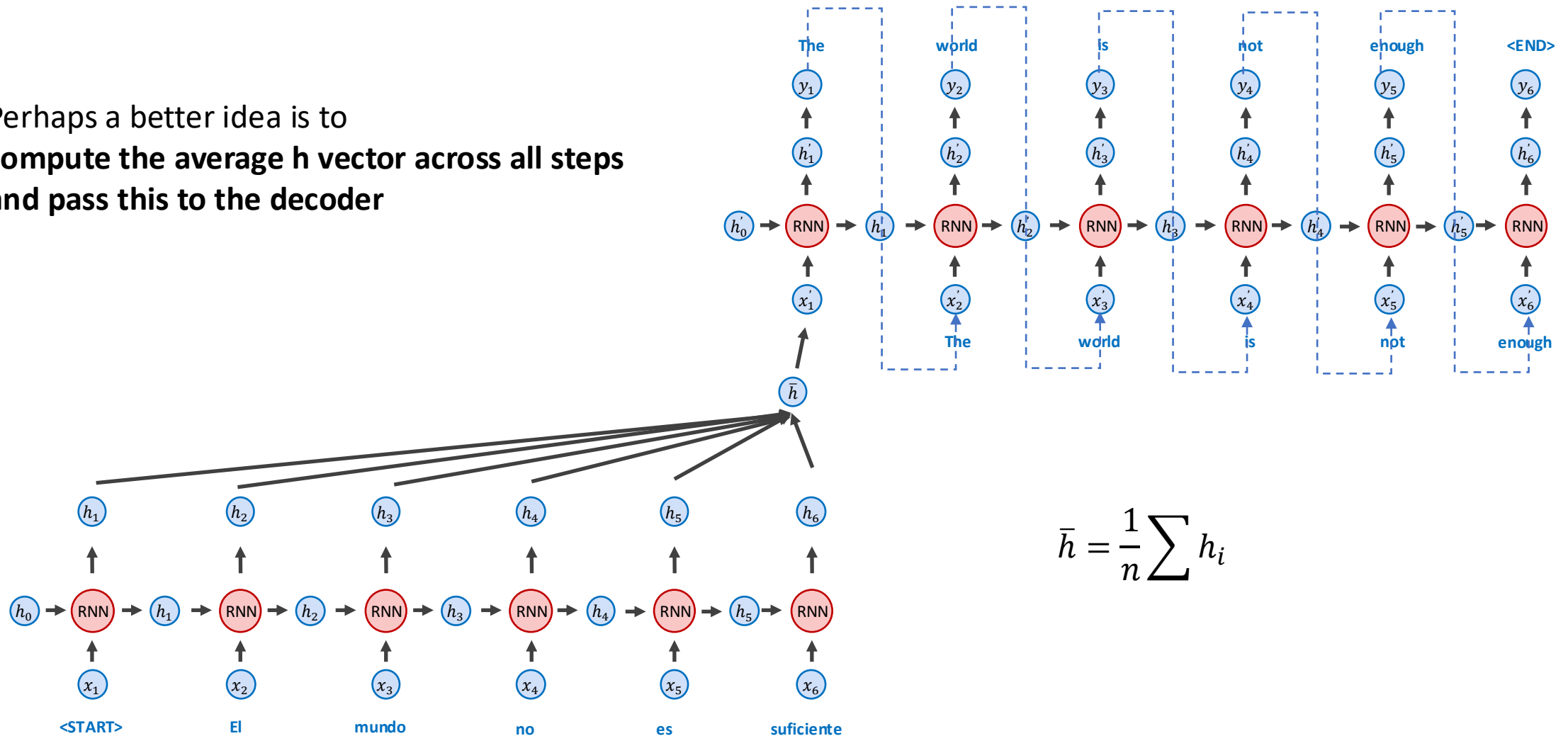


# RNNs for Machine Translation Seq-to-Seq



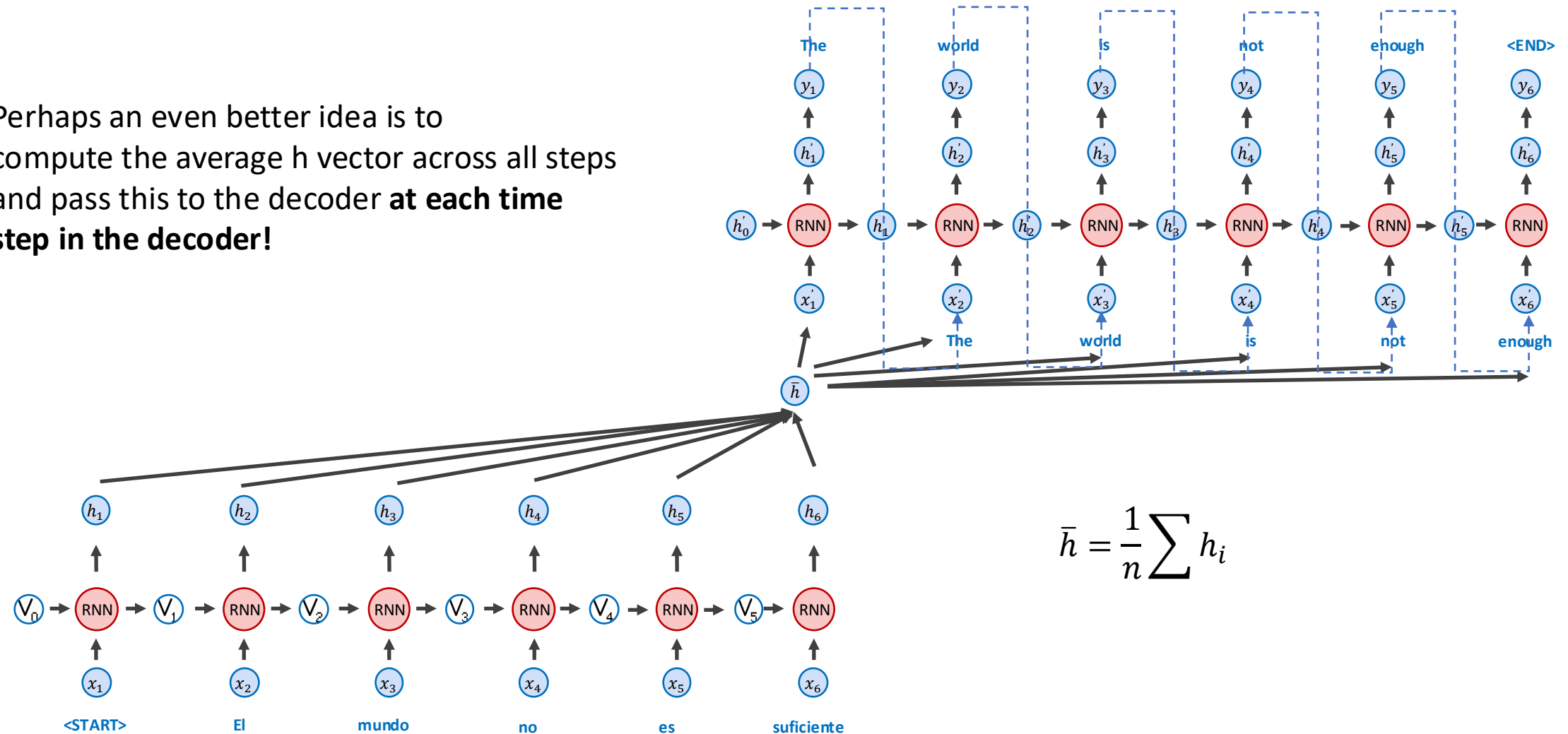
# RNNs for Machine Translation Seq-to-Seq

Perhaps a better idea is to **compute the average h vector across all steps and pass this to the decoder**



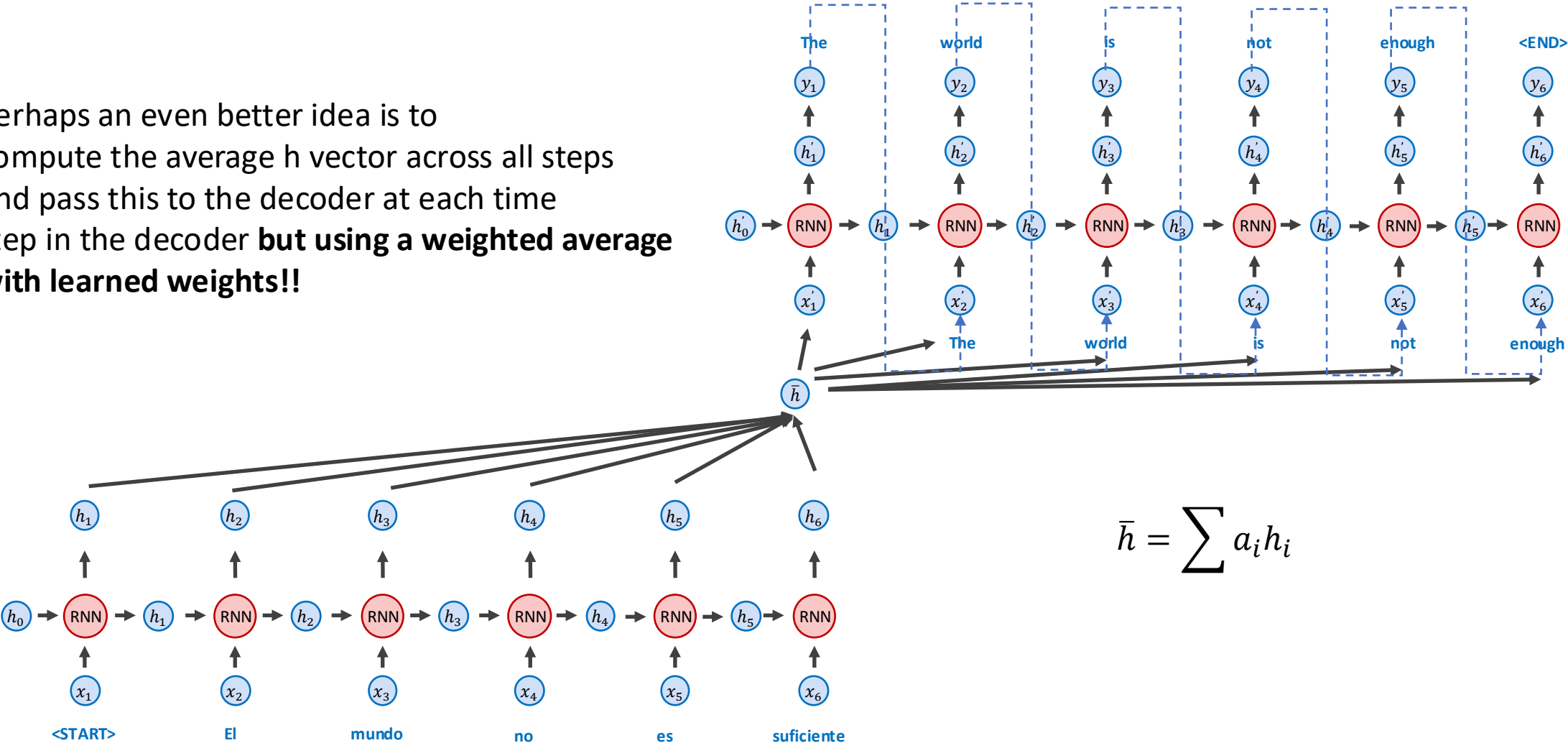
# RNNs for Machine Translation Seq-to-Seq

Perhaps an even better idea is to compute the average  $h$  vector across all steps and pass this to the decoder **at each time step in the decoder!**



# RNNs for Machine Translation Seq-to-Seq

Perhaps an even better idea is to compute the average  $h$  vector across all steps and pass this to the decoder at each time step in the decoder **but using a weighted average with learned weights!!**

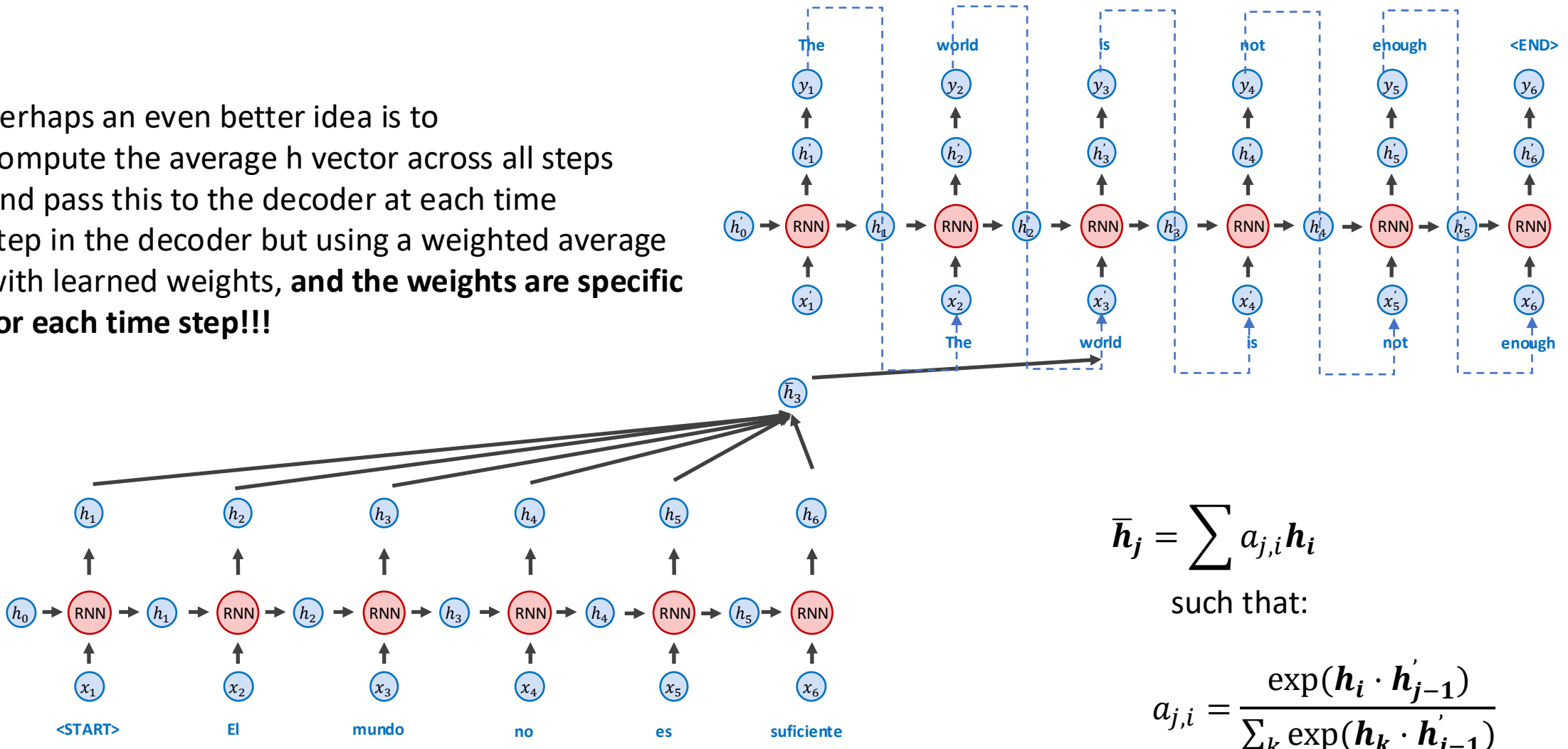


$$\bar{h} = \sum a_i h_i$$

# RNNs for Machine Translation Seq-to-Seq

Perhaps an even better idea is to compute the average  $h$  vector across all steps and pass this to the decoder at each time step in the decoder but using a weighted average with learned weights, **and the weights are specific for each time step!!!**

Only showing the third time step encoder-decoder connection



$$\bar{h}_j = \sum a_{j,i} h_i$$

such that:

$$a_{j,i} = \frac{\exp(h_i \cdot h'_{j-1})}{\sum_k \exp(h_k \cdot h'_{j-1})}$$

# NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

**Dzmitry Bahdanau**

Jacobs University Bremen, Germany

**KyungHyun Cho**   **Yoshua Bengio\***

Université de Montréal

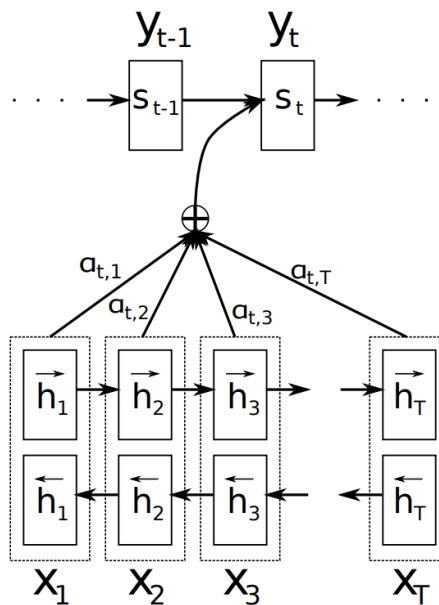


Figure 1: The graphical illustration of the proposed model trying to generate the  $t$ -th target word  $y_t$  given a source sentence  $(x_1, x_2, \dots, x_T)$ .

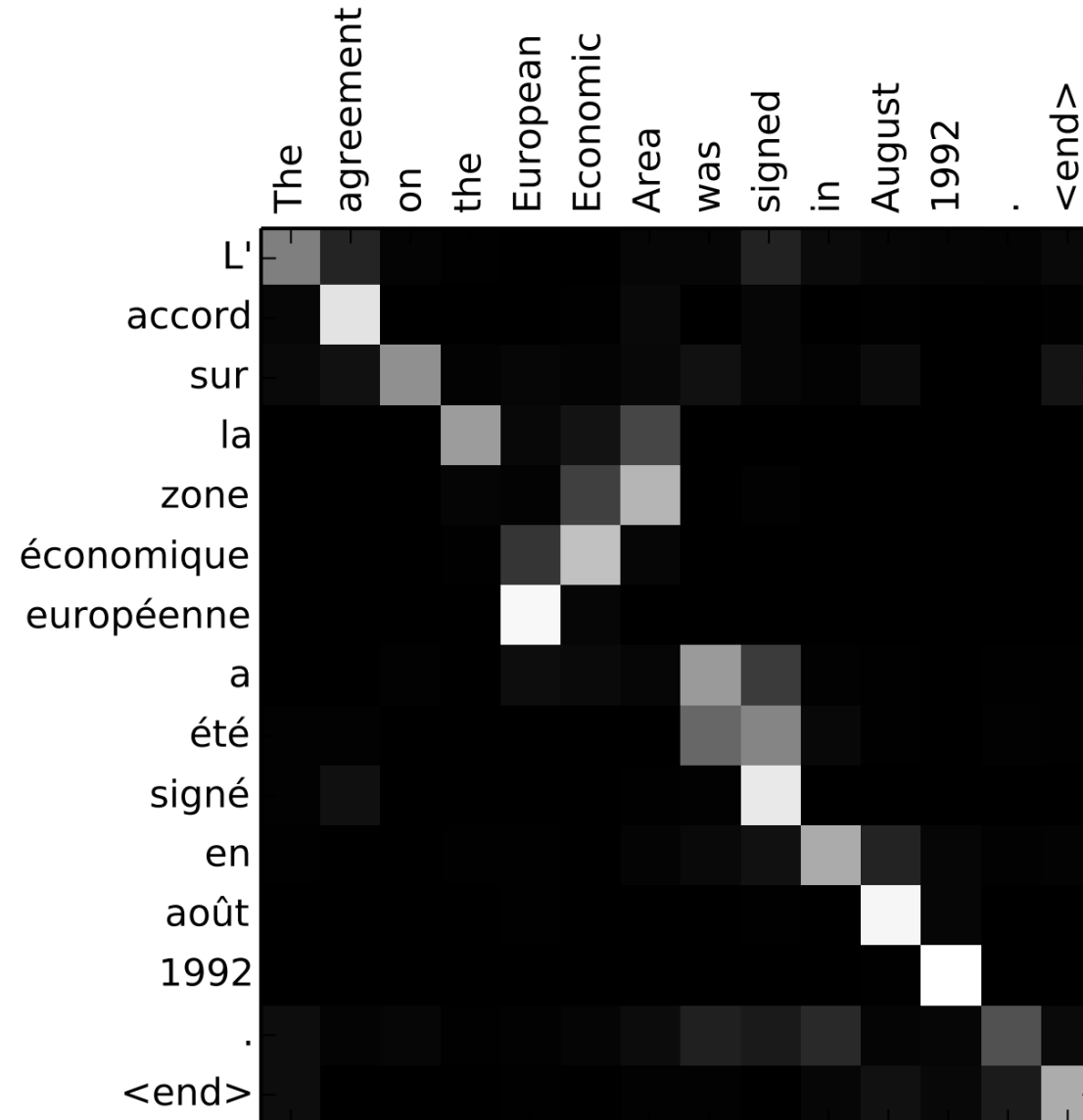
Let's take a look at one of the first papers introducing this idea.

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$$

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

$$e_{ij} = a(s_{i-1}, h_j)$$

# Let's look at the Attention weights



# Transformers: 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  
lukaszkaizer@google.com

**Illia Polosukhin\* ‡**  
illia.polosukhin@gmail.com

# Transformers: 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

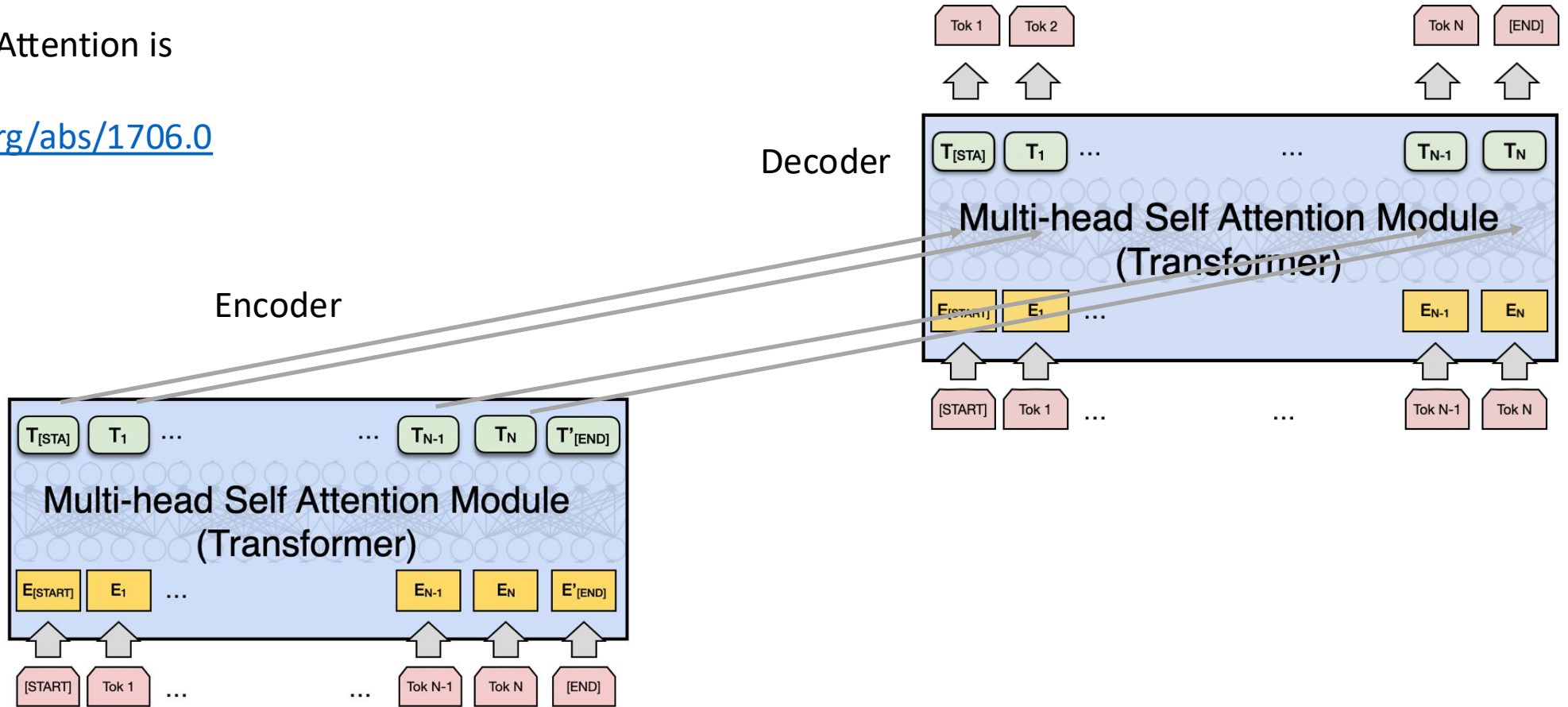
**Łukasz Kaiser\***  
Google Brain  
lukaszkaizer@google.com

**Illia Polosukhin\* ‡**  
illia.polosukhin@gmail.com

# Attention is All you Need (no RNNs)

Vaswani et al. Attention is all you need

<https://arxiv.org/abs/1706.03762>



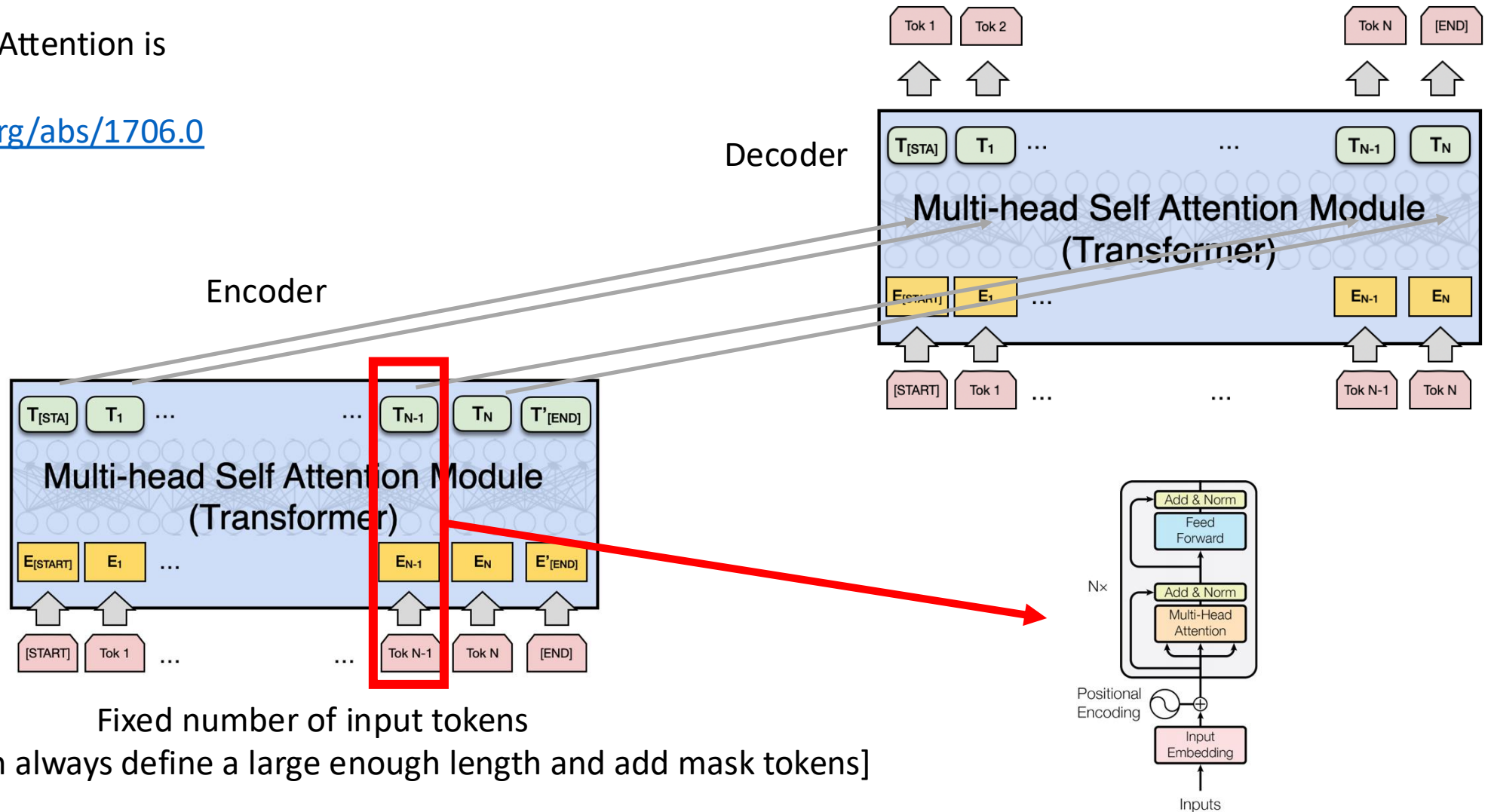
Fixed number of input tokens

[but hey! we can always define a large enough length and add mask tokens]

# Attention is All you Need (no RNNs)

Vaswani et al. Attention is all you need

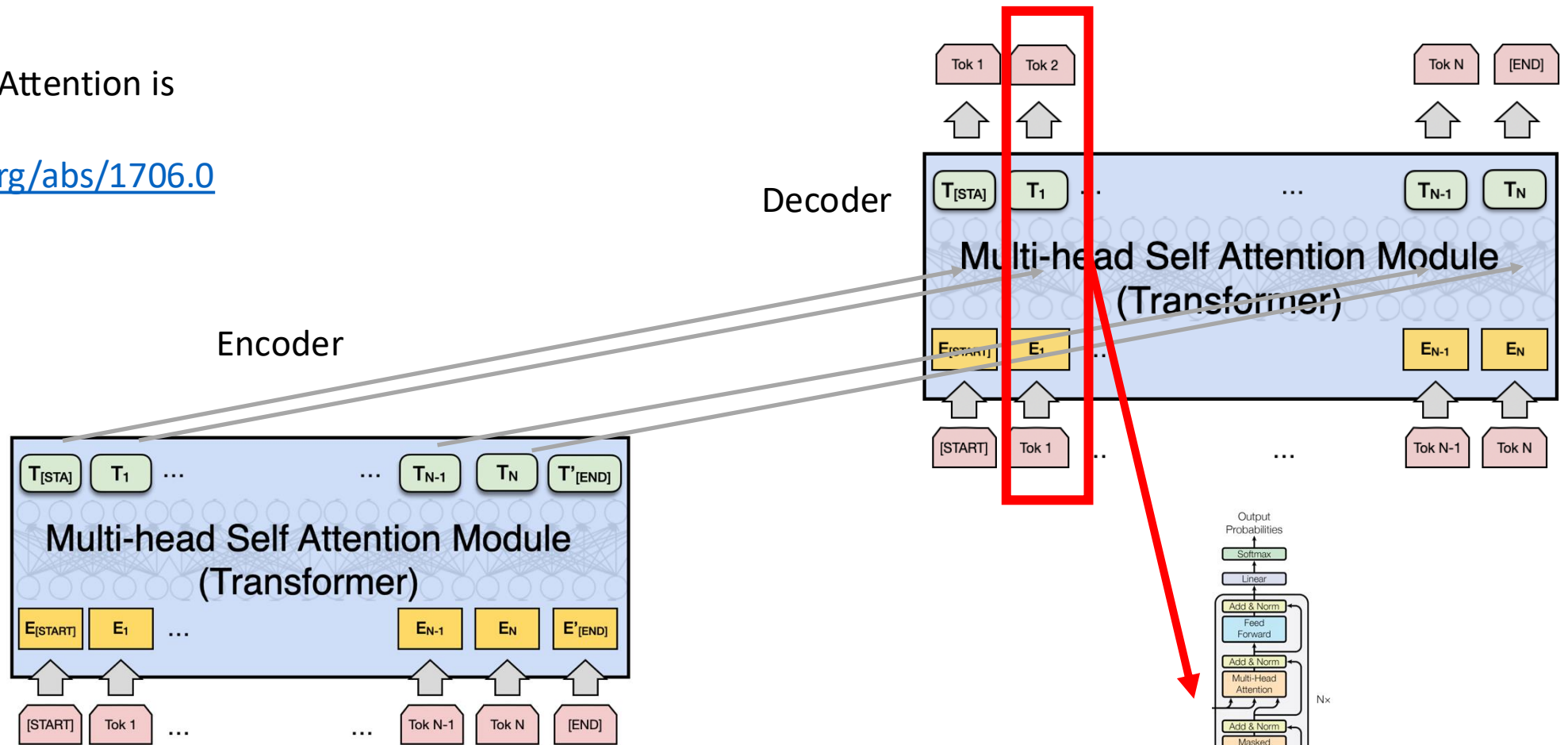
<https://arxiv.org/abs/1706.03762>



# Attention is All you Need (no RNNs)

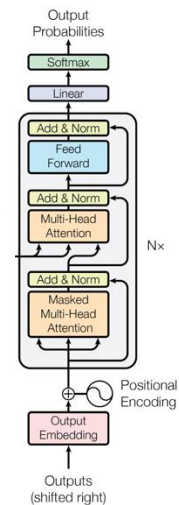
Vaswani et al. Attention is all you need

<https://arxiv.org/abs/1706.03762>



Fixed number of input tokens

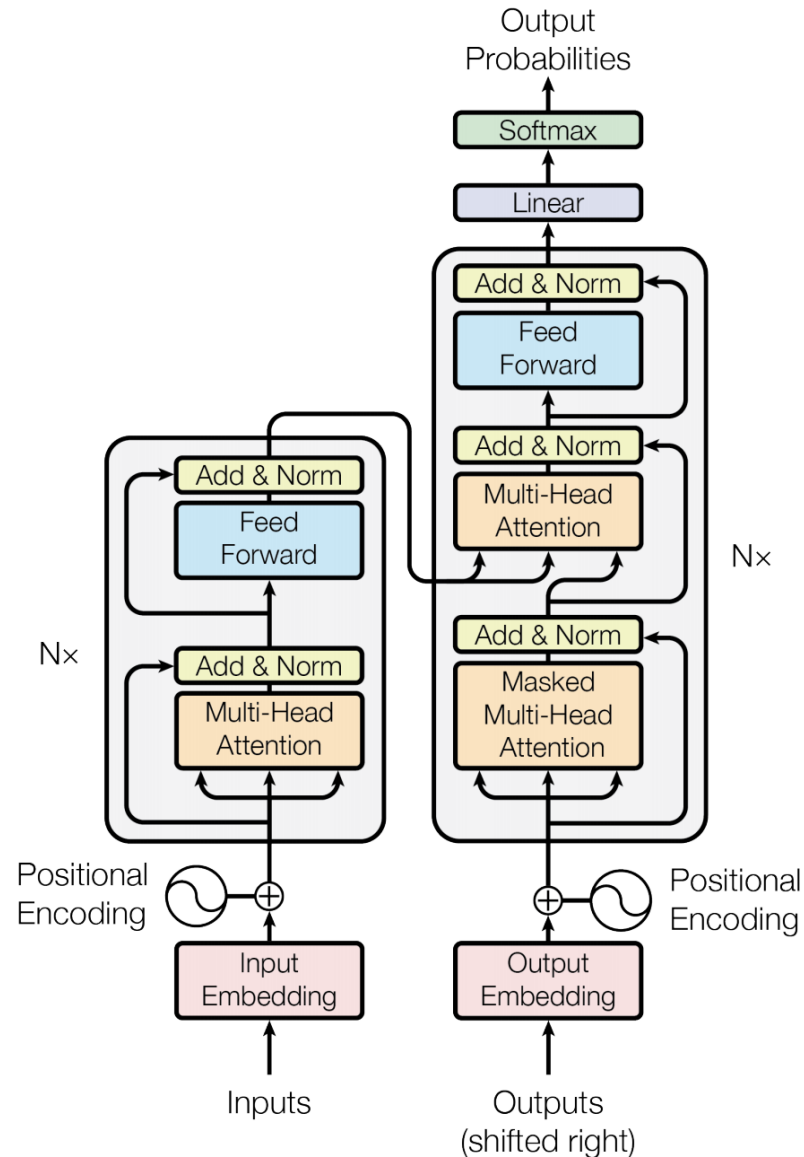
[but hey! we can always define a large enough length and add mask tokens]



# We can also draw this as in the paper:

Vaswani et al. Attention is all you need

<https://arxiv.org/abs/1706.03762>



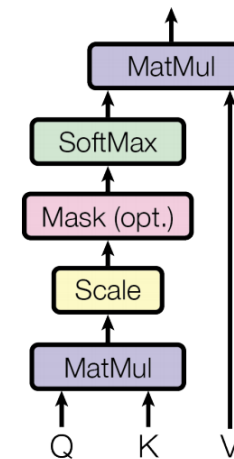
# Regular Attention: + Scaling factor

Vaswani et al. Attention is  
all you need

<https://arxiv.org/abs/1706.03762>

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

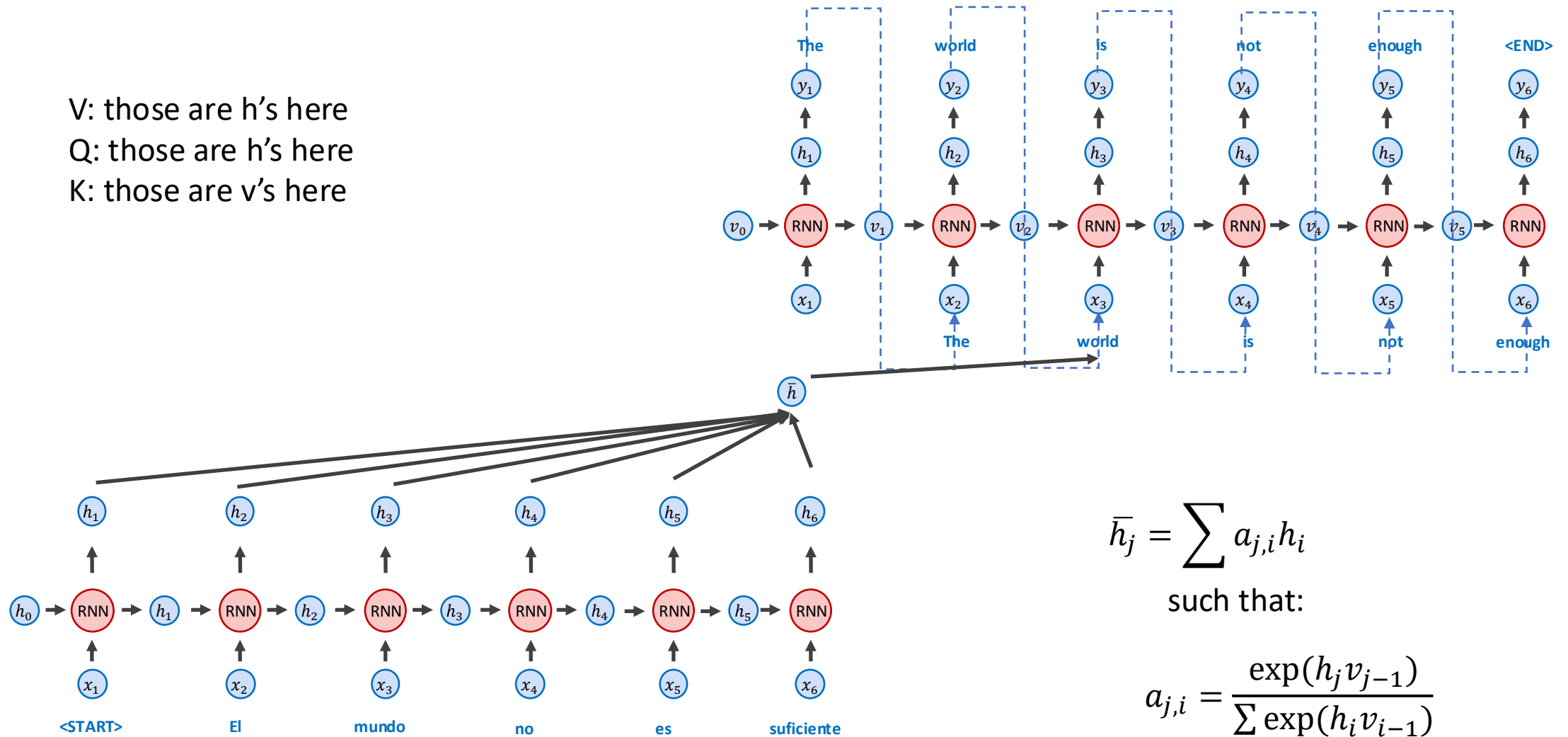
Scaled Dot-Product Attention



# This is not unlike what we already used before

Only showing the third time step encoder-decoder connection

V: those are h's here  
 Q: those are h's here  
 K: those are v's here



$$\bar{h}_j = \sum a_{j,i} h_i$$

such that:

$$a_{j,i} = \frac{\exp(h_j v_{j-1})}{\sum \exp(h_i v_{i-1})}$$

# Multi-head Attention: Do not settle for just one set of attention weights.

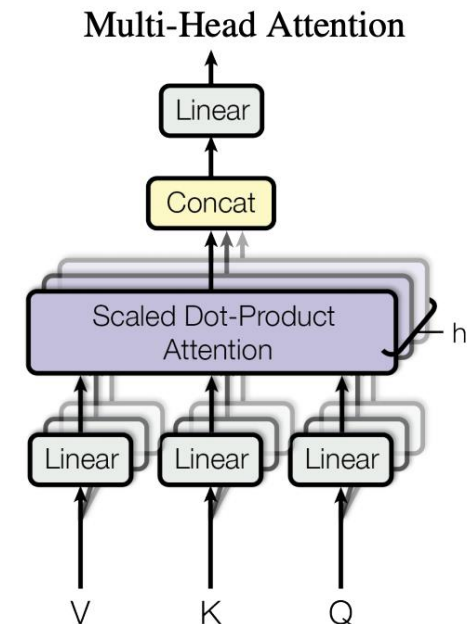
Vaswani et al. Attention is all you need

<https://arxiv.org/abs/1706.03762>

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O$$

where  $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$

Where the projections are parameter matrices  $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$ ,  $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$ ,  $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$  and  $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$ .

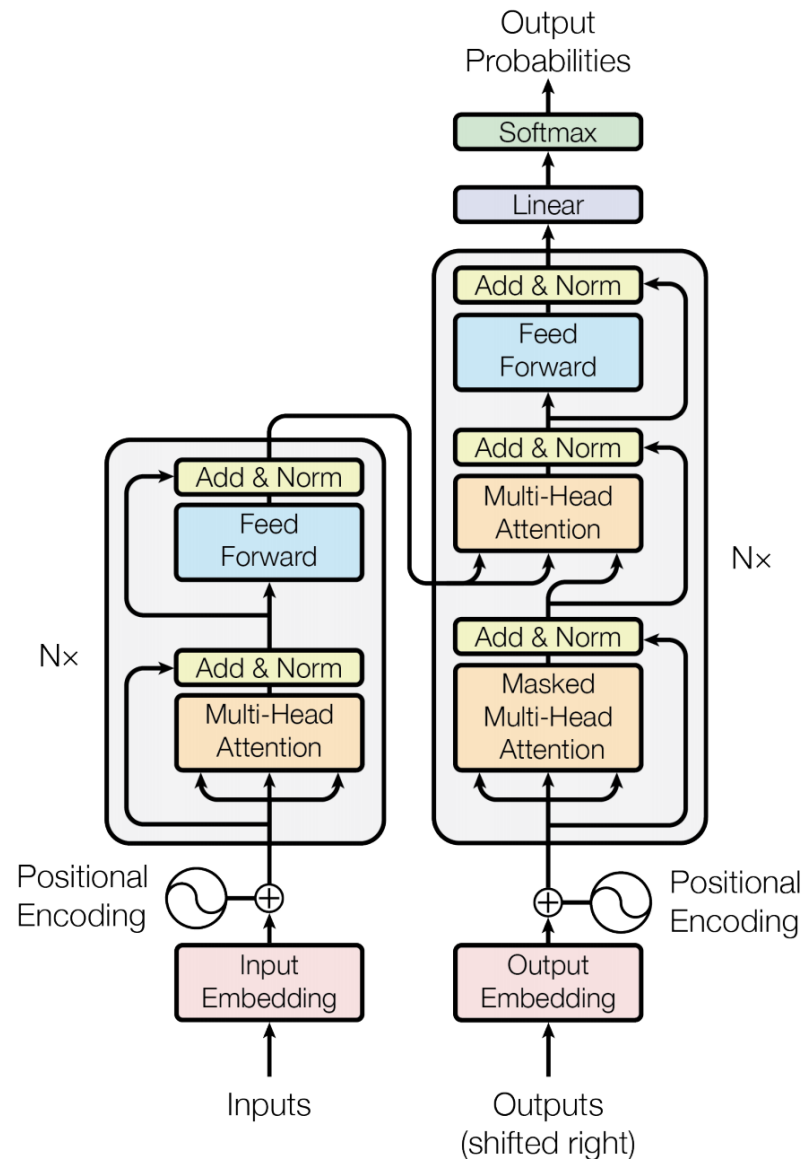


We can lose track of position since we are aggregating across all locations

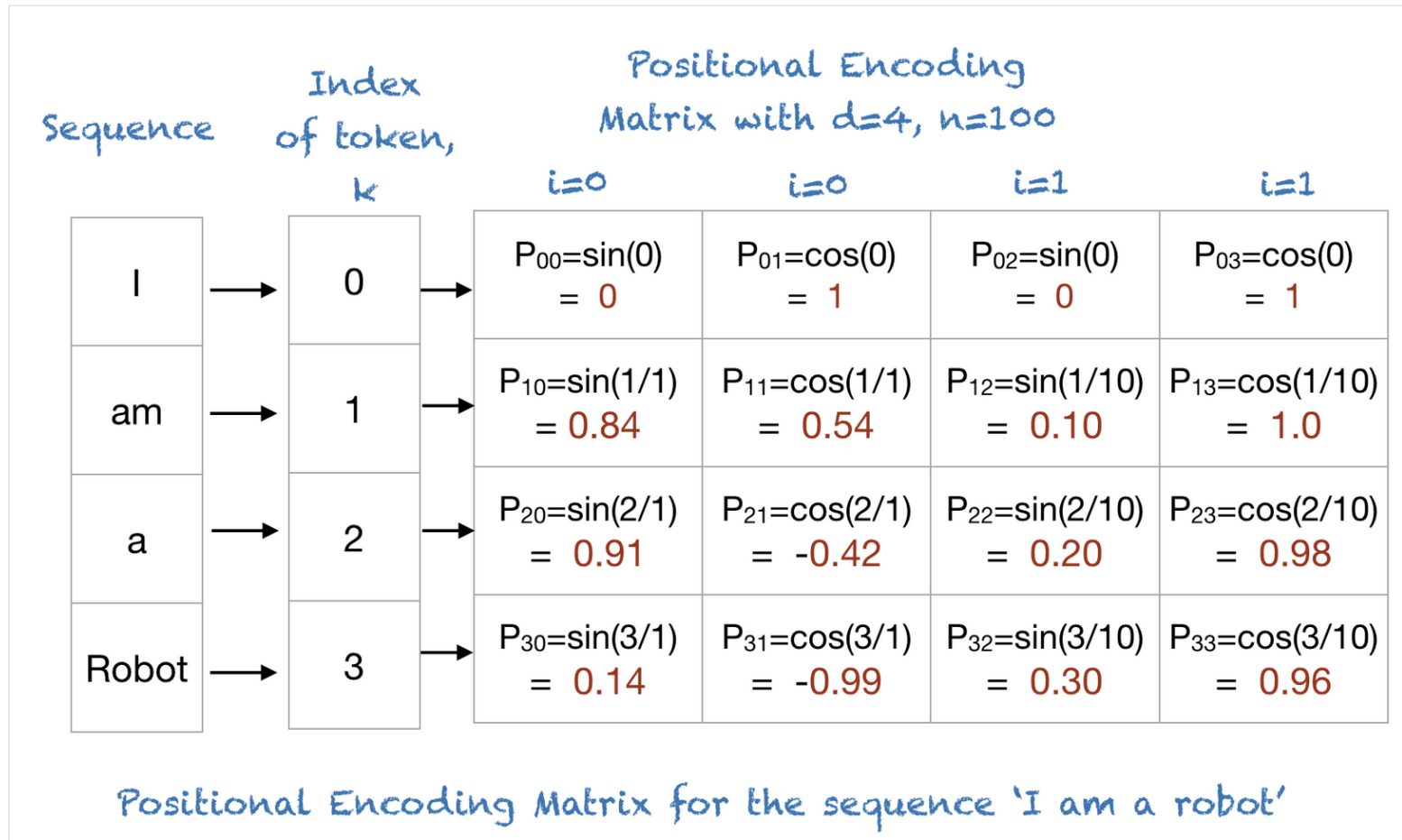
Vaswani et al. Attention is all you need

<https://arxiv.org/abs/1706.03762>

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}})$$
$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}})$$



# Positional Encodings

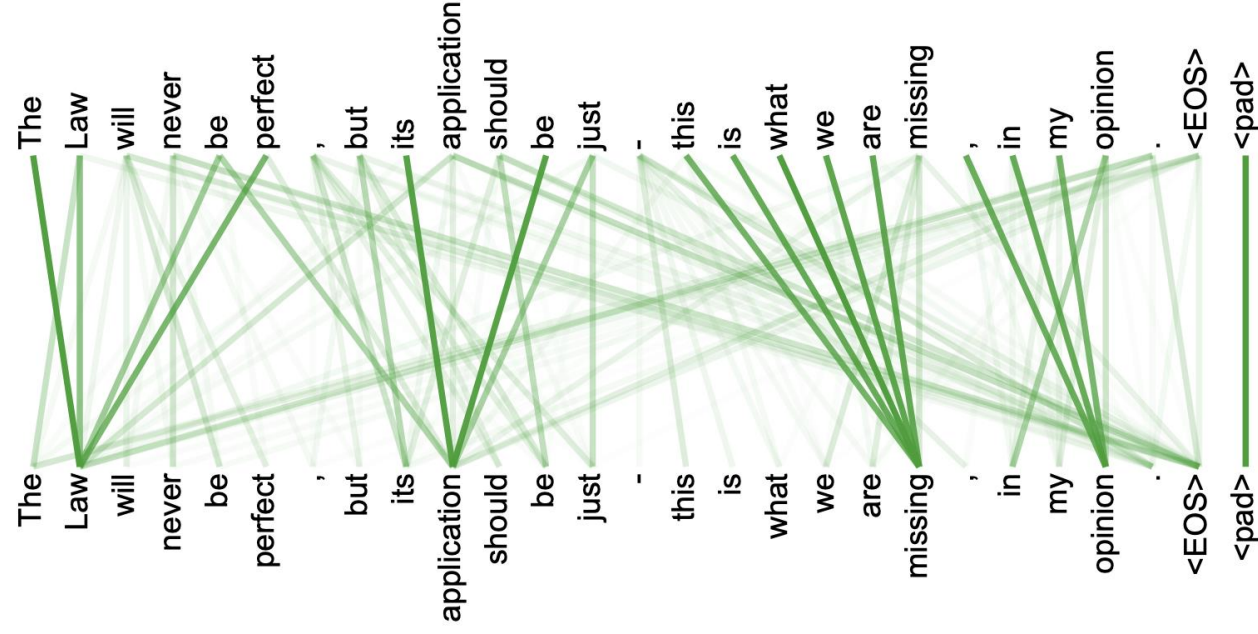
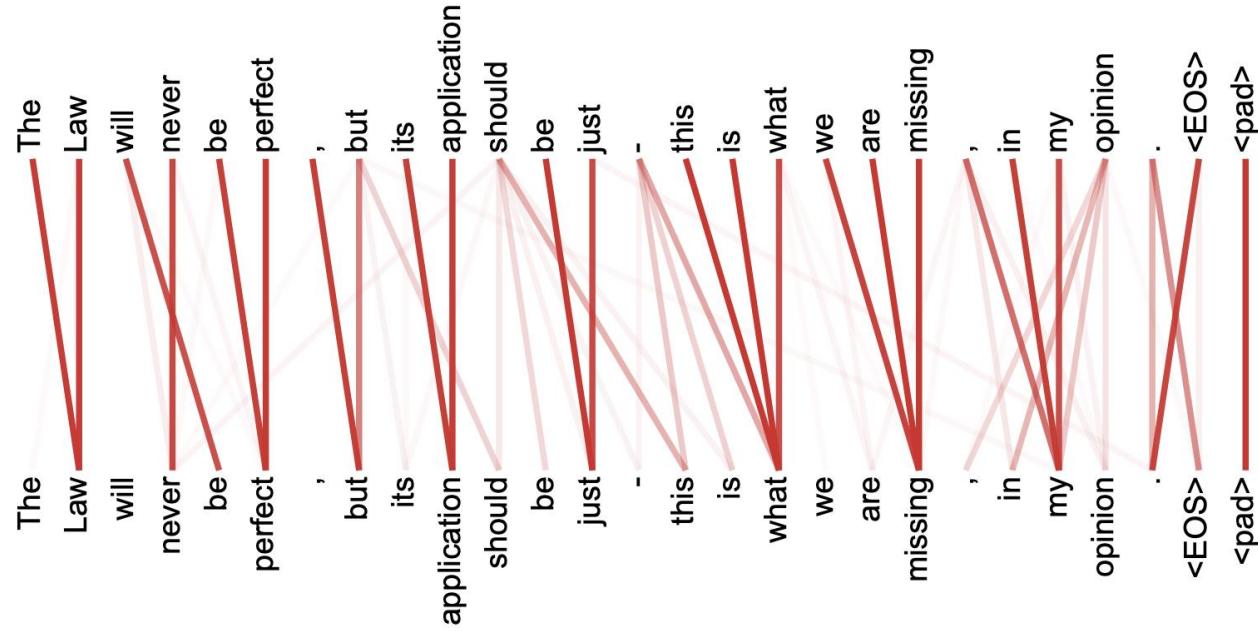


<https://machinelearningmastery.com/a-gentle-introduction-to-positional-encoding-in-transformer-models-part-1/>

# Other Positional Encodings

- **Learnable Absolute Positional Encodings:** Initialize them with random weights and learn them. However fixed for each position. BERT Models use this.
- **Relative Positional Encodings:** Encode distances between tokens instead of absolute positions. Used in moderns such as T5 from Google.
- **RoPE: Rotary Positional Encodings:** Position information is encoded through rotation in multidimensional space. Models such as LLaMA use this.
- **NoPE? No Positional Encodings?** Let's hope the model learns through attention weights?

Multi-headed attention weights are harder to interpret obviously

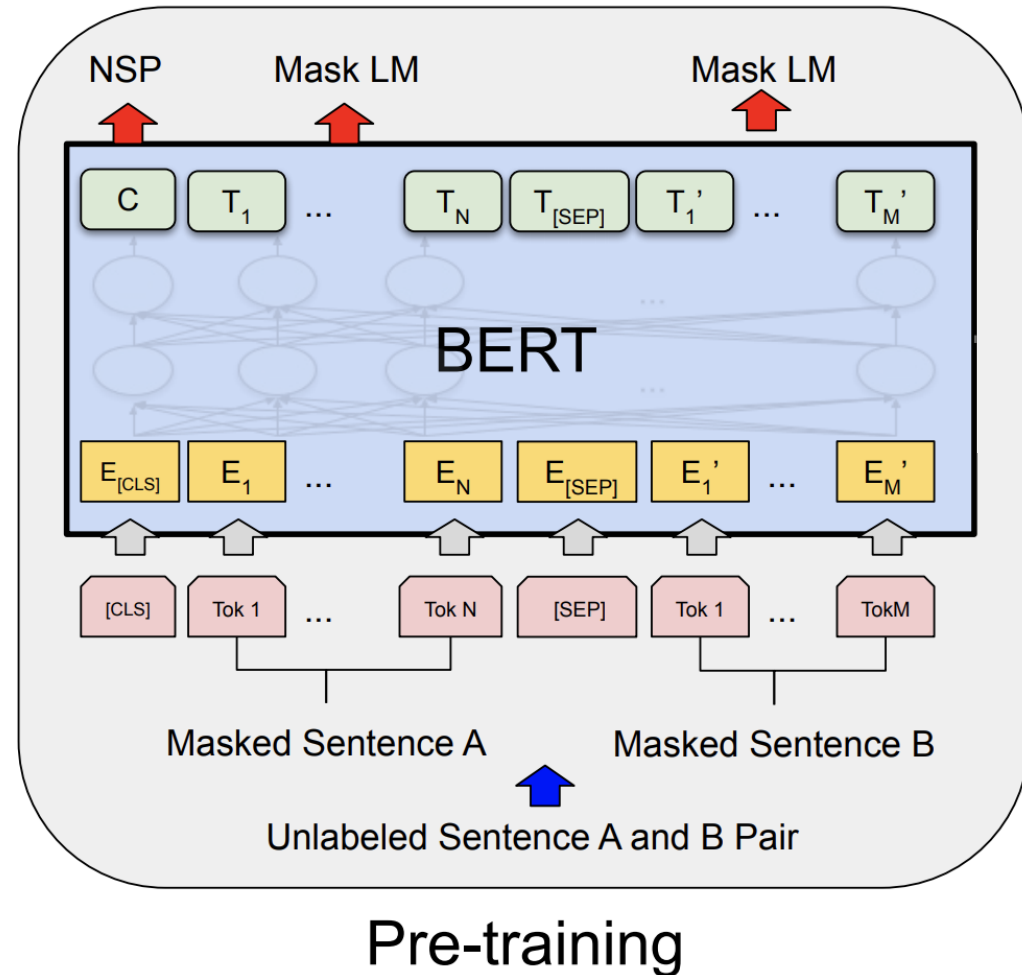


# The BERT Encoder Model (October, 2018)

Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding . <https://arxiv.org/abs/1810.04805>

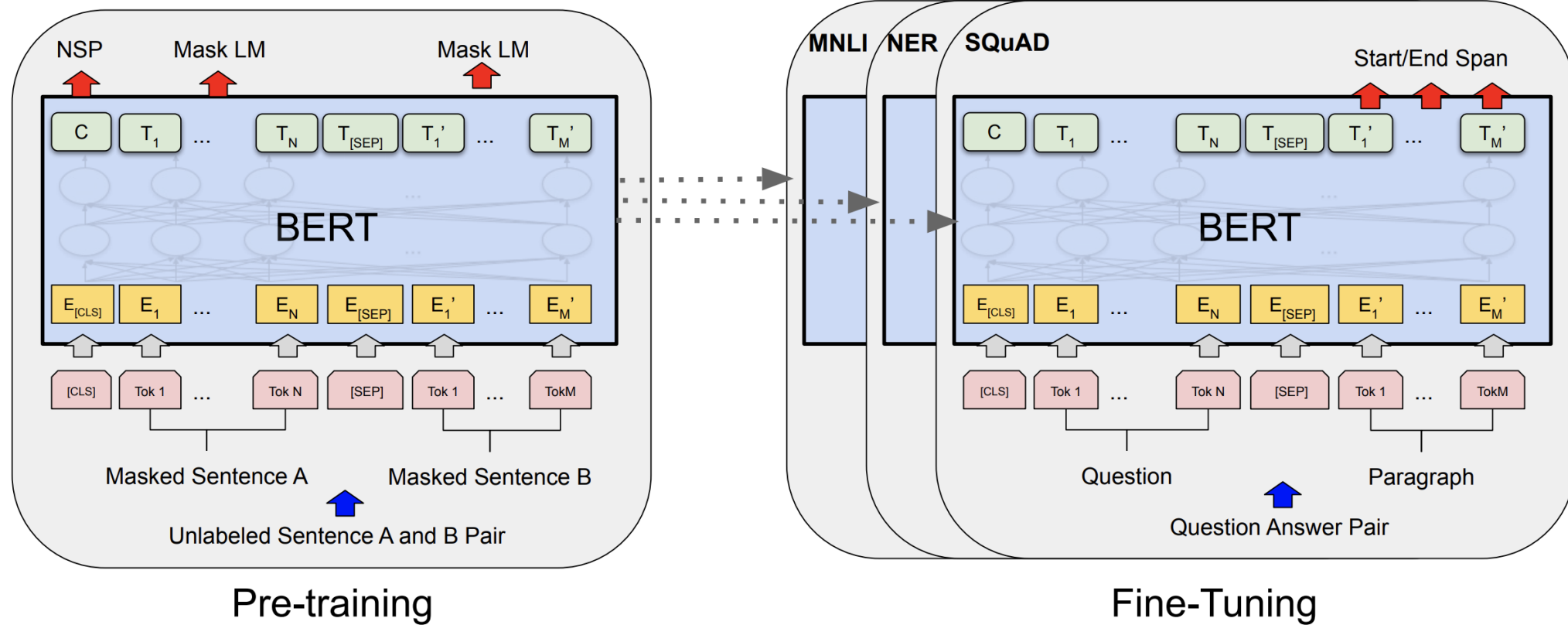
## Important things to know

- No decoder
- Train the model to fill-in-the-blank by masking some of the input tokens and trying to recover the full sentence.
- The input is not one sentence but two sentences separated by a [SEP] token.
- Also try to predict whether these two input sentences are consecutive or not.



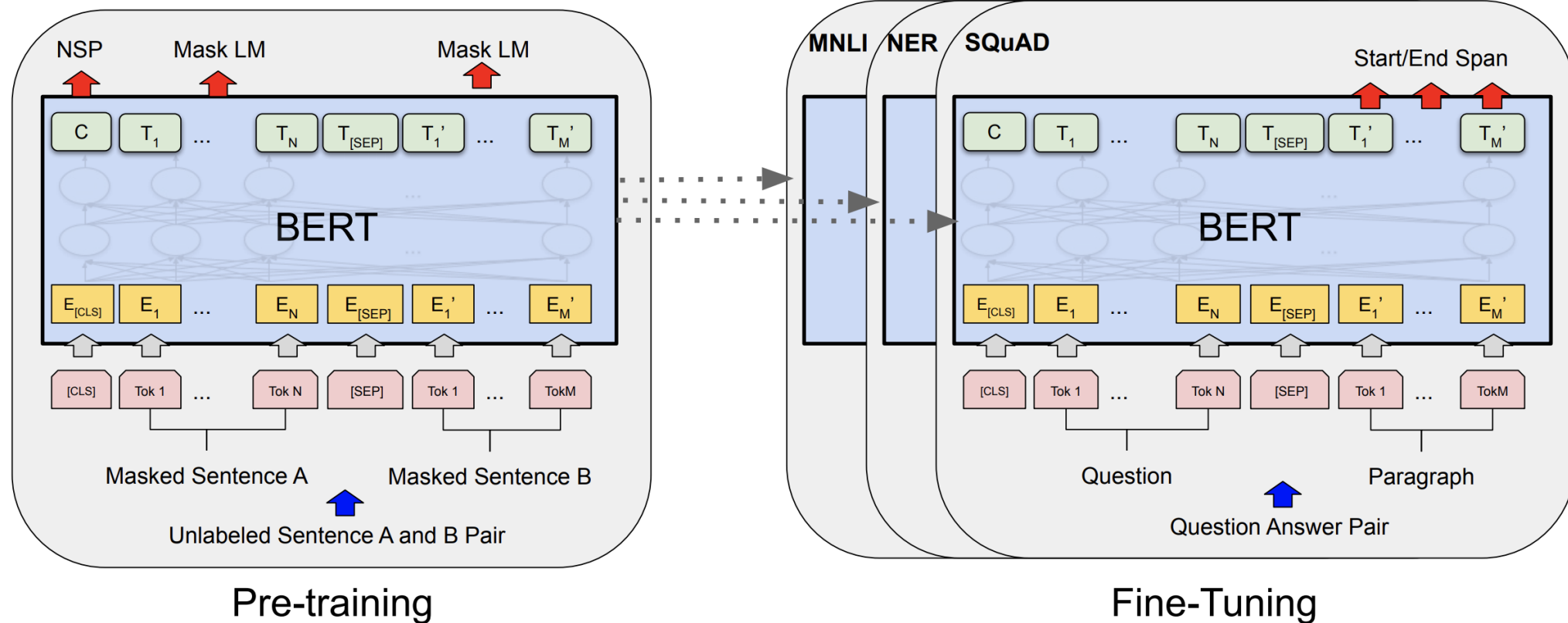
# The BERT Encoder Model

Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding . <https://arxiv.org/abs/1810.04805>

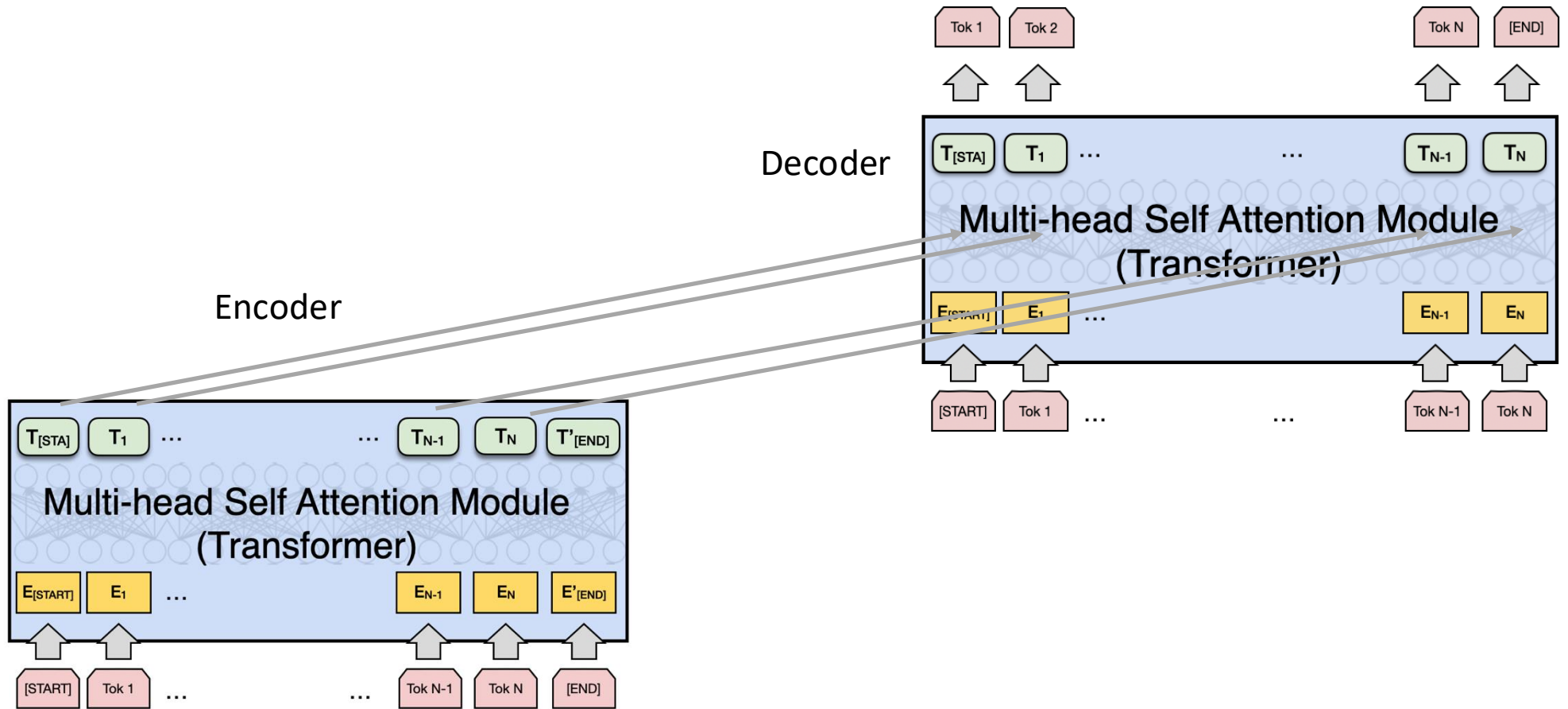


# The BERT Encoder-only Model

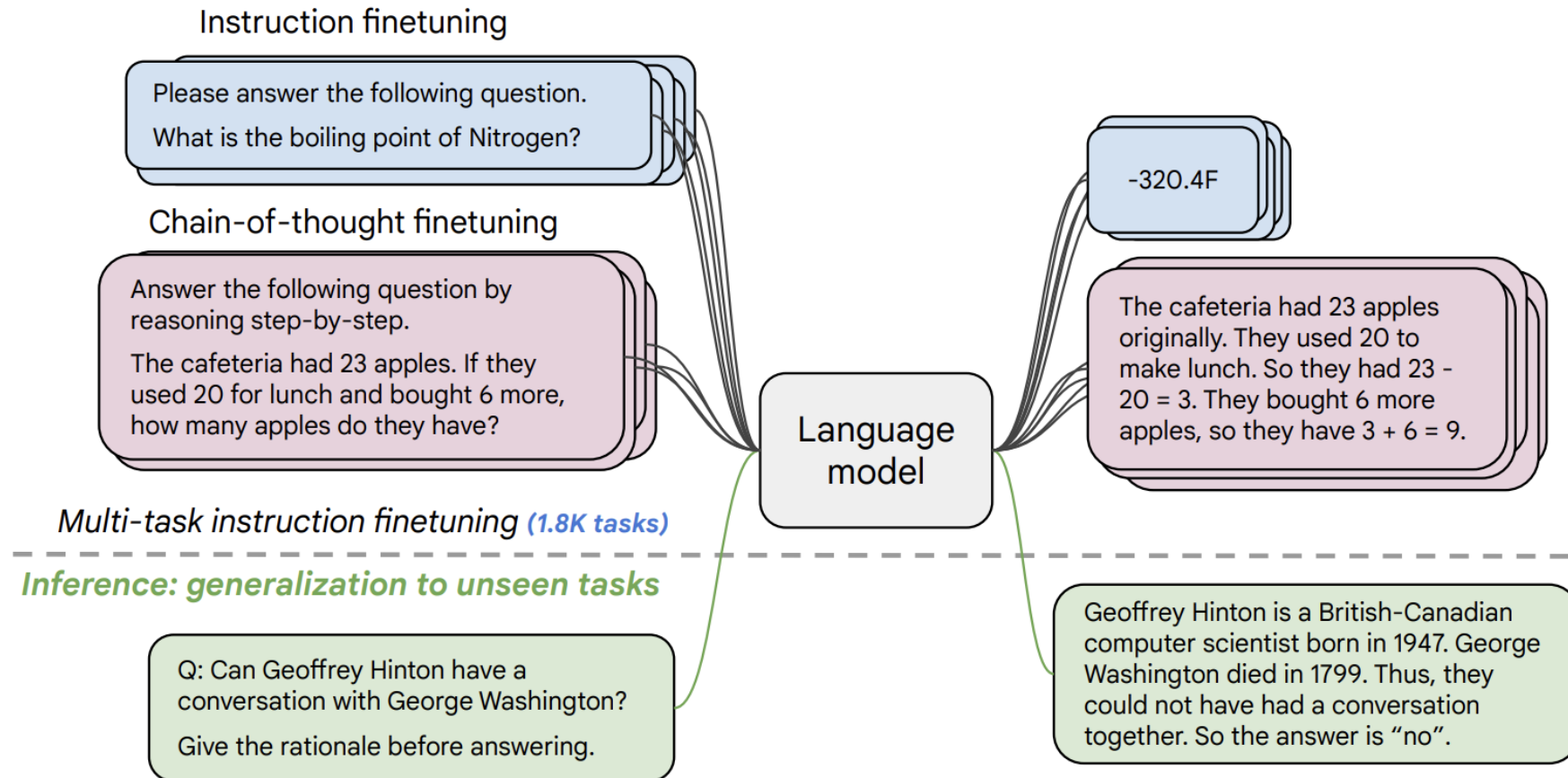
Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding . <https://arxiv.org/abs/1810.04805>



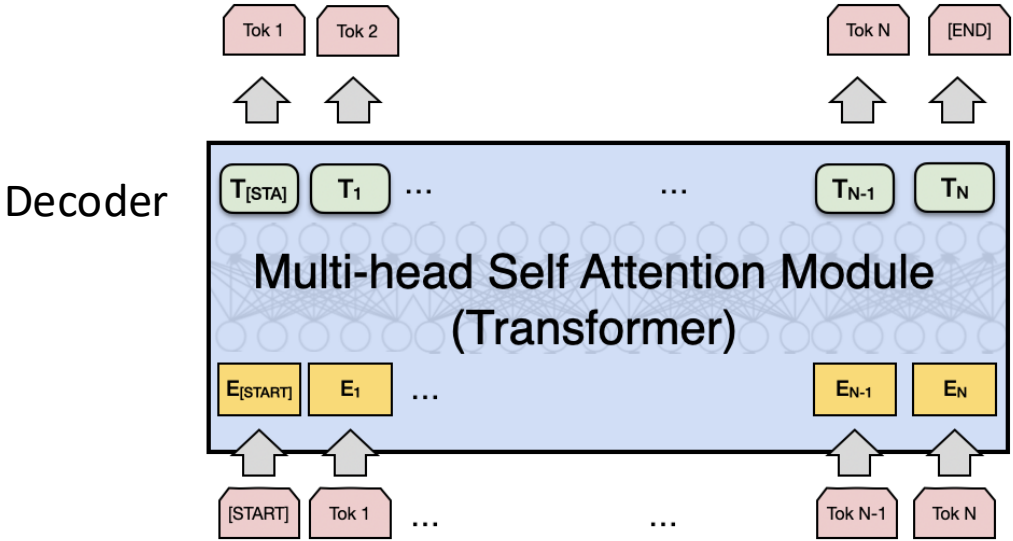
# The T5 Encoder-Decoder Model



# Instruction Tuning (FLAN-T5 by Google)



# The GPT-2, GPT-3 Decoder-only Model



# The GPT-2 Model (Feb, 2019)

---

## **Language Models are Unsupervised Multitask Learners**

---

**Alec Radford<sup>\* 1</sup> Jeffrey Wu<sup>\* 1</sup> Rewon Child<sup>1</sup> David Luan<sup>1</sup> Dario Amodei<sup>\*\* 1</sup> Ilya Sutskever<sup>\*\* 1</sup>**

<https://openai.com/blog/better-language-models/>

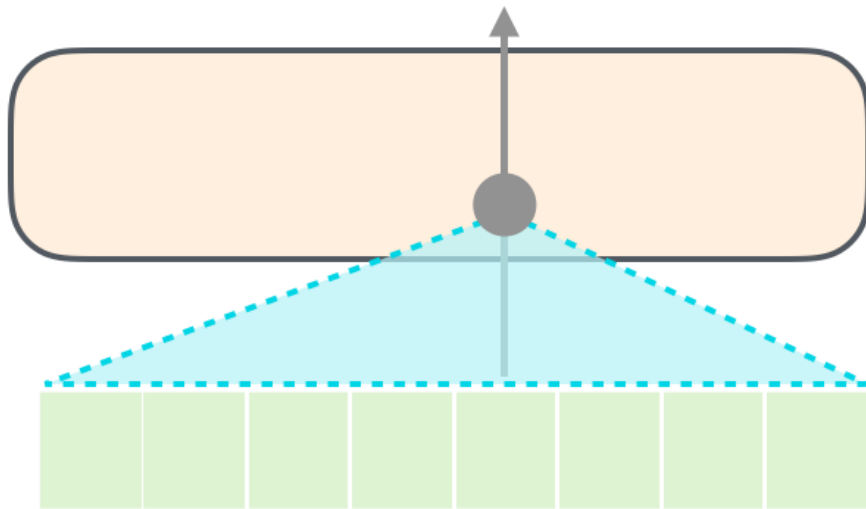
# The GPT-2 Model



# The GPT-2 Model

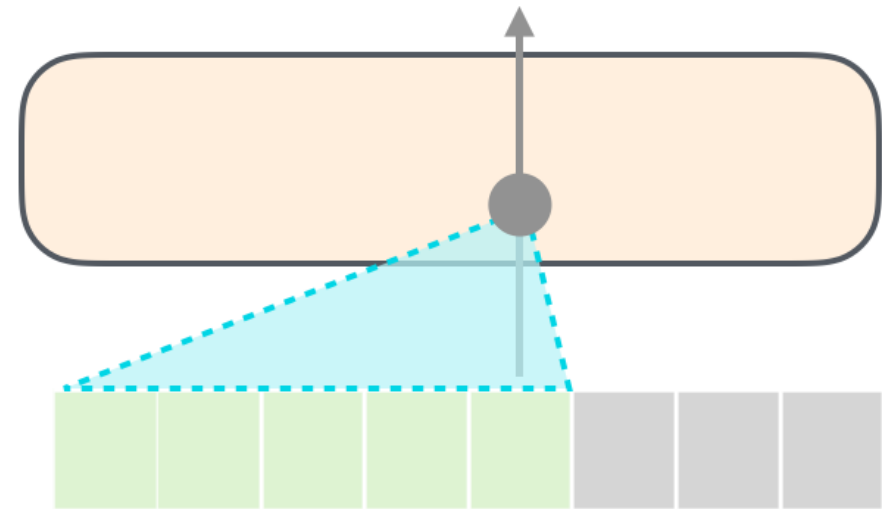
## BERT

### Self-Attention

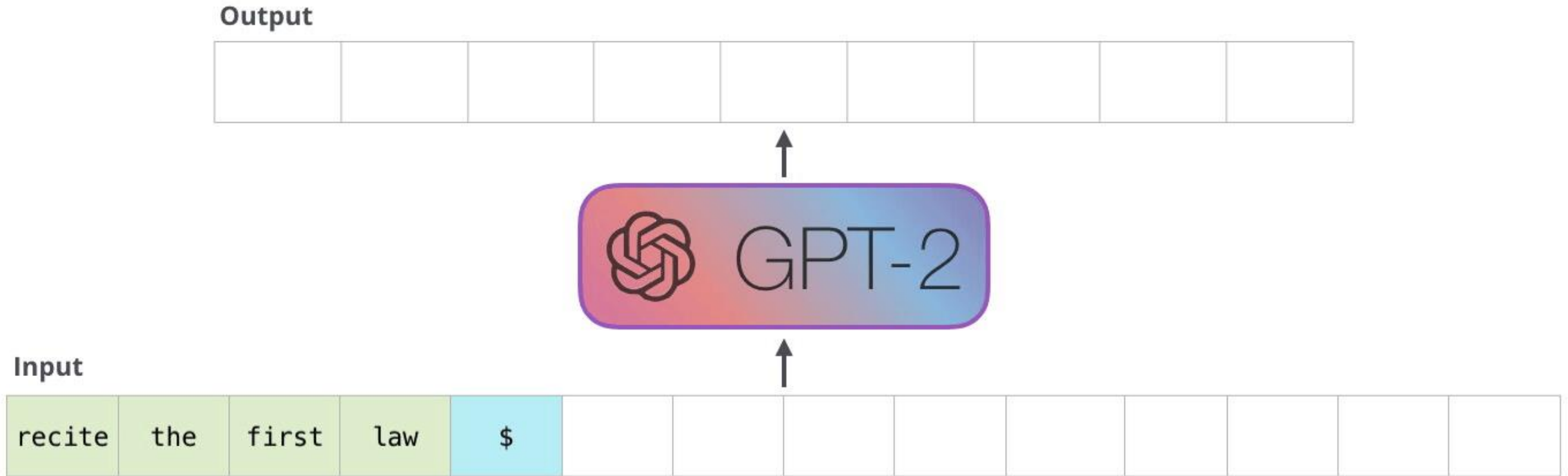


## GPT

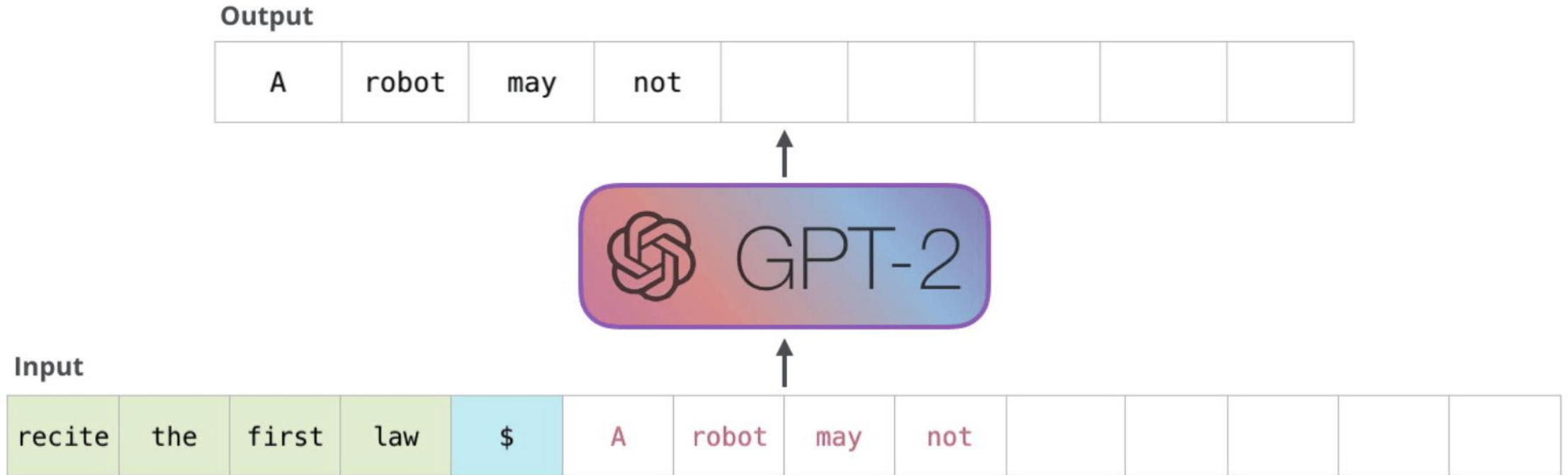
### Masked Self-Attention



# The GPT-2 Model



# The GPT-2 Model



# The GPT-2 Model



# GPT-1 vs GPT-2 vs GPT-3

	GPT-1	GPT-2	GPT-3
Parameters	117 Million	1.5 Billion	175 Billion
Decoder Layers	12	48	96
Context Token Size	512	1024	2048
Hidden Layer	768	1600	12288
Batch Size	64	512	3.2M

# GPT-3 (July, 2020)

Model Name	$n_{\text{params}}$	$n_{\text{layers}}$	$d_{\text{model}}$	$n_{\text{heads}}$	$d_{\text{head}}$	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	$6.0 \times 10^{-4}$
GPT-3 Medium	350M	24	1024	16	64	0.5M	$3.0 \times 10^{-4}$
GPT-3 Large	760M	24	1536	16	96	0.5M	$2.5 \times 10^{-4}$
GPT-3 XL	1.3B	24	2048	24	128	1M	$2.0 \times 10^{-4}$
GPT-3 2.7B	2.7B	32	2560	32	80	1M	$1.6 \times 10^{-4}$
GPT-3 6.7B	6.7B	32	4096	32	128	2M	$1.2 \times 10^{-4}$
GPT-3 13B	13.0B	40	5140	40	128	2M	$1.0 \times 10^{-4}$
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128	3.2M	$0.6 \times 10^{-4}$

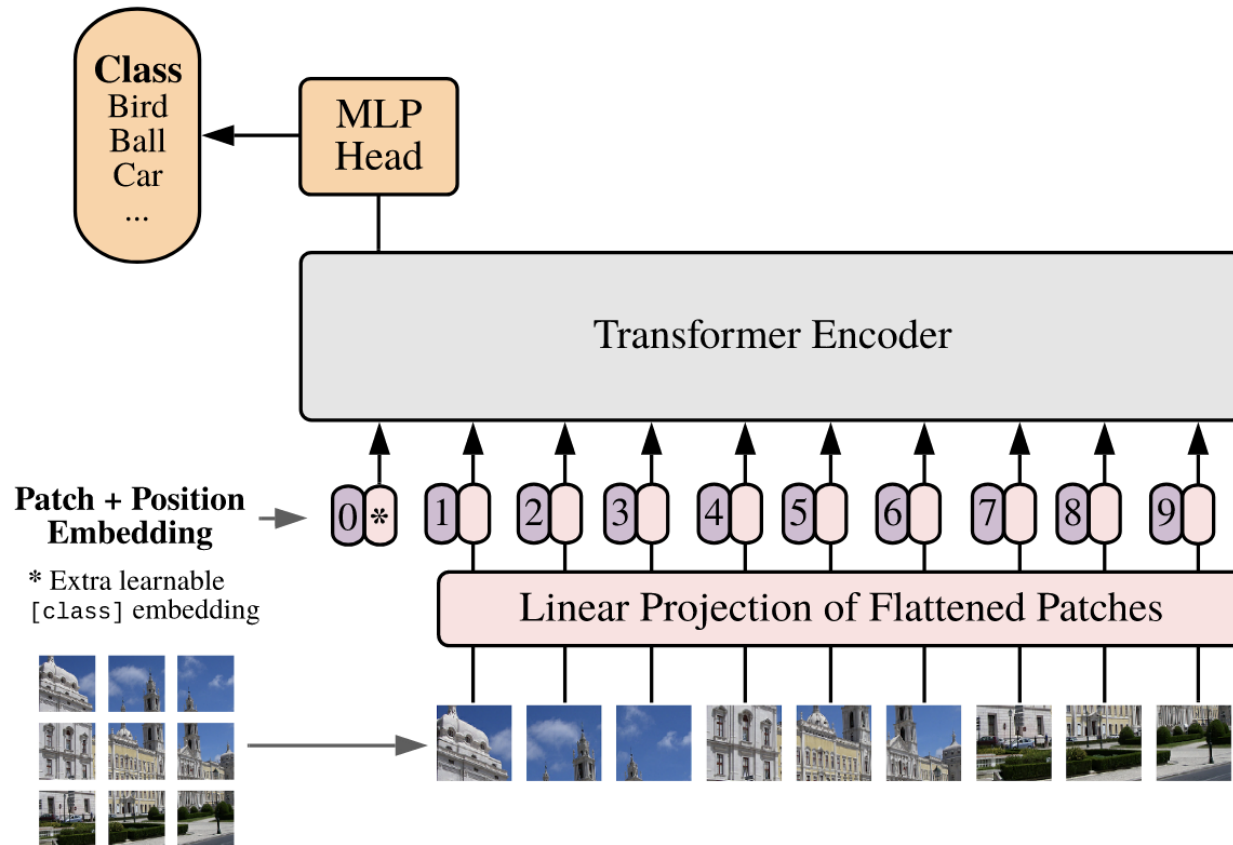
# GPT family keeps growing

- GPT-3.5
- GPT-3.5-turbo
- GPT-4, GPT 4.1
- GPT-4-turbo
- GPT-4o
- o1, o3, o3-pro
- GPT-5.2 (Thinking)

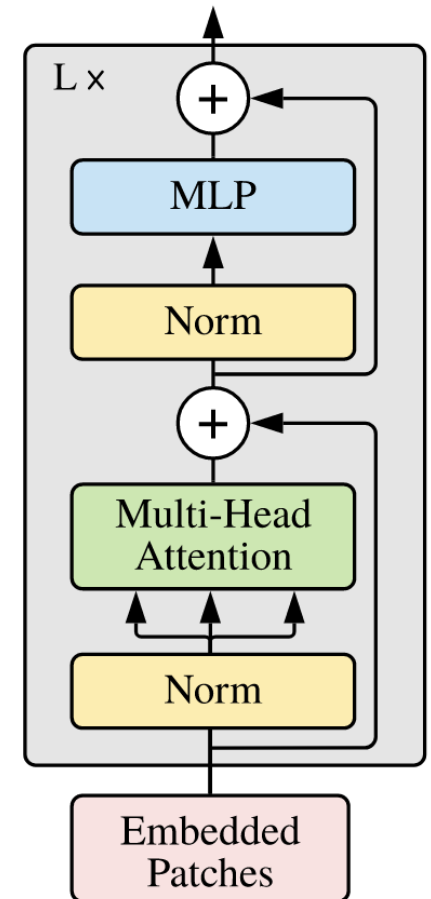
## Competitors

- Gemini family (Gemini Pro) (Google) (Gemini Pro 3.0)
- Mistral 7xMoE (Open Source by Mistral.ai)
- Llama-2, Llama-3(Open Source by Meta AI), Llama-4
- Qwen3.5, Qwen3, Qwen2.5 (Alibaba)
- DeepSeekV3, DeepSeek, DeepSeek-R1 (Open Source by DeepSeek Team)
- Claude Opus 4.6 (Extended), Claude3.5 Sonnet, Haiku, etc (Anthropic)
- Grok3, Grok4 (Twitter/xAI)

# Vision Transformers



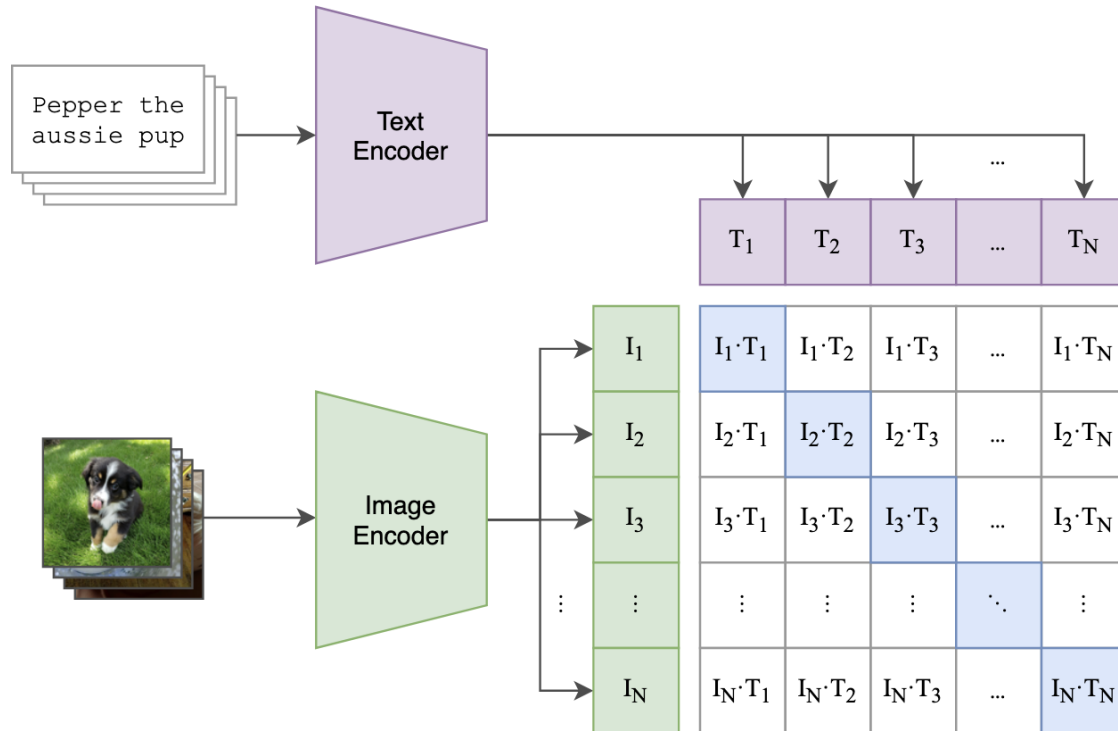
## Transformer Encoder



<https://arxiv.org/abs/2010.11929>

**An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale**  
[Alexey Dosovitskiy](#), [Lucas Beyer](#), [Alexander Kolesnikov](#), [Dirk Weissenborn](#), [Xiaohua Zhai](#), [Thomas Unterthiner](#), [Mostafa Dehghani](#), [Matthias Minderer](#), [Georg Heigold](#), [Sylvain Gelly](#), [Jakob Uszkoreit](#), [Neil Houlsby](#)

# The CLIP Model



$$L = \sum_k \ell(I_k T_k)$$

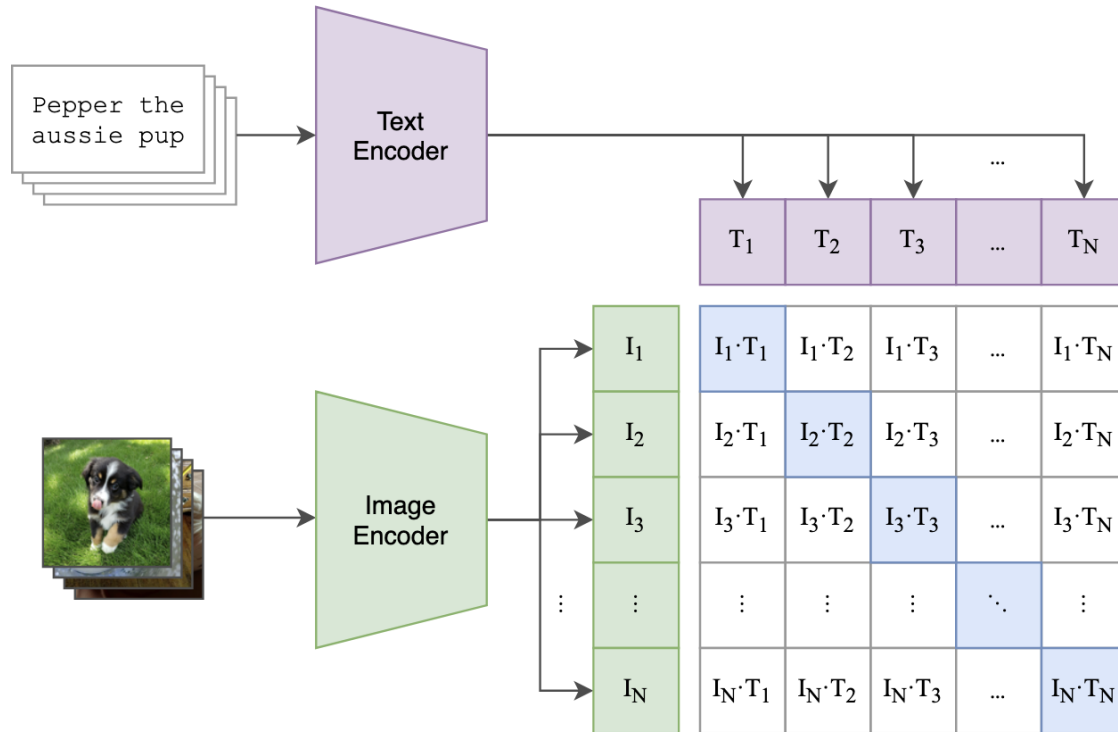
$$\ell(I_k T_k) = -\log \left( \frac{\exp(\text{sim}(I_k, T_k))}{\sum_{t=1}^{2N} 1[k \neq i] \exp(\text{sim}(I_k, T_t))} \right)$$

<https://arxiv.org/abs/2103.00020>

**Learning Transferable Visual Models From Natural Language Supervision**

[Alec Radford](#), [Jong Wook Kim](#), [Chris Hallacy](#), [Aditya Ramesh](#), [Gabriel Goh](#),  
[Sandhini Agarwal](#), [Girish Sastry](#), [Amanda Askell](#), [Pamela Mishkin](#), [Jack Clark](#),  
[Gretchen Krueger](#), [Ilya Sutskever](#)

# The CLIP Model



$$L = \sum_k \ell_1(I_k T_k) + \ell_2(I_k T_k)$$

$$\ell_1(I_k T_k) = -\log \left( \frac{\exp(\text{sim}(I_k, T_k))}{\sum_{t=1}^{2N} 1[k \neq i] \exp(\text{sim}(I_k, T_t))} \right)$$

$$\ell_2(I_k T_k) = -\log \left( \frac{\exp(\text{sim}(I_k, T_k))}{\sum_{t=1}^{2N} 1[k \neq i] \exp(\text{sim}(I_t, T_k))} \right)$$

<https://arxiv.org/abs/2103.00020>

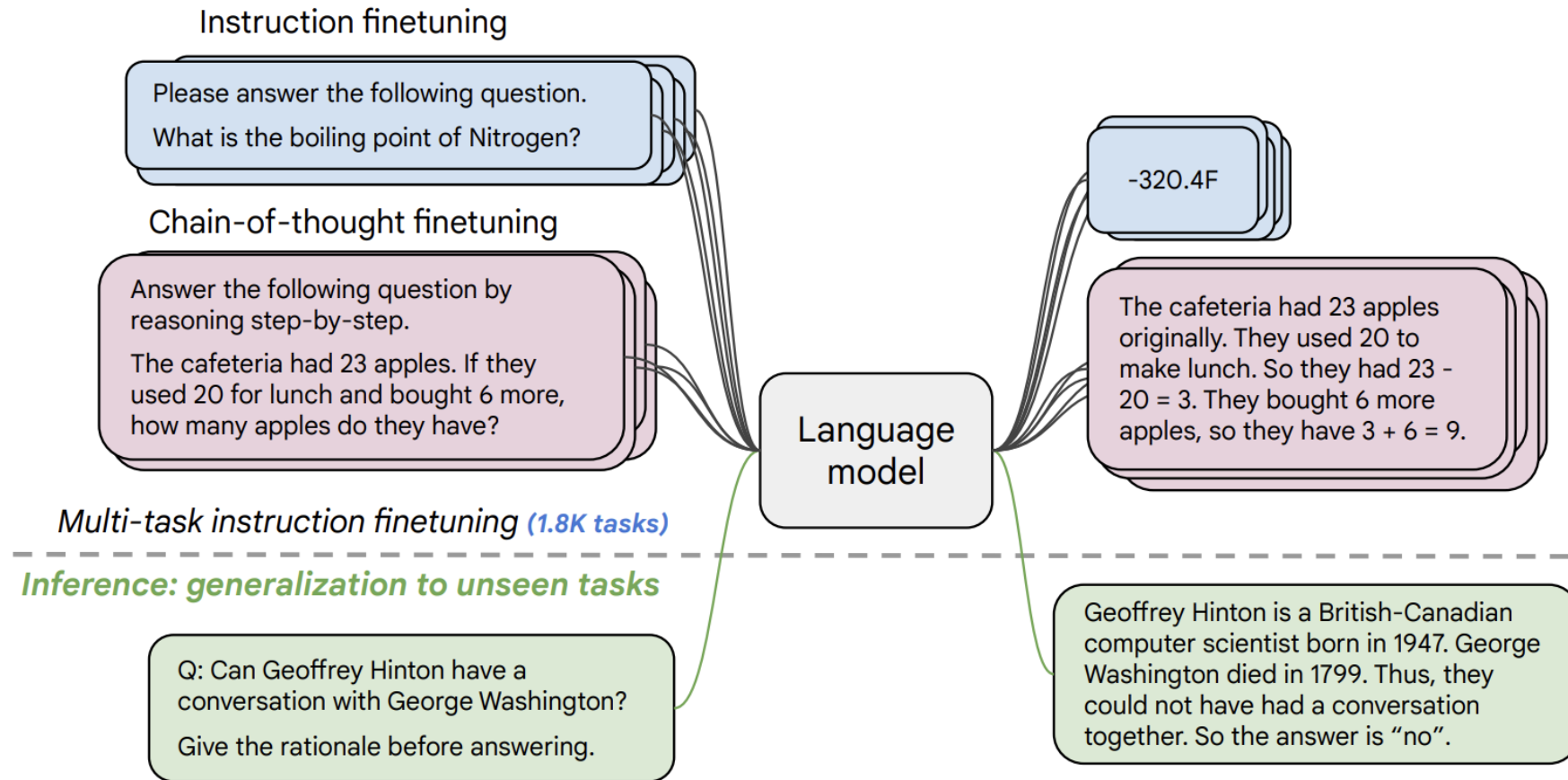
**Learning Transferable Visual Models From Natural Language Supervision**

[Alec Radford](#), [Jong Wook Kim](#), [Chris Hallacy](#), [Aditya Ramesh](#), [Gabriel Goh](#),  
[Sandhini Agarwal](#), [Girish Sastry](#), [Amanda Askell](#), [Pamela Mishkin](#), [Jack Clark](#),  
[Gretchen Krueger](#), [Ilya Sutskever](#)

# Next Work Prediction is limited

- Predicting the next word can lead to intelligent behavior such as the one exemplified earlier however this still limited
- What makes some of the new LLMs special? ChatGPT (GPT-3.5, 3.5 Turbo, 4, 4-turbo), FLAN-T5, OPT-IML
  - SFT: Supervised Finetuning (Curated Input/Output Instruction Sets)
  - DPO: Direct Preference Optimization
  - PPO: Proximal Policy Optimization (Reinforcement Learning with Human Feedback)
  - GRPO: Group-relative Policy Optimization (DeepSeek)

# Instruction Tuning (e.g. FLAN-T5 by Google)



# FLAN-T5

## Model input (Disambiguation QA)

Q: In the following sentences, explain the antecedent of the pronoun (which thing the pronoun refers to), or state that it is ambiguous.

Sentence: The reporter and the chef will discuss their favorite dishes.

Options:

- (A) They will discuss the reporter's favorite dishes
- (B) They will discuss the chef's favorite dishes
- (C) Ambiguous

A: Let's think step by step.

## Before instruction finetuning

The reporter and the chef will discuss their favorite dishes.

The reporter and the chef will discuss the reporter's favorite dishes.

The reporter and the chef will discuss the chef's favorite dishes.

The reporter and the chef will discuss the reporter's and the chef's favorite dishes.

✘ (doesn't answer question)

# FLAN-T5

## Model input (Disambiguation QA)

Q: In the following sentences, explain the antecedent of the pronoun (which thing the pronoun refers to), or state that it is ambiguous.

Sentence: The reporter and the chef will discuss their favorite dishes.

Options:

- (A) They will discuss the reporter's favorite dishes
- (B) They will discuss the chef's favorite dishes
- (C) Ambiguous

A: Let's think step by step.

## After instruction finetuning

The reporter and the chef will discuss their favorite dishes does not indicate whose favorite dishes they will discuss. So, the answer is (C). ✓

# Questions?