NLP-101-Introduction

Advanced NLP

Ali Abdelaal, Language Engineer @Apple

Session Agenda

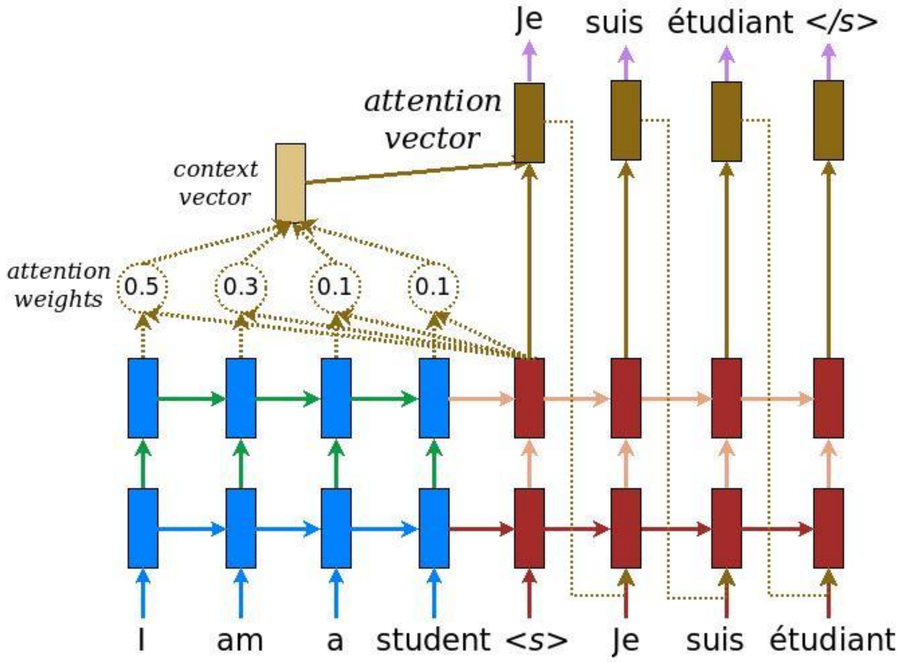
- NMT evaluation
- seq2seq tasks (summarization, question answering)
- Transformers
  - GPT, BERT, T5
- Transfer learning
- Knowledge distillation

NMT Recap

NMT Recap

NMT

- NMT is a form of a seq2seq problem.
- An encoder grasps info from input
- A decoder generates the translation
- An attention layer helps the decoder focus on important pieces from the input
- Each decoder output is reused to generate the following output.



NMT Evaluation

BLEU (BiLingual Evaluation Understudy)

NMT Evaluation

- Compare predicted translation (candidate) to the reference one (from a human)
- Ranges from 0 to 1, 1 is the best.
- It tries to find how many words from candidate appear in the reference.
- It doesn't consider semantic meaning or sentence structure.
- You can think of it as a precision metric for the translation.

BLEU (BiLingual Evaluation Understudy)

NMT Evaluation

candidate	I	I	am	I	
reference 1	Younes	said	I	am	hungry
reference 2	He	said	I	am	hungry
	Count:	**1**			

source

BLEU (BiLingual Evaluation Understudy)

NMT Evaluation

candidate	I	I	am	I	
reference 1	Younes	said	I	am	hungry
reference 2	He	said	I	am	hungry

Count: **2**

source

BLEU (BiLingual Evaluation Understudy)

NMT Evaluation

candidate	I	I	am	I	
reference 1	Younes	said	I	am	hungry
reference 2	He	said	I	am	hungry

Count: **3**

source

BLEU (BiLingual Evaluation Understudy)

NMT Evaluation

candidate	I	I	am	I	
reference 1	Younes	said	I	am	hungry
reference 2	He	said	I	am	hungry

Count: **4**

source

BLEU (BiLingual Evaluation Understudy)

NMT Evaluation

candidate	I	I	am	I	
reference 1	Younes	said	I	am	hungry
reference 2	He	said	I	am	hungry

Count: **4 / len(text) = 4/4 = 1**

source

BLEU modified

NMT Evaluation

candidate	I	I	am	I	
reference 1	Younes	said	I	am	hungry
reference 2	He	said	I	am	hungry

Count: **1**

source

BLEU modified

NMT Evaluation

candidate	I	I	am	I	
reference 1	Younes	said		am	hungry
reference 2	He	said		am	hungry

Count: **1**

source

BLEU modified

NMT Evaluation

candidate	I	I	am	I	
reference 1	Younes	said		am	hungry
reference 2	He	said		am	hungry

Count: **2**

source

BLEU modified

NMT Evaluation

candidate	I	I	am	I	
reference 1	Younes	said			hungry
reference 2	He	said			hungry
	Count:	**2**			

source

BLEU modified

NMT Evaluation

candidate	I	I	am	I	
reference 1	Younes	said			hungry
reference 2	He	said			hungry

Count: **2 / len(text) = 2/4 = 0.5**

source

ROUGE-N Score

NMT Evaluation

- Recall-Oriented Understudy for Gisting Evaluation.
- Compare the reference to predicted translation (candidate) .
- Ranges from 0 to 1, 1 is the best.
- It tries to find how many words from reference appear in the candidate.
- It doesn't consider semantic meaning or sentence structure.
- You can think of it as a recall metric for the translation.

ROUGE-N Score

NMT Evaluation

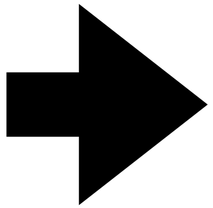
candidate	I	I	am	I	
reference 1	Younes	said	I	am	hungry
reference 2	He	said	I	am	hungry
	Count 1:	**2/5**		**Count 2:**	**2/5**

source

F1 score
NMT Evaluation

candidate	I	I	am	I	
reference 1	Younes	said	I	am	hungry
reference 2	He	said	I	am	hungry

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$



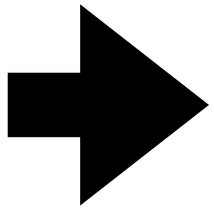
$$F1 = 2 \times \frac{BLEU \times ROUGE\text{-}N}{BLEU + ROUGE\text{-}N}$$

source

F1 score
NMT Evaluation

candidate	I	I	am	I	
reference 1	Younes	said	I	am	hungry
reference 2	He	said	I	am	hungry

$$F1 = 2 \times \frac{BLEU \times ROUGE-N}{BLEU + ROUGE-N}$$



$$F1 = 2 \times \frac{0.5 \times 0.4}{0.5 + 0.4} = \frac{4}{9} \approx 0.44$$

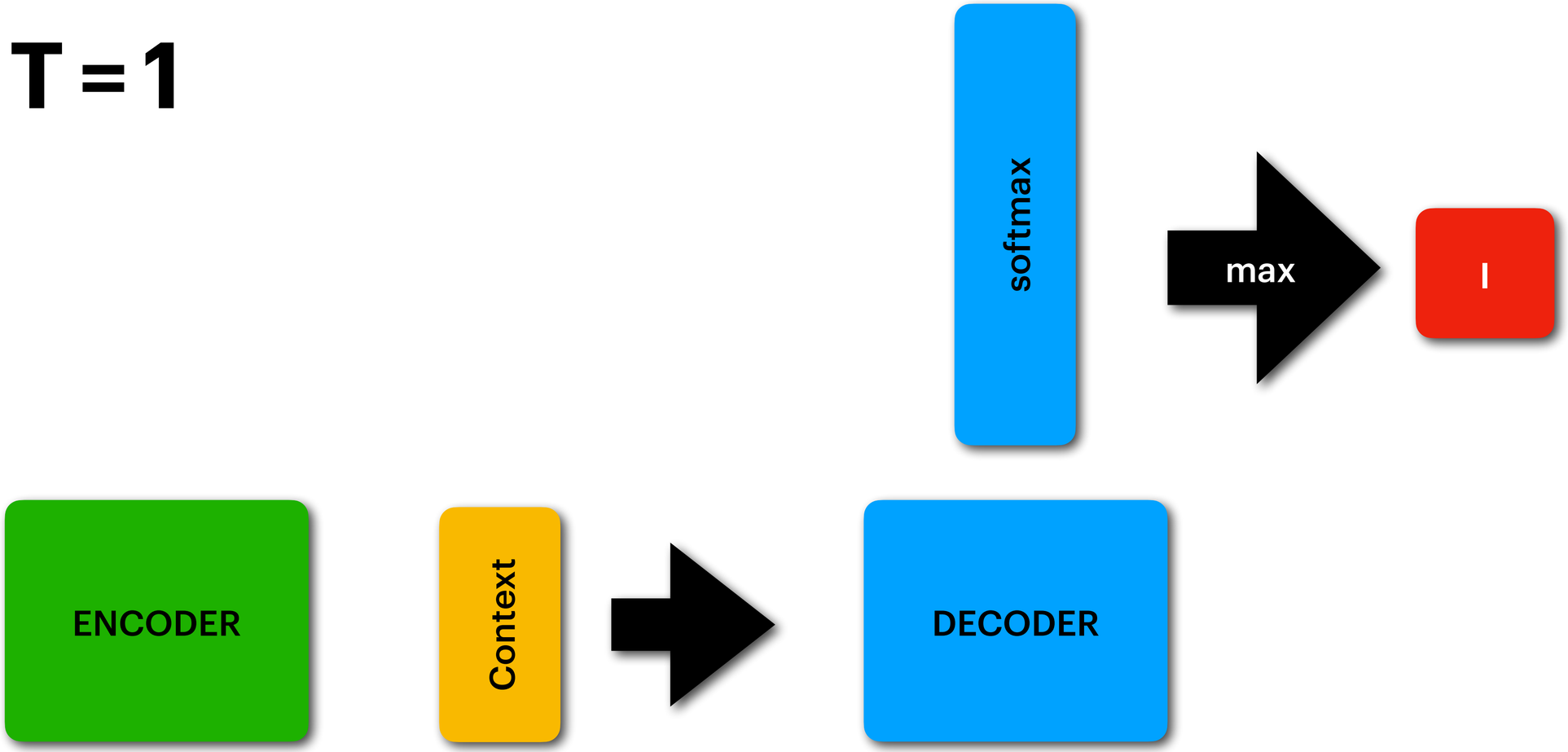
source

Back to decoder

Decoding output

NMT

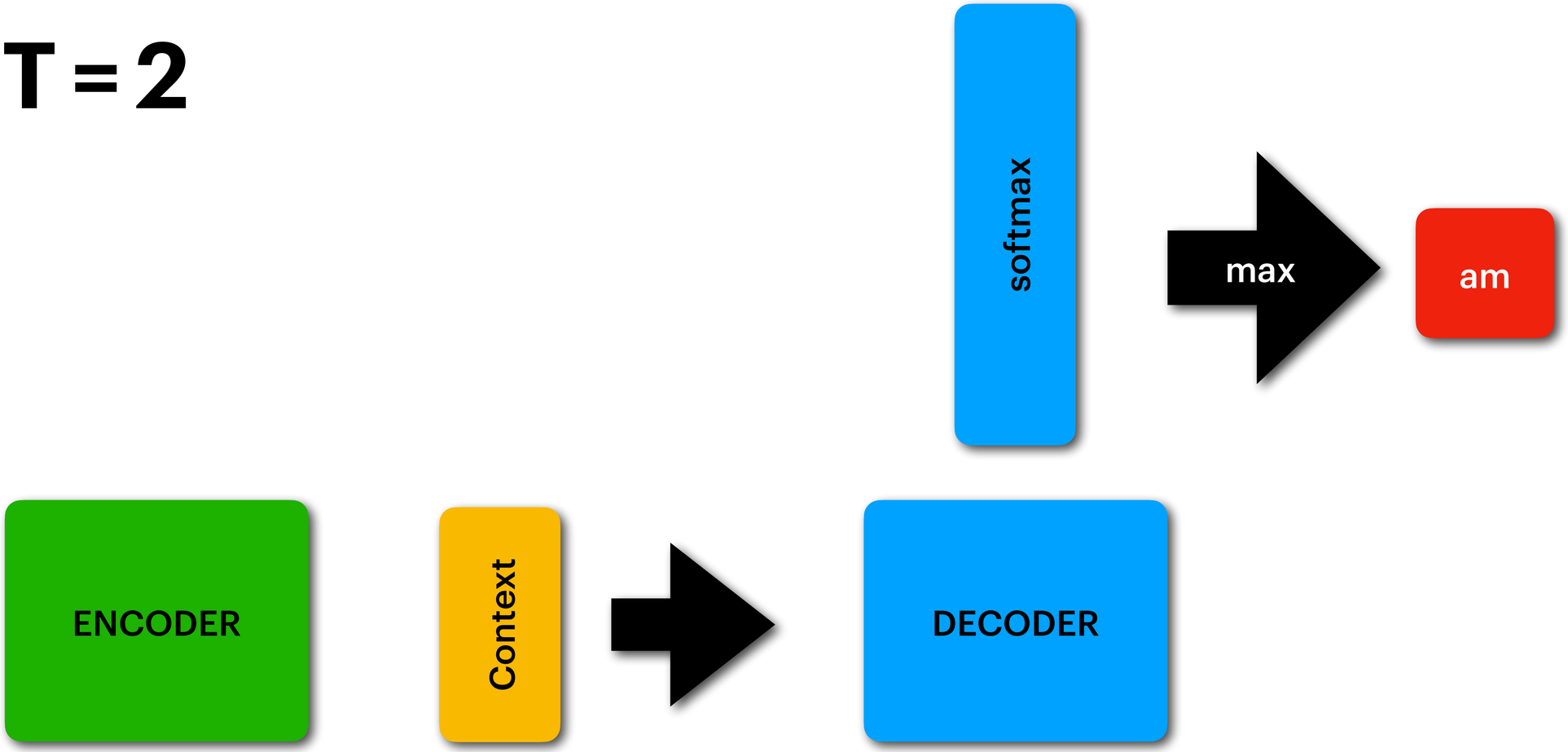
T = 1



Decoding output

NMT

T = 2

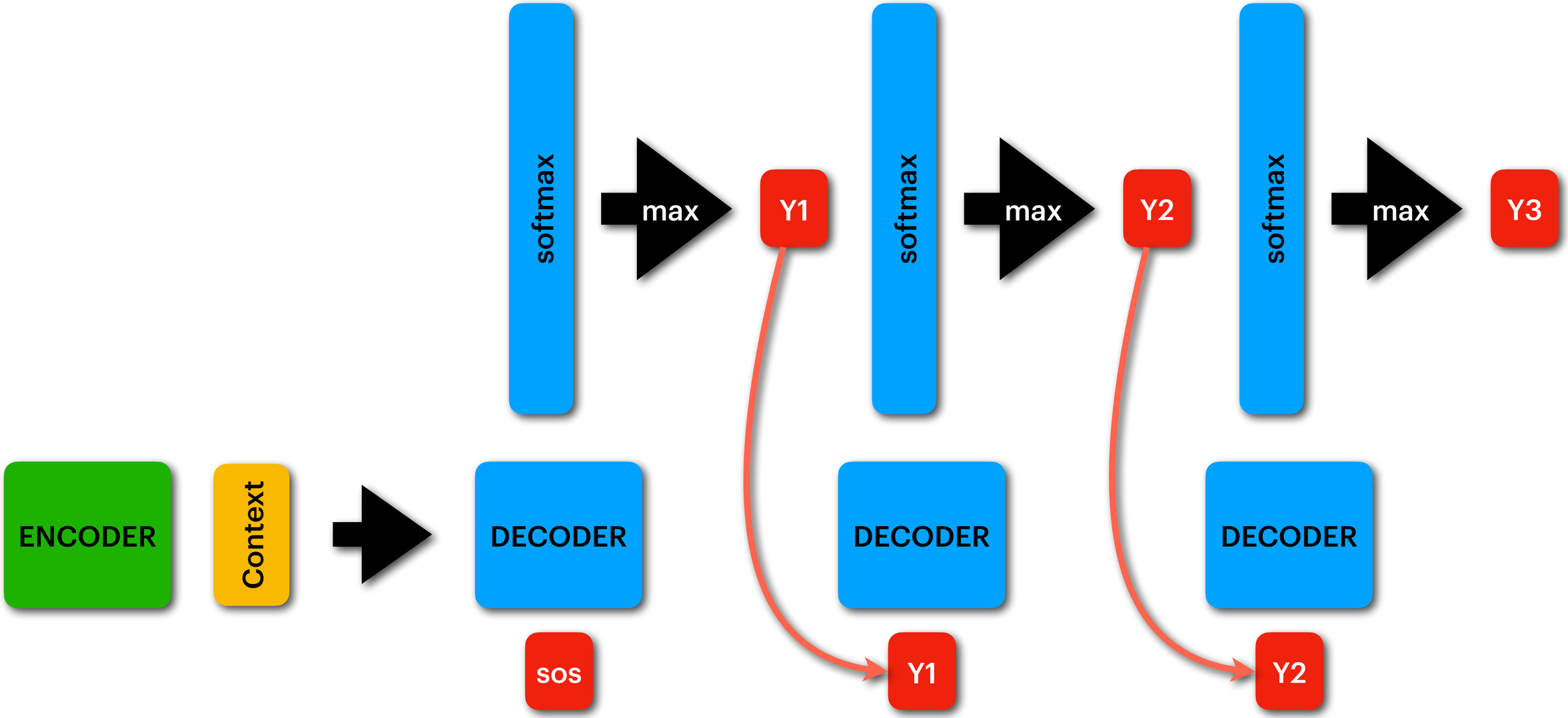


je suis étudiant

I

Greedy decoding

NMT



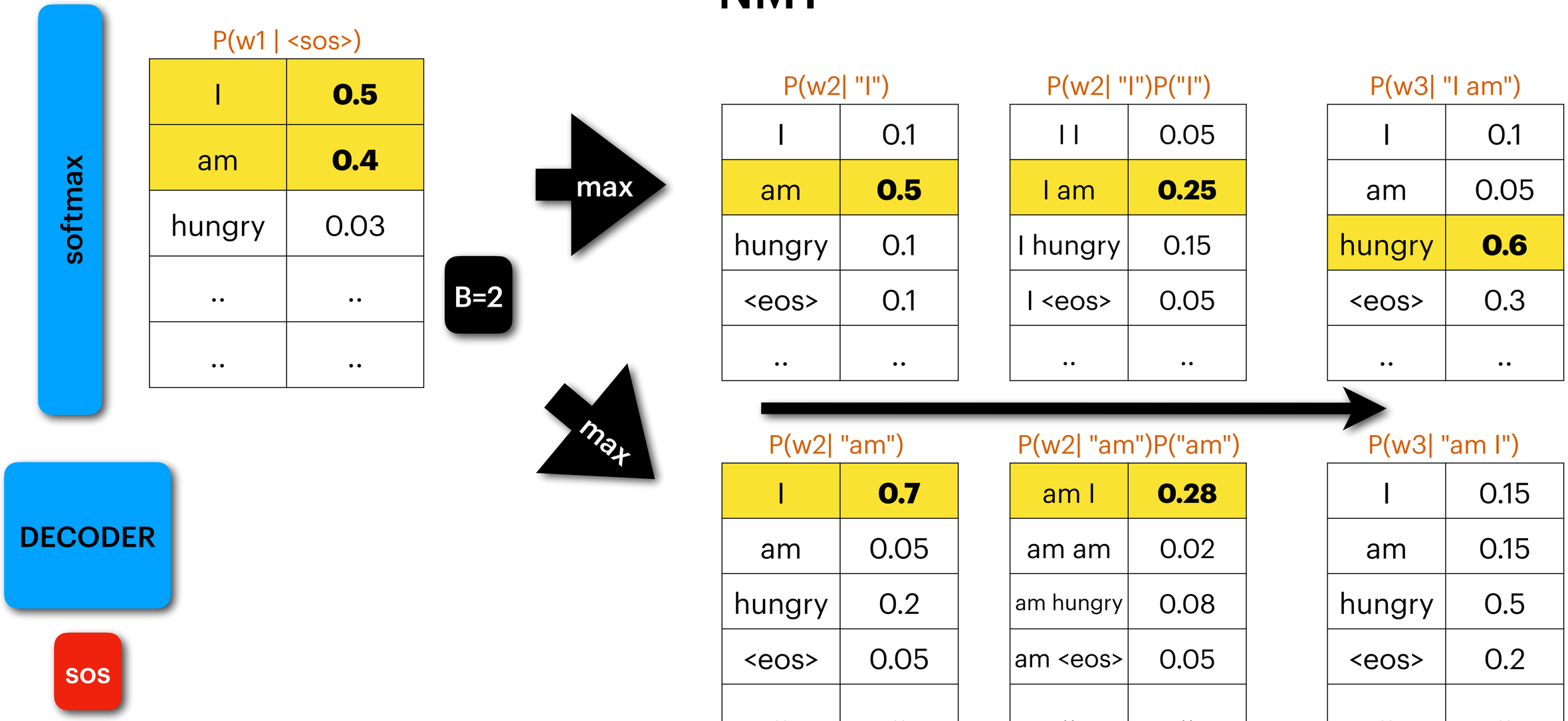
Beam Search

NMT

- Greedy decoding select the best word at each step, but this doesn't mean it is the best word overall.
- Beam search tries different paths and choose the best overall.
- Beam search with enough window helps pick the best word according to past words not just word.

Beam Search

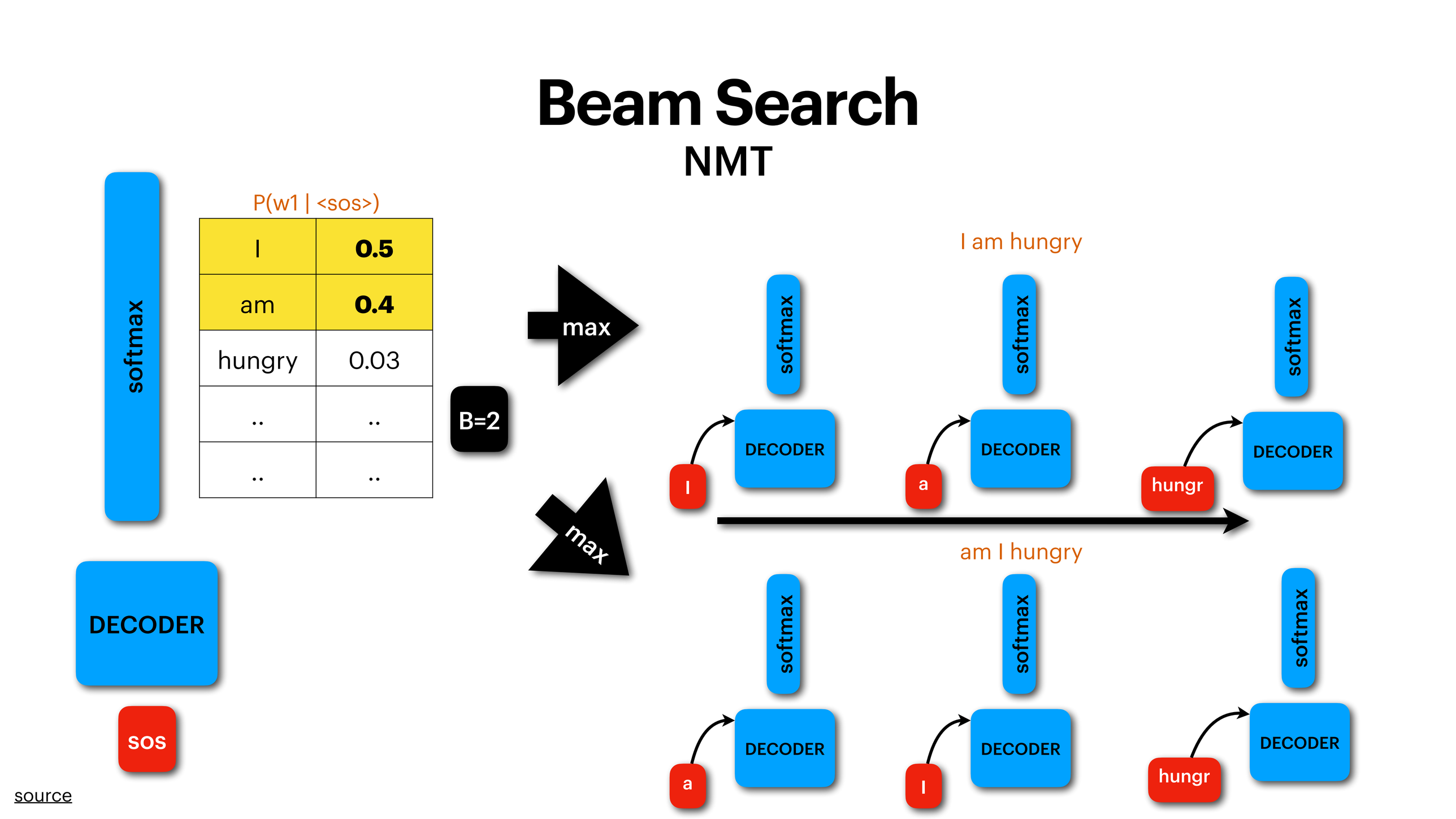
NMT



source

Beam Search

NMT



source

Conclusion

NMT

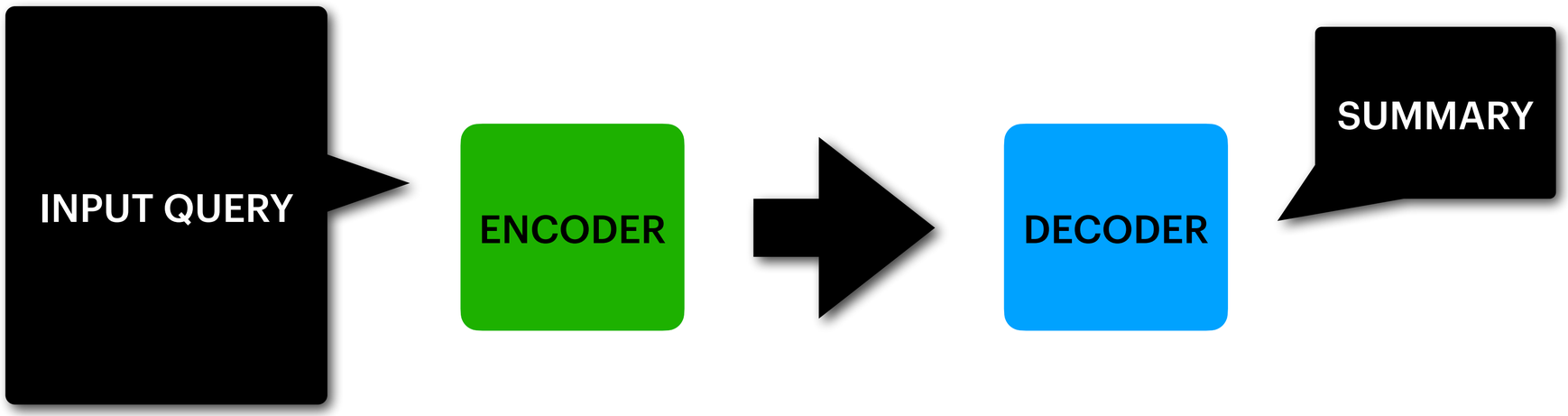
- NMT is a seq2seq problem.
- We use BLEU score, ROUGE-N and more metrics to assess the translation.
- Beam search is used to find the best overall translation with the higher overall probability.

More on seq2seq

Text Summarization

seq2seq

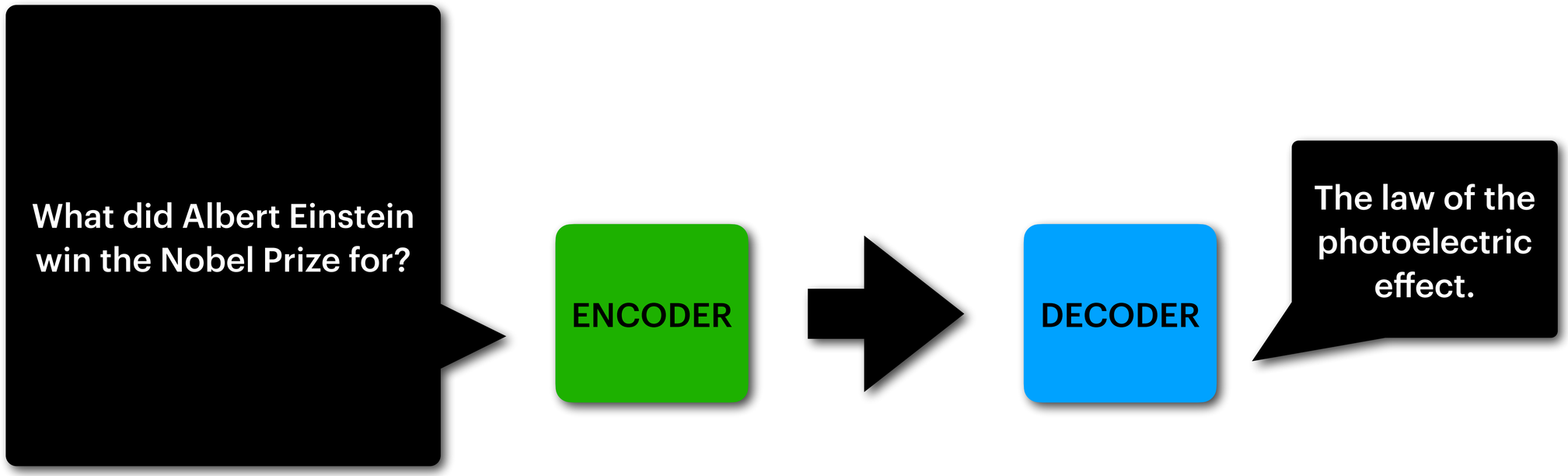
- In summarization task the output sequence is the summary for the input context.
- We can use BLEU and ROUGE to assess the summarization.
- Attention is very useful here too.



Question Answering

seq2seq

- Open-Domain QA is a form of tasks where the model is only given a question and is required to answer it with no context

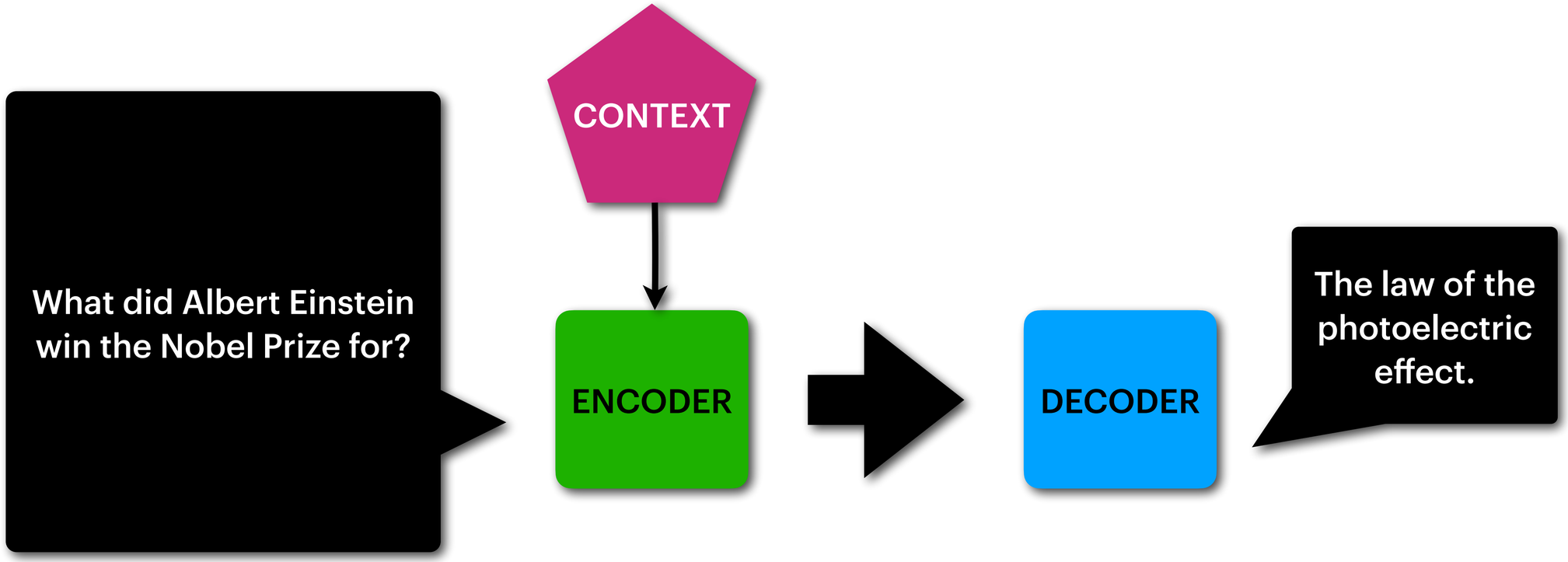


source

Question Answering

seq2seq

- Reading comprehension QA is a task where a context is provided to the model.



source

RNN Limitations

RNN Limitations

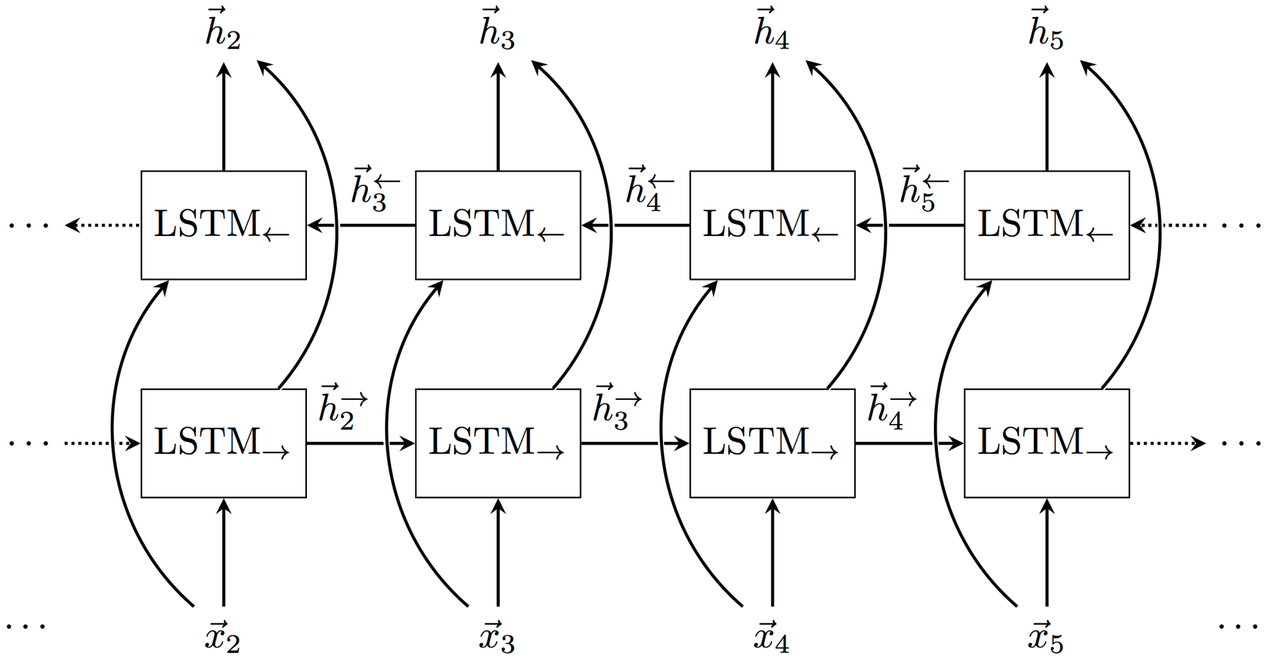
seq2seq

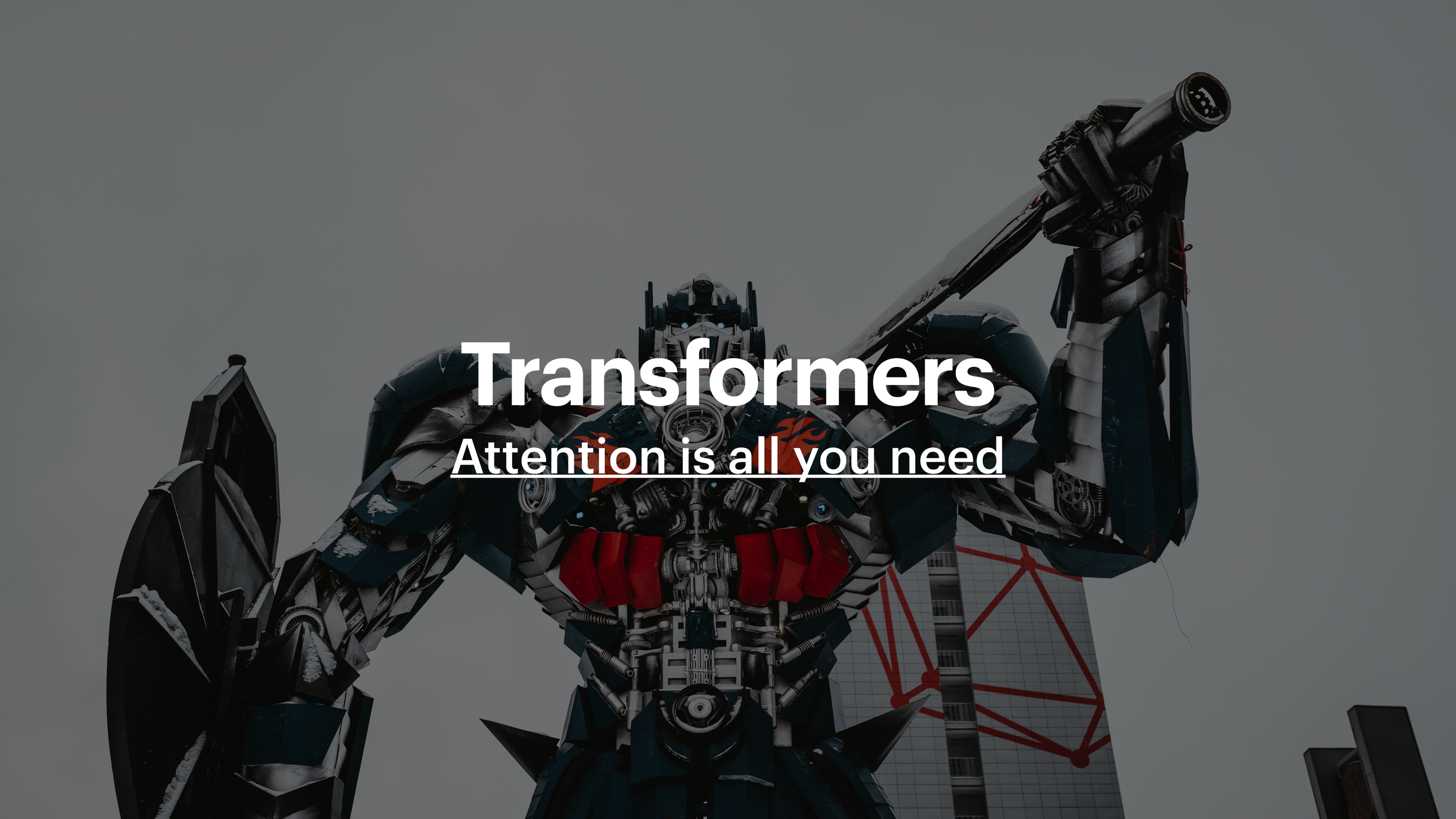
- Information bottleneck happens when the encoder tries to fit all info in one vector.
- Hard to parallelize due to sequential nature.
- Can't see both direction (even in BiDirectional) as every direction network sees in one direction only.

Bi-directional RNN

seq2seq

- This is considered as one layer but in facts two LSTMs are in work here.
- Every LSTM sees in one direction and can't see the whole sentence while encoding.




Transformers
Attention is all you need

What is a transformer

Transformers

- Transformer model was introduced by Google in the very famous paper Attention is all you need.
- The model is built based on attention and got rid of RNNs.
- Transformer model reshaped the NLP industry as it enabled a way to optimize large model by parallelizing model operations and making use of GPUs/TPUs.
- The original paper introduced the transformer in NMT task and had a better BLEU score than all previous SOTA models.

What is a transformer

Transformers

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

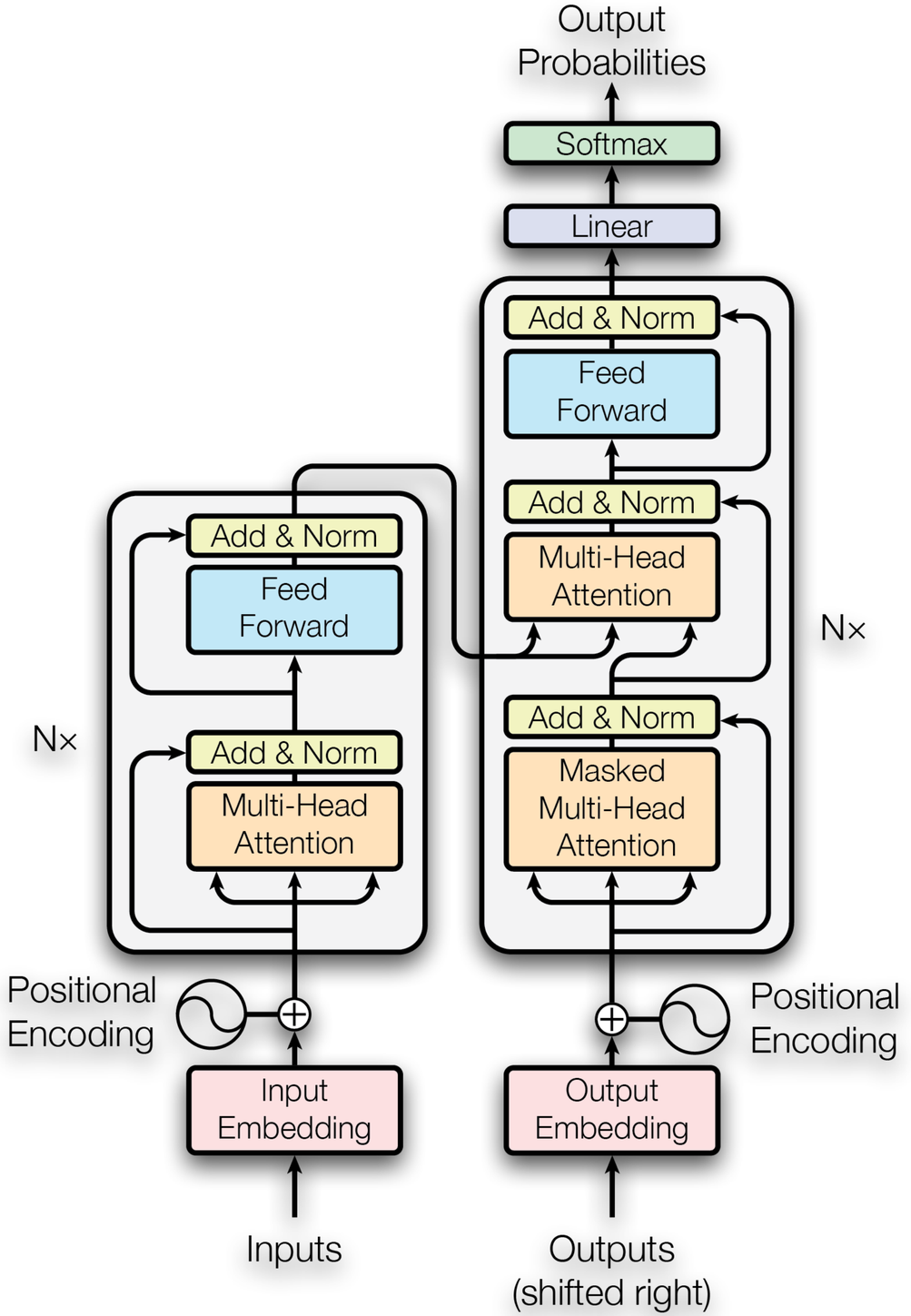
Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

source

Architecture

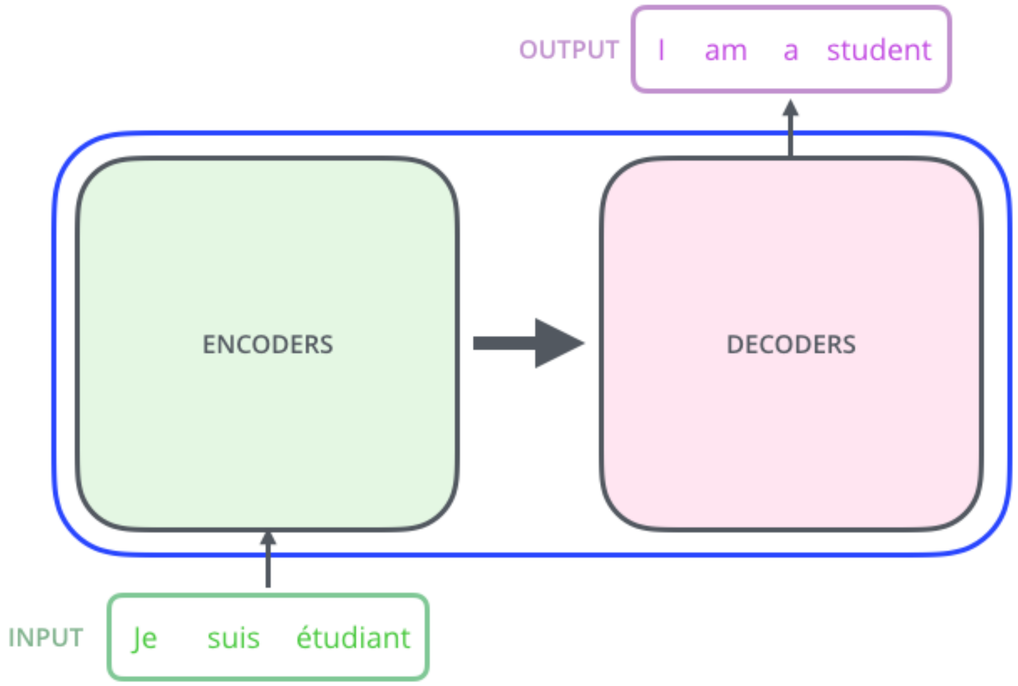
Transformer

- Positional encoding is used to addresses the position.
- Multi-head attention is used to learn context of words.
- All the components can be parallelized and thus make use of large hardware/data.



How it works
Transformer

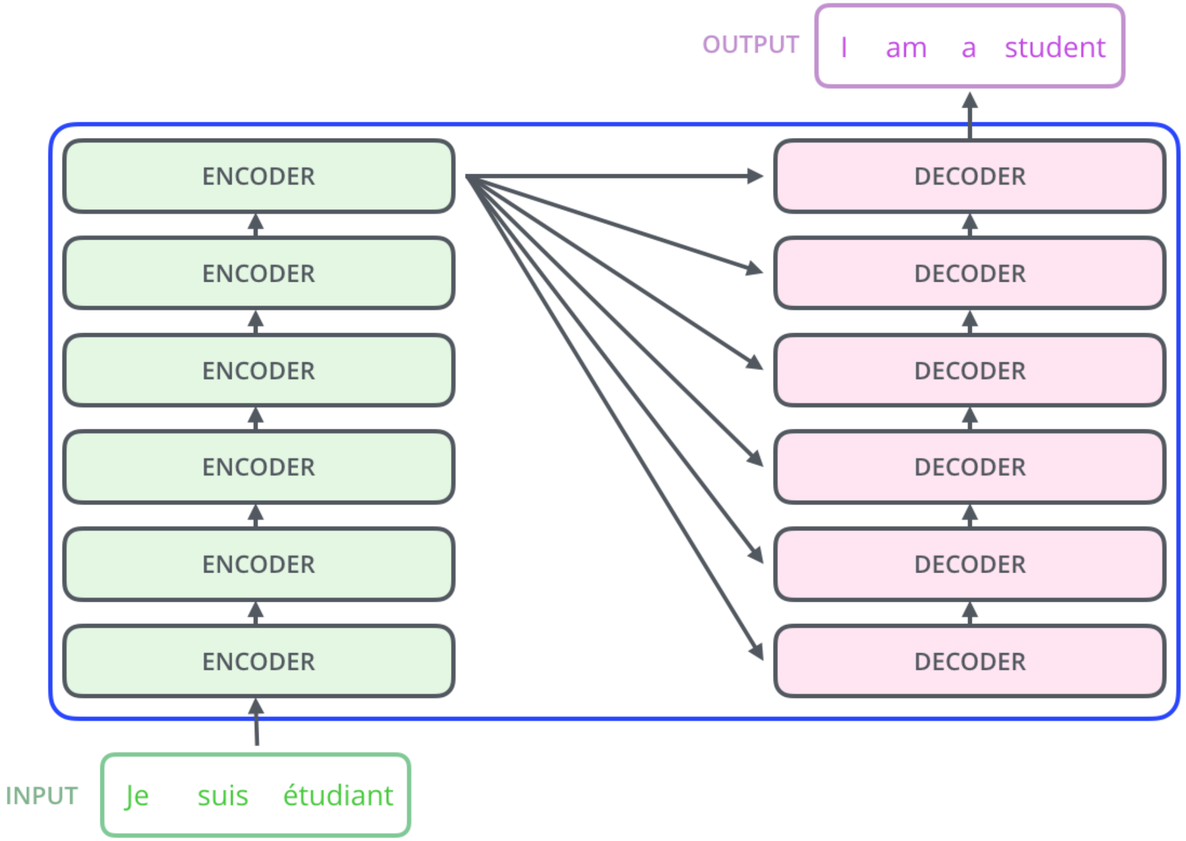
- The main idea is the same here, seq2seq model
- An encoder and a decoder are used to generate context and generate translation.
- Attention is the core building block



source

How it works
Transformer

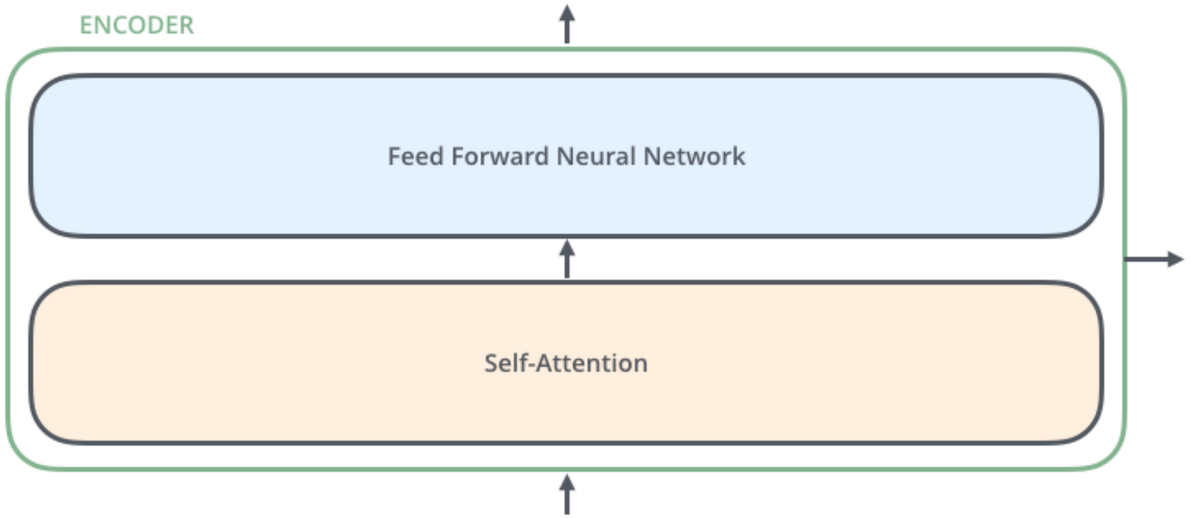
- encoder and decoder are a stack of multiple ones (6 layers in the original paper)



source

How it works
Transformer

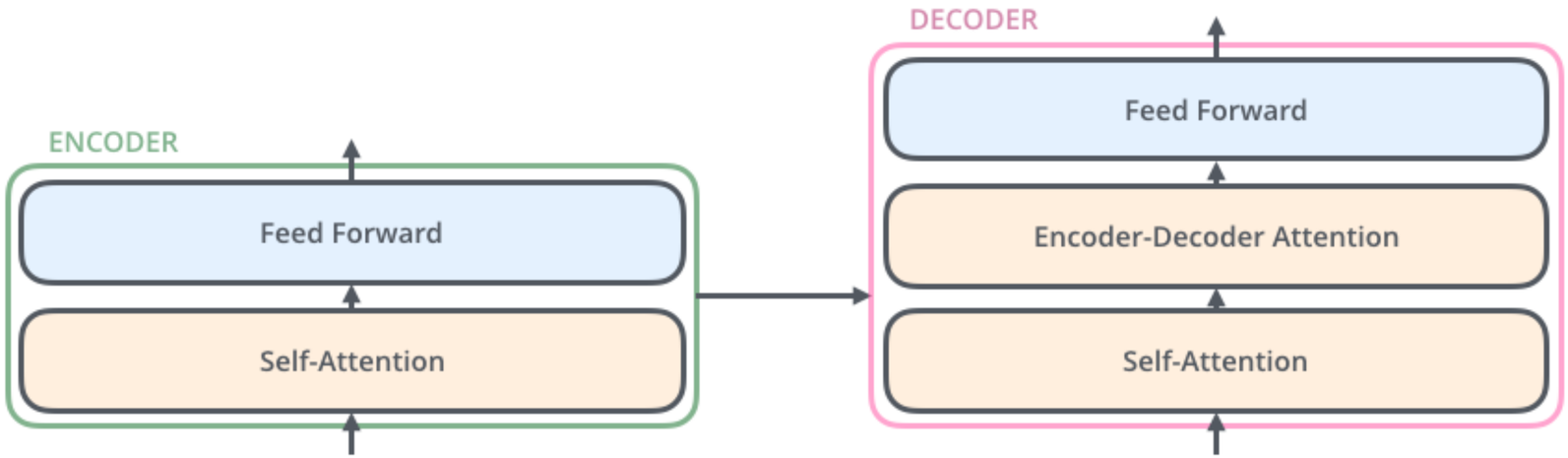
- The encoder itself consists of a **self-attention** layer and a FCN.
- **self-attention** layer helps the encoder look at other words in the input sentence as it encodes a specific word.



source

How it works
Transformer

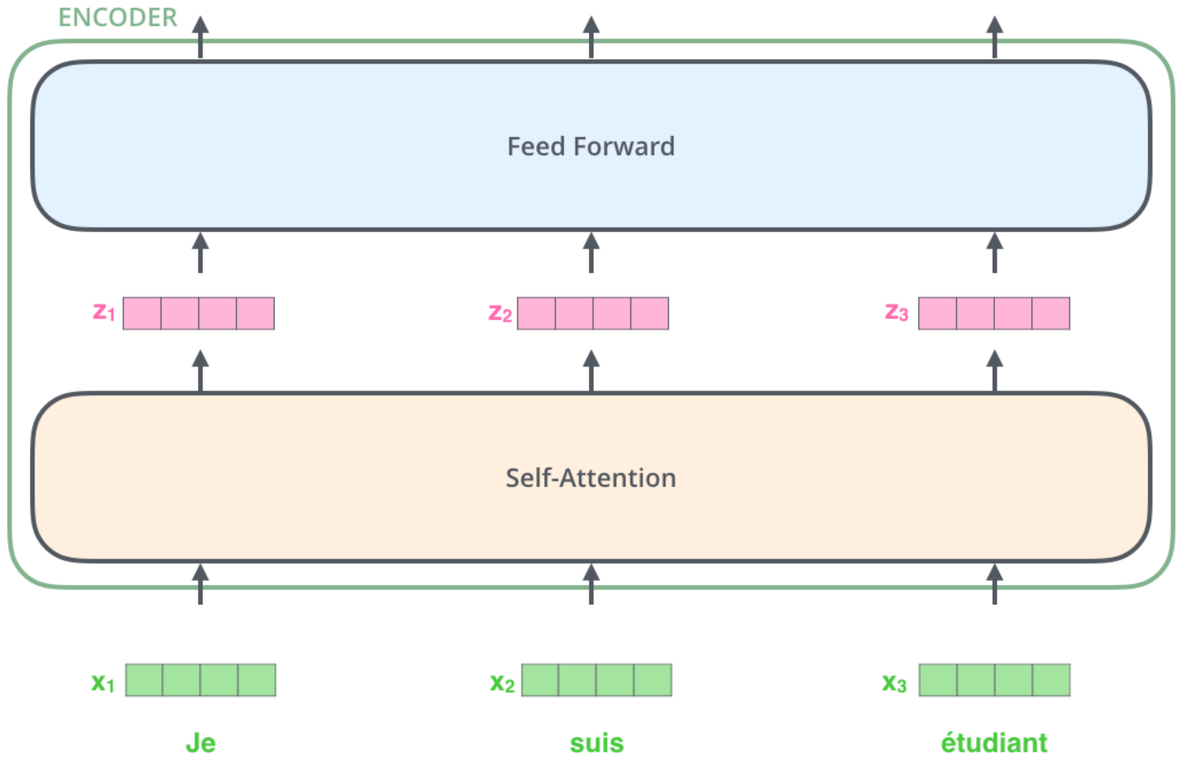
- Decoder is the same as encoder with and addition of an attention layer that helps it focusing on important parts from the encoder.



source

How it works
Transformer

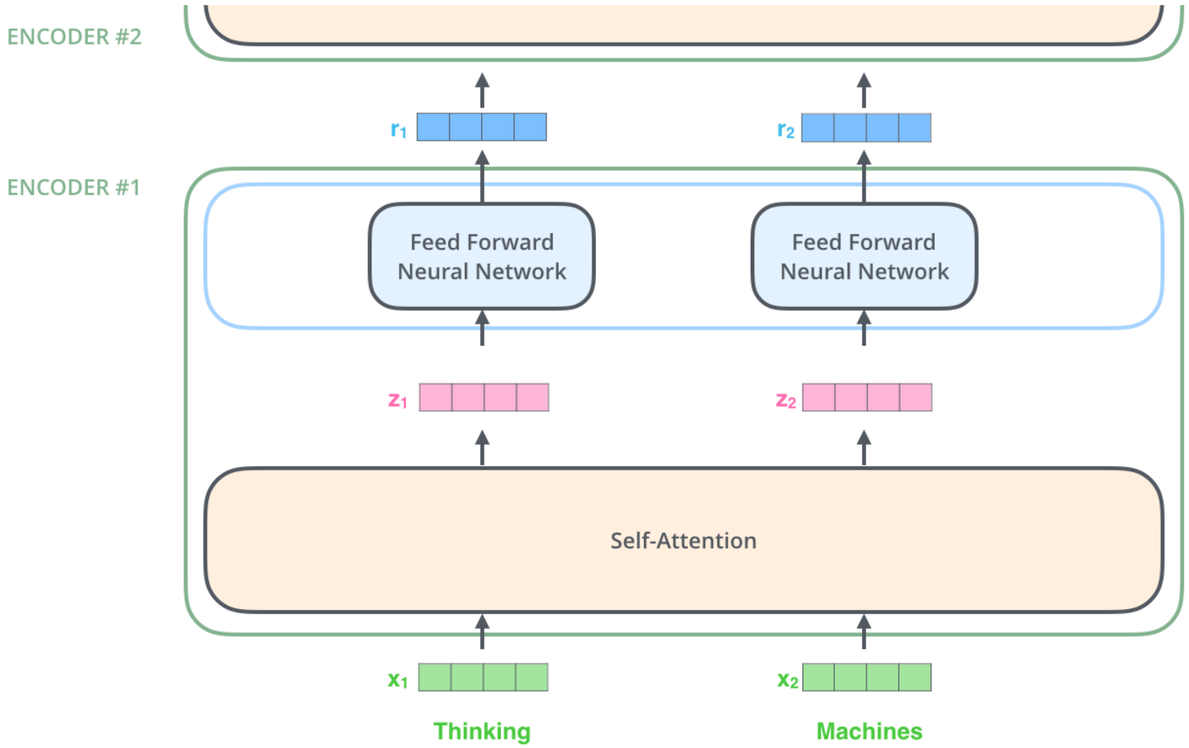
- The first encoder input is the word embeddings for the input words.
- Words enters in form of vectors (embeddings)



source

How it works
Transformer

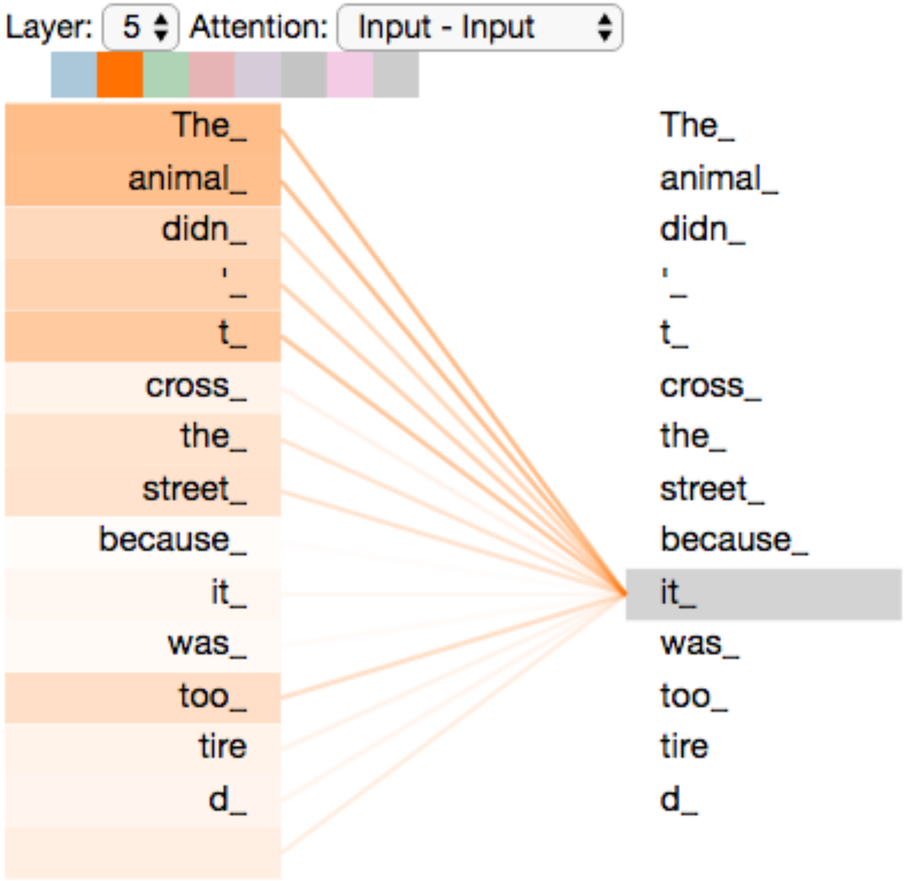
- following encoders input is the output of the previous encoders.
- You can see already each word goes to it's own path, **no dependency** is required between each word and it's previous one, which makes it easy to parallelize this model.



source

Self-Attention
Transformer

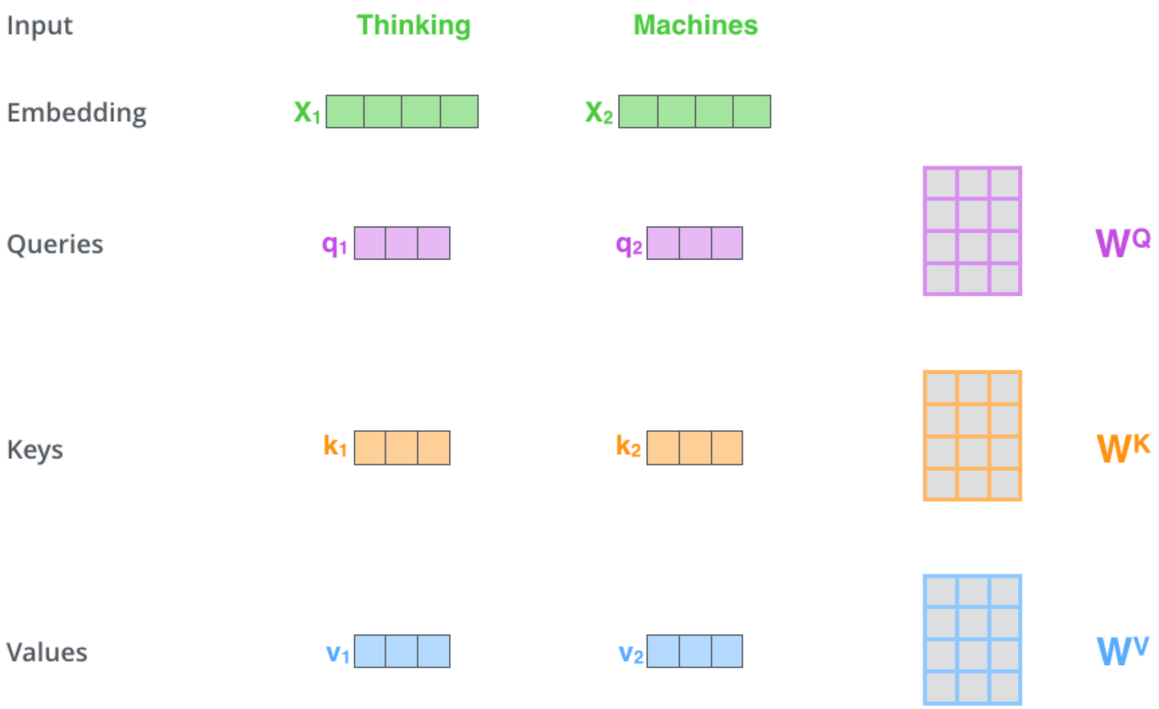
- self attention allows the model to look at other positions in the input sequence for clues that can help lead to a better encoding for this word.
- It help changing the meaning/ representation of the word based on it's context.



source

Self-Attention
Transformer

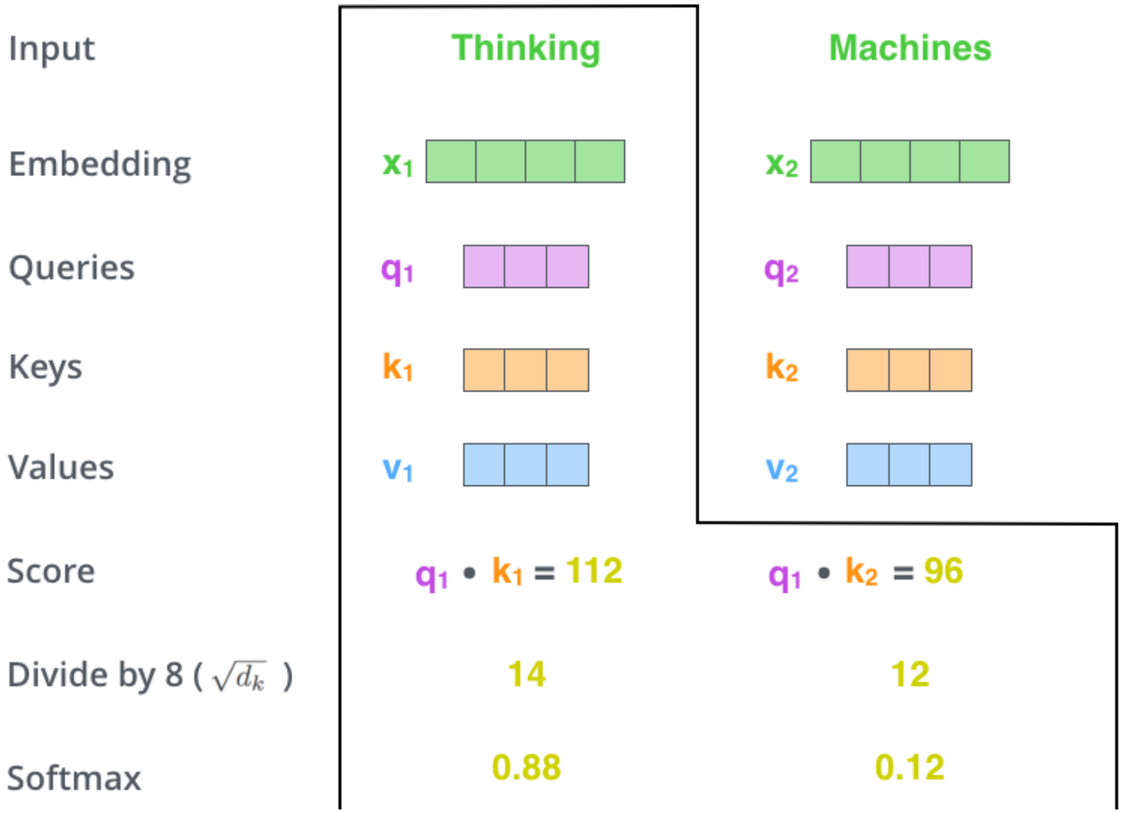
- It works by creating three vectors for each word.
- Query, key and value.
- Each one is created by a weights matrix that is learnable.
- multiplying **x** by and **W** generates the corresponding vector.



source

Self-Attention
Transformer

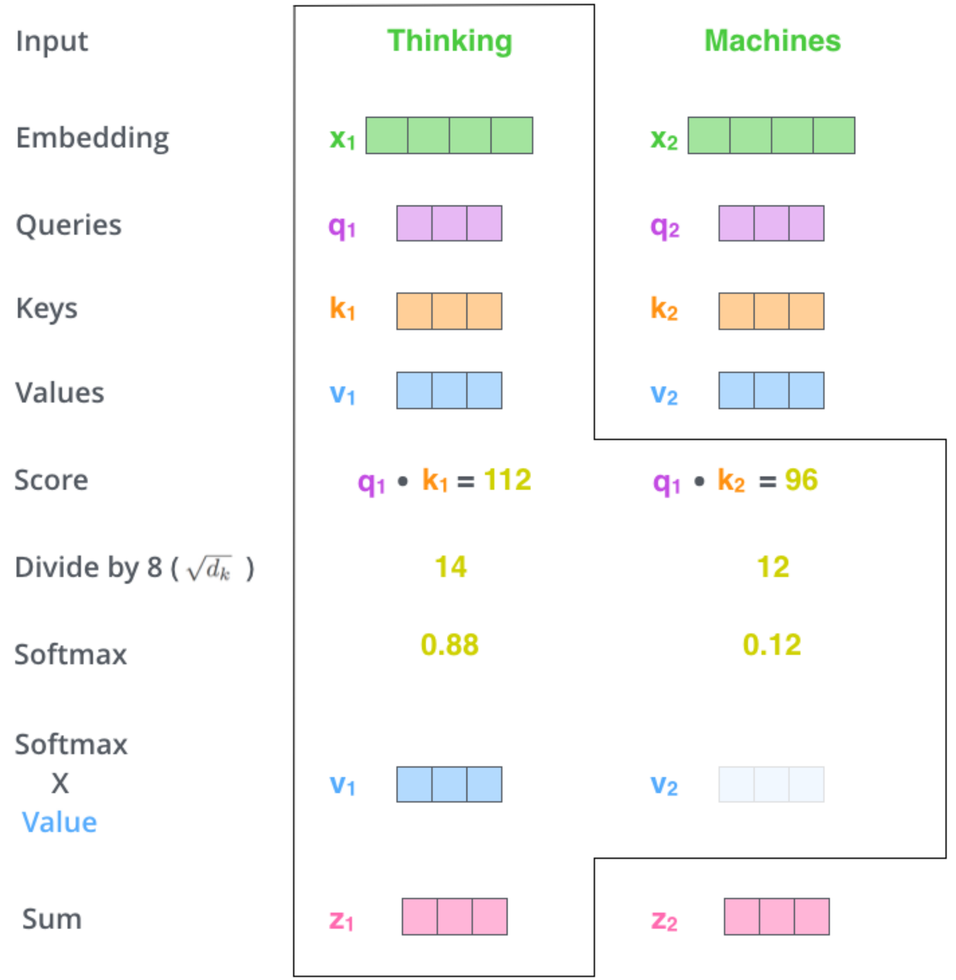
- To calculate self-attention for first word "Thinking" we multiply it's **Query** vector by all the **Key** vectors of every token to get a score.
- The score is then normalized and softmax-ed to get the weight of each key word w.r.t the query word.



source

Self-Attention
Transformer

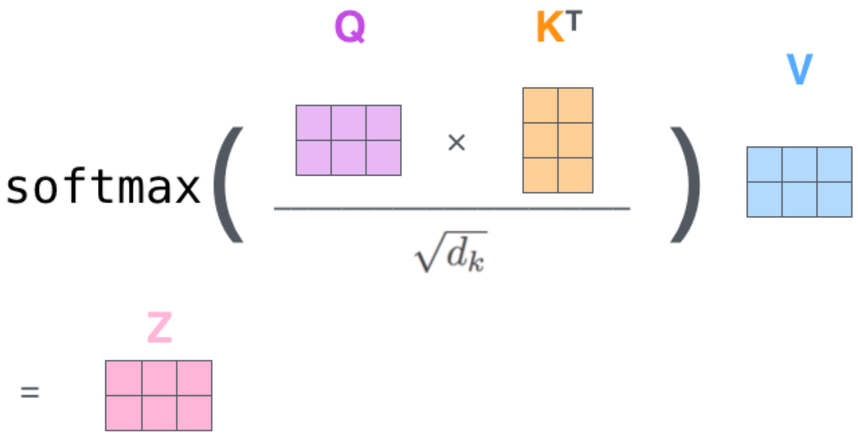
- The **Value** vectors are then scaled by the score and summed per word.
- What we did for the first word is done for every single word, producing a **Z** vector for every word that carries meaning of it.



source

Self-Attention
Transformer

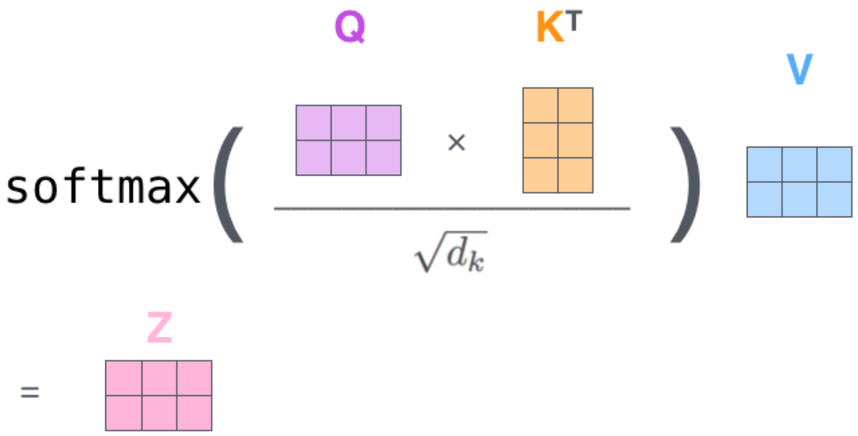
- The previous operations can be calculated as matrix multiplication as follows.

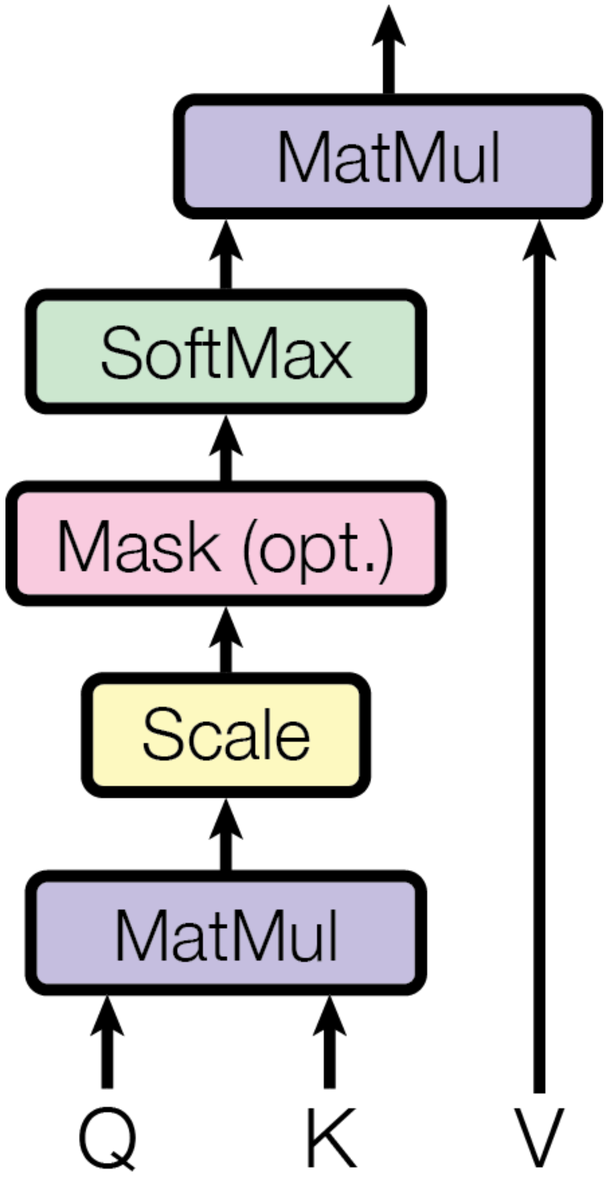


source

Self-Attention
Transformer

- Another annotation for the self-attention mechanism.

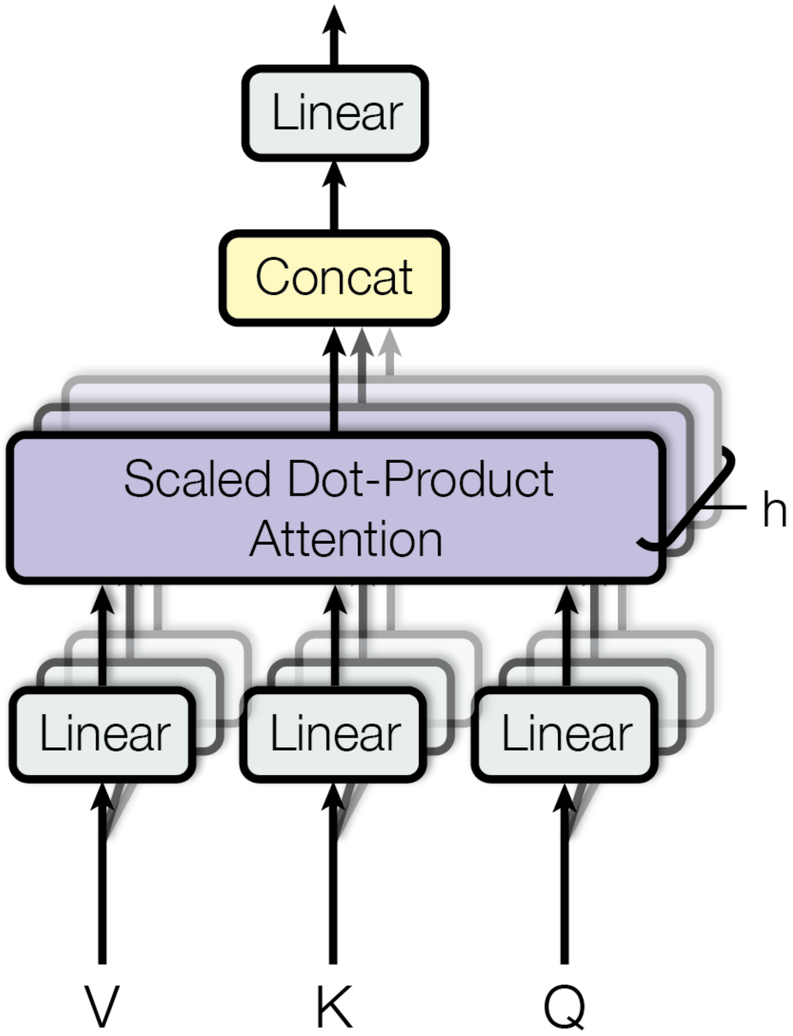




source

Self-Attention
Transformer

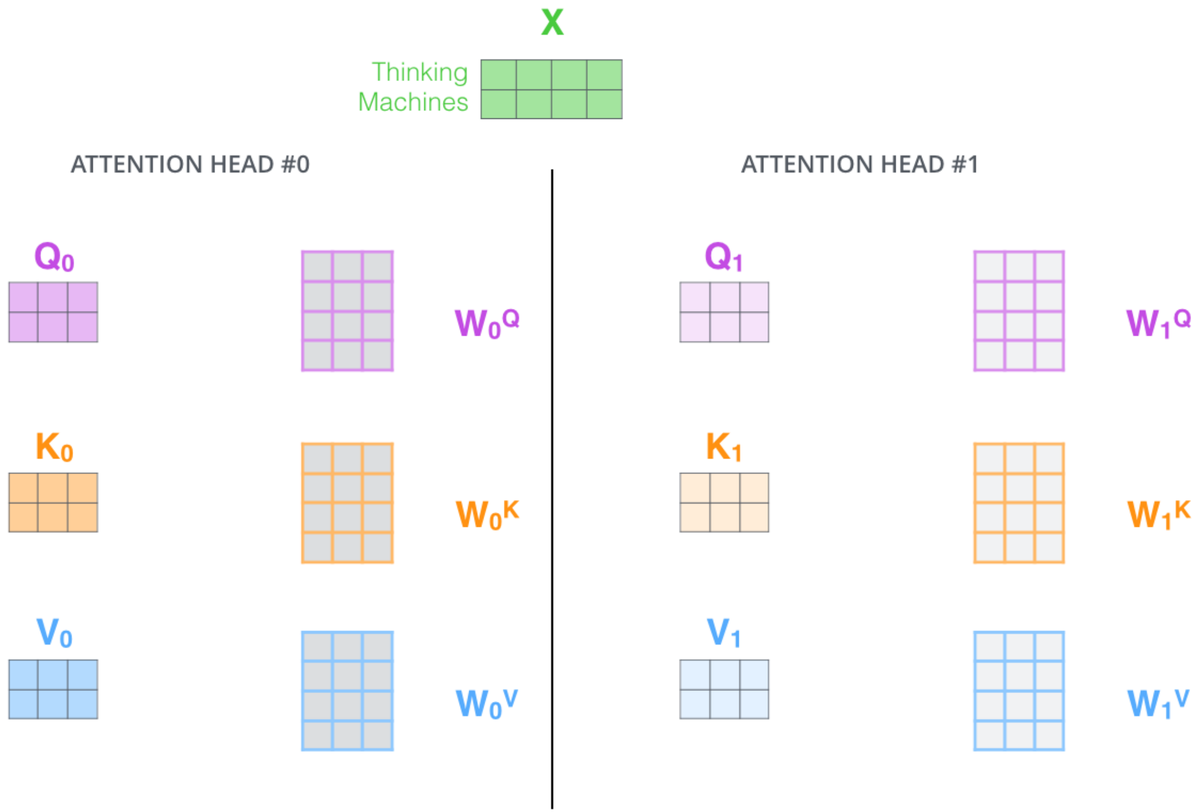
- A multi-headed attention is used.
- This enables multiple representation space for the features.
- Think of it as more layers in NN, it adds more learning space to the model.



source

Self-Attention
Transformer

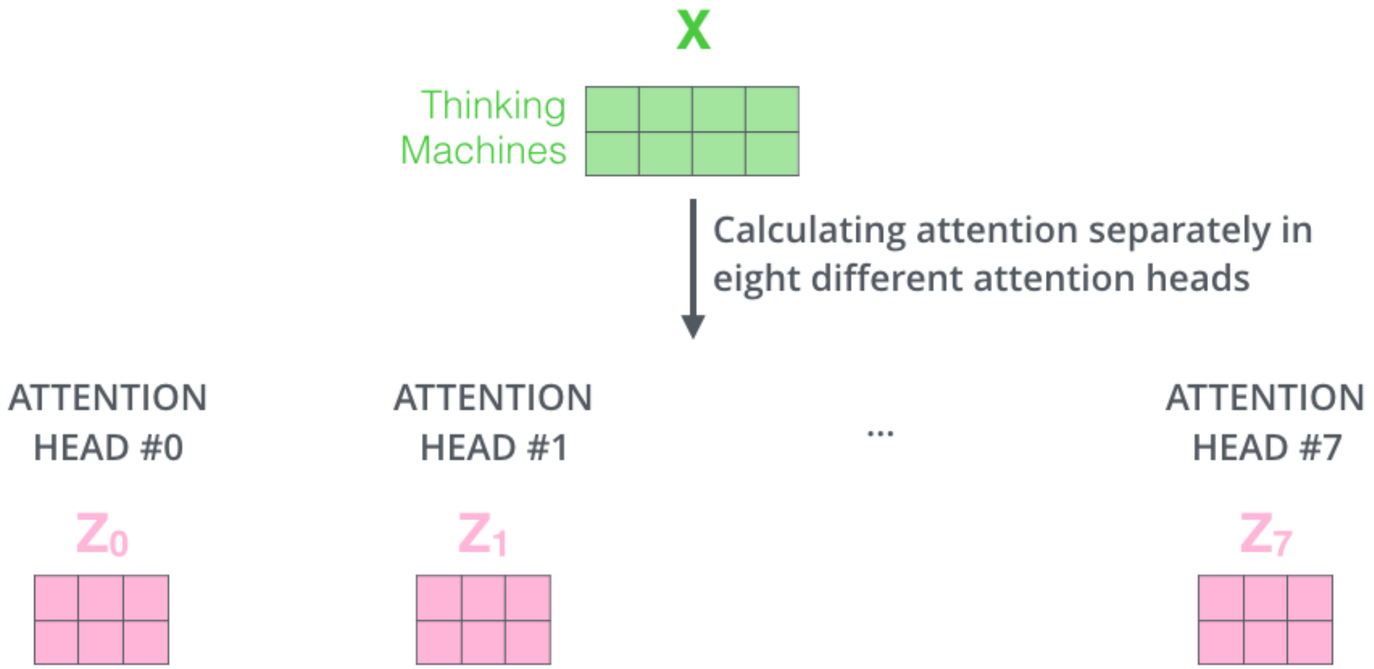
- A multi-headed attention is used.
- This enables multiple representation space for the features.
- Think of it as more layers in NN, it adds more learning space to the model.



source

Self-Attention
Transformer

- each head generate a different **z** vector.

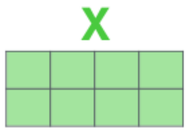


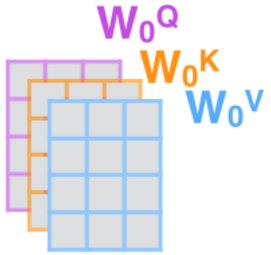
source

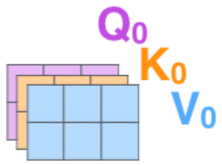
Self-Attention

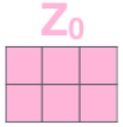
Transformer

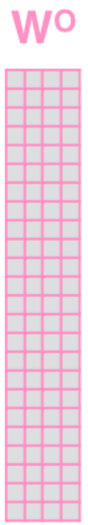
Thinking
Machines

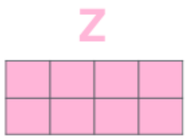




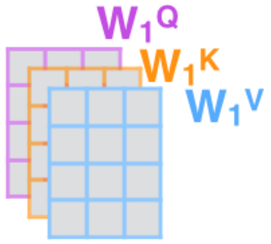


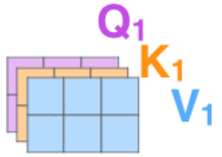


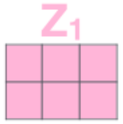


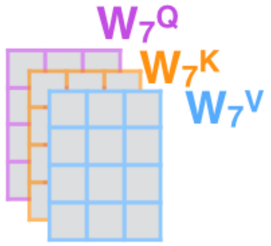


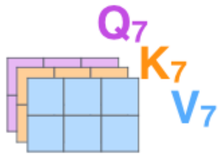
* In all encoders other than #0,
we don't need embedding.
We start directly with the output
of the encoder right below this one

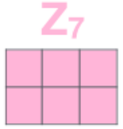








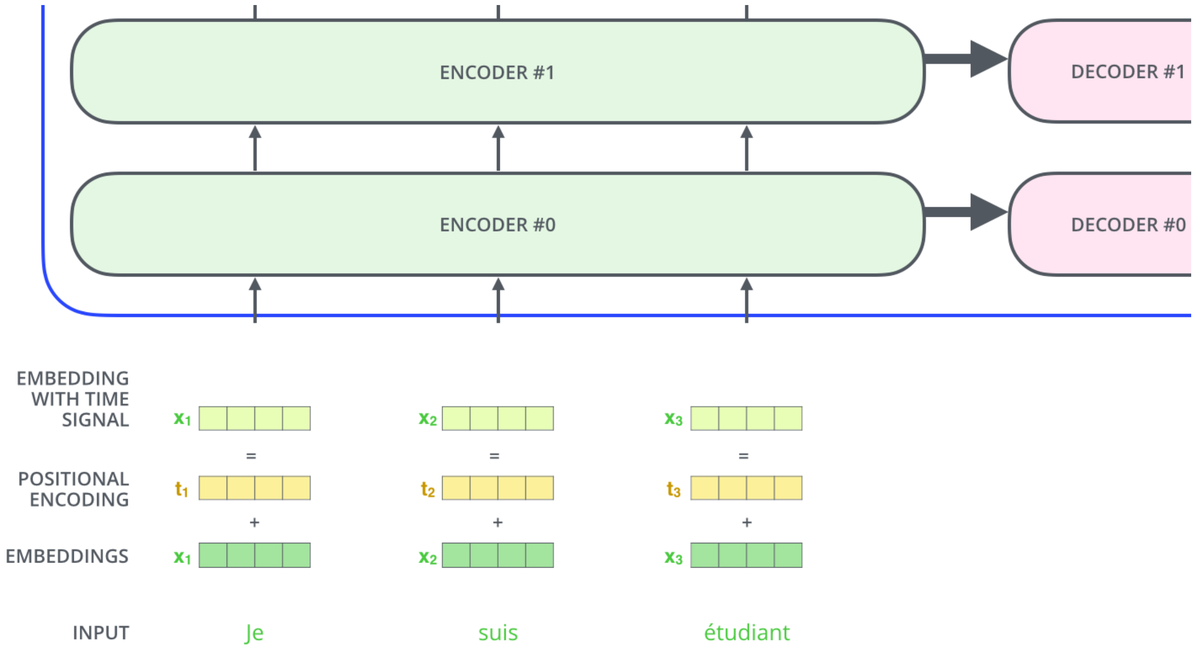




source

Positional Encoding
Transformer

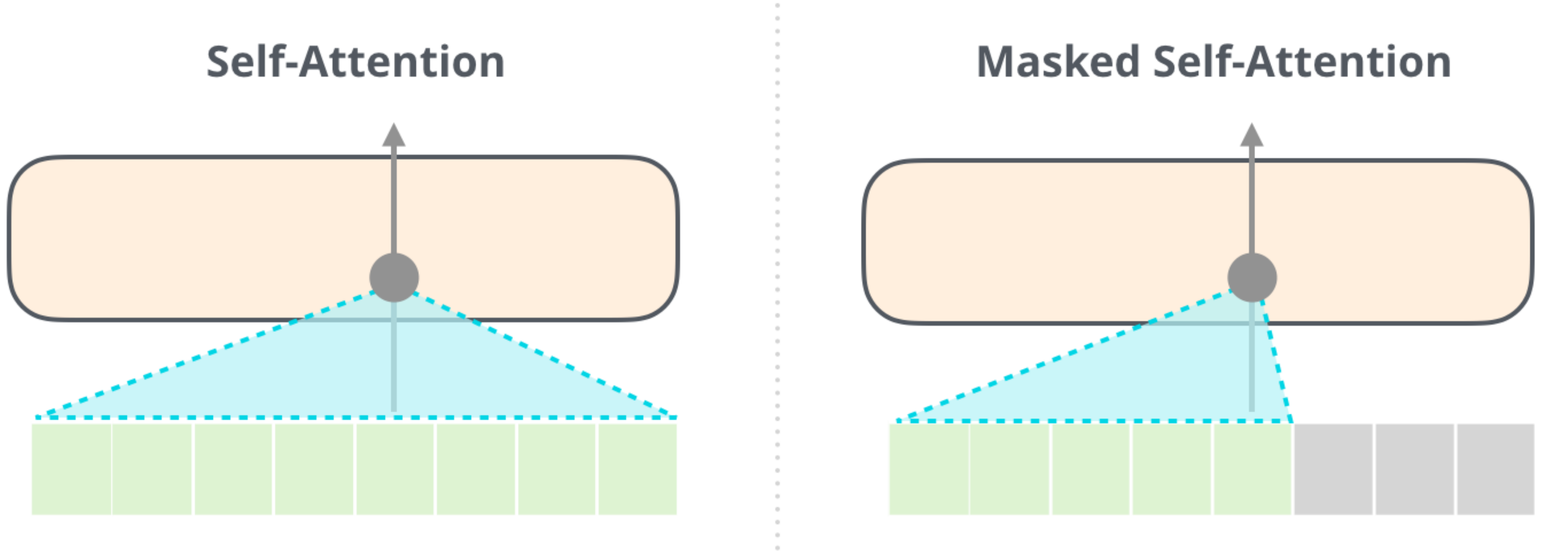
- To get a sense of position, we add a positional encoder to give the model a way to know if words are close to each.
- The positional encodings follow a specific pattern.



source

Decoder
Transformer

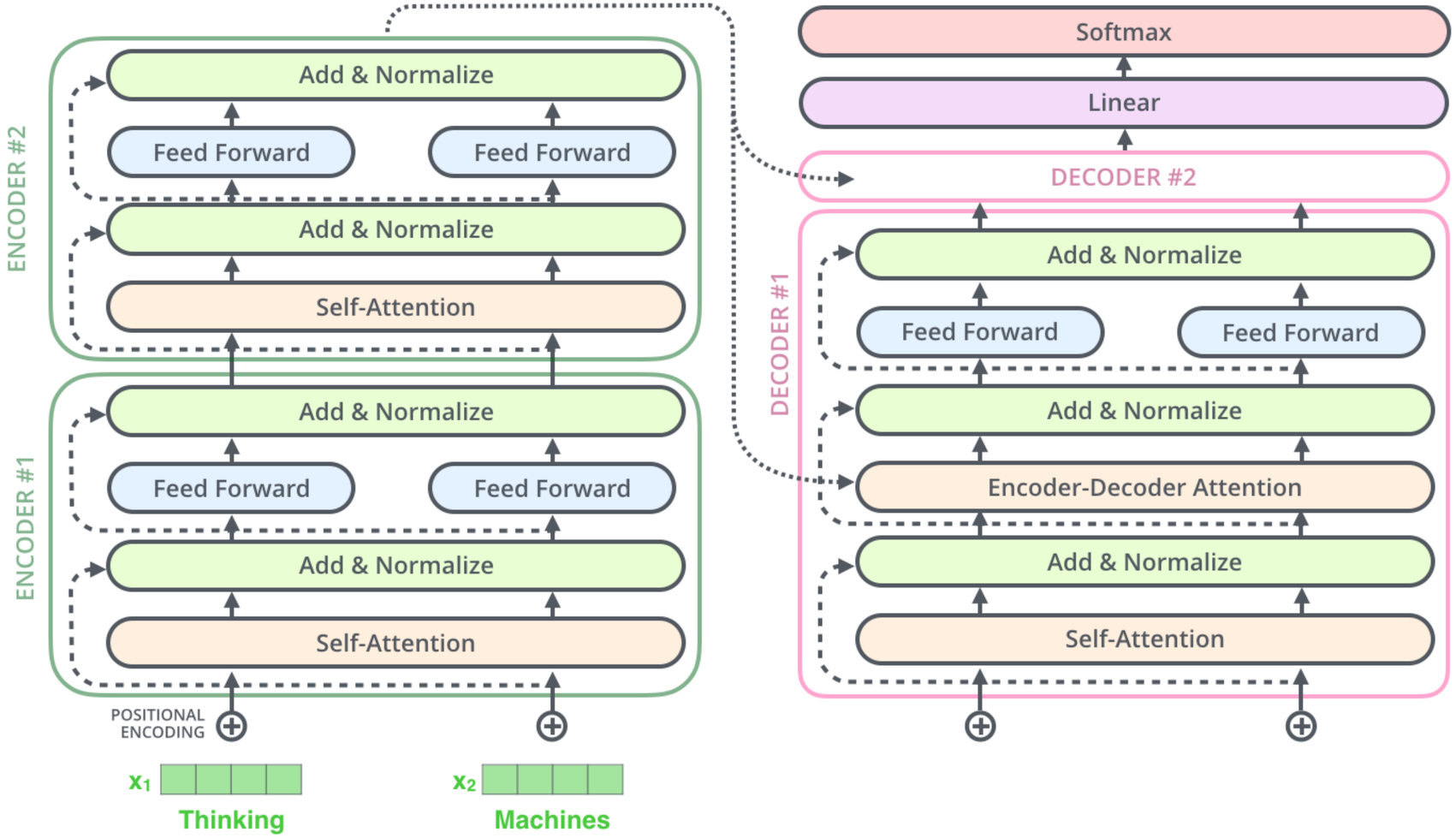
- The decoder works exactly as the encoder expect for the mask.
- Every future token is masked so the decoder can't attend to future tokens while training, as they won't be available in inference.



source

Decoder

Transformer

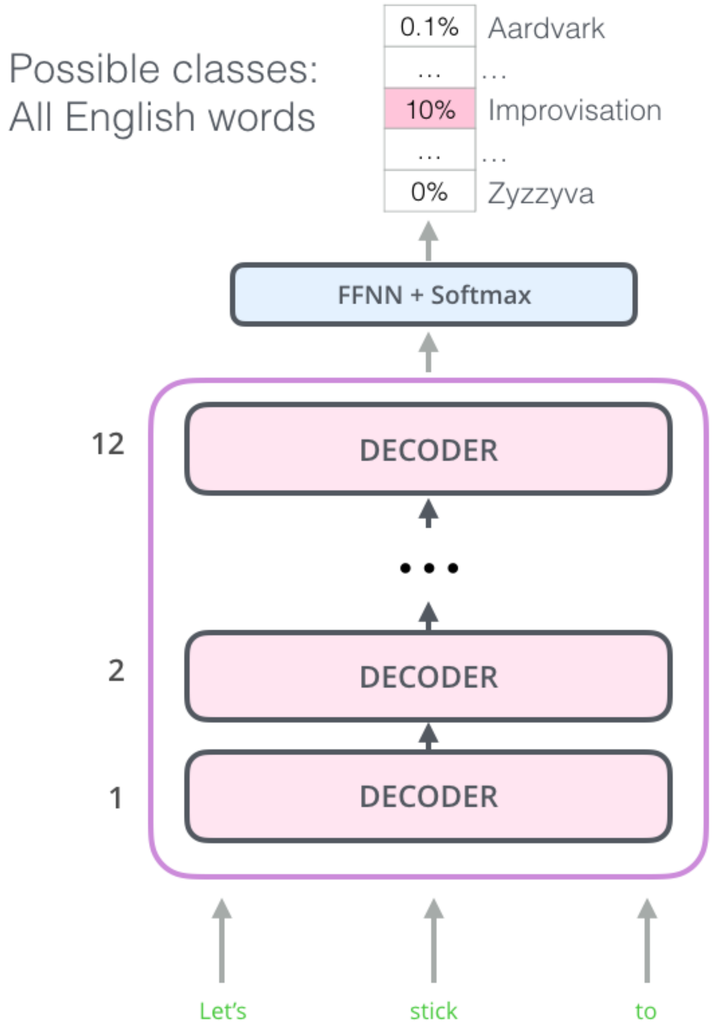


source

The dawn of transformers

OpenAI: GPT
Transformers

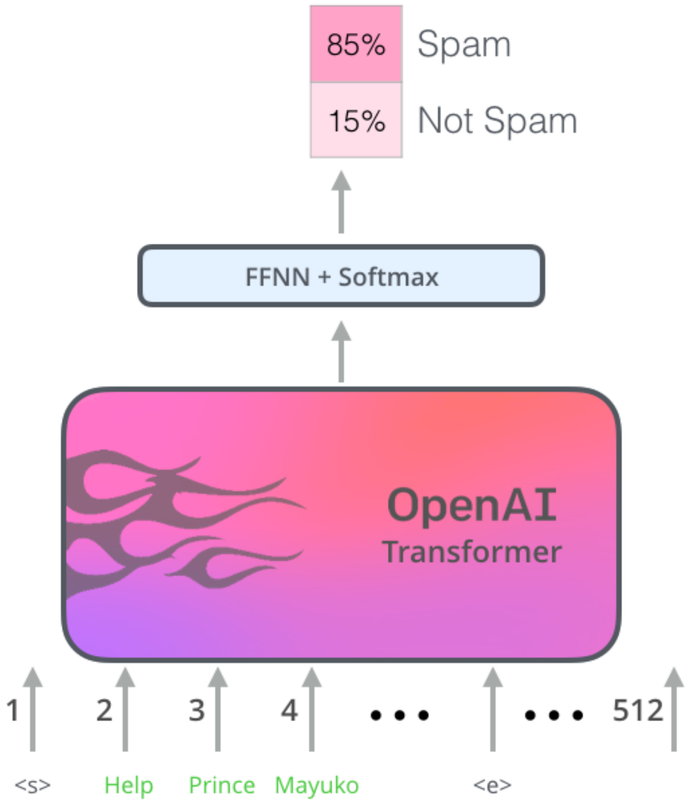
- It uses a stack of 12 decoder unit.
- No encoder is used.
- It trains on language modeling task.
- The objective is to predict the next word.



source

OpenAI: GPT
Transformers

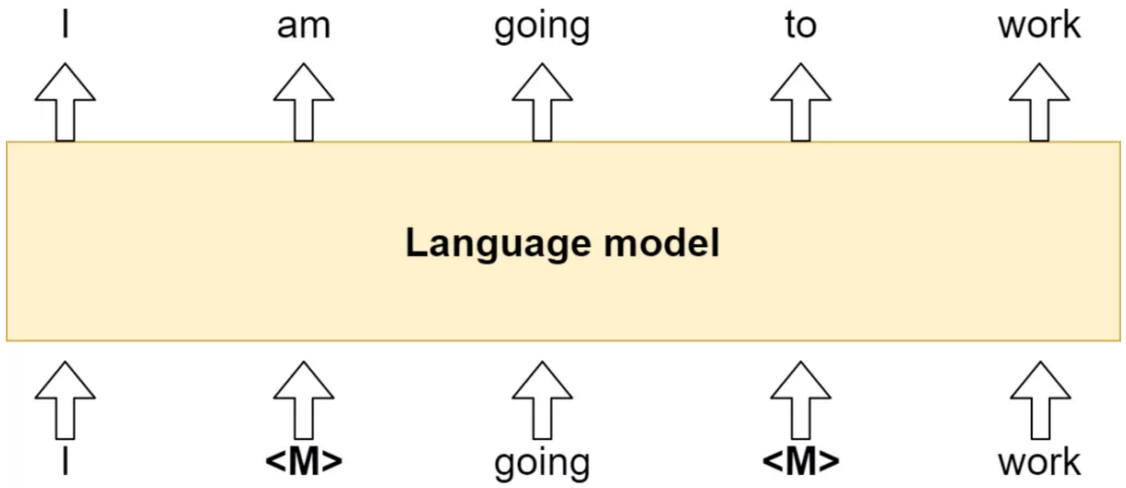
- The model is then used in downstream tasks.
- The decoder has a drawback, it can't attend to future tokens.



source

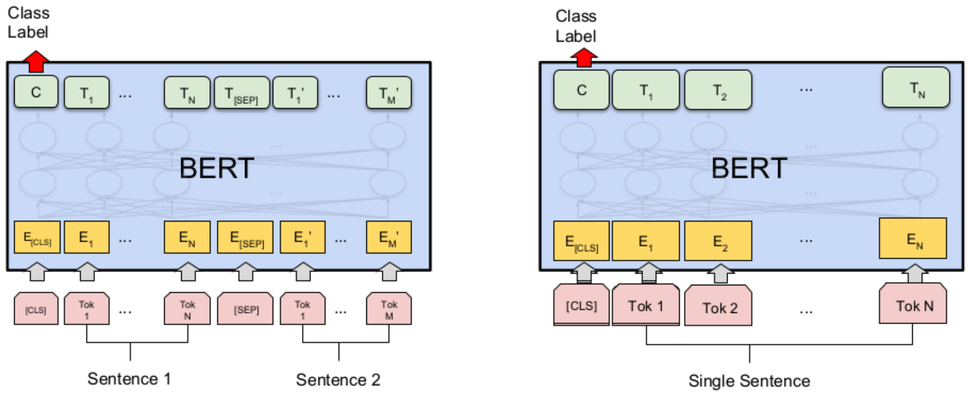
Google: BERT
Transformers

- Uses encoder instead of decoder.
- Trained on masked language modeling task, BERT objective is to predict masked token.



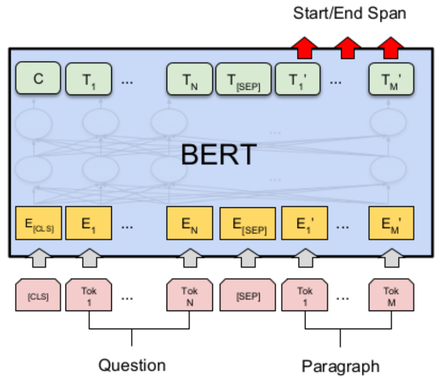
Google: BERT
Transformers

- BERT can be used for almost every NLP task

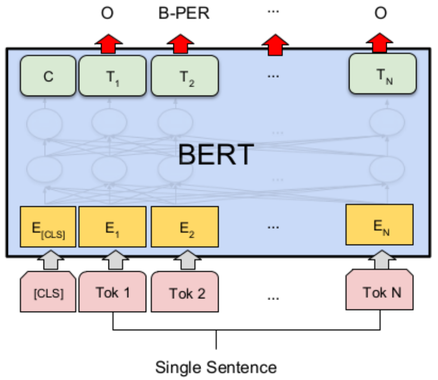


(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA



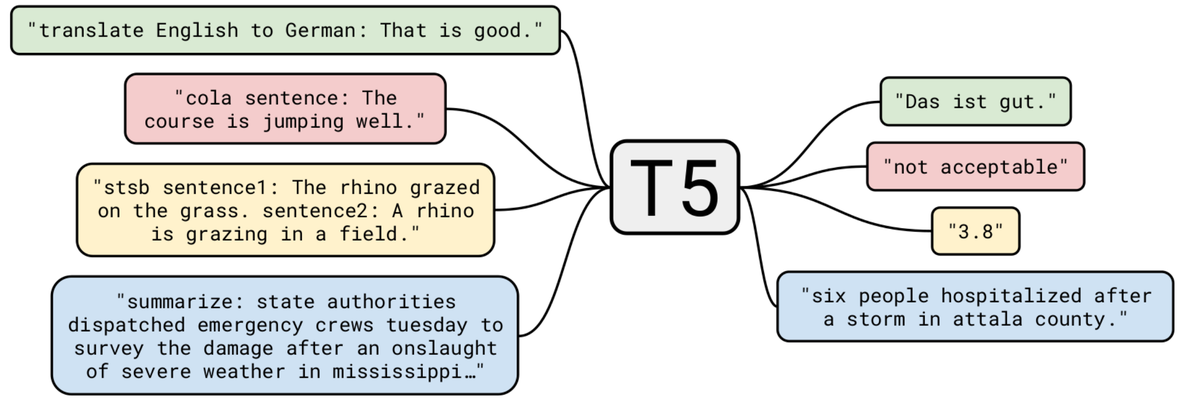
(c) Question Answering Tasks: SQuAD v1.1



(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Google: T5
Transformers

- T5 was introduced by Google in early 2020, it's an encoder-decoder architecture.
- The model was trained on the Colossal Clean Crawled Corpus (C4) which is a huge corpus covering differing kind of tasks.
- The objective of the model is to fill the blank.



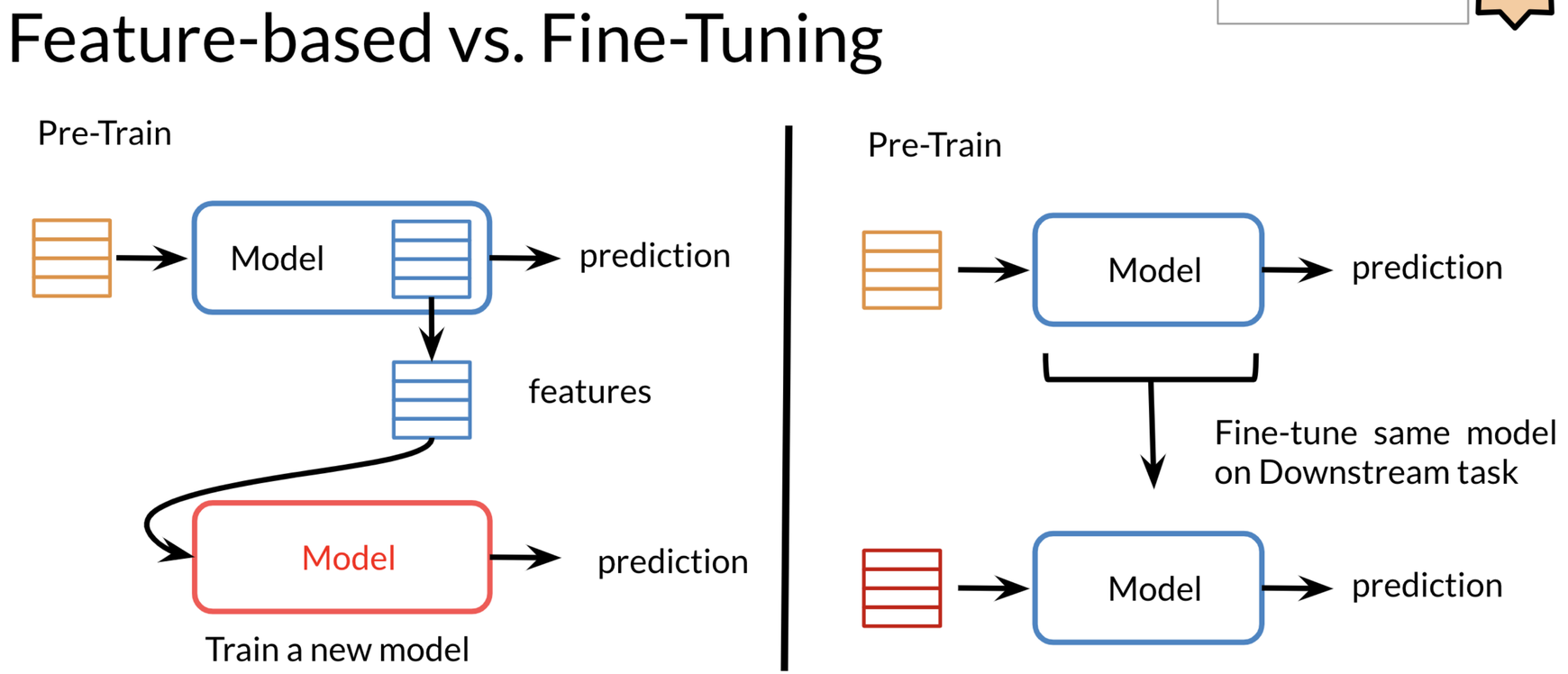
source

Transfer Learning

Feature-based vs Fine-Tuning

Transfer Learning

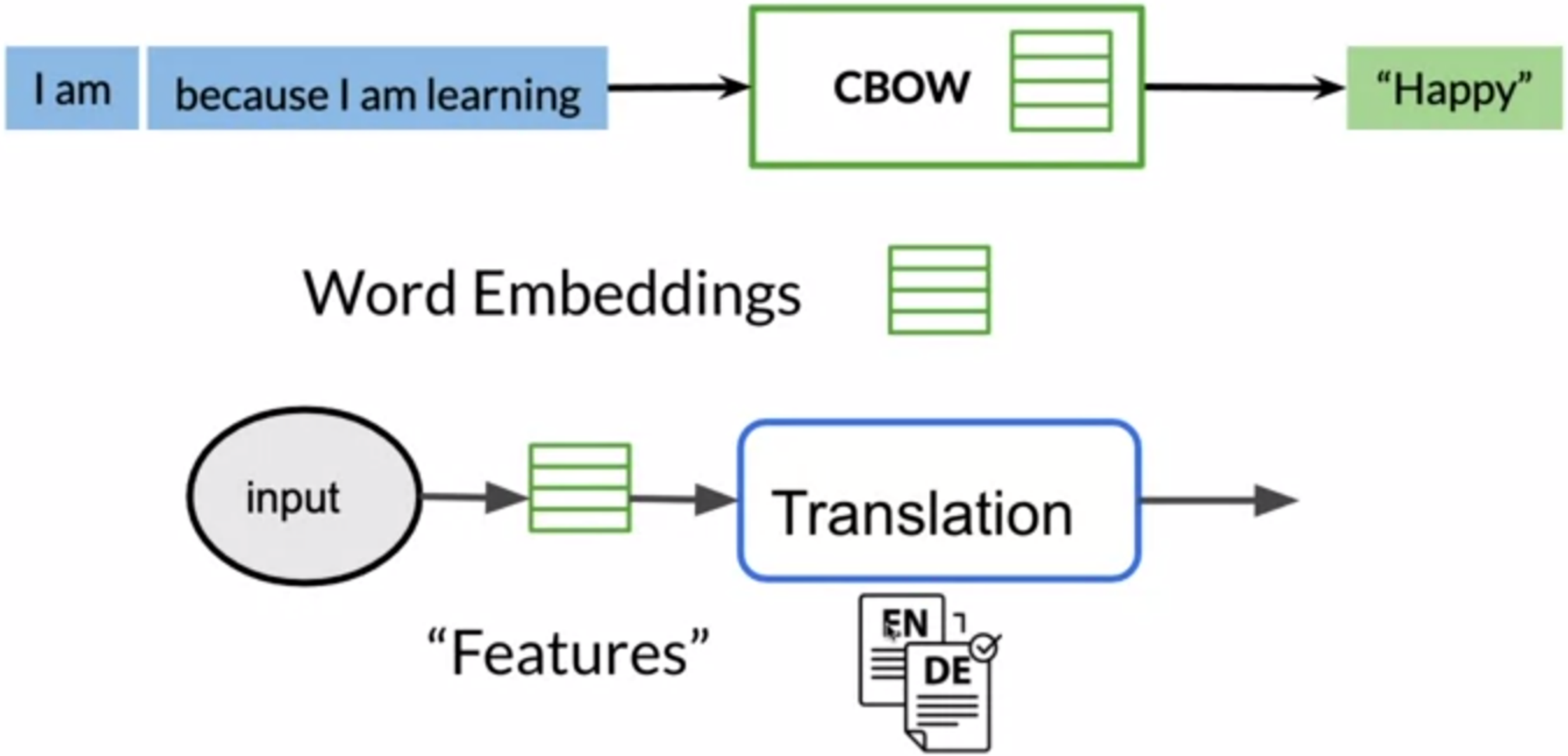
Transfer 1



source

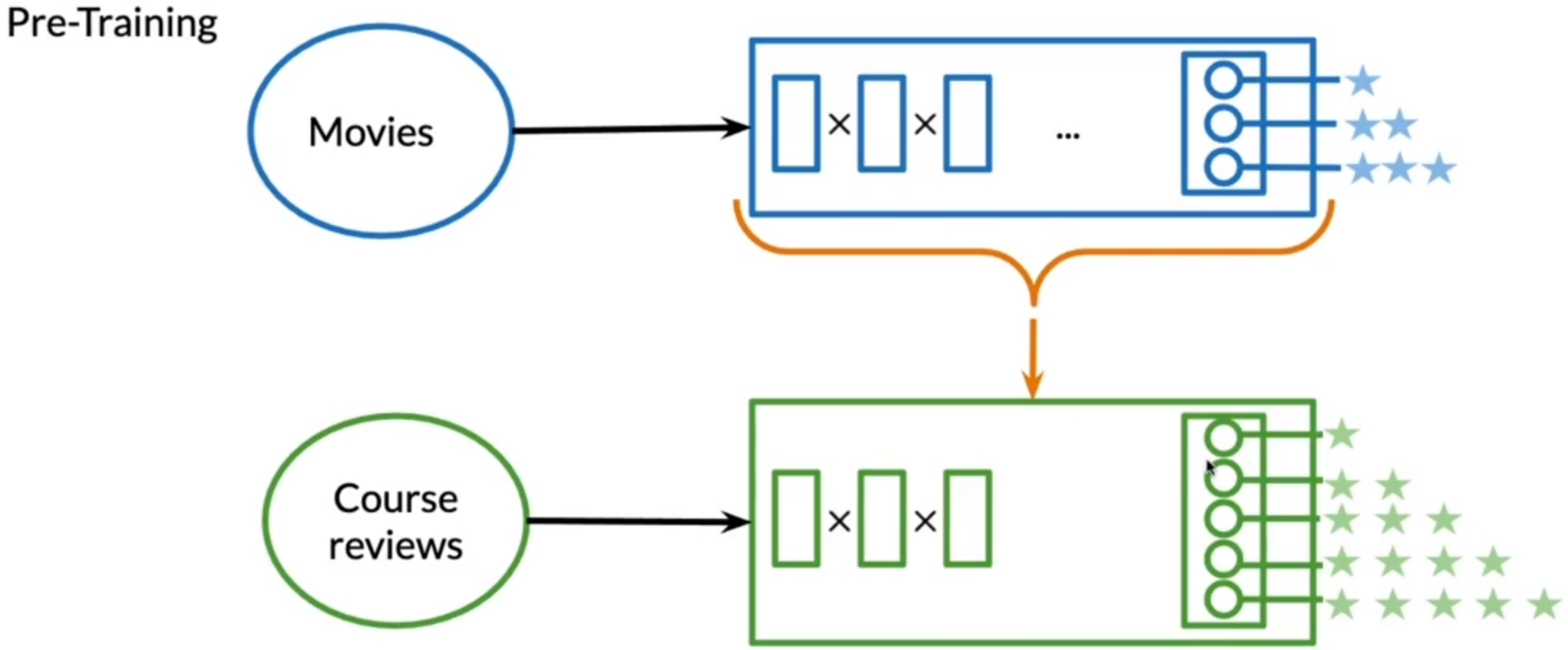
Feature-based

Transfer Learning



source

Fine-Tuning
Transfer Learning

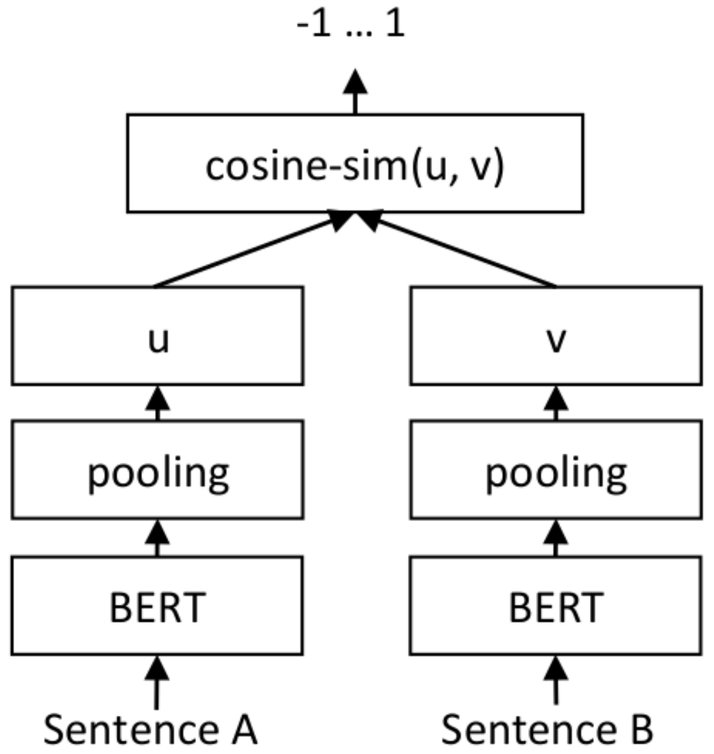


source

Siamese Network

Transfer Learning

- We can use Siamese Network to fine-tune a model on a specific domain.
- This architecture is very useful in sentence similarity tasks.
- We can transfer model knowledge to new domains.

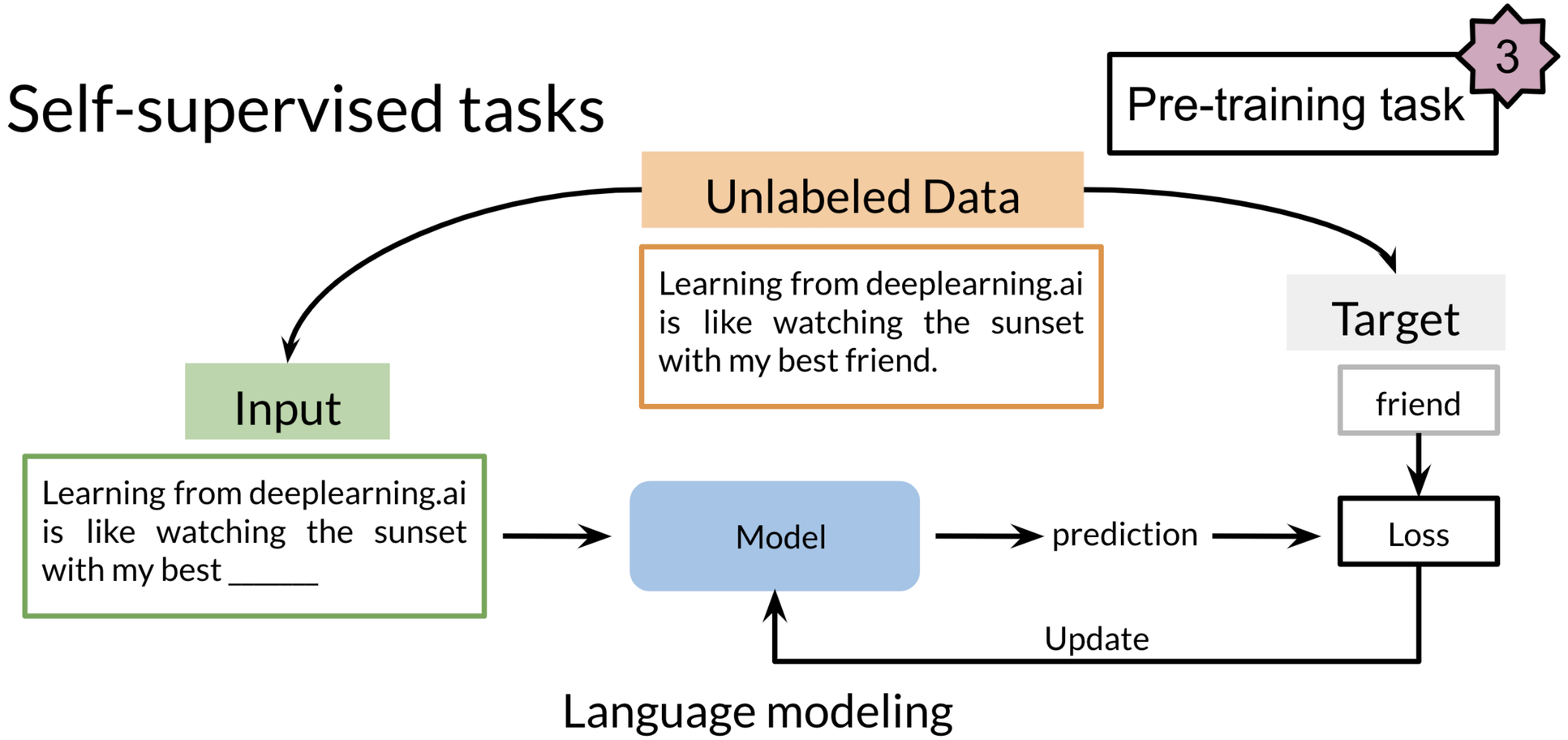


source

Self Supervised Learning

Self-Supervised learning

Transfer Learning



source

Knowledge Distillation

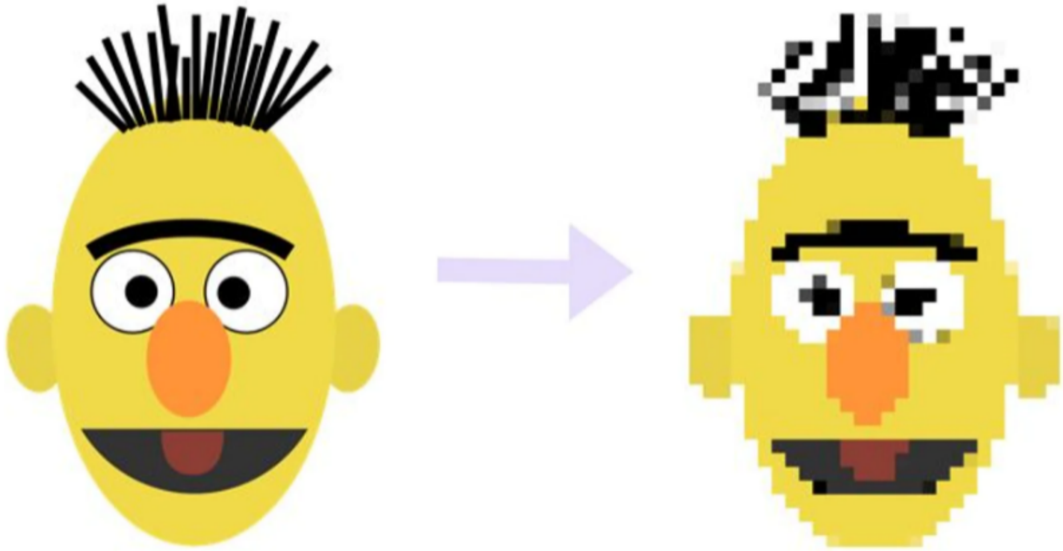
Transformers limitations

Knowledge Distillation

- Transformer based models are huge in terms of compute power and memory consumption.
- They need GPUs to offer a reasonable inference time.
- We can reduce the number of parameters (weights), or reduce the precision of them for example going from **float64** to **float16**.

Quantization
Knowledge Distillation

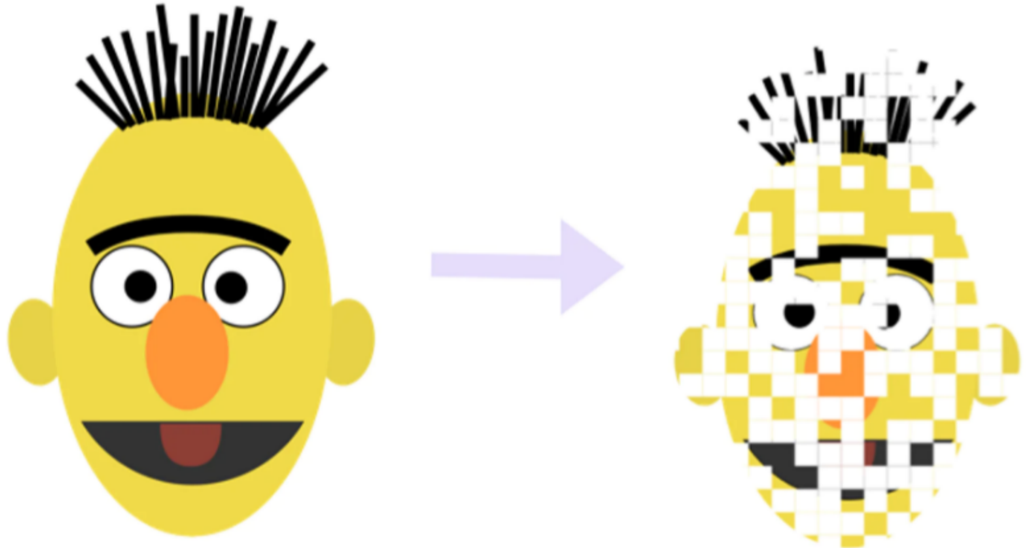
- Quantization works by reducing the precision of the weights to 8-bits.
- This significantly reduce memory usage of the model but also the accuracy of the model.
- The trade-off most of the time is worth it.



source

Pruning
Knowledge Distillation

- Pruning remove some weights that is very insignificant.
- The importance of the weight is decided based on it's value, when it is close to 0 it's not important.
- The enhancement depends on the sparse matrix implementation that you use.
- Also some pruning techniques removes neurons or even a complete weight matrix (like remove an attention head)

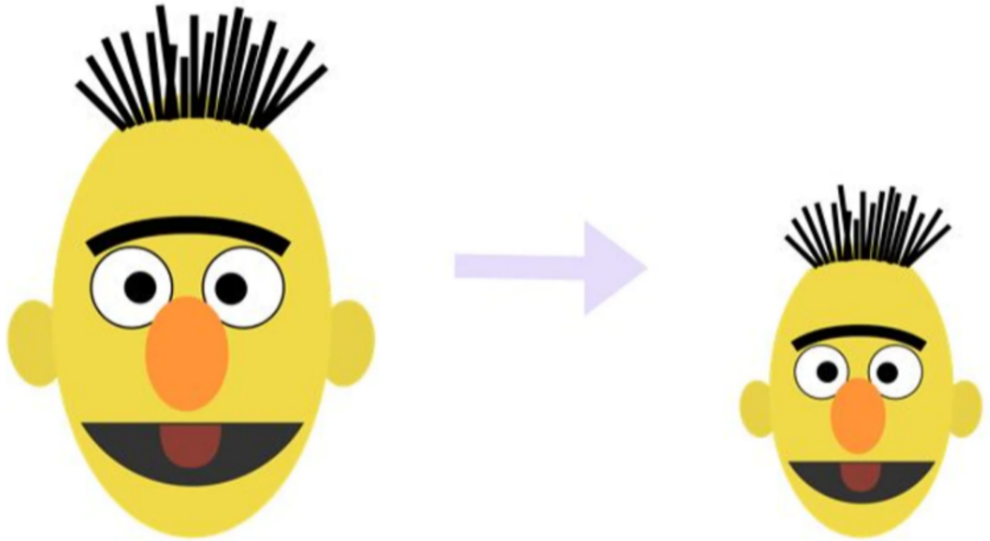


source

Knowledge distillation

Knowledge Distillation

- After training a big and slow model (the teacher), a smaller model (the student) is trained to mimic the teacher's behavior - whether its outputs or its internal data representations.
- The student trains to copy the knowledge of teacher.
- The loss of the model is a cross-entropy over the soft targets (probabilities of the teachers) not the hard targets.



source

HuggingFace

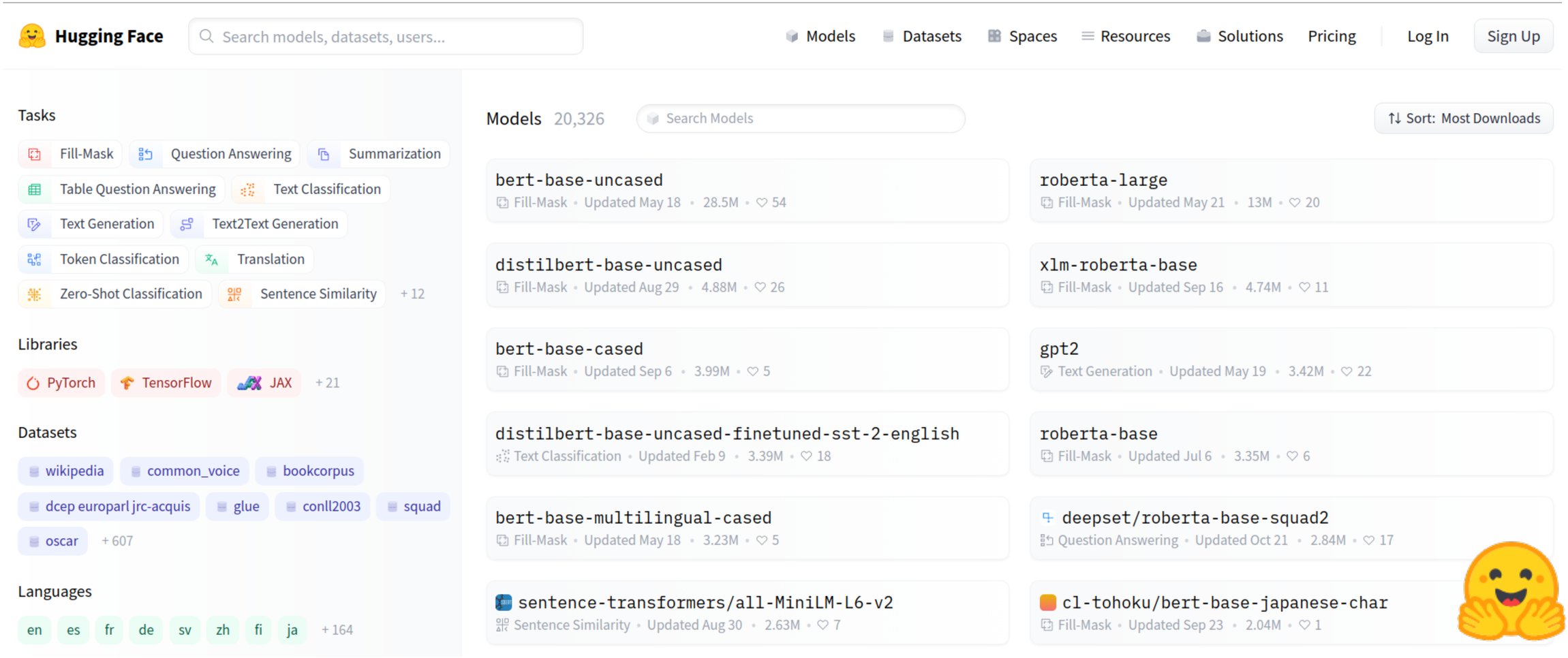
HF Hub

HuggingFace

- HF is an open source community with a hub for wide range of pre-trained deep learning models.
- HF also provide wide range of datasets for lots of tasks.
- transformers, datasets are the most famous libraries huggingface has provided.

HF Hub

HuggingFace



Notebooks - 1

Transformers introduction

Notebooks - 2

Sentence Transformers

Notebooks - 3

ONNX

Questions ?

Thank you !