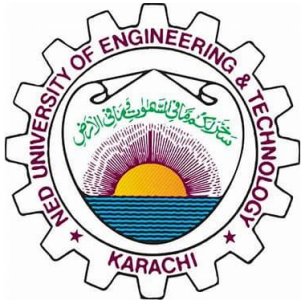


BERT

(Bi-Directional Encoder Representations from Transformer)

A Short Introduction

By Tariq Jamil
PGD-DS (BII)

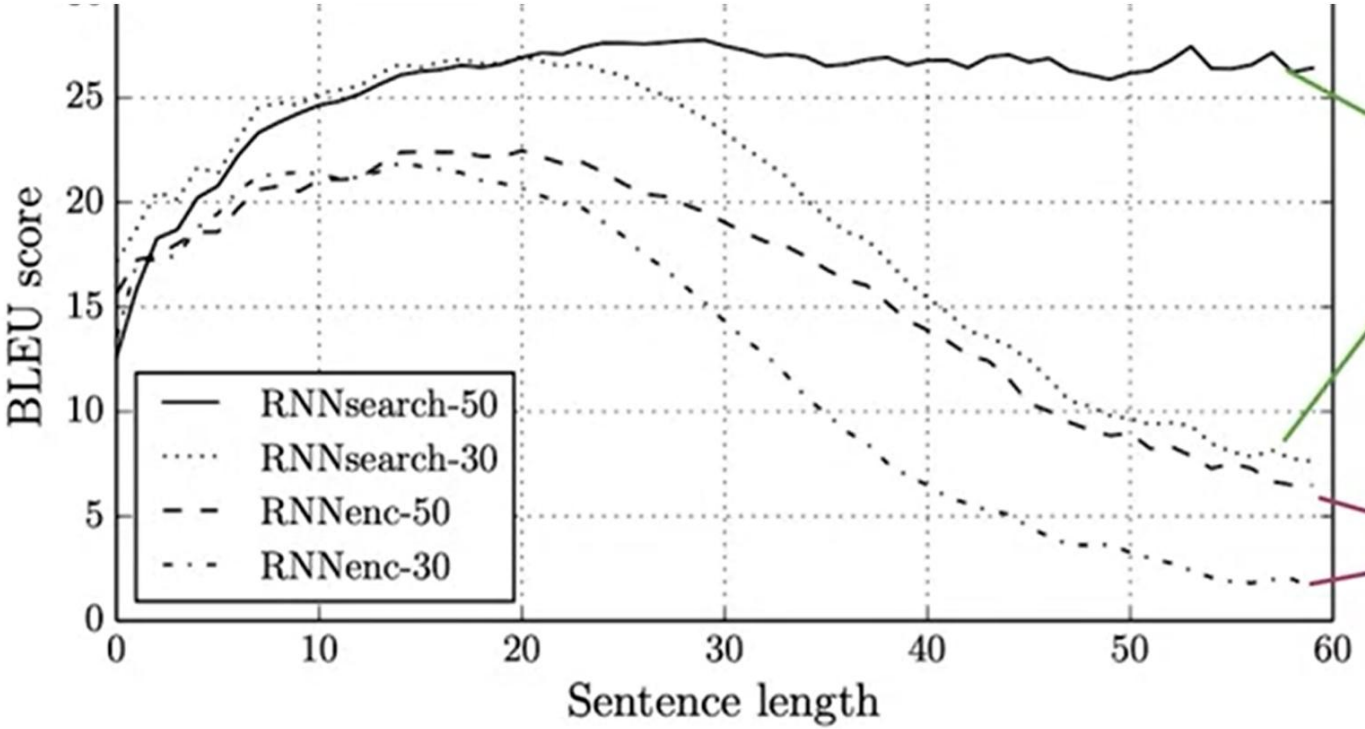
Traditional Approach / LSTM

Seq2Seq shortcomings

- Variable-length sentences + fixed-length memory =

- As sequence size increases, model performance decreases

Seq2Seq Models Comparison

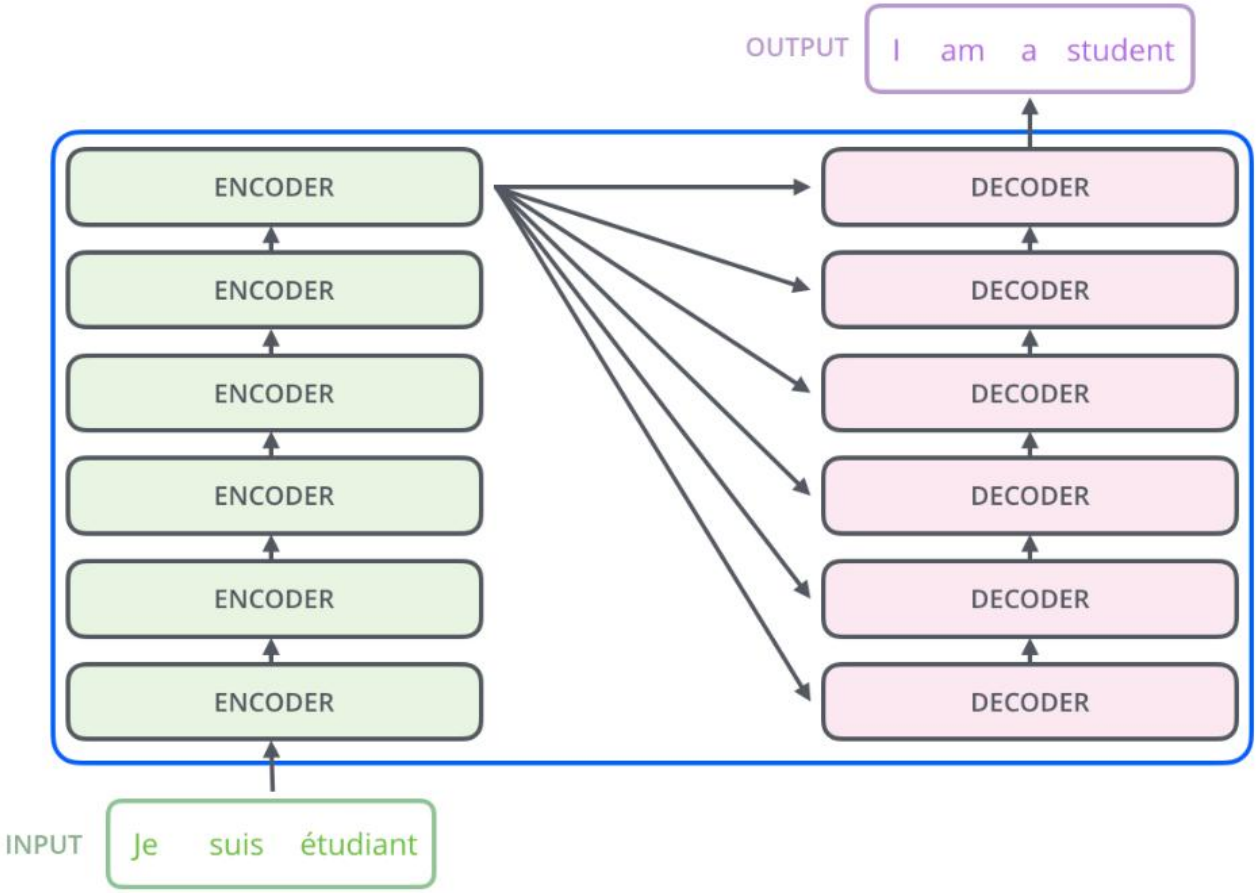


Seq2Seq with Attention

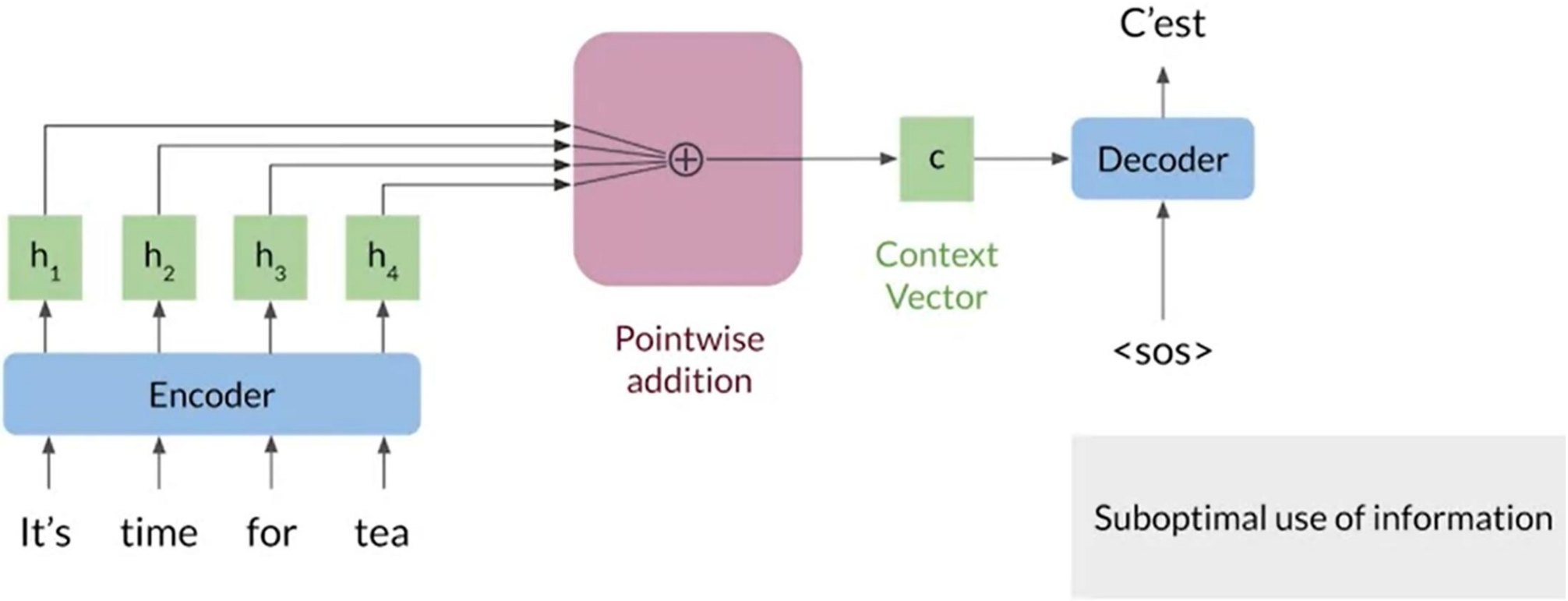
Traditional Seq2Seq Models

Greater BLEU is better

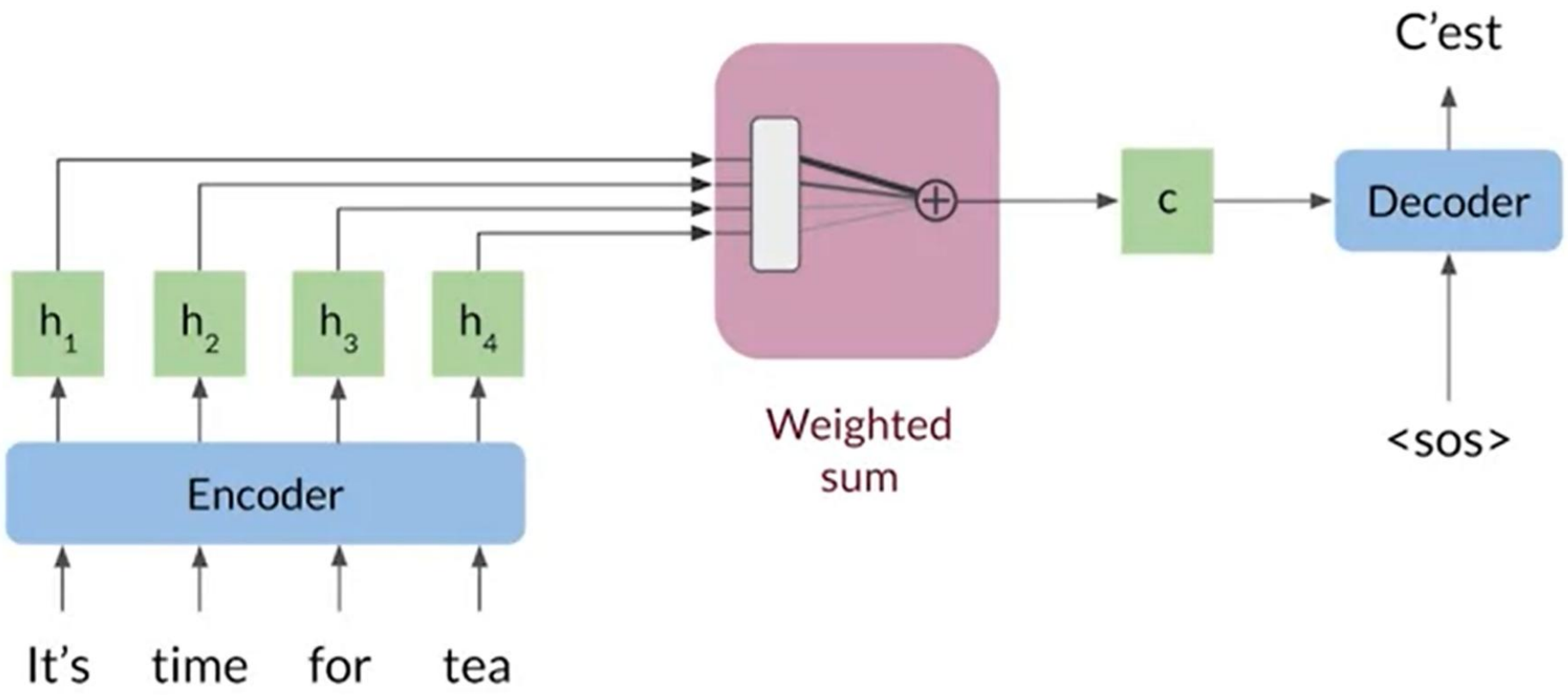
Transformer | Block Diagram



Transformers | Traditional Approach



Transformers | with Self Attention



Understanding - Self Attention

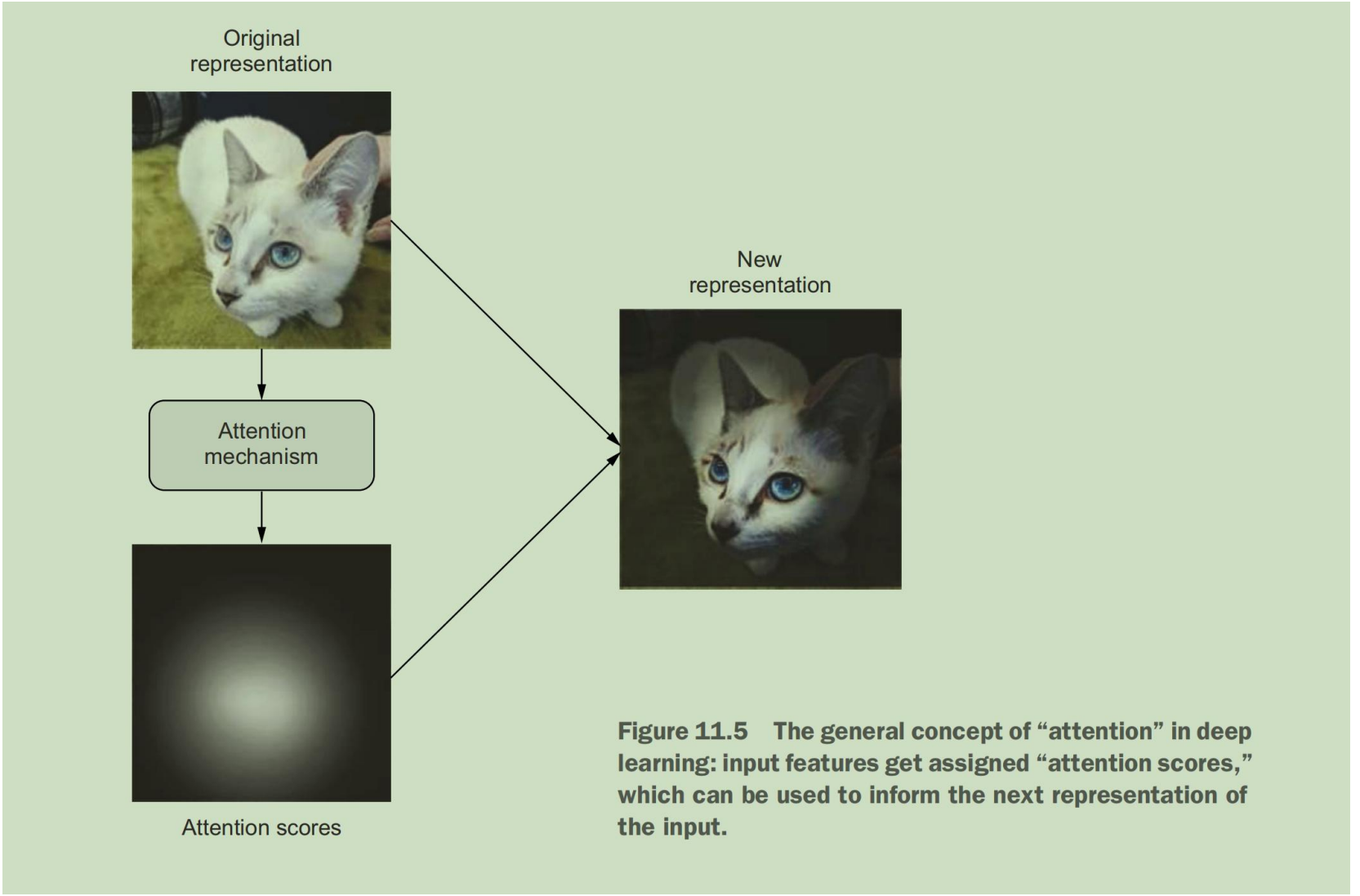
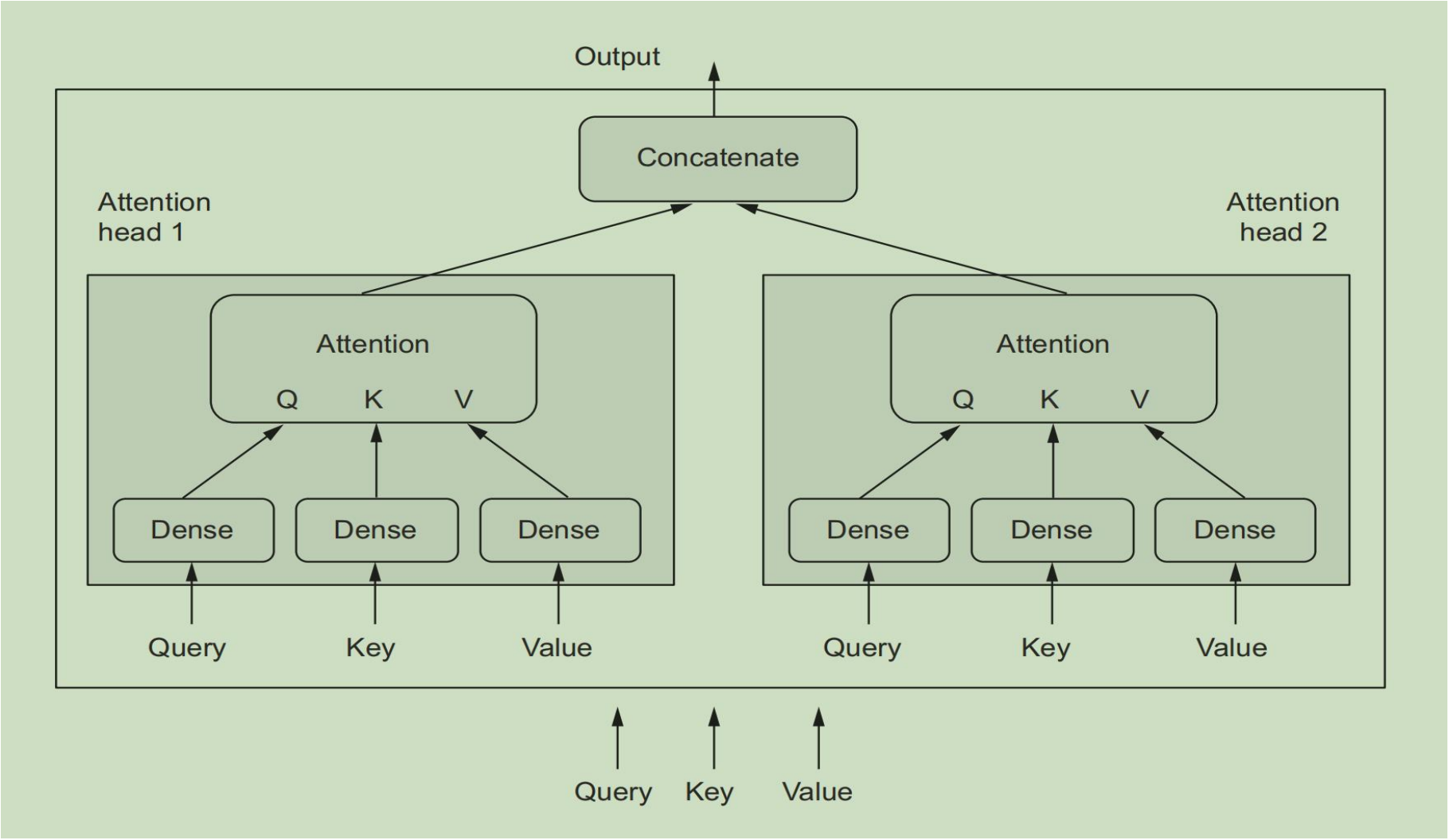
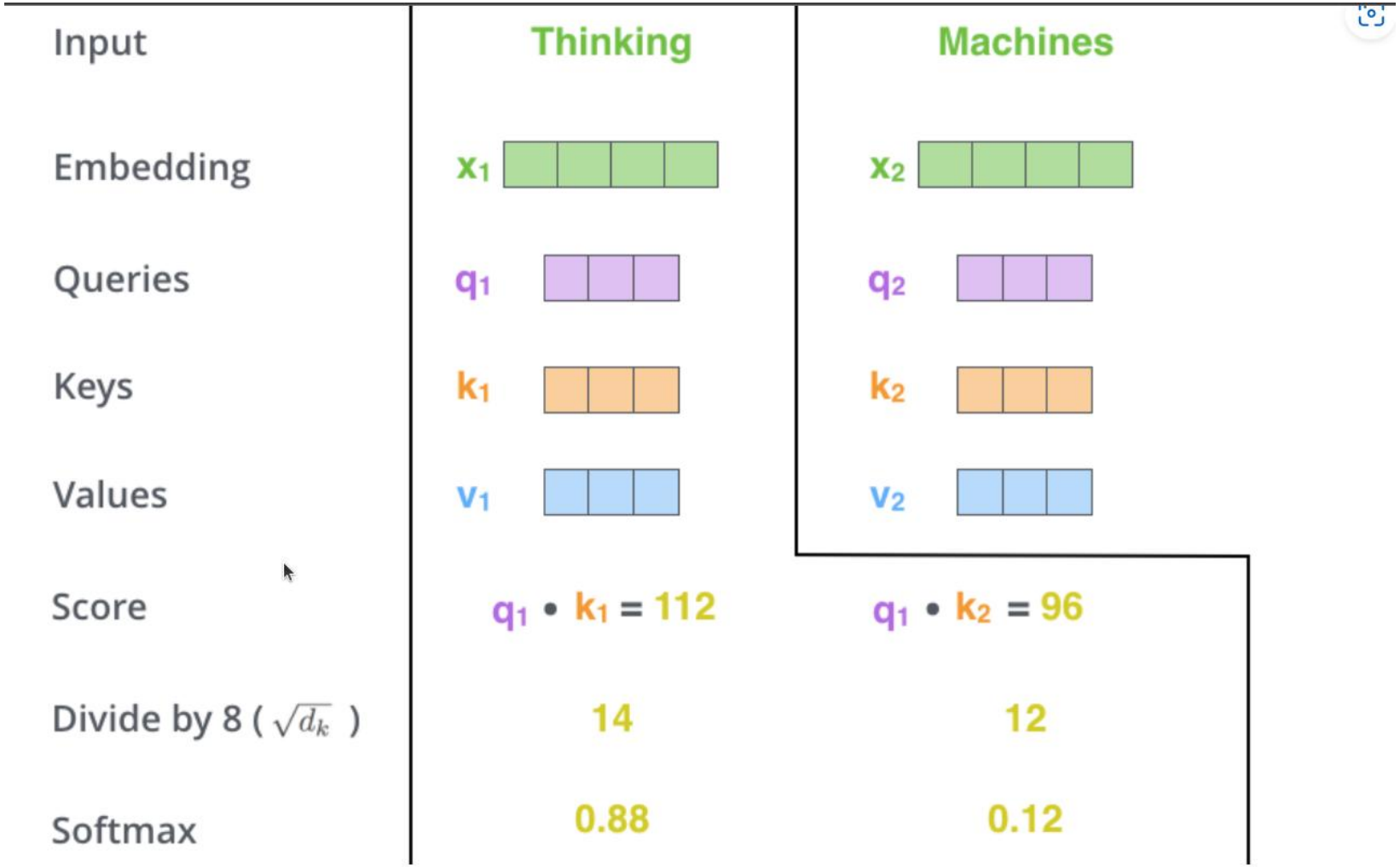


Figure 11.5 The general concept of "attention" in deep learning: input features get assigned "attention scores," which can be used to inform the next representation of the input.

Self Attention

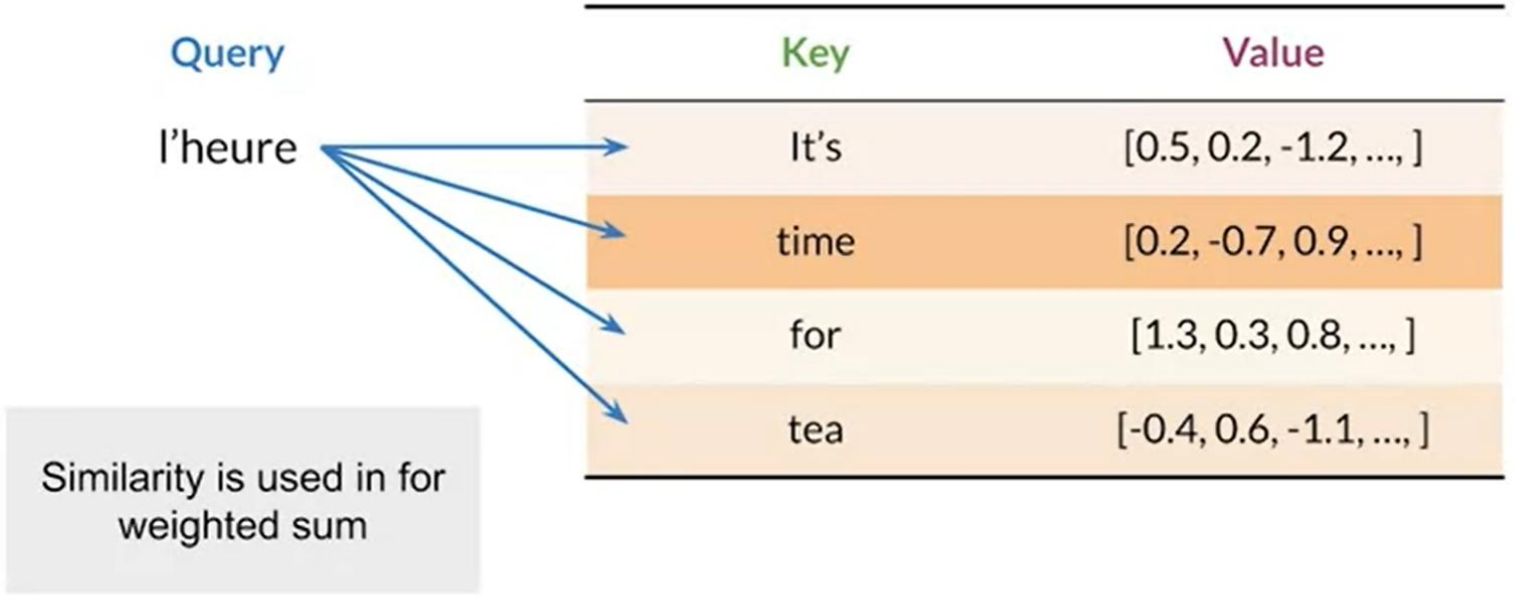


Process Flow

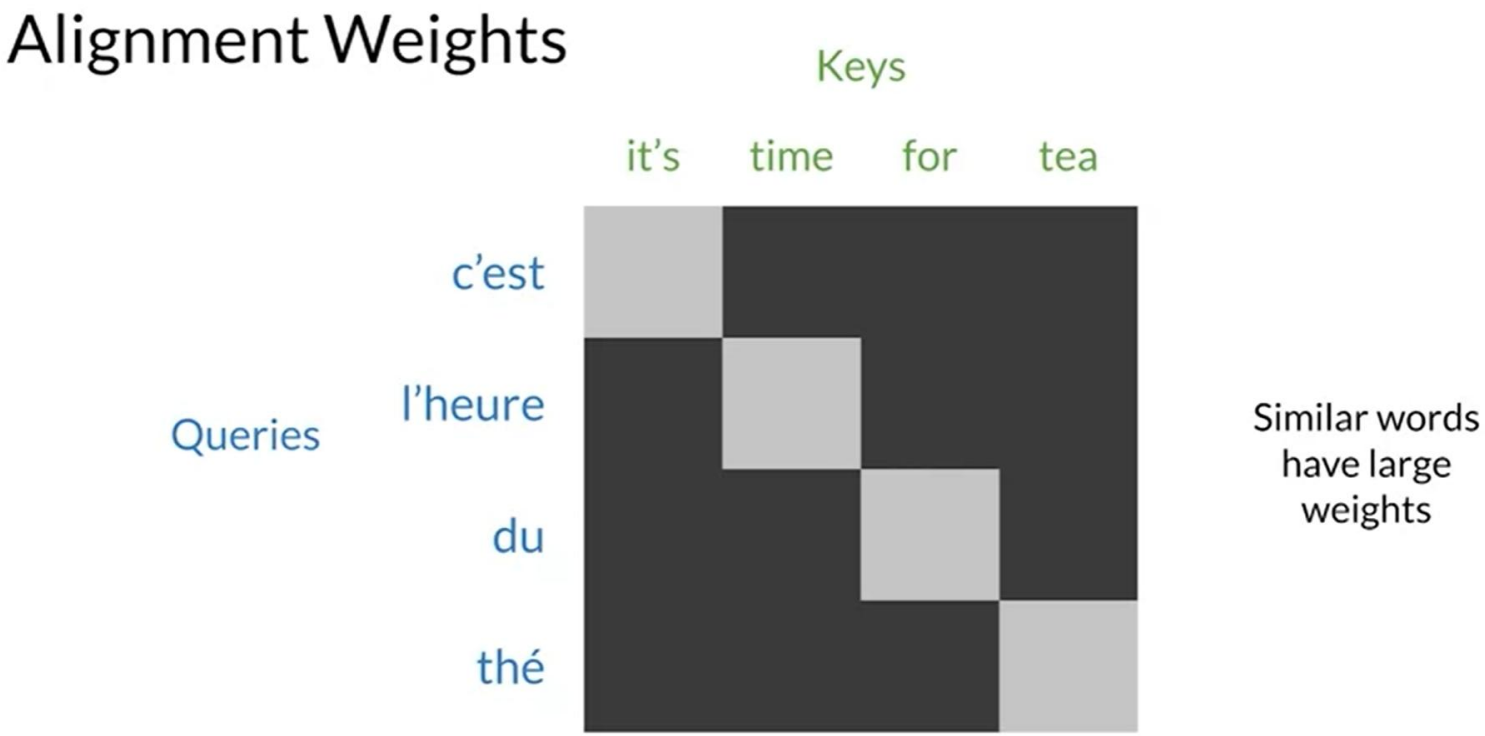


Attention - components

Queries, Keys, Values

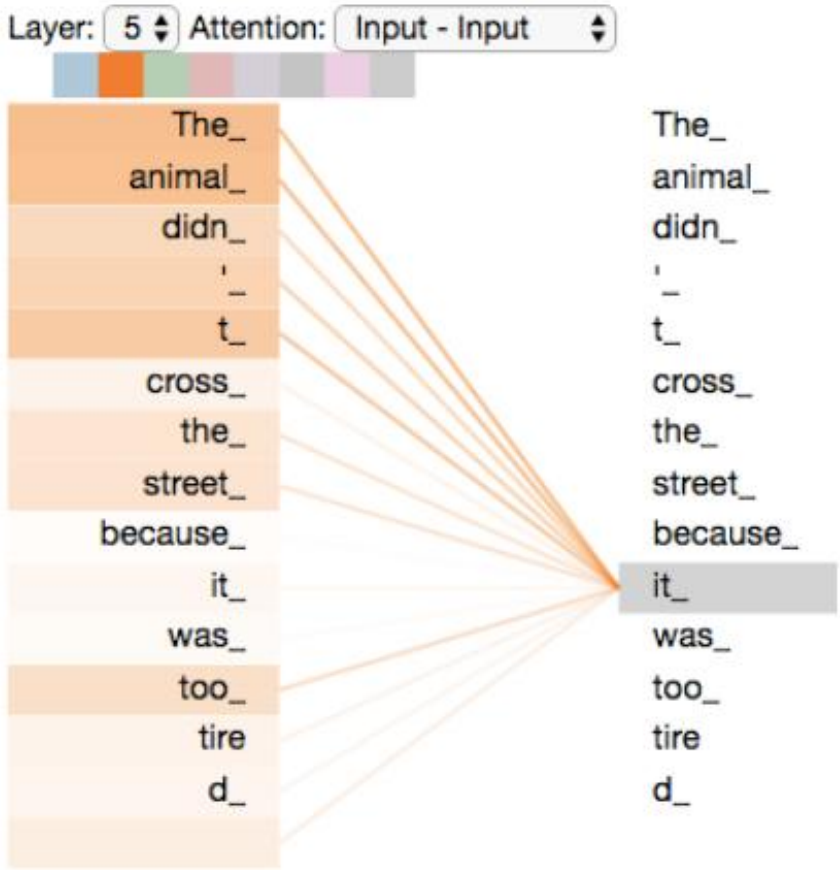


Similarity Score



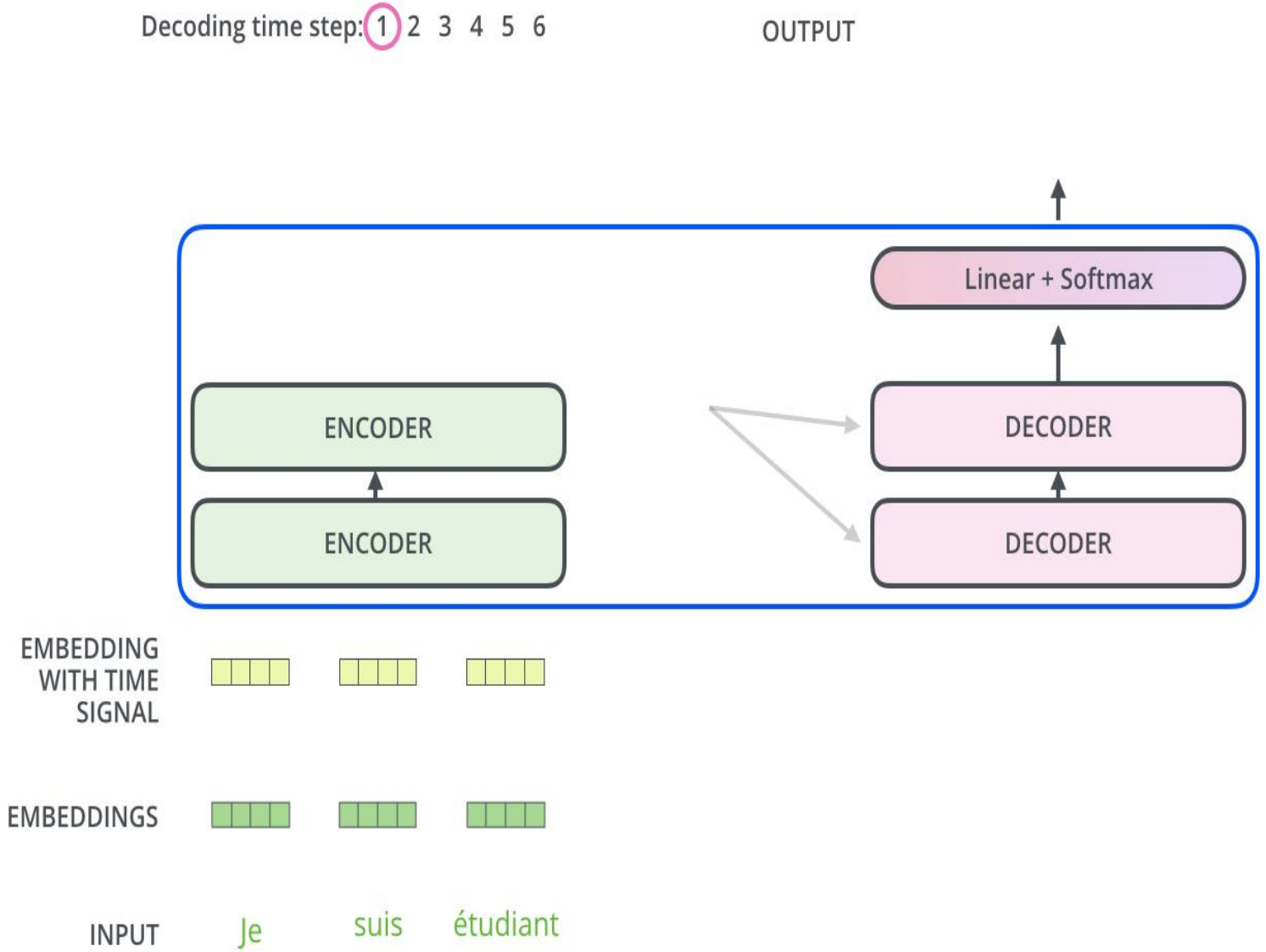
Example - Why need self attention

"The animal didn't cross the street because it was too tired"



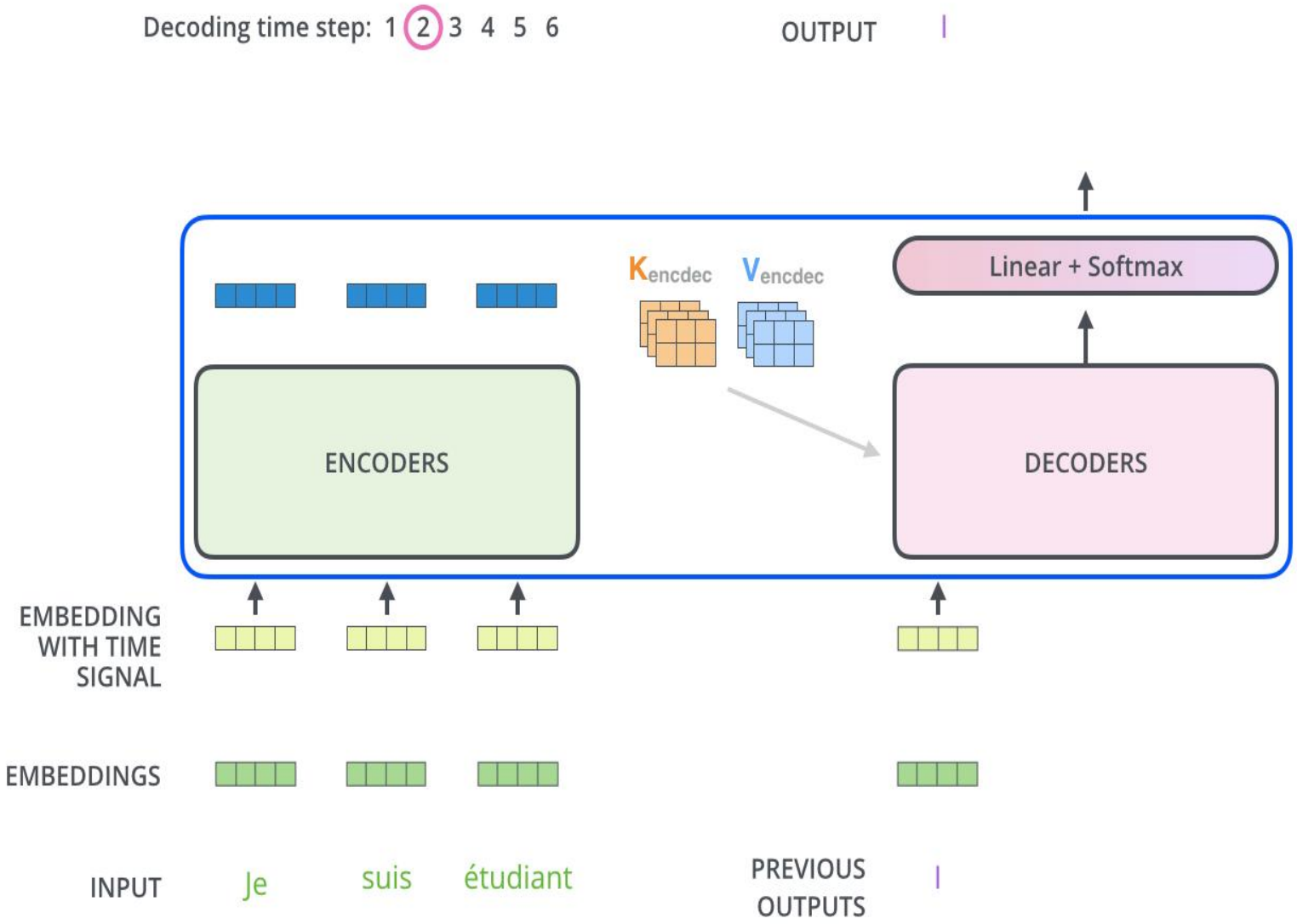
Visualization

http://jalammar.github.io/im



Visualization

http://jalammar.github.io/image



BERT | A Transformer with Attention and bidirection Training

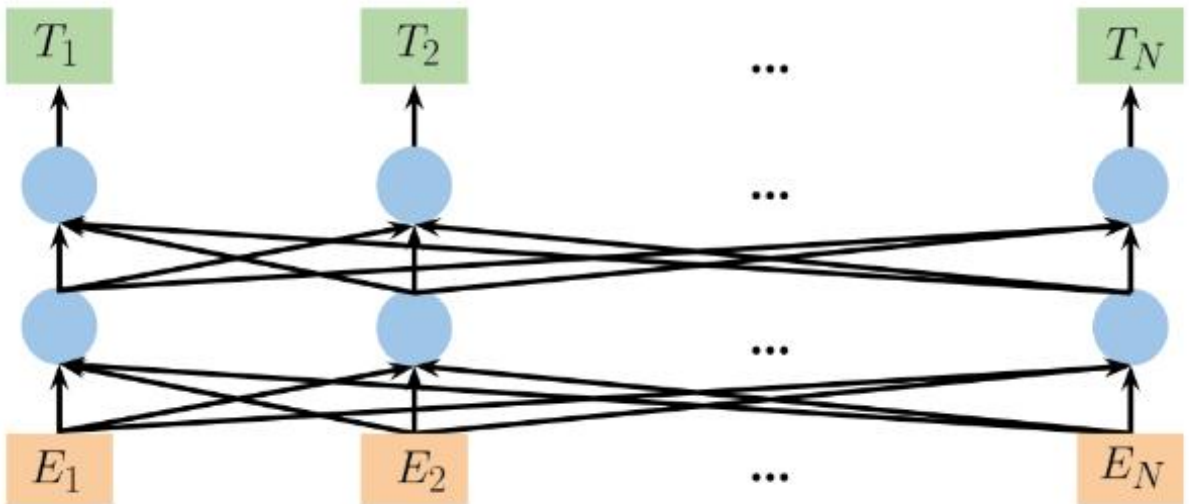
- A multi layer bidirectional transformer
- Positional embeddings
- BERT_base:
  - 12 layers (12 transformer blocks)
  - 12 attentions heads
  - 110 million parameters

BERT | Architecture

Bidirectional Encoder Representations from Transformers (BERT)

You will now learn about the BERT architecture and understand how the pre-training works.

- Makes use of transfer learning/pre-training:

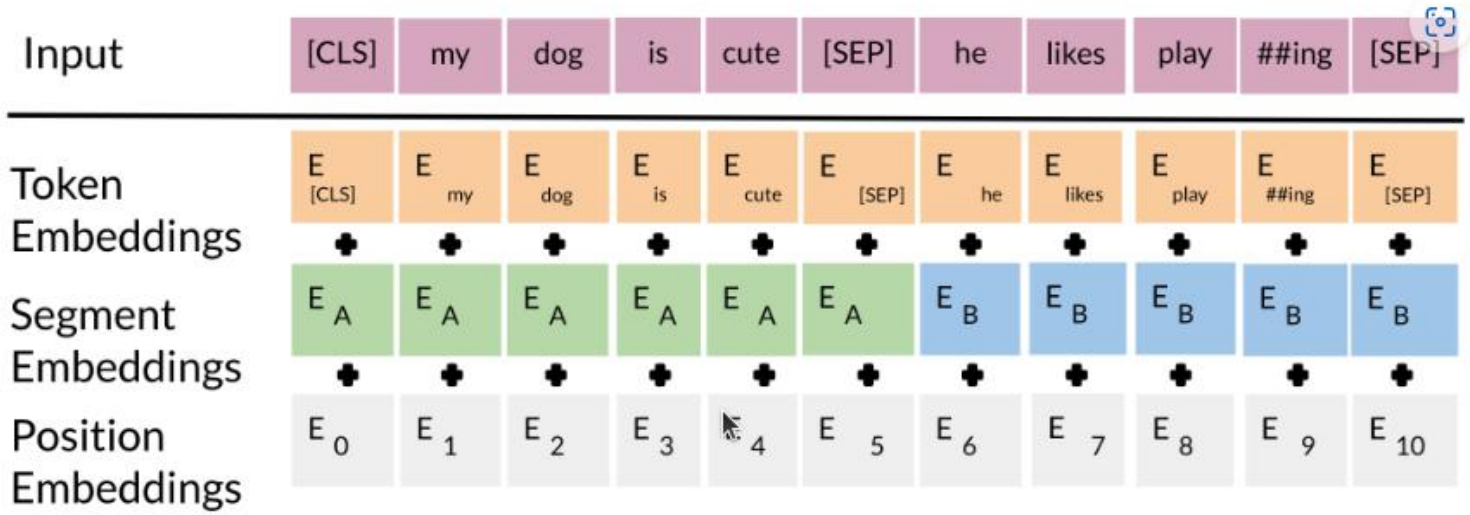


BERT | Training

- Choose 15% of the tokens at random: mask them 80% of the time, replace them with a random token 10% of the time, or keep as is 10% of the time.
- There could be multiple masked spans in a sentence
- Next sentence prediction is also used when pre-training.

BERT | Training

We will first start by visualizing the input.



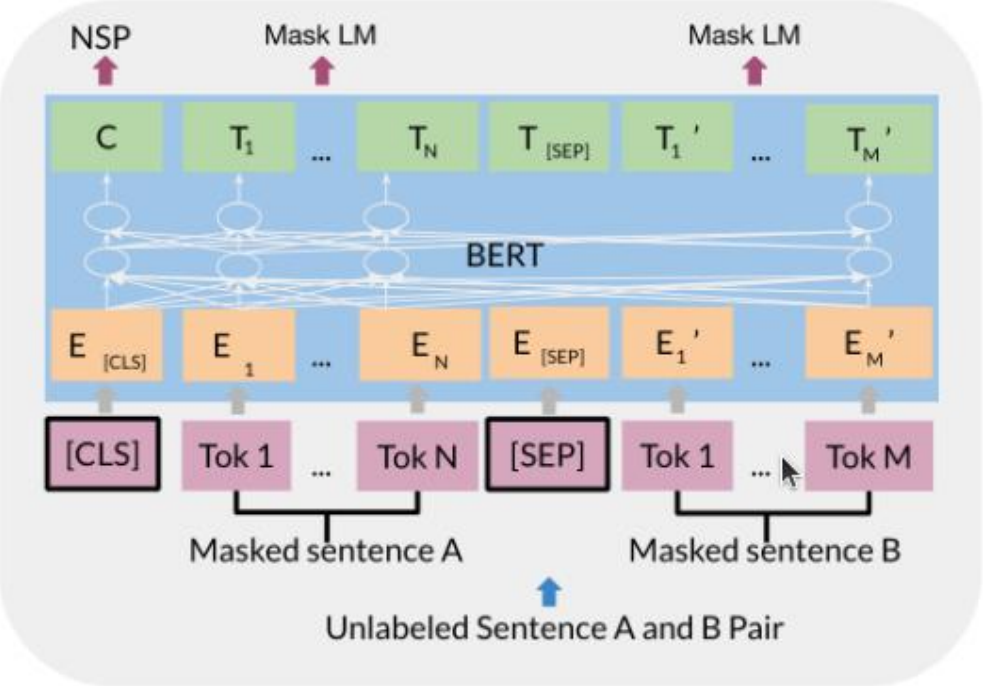
The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

The input embeddings: you have a CLS token to indicate the beginning of the sentence and a sep to indicate the end of the sentence

The segment embeddings: allows you to indicate whether it is sentence a or b.

Positional embeddings: allows you to indicate the word's position in the sentence.

BERT | Training



- [CLS]: a special classification symbol added in front of every input
- [SEP]: a special separator token

The C token in the image above could be used for classification purposes. The unlabeled sentence A/B pair will depend on what you are trying to predict, it could range from question answering to sentiment. (in which case the second sentence could be just empty). The BERT objective is defined as follows:

BERT | Use Cases

Text encoding

Text Summarization

Response Selection

Question Answering

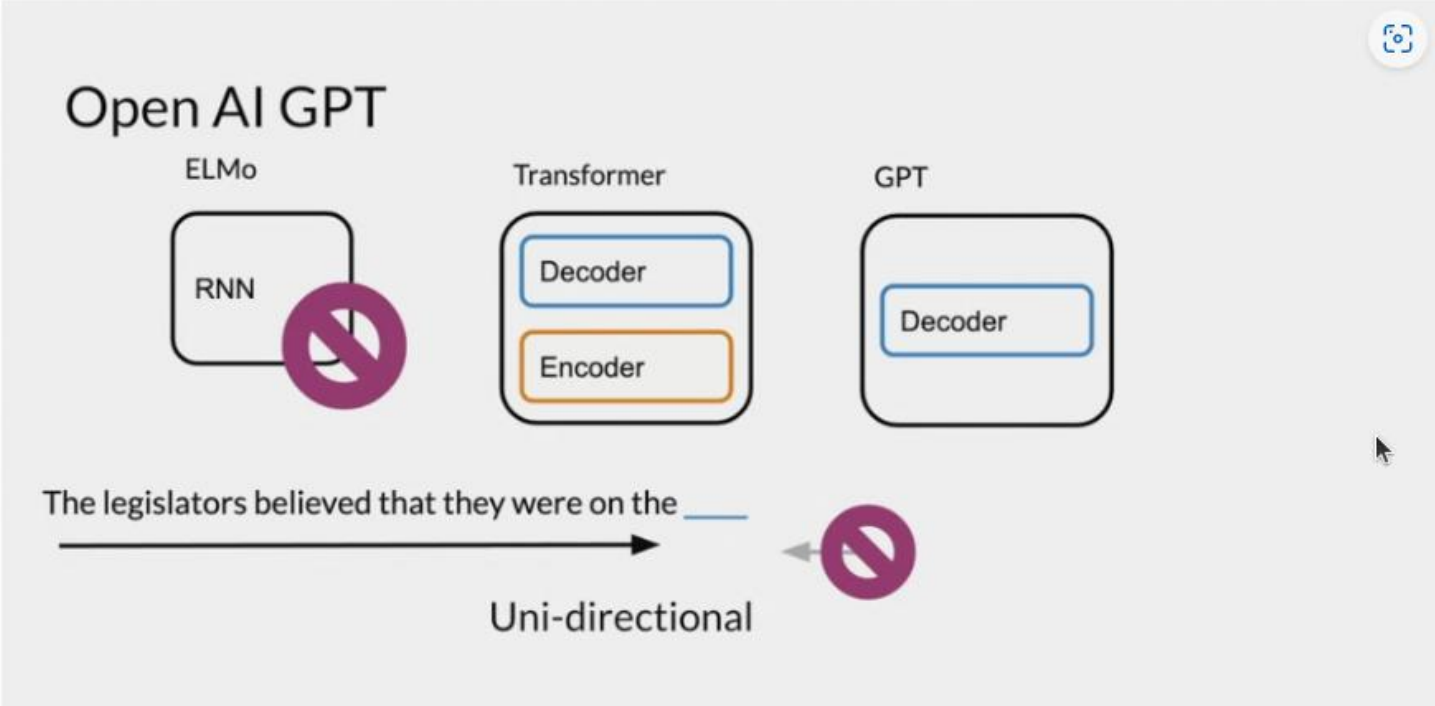
Similarity retrieval

And more...

Language Models | General

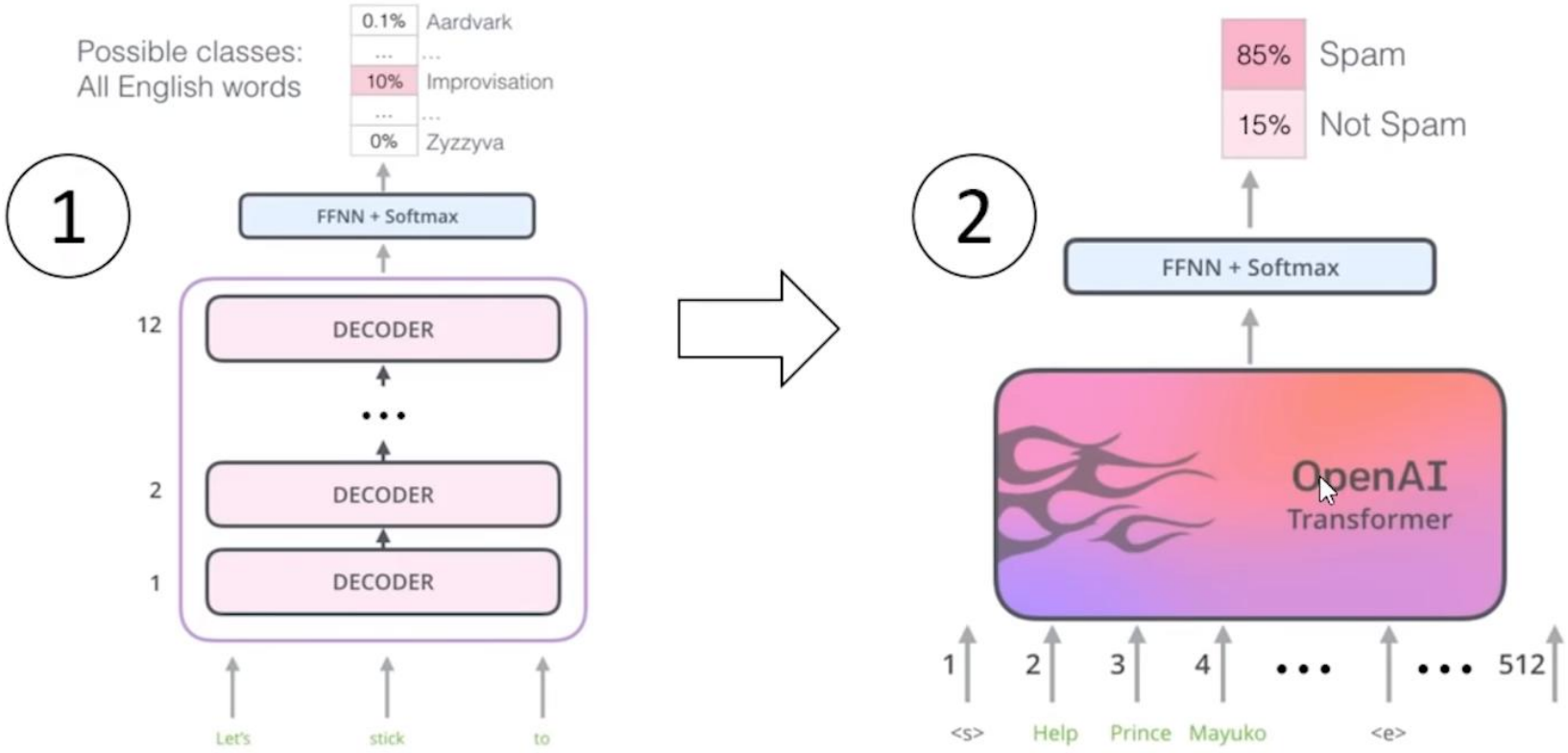
ELMo, GPT, BERT, T5

Around 2:50, Younes *incorrectly* mentions that ELMo is uni-directional. Please note, ELMo is **bi-directional**.



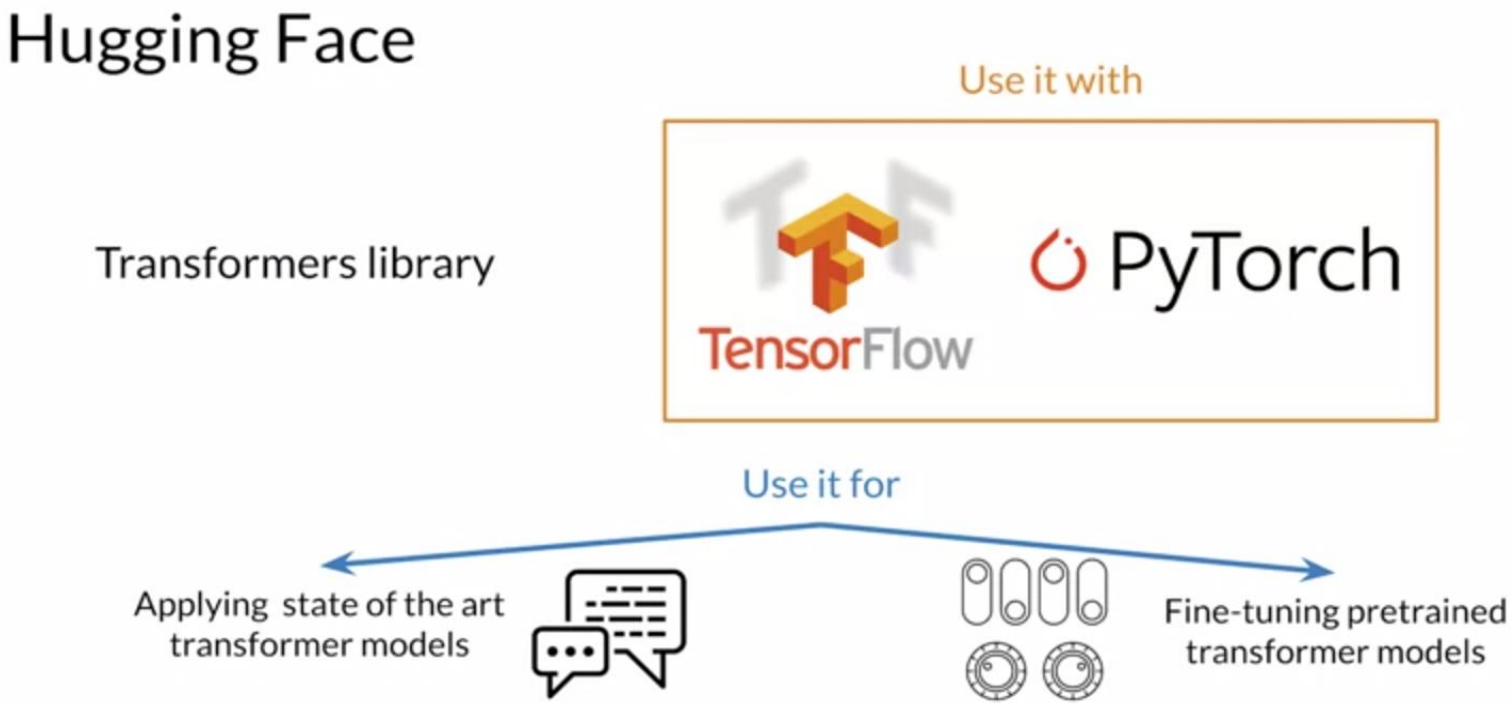
GPT

OpenAI Transformer



1. Pre-train a Transformer's decoder for language modeling
2. Train it on, for example, a sentence classification task

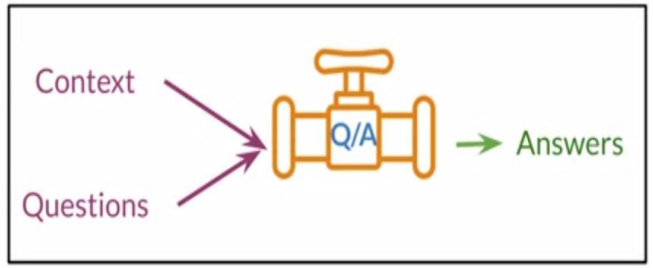
Hugging Face | A BERT & Transformers- Echo System

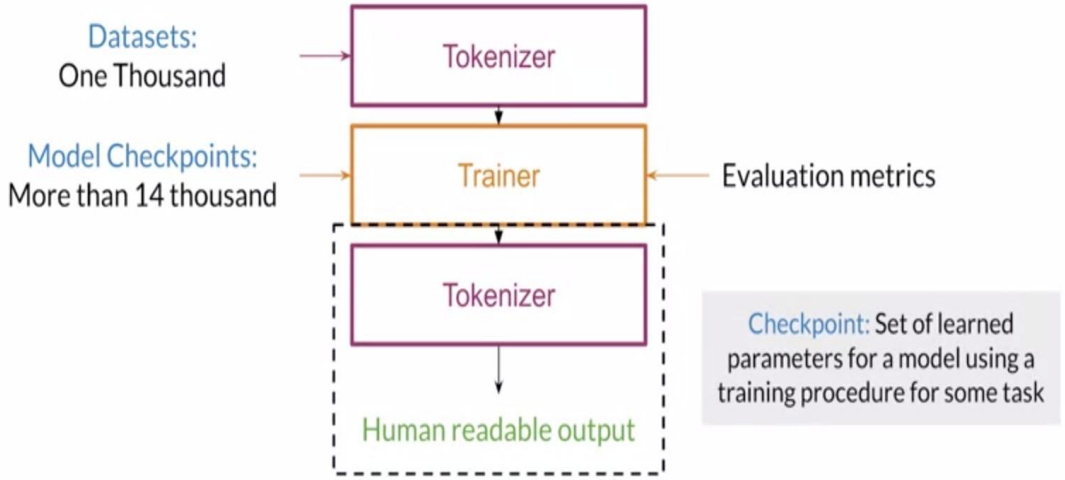


Hugging Face | Using Transformers

Pipelines

1. **Pre-processing** your inputs
2. **Running** the model
3. **Post-processing** the outputs





Model Checkpoints

Model Checkpoints:
More than 15 thousand
(and increasing)

Upload the architecture and weights with 1 line of code!

Model	Dataset	Name in 🤗
DistilBERT	Stanford Question Answering Dataset (SQuAD)	distilbert-base-cased-distilled-squad
BERT	Wikipedia and Book Corpus	bert-base-cased

Hugging Face | Code Example

- C4_W3_1_Question_Answering_with_BERT_and_HuggingFace_Pytorch_tydiqa.ipynb - Colaboratory

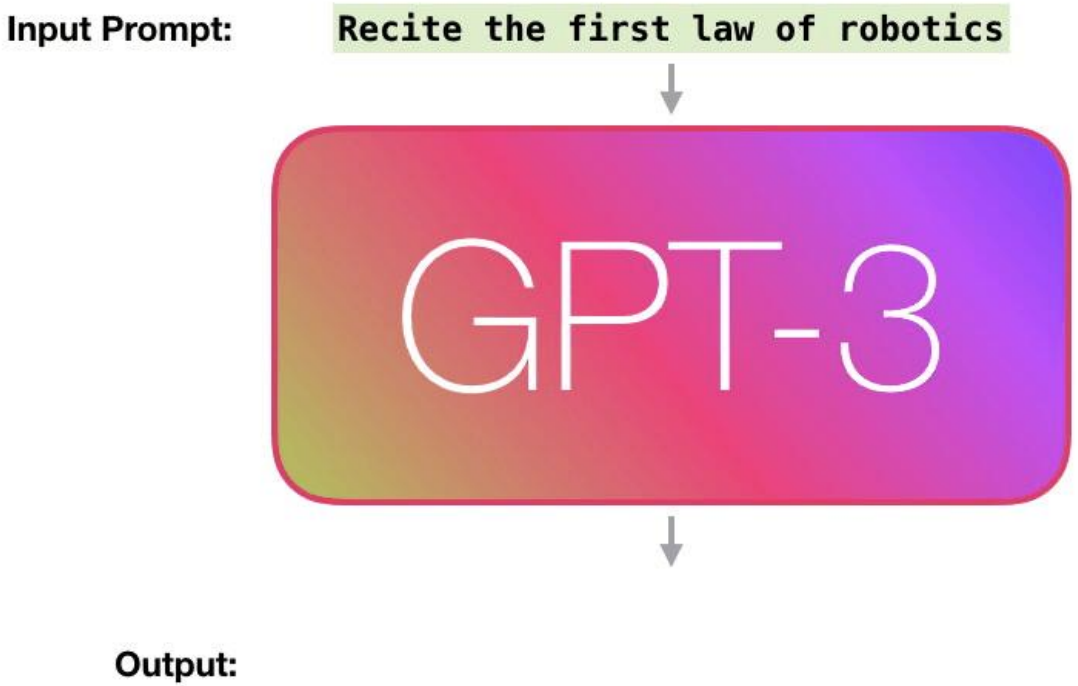
https://colab.research.google.com/drive/1O4LvdhHw6Zx7Kd43HK-p5a1rtsHUEia5#scrollTo=oDG5fgap-N7I

GPT3 , and ChatGPT

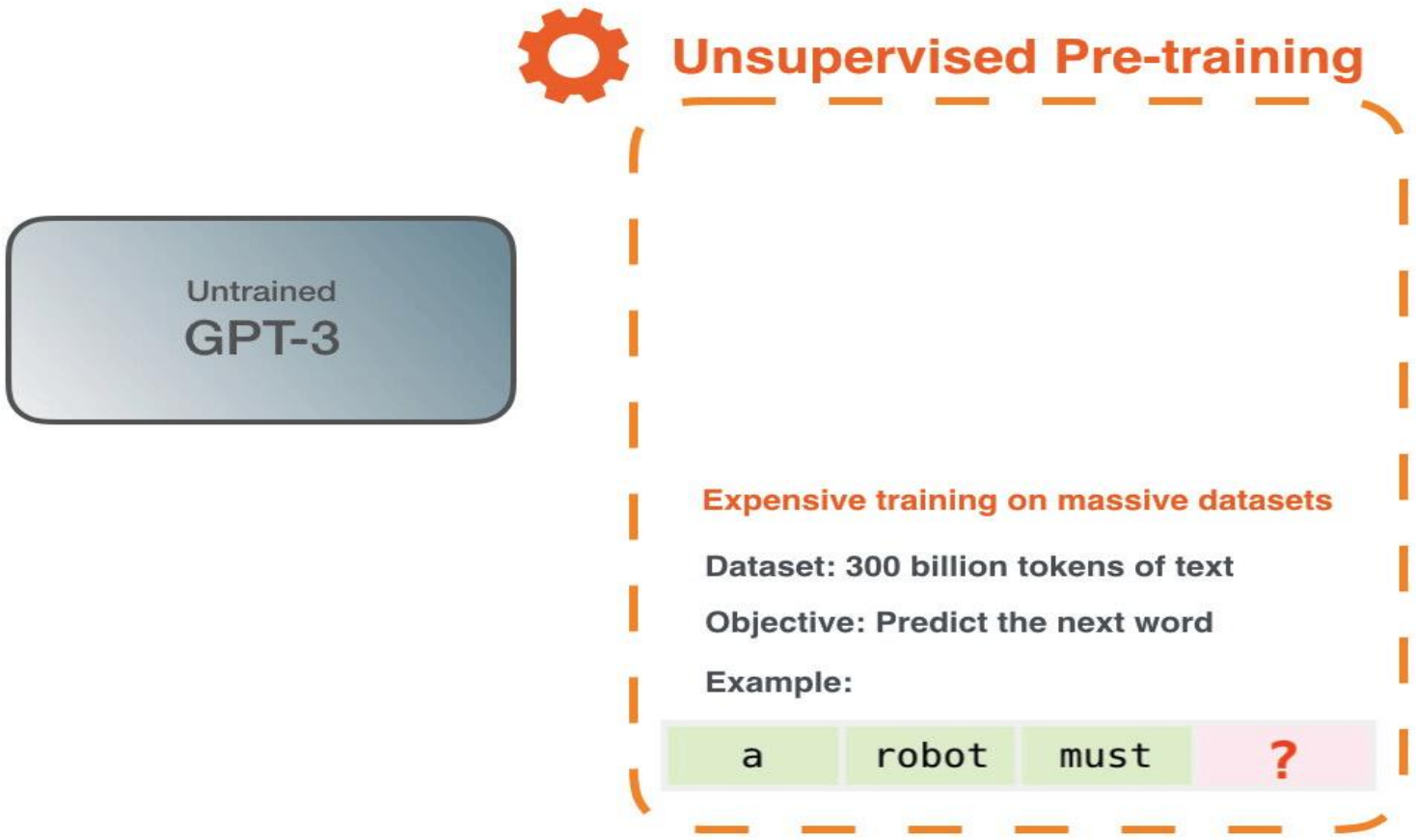
- GPT3 - is a Language Model which may be used in variety of Languange agnostic applications

- ChatGPT - is a specifically designed for conversational Langauge Applications.

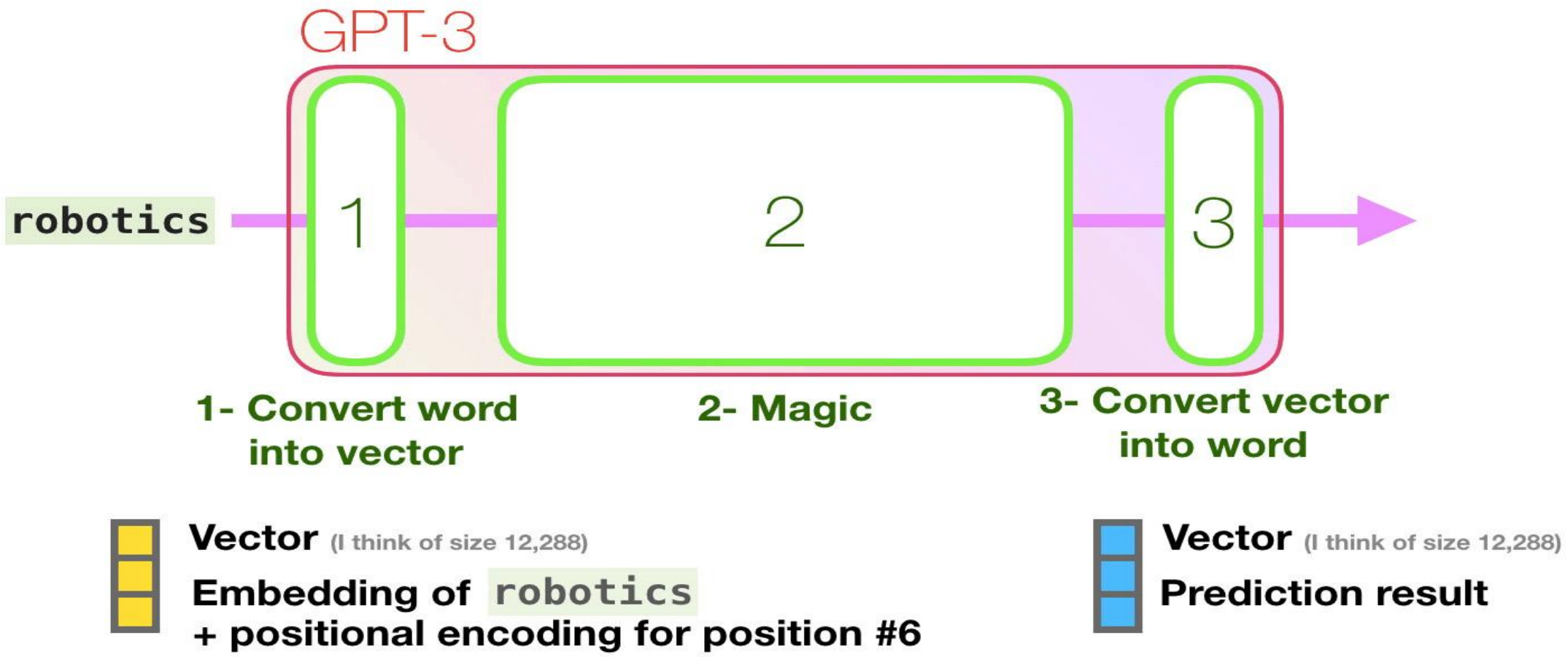
GPT-3



GPT3


Unsupervised Pre-training
Untrained
GPT-3
Expensive training on massive datasets
Dataset: 300 billion tokens of text
Objective: Predict the next word
Example:
a
robot
must
?

GPT3


GPT-3
robotics
1
2
3
1- Convert word into vector
2- Magic
3- Convert vector into word
Vector (I think of size 12,288)
Embedding of robotics
+ positional encoding for position #6
Vector (I think of size 12,288)
Prediction result

GPT - Fine Tuning