Machine Learning 2

DS 4420 - Spring 2020

Transformers

Byron C. Wallace

Material in this lecture derived rom materials created by Jay Alammar (http://jalammar.github.io/illustrated-transformer/)

Some housekeeping

- First, let's talk midterm…

Some housekeeping

- First, let's talk midterm…
- Mean: 70 (from 30s to high 90s)

Some housekeeping

- First, let's talk midterm…
- Mean: 70 (from 30s to high 90s)
- I miscalibrated Q2 (average: 56%)

Some housekeeping

- First, let's talk midterm…
- Mean: 70 (from 30s to high 90s)
- I miscalibrated Q2 (average: 56%)
  - ★ I gave back to 5 points to everyone (mean now 75)

Some housekeeping

- First, let's talk midterm…
- Mean: 70 (from 30s to high 90s)
- I miscalibrated Q2 (average: 56%)
  - ★ I gave back to 5 points to everyone (mean now 75)
  - ★ We are releasing an **optional** bonus assignment that covers the same content as Q2 — you can use this to make up *up to half* (12.5) points on said question. This will be released tonight; due date is flexible.

HW 4

- HW 4 will be released soon; due 3/24 (Tuesday)

Projects!

- **THURSDAY 3/13 Project proposal is due!**
- **TUESDAY 3/17 Project pitches in class!**

A remote possibility

- There is a (increasingly) non-zero chance that Northeastern will move to holding all classes remotely in the coming days/weeks

- In this case: Remote / recorded lectures; on-demand office hours, remotely; project presentations (+ pitches) will also have to be remote or recorded (will figure out!)

- Keep an eye on Piazza for more updates

Today

- Will introduce *transformer* networks, which are a type of neural networks that have come to dominate in NLP

Today

- Will introduce *transformer* networks, which are a type of neural networks that have come to dominate in NLP
- To get there, will first review RNNs briefly

RNNs

- Review [on board]

Transformers

- Hey, maybe we can get rid of recurrence!

Attention mechanisms

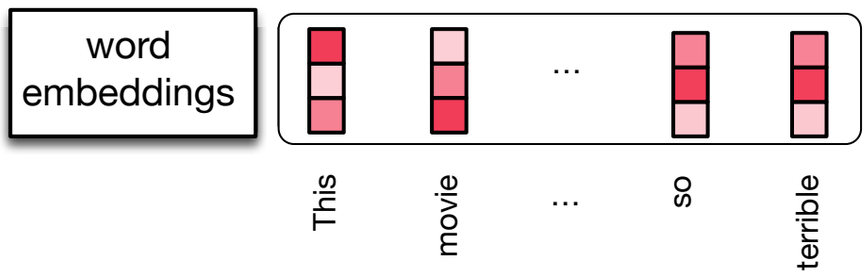
This

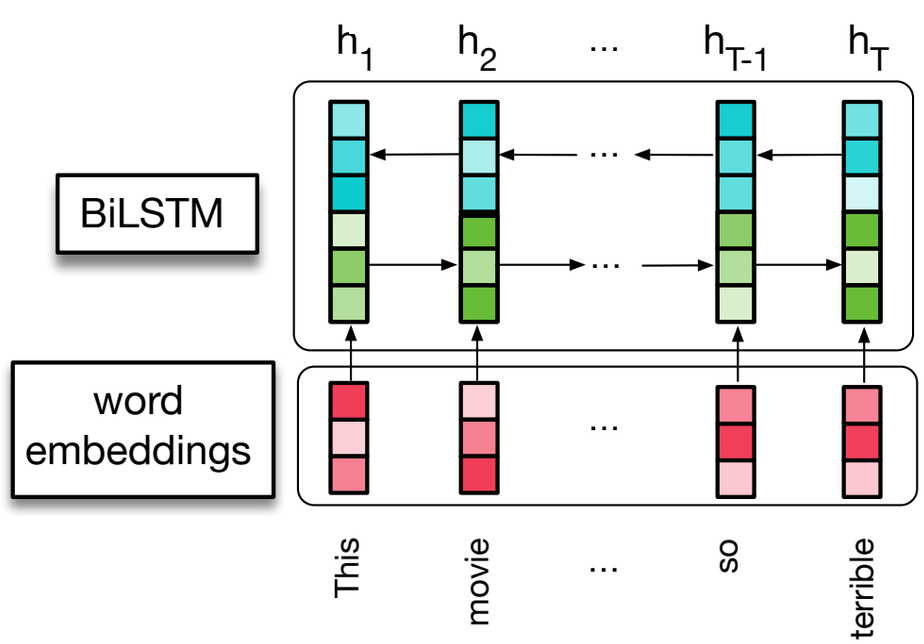
movie

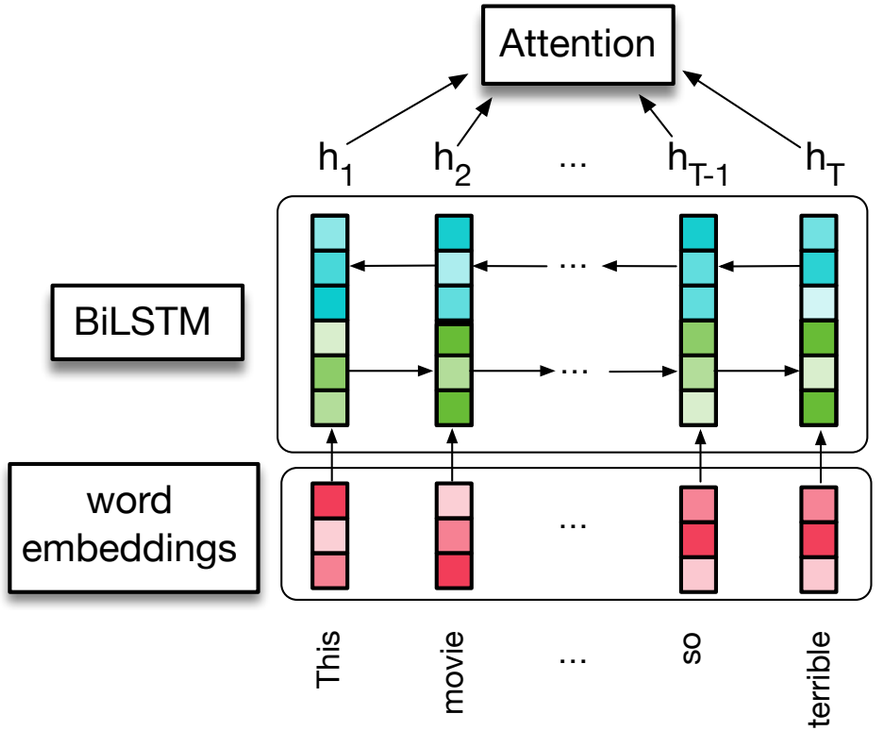
⋮

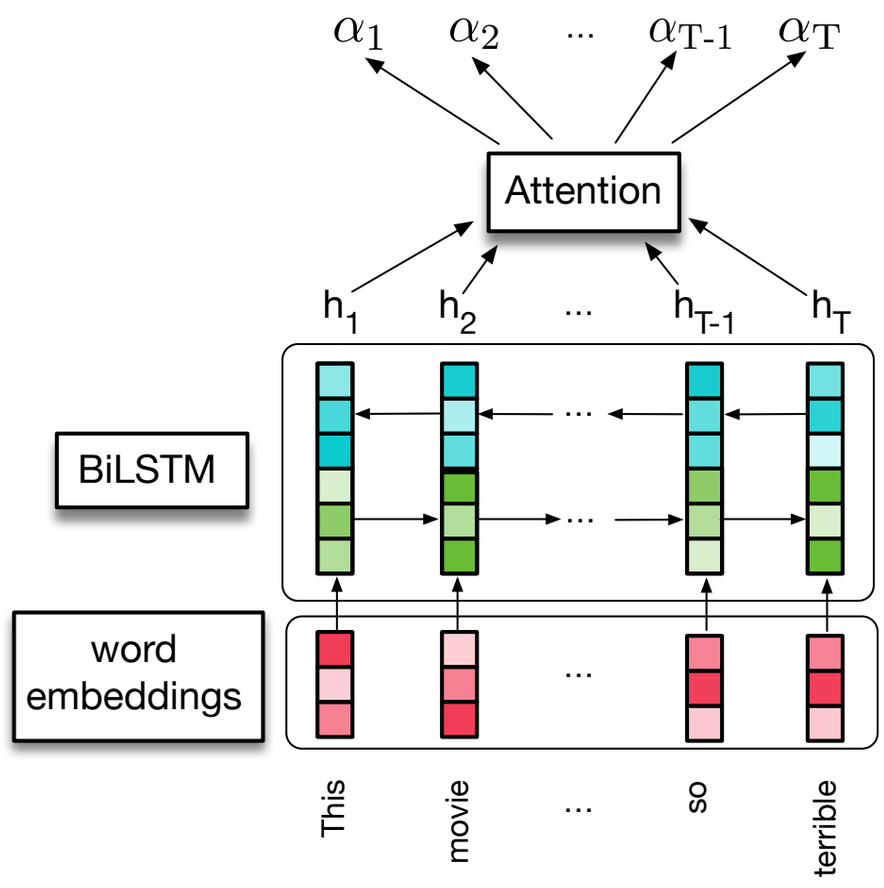
so

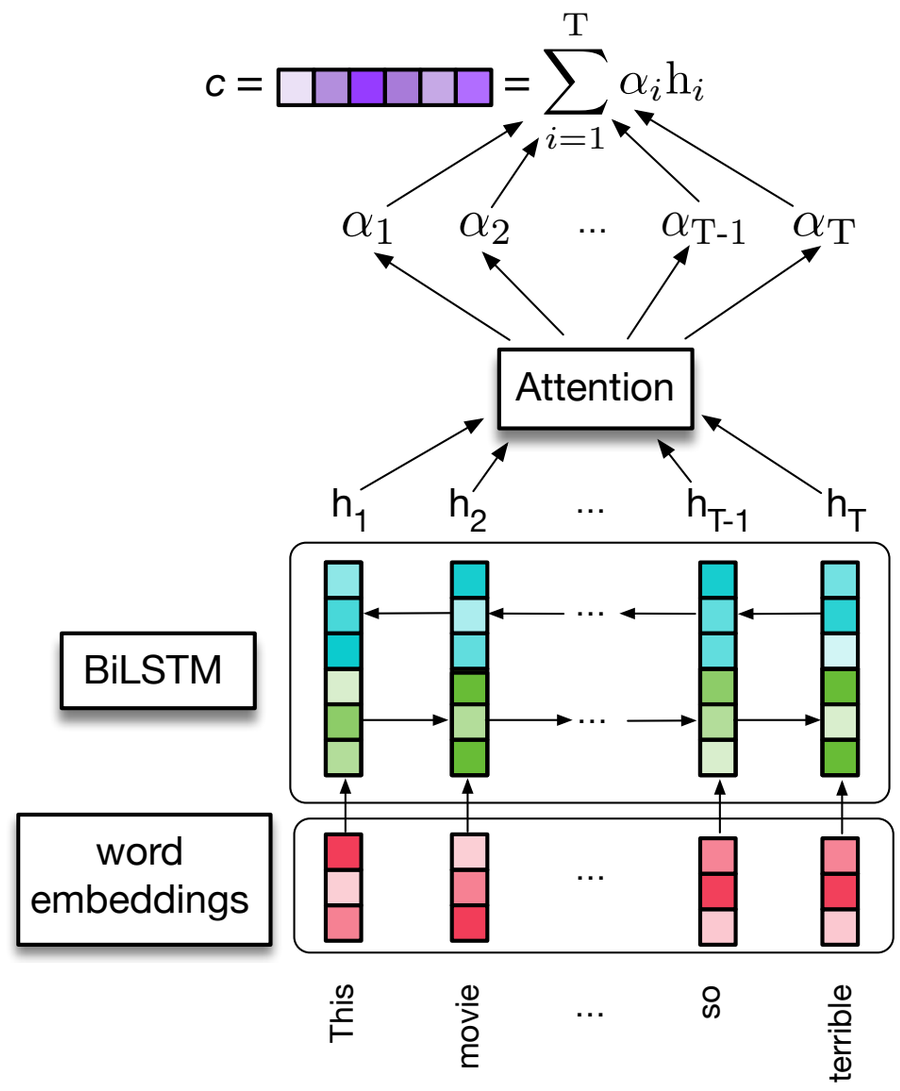
terrible

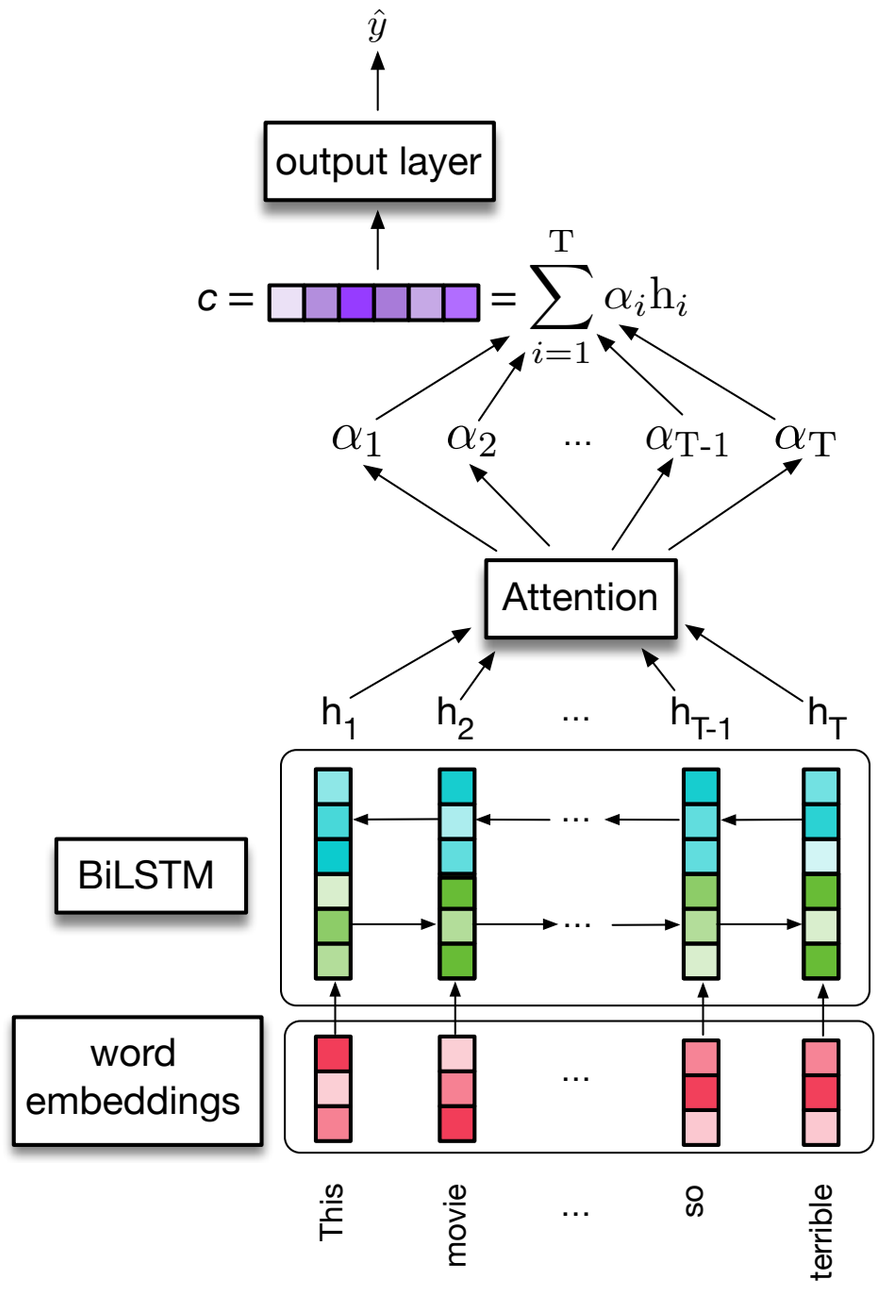

word
embeddings
This
movie
...
...
so
terrible


h1
h2
...
hT-1
hT
BiLSTM
word embeddings
This
movie
...
so
terrible


Attention
h1
h2
...
hT-1
hT
BiLSTM
word embeddings
This
movie
...
so
terrible


α_1
α_2
...
α_{T-1}
α_T
Attention
h_1
h_2
...
h_{T-1}
h_T
BiLSTM
word embeddings
This
movie
...
so
terrible


c =
$\sum_{i=1}^{T} \alpha_i h_i$
α_1
α_2
...
α_{T-1}
α_T
Attention
h_1
h_2
...
h_{T-1}
h_T
BiLSTM
word embeddings
This
movie
...
so
terrible


$\hat{y}$
output layer
$c =$
$= \sum_{i=1}^{T} \alpha_i h_i$
α_1 α_2 ... α_{T-1} α_T
Attention
h_1 h_2 ... h_{T-1} h_T
BiLSTM
word embeddings
This movie ... so terrible

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

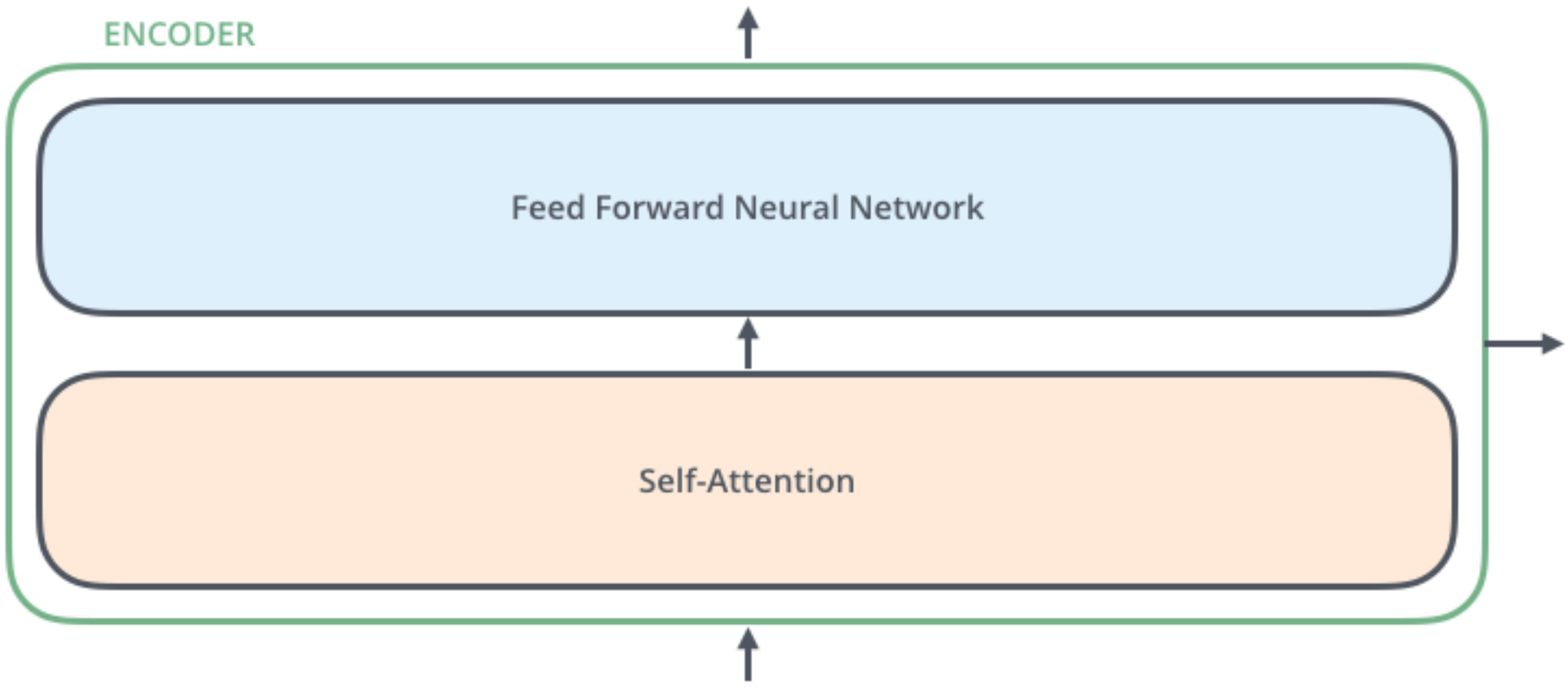
Łukasz Kaiser[*]
Google Brain
lukaszkaiser@google.com

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 less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Transformer block

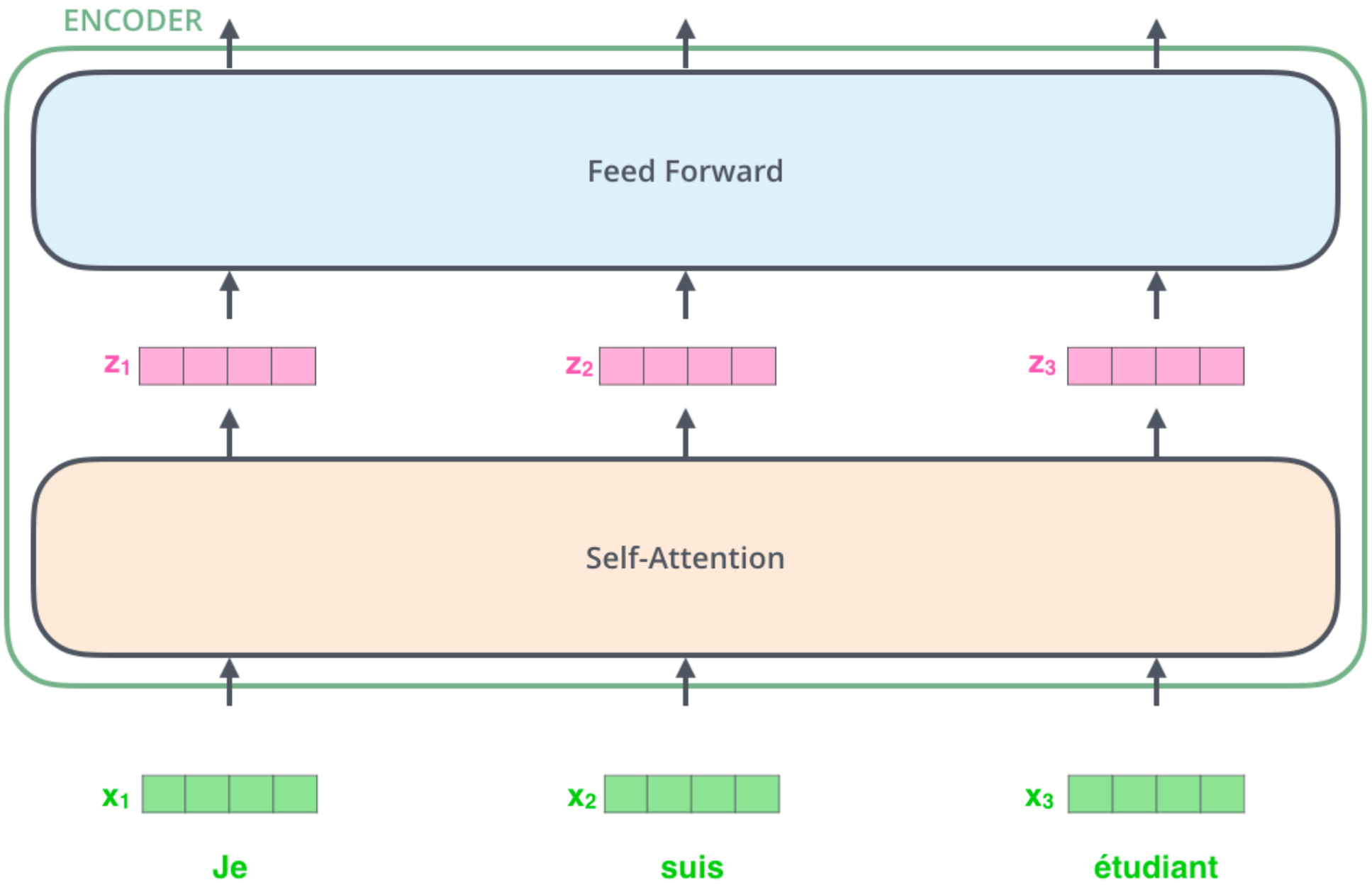


source: http://jalammar.github.io/illustrated-transformer/

First, embed

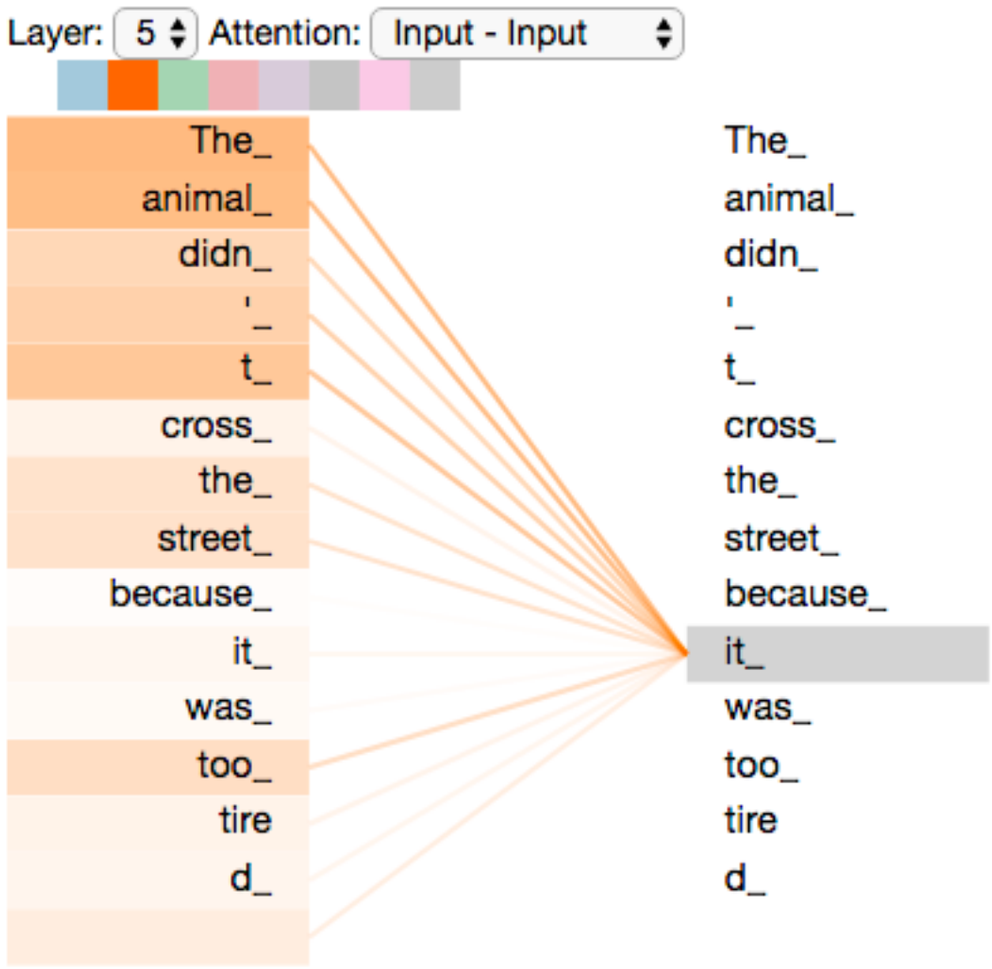
source: http://jalammar.github.io/illustrated-transformer/

Then transform

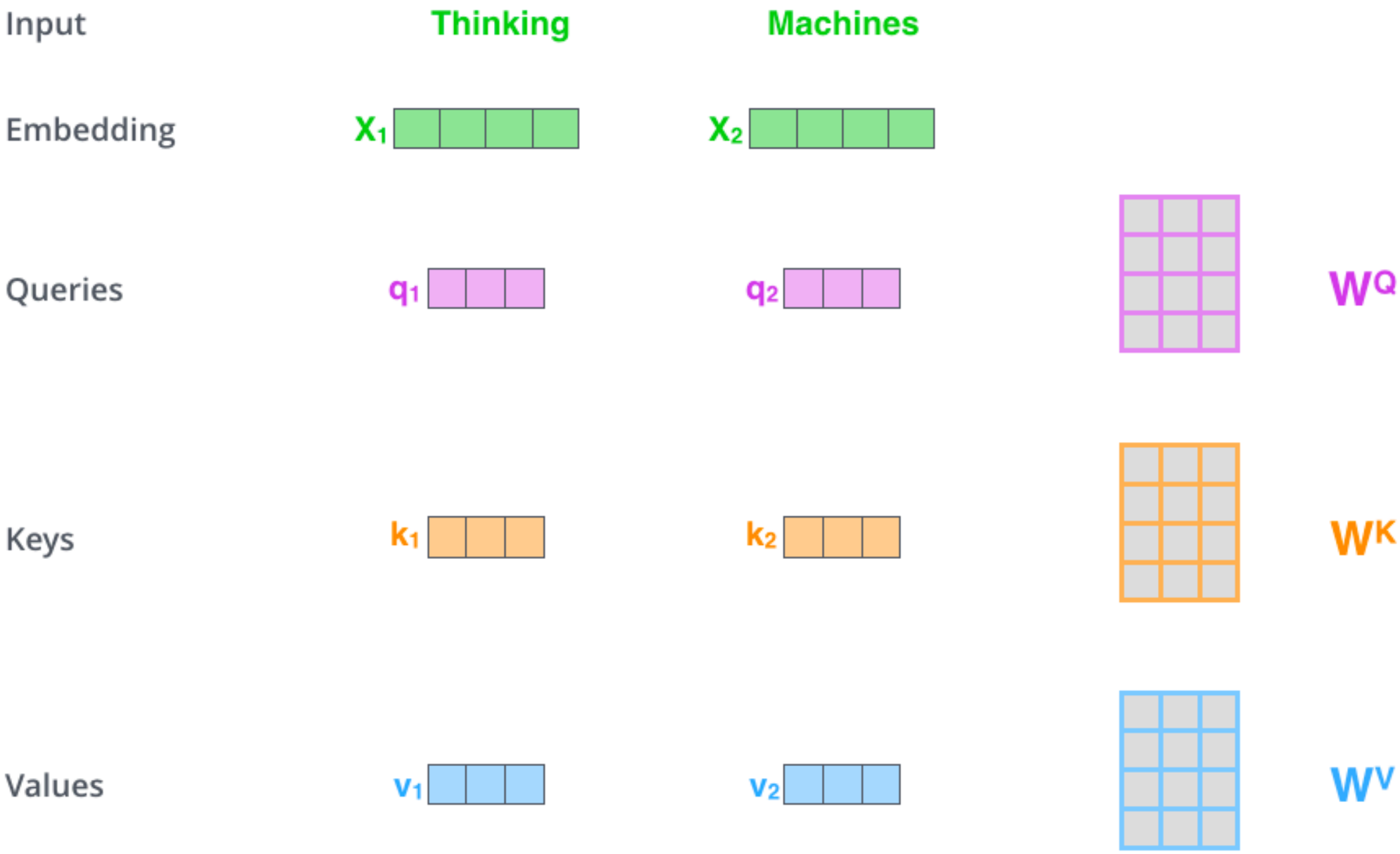


source: http://jalammar.github.io/illustrated-transformer/

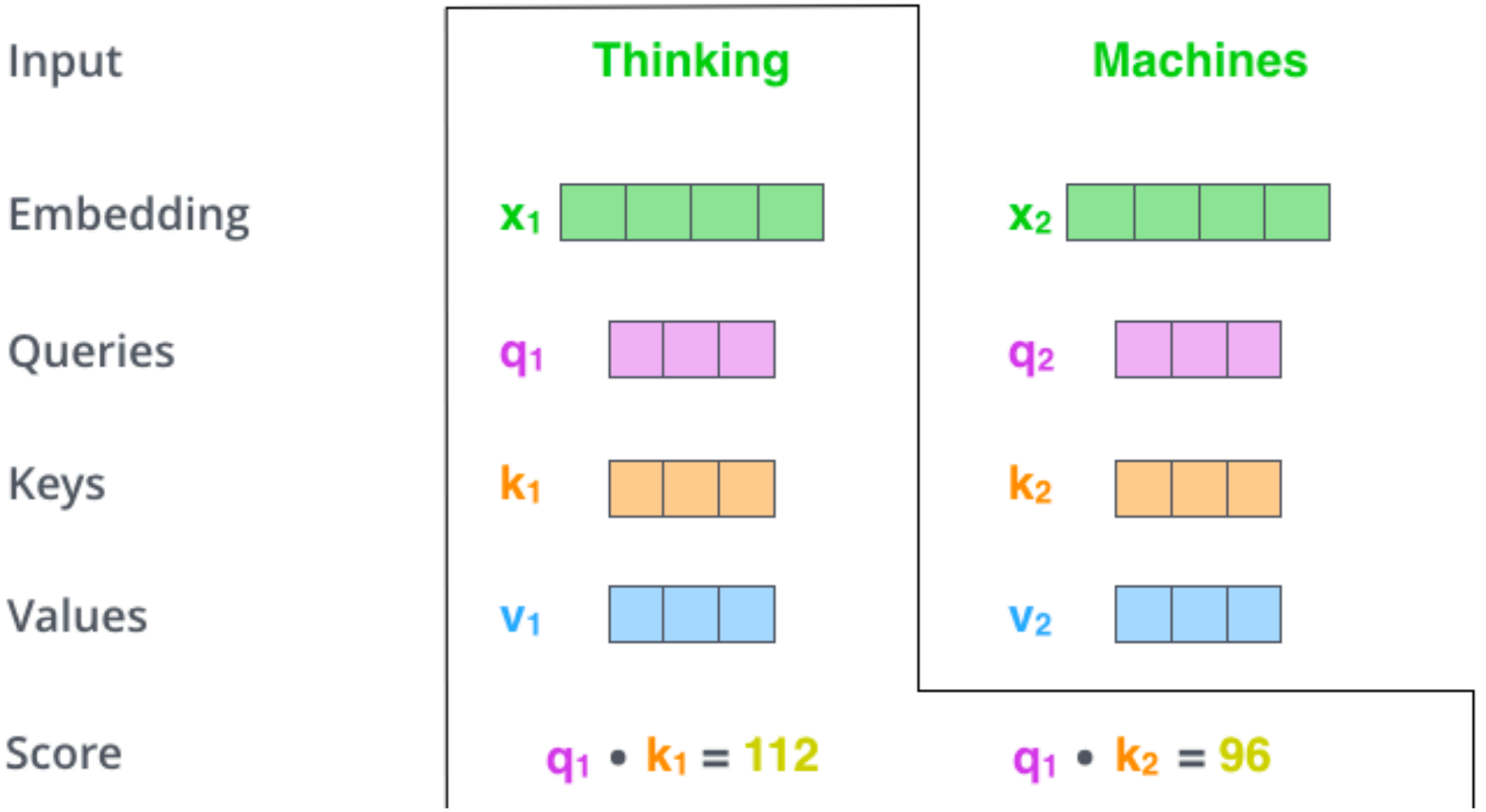
What is “self-attention”?



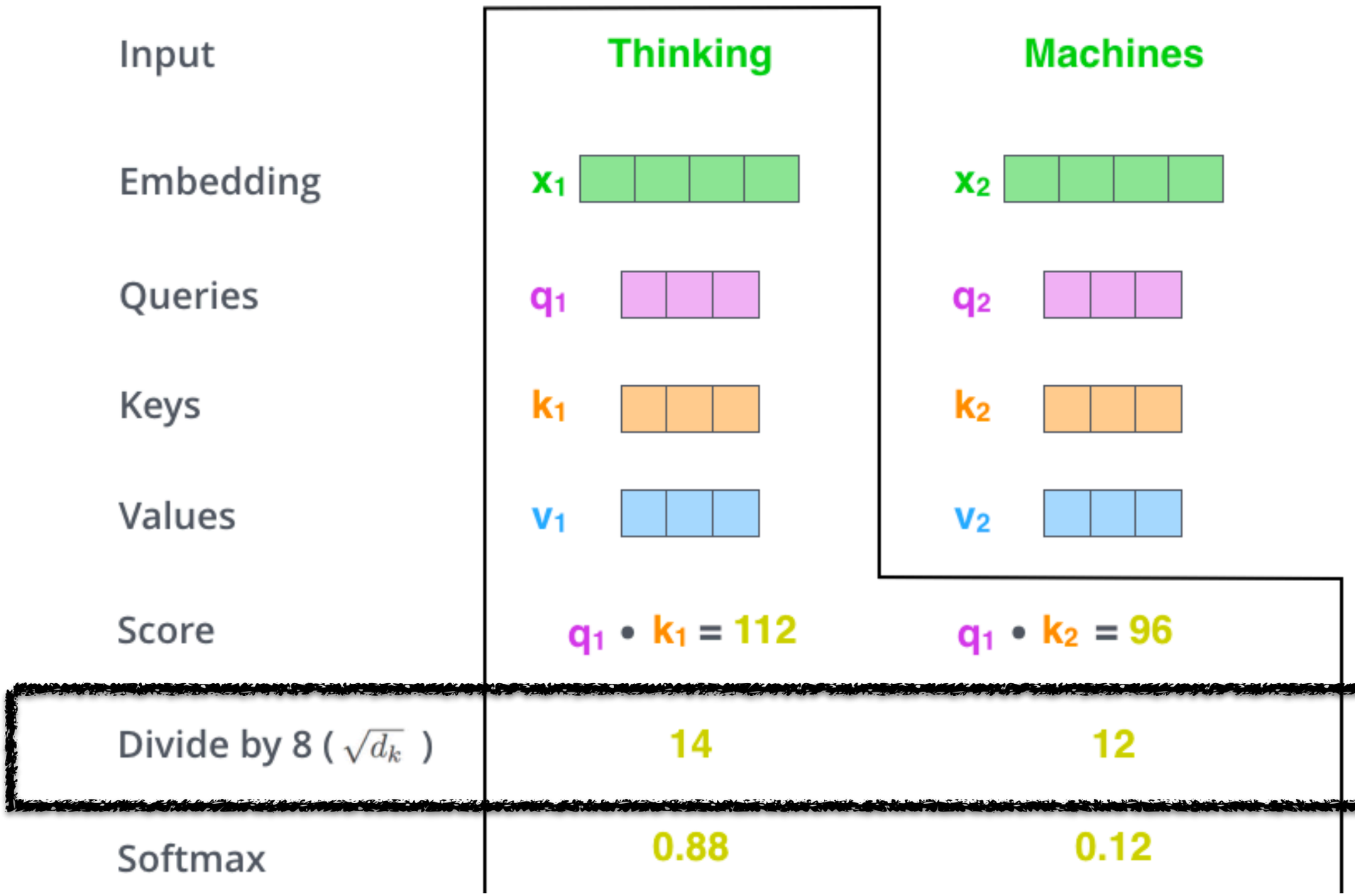
source: http://jalammar.github.io/illustrated-transformer/



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

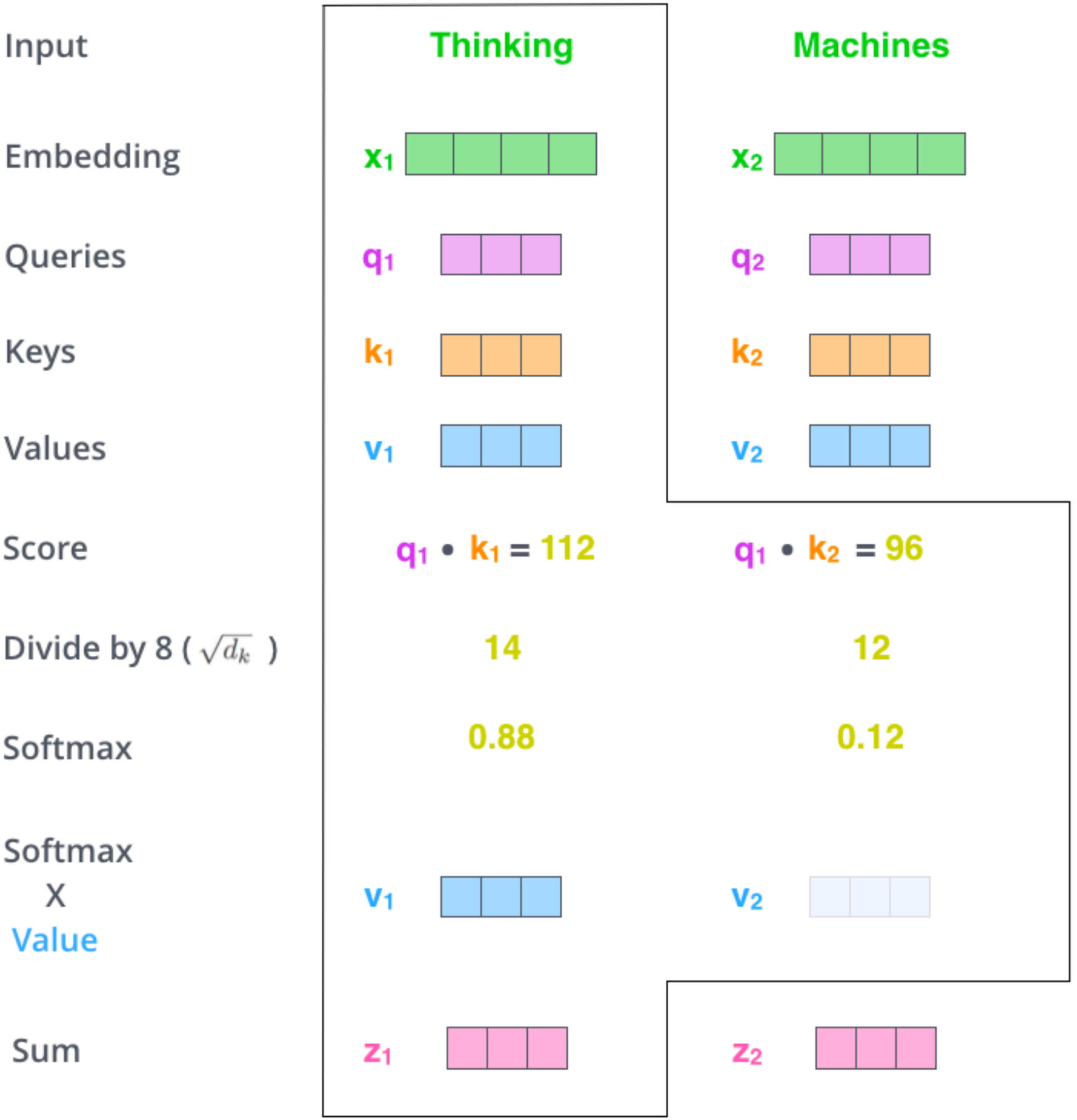


source: http://jalammar.github.io/illustrated-transformer/

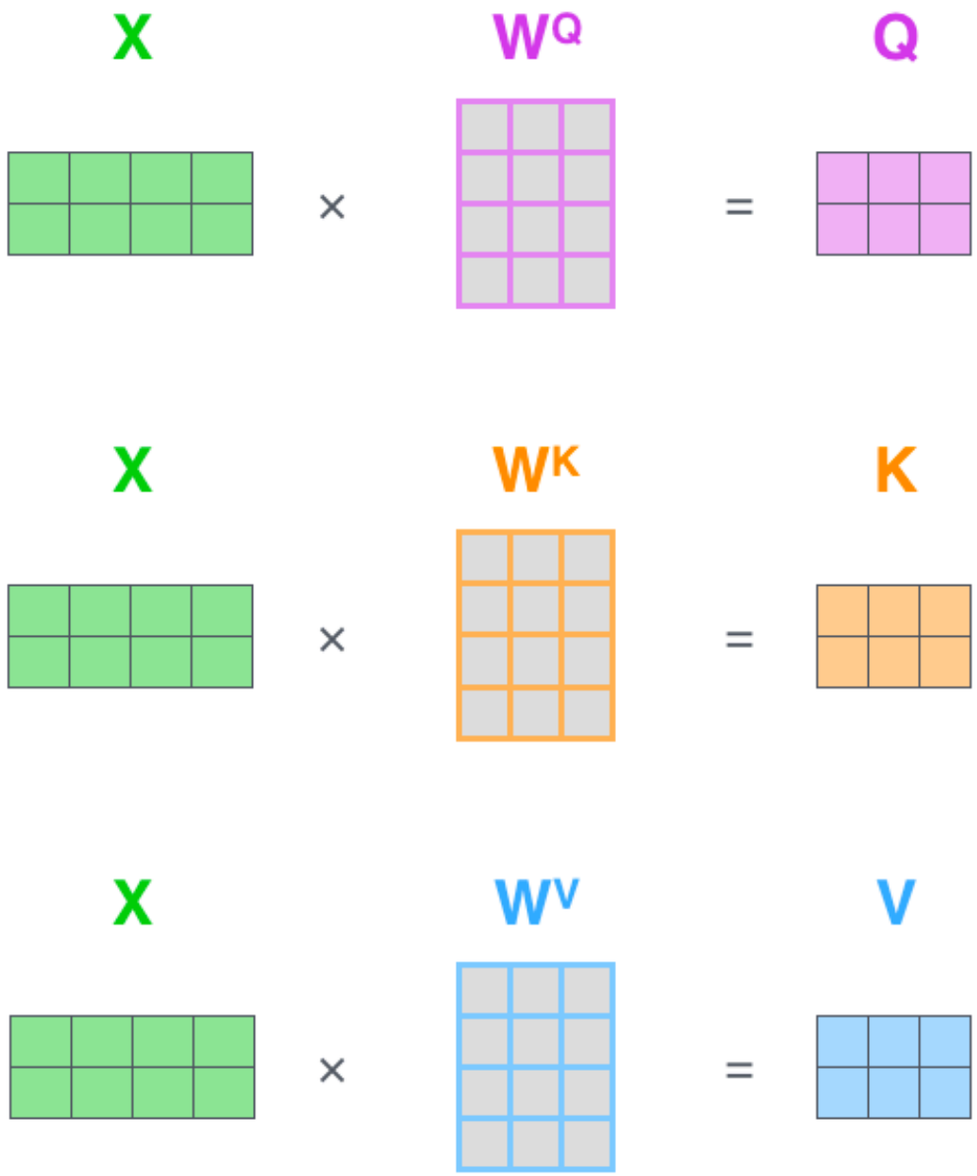


This one weird trick

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


Input
Thinking
Machines
Embedding
x1
x2
Queries
q1
q2
Keys
k1
k2
Values
v1
v2
Score
q1 • k1 = 112
q1 • k2 = 96
Divide by 8 (√dk)
14
12
Softmax
0.88
0.12
Softmax
X
Value
v1
v2
Sum
z1
z2

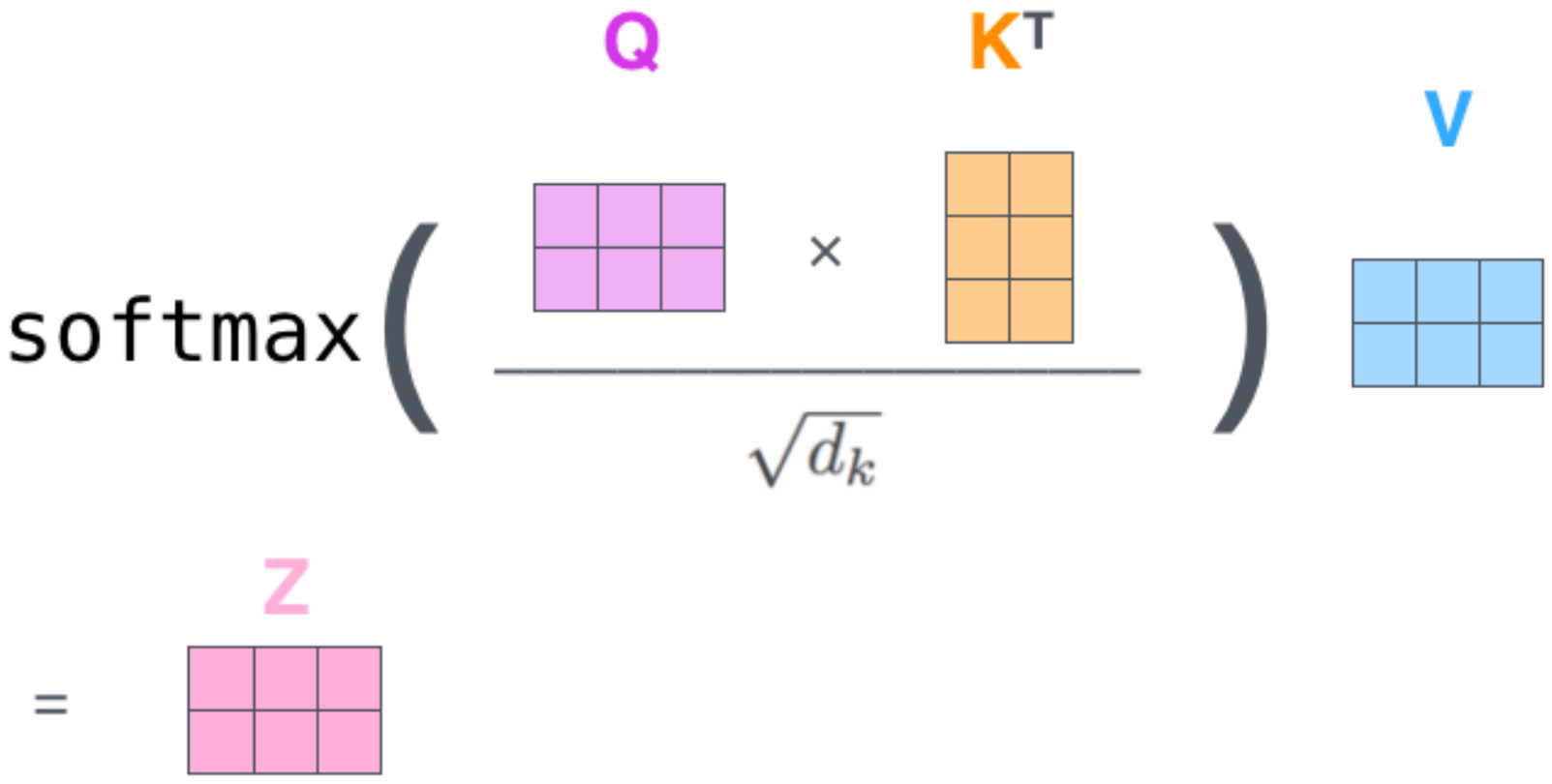
In matrices



Learned

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

In matrices



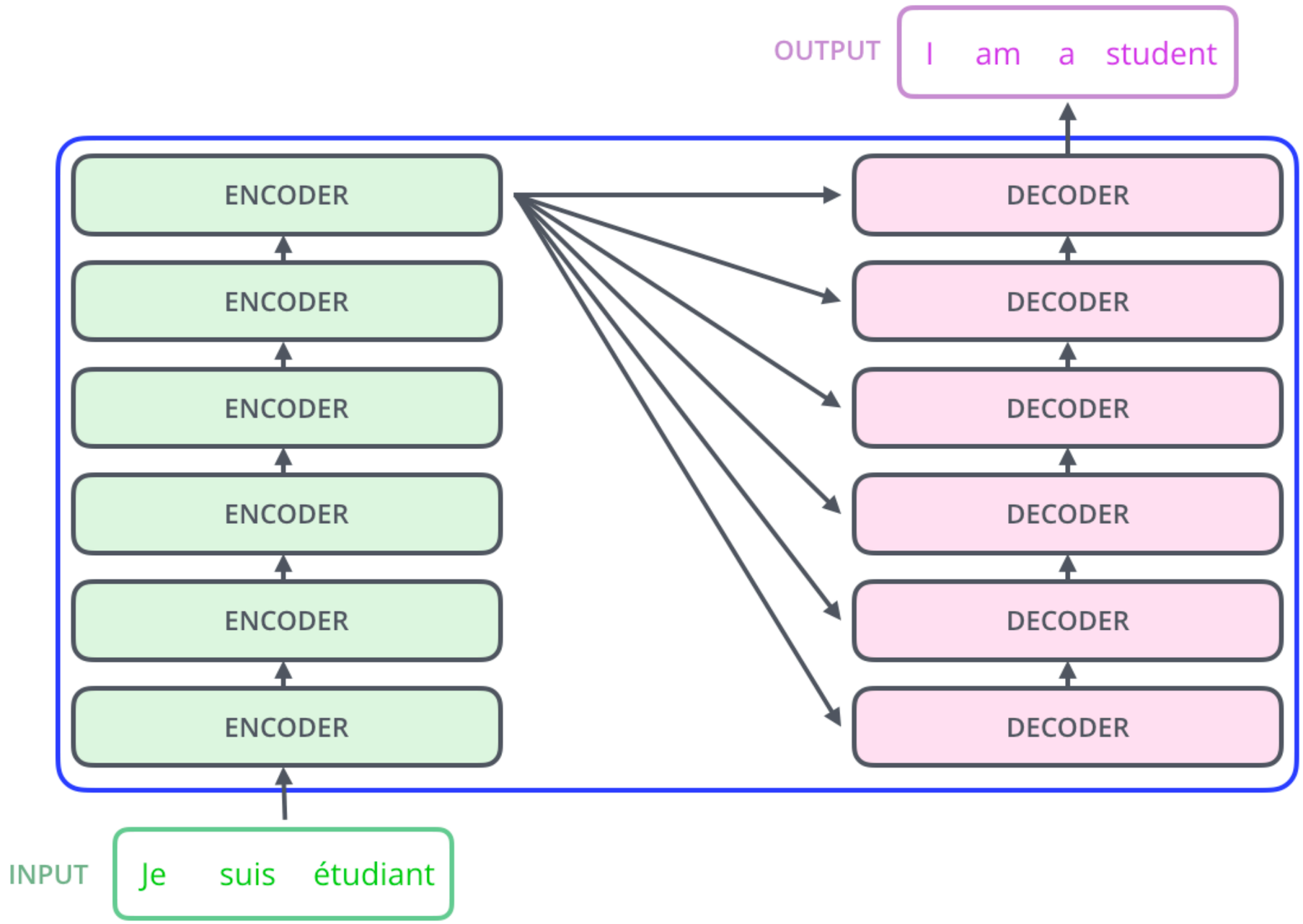
source: http://jalammar.github.io/illustrated-transformer/

Let’s implement…

[notebook TODOs 1 & 2]

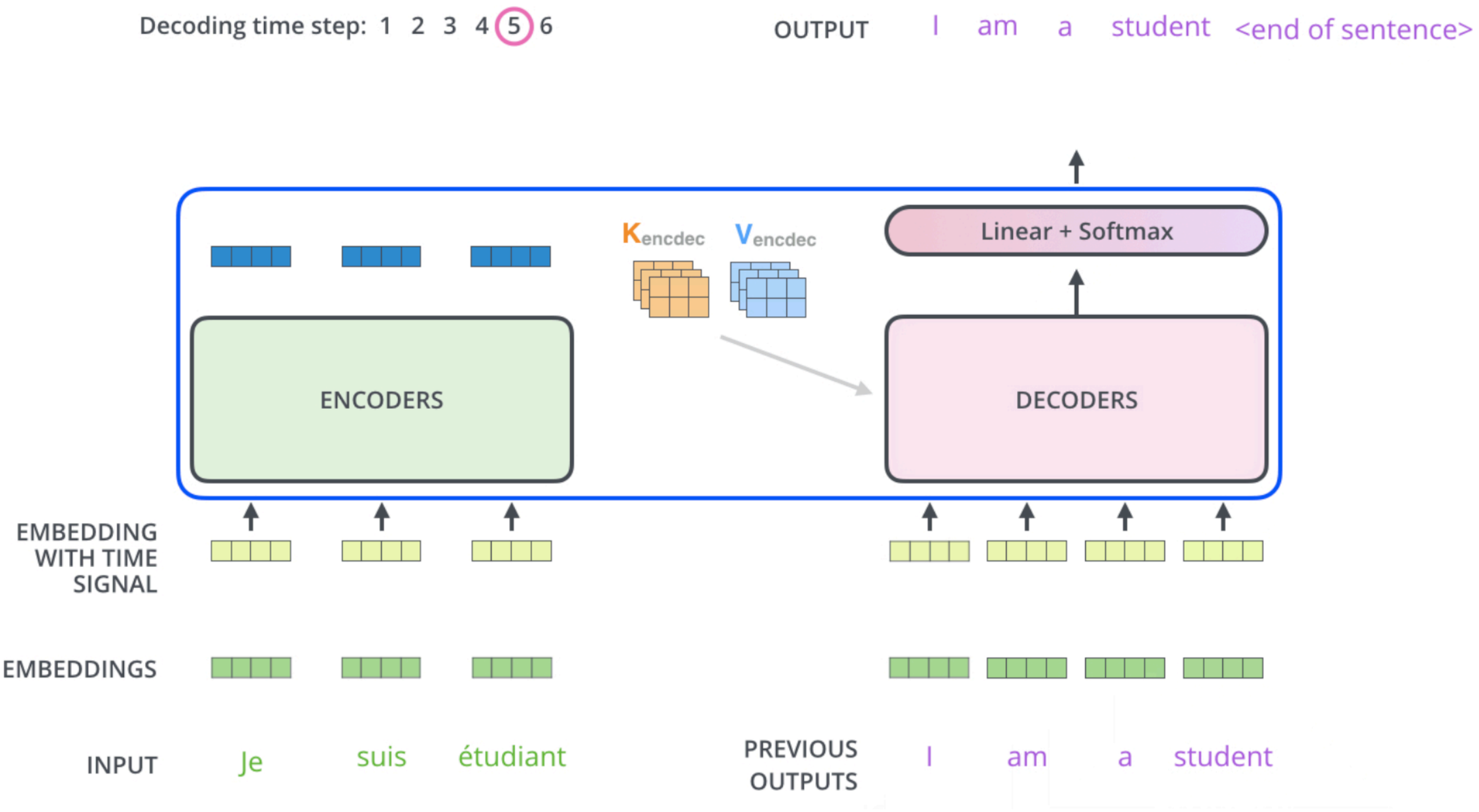
OK, but what is it used for?

Translation



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

Translation



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

Language modeling

https://talktotransformer.com/

BERT

BERT

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Pre-train (self-supervise) then *fine-tune*: A winning combo

SQuAD1.1 Leaderboard

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar et al. '16)	82.304	91.221
1 Oct 05, 2018	BERT (ensemble) *Google AI Language* https://arxiv.org/abs/1810.04805	**87.433**	**93.160**
2 Sep 09, 2018	nlnet (ensemble) *Microsoft Research Asia*	85.356	91.202
3 Jul 11, 2018	QANet (ensemble) *Google Brain & CMU*	84.454	90.490

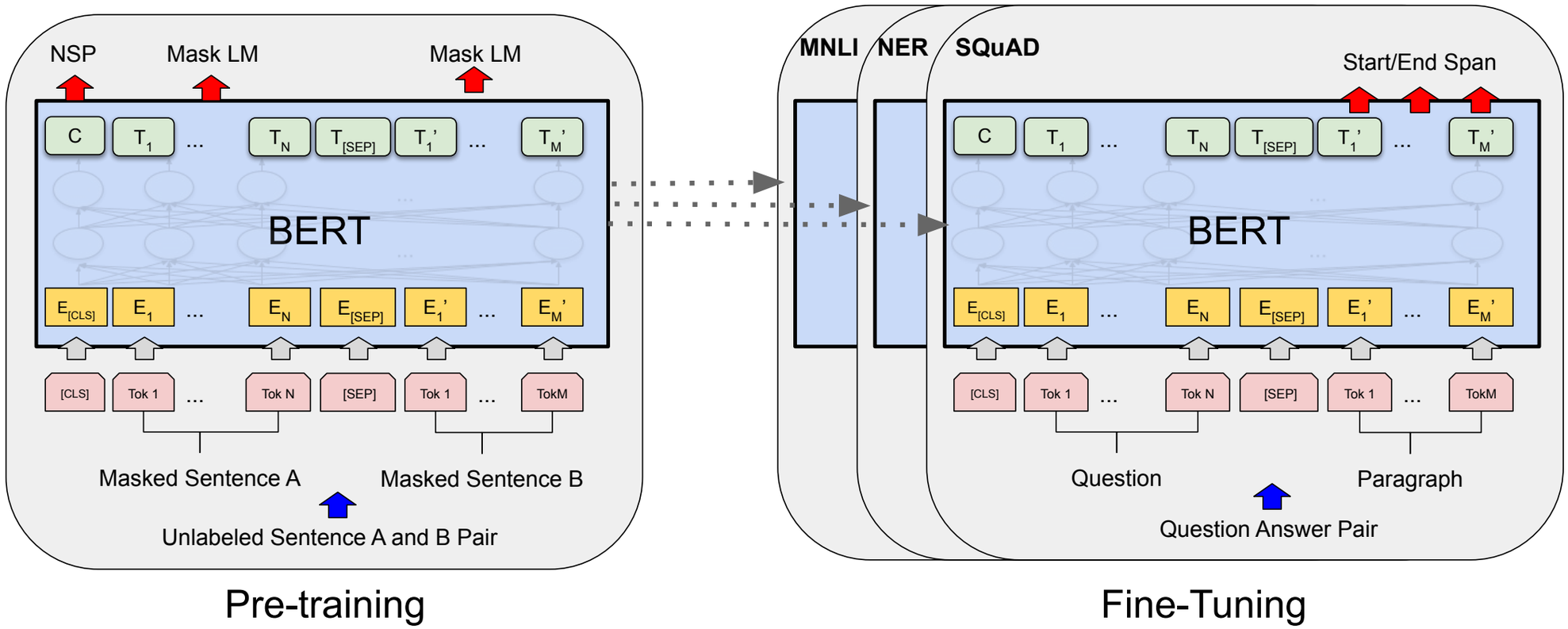
Glue Benchmark Leaderboard

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**91.1**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**81.9**

This is a thing now

A Primer in BERTology: What we know about how BERT works

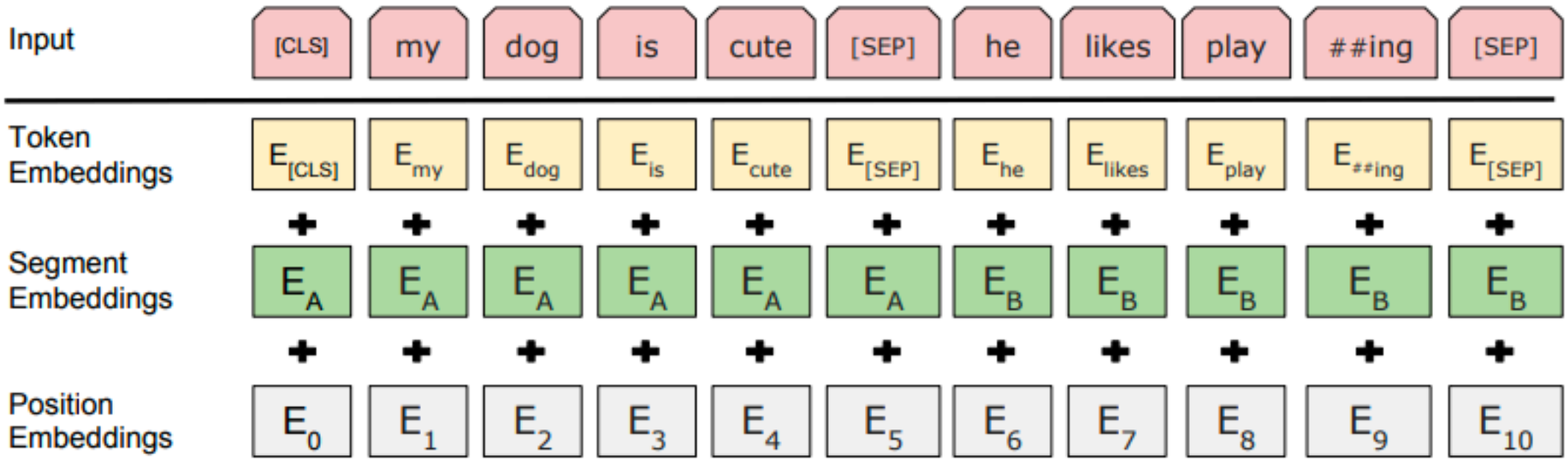
Anna Rogers, Olga Kovaleva, Anna Rumshisky
Department of Computer Science, University of Massachusetts Lowell
Lowell, MA 01854
{arogers, okovalev, arum}@cs.uml.edu



BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Self-Supervise an Encoder



BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Self-Supervise an Encoder

The cat is very cute

Self-Supervise an Encoder

The cat is very cute

X	*The [MASK] is very cute*
y	*cat*

Let’s implement …

[notebook TODO 3]

BERT details we did not consider

- BERT actually uses *word-pieces* rather than entire words

BERT details we did not consider

- BERT actually uses *word-pieces* rather than entire words
- Also uses “positional” embeddings in the inputs to give a sense of “location” in the sequence

BERT details we did not consider

- BERT actually uses *word-pieces* rather than entire words
- Also uses “positional” embeddings in the inputs to give a sense of “location” in the sequence
- Multiple self-attention “heads”

BERT details we did not consider

- BERT actually uses *word-pieces* rather than entire words
- Also uses “positional” embeddings in the inputs to give a sense of “location” in the sequence
- Multiple self-attention “heads”
- Deeper (12+ layers)
- Residual + layer norms (prevents explosions/NaNs)

For a more detailed implementation …

- See Sasha Rush's excellent "annotated transformer": http://nlp.seas.harvard.edu/2018/04/03/attention.html