

Evolution of LLMs and the Current State of the Art.

Hai Le

Research Engineer, NLP Lab @ **Skoltech**
Invited Lecturer, **HSE**

Who Am I?

- Hi! My name is Hai (no pun intended)
- Graduated from **University of Maryland** (USA) & **Skoltech**
- Currently a NLP researcher at **Skoltech**, and an incoming PhD student at **Imperial College London**
- Research Interest: **human-centered language modelings**:
 - How can we better **understand** LLMs?
 - How can we make LLMs more **accessible** to researchers and engineers



Skoltech

**Imperial College
London**

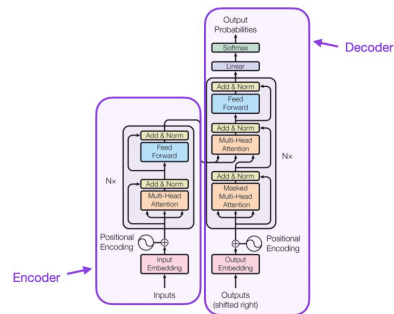


Figure 1: The Transformer - model architecture.

Agenda — Part 1

Part 1: Pre-Transformers & Attention: How the transformer architecture revolutionized NLP.

- Brief NLP Intro, Application, and Today's Focus
- “Prehistoric” Models: N-Gram based models
- RNNs (Recurrent Neural Networks) and LSTMs (Long Short Term Memories)
- Attention & Transformer

Agenda — Part 2

Part 2: Post Transformers: LLMs built on the transformer architecture; the past, present & future (state of the art)

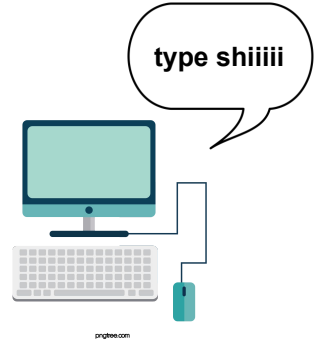
- Transformer-based encoder models (Bert-like)
 - BERT, RoBERTa
- Transformer-based decoder models
 - GPT Family, LLaMA, Vicuna
- Transformer-based encoder-decoder models
 - T5, BART



Part 1

Pre-Transformers & Attention: How the transformer architecture revolutionized NLP.

What is Natural Language Processing?



- We are currently communicating effortlessly in English.
- Not so “effortlessly” to replicate for computers.
 - How can computers “understand” the contents of natural language, including the contextual nuances of the language within them.
- NLP: ability to support and manipulate human language, enabling computers to **understand**, **interpret**, and **generate** human language in a valuable way.
 - Process large natural language datasets (ie text corpora or speech corpora) using ML approaches computer
- Example of NLP application: summarization, sentiment analysis, text classification, named entity recognition, **machine translation**

Language Modeling?

- Language modeling is the task of predicting what word comes next
 - The student opened their ____ (books | laptops | exams | minds |...)
- More formally: given a sequence of words x_1, x_2, \dots, x_t , compute the **probability distribution** of the next word x_{t+1}
- A system that does it is called a language model (LM)
 - A probabilistic multi-class classifier: $P(x_{t+1} | x_1, x_2, \dots, x_t)$
- Usages includes generating a translation (if the model is conditional on the input text)

Pre-historic Models: Statistical Language Modelings ~ N-Gram

Idea: Determine the probability of a sequence of words in a given language. The model is based on the assumption that the **probability of the next word** in a sequence **depends only on a fixed size window of the previous words**.

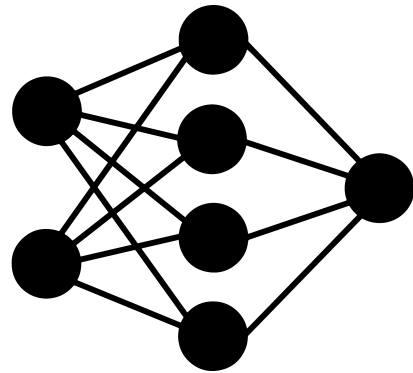
$$P(\mathbf{x}^{(t+1)} | \mathbf{x}^{(t)}, \dots, \mathbf{x}^{(1)}) = P(\mathbf{x}^{(t+1)} | \overbrace{\mathbf{x}^{(t)}, \dots, \mathbf{x}^{(t-n+2)}}^{n-1 \text{ words}}) \quad (\text{assumption})$$

$$\begin{array}{l} \text{prob of a } n\text{-gram} \rightarrow \\ \text{prob of a } (n-1)\text{-gram} \rightarrow \end{array} \begin{array}{c} P(\mathbf{x}^{(t+1)}, \mathbf{x}^{(t)}, \dots, \mathbf{x}^{(t-n+2)}) \\ = \\ P(\mathbf{x}^{(t)}, \dots, \mathbf{x}^{(t-n+2)}) \end{array} \quad \left| \begin{array}{l} \text{(definition of} \\ \text{conditional prob)} \end{array} \right.$$

- **Question:** How do we get these n -gram and $(n-1)$ -gram probabilities?
- **Answer:** By **counting** them in some large corpus of text!

$$\approx \frac{\text{count}(\mathbf{x}^{(t+1)}, \mathbf{x}^{(t)}, \dots, \mathbf{x}^{(t-n+2)})}{\text{count}(\mathbf{x}^{(t)}, \dots, \mathbf{x}^{(t-n+2)})} \quad (\text{statistical approximation})$$

Neural Language Models



Neural Net-based language > n-gram language models:

- neural language models don't need **smoothing**
- they can handle **much longer history** (longer sequence)
- they can **generalize** over contexts of similar words
 - word embeddings / distributed representations

(+) a neural language model has much **higher predictive accuracy** than an n-gram language model!

(-) neural net language models are strikingly **slower** to train than traditional language models

Machine Translation

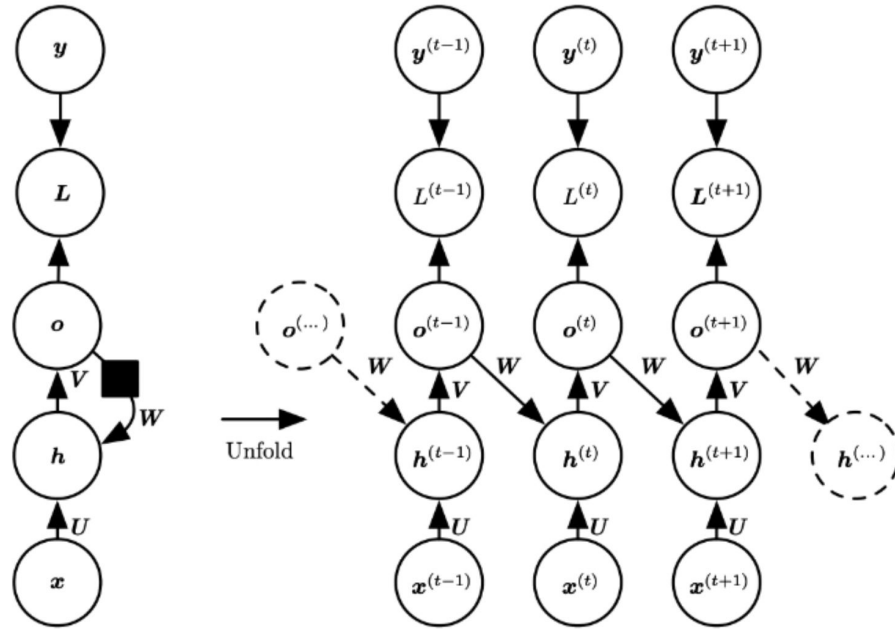


What does one need for this “NLP Magic”:

- Understand individual words
- Understand interaction between words (syntax)
- Translate the words (given the context)
- Compose a meaningful and fluent text

How does this list affect this
**“transformer
architecture/attention
mechanism”** ?

Recurrent Neural Network (RNN)



Feed-forward neural
networks

Rolled out over time



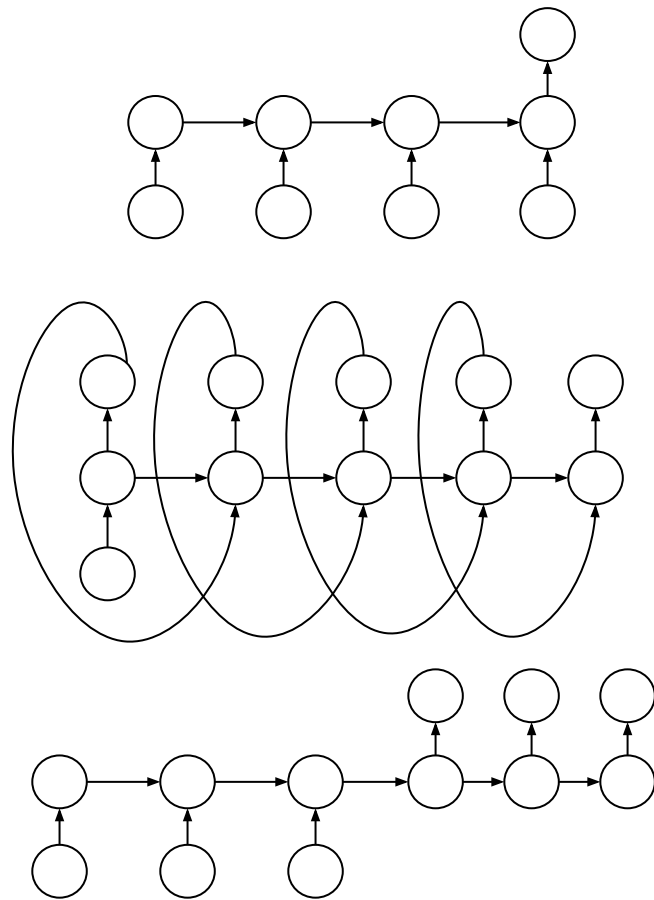
Sequence
Data!

Types of RNNs and Its Downfall

- Vector to Sequence RNNs
 - Image description
- Sequence to Vector RNNs
 - Movie review classification
- Sequence to Sequence
 - Machine translation

Disadvantages:

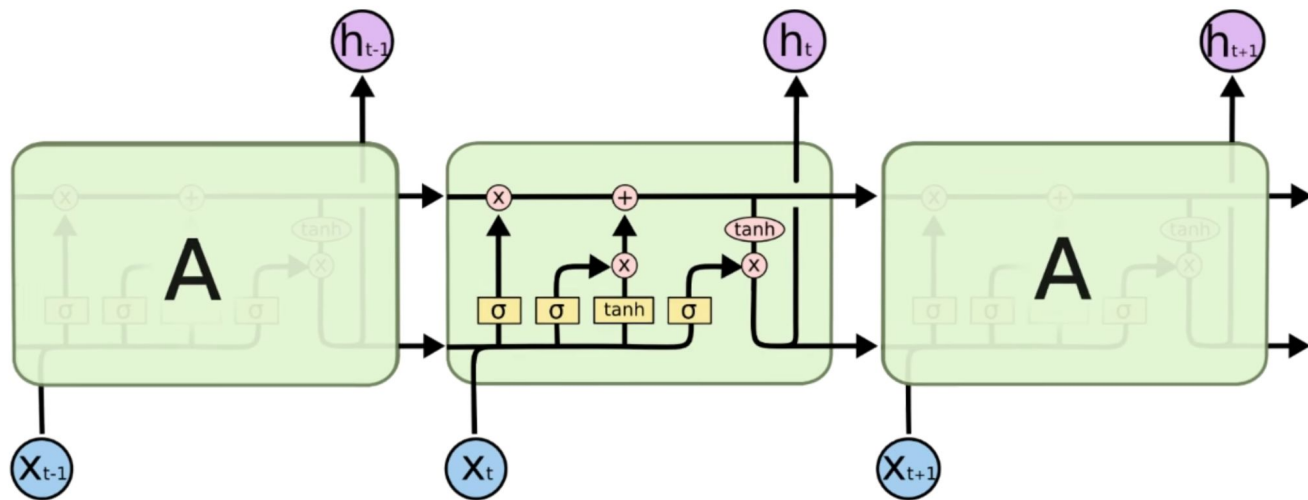
- Very **slow** to train
- **Vanishing gradients** for long sequences
- **Short term** memory



What about Long Short Term Memories (LSTM)?

Longer memory & reference window

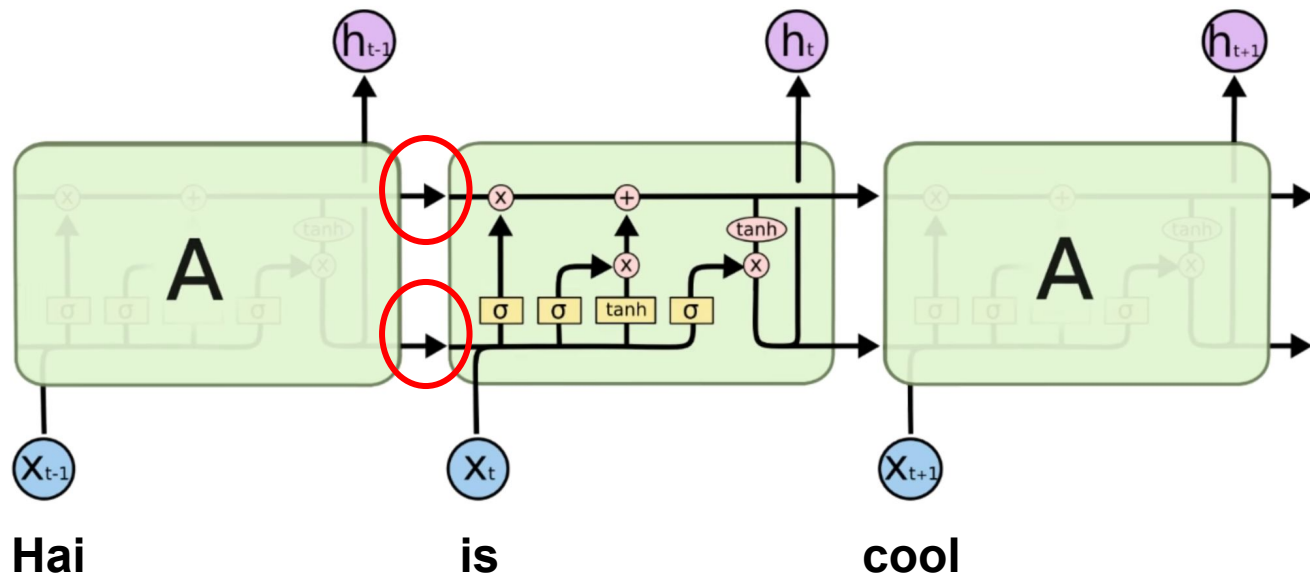
Longer sequences!



RNNs are slow, but LSTM are even **SLOWER**

What about Long Short Term Memories (LSTM)?

Not the best at capture true meaning/context of text



Everything is passed **sequentially**, and with **time steps**

Transformers — Problems Addressed



What does one need for this “NLP Magic”:

- Understand individual words
- Understand interaction between words (syntax)
- Translate the words (given the context)
- Compose a meaningful and fluent text

Subword vocabulary & embeddings

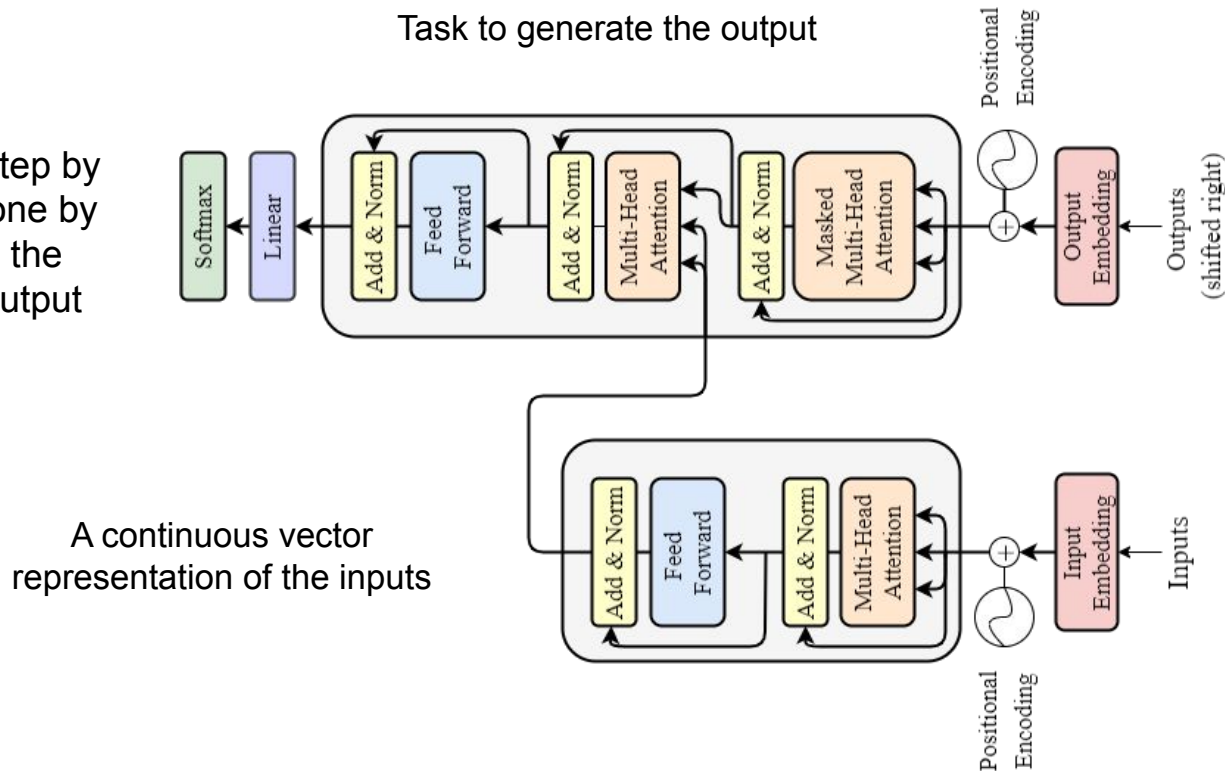
Encoder self attention

Cross attention

Encoder self attention

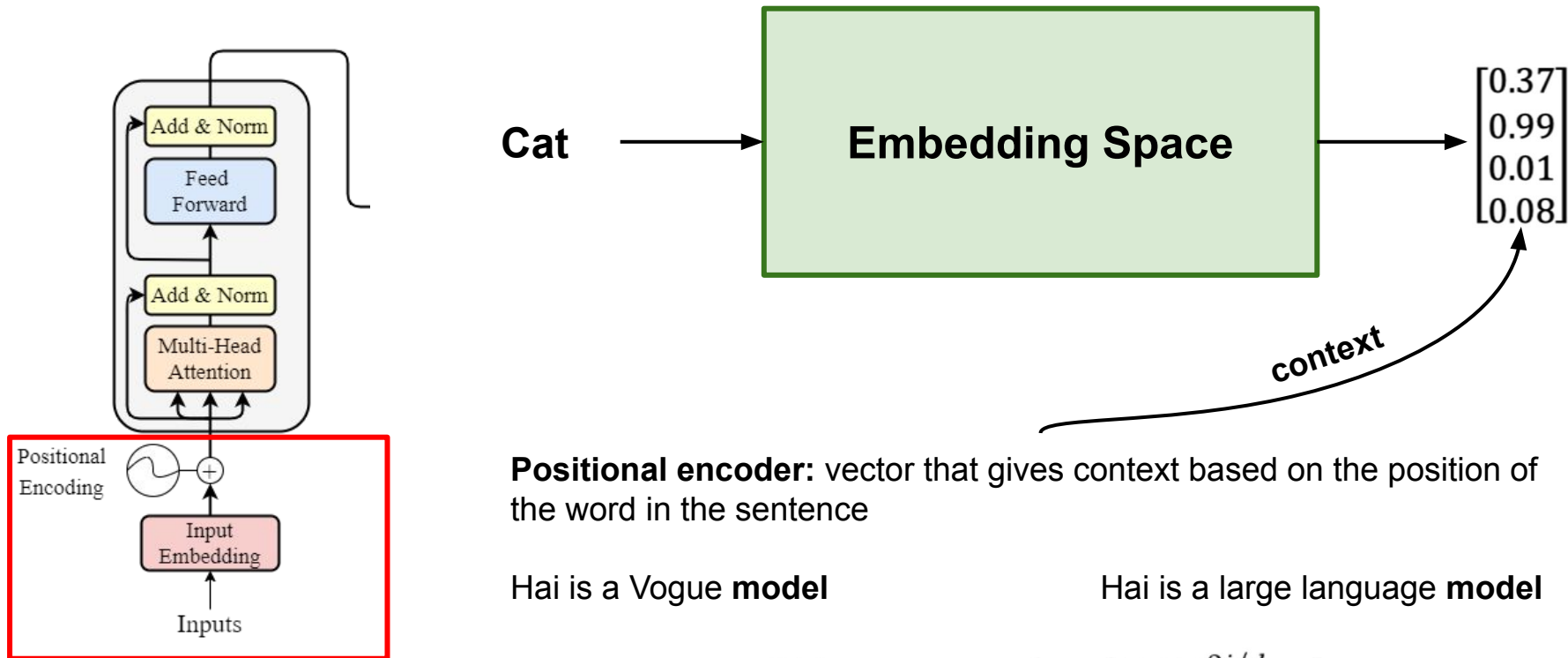
Transformers Architecture: An Overview

“Auto-regressive”: Step by step generate output one by one, while being fed the previous generated output



Vaswani et al, 2017. Attention is all you need.

Transformers Architecture: Encoder Overview



Positional encoder: vector that gives context based on the position of the word in the sentence

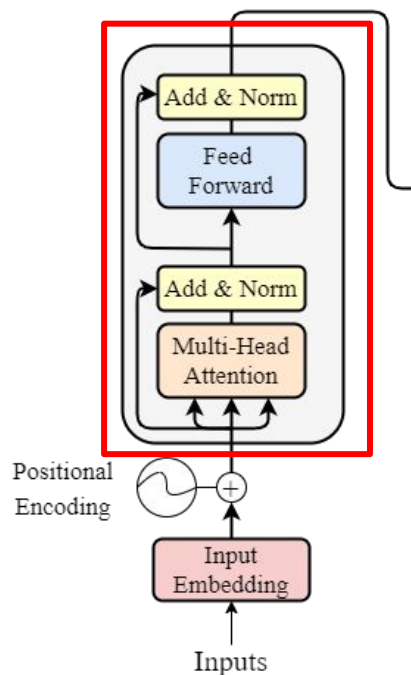
Hai is a Vogue **model**

Hai is a large language **model**

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

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

Transformers Architecture: Encoder Overview

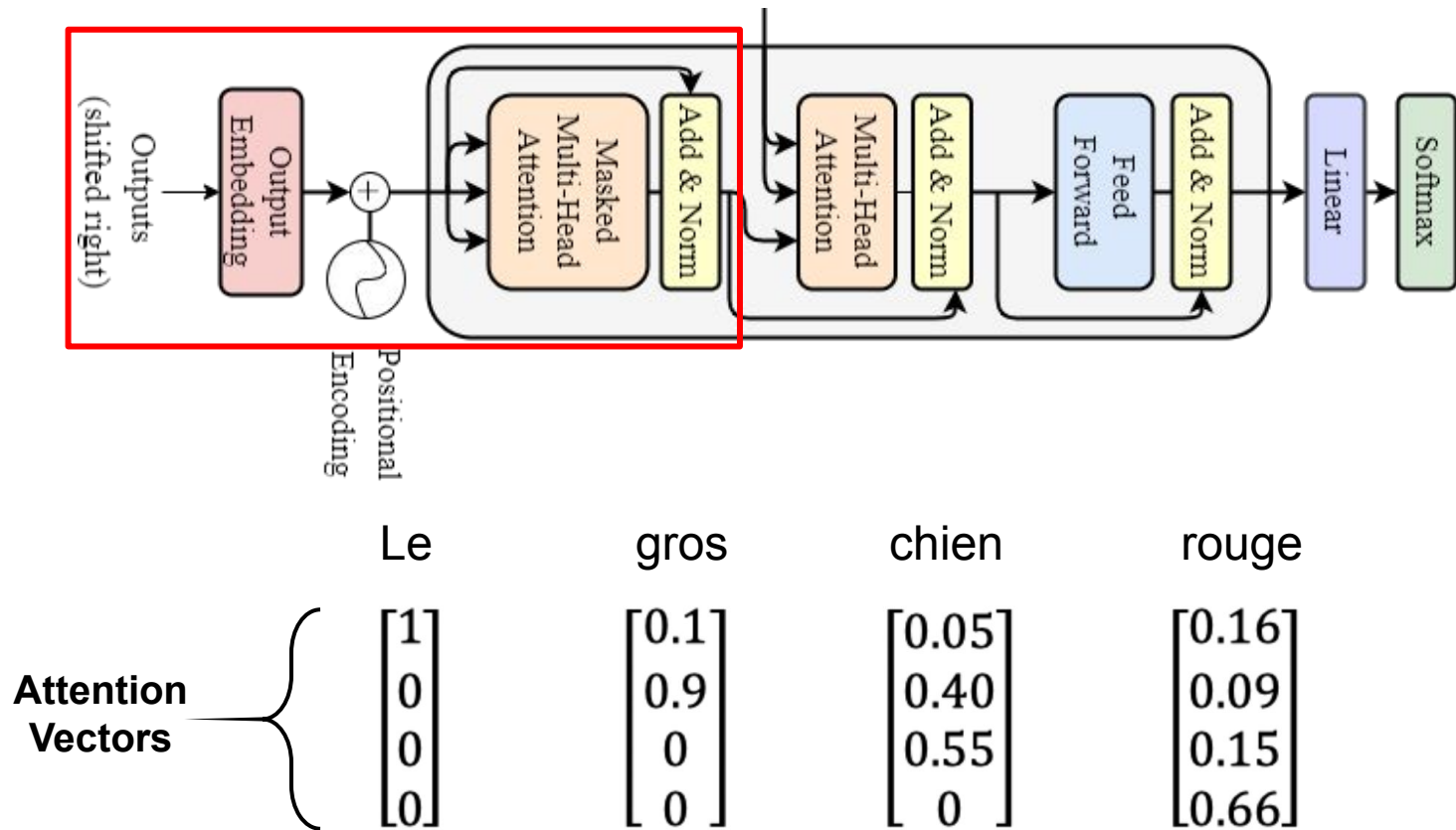


Attention: How is the i^{th} word relevant to the rest of the words in this sentence?

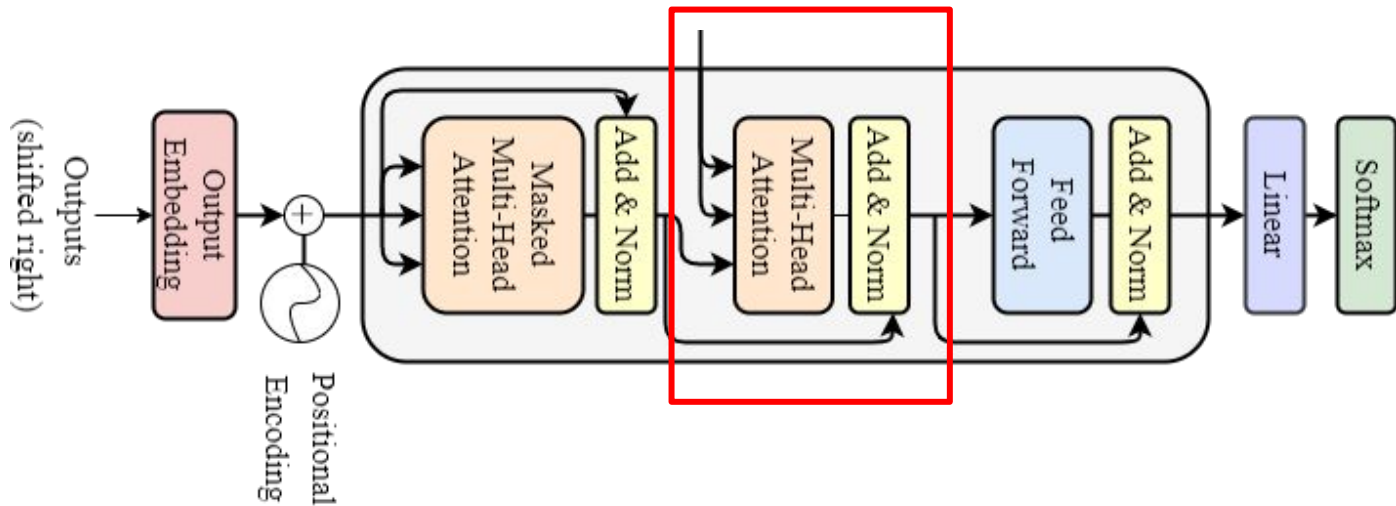
The	$[0.71 \quad 0.04 \quad 0.07 \quad 0.18]^T$
big	$[0.01 \quad 0.84 \quad 0.02 \quad 0.13]^T$
red	$[0.09 \quad 0.05 \quad 0.62 \quad 0.24]^T$
dog	$[0.03 \quad 0.03 \quad 0.03 \quad 0.91]^T$

⏟
Attention Vectors

Transformers Architecture: Decoder Overview



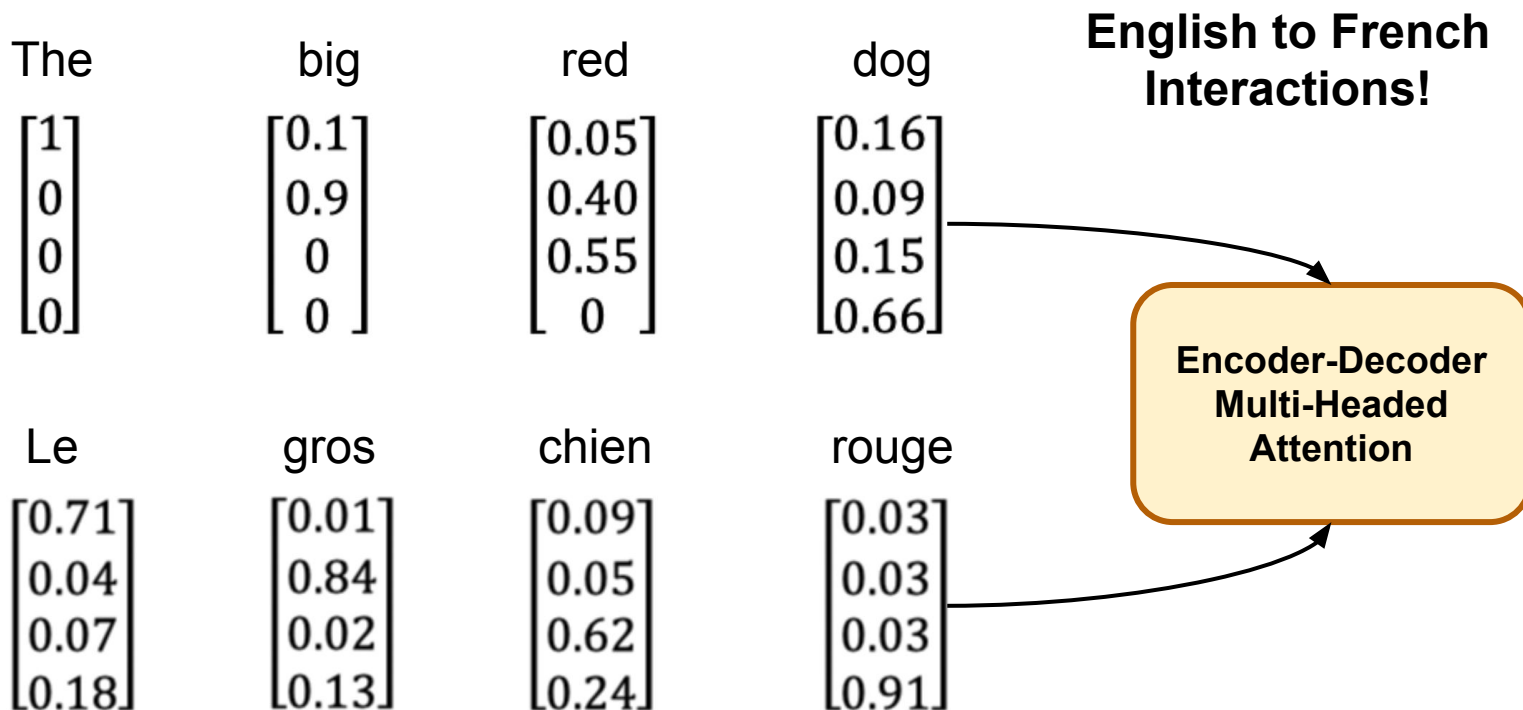
Transformers Architecture: Decoder Overview



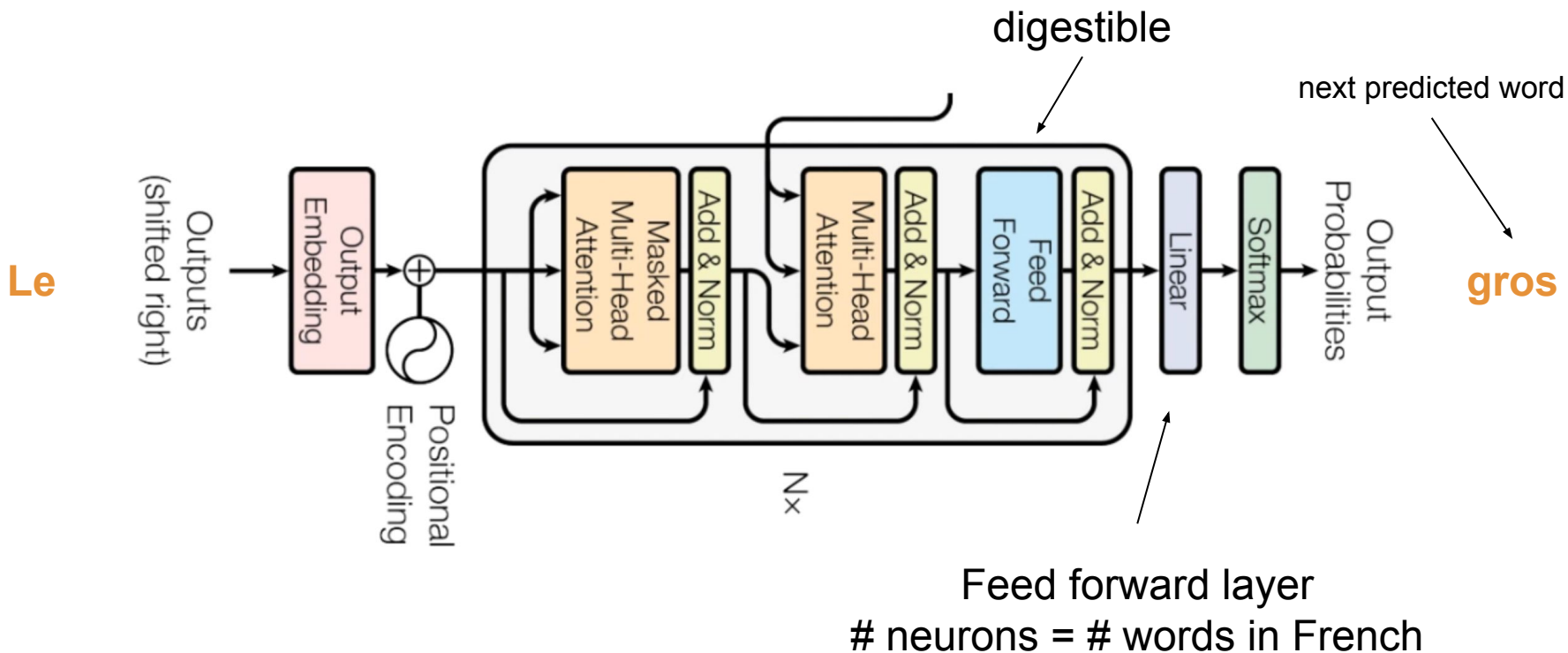
Attention vectors and the **encoder vectors** are fed into another Multi-Headed Attention block

- Encapsulates the English-French Interactions

Transformers Architecture: Decoder Overview

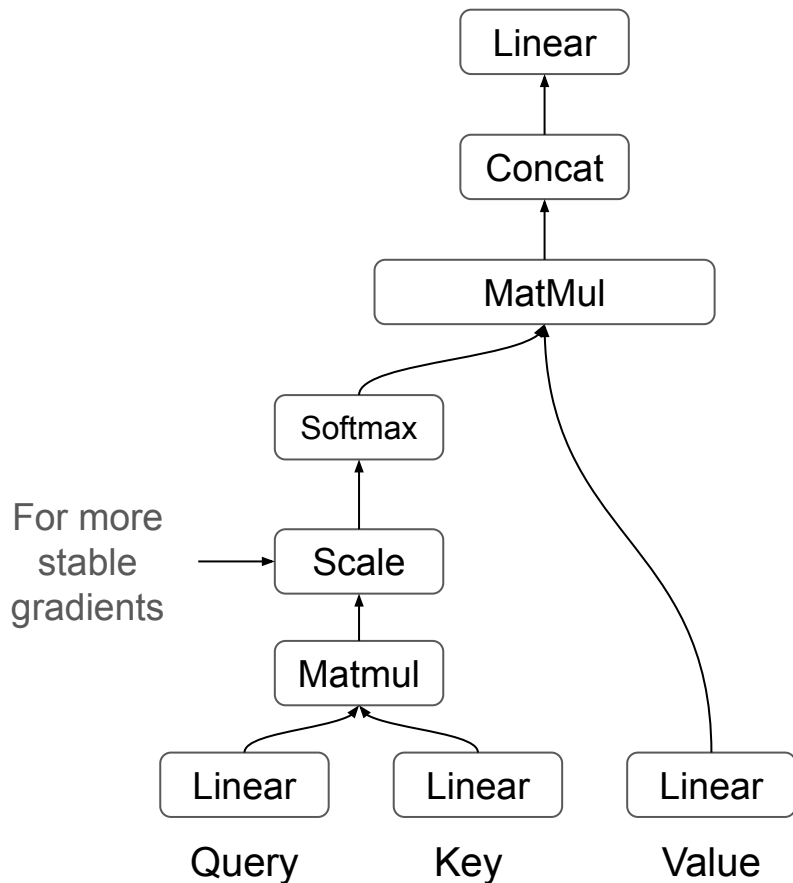


Transformers Architecture: Decoder Overview



Transformers Architecture: Encoder Deeper View

Q, K, V created from feeding input text into a fully connected layer. Q, K, V are split into n vectors
N attention heads = Multi-head attention block



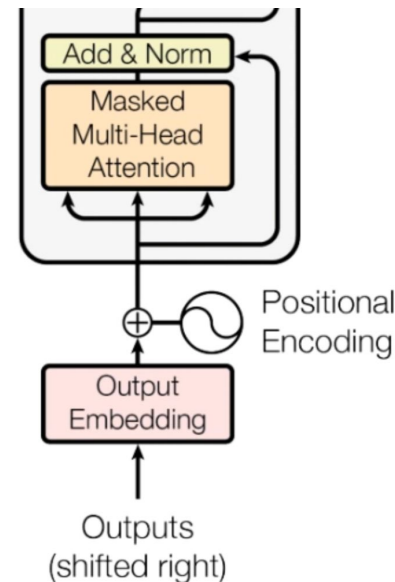
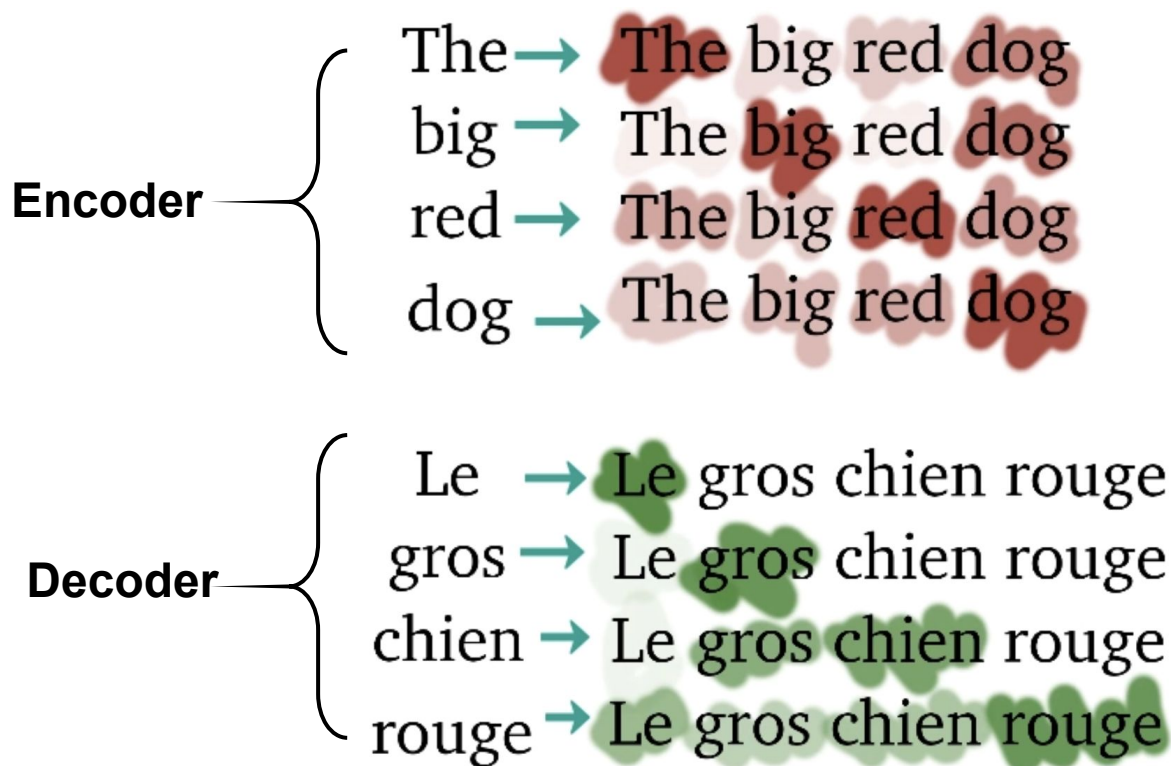
Attention weights

	the	big	red	dog
the	0.71	0.04	0.07	0.18
big	0.01	0.84	0.02	0.13
red	0.09	0.05	0.62	0.24
dog	0.03	0.03	0.03	0.91

Outputs = Attention weights x Value

Transformers Architecture: Decoder Deeper View

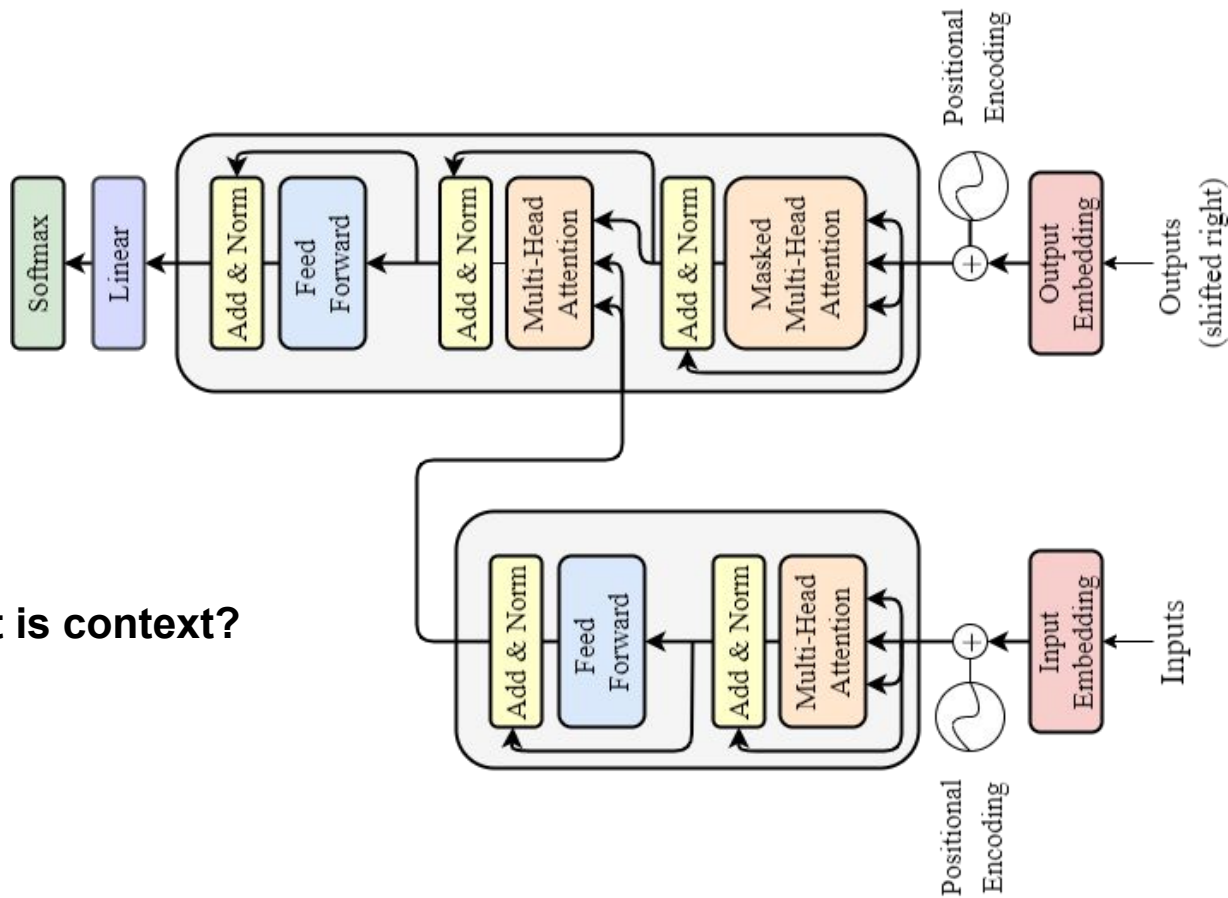
Masked Multi-Headed Attention?



Transformer Encoder & Decoder Summarized

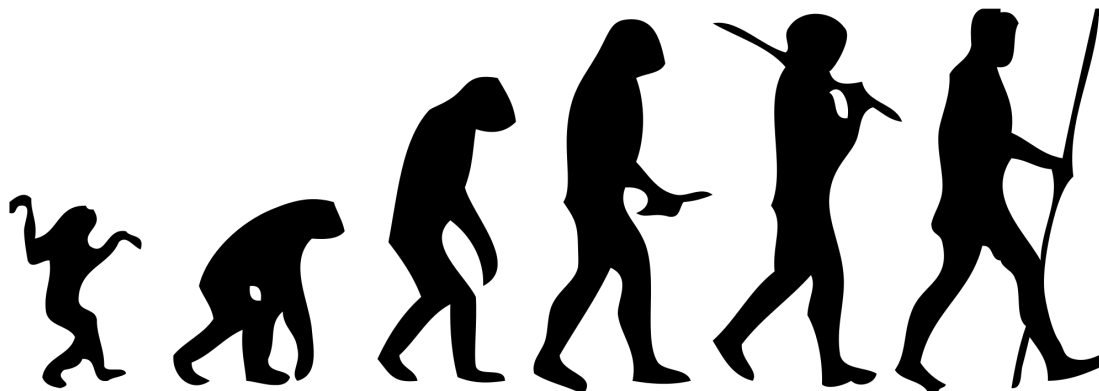
How to map
English words to
French words?

What is English? What is context?

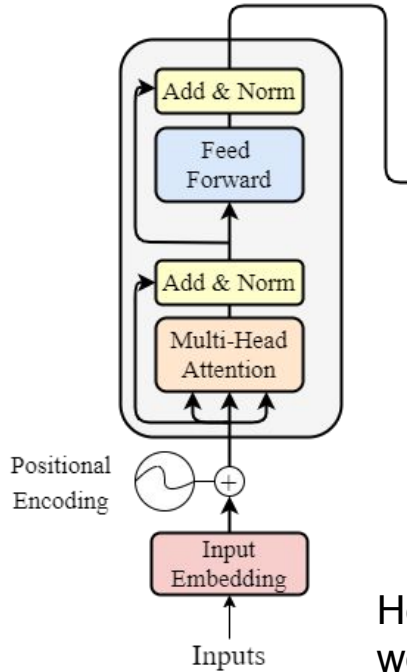


Part 2

Post Transformers: The Past, Present & Future of Transformer-based LLMs

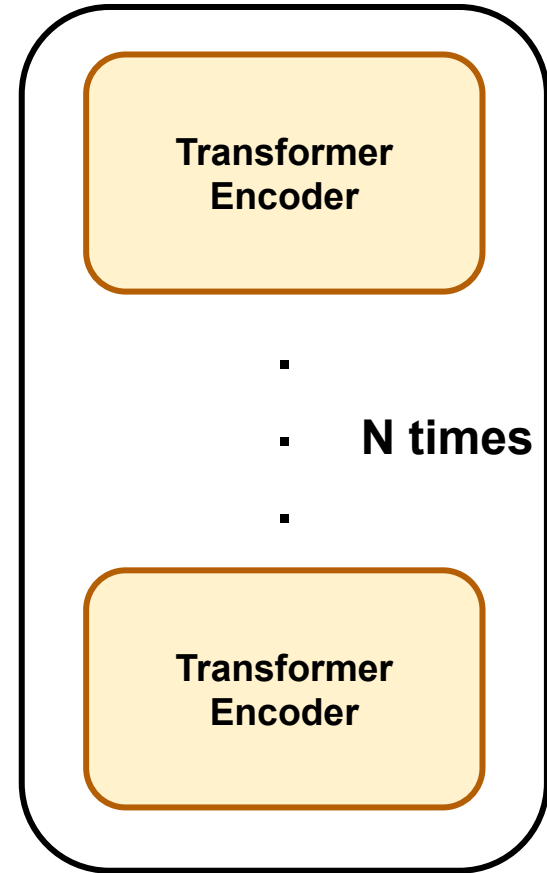


Stacking Encoders?

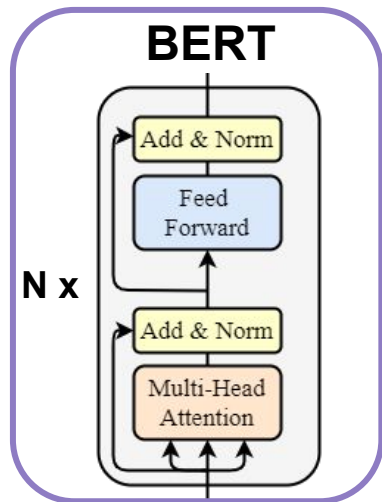


encodes the input
text to **some
representation
with attention
information**

Helps decoder to focus on a certain
word during the decoding process



Bidirectional Encoder Representation of Transformer (BERT)



Problems to solve:

- Question Answering
- Sentiment Analysis
- Text Summarization
- Many more tasks

Language understanding

Solution:

- **Pretraining** this stacked transformer encoders architecture (BERT) to understand language
- **Finetune** BERT to learn a specific task

BERT Pretraining Overview

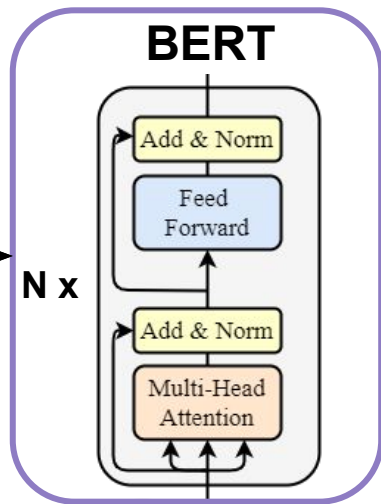
Goal: “What is language?” “What is context?”

Masked Language Modeling (MLM)

The [MASK1]
brown fox
[MASK2] over the
lazy dog.

Next Sentence Prediction (NSP)

A: Hai has a good
tate in fashio
B: His fits are
always dusty.

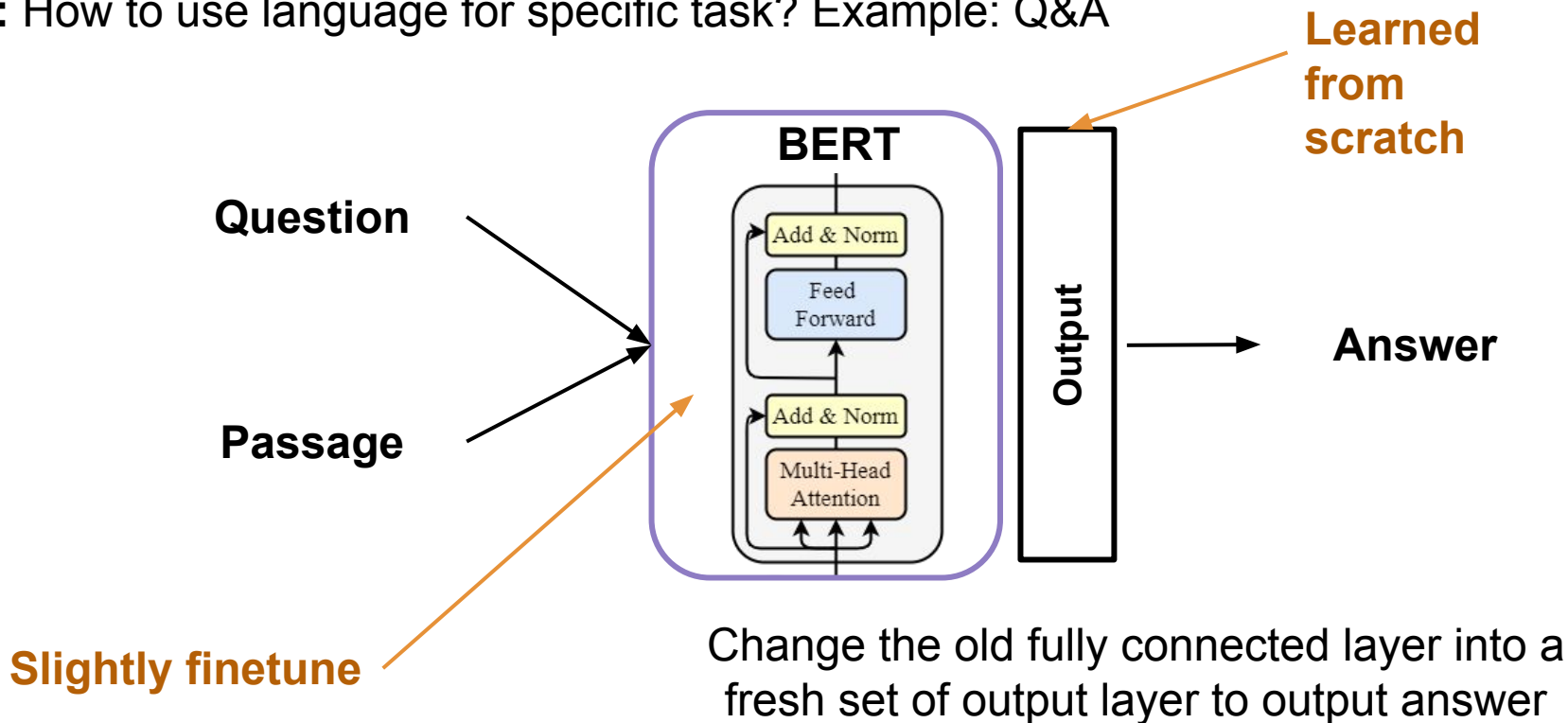


[MASK1] = quick
[MASK2] = jumped

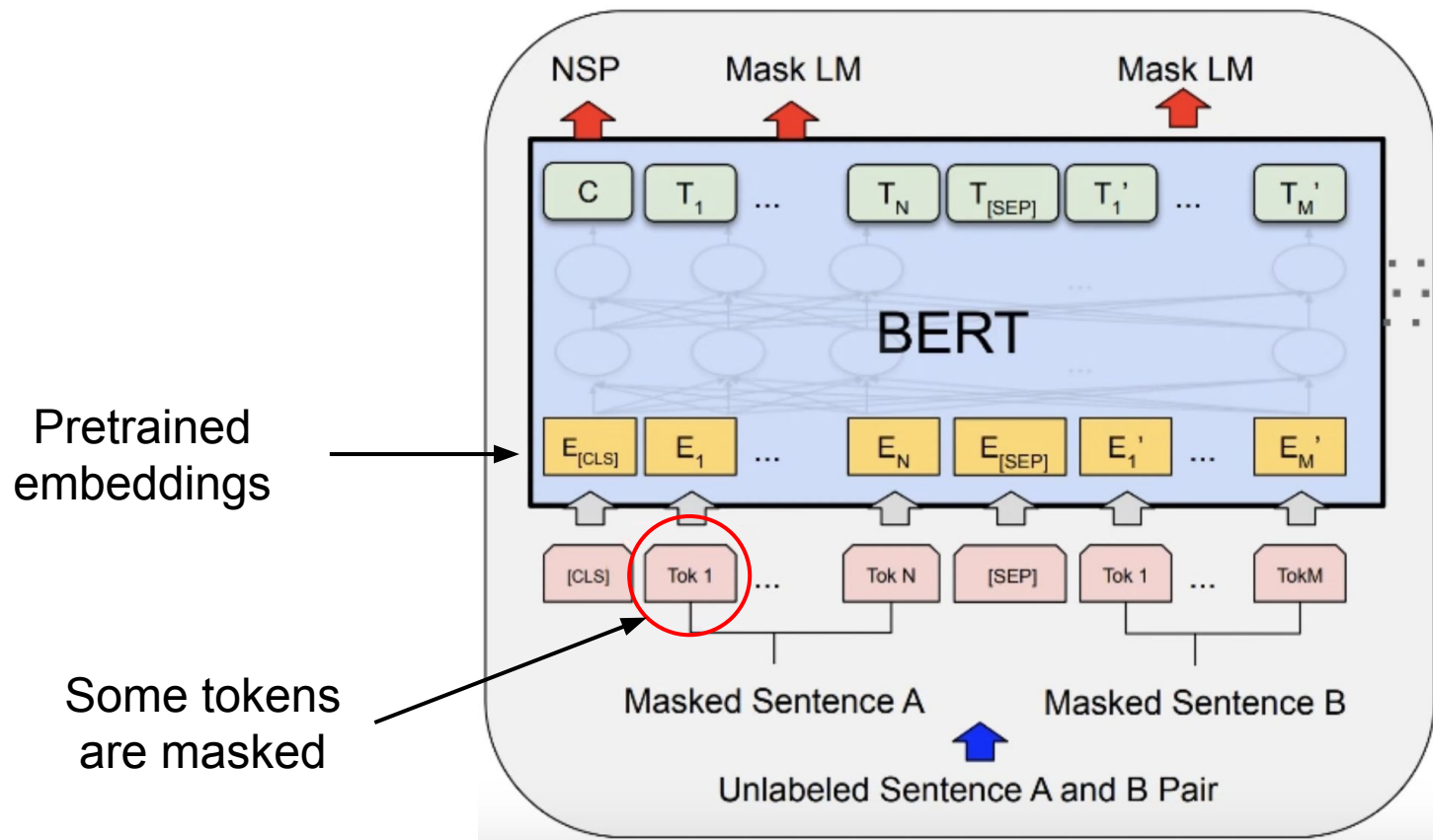
**No, Sentence B
does not follow
sentence A.**

BERT Finetuning Overview

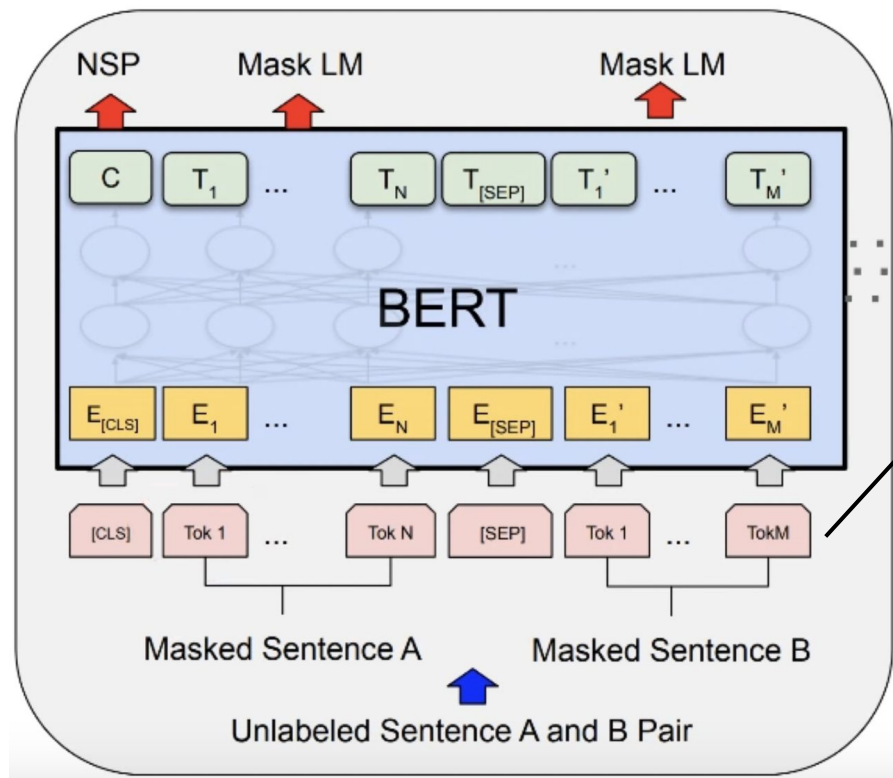
Goal: How to use language for specific task? Example: Q&A



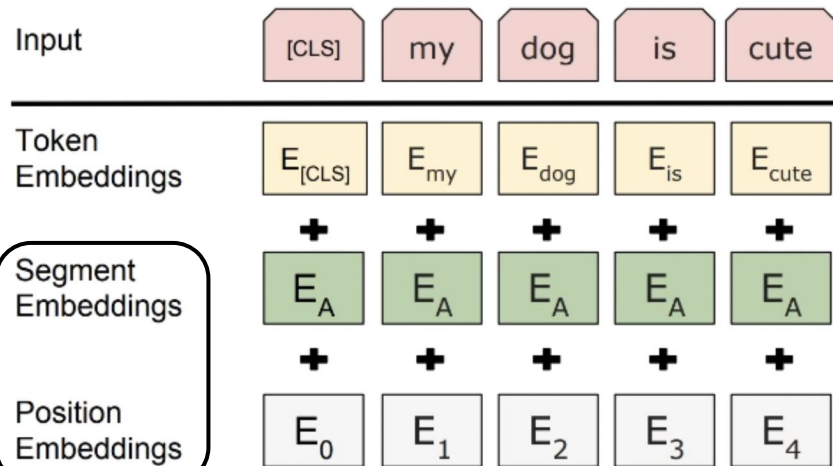
BERT Pretraining Deeper View



BERT Pretraining Deeper View



CrossEntropy(Softmax(Output word vector)**,** Actual Word**)**

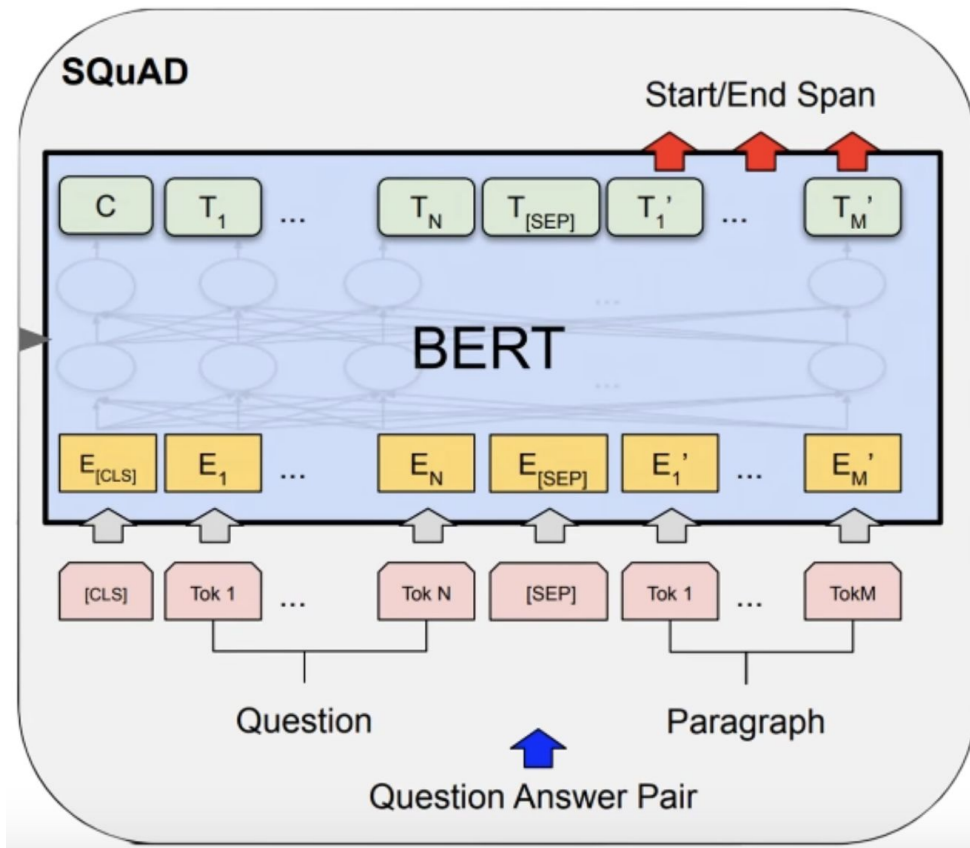


Account for ordering of the inputs

BERT Finetuning Deeper View

Change output to display the answer's text

Change input to take in question & passage



BERT's Performance

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	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
BERT _{BASE}	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
BERT _{LARGE}	86.7/85.9	72.1	92.7	94.9	60.5	86.5	89.3	70.1	82.1

MultiNLI

Premise: Hills and mountains are especially sanctified in Jainism.

Hypothesis: Jainism hates nature.

Label: Contradiction

CoLA

Sentence: The wagon rumbled down the road





Label: Acceptable

Sentence: The car honked down the road

Label: Unacceptable

Why is BERT so good?

- . Very **large** model & lots of compute!
- . Moderately large corpora (3.2B words),
- . **Deep bidirectional contextualized** word representation
(seeing all words when encoding each word in each layer)

ULMfit	GPT	BERT	GPT-2
Jan 2018	June 2018	Oct 2018	Feb 2019
Training:	Training	Training	Training
1 GPU day	240 GPU days	256 TPU days ~320–560 GPU days	~2048 TPU v3 days according to a reddit thread
			

BERT's Relative!

- mBERT
- **RoBERTa**
- XLM-RoBERTa (XLM-R)
- Electra
- DeBERTa
- and many-many others!



BERT→RoBERTa

RoBERTa: Robustly optimized **BERT** approach.

After BERT, there have been many improvements on pretrained transformer encoder.

However, RoBERTa was the **biggest** leap

Subword embeddings



- **Dynamic masking**
- Change of pretrained word embeddings: **30k WordPiece** → **50k byte-level BPE vocabulary**
- **Removal** of NSP (next sentence prediction)
- Careful selection of optimization hyperparameters for pretraining and finetuning
- Larger dataset & larger batch size

RoBERTa Masking

- **BERT**: data was **duplicate 10 times** and perturbed (masked, etc.) before training for 40 epochs
- **RoBERTa**: randomly perturbs inputs during training.
 - No need to duplicate large datasets
 - More diverse data when training more epochs
 - Similar or better results

Masking	SQuAD 2.0	MNLI-m	SST-2
reference	76.3	84.3	92.8
<i>Our reimplementation:</i>			
static	78.3	84.3	92.5
dynamic	78.7	84.0	92.9

Table 1: Comparison between static and dynamic masking for BERT_{BASE}. We report F1 for SQuAD and accuracy for MNLI-m and SST-2. Reported results are medians over 5 random initializations (seeds). Reference results are from [Yang et al. \(2019\)](#).

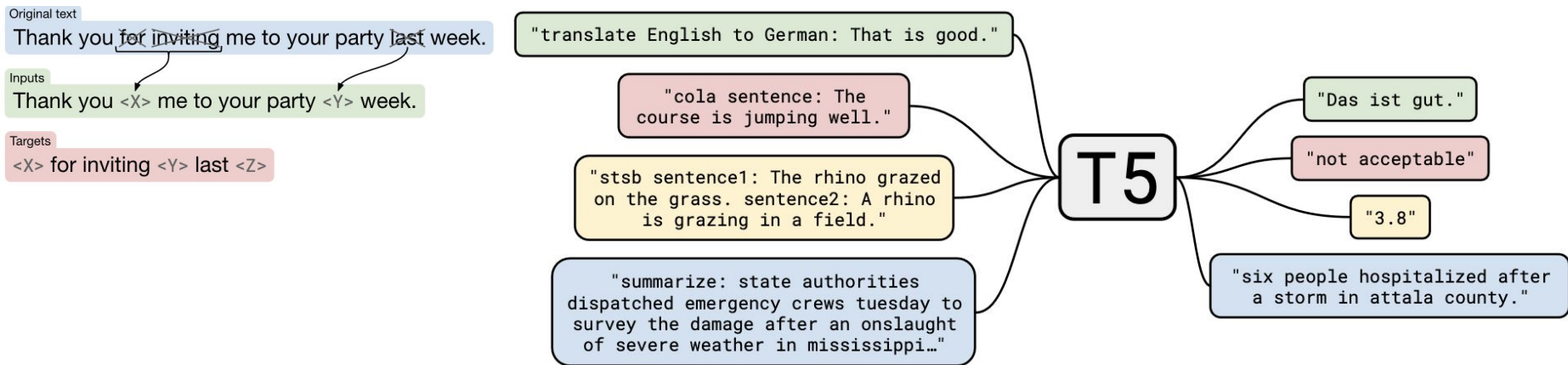
RoBERTa Performance

	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	WNLI	Avg
<i>Single-task single models on dev</i>										
BERT _{LARGE}	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-	-
XLNet _{LARGE}	89.8/-	93.9	91.8	83.8	95.6	89.2	63.6	91.8	-	-
RoBERTa	90.2/90.2	94.7	92.2	86.6	96.4	90.9	68.0	92.4	91.3	-
<i>Ensembles on test (from leaderboard as of July 25, 2019)</i>										
ALICE	88.2/87.9	95.7	90.7	83.5	95.2	92.6	68.6	91.1	80.8	86.3
MT-DNN	87.9/87.4	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0	87.6
XLNet	90.2/89.8	98.6	90.3	86.3	96.8	93.0	67.8	91.6	90.4	88.4
RoBERTa	90.8/90.2	98.9	90.2	88.2	96.7	92.3	67.8	92.2	89.0	88.5

Table 5: Results on GLUE. All results are based on a 24-layer architecture. BERT_{LARGE} and XLNet_{LARGE} results are from [Devlin et al. \(2019\)](#) and [Yang et al. \(2019\)](#), respectively. RoBERTa results on the development set are a median over five runs. RoBERTa results on the test set are ensembles of *single-task* models. For RTE, STS and MRPC we finetune starting from the MNLI model instead of the baseline pretrained model. Averages are obtained from the GLUE leaderboard.

Transformer Encoder-Decoder LLM - T5

Text-to-Text Transfer Transformer (**T5**), trained on **C4** (large common crawl dataset); MLM style

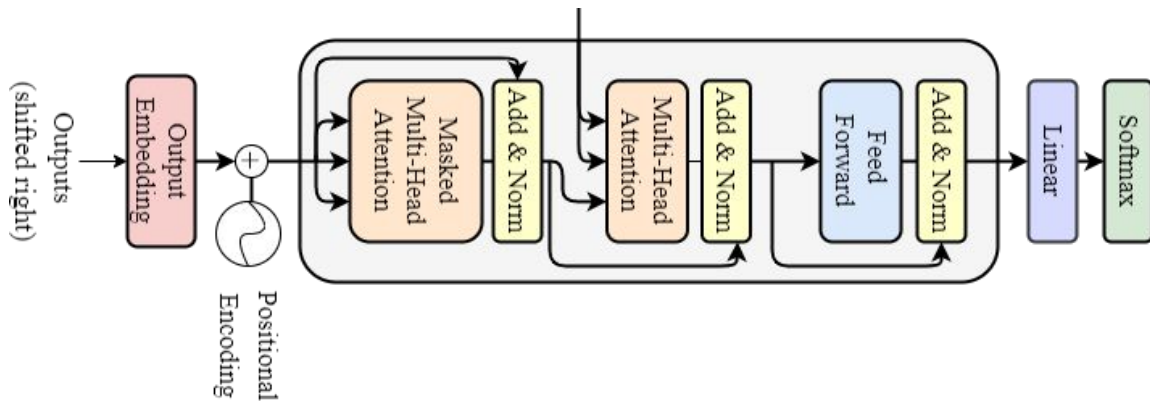


T5 Performance

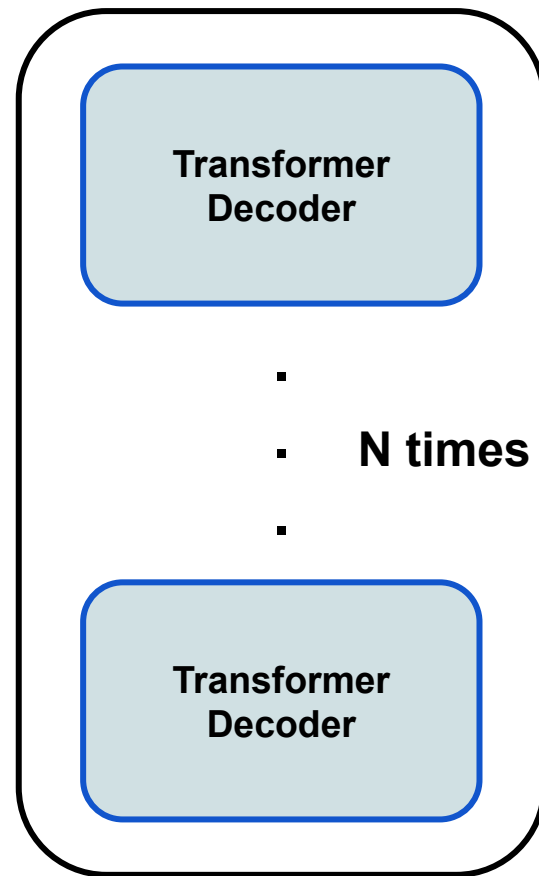
Model	GLUE Average	CoLA Matthew's	SST-2 Accuracy	MRPC F1	MRPC Accuracy	STS-B Pearson	STS-B Spearman
Previous best	89.4 ^a	69.2 ^b	97.1 ^a	93.6^b	91.5^b	92.7 ^b	92.3 ^b
T5-Small	77.4	41.0	91.8	89.7	86.6	85.6	85.0
T5-Base	82.7	51.1	95.2	90.7	87.5	89.4	88.6
T5-Large	86.4	61.2	96.3	92.4	89.9	89.9	89.2
T5-3B	88.5	67.1	97.4	92.5	90.0	90.6	89.8
T5-11B	90.3	71.6	97.5	92.8	90.4	93.1	92.8

Model	SQuAD EM	SQuAD F1	SuperGLUE Average	BoolQ Accuracy	CB F1	CB Accuracy	COPA Accuracy
Previous best	90.1 ^a	95.5 ^a	84.6 ^d	87.1 ^d	90.5 ^d	95.2 ^d	90.6 ^d
T5-Small	79.10	87.24	63.3	76.4	56.9	81.6	46.0
T5-Base	85.44	92.08	76.2	81.4	86.2	94.0	71.2
T5-Large	86.66	93.79	82.3	85.4	91.6	94.8	83.4
T5-3B	88.53	94.95	86.4	89.9	90.3	94.4	92.0
T5-11B	91.26	96.22	88.9	91.2	93.9	96.8	94.8

Stacking Decoders?

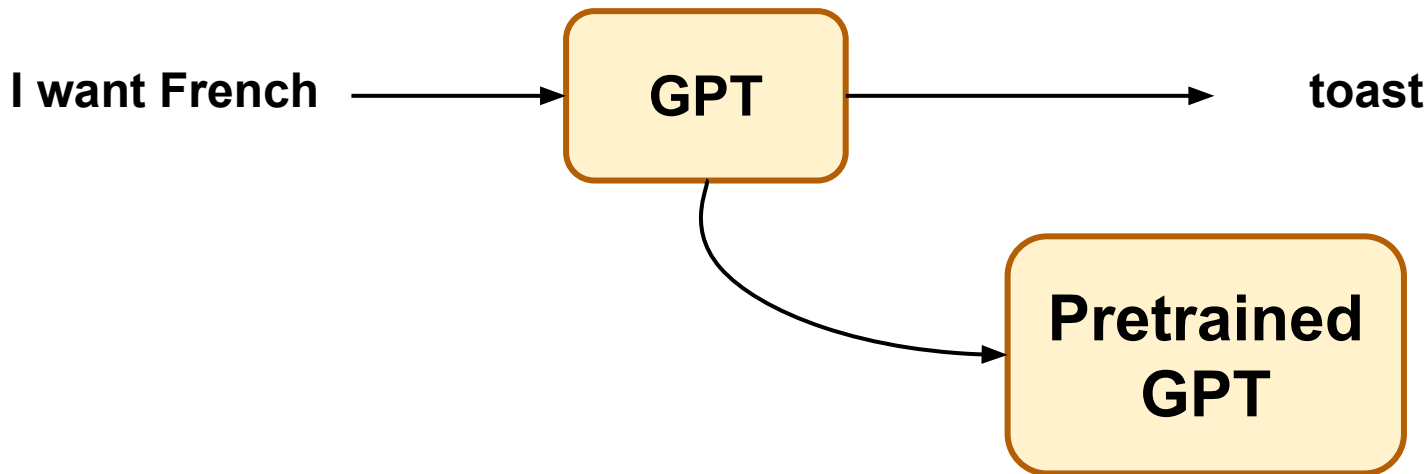


Autoregressively generate the next word in the sequence; taking in the previous generate words from the decoder and the embeddings from the encoder



Generative Pretrained Transformer (GPT)

- GPT: Pretrained on general language data to answer “what is language?”
- Feed input sequences and predict their next token




Chat Generative Pretrained Transformer (ChatGPT)

Using the Pretrained GPT as a language model, we **finetune**:

- ChatGPT: Firstly, the model underwent **supervised fine-tuning**: human trainers' dialogues to emulate human vs. AI
 - Conversational use cases, hence the Chat in GPT
- Then, the model underwent **reinforcement learning**: assessing AI trainers' response in a dialogue
 - Process of achieving some goal using **rewards**

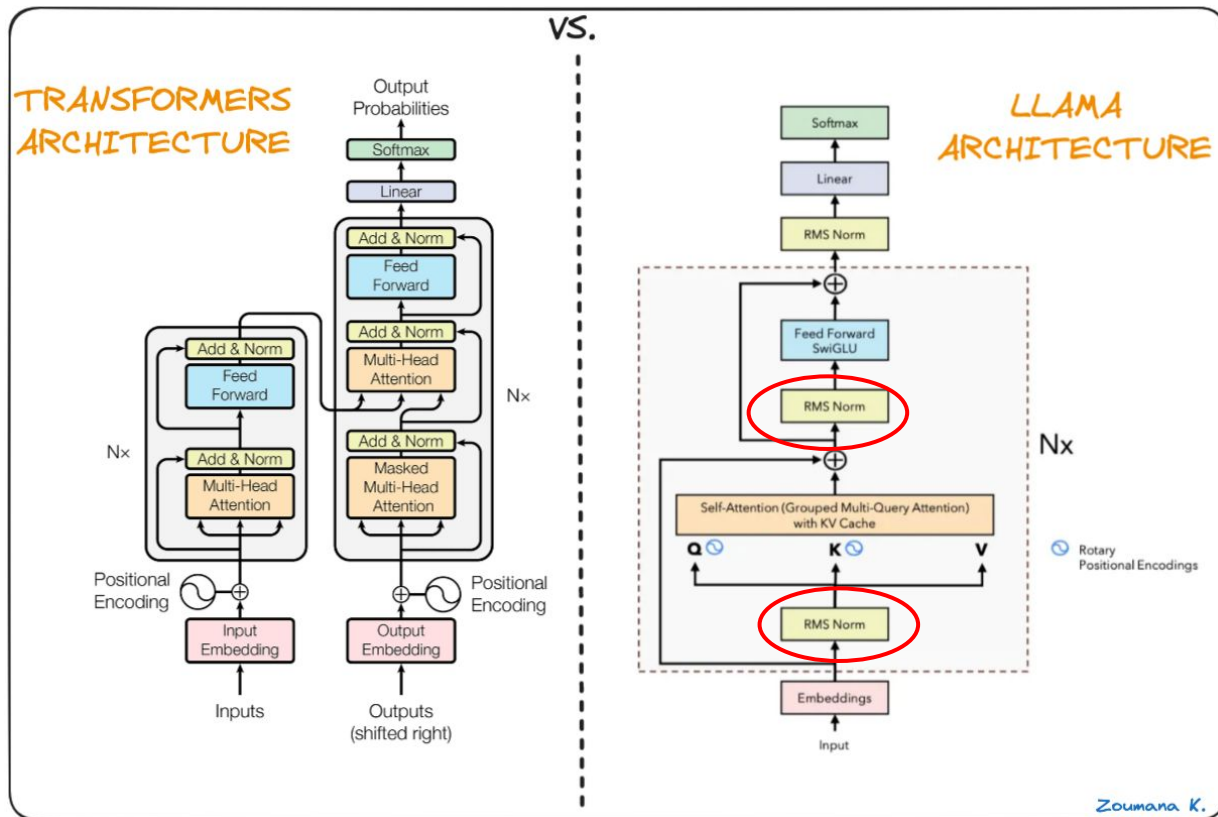
ChatGPT's Reinforcement Learning

	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	+10

- Agent
- Reward
- State
- Action
- Policy

GPT → LLaMA (Large Language Model Meta AI)

- **Decoder only** model, similar to the GPT family.
 - Stacked decoders



LlaMA > GPT?

- Open source.
- Public data.
- Limited GPU requirements for finetuning.
- **Smaller** architecture, **more** training data
 - 7 billions to 70 billions vs. 175 billions.
 - Comparable performance, sometimes even **better!**
- **Fast** inference!

Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

LlaMA Performance

		BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA
GPT-3	175B	60.5	81.0	-	78.9	70.2	68.8	51.4	57.6
Gopher	280B	79.3	81.8	50.6	79.2	70.1	-	-	-
Chinchilla	70B	83.7	81.8	51.3	80.8	74.9	-	-	-
PaLM	62B	84.8	80.5	-	79.7	77.0	75.2	52.5	50.4
PaLM-cont	62B	83.9	81.4	-	80.6	77.0	-	-	-
PaLM	540B	88.0	82.3	-	83.4	81.1	76.6	53.0	53.4
LLaMA	7B	76.5	79.8	48.9	76.1	70.1	72.8	47.6	57.2
	13B	78.1	80.1	50.4	79.2	73.0	74.8	52.7	56.4
	33B	83.1	82.3	50.4	82.8	76.0	80.0	57.8	58.6
	65B	85.3	82.8	52.3	84.2	77.0	78.9	56.0	60.2

Transformers → Large Language Modelings

To conclude:

- Transformer architecture; given enough resource, can look back and forward **without limitations: real contexts & meaning** of language.
 - Unlike RNNs, LSTMs, GRUs!
- **Incredibly fast**, due to the ability to **fully** utilize parallelization
- With transformer, we give rise to LLMs as we know it today (BERT, RoBERTa, ELECTRA, GPT, LLaMA, Mistral)
 - Building brick by brick, what's next?

Concluding Thoughts

Email: hai.le@skol.tech

Telegram: @hlet0