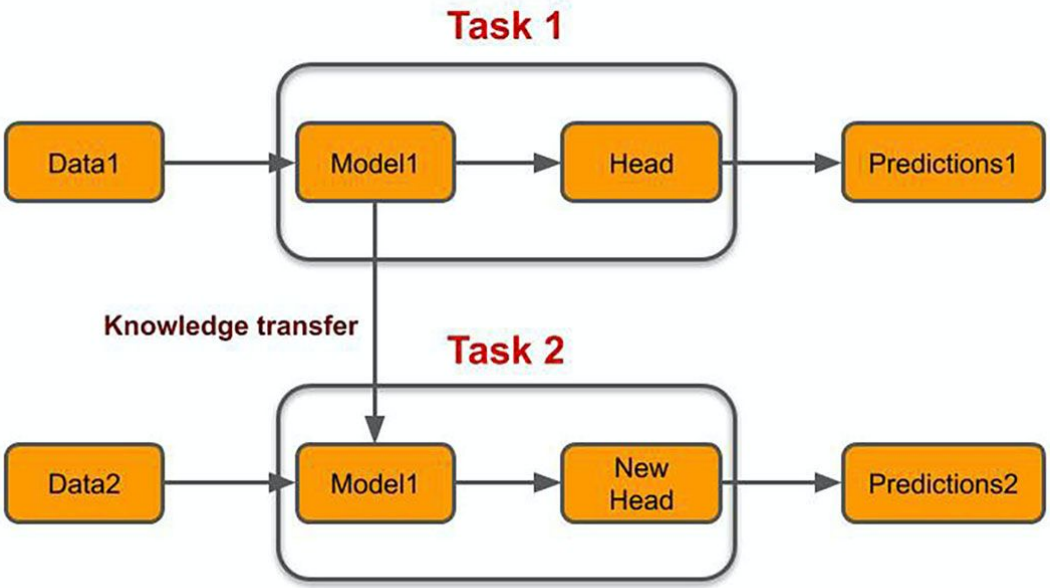
Transformers+

Based on:
https://jalammar.github.io/illustrated-bert/
https://jalammar.github.io/how-gpt3-works-visualizations-animations/

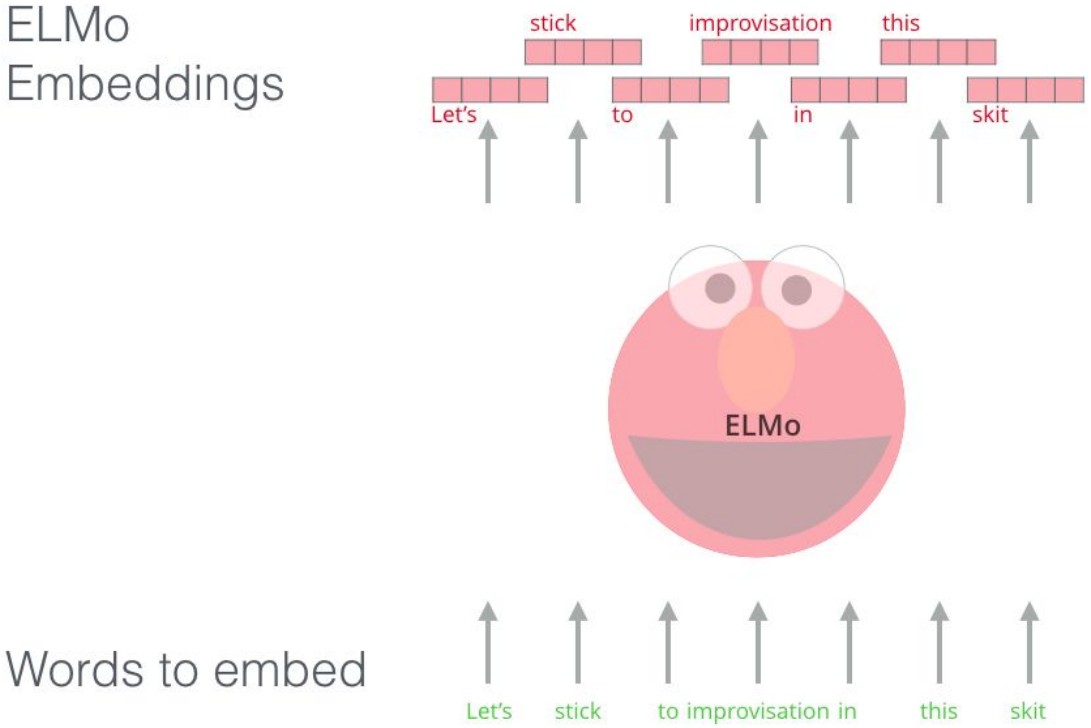
Timeline of some major projects

Transfer Learning

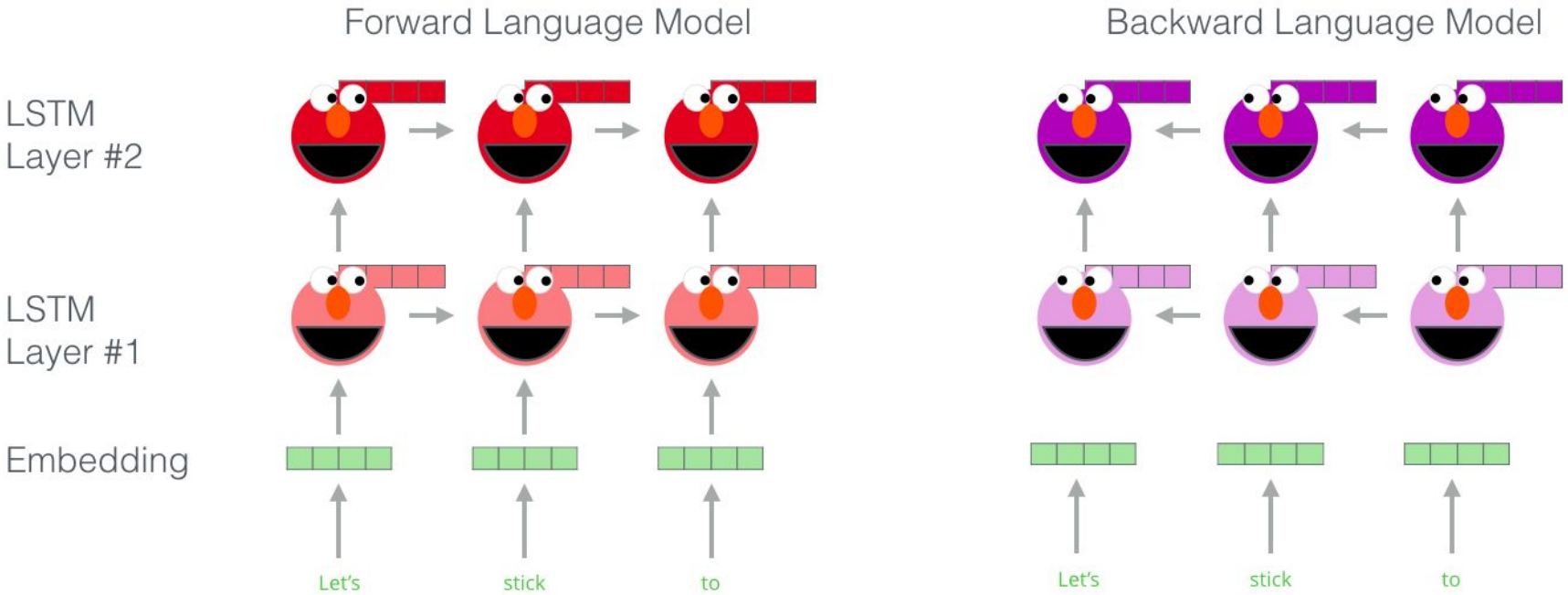


ELMO: Embeddings from Language Models

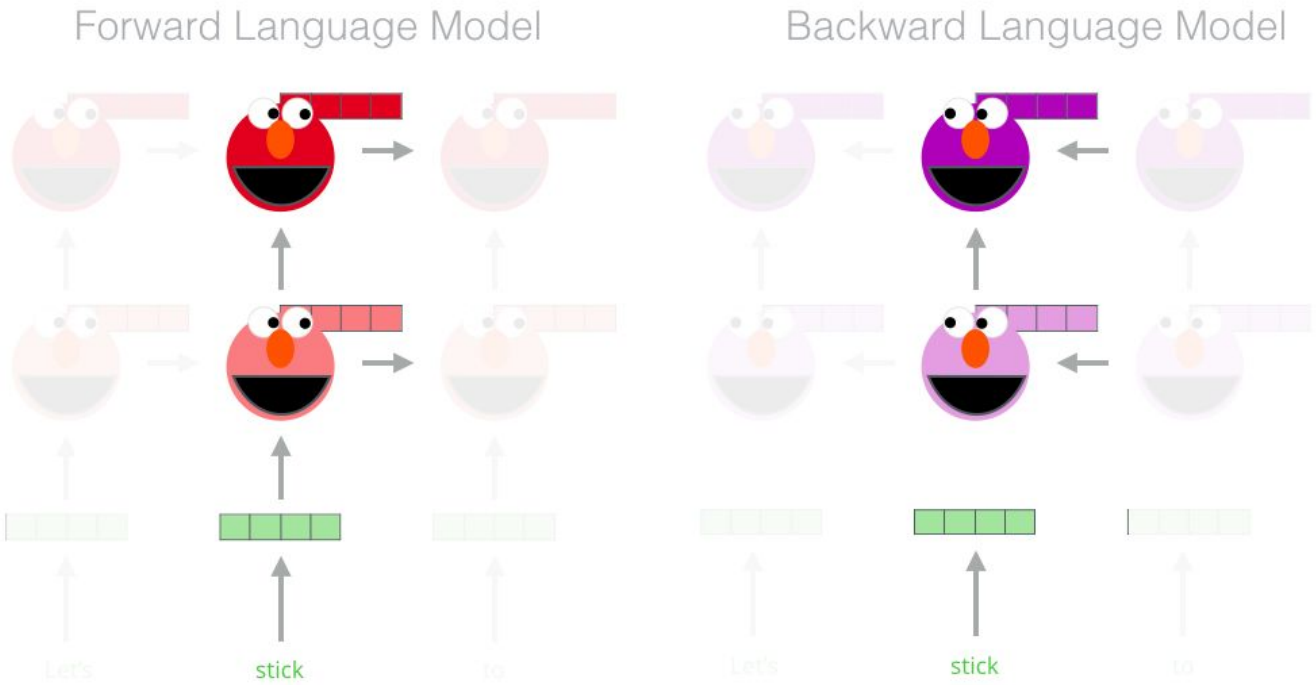
ELMO


ELMo
Embeddings
stick
improvisation
this
Let's
to
in
skit
ELMo
Words to embed
Let's
stick
to improvisation in
this
skit

ELMO


Forward Language Model
Backward Language Model
LSTM
Layer #2
LSTM
Layer #1
Embedding
Let's
stick
to
Let's
stick
to

ELMO

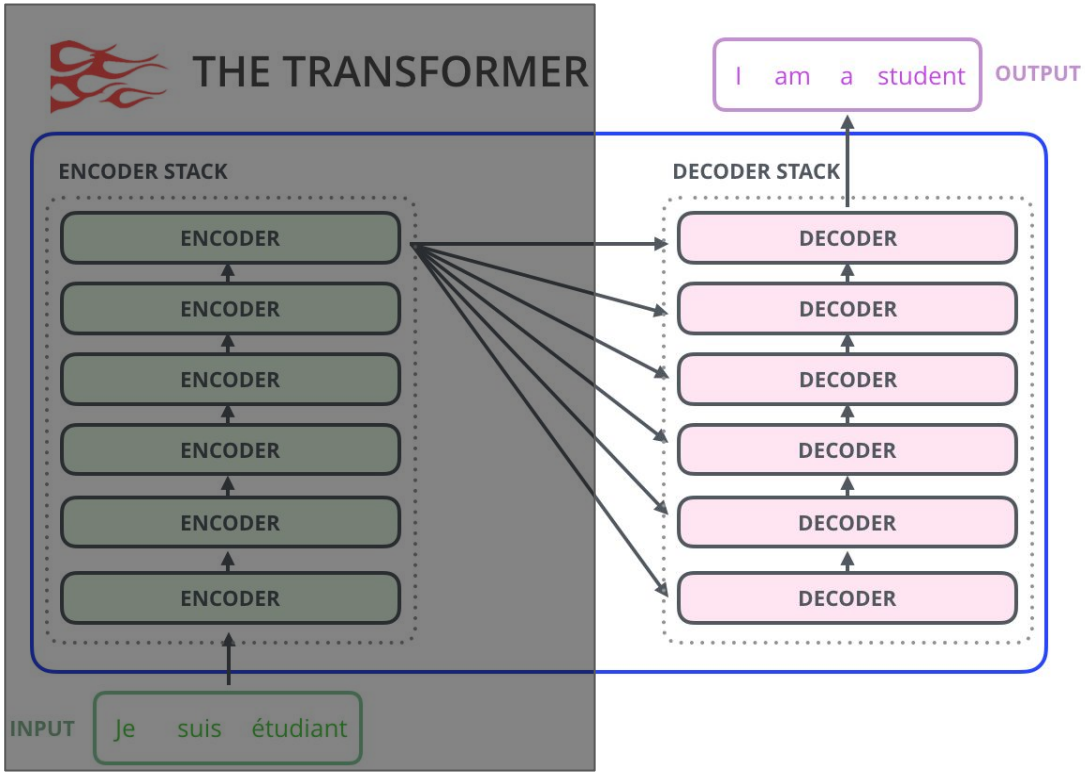

Forward Language Model
Backward Language Model
stick
stick

ELMO

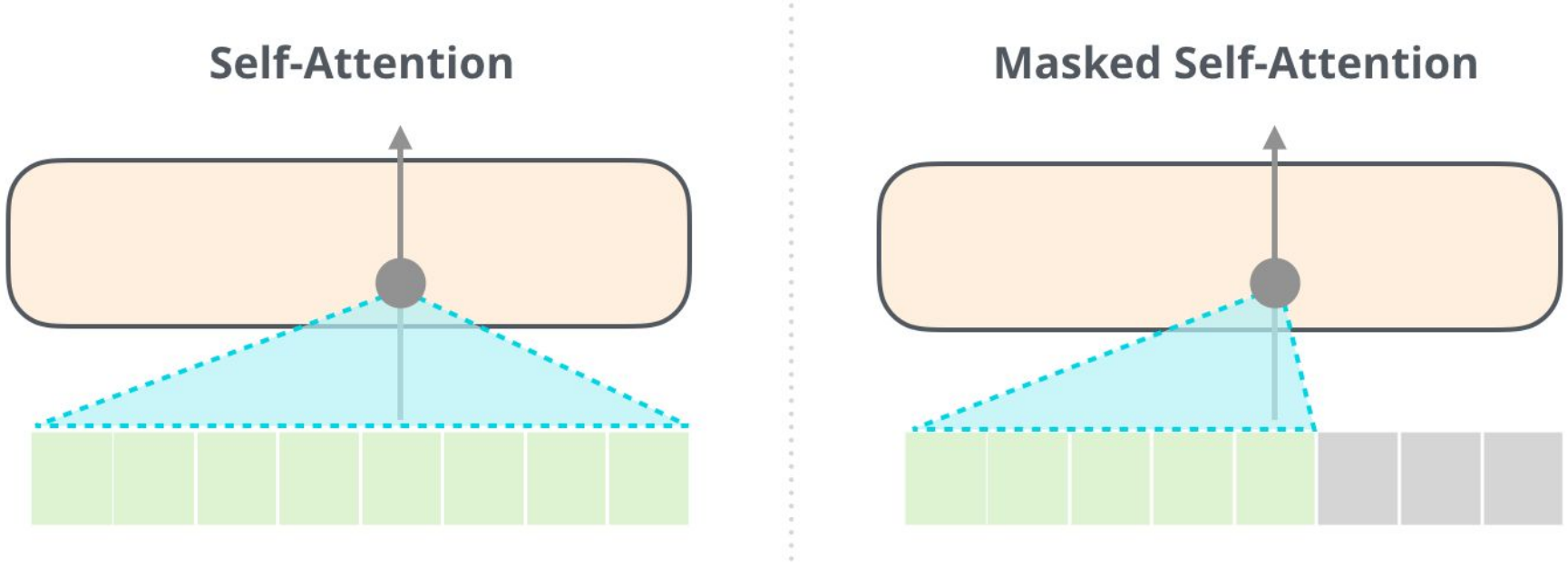
Task	Previous SOTA		Our baseline	ELMo + baseline	Increase (absolute/ relative)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

Paper link
Elmo weights

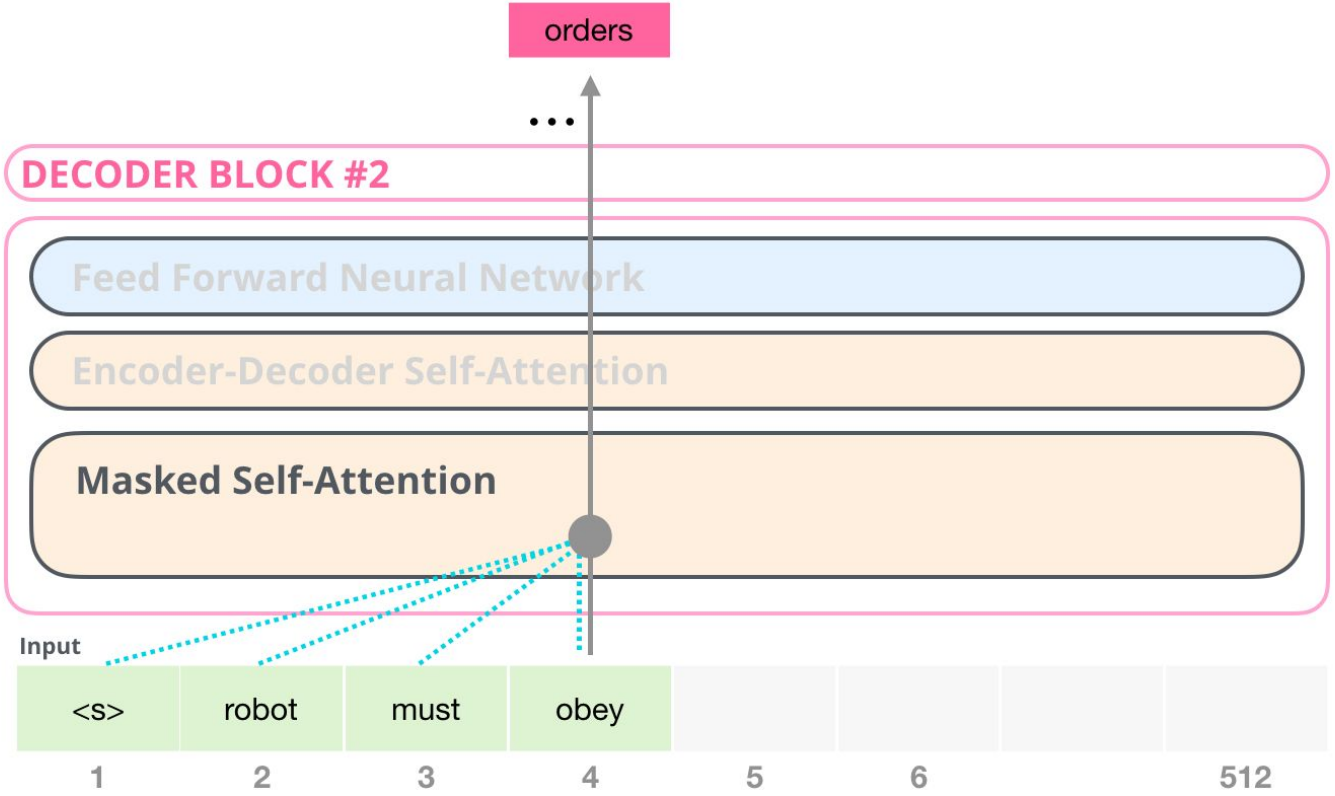
GPT: Generative Pretrained Transformers



GPT



GPT


orders
...
DECODER BLOCK #2
Feed Forward Neural Network
Encoder-Decoder Self-Attention
Masked Self-Attention
Input
<s>
robot
must
obey
1
2
3
4
5
6
512

GPT

	Features				Labels
	position: 1	2	3	4	
Example: 1	robot	must	obey	orders	must
2	robot	must	obey	orders	obey
3	robot	must	obey	orders	orders
4	robot	must	obey	orders	<eos>

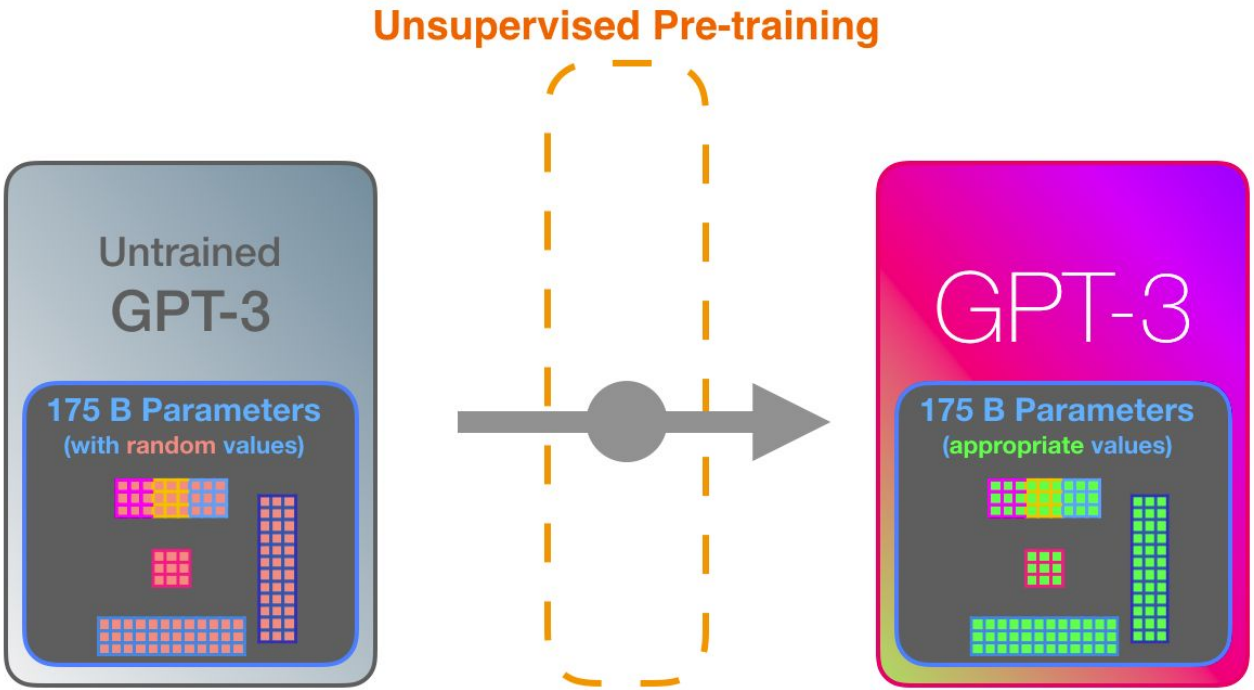
GPT

Text: Second Law of Robotics: A robot must obey the orders given it by human beings

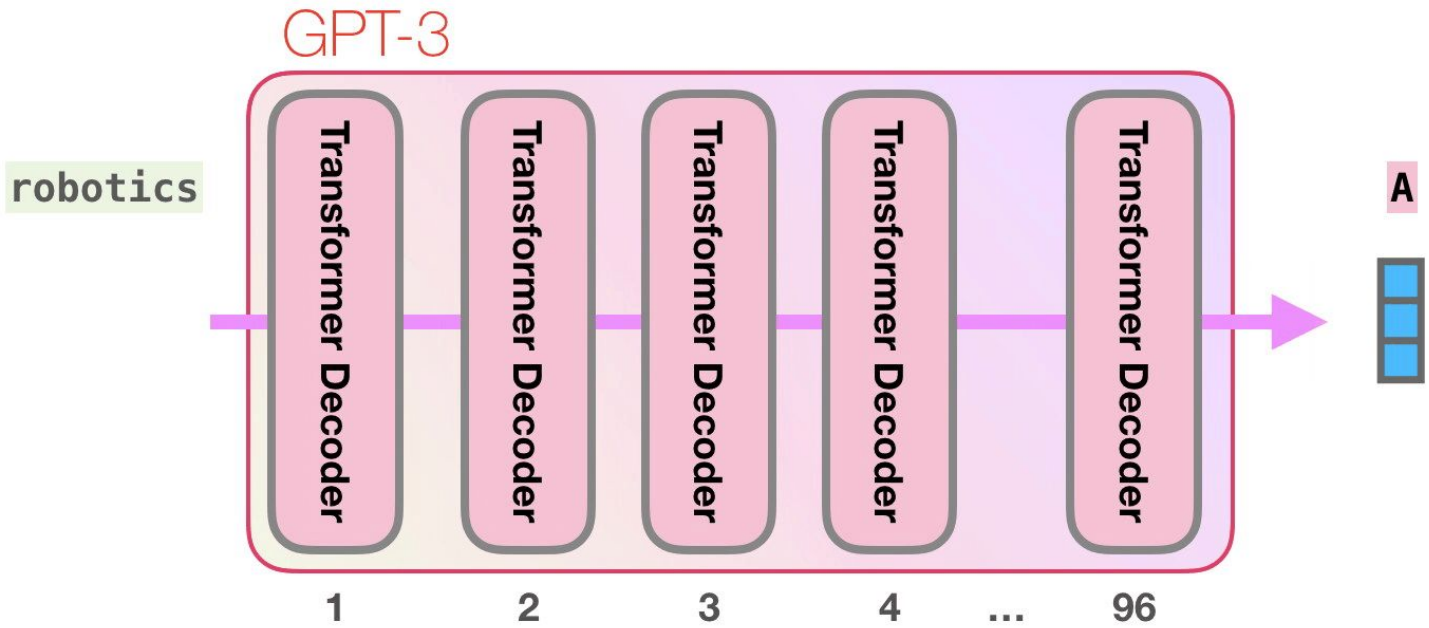
Generated training examples

Example #	Input (features)	Correct output (labels)
1	Second law of robotics :	a
2	Second law of robotics : a	robot
3	Second law of robotics : a robot	must
...		

GPT


Unsupervised Pre-training
Untrained
GPT-3
175 B Parameters
(with random values)
GPT-3
175 B Parameters
(appropriate values)

GPT

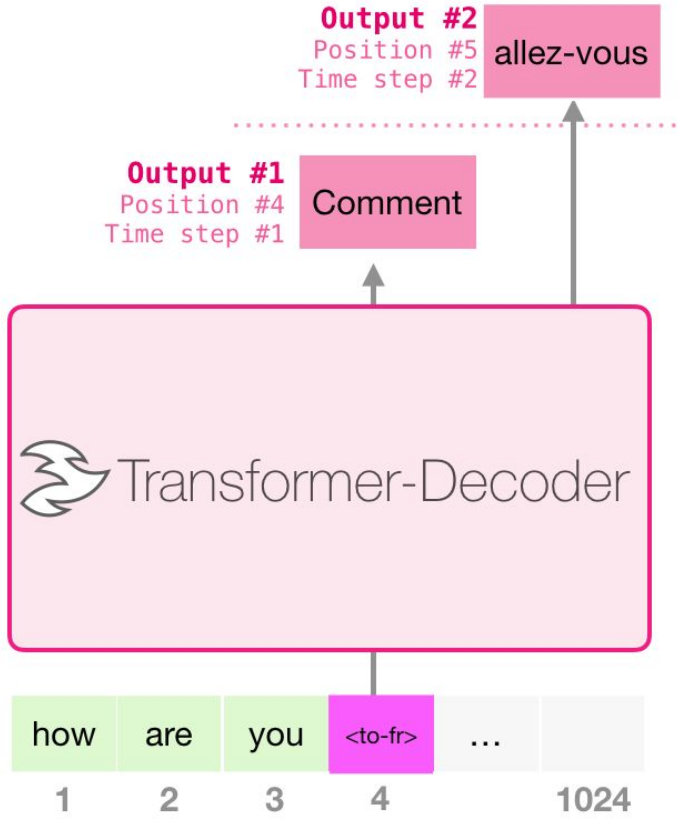

GPT-3
robotics
Transformer Decoder
Transformer Decoder
Transformer Decoder
Transformer Decoder
Transformer Decoder
A
1
2
3
4
...
96

Example: Machine Translation

An encoder is not required to conduct translation

Training Dataset

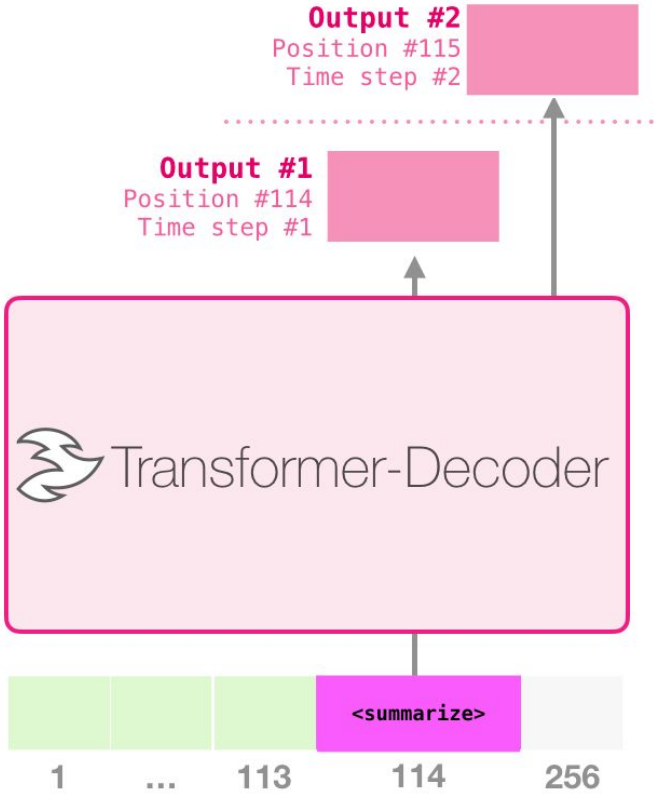
I	am	a	student	<to-fr>	je	suis	étudiant
let	them	eat	cake	<to-fr>	Qu'ils	mangent	de
good	morning	<to-fr>	Bonjour				



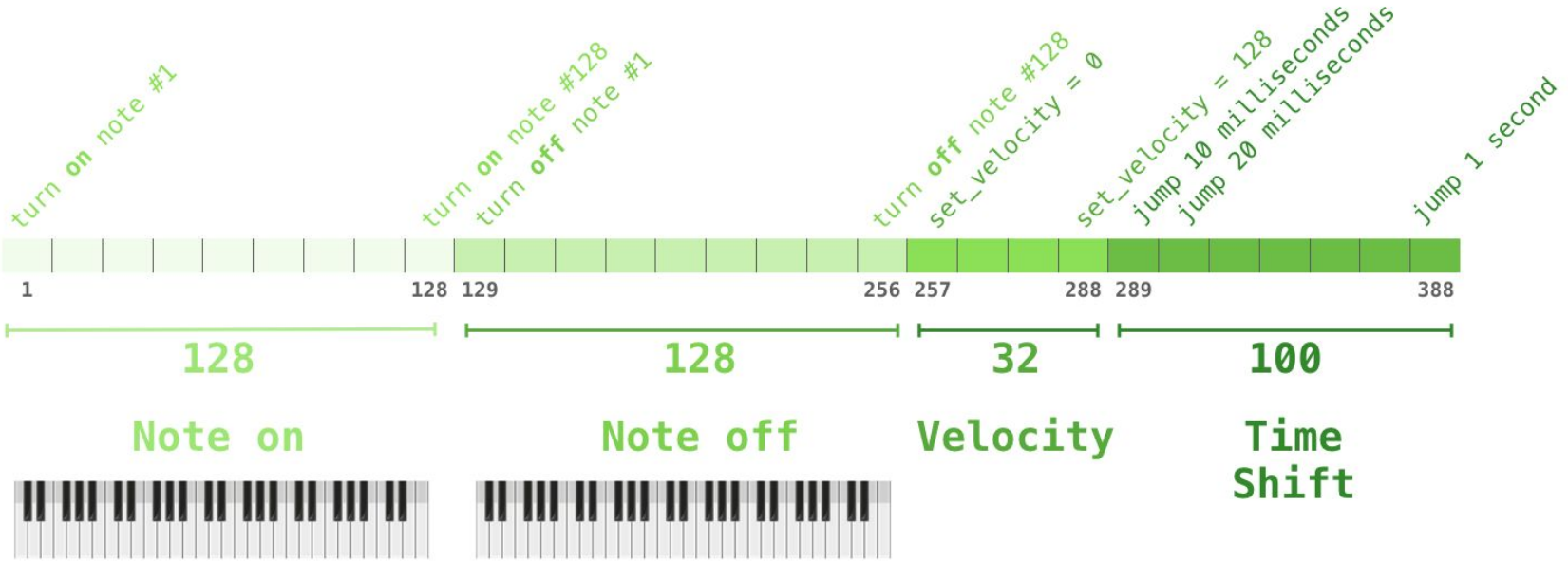
Examples: Summarization

Training Dataset

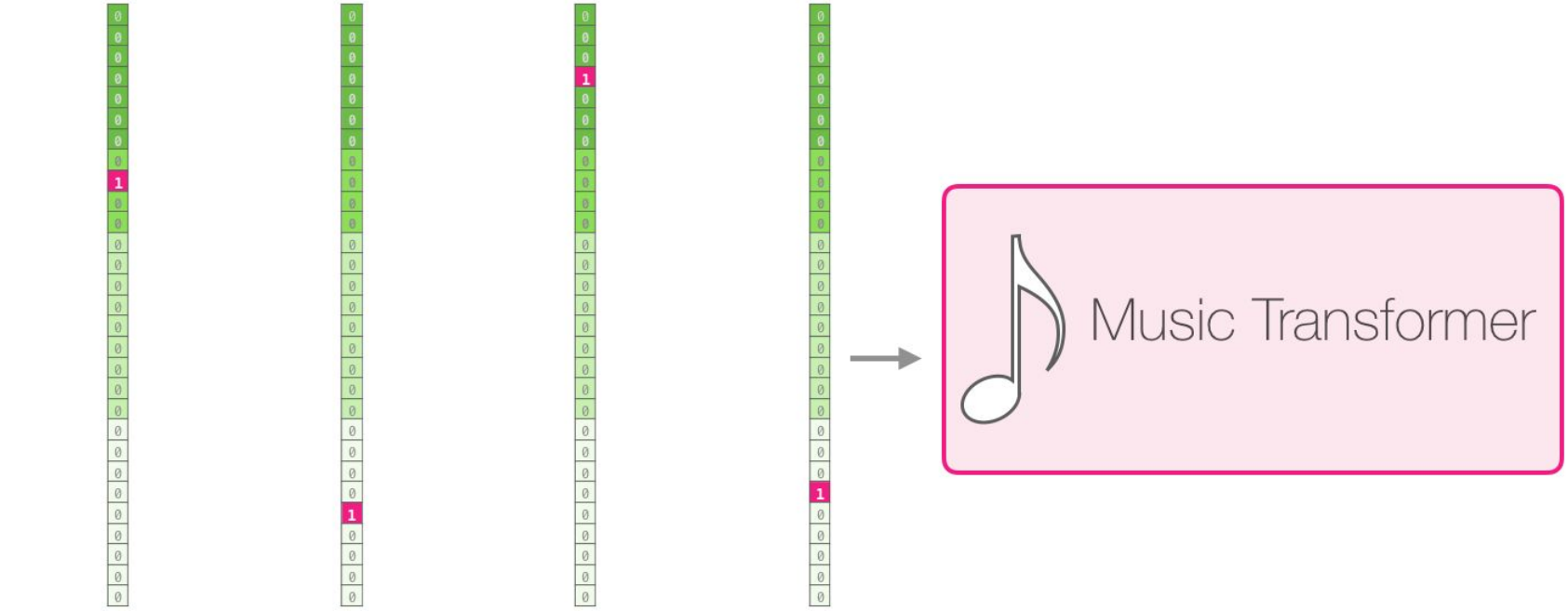
Article #1 tokens		<summarize>	Article #1 Summary
Article #2 tokens	<summarize>	Article #2 Summary	padding
Article #3 tokens		<summarize>	Article #3 Summary



Examples: Music Generation

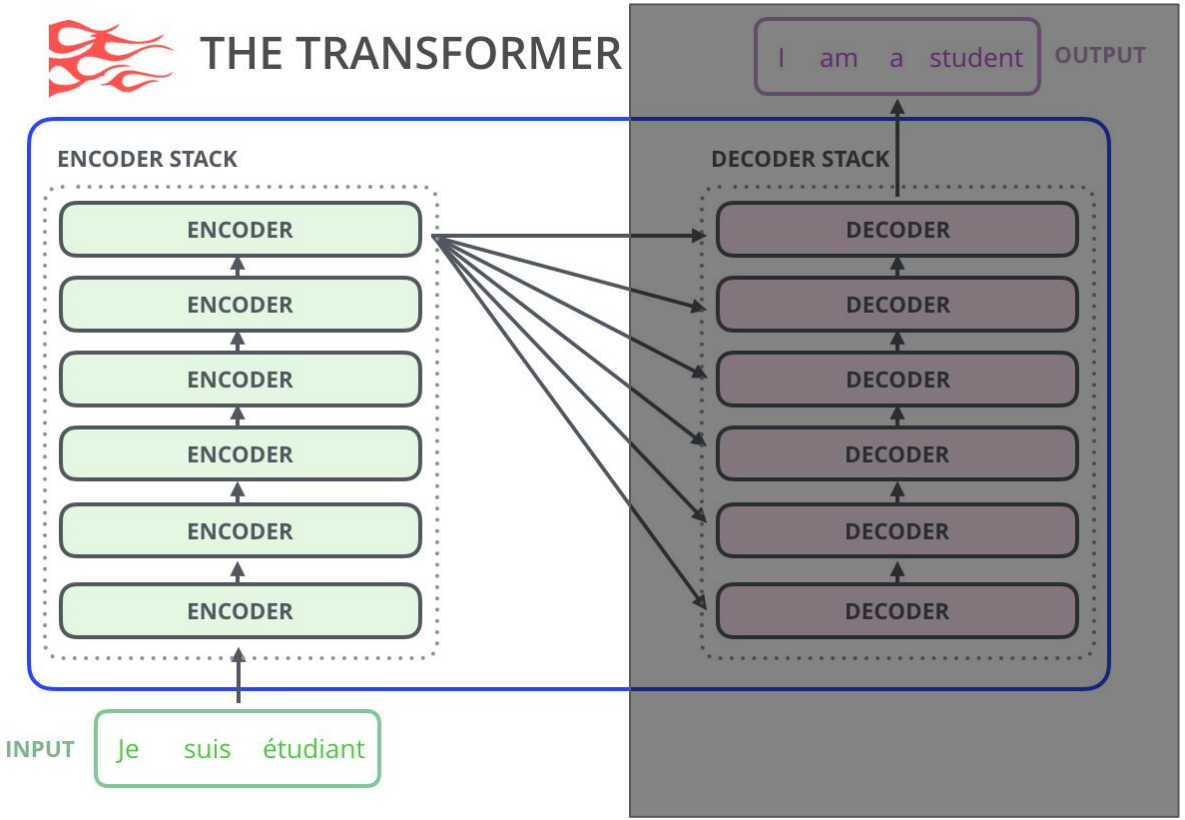


Examples: Music Generation



BERT: Bidirectional Encoder Representation from Transformer

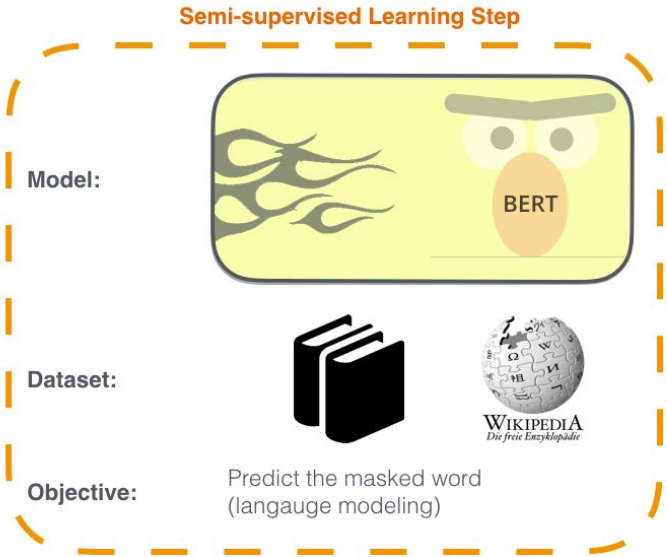
BERT


THE TRANSFORMER
I am a student
OUTPUT
ENCODER STACK
DECODER STACK
ENCODER
ENCODER
ENCODER
ENCODER
ENCODER
ENCODER
DECODER
DECODER
DECODER
DECODER
DECODER
DECODER
INPUT
Je suis étudiant

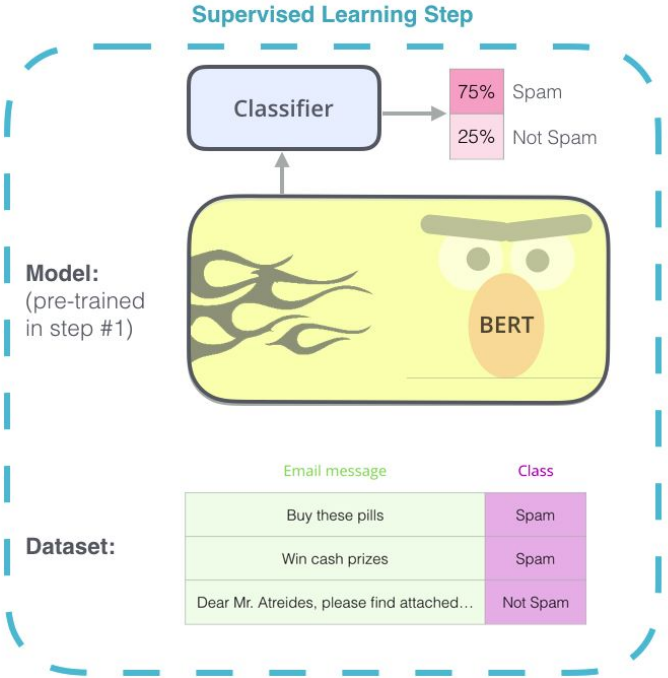
BERT

1 - Semi-supervised training on large amounts of text (books, wikipedia..etc).

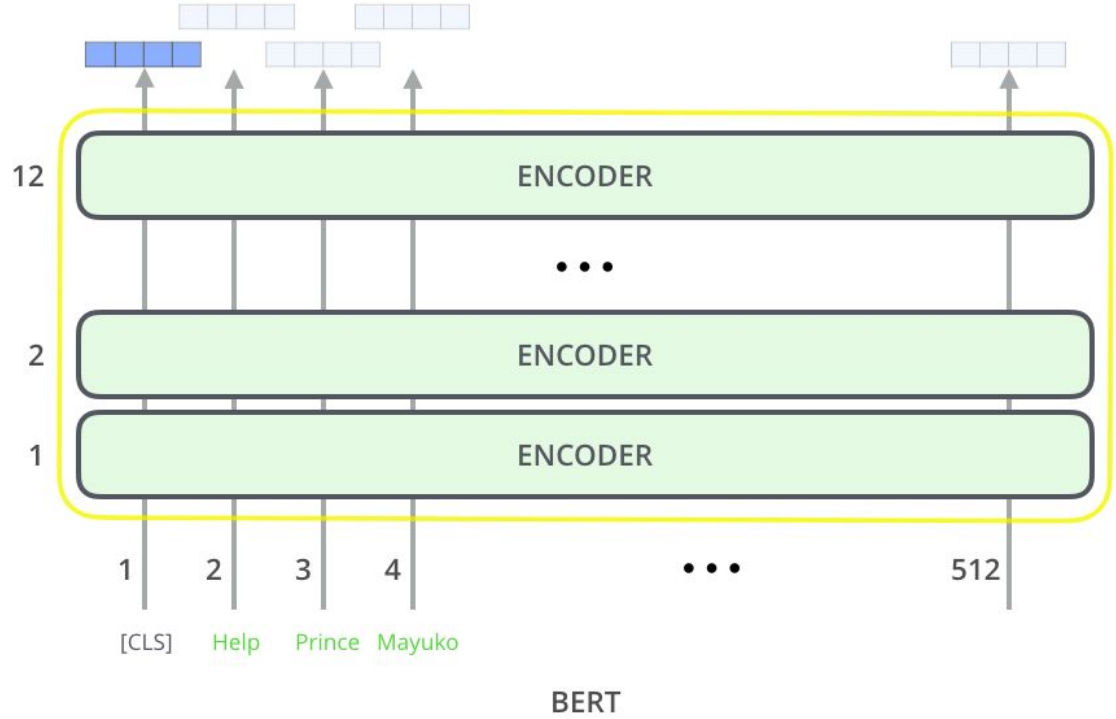
The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.



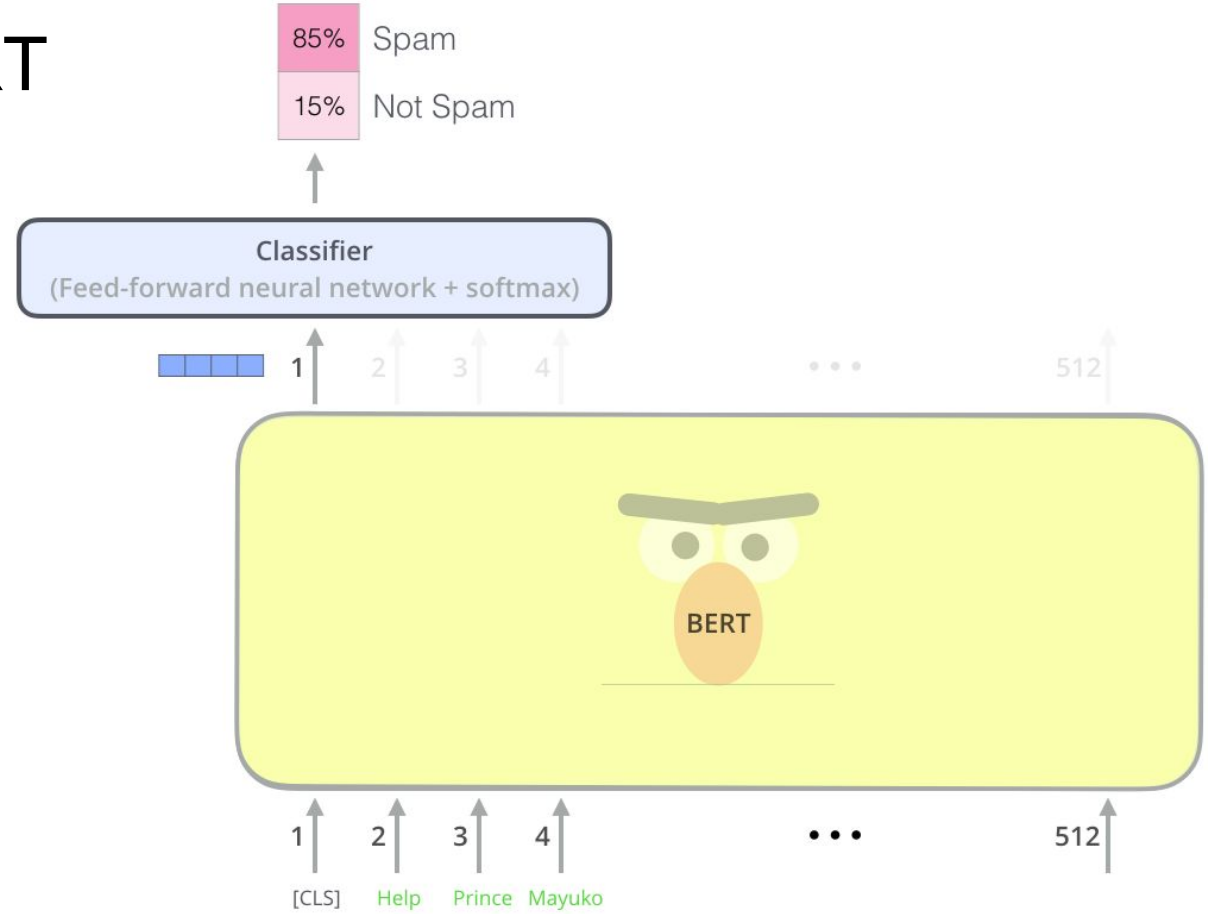
2 - Supervised training on a specific task with a labeled dataset.



BERT


12
ENCODER
2
ENCODER
1
ENCODER
1
2
3
4
512
[CLS]
Help
Prince
Mayuko
BERT

BERT


85%
Spam
15%
Not Spam
Classifier
(Feed-forward neural network + softmax)
1
2
3
4
512
BERT
1
2
3
4
512
[CLS]
Help
Prince
Mayuko

BERT pretraining: Masked Language Model

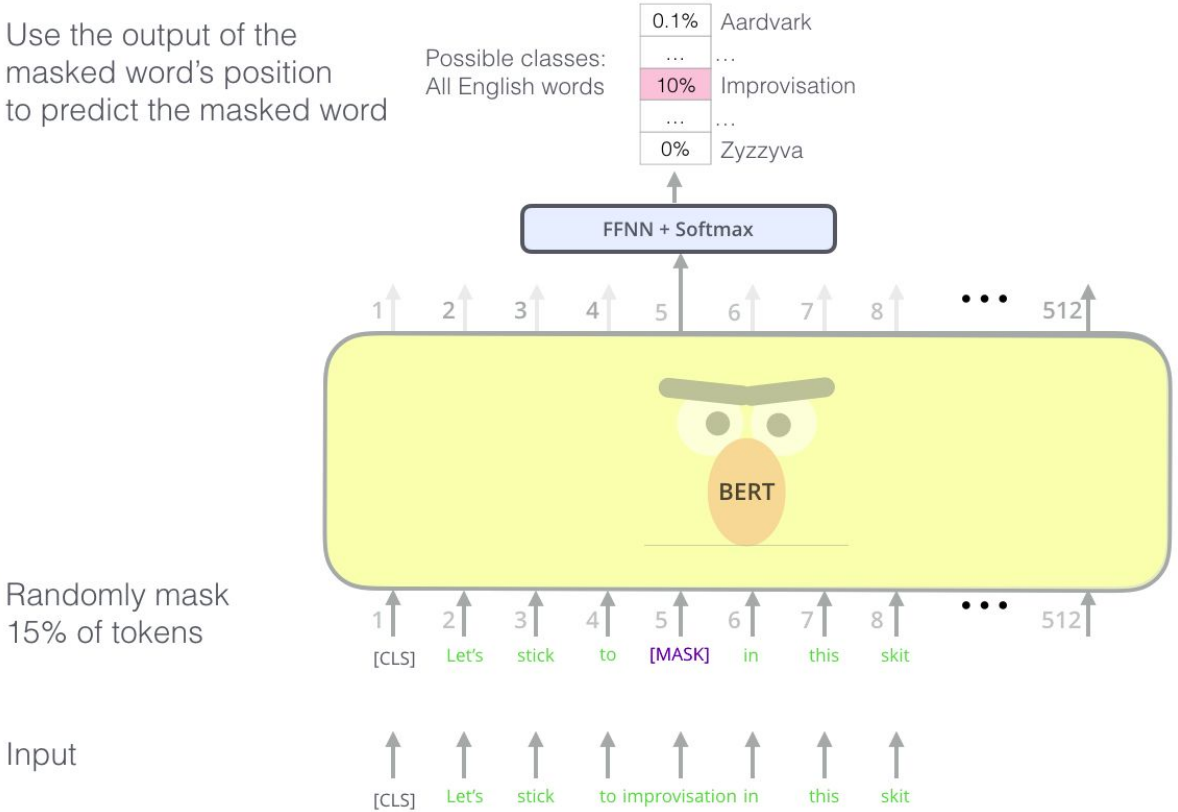
- **Solution**: Mask out k% of the input words, and then predict the masked words
- Too little masking: Too expensive to train
- Too much masking: Not enough context

BERT pretraining: Masked Language Model

- 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`

BERT pretraining: Masked Language Model



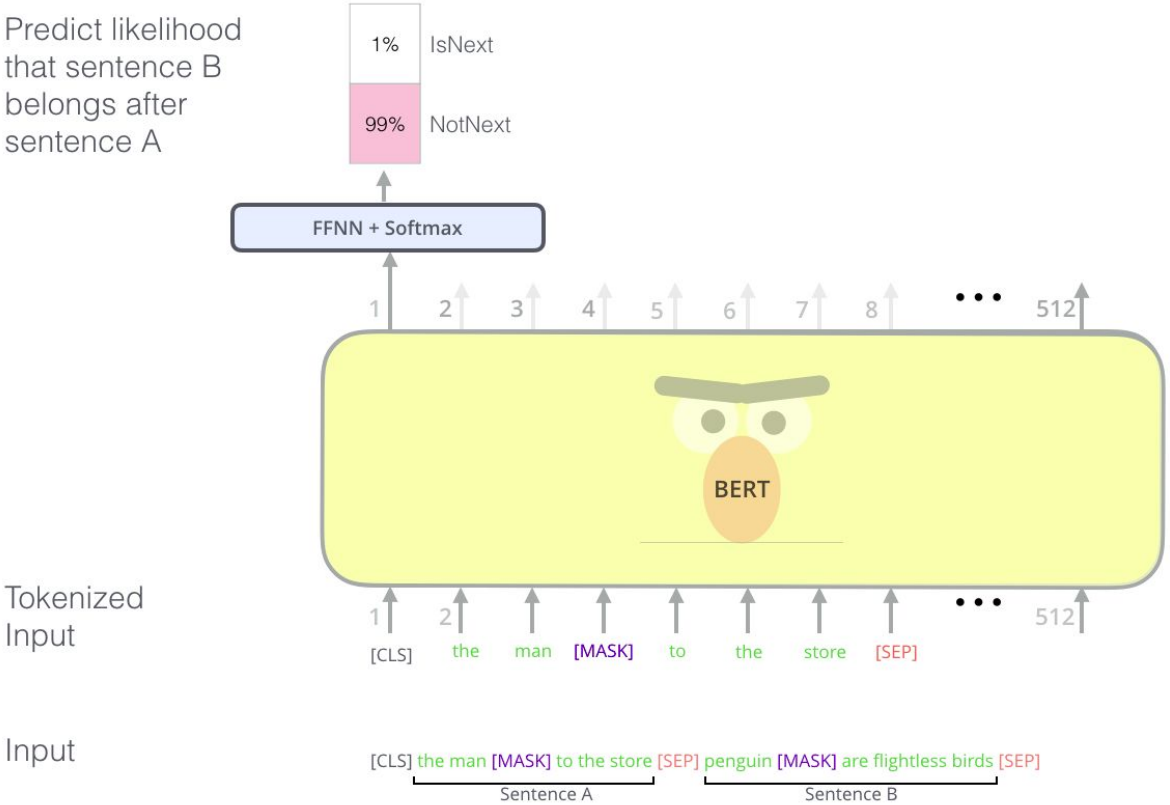
BERT pretraining: Next Sentence Prediction

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

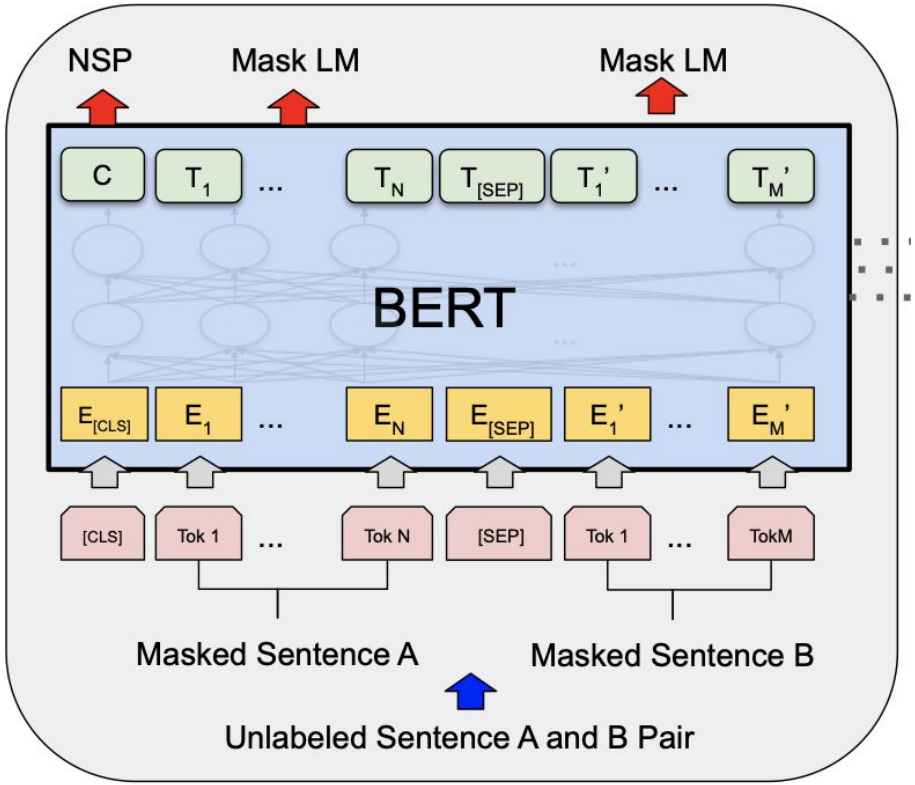
Sentence A = `The man went to the store.`
Sentence B = `He bought a gallon of milk.`
Label = `IsNextSentence`

Sentence A = `The man went to the store.`
Sentence B = `Penguins are flightless.`
Label = `NotNextSentence`

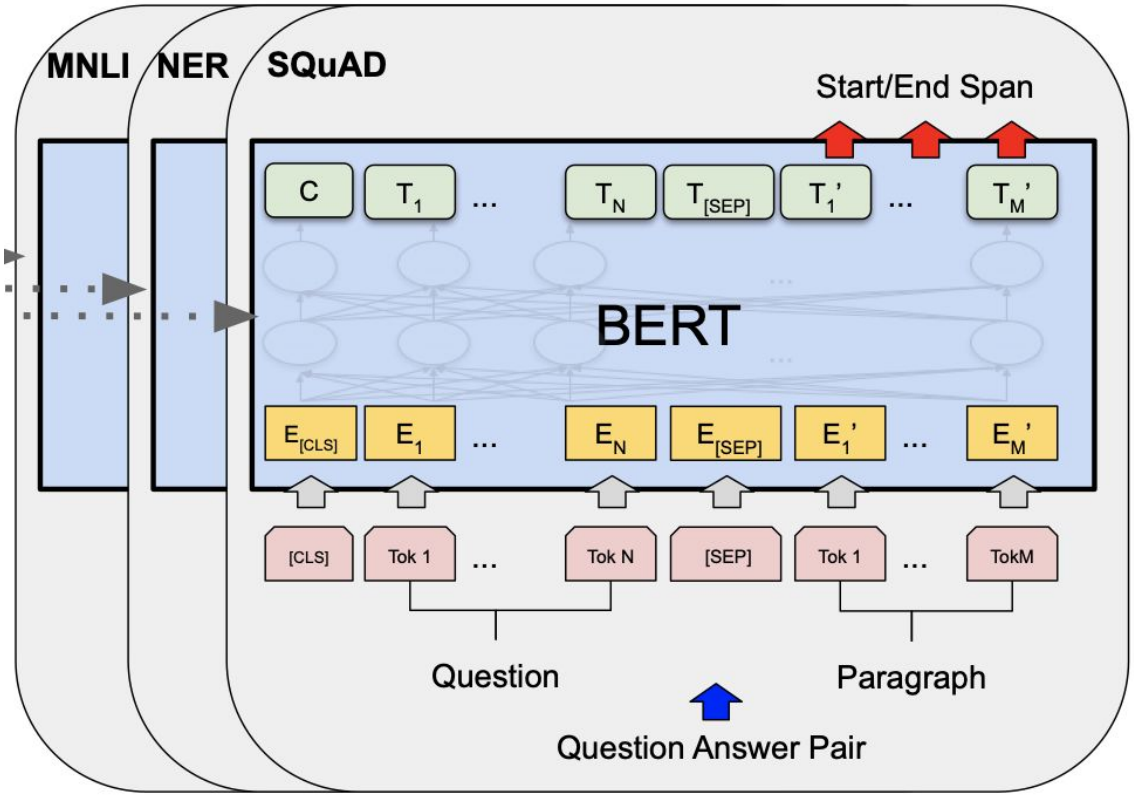
BERT pretraining: Next Sentence Prediction



BERT pretraining

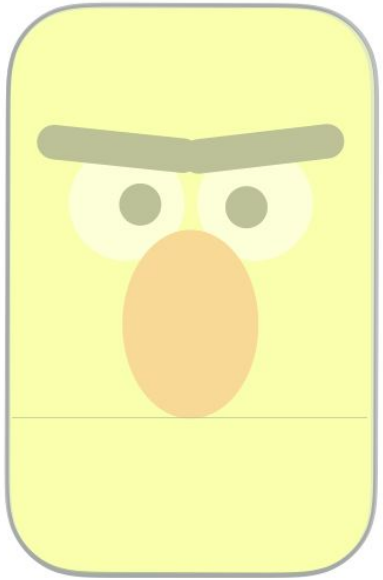


BERT: fine-tuning



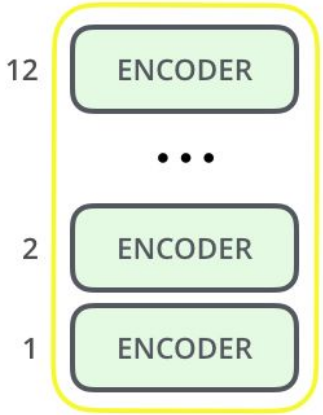
BERT

BERT$_{BASE}$

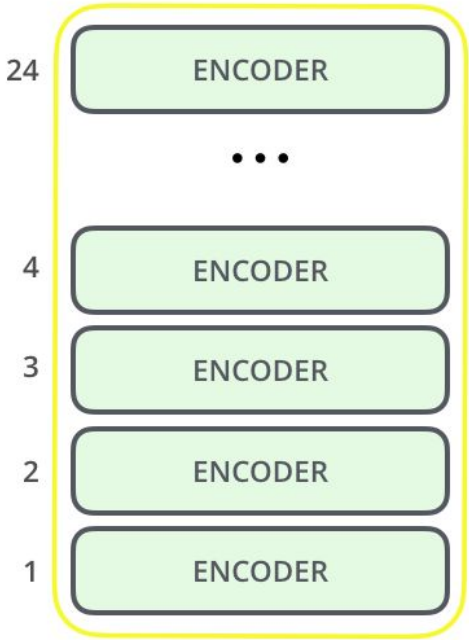


BERT$_{LARGE}$

BERT



$BERT_{BASE}$



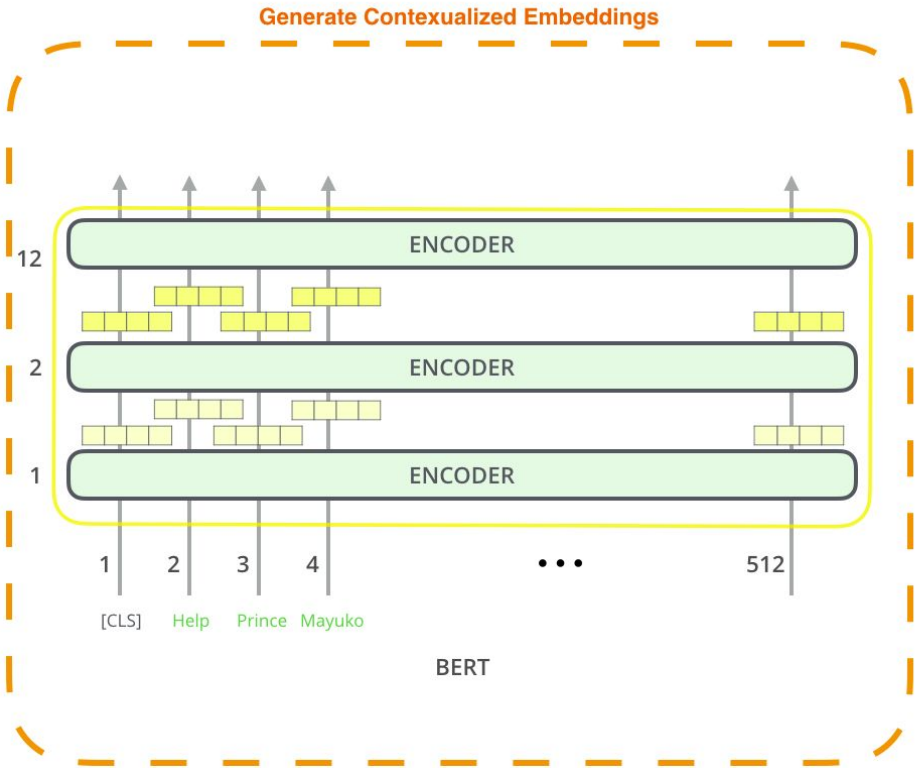
$BERT_{LARGE}$

- 768 and 1024 hidden units
- attention heads 12 and 16
- default Transformer configuration:
  6 encoder layers, 512 hidden units, 8 attention head

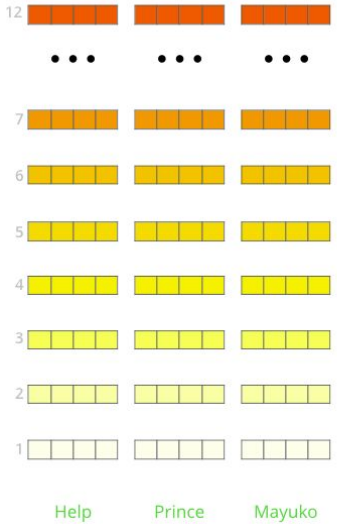
BERT

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
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
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

BERT: feature extraction



The output of each encoder layer along each token's path can be used as a feature representing that token.



But which one should we use?

BERT: feature extraction