UNDERSTANDING TEXT WITH BERT

Scaleway

24/07/2019, Paris NLP Meetup

OUTLINE

- What does it mean to **understand text**?
- **Sequence to sequence** models in the “old days”
- The age of **Transformers**
- **BERT**
- **Demo**: how do we use BERT

REFERENCES

The original papers:

Attention is All You Need (2017)

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2018)

Visualizing A Neural Machine Translation Model by Jay Alammar

Very nice visualisations of Sequence to Sequence models

The Illustrated Transformer by Jay Alammar

Transformer architecture illustrated

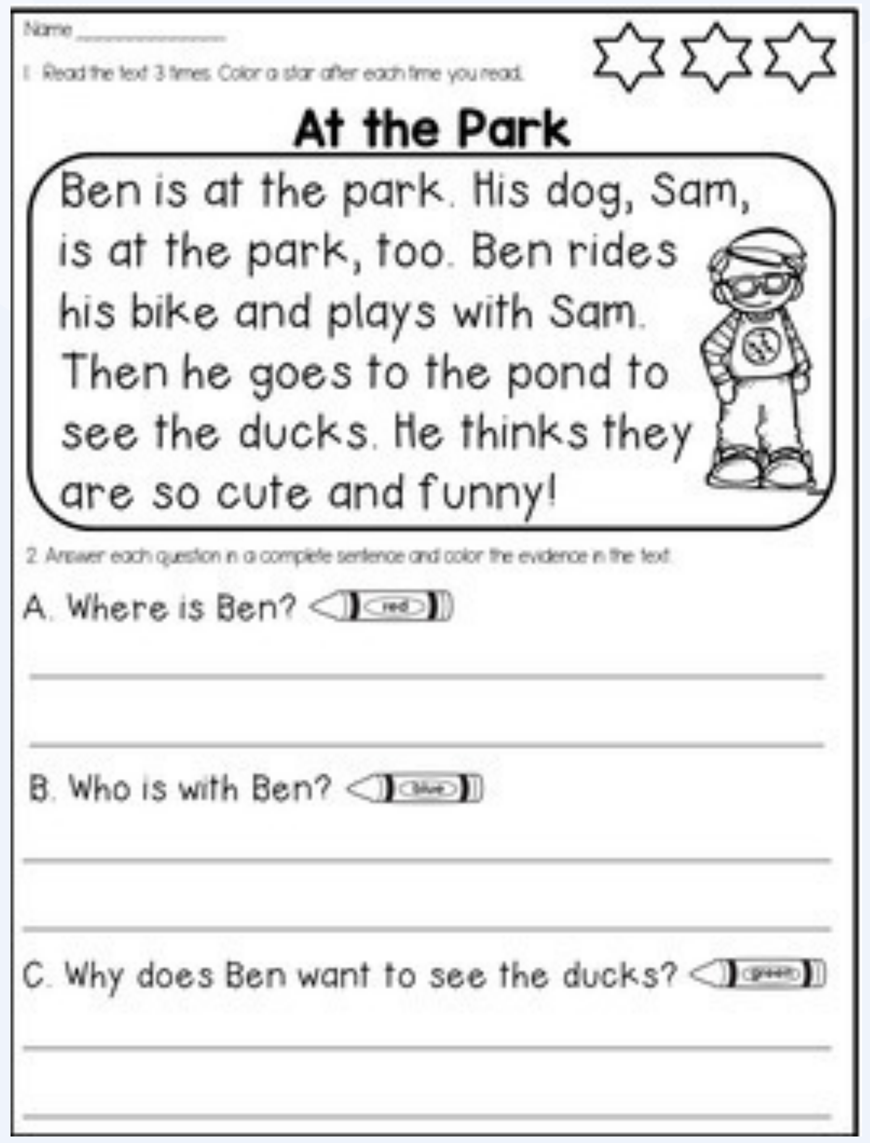
The Annotated Transformer by Harvard NLP

A walkthrough of the Pytorch implementation

READING COMPREHENSION

READING COMPREHENSION ~ QUESTION ANSWERING

- Humans are trained in reading comprehension by answering questions based on a text they read
- Machine Reading Comprehension → Question Answering → Span Extraction


Name ____________

1. Read the text 3 times. Color a star after each time you read.

At the Park

Ben is at the park. His dog, Sam, is at the park, too. Ben rides his bike and plays with Sam. Then he goes to the pond to see the ducks. He thinks they are so cute and funny!

2. Answer each question in a complete sentence and color the evidence in the text.

A. Where is Ben? red

B. Who is with Ben? blue

C. Why does Ben want to see the ducks? green

Context: “Architecturally, the school has a Catholic character. Atop the Main Building's gold dome is a golden statue of the Virgin Mary. Immediately in front of the Main Building and facing it, is a copper statue of Christ with arms upraised with the legend "Venite Ad Me Omnes". Next to **the Main Building** is the Basilica of the Sacred Heart. Immediately behind the basilica is the Grotto, a Marian place of prayer and reflection. It is a replica of the grotto at Lourdes, France where the Virgin Mary reputedly appeared to Saint Bernadette Soubirous in 1858. At the end of the main drive (and in a direct line that connects through 3 statues and the Gold Dome), is a simple, modern stone statue of Mary.”

Question: “The Basilica of the Sacred heart at Notre Dame is beside to which structure?”

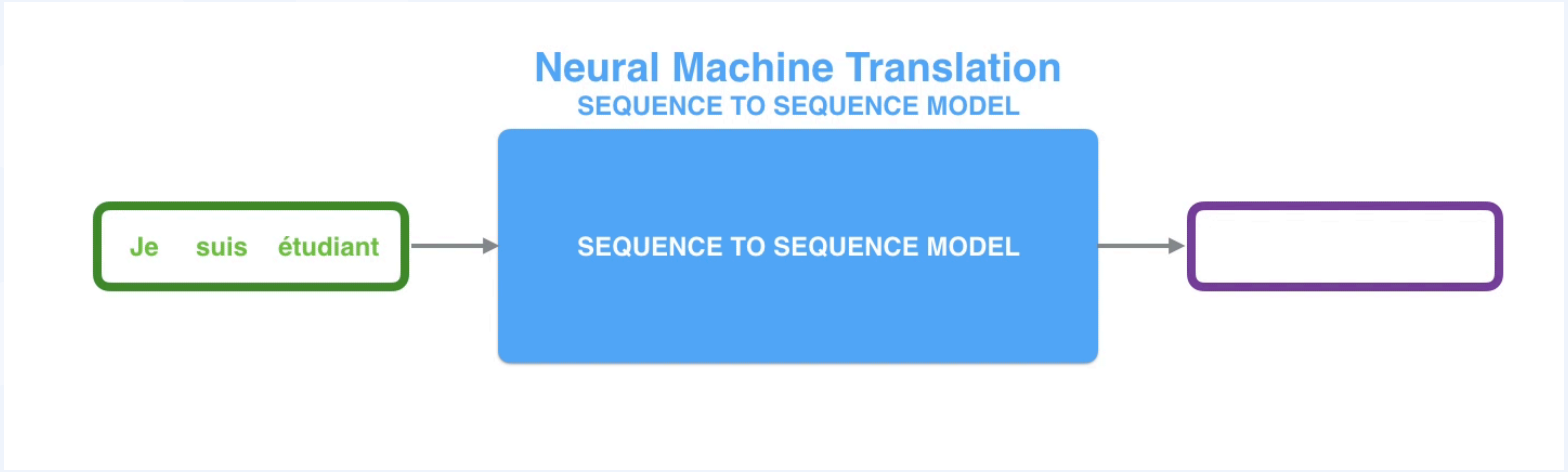
Answer: start_position: 49, end_position: 51

- Stanford Question Answering Dataset (SQuAD) dataset

A LONG TIME AGO* IN A GALAXY [NOT SO] FAR AWAY

** circa 2014*

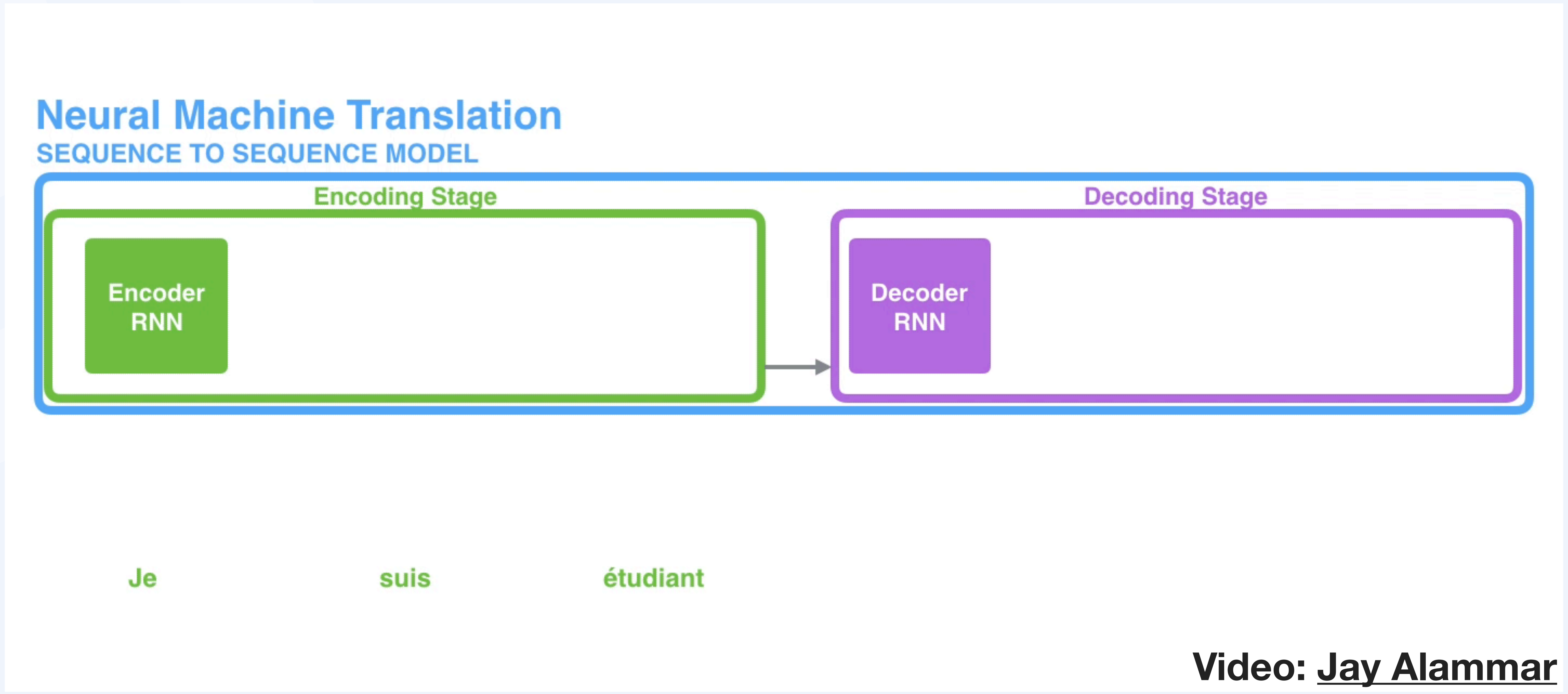
NLP: SEQUENCE TO SEQUENCE


Neural Machine Translation
SEQUENCE TO SEQUENCE MODEL
Je suis étudiant
SEQUENCE TO SEQUENCE MODEL

Video: Jay Alammar

- Seq2Seq: both the input and the output are **sequences**
- Sequence that consists of **tokens** in some **vocabulary**
- **Order matters**
- Used across many different NLP tasks: Machine translation (used by Google Translate), Question answering, Part of speech tagging, etc

RNN ENCODER-DECODER



- Each input token gets fed into the **Encoder**, the resulting **hidden** state gets passed on
- The **final hidden** state to come out of the Encoder ~ meaning of the entire input sequence
- The **Decoder** generates outputs one token at a time
- Often previous outputs serve as additional inputs for the **Decoder**

PROBLEMS WITH THE RNN APPROACH

1. The meaning of the entire input sequence gets represented by a single fixed-size vector

"Je suis étudiant."

"It was a wrong number that started it, the telephone ringing three times in the dead of night, and the voice on the other end asking for someone he was not."

Solution: the Attention mechanism

2. Sequential computation → non parallelizable within the training examples

Solution: CNNs instead of RNNs

PROBLEMS WITH THE RNN APPROACH

1. The meaning of the entire input sequence gets represented by a single fixed-size vector

“Je suis étudiant.”

“It was a wrong number that started it, the telephone ringing three times in the dead of night and the voice on the other end asking for someone he was not.”

Solution: the Attention mechanism

2. Sequential computation → non parallelizable within the training examples

Solution: CNNs instead of RNNs

PROBLEMS WITH THE RNN APPROACH

1. The meaning of the entire input sequence gets represented by a single fixed-size vector

“Je suis étudiant.”

“It was a wrong number that started it, the telephone ringing three times in the dead of night, and the voice on the other end asking for someone he was not.”

Solution: the Attention mechanism

2. Sequential computation → non parallelizable within the training examples

Solution: CNNs instead of RNNs

PROBLEMS WITH THE RNN APPROACH

1. The meaning of the entire input sequence gets represented by a single fixed-size vector

“Je suis étudiant.”

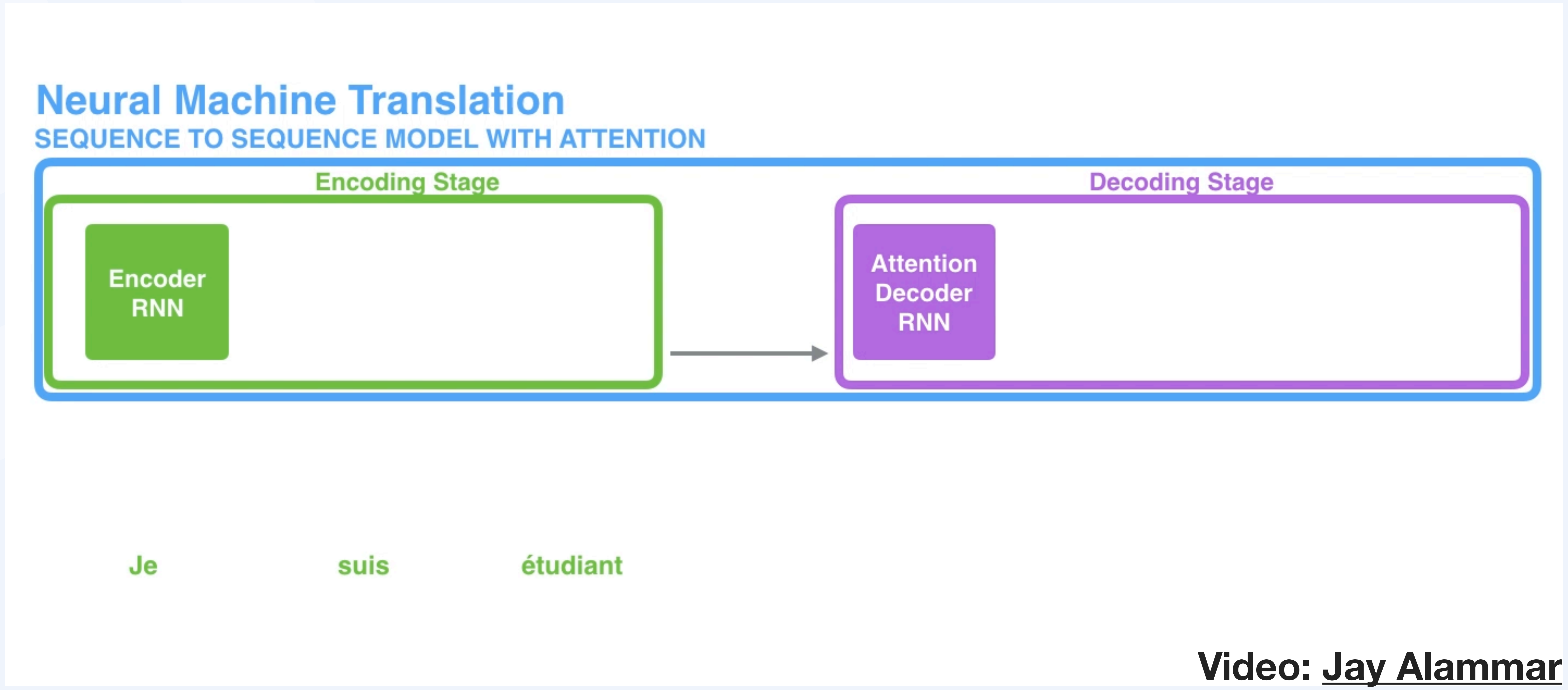
“It was a wrong number that started it, the telephone ringing three times in the dead of night, and the voice on the other end asking for someone he was not.”

Solution: **the Attention mechanism**

2. Sequential computation → non parallelizable within the training examples

Solution: CNNs instead of RNNs

ATTENTION MECHANISM



Video: Jay Alammar

- Attention allows the model to *attend to* the relevant parts of the input sequence
- The **Encoder** passes *all* the hidden states to the **Decoder**
- **Decoder**: softmax over those input hidden states to determine what is most relevant
- **RNN architecture**

PROBLEMS WITH THE RNN APPROACH

1. The meaning of the entire input sequence gets represented by a single fixed-size vector.

"Je suis étudiant."

"It was a wrong number that started it, the telephone ringing three times in the dead of night, and the voice on the other end asking for someone he was not."

Solution: **the Attention mechanism**

2. Sequential computation → non parallelizable within the training examples

Solution: CNNs instead of RNNs

PROBLEMS WITH THE RNN APPROACH

1. The meaning of the entire input sequence gets represented by a single fixed-size vector.

"Je suis étudiant."

"It was a wrong number that started it, the telephone ringing three times in the dead of night, and the voice on the other end asking for someone he was not."

Solution: **the Attention mechanism**

2. Sequential computation → non parallelizable within the training examples

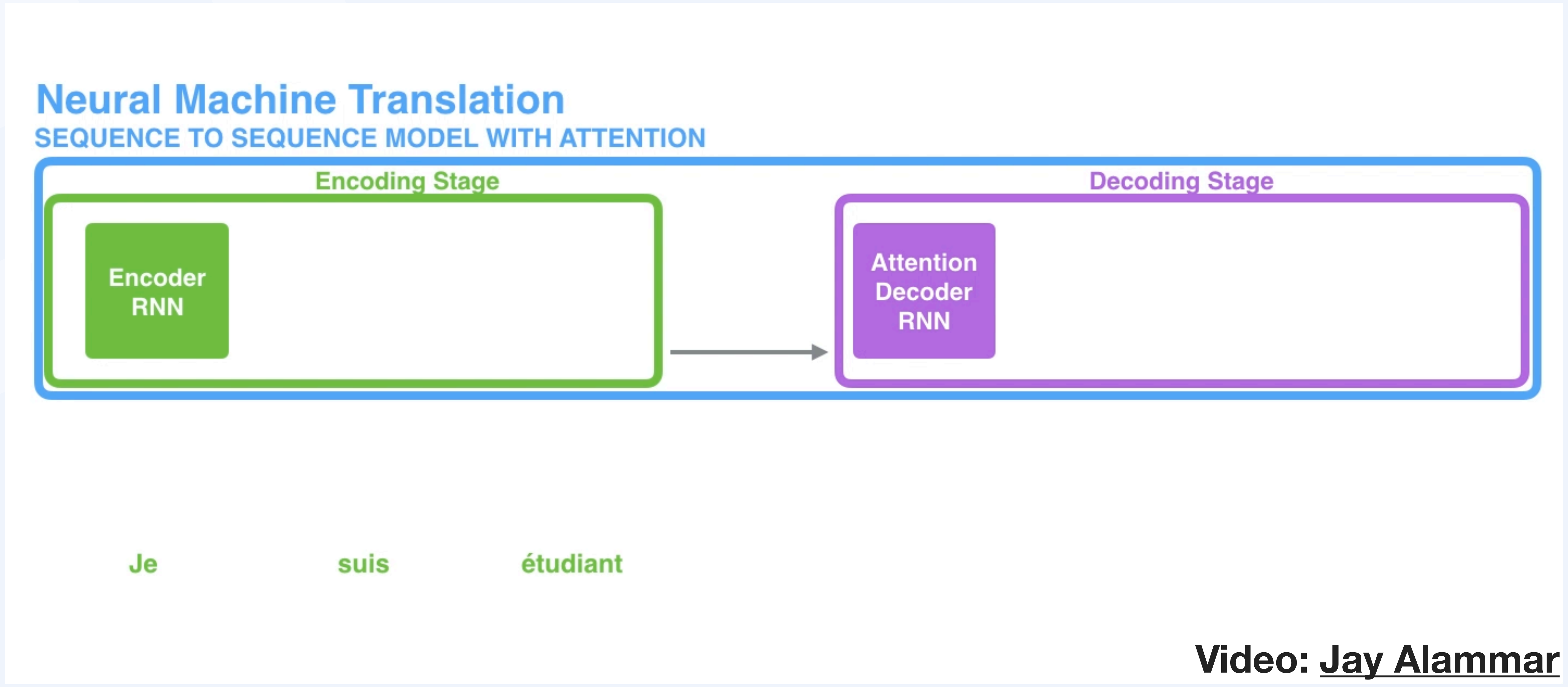
Solution: CNNs instead of RNNs

Better solution?..

ATTENTION IS ALL YOU NEED

Google, 2017

TRANSFORMERS



Video: Jay Alammar

- Keep the Attention and **throw away the RNN**
- **Three types of Attention**: Encoder-Decoder Attention, Encoder Self-Attention and Decoder Self-Attention
- To keep track of order in the input sequence: use **positional encoding**

TRANSFORMERS

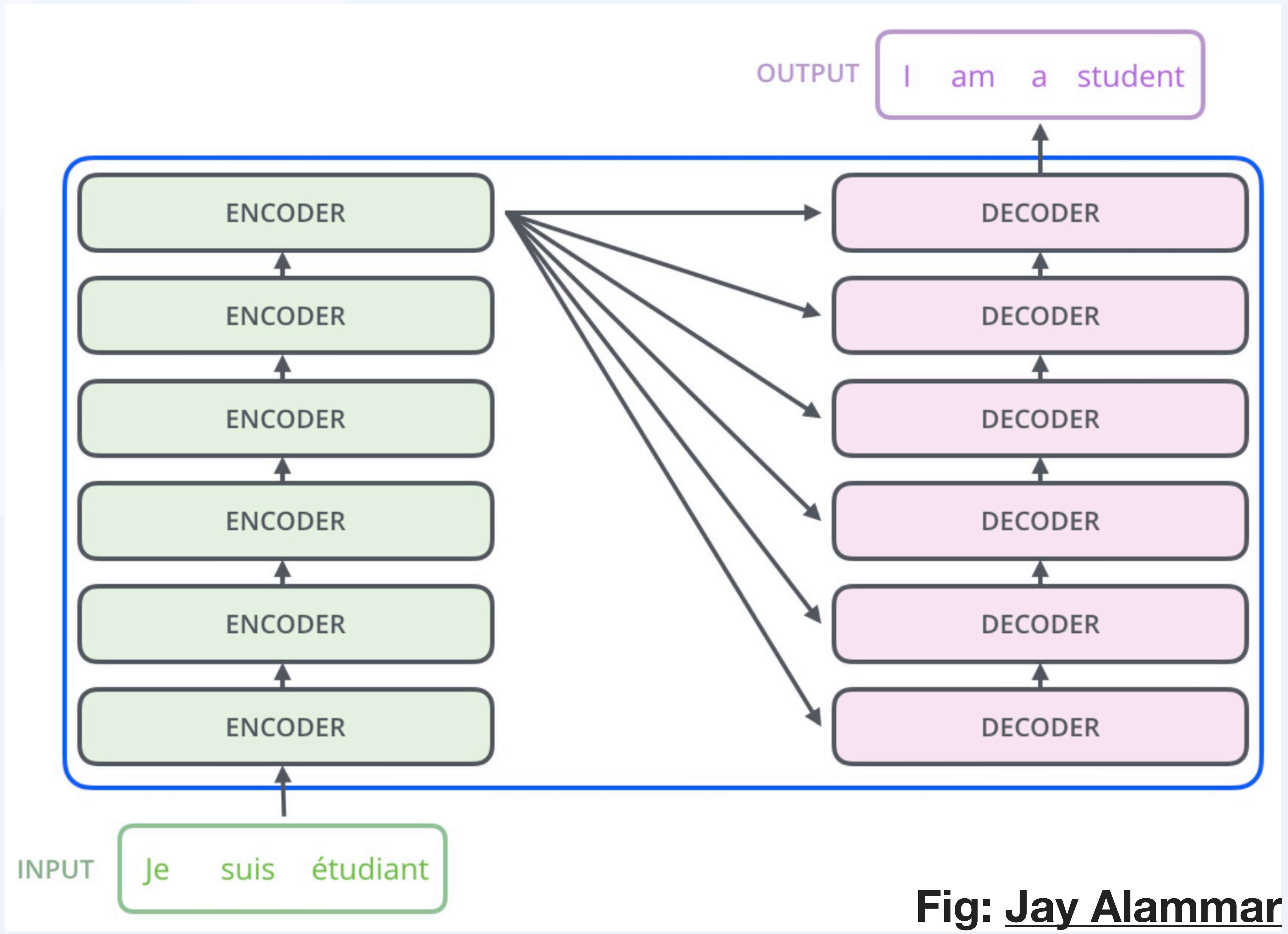


Fig: Jay Alammar

- Keep the Attention and **throw away the RNN**
- **Three types of Attention**: Encoder-Decoder Attention, Encoder Self-Attention and Decoder Self-Attention
- To keep track of order in the input sequence: use **positional encoding**

TRANSFORMERS

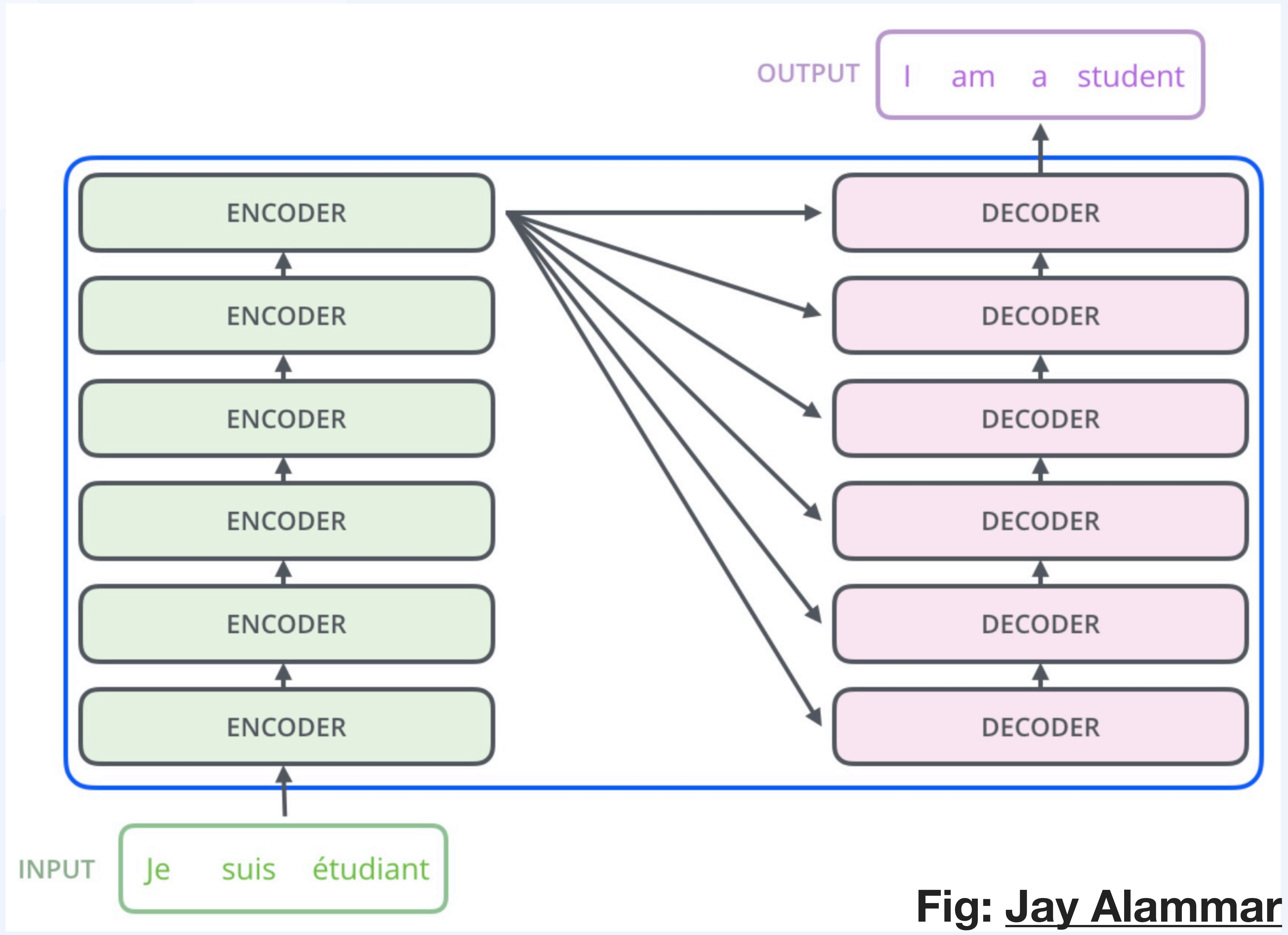


Fig: Jay Alammar

Encoder-Decoder Attention

- Each input goes through **Encoder**
- Final output of **Encoder** = input for each layer of the **Decoder**

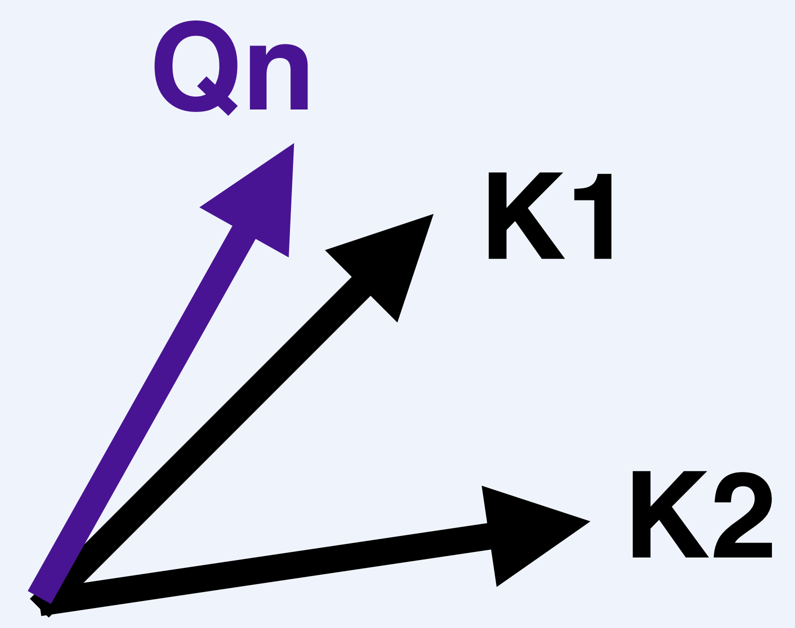
- Keep the Attention and **throw away the RNN**
- **Three types of Attention**: Encoder-Decoder Attention, Encoder Self-Attention and Decoder Self-Attention
- To keep track of order in the input sequence: use **positional encoding**

ATTENTION: K, V, Q

- **X**: input to the Attention layer
- Three new spaces: **K** (keys), **V** (values) and **Q** (queries)
- $\mathbf{K} = \mathbf{X}\,\mathbf{W}^{\mathbf{K}}$ $\mathbf{V} = \mathbf{X}\,\mathbf{W}^{\mathbf{V}}$ $\mathbf{Q} = \mathbf{X}\,\mathbf{W}^{\mathbf{Q}}$
- What you need to learn while training: matrices $\mathbf{W}^{\mathbf{K}}$,$\mathbf{W}^{\mathbf{V}}$ and $\mathbf{W}^{\mathbf{Q}}$
- How Attention is used: $\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$
- Encoder-Decoder Attention:
  - **Keys** and **Values** from the **Encoder**
  - **Queries** from the **Decoder**

ATTENTION (THE WAY I THINK ABOUT IT)

- A grossly oversimplified explanation of **K**, **V** and **Q**



- Example:
  Input sequence: Je suis étudiant
  X1 = “je”; X2 = “suis”; X3 = “étudiant”
- **K**eys: index the kinds of information that are contained in the input sequence
  E.g. K1 = subject; K2 = description of the subject
- **V**alues: correspond to the tokens **X** in the input sequence, e.g. V1→X1; V2→X3 or V2→X2+X3
- **Q**ueries: what is the information we are looking for (at some point we refer to as *n*)
  Let’s say it is the subject, then Qn*K1 is maximised and V1 is the value that gets the **attention**

SELF-ATTENTION

- Self-Attention for the Encoder and the Decoder

- Each input in the sequence gets represented in terms of the other inputs in the sequence

  The **animal** did not cross the road because **it** was too tired.

  The animal did not cross the **road** because **it** was too wide.

- In the **Decoder Self-Attention**, the outputs get represented only in terms of the outputs produced *so far* (uni-directional)

POSITIONAL ENCODING

- Without RNN, how do we keep track of **order inside the input sequence**?
- Fixed positional encoding using *sin* and *cos:*

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

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

pos = position of the token in the sequence
d_{model} = dimensionality of the word embeddings **and** positional encodings (512 in the Transformer paper)

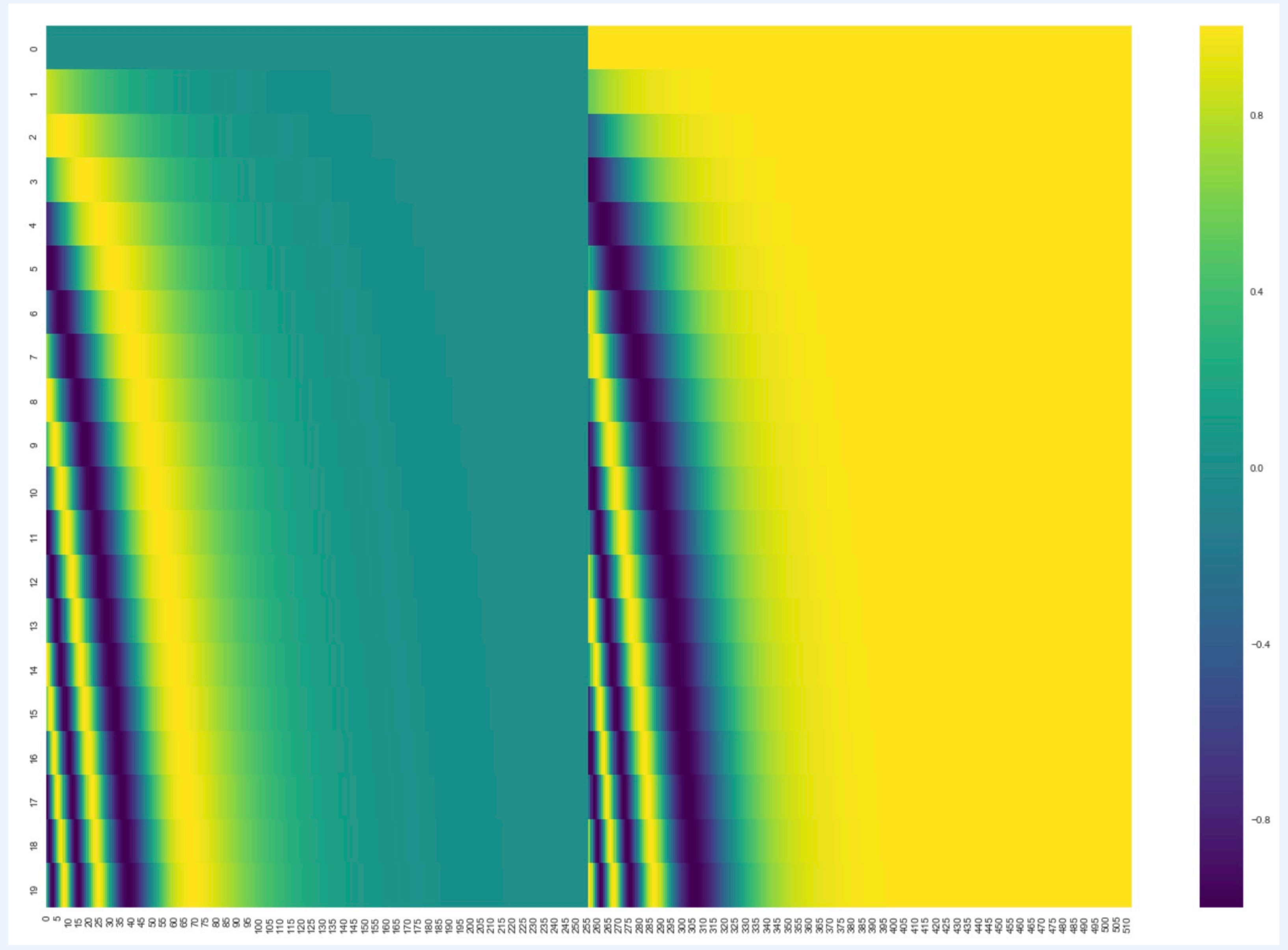
POSITIONAL ENCODING

- Without RNN, how do we keep track of **order inside the input sequence**?
- Fixed positional encoding using *sin* and *cos:*

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

  pos = position of the token in the sequence
  d_{model} = dimensionality of the word embeddings **and** positional encodings (512 in the Transformer paper)

Left side:	*sin*;	right side: *cos*
Vertical:	20 positions (0..19)	
Horizontal:	d_{model} (0…511)	



- Benefit of using periodic functions over absolute position encoding: at inference time, we can make sense of sentences longer than any in our training set

Fig: Jay Alammar

BERT

Bidirectional Encoder Representations from Transformers

BERT: BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS

- Why Bi-Directional?

  BERT keeps the **Encoder**, whose **Self-Attention is bi-directional**

- BERT: the Encoder of a Transformer pre-trained on two types of tasks
  - Masked Language Model (MLM):
    - Mask 15% of words in your corpus
    - Add a classification layer of top of the Encoder to find a masked word
  - Next Sentence Prediction:
    - Feed sequences A and B into the Transformer
    - Does B come after A?

DEMO

See the Jupyter notebook here