

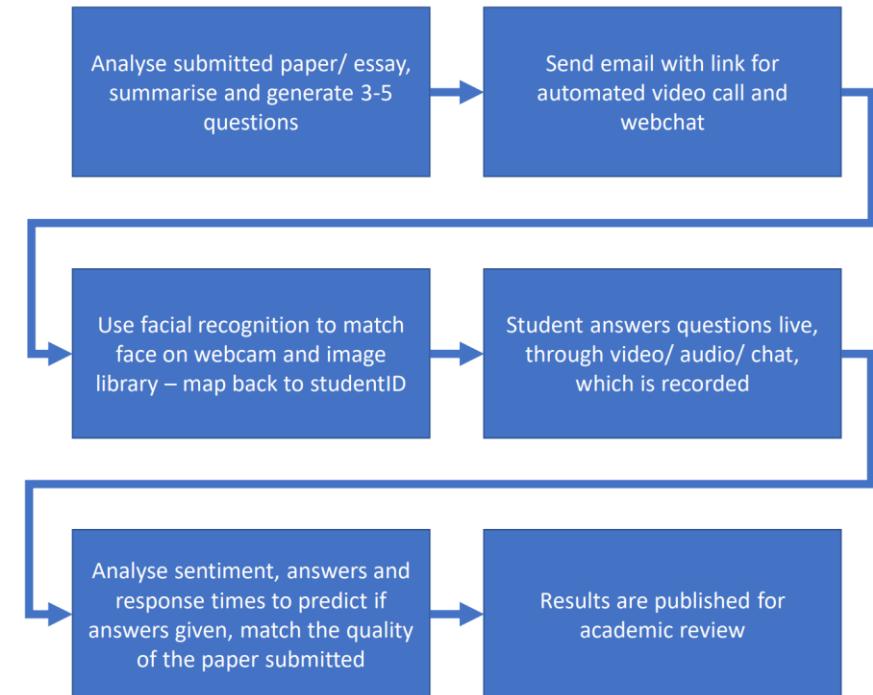


Automatic Exam Marking Project

Phase 1: Exam Management Product

Product ML Engine

Required Core Modules:
Bert & Transformers



Automatic Exam Marking Project

Phase 1: Exam Management Product

PART 1 : Language Understanding Models & Bert Overview

PART 2 : Transformer Architecture Overview &
Encoder Part Technical Detail

PART 3 : Bert Technical Detail & Fine Tuning Examples

PART 4 : New Task & Code Analysis Definition

PART 5 : References

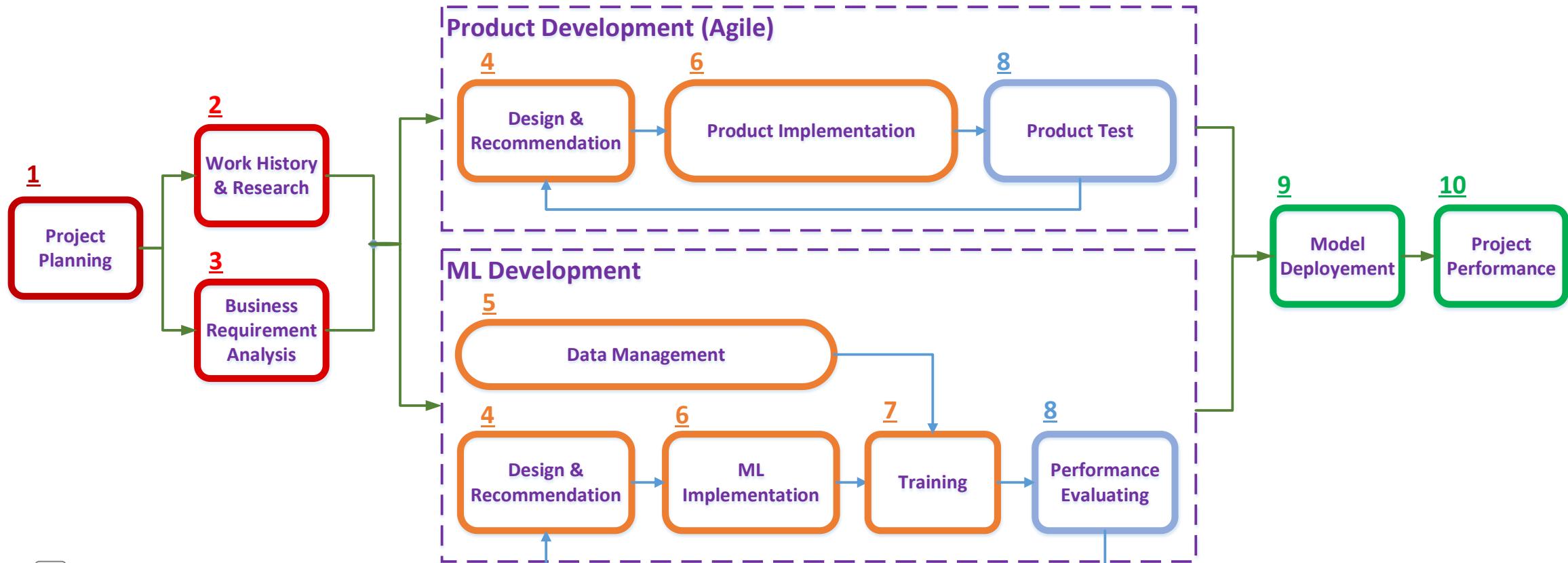
Project Current Status



MACQUARIE
University



Automatic Exam Marking - Project Phases





Automatic Exam Marking Project

Phase 1: Exam Management Product

Product ML Engine

**PART 1 : Language Understanding Models
&
Bert Overview**

NLP Tasks

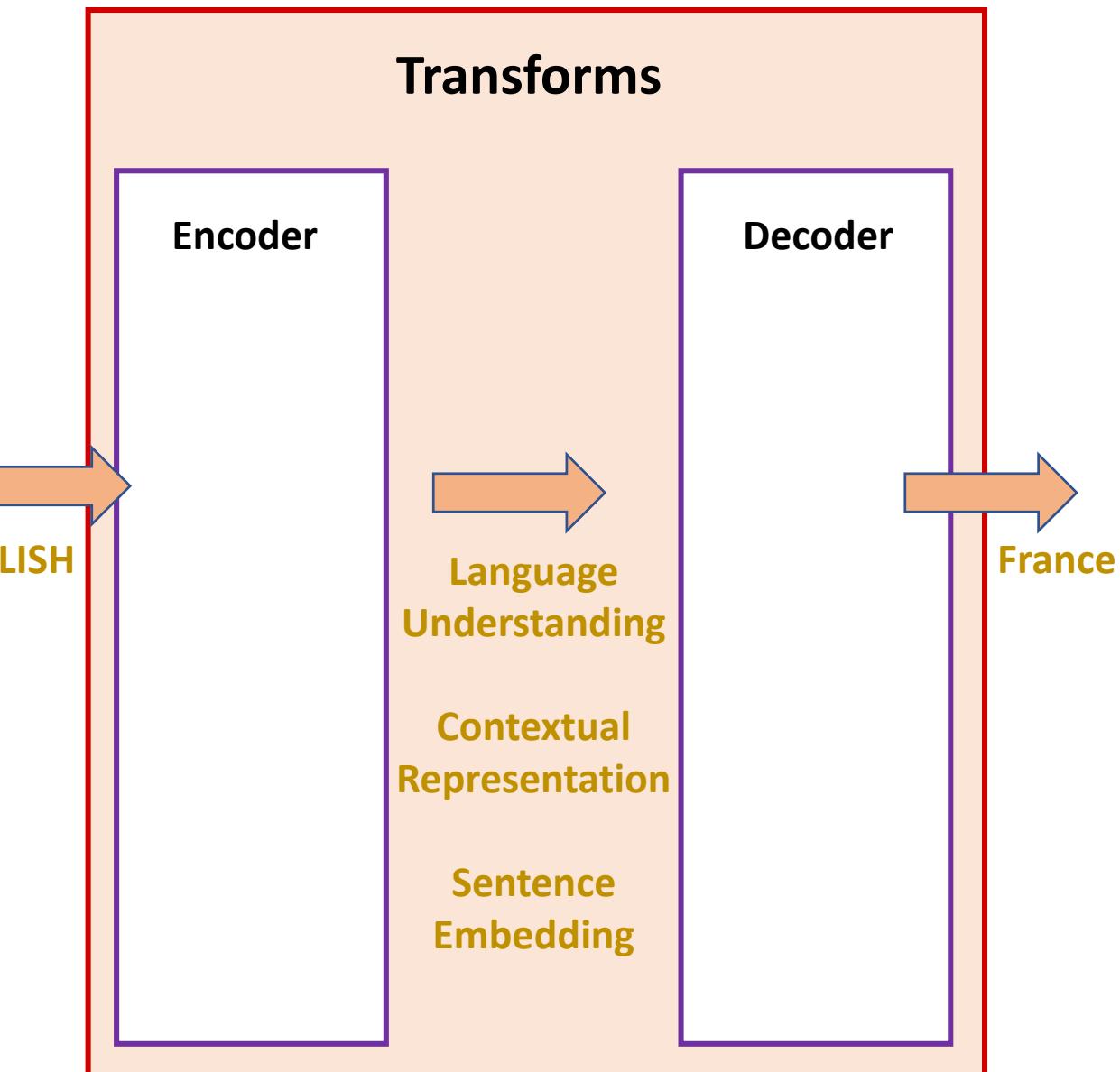
Pre-train Phase

Language Understanding

Model Tuning

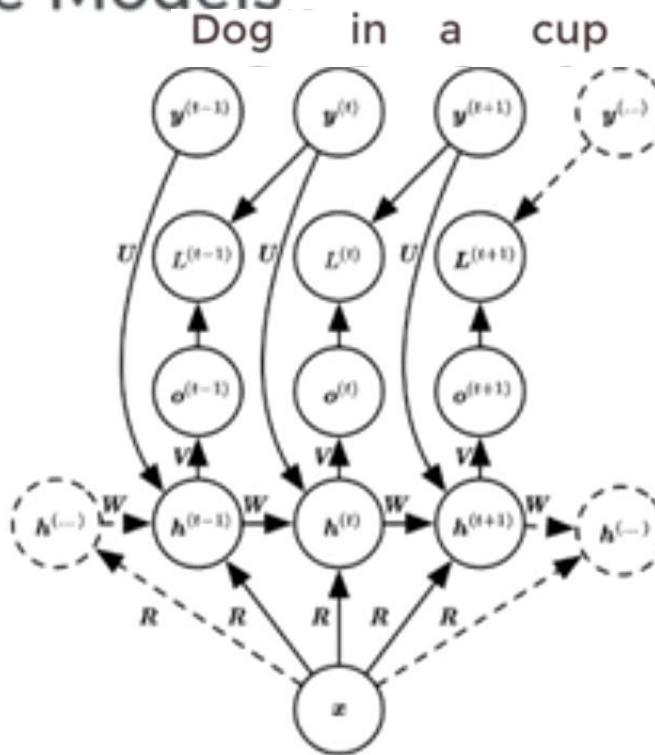
Classification
Question Answering
Passage Topic
Sentiment Analysis

Generating New Content



Recurrent Neural Networks

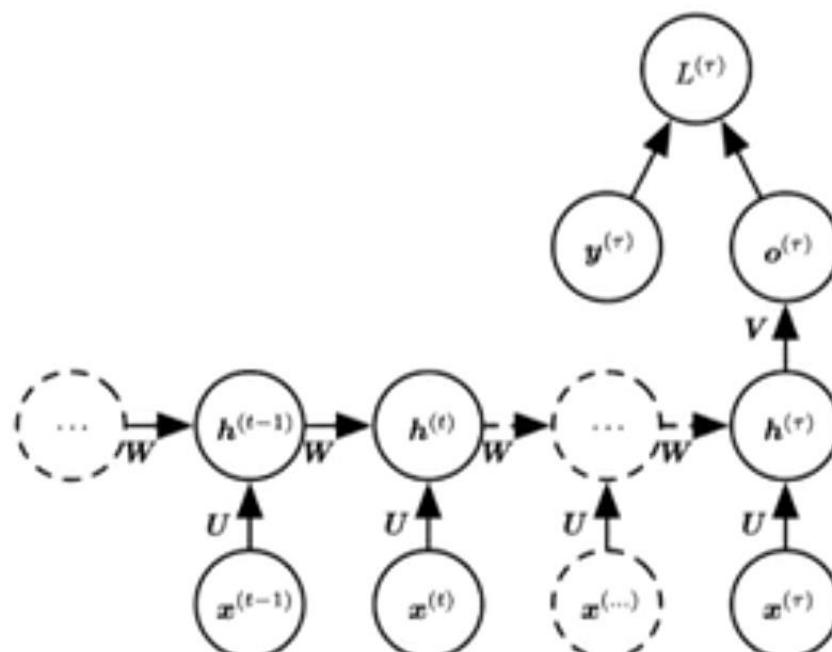
1. Vector-Sequence Models



Recurrent Neural Networks

2. Sequence-Vector Models

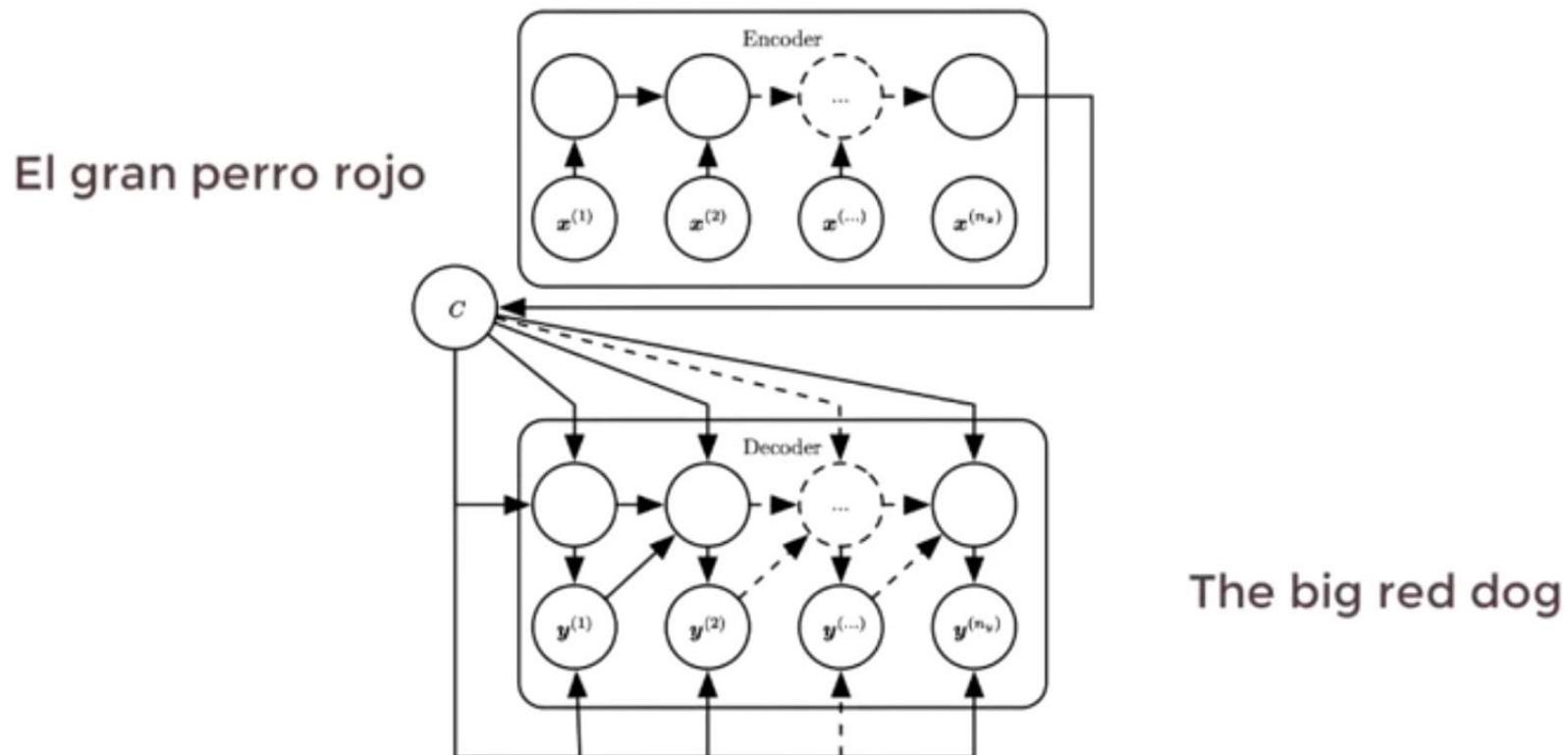
[0.11] good
[0.89] bad



The main character sucked

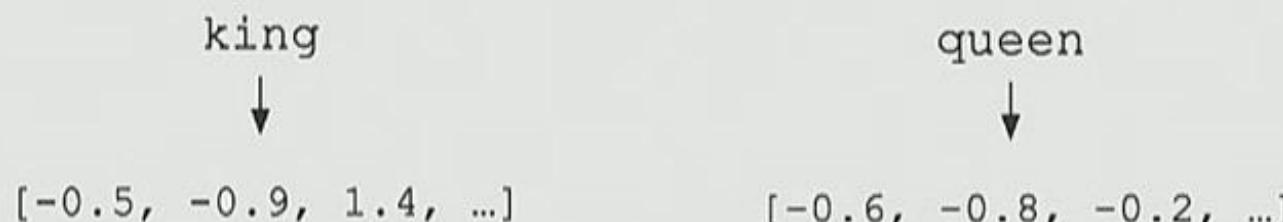
Recurrent Neural Networks

3. Sequence-Sequence Models

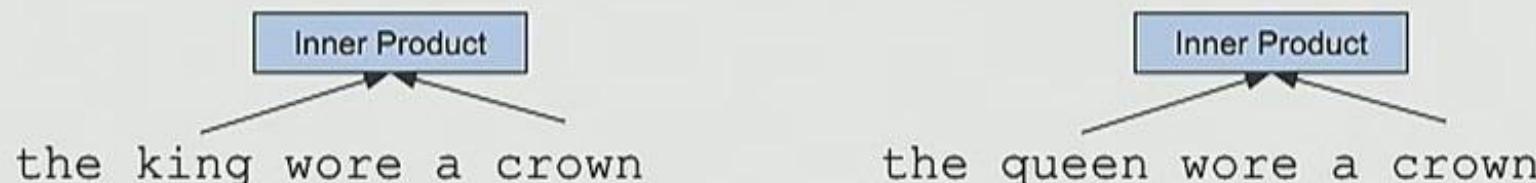


Pre-training in NLP

- Word embeddings are the basis of deep learning for NLP

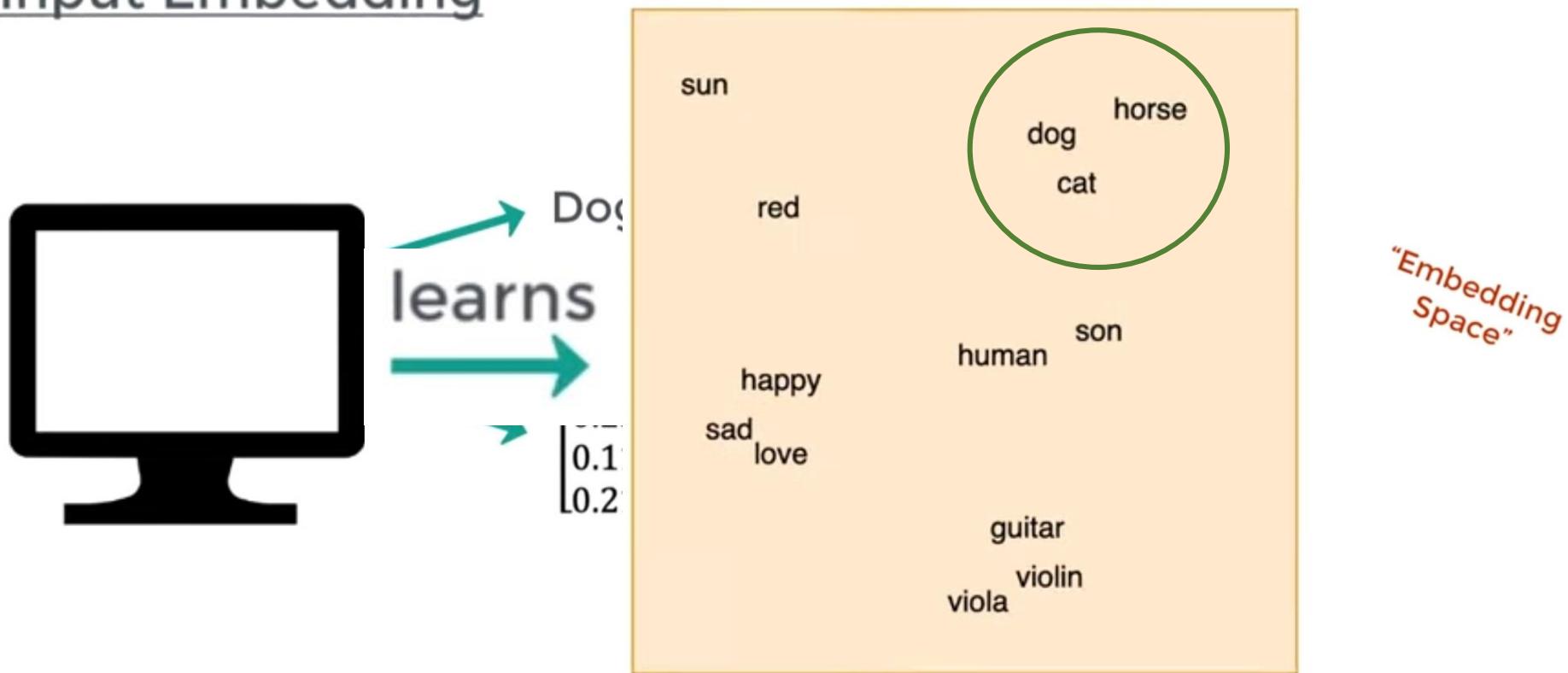


- Word embeddings (word2vec, GloVe) are often pre-trained on text corpus from co-occurrence statistics



Transformer Components

Input Embedding



Transformer Components

Input Embedding

dog 

AJ's **dog** is a cutie

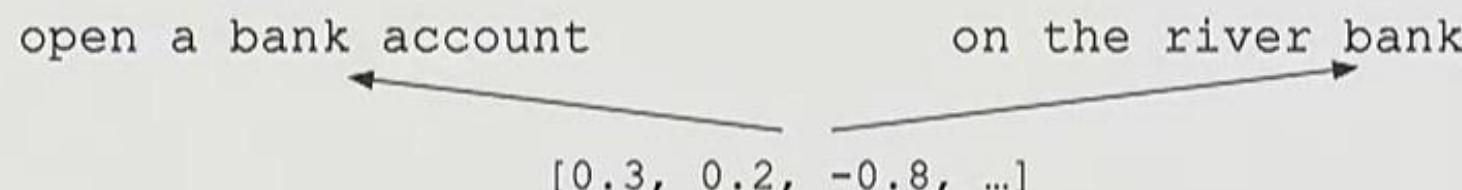
AJ looks like a **dog**



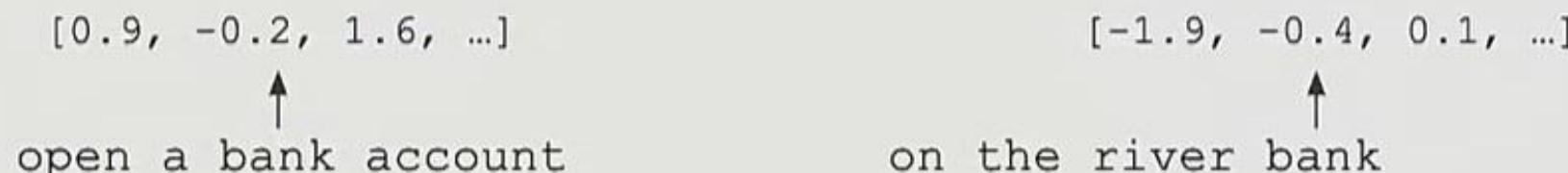
$$\begin{bmatrix} 0.37 \\ 0.99 \\ 0.01 \\ 0.08 \end{bmatrix}$$

Contextual Representations

- **Problem:** Word embeddings are applied in a context free manner



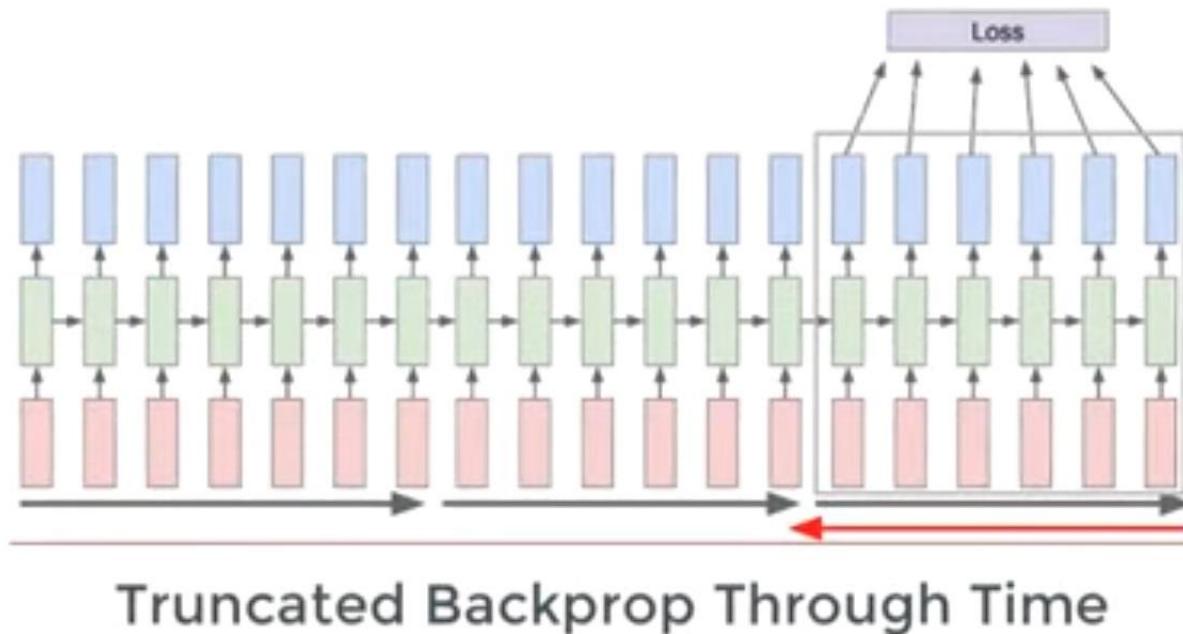
- **Solution:** Train contextual representations on text corpus



Recurrent Neural Networks

Disadvantages

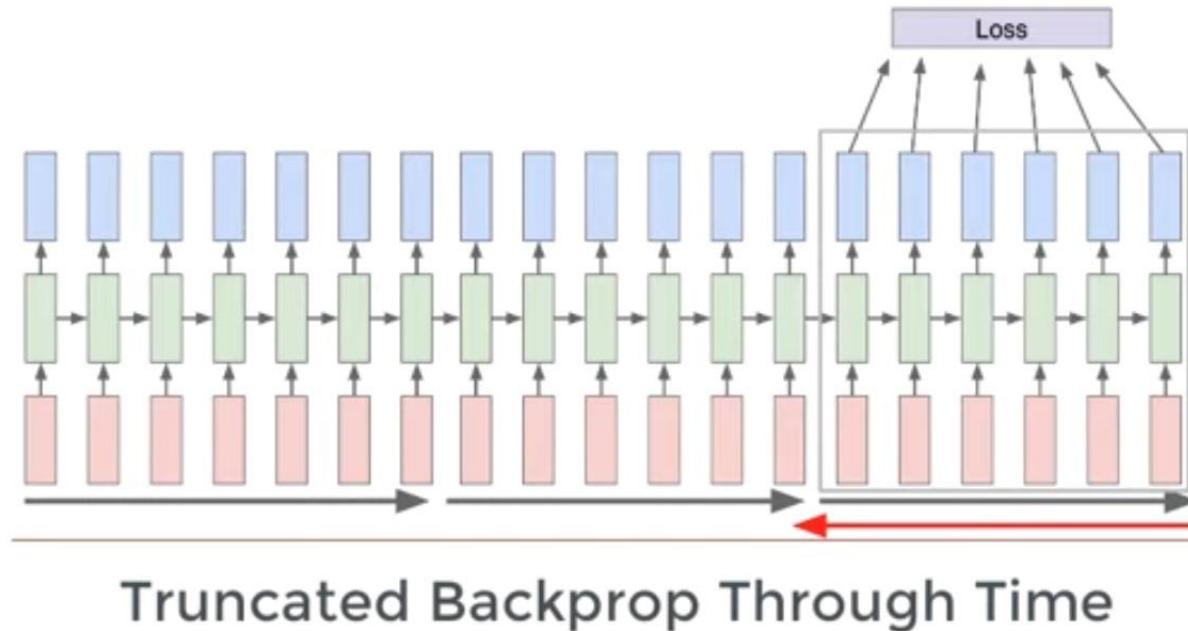
1. Slow to train.



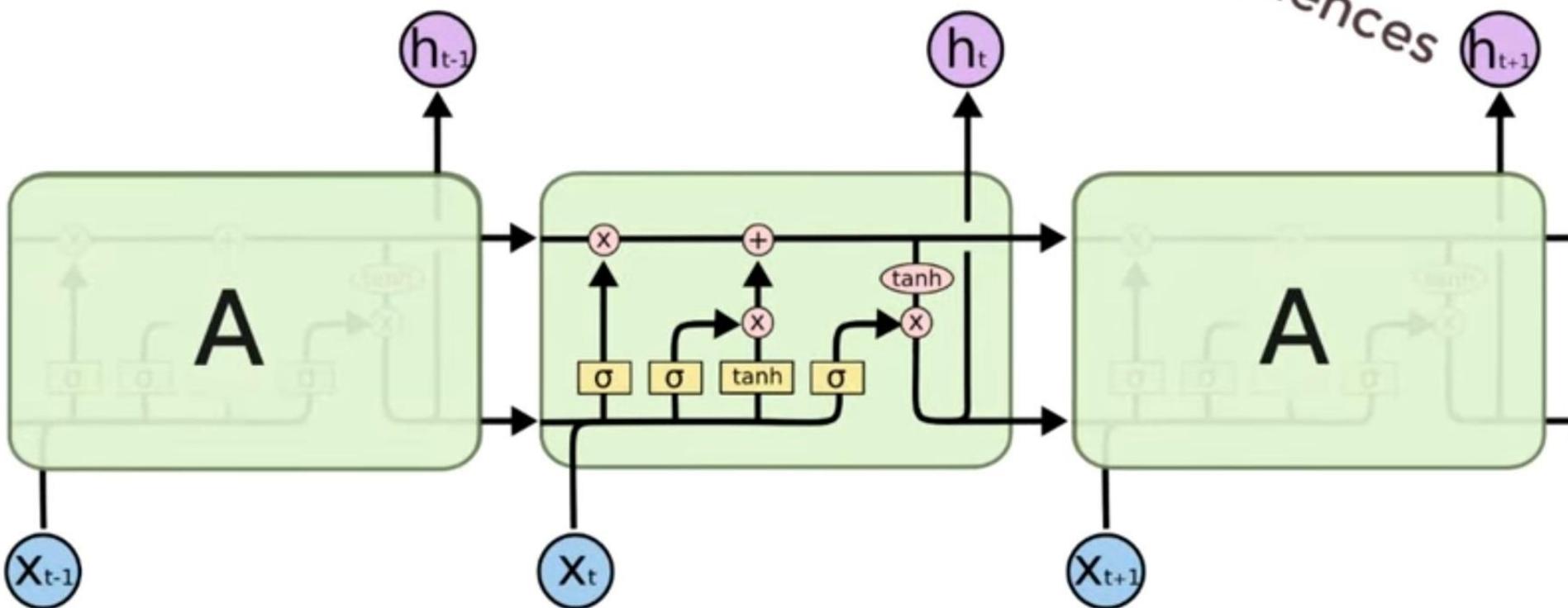
Recurrent Neural Networks

Disadvantages

1. Slow to train.
2. Long sequences lead to vanishing/exploding gradients
3. It not bidirectional



LSTM Networks

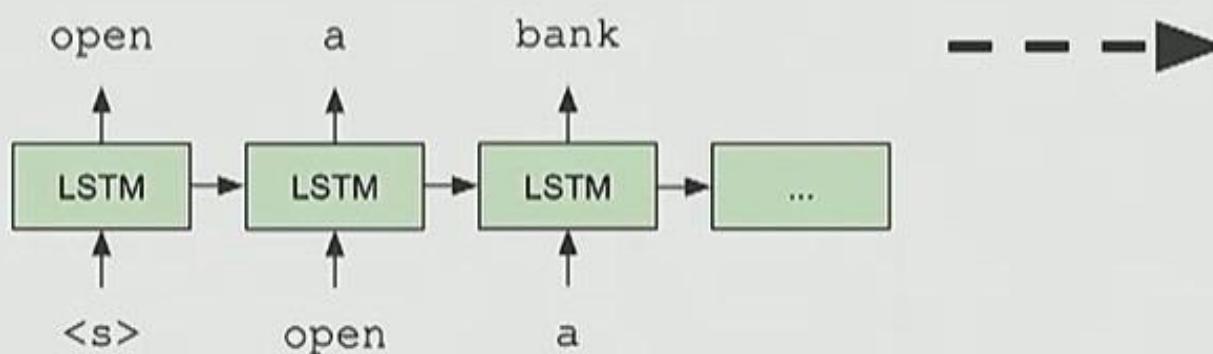


The repeating module in an LSTM contains four interacting layers.

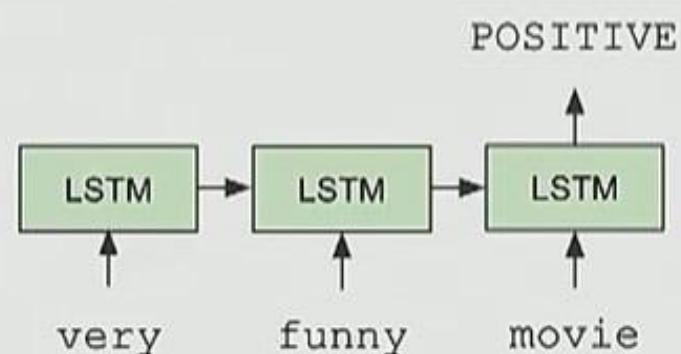
History of Contextual Representations

- *Semi-Supervised Sequence Learning*, Google, 2015

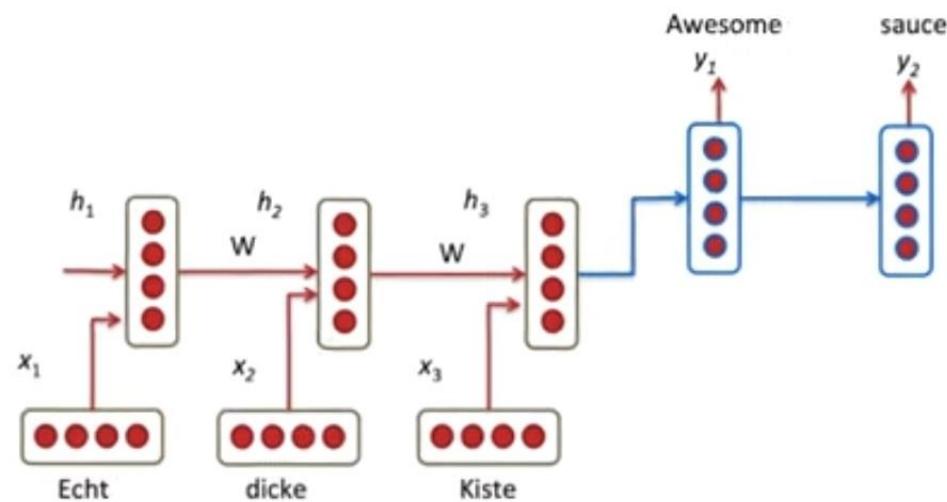
**Train LSTM
Language Model,**



**Fine-tune on
Classification Task**



LSTM

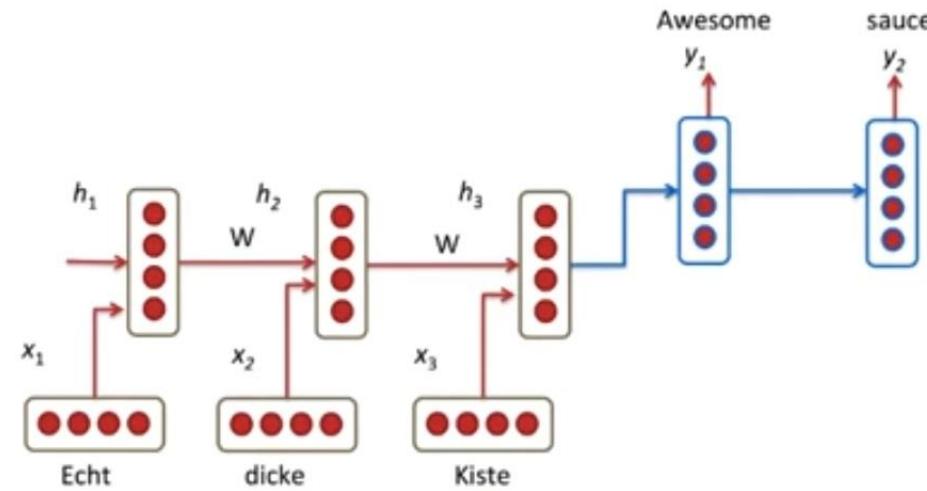


Source [Blog]: Attention & Memory in Deep Learning (Britz, 2017)

LSTM Vs Transformer

LSTM Networks

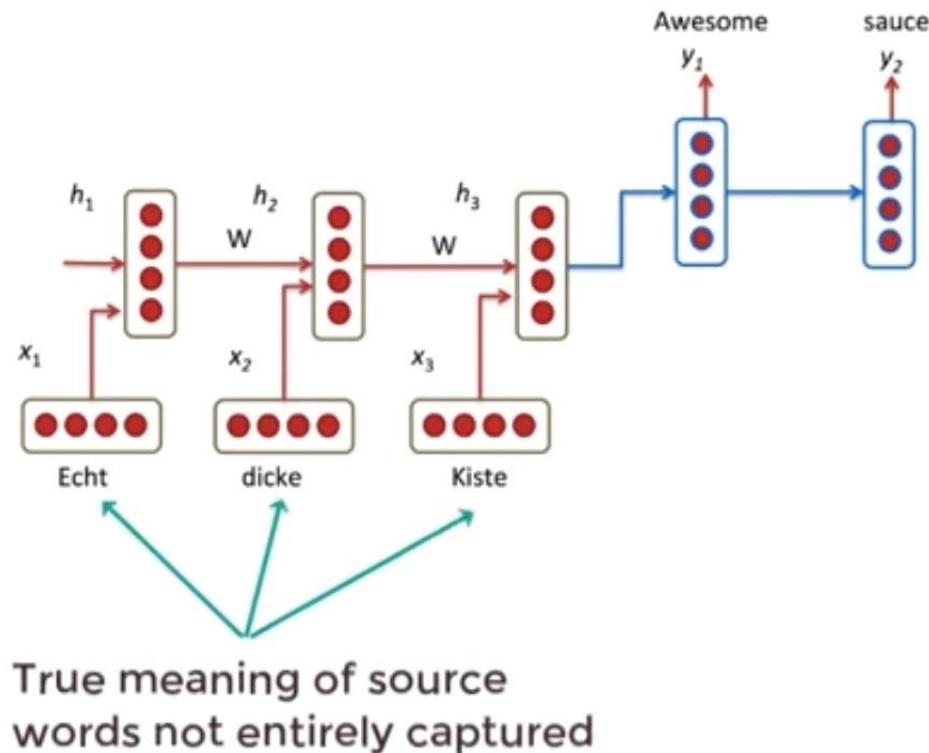
I. Slow



Source [Blog]: Attention & Memory in Deep Learning (Britz, 2017)

LSTM

- 1. Slow
- 2. Not truly Bidirectional



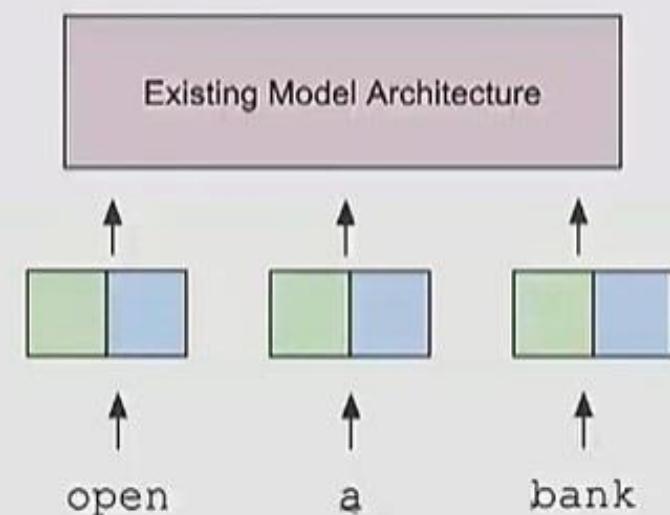
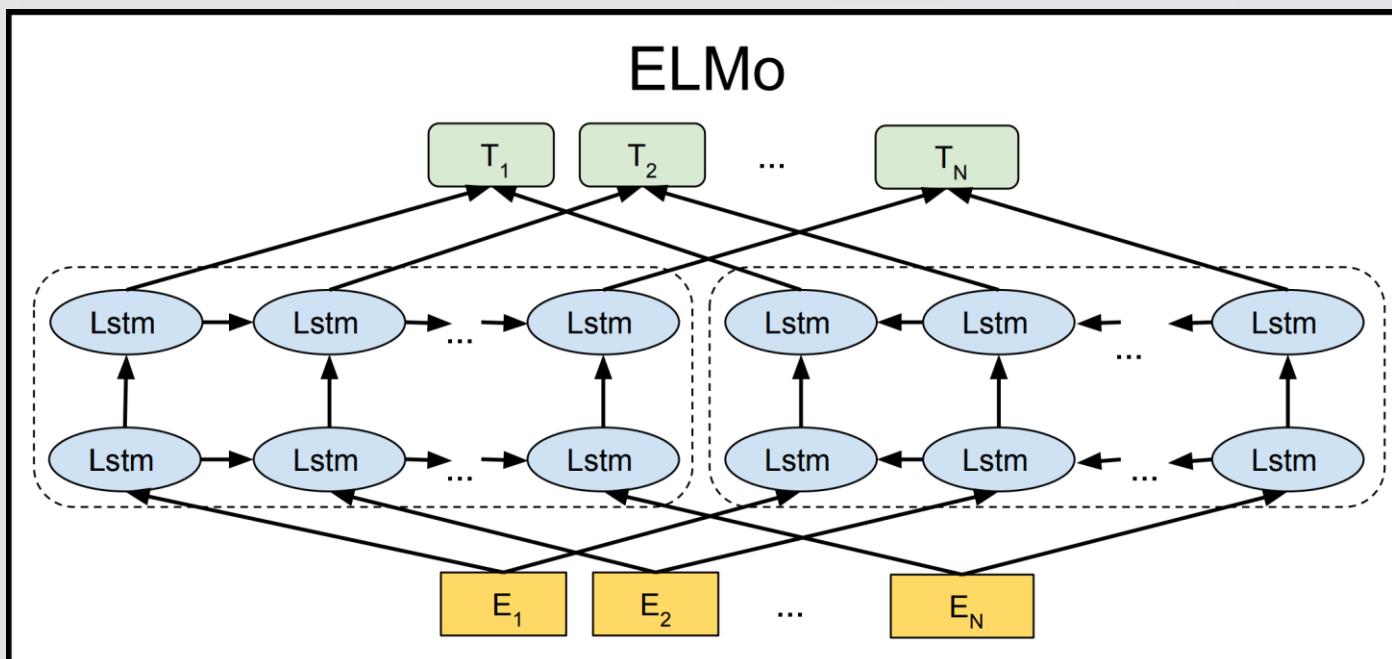
Source [Blog]: Attention & Memory in Deep Learning (Britz, 2017)

History of Contextual Representations

- *ELMo: Deep Contextual Word Embeddings*, AI2 & University of Washington, 2017

Train Separate Left-to-Right and Right-to-Left LMs

Apply as “Pre-trained Embeddings”



Model Architecture

- Empirical advantages of Transformer vs. LSTM:
 1. Self-attention == no locality bias
 - Long-distance context has “equal opportunity”
 2. Single multiplication per layer == efficiency on TPU
 - Effective batch size is number of words, not sequences

Transformer

X_0_0	X_0_1	X_0_2	X_0_3
X_1_0	X_1_1	X_1_2	X_1_3

$$\times \quad W$$

LSTM

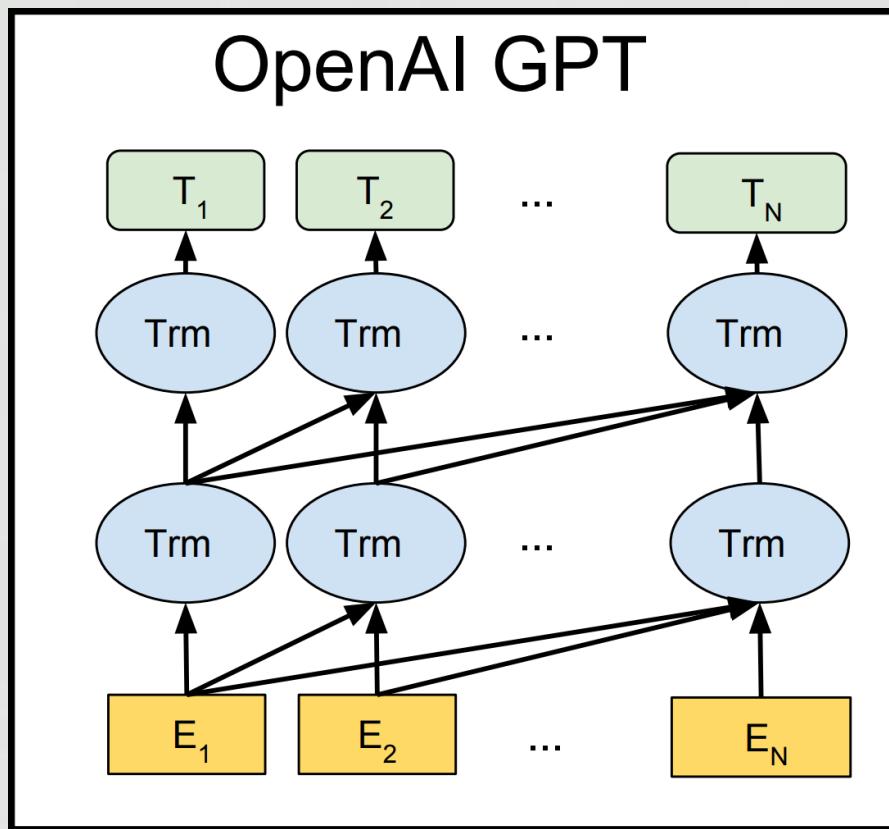
X_0_0	X_0_1	X_0_2	X_0_3
X_1_0	X_1_1	X_1_2	X_1_3

$$\times \quad W$$

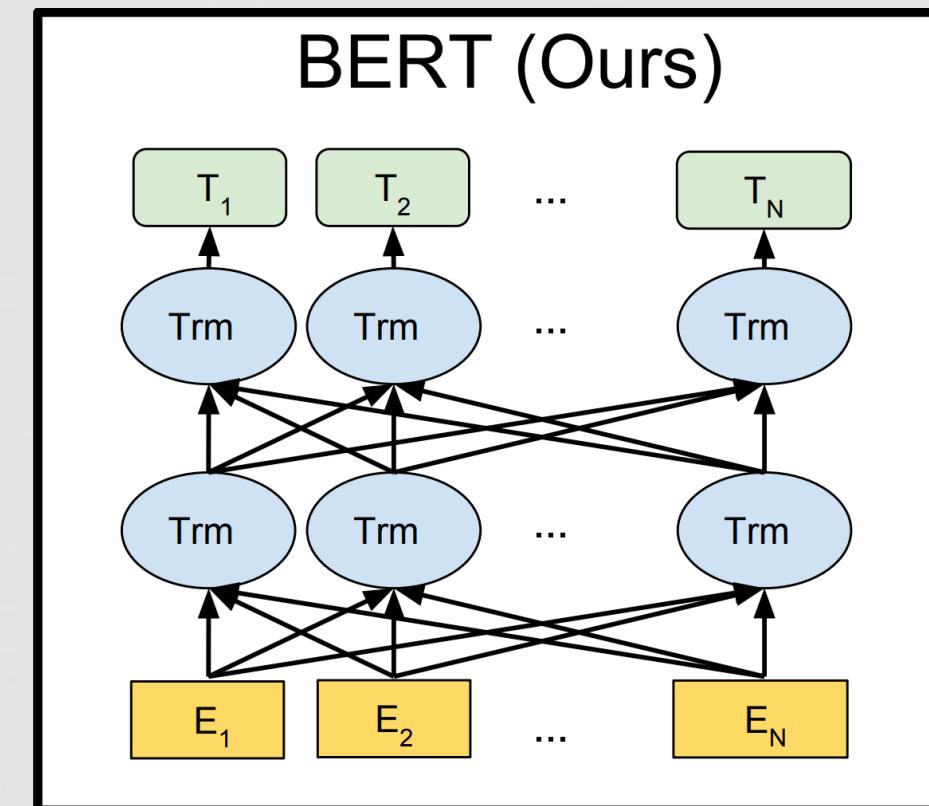
Unidirectional vs. Bidirectional Models

Unidirectional context

Build representation incrementally

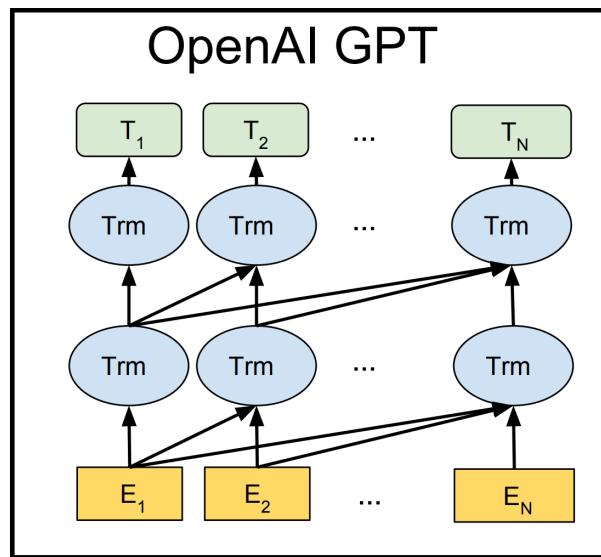


Bidirectional context
Words can “see themselves”



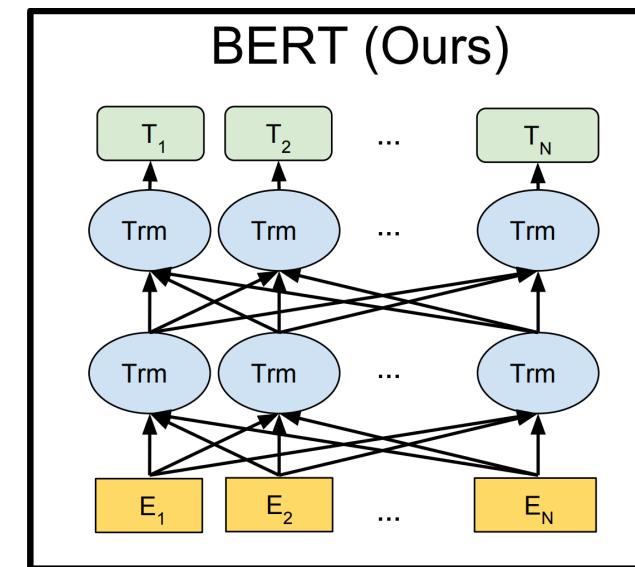
Unidirectional vs. Bidirectional Models

Unidirectional



It is good to understand a Language
But it required to input one by one
It uses for specific task
Training Task is: next word prediction

Bidirectional



It gives us better Understanding because of Bidirectional
It input all words simultaneously
It is not good for all part generation tasks
So how we can train the model to language understanding
Different Training Tasks

Masked LM

- **Solution:** Mask out $k\%$ of the input words, and then predict the masked words
 - We always use $k = 15\%$

- Too little masking: Too expensive to train
 - Too much masking: Not enough context

Next Sentence Prediction

- To learn *relationships* between sentences, predict whether Sentence B is actual sentence that proceeds Sentence A, or a random sentence

Sentence A = The man went to the store.

Sentence B = He bought a gallon of milk.

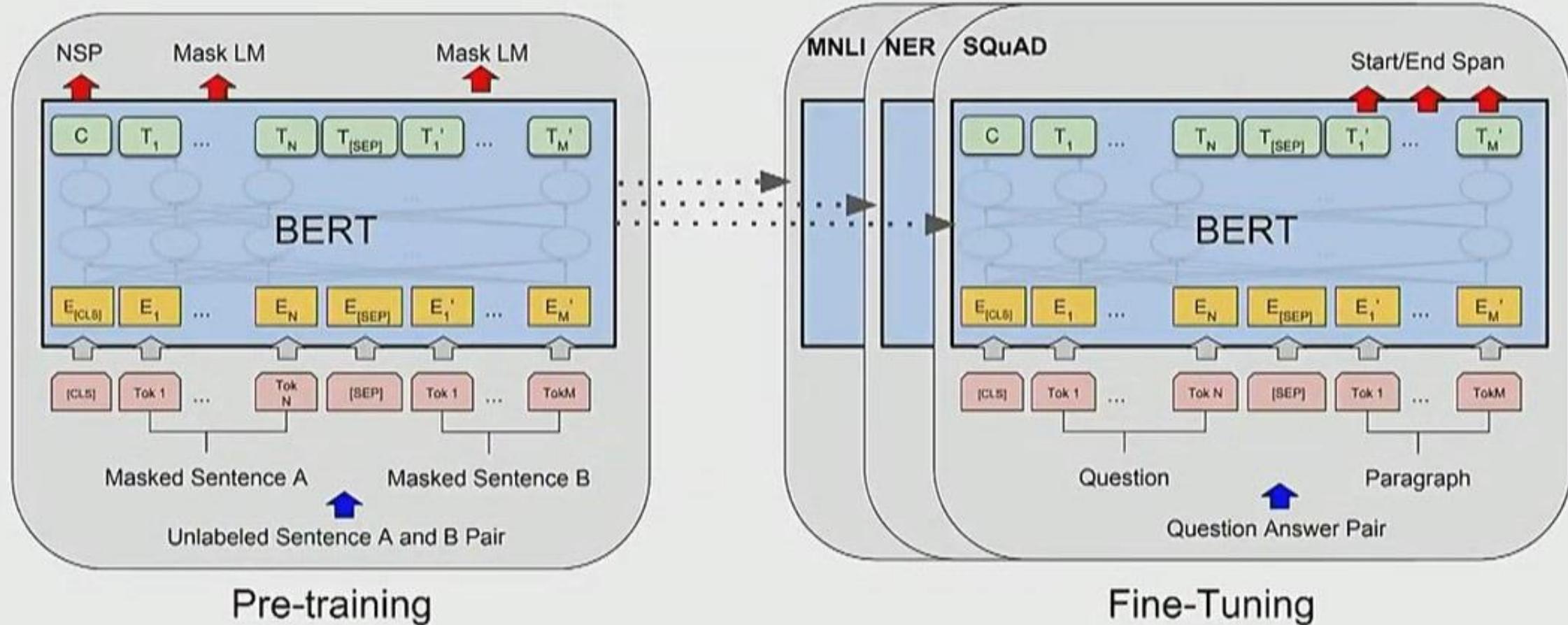
Label = IsNextSentence

Sentence A = The man went to the store.

Sentence B = Penguins are flightless.

Label = NotNextSentence

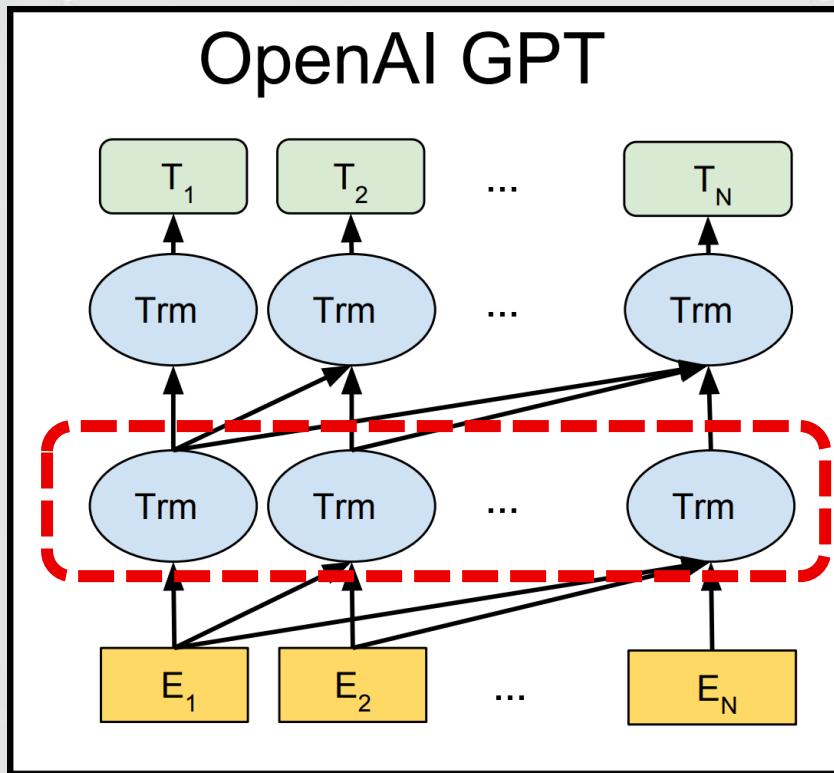
Fine-Tuning Procedure



Unidirectional vs. Bidirectional Models

Unidirectional context

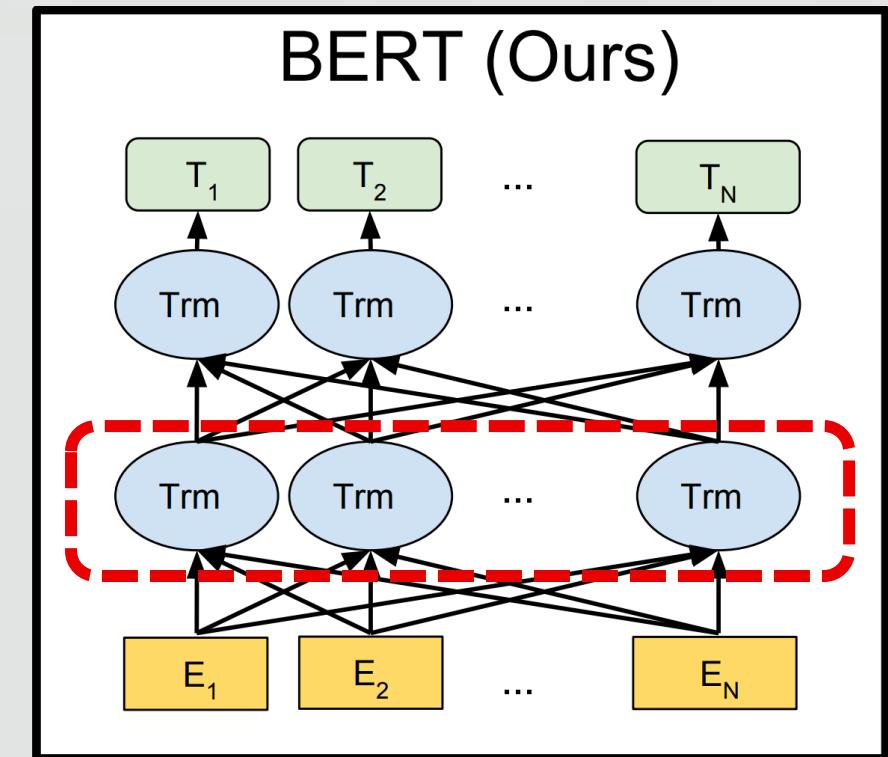
Build representation incrementally



Transformer Decoder Block

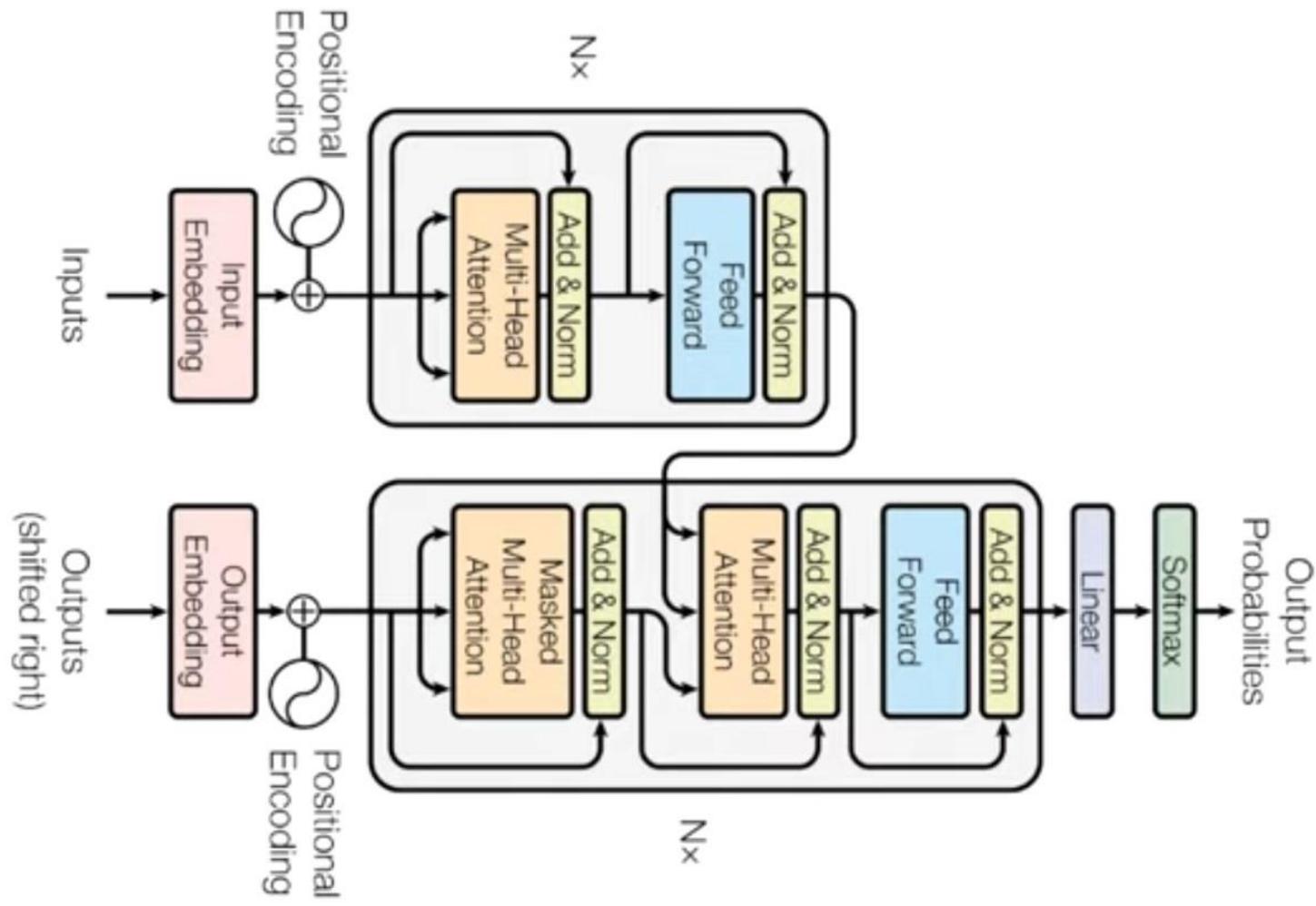
Bidirectional context

Words can “see themselves”



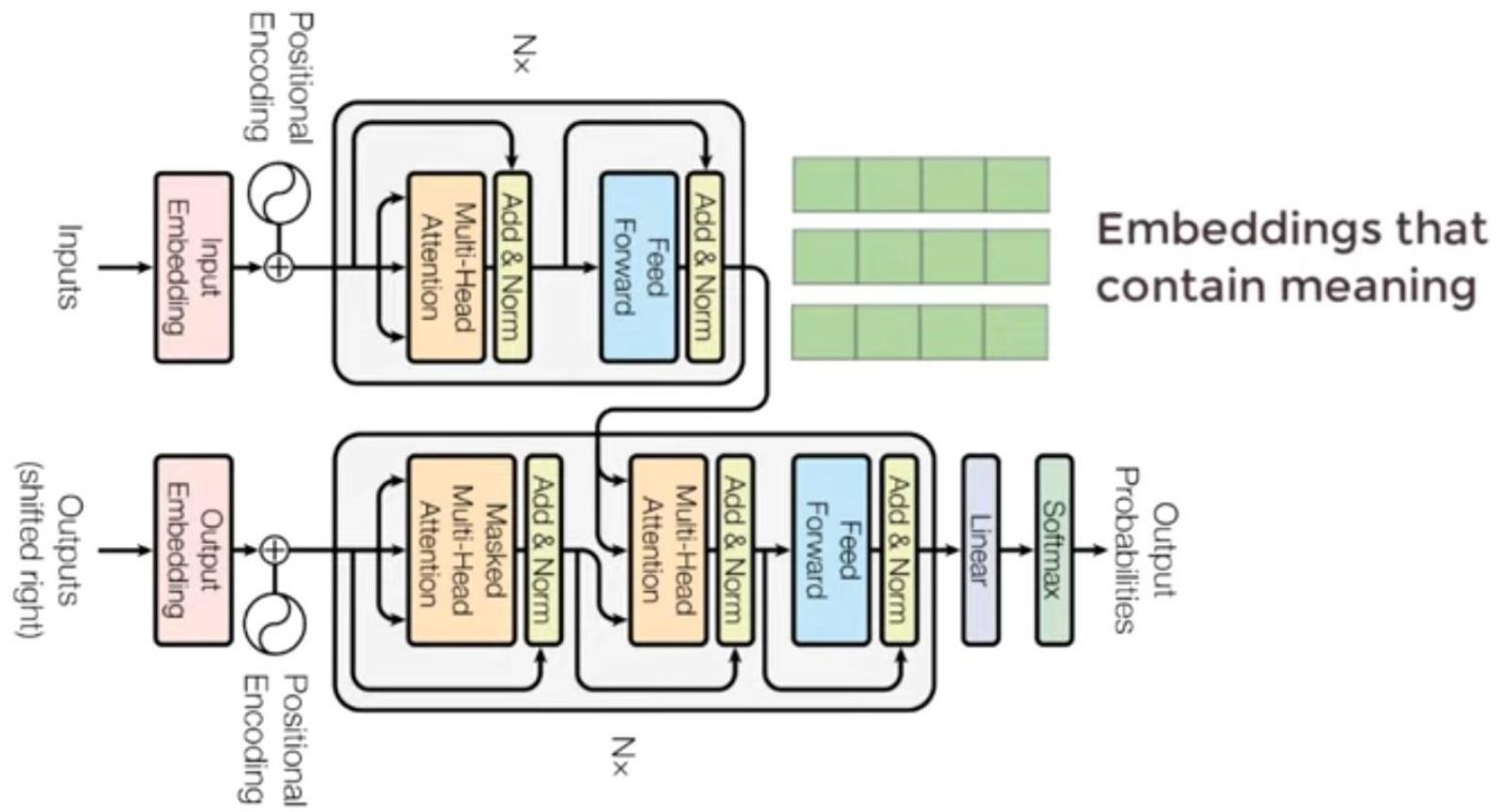
Transformer Encoder Block

Transformer Flow

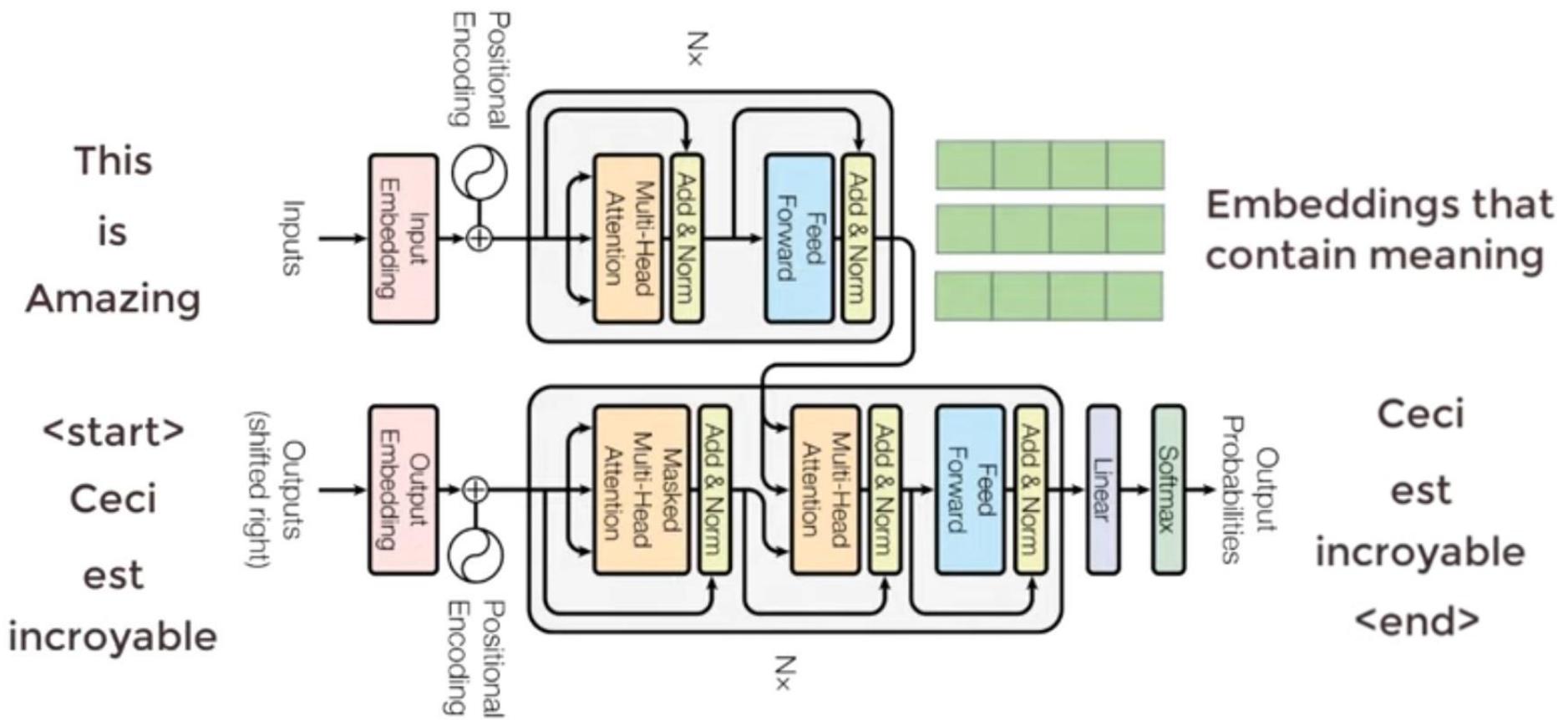


Transformer Flow

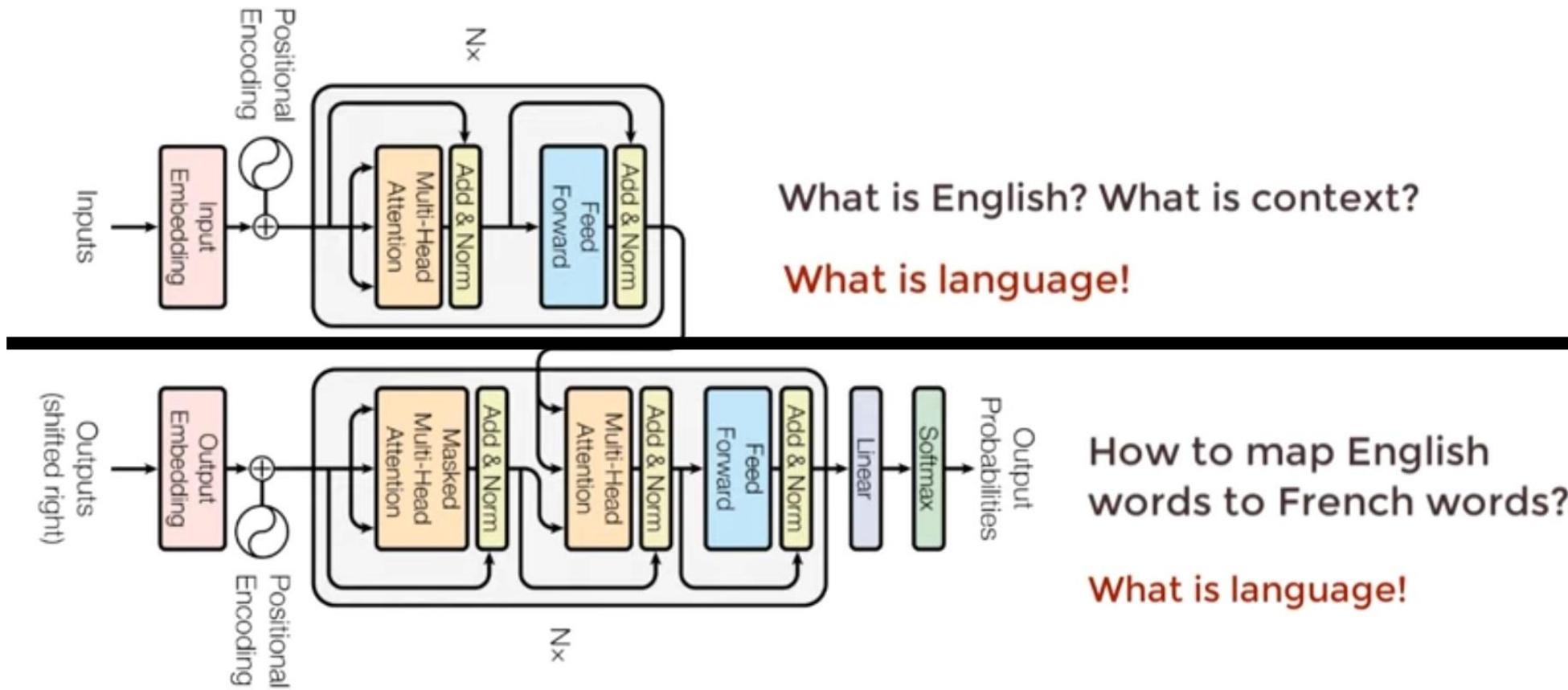
This
is
Amazing



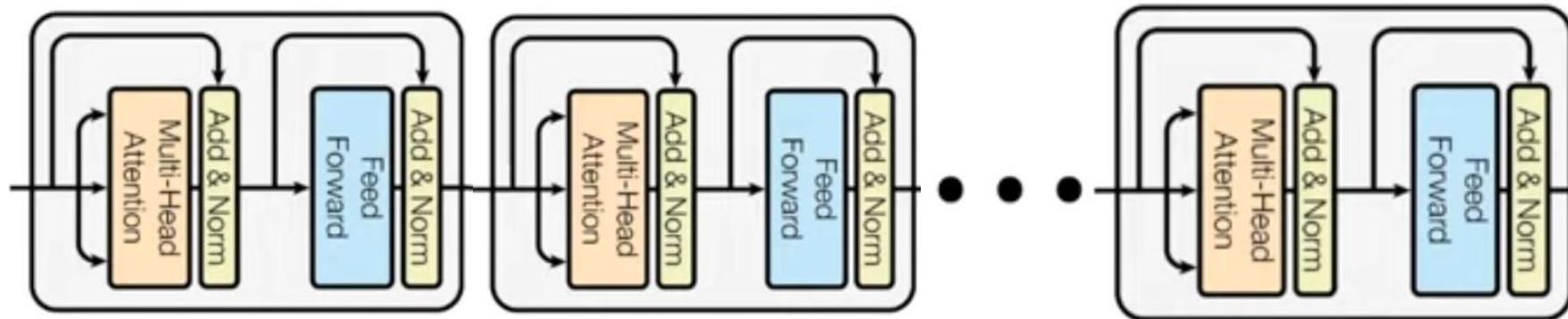
Transformer Flow



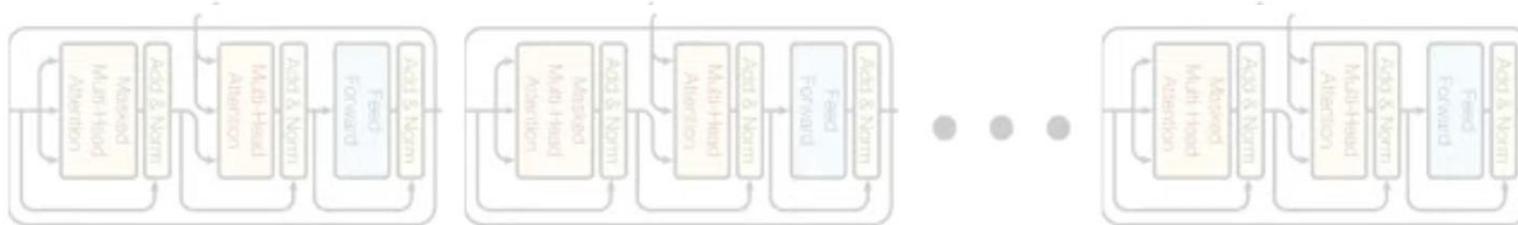
Transformer Flow



Transformer Flow

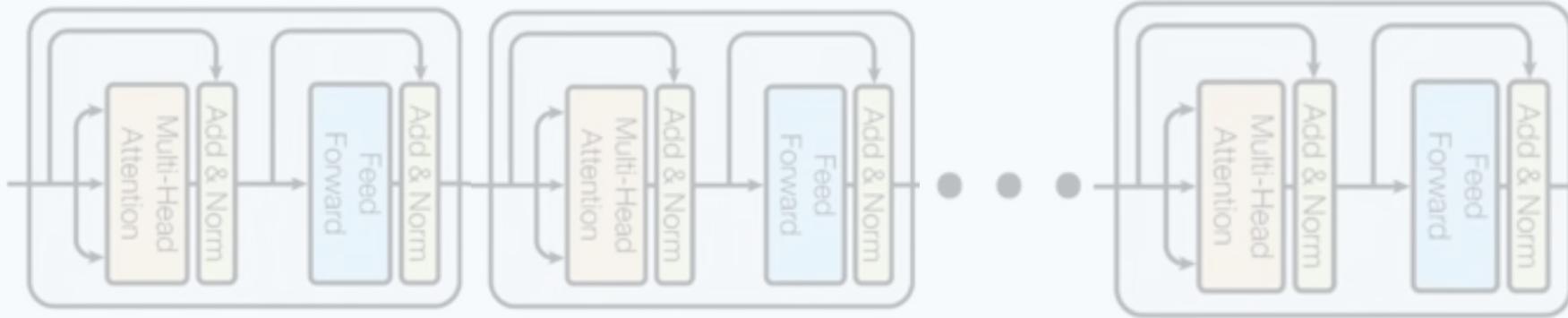


Bidirectional Encoder Representation from Transformers

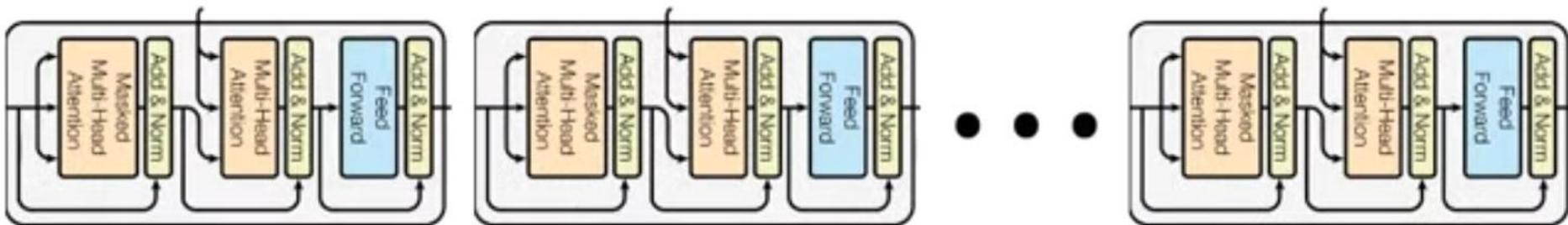


GPT

Transformer Flow



Bidirectional Encoder Representation from Transformers



G^PT



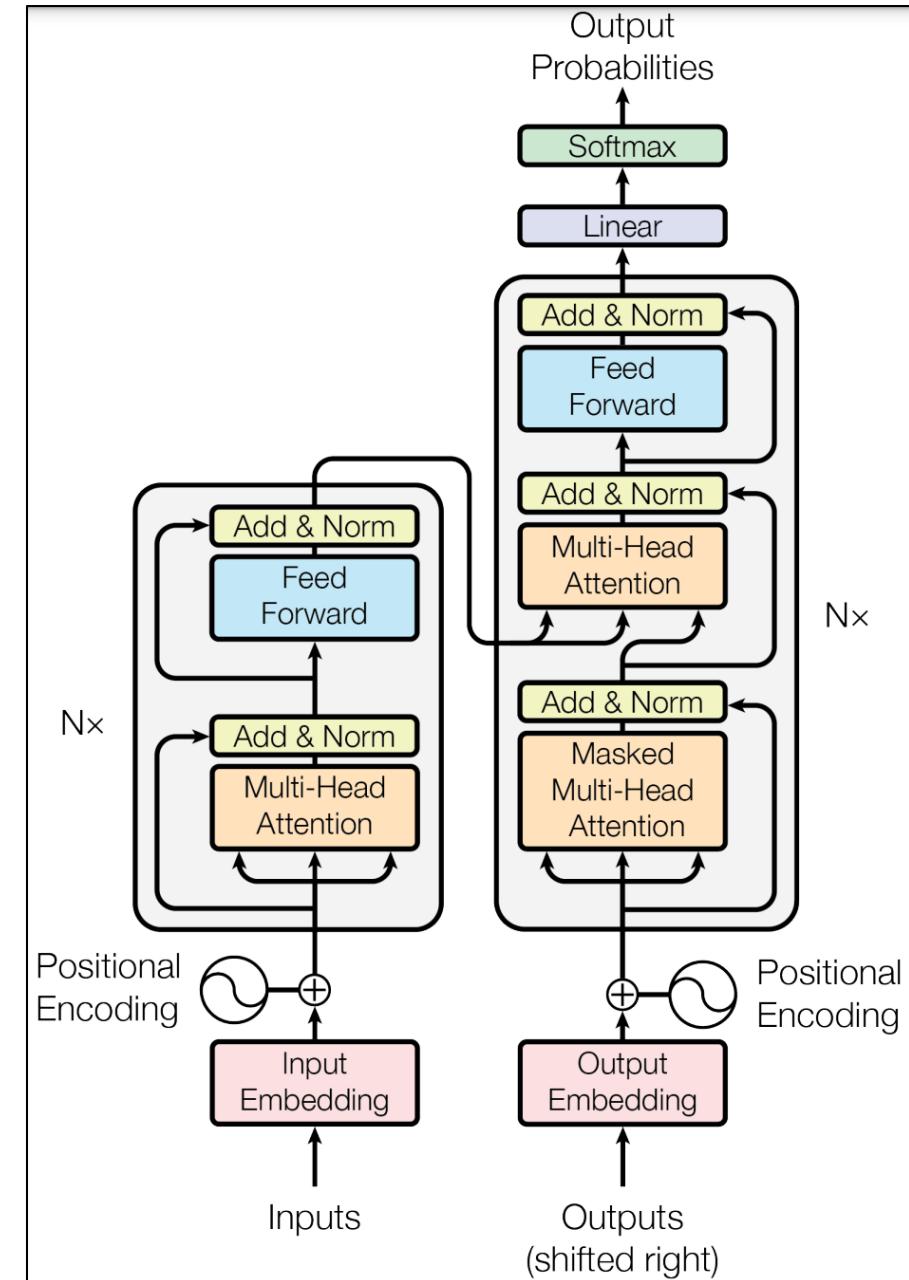
Automatic Exam Marking Project

Phase 1: Exam Management Product

Product ML Engine

PART 2 : Transformer Architecture Overview
&
Encoder Part Technical Detail

Transformer Flow



Transformer Components

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

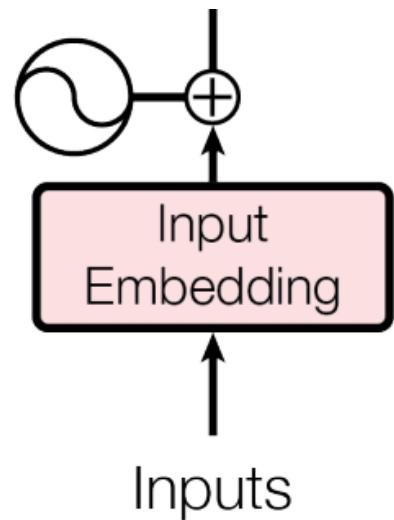
AJ's **dog** is a cutie → Position 2

AJ looks like a **dog** → Position 5

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

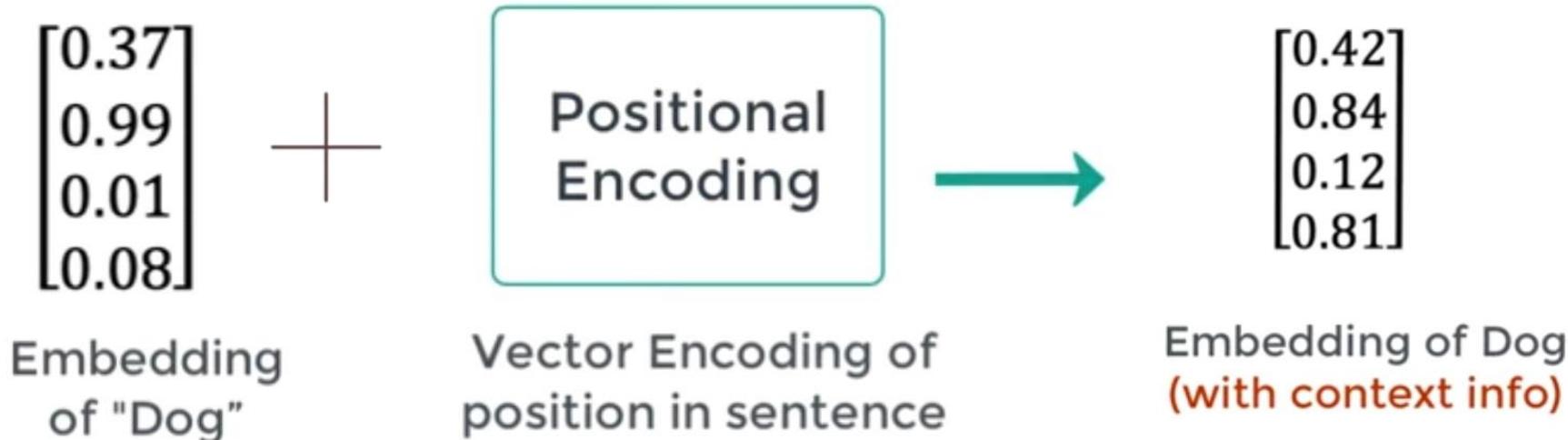
$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{\text{model}}})$$

Positional
Encoding



Transformer Components

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



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

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

NLP Tasks

Pre-train Phase

Language
Understanding

Model Tuning

Classification
Question Answering
Passage Topic
Sentiment Analysis

Required Language understanding:

the meaning and concept of each word of input based on word's position in a passage by considering the whole meaning of the passage. This is a process to provide contextual meaning of the words.

Two Step:

Pre-train for language understanding phase and Fine tuning for customizing the pre-trained model to doing an specific NLP task.

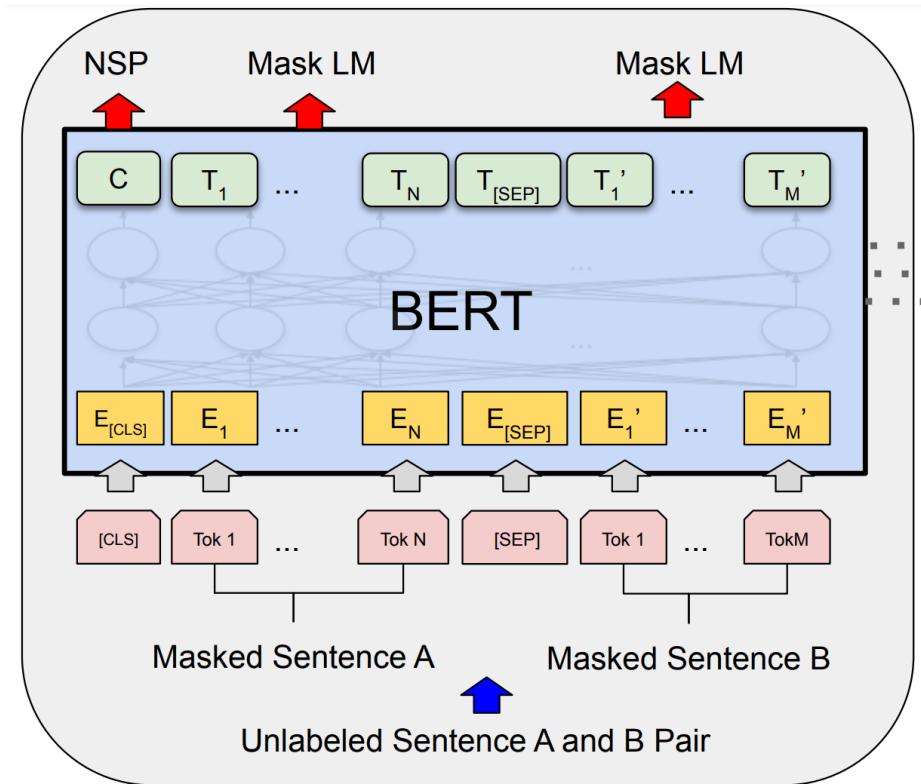
Advantageous of Pre-training:

- 1- Opportunity to use free and huge amount of unsupervised data to have a good pre-training and language understanding
- 2- decreasing execution time of fine tuning training, because we already know language and in fine tuning we need to learn few parameters for a specific task

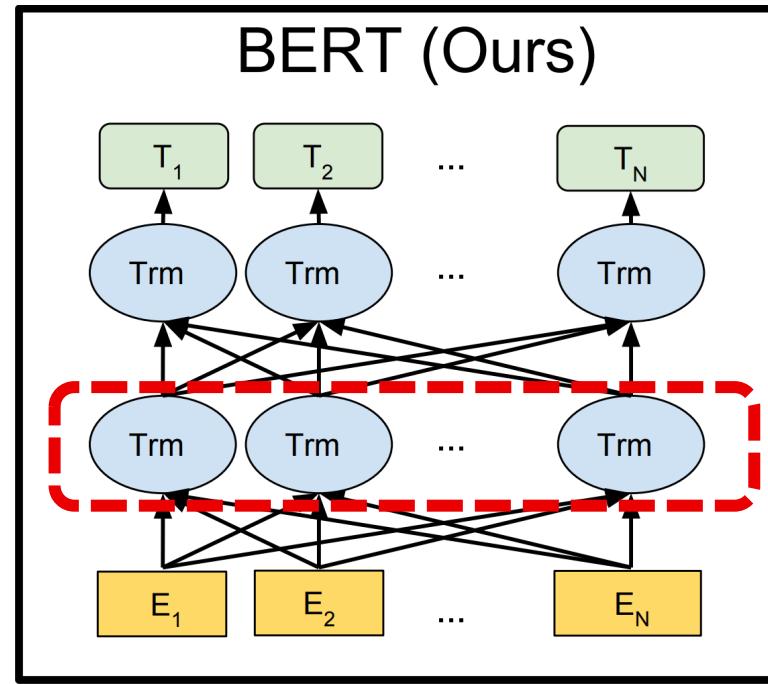
NLP Tasks

	Problem	Input	Pre-training Learning Tasks
RNN	1-Slowness 2- Gradient Vanishing and exploding 3- Unidirectional	One word in each Time stamp	Next Word Prediction
LSTM	1-Slowness 2- Unidirectional	One word in each Time stamp	Next Word Prediction
ELMO	1-Slowness 2- Semi- Bidirectional	One word in each Time stamp	Next Word Prediction
Open AI	1- Unidirectional (Useful for specific tasks)	Push all word simultaneously/ One word in each time stamp	Next Word Prediction
Bert	1- Not applicable for some tasks	Push all word simultaneously	Mask Prediction Next Sentence Prediction

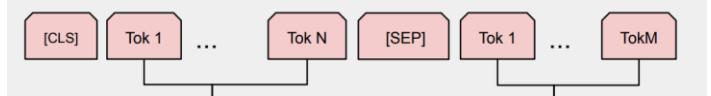
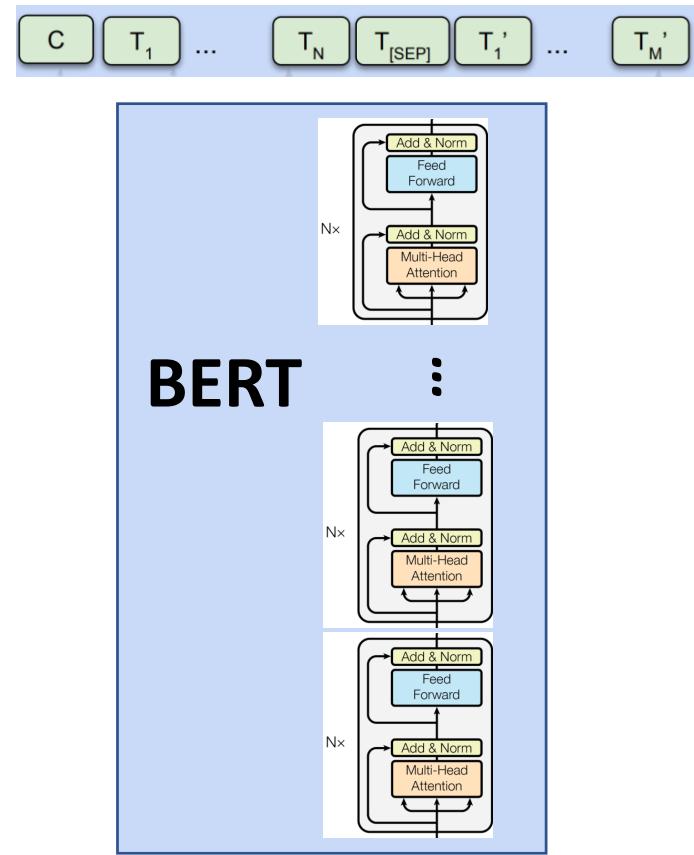
BERT Architecture



Pre-training



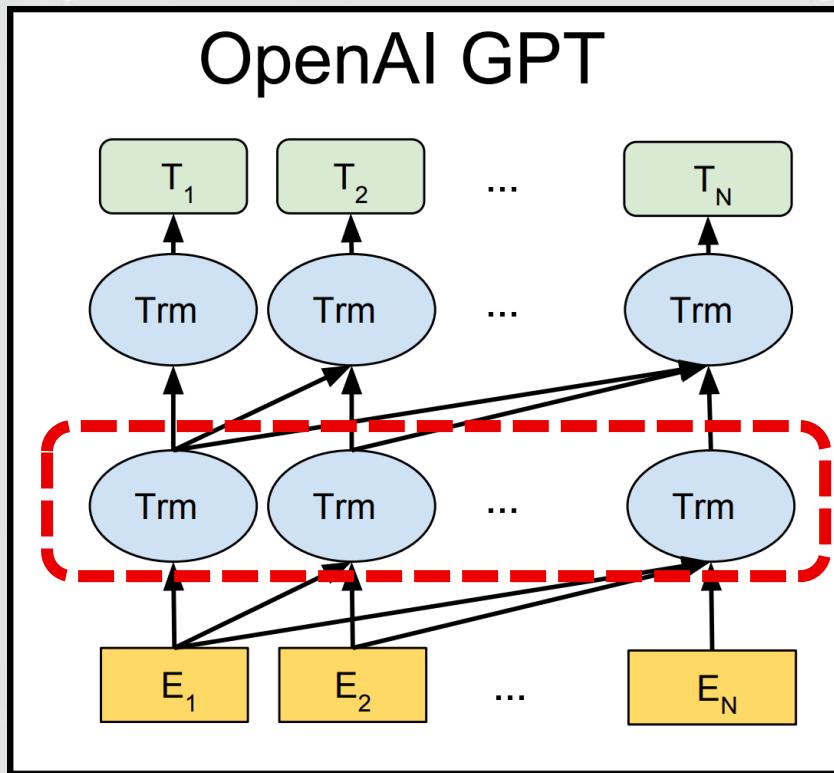
Transformer Encoder Block



Unidirectional vs. Bidirectional Models

Unidirectional context

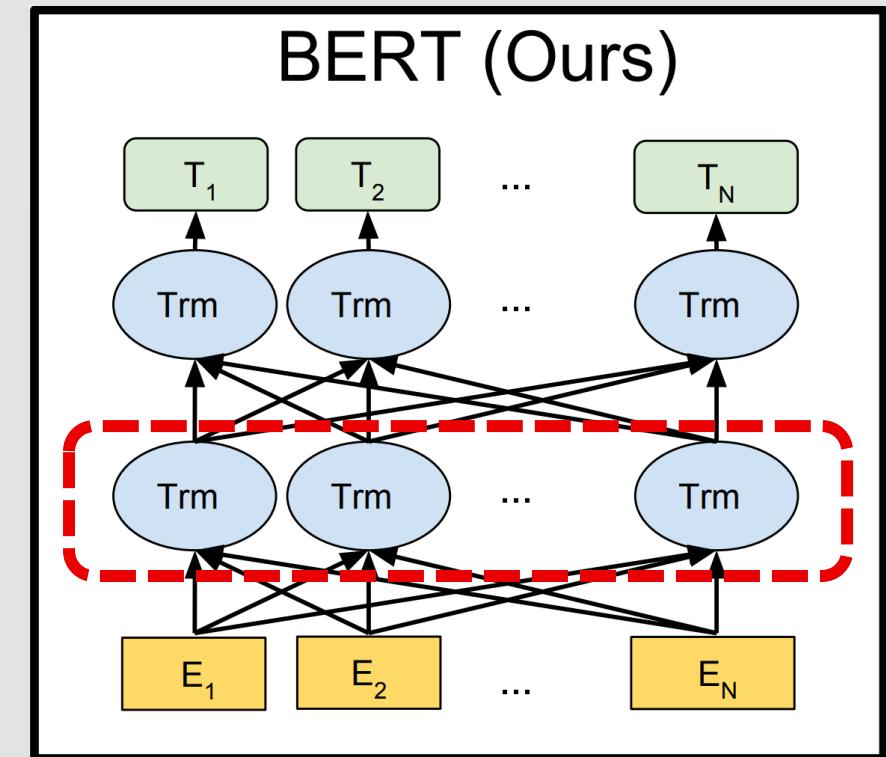
Build representation incrementally



Transformer Decoder Block

Bidirectional context

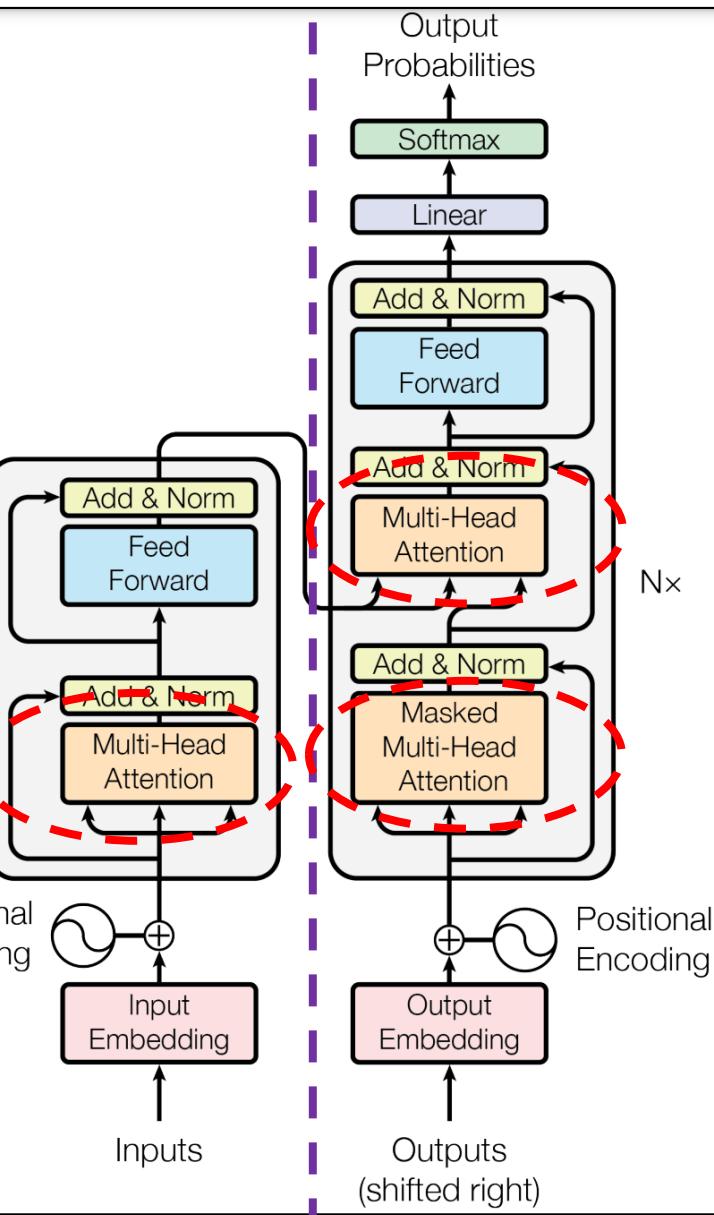
Words can “see themselves”



Transformer Encoder Block

Encoder

Decoder



Transformer Architecture / Attention Matrix

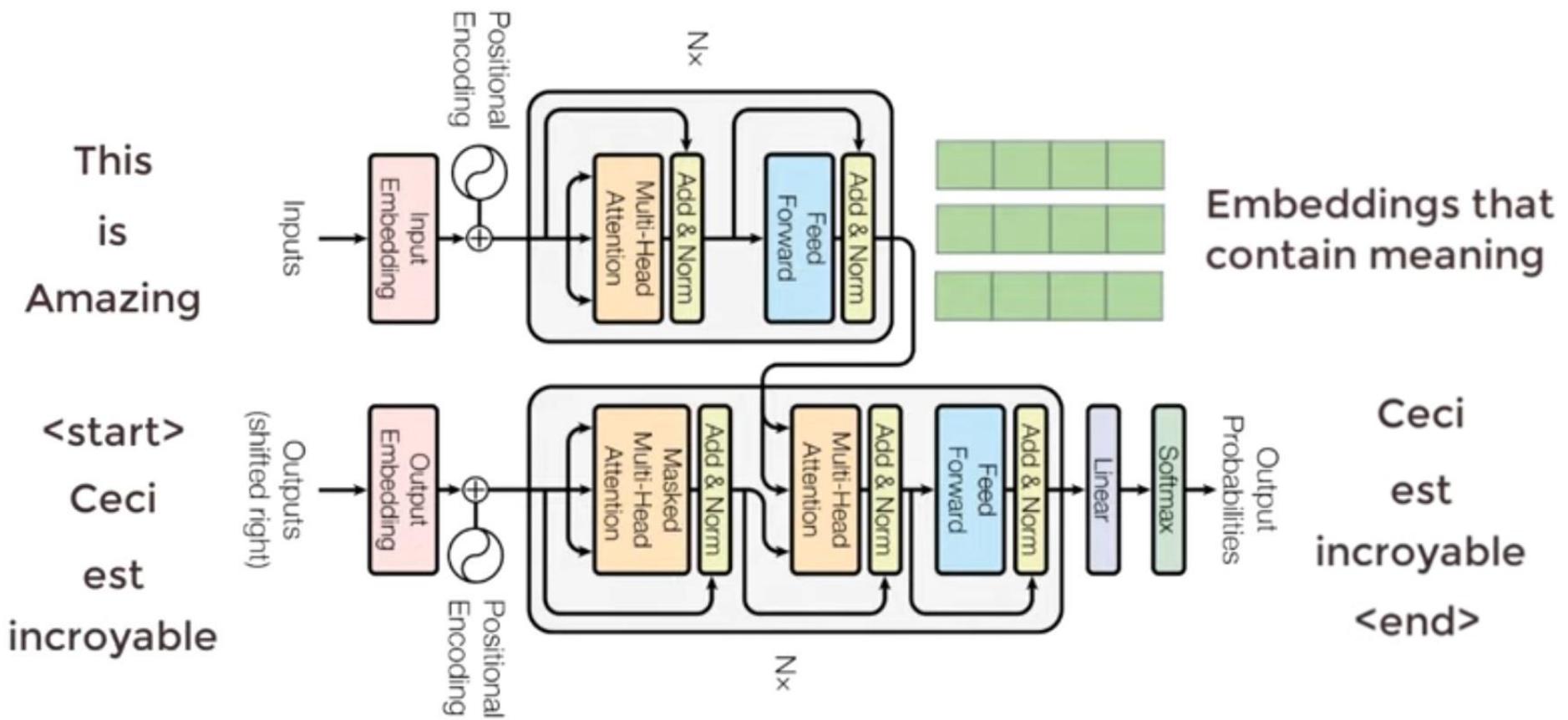
Focus

The → The big red dog
big → The big red dog
red → The big red dog
dog → The big red dog

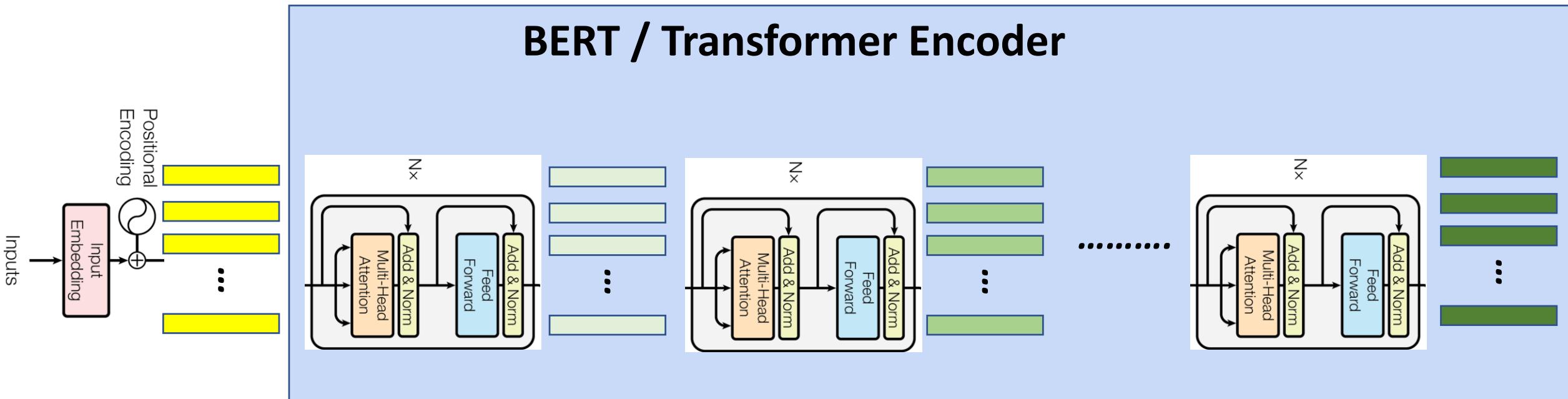
Attention Vectors

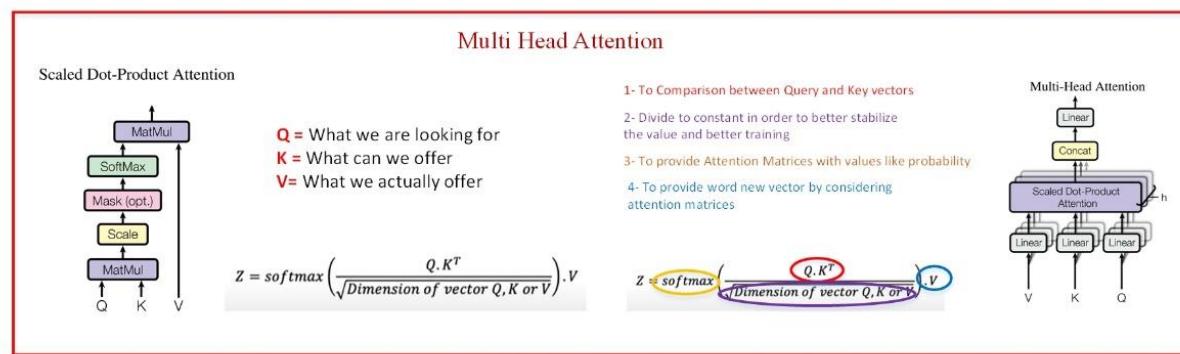
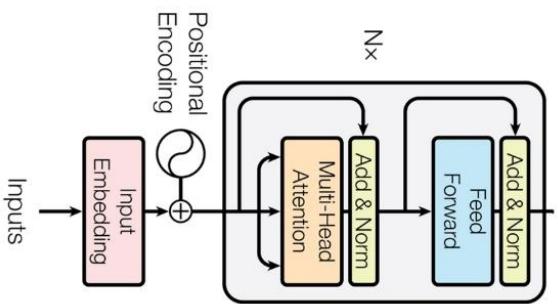
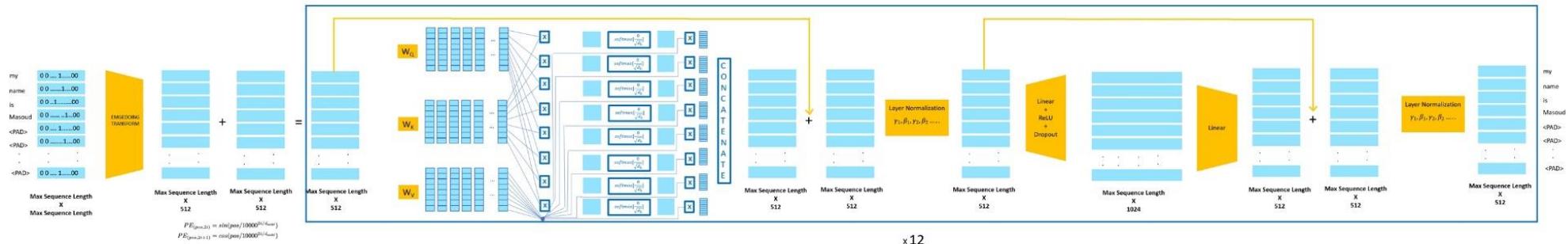
[0.71	0.04	0.07	0.18]
[0.01	0.84	0.02	0.13]
[0.09	0.05	0.62	0.24]
[0.03	0.03	0.03	0.91]

Transformer Flow



Transformer Encoder & Bert Block or Layer





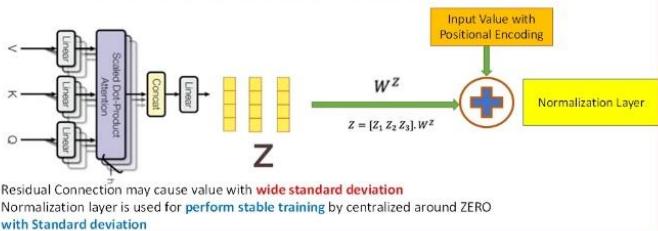
Transformer Decoder Block Workflow



Residual Connection and Normalizing

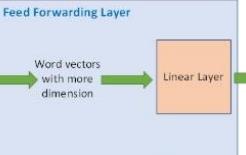
Add the result of Multi-Head attention layer with input value alongside residual connection and perform Normalization Layer

For very deep network the back propagation eventually leads to very small gradients => no parameters update properly and network does not learn



Transformer Flow – Feed Forwarding

Residual Connection and Normalizing



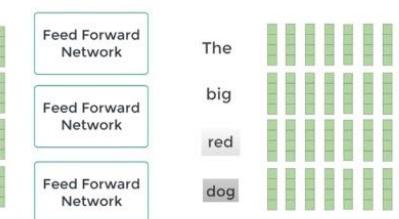
Linear Layer is going to help better interaction among Heads because of just doing concatenation (pushing all Heads to interact with eachother cross the layers)

Relu activation function helps the network understand better more complex pattern

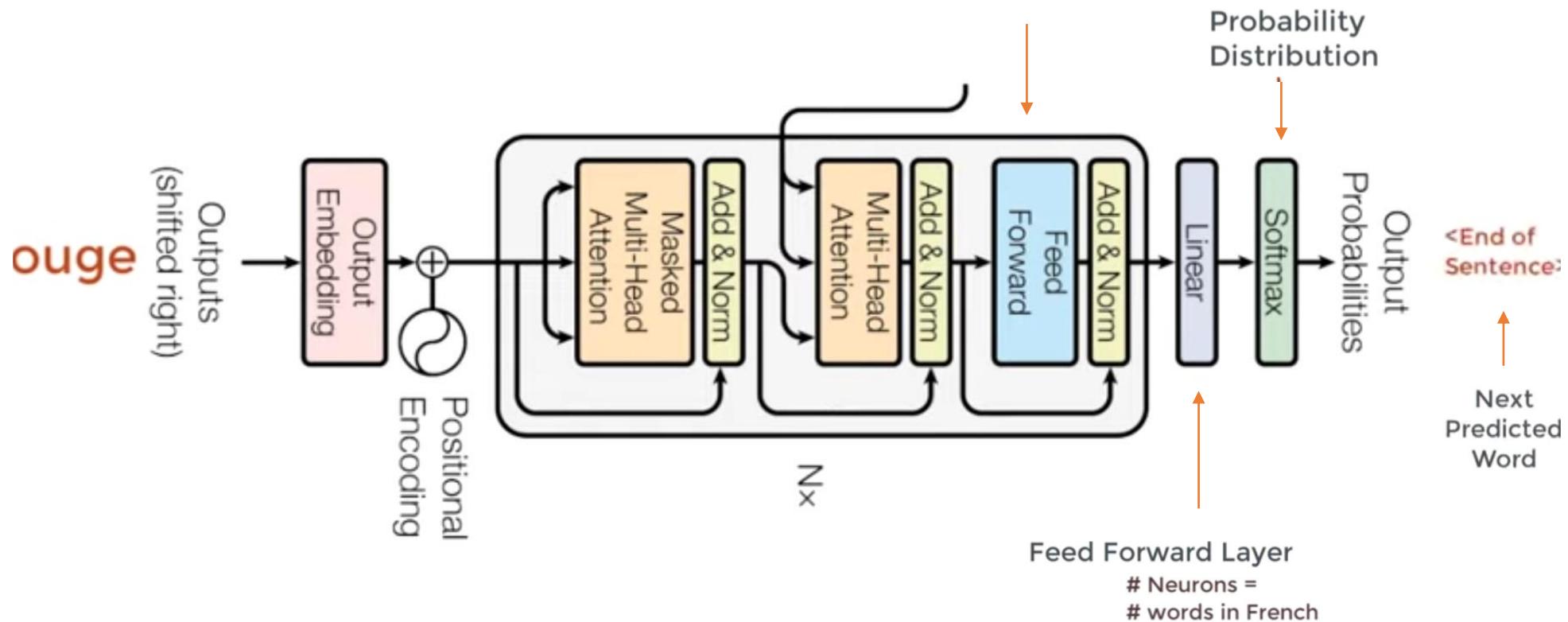
Droput randomly turns off neurons within feed forward layer in order to network better able to generalize and to avoid memorize a specific kind of pattern

Transformer Flow – Feed Forwarding

Residual Connection and Normalizing



Transformer Components



Decoder

Self
Attention

Le → Le gros chien rouge
gros → Le gros chien rouge
chien → Le gros chien rouge
rouge → Le gros chien rouge

$$\begin{bmatrix} 0.16 \\ 0.09 \\ 0.05 \\ 0.40 \end{bmatrix} \begin{bmatrix} 0.1 \\ 0.9 \\ 0.55 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



Automatic Exam Marking Project

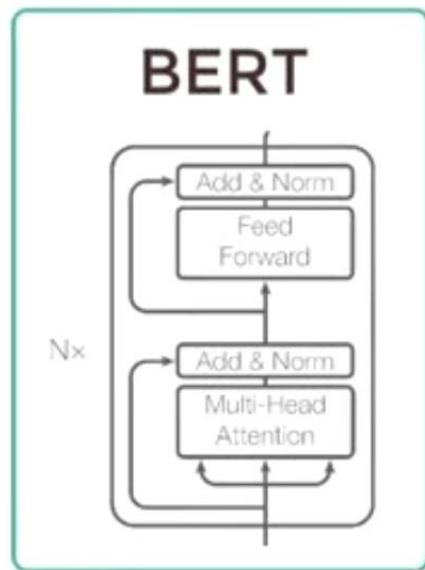
Phase 1: Exam Management Product

Product ML Engine

**PART 3 : Bert Technical Detail
&
Fine Tuning Examples**

Bidirectional Encoder Representation from Transformers

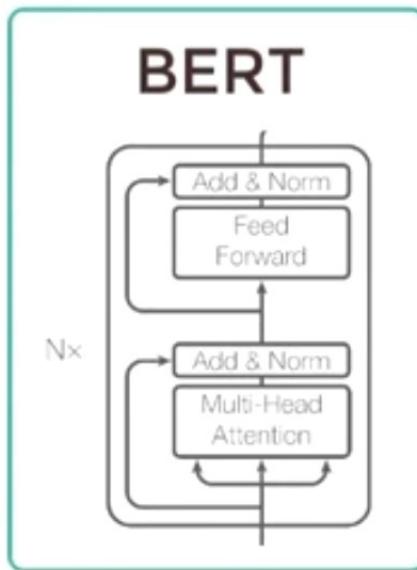
Problems to Solve



- Neural Machine Translation
- Question Answering
- Sentiment Analysis
- Text summarization

Bidirectional Encoder Representation from Transformers

Problems to Solve



3 passes of explanation

- Neural Machine Translation
- Question Answering
- Sentiment Analysis
- Text summarization

Needs Language understanding

How to solve Problems (BERT Training)

- Pretrain BERT to understand language
- Fine tune BERT to learn specific task

Bidirectional Encoder Representation from Transformers

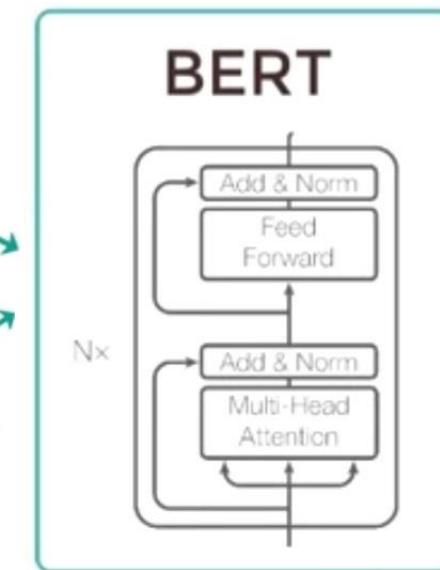
Pretraining (Pass 1) : “What is language? What is context?”

Masked Language Model (MLM)

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

Next Sentence Prediction (NSP)

A: Ajay is a cool dude.
B: He lives in Ohio



[MASK1] = quick
[MASK2] = jumped

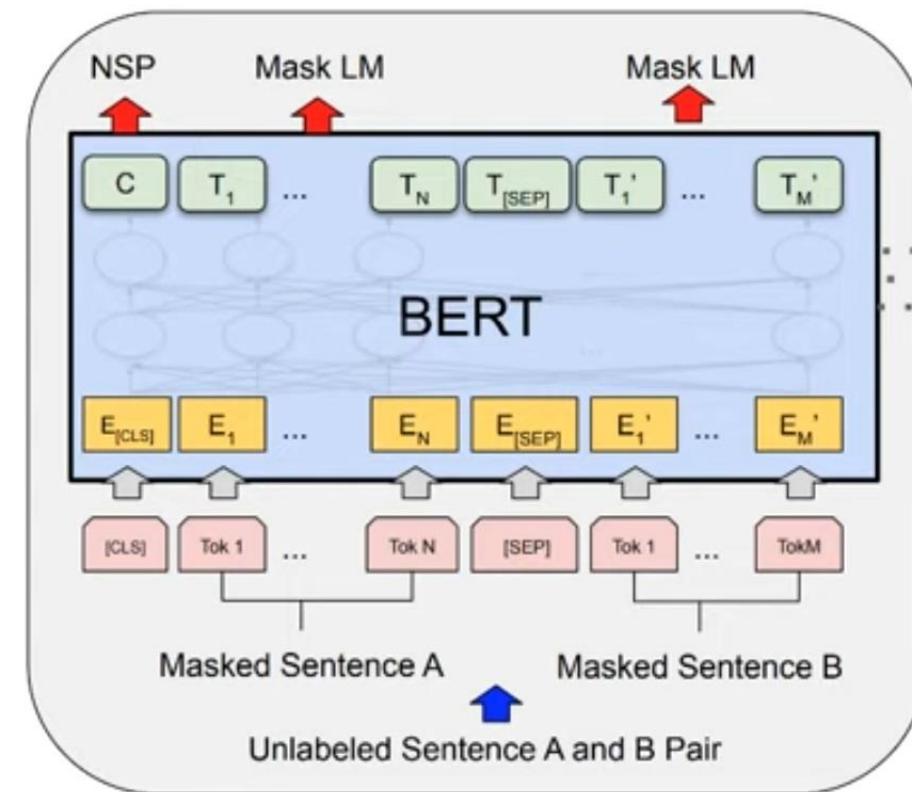
Yes. Sentence B follows sentence A

Bidirectional Encoder Representation from Transformers

Pretraining (Pass 2)

Problems to train on simultaneously:

1. Masked Language Modeling (Mask LM)
2. Next Sentence Prediction (NSP)



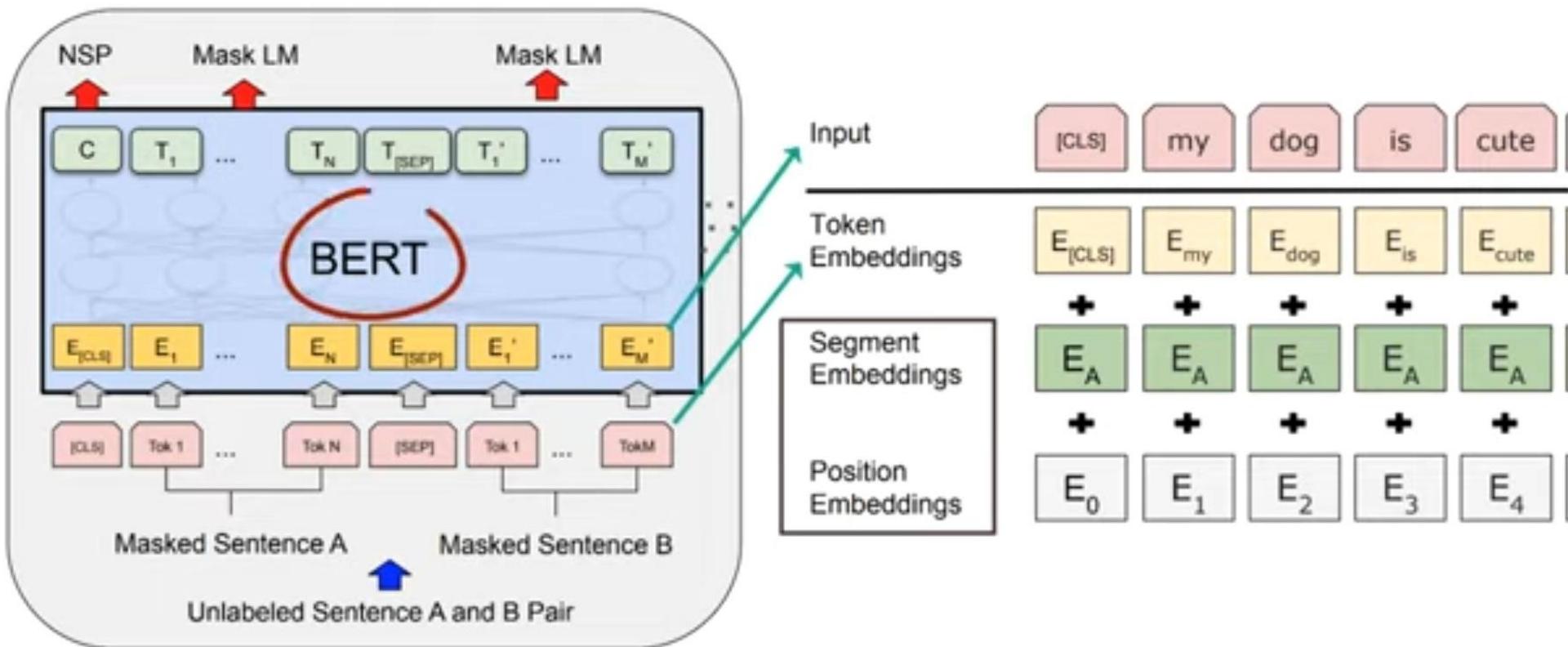
Source: BERT: Pre-training of deep bidirectional Transformers for language understanding (Devlin et al., 2019)

Masked LM

- Problem: Mask token never seen at fine-tuning
- Solution: 15% of the words to predict, but don't replace with [MASK] 100% of the time. Instead:
 - 80% of the time, replace with [MASK]
went to the store → went to the [MASK]
 - 10% of the time, replace random word
went to the store → went to the running
 - 10% of the time, keep same
went to the store → went to the store

Bidirectional Encoder Representation from Transformers

Pretraining (Summary)

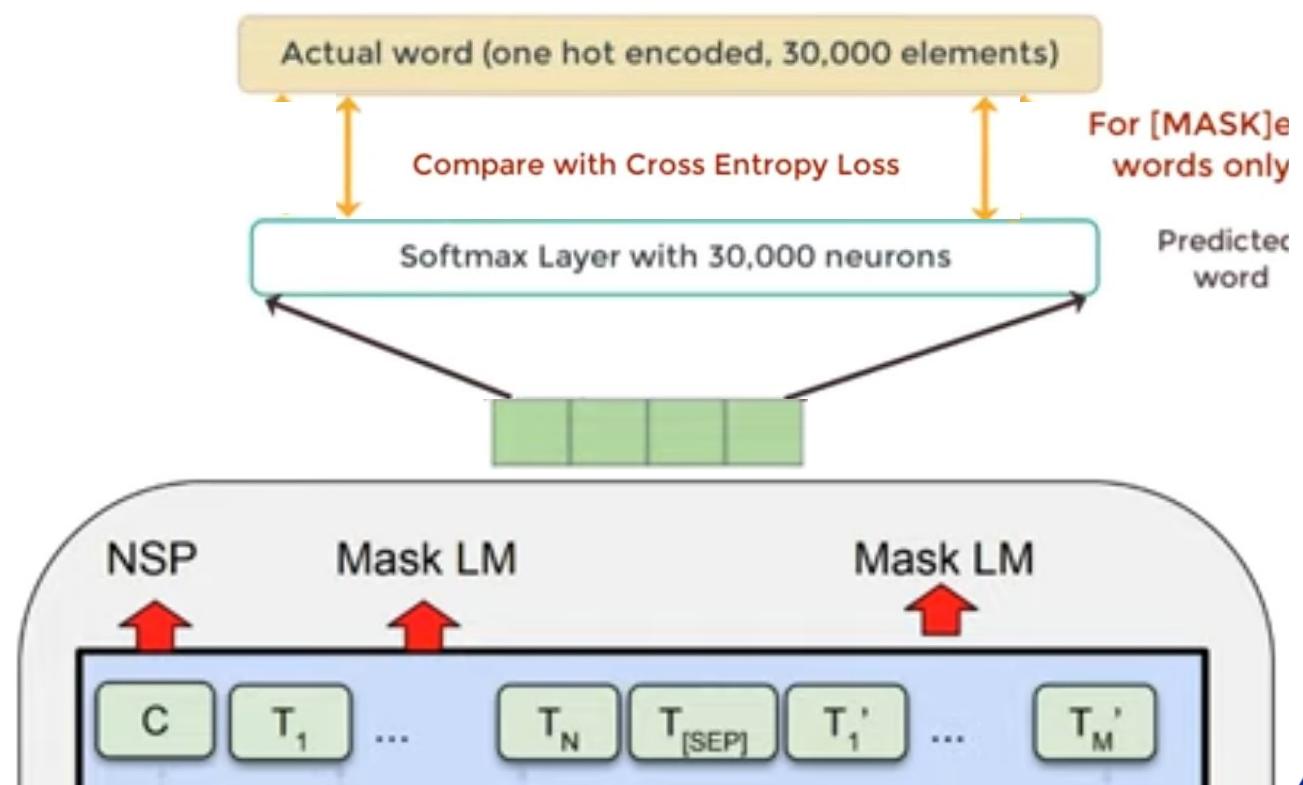


Bidirectional Encoder Representation from Transformers

Pretraining (Pass 3)

Word vectors T_i have the same size.

Word vectors T_i are generated simultaneously



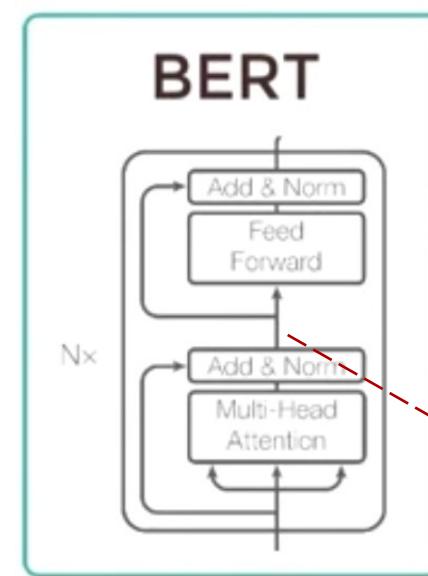
Bidirectional Encoder Representation from Transformers

Fine Tuning (Pass 1): “How to use language for specific task?”

Fine tuned Q & A

Question

Passage

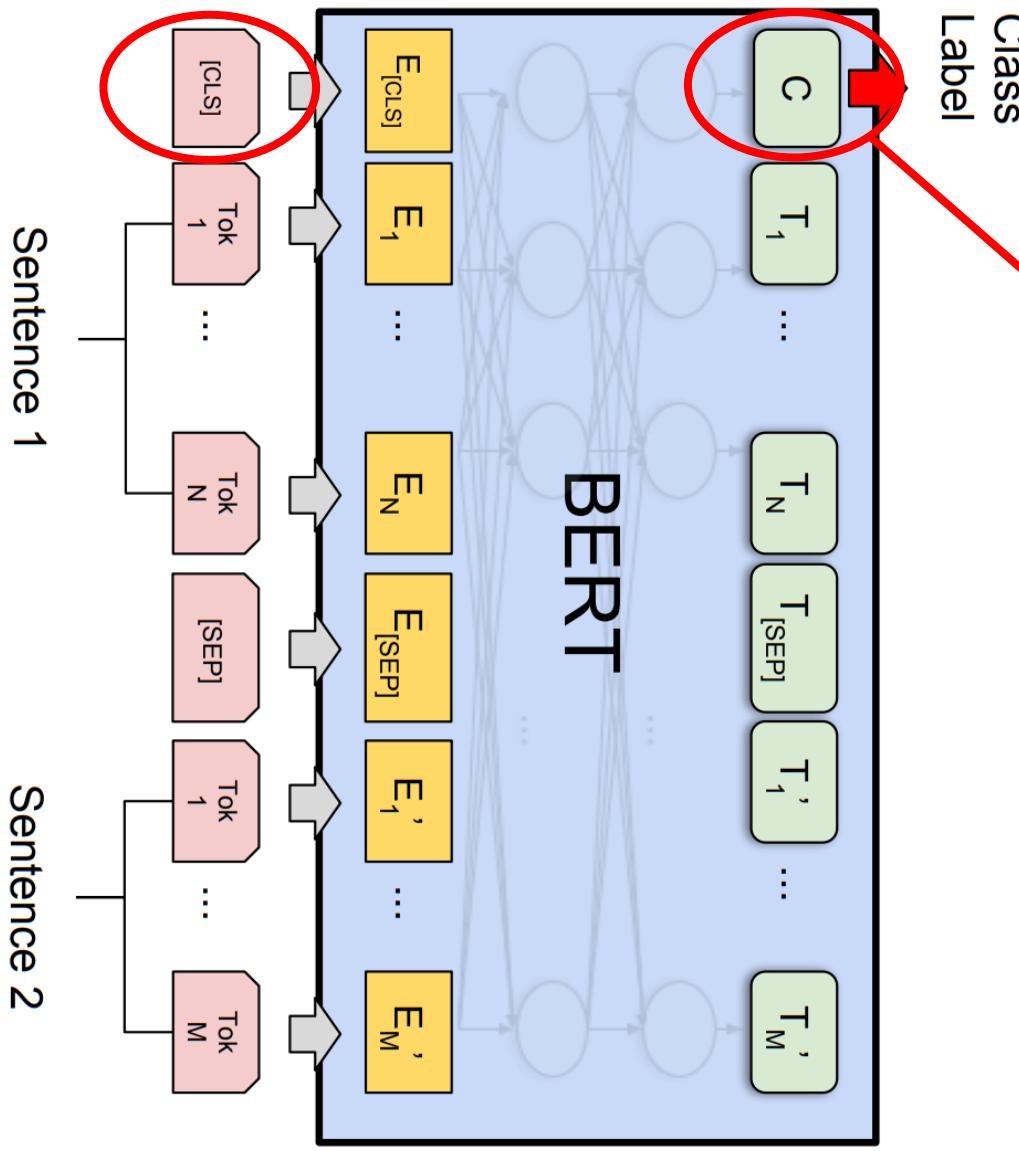


Learned From
scratch

Answer

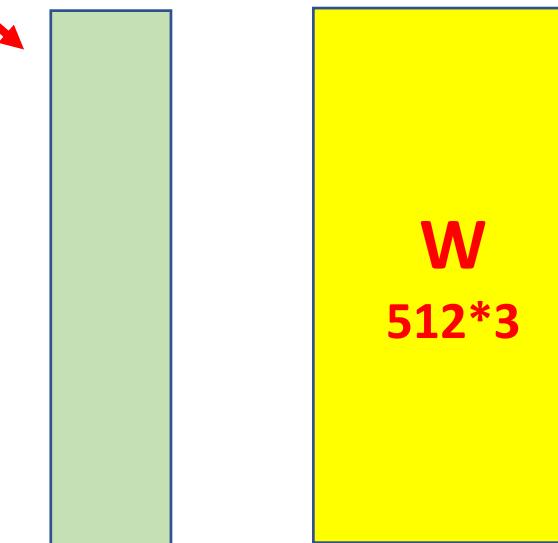
Fine Tuned

Bert Tuning For



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

Given a pair of sentences, predict the second is an entailment, contradiction or neutral with the first



entailment

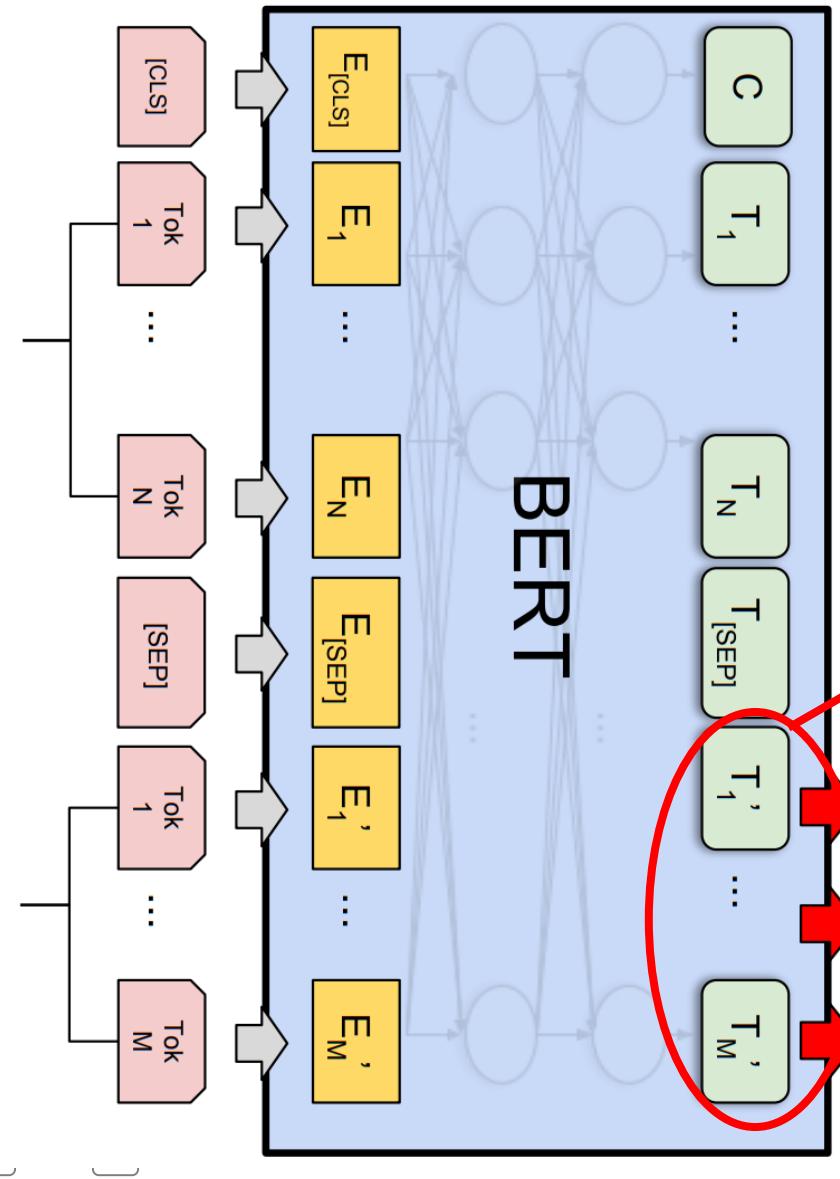
Contra
diction

neutral

3 Label
output

Bert Tuning For

Question



(c) Question Answering Tasks: SQuAD v1.1

- Input Question:

Where do water droplets collide with ice crystals to form precipitation?

- Input Paragraph:

... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. ...

- Output Answer:

within a cloud

Probability of each output token for Start/End/Spam

Compute to produce inner Product

QS

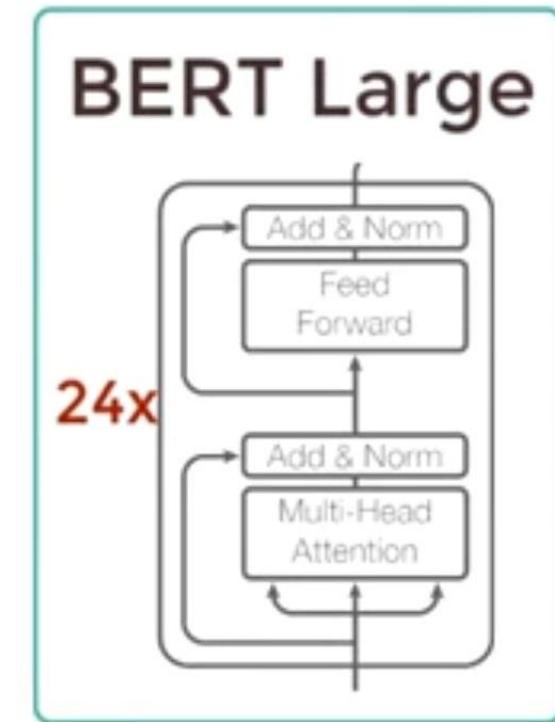
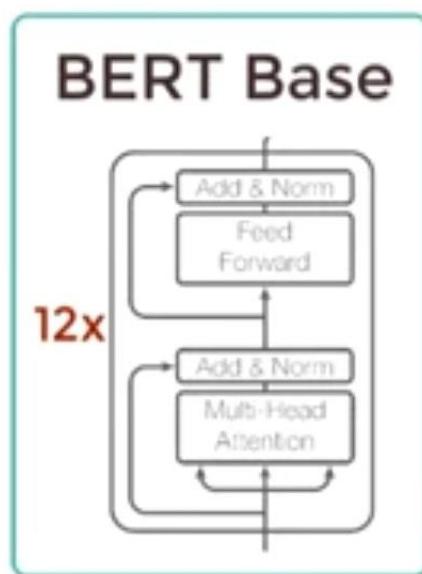
QE

Start/End Span
512* Size of Paragraph

3* Size of Paragraph

Bidirectional Encoder Representation from Transformers

Performance



340M Parameters

Model Details

- Data: Wikipedia (2.5B words) + BookCorpus (800M words)
- Batch Size: 131,072 words (1024 sequences * 128 length or 256 sequences * 512 length)
- Training Time: 1M steps (~40 epochs)
- Optimizer: AdamW, 1e-4 learning rate, linear decay
- BERT-Base: 12-layer, 768-hidden, 12-head
- BERT-Large: 24-layer, 1024-hidden, 16-head
- Trained on 4x4 or 8x8 TPU slice for 4 days



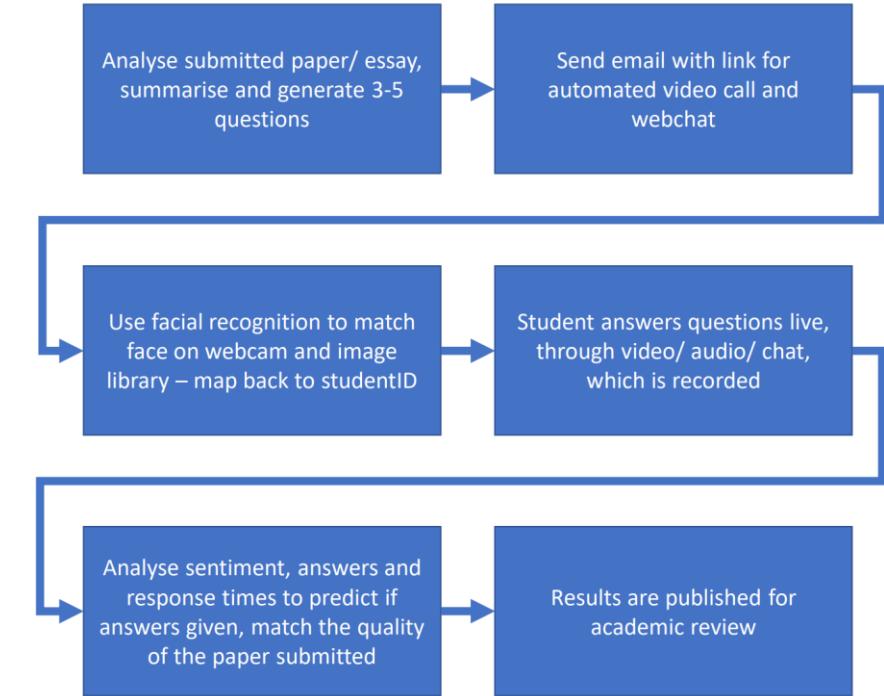
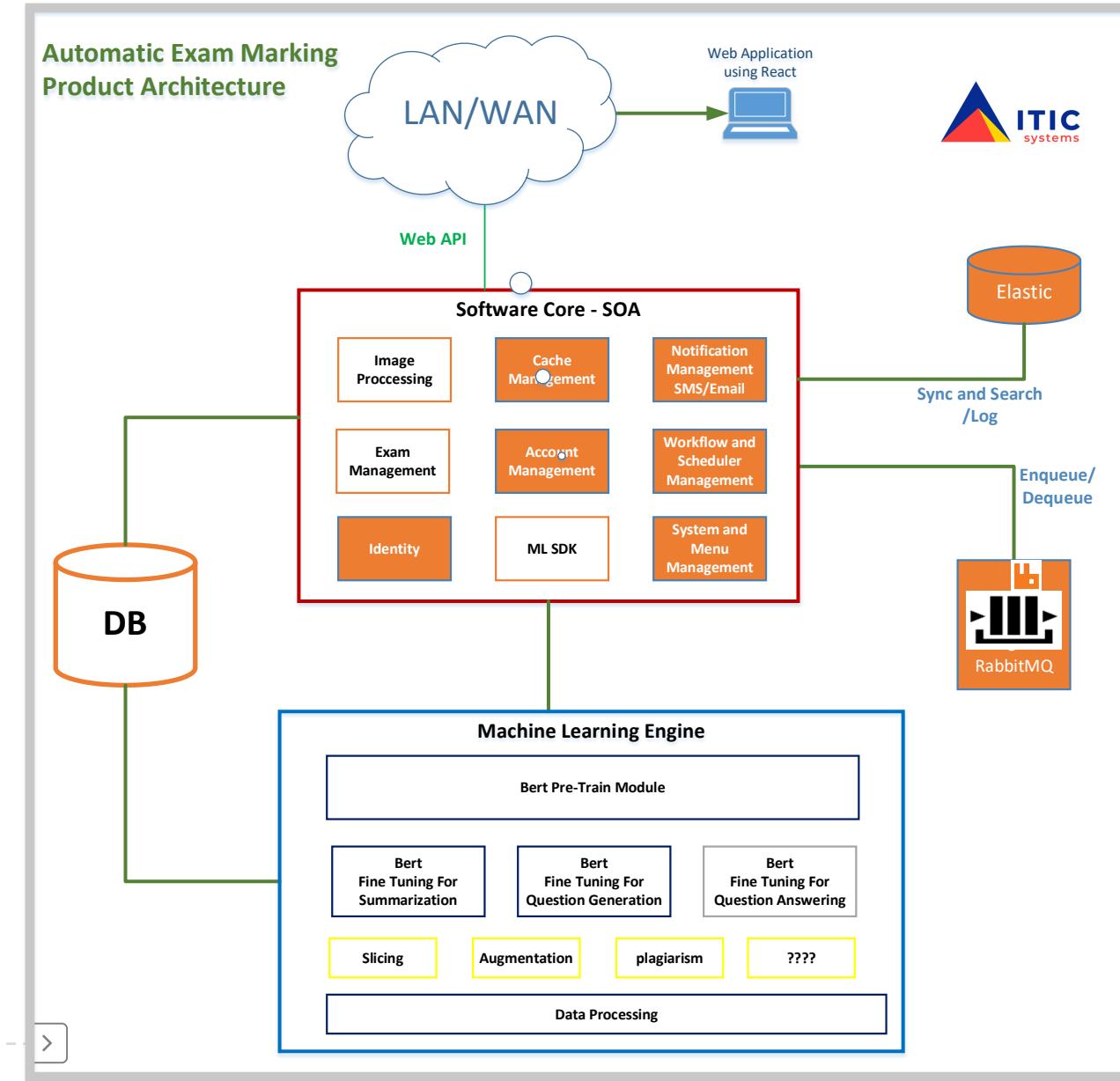
Automatic Exam Marking Project

Phase 1: Exam Management Product

Product ML Engine

PART 4 : New Task
&
Code Analysis Definition

Task 4: Existing GAN For Generation Essay DataSet



Task 5: Bert Pre-traning and Fine Tuning Code Analysis

Expected Task Outputs

Providing Code Analysis Document
For Bert Pre-training

Python Code:

[queries. https://github.com/jina-ai/clip-as-service](https://github.com/jina-ai/clip-as-service)

Providing Code Analysis Document
For Existing Bert Fine Tuning Phases for Specific Task

**Text Summarization
Question Generating
Answers Marking**

Task 4: Existing GAN For Generation Essay DataSet

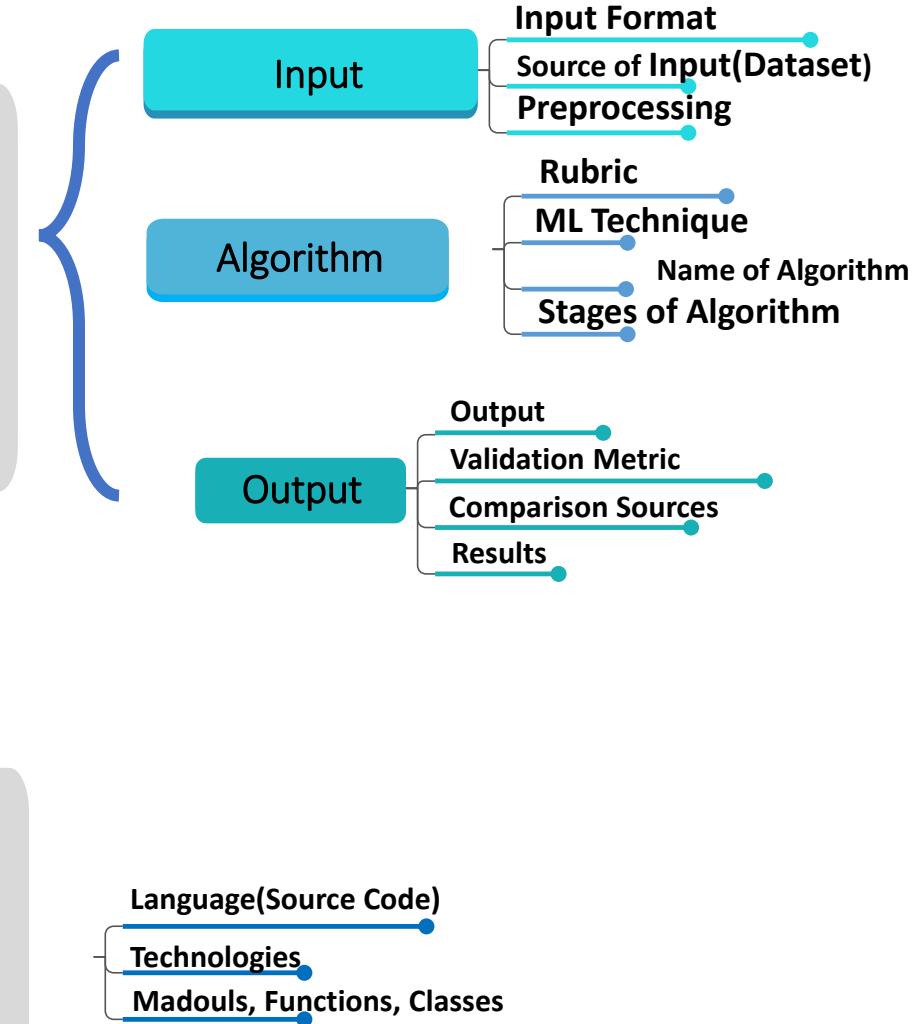
Research & Code Analysis

Park, Y. H., Choi, Y. S., Park, C. Y., & Lee, K. J. (2022). EssayGAN: Essay Data Augmentation Based on Generative Adversarial Networks for Automated Essay Scoring. *Applied Sciences*, 12(12), 5803.

Expected Task Outputs

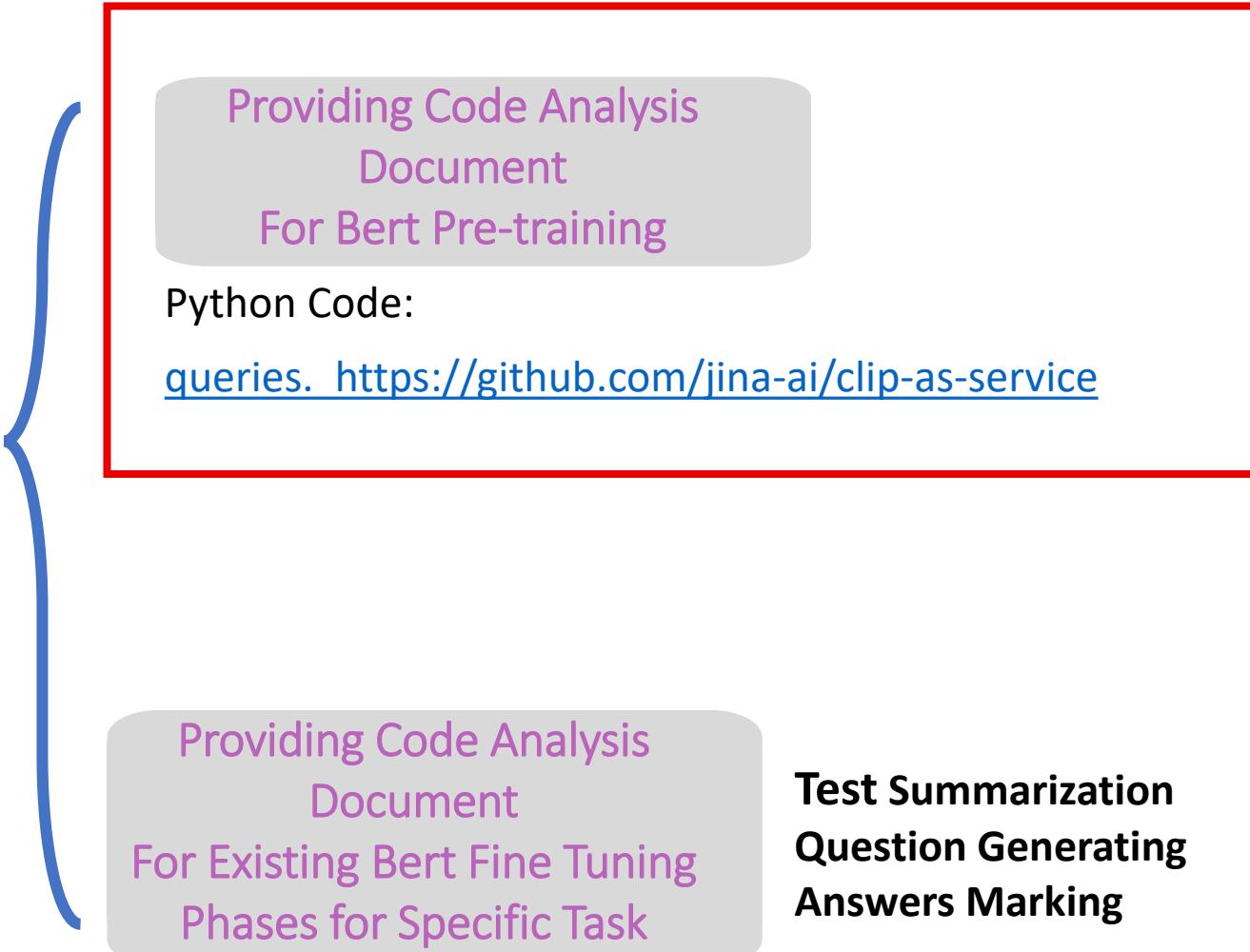
Provide a presentation about details of the using ML model

Providing Code Analysis Document



Task 5 & 7 & 8 & 9: Bert Pre-traing and Fine Tuning Code Analysis

Expected Task Outputs

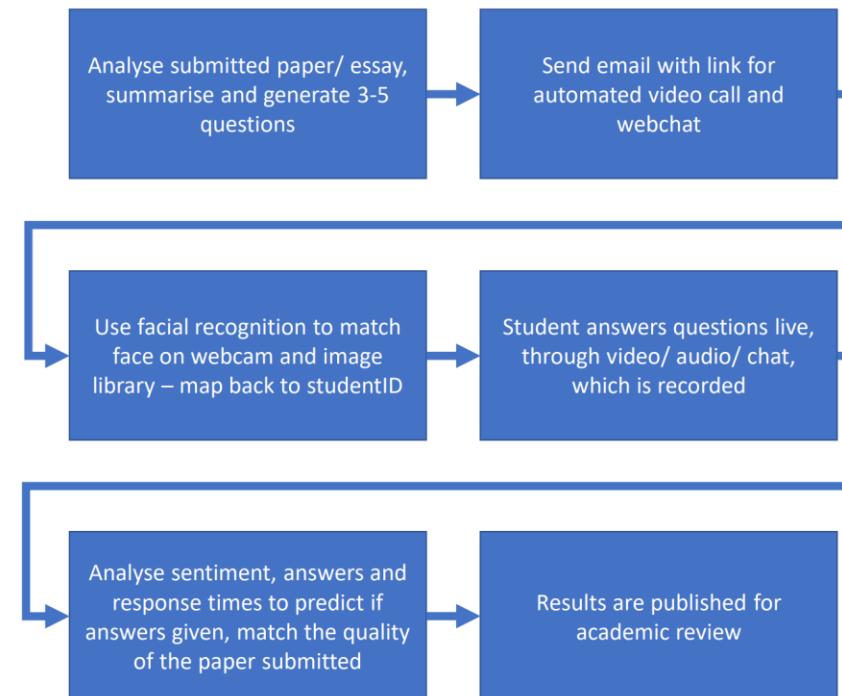


Code Analysis – Required Output

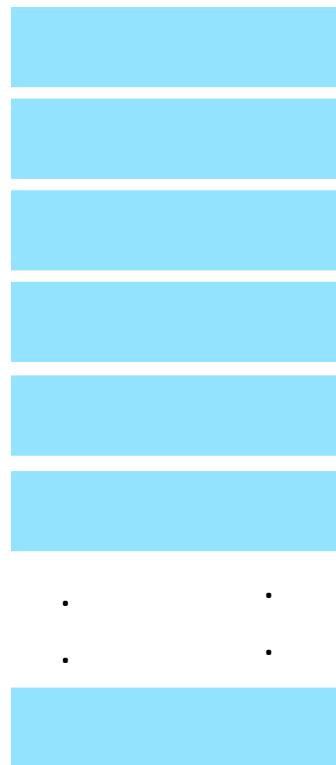
ITEM	Description of usability	How We can Use It in Our problem	Using in which Part of Code	Used Item	Variable Description	Parameters & Variables	Output	Input	Description per Line
Function A				B, C					
Classes B									
Libraries C									
Data Base					Overall Data Model				
					Data Scale				
					Data Type				
					Relations & Keys				

Task 10: Analyzing Existing methods for plagiarism

- Find Similarities between existing source of question



my	0 0 1.....00
name	0 01....00
is	0 0 ..1.....00
Masoud	0 01...00
<PAD>	0 0 1.....00
<PAD>	0 01....00
.	.
.	.
<PAD>	0 0 1.....00



+



=



Max Sequence Length
X
Max Sequence Length

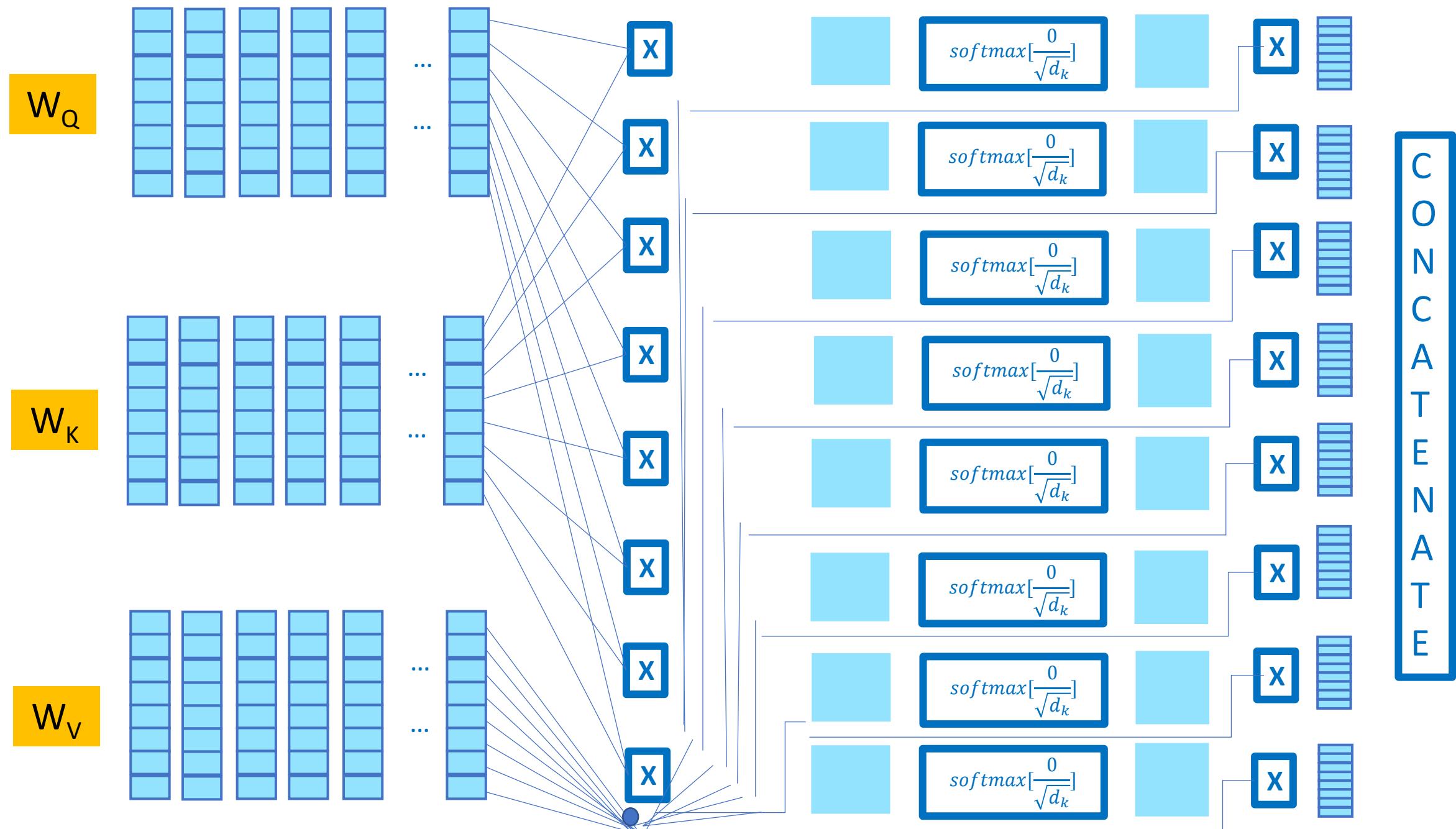
Max Sequence Length
X
512

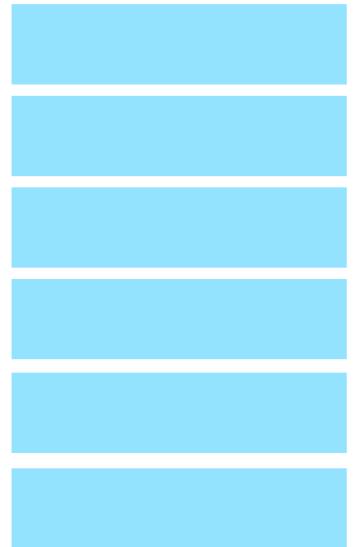
Max Sequence Length
X
512

Max Sequence Length
X
512

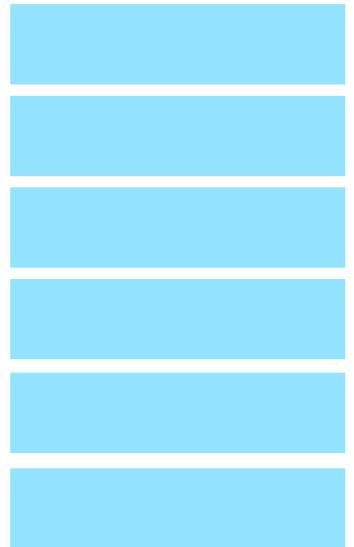
$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{\text{model}}})$$

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

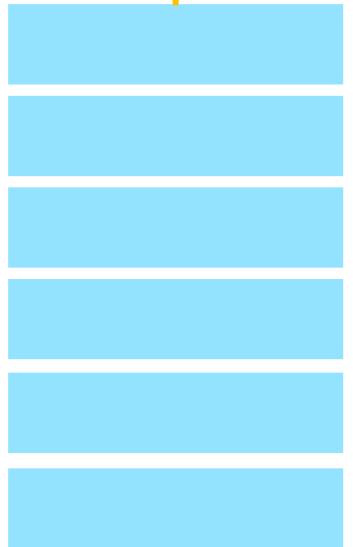




+



Layer Normalization
 $\gamma_1, \beta_1, \gamma_2, \beta_2 \dots$

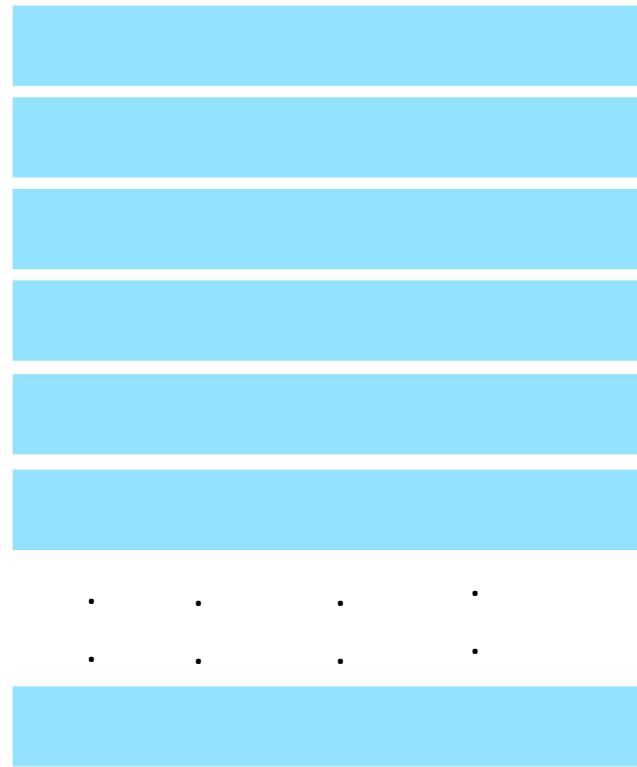


Linear
+
ReLU
+
Dropout

Max Sequence Length
 X
512

Max Sequence Length
 X
512

Max Sequence Length
 X
512



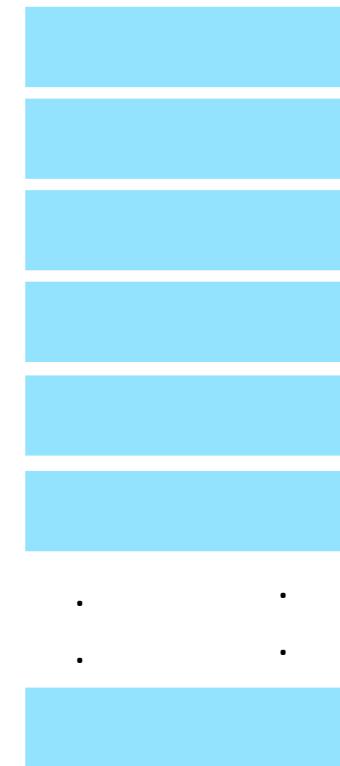
Max Sequence Length
 x
1024



Max Sequence Length
 x
512



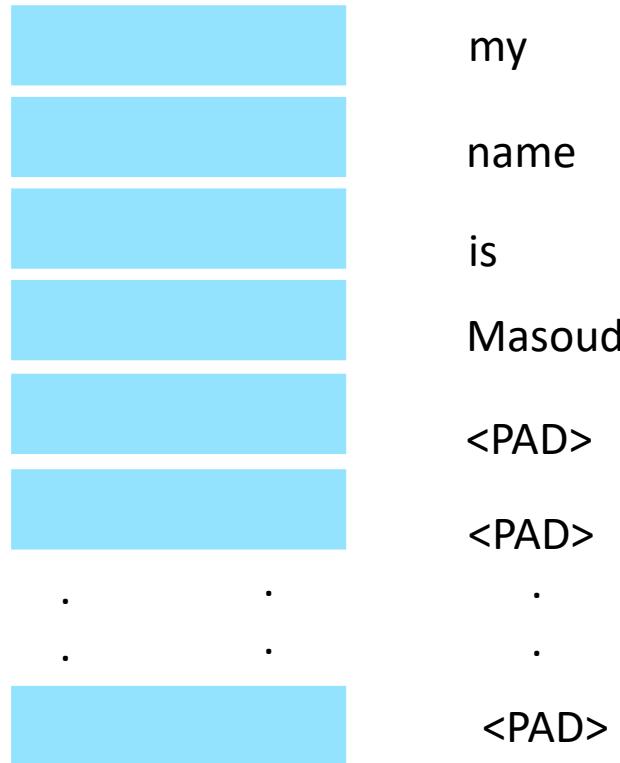
+



Max Sequence Length
 x
512

Layer Normalization

$\gamma_1, \beta_1, \gamma_2, \beta_2 \dots$



Max Sequence Length
X
512



Automatic Exam Marking Project

Phase 1: Exam Management Product

Product ML Engine

PART 5 : References

References

Bert Overview

<https://www.youtube.com/watch?v=xI0HHN5XKDo>

Bert Paper Overview

<https://www.youtube.com/watch?v=-9evrZnBorM>

Transformer Neural Networks - EXPLAINED! (Attention is all you need)

<https://www.youtube.com/watch?v=TQQIZhbC5ps>

Stanford Bert

<https://www.youtube.com/watch?v=knTc-NQSjKA>

Stanford Transfomer and self attention

<https://www.youtube.com/watch?v=5vcj8kSwBCY>

Self attention in Transformer with code

<https://www.youtube.com/watch?v=QCJQG4DuHT0&list=PLTl9hO2Oobd97qfWC40gOSU8C0iu0m2l4>

Multi Head Attention in Transformer Neural Networks with Code!

<https://www.youtube.com/watch?v=HQn1QKQYXVg>

References

Architecture connection & Normalization layer

<https://www.youtube.com/watch?v=G45TuC6zRf4&list=PLTl9hO2Oobd97qfWC40gOSU8C0iu0m2l4&index=5>

Positional Encoding in Transformer Neural Networks Explained

<https://www.youtube.com/watch?v=ZMxVe-HK174&list=PLTl9hO2Oobd97qfWC40gOSU8C0iu0m2l4&index=4>

Blowing up the Transformer Encoder!

<https://www.youtube.com/watch?v=Qwfu0NhjbkI&list=PLTl9hO2Oobd97qfWC40gOSU8C0iu0m2l4>

Blowing up Transformer Decoder architecture

<https://www.youtube.com/watch?v=ekg-hoob0SM&list=PLTl9hO2Oobd97qfWC40gOSU8C0iu0m2l4&index=8>

NLP Project in 20 Minutes Using BERT

<https://www.youtube.com/watch?v=iiwEW-sg9KE>

Sample Topic Modeling problem with using bert

<https://www.youtube.com/watch?v=TLPmlVeEf1k>