BERT Fine-Tuning

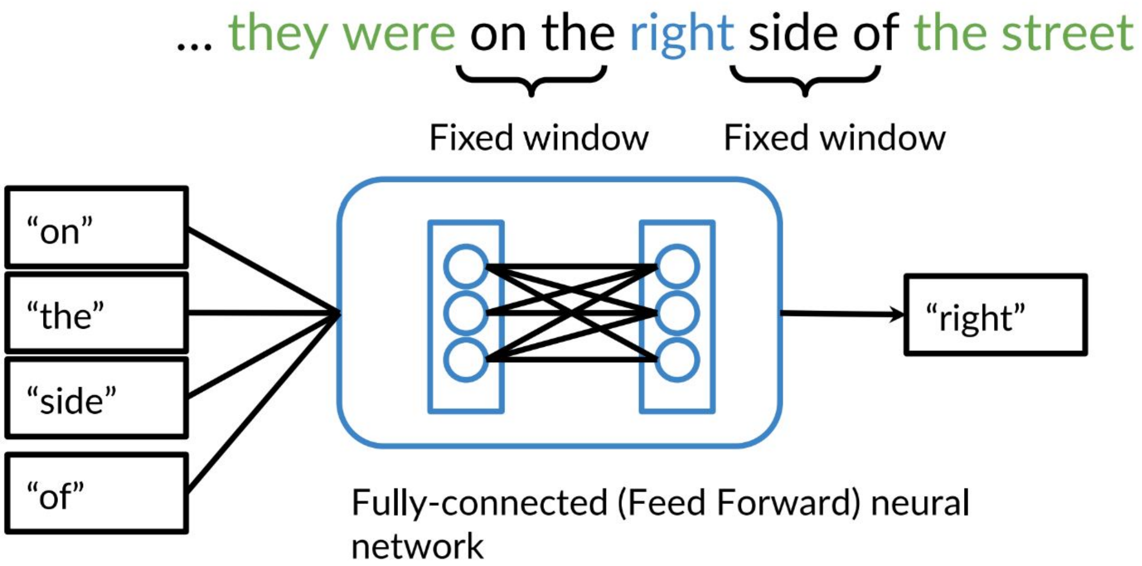
Context

... right ...

... they were on the right ...

... they were on the right side of the street

Continuous Bag of Words



Need more context?

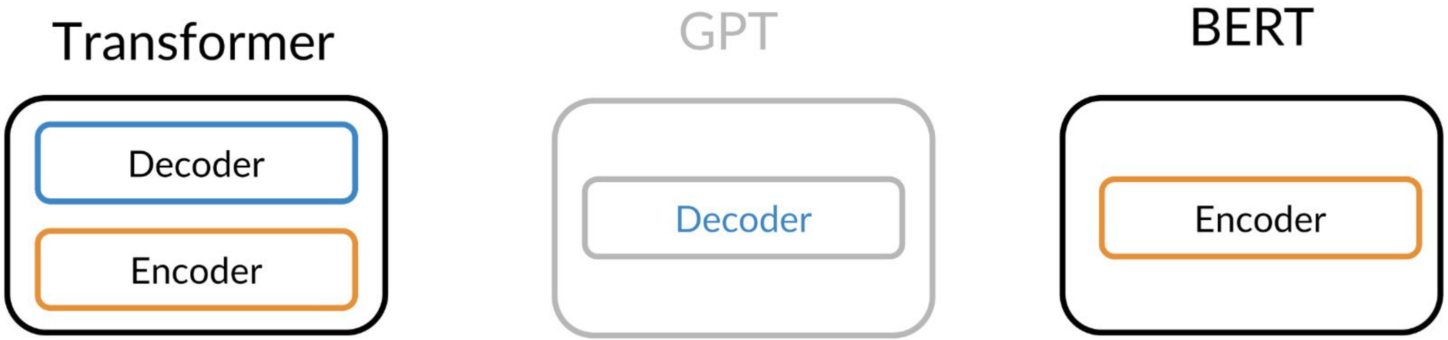
... they were $\underbrace{\text{on the}}_{\text{Fixed window}}$ right $\underbrace{\text{side of}}_{\text{Fixed window}}$ the street.

... they were on the right side of history.

Use all context words

The legislators believed that they were on the right side of history, so they changed the law.

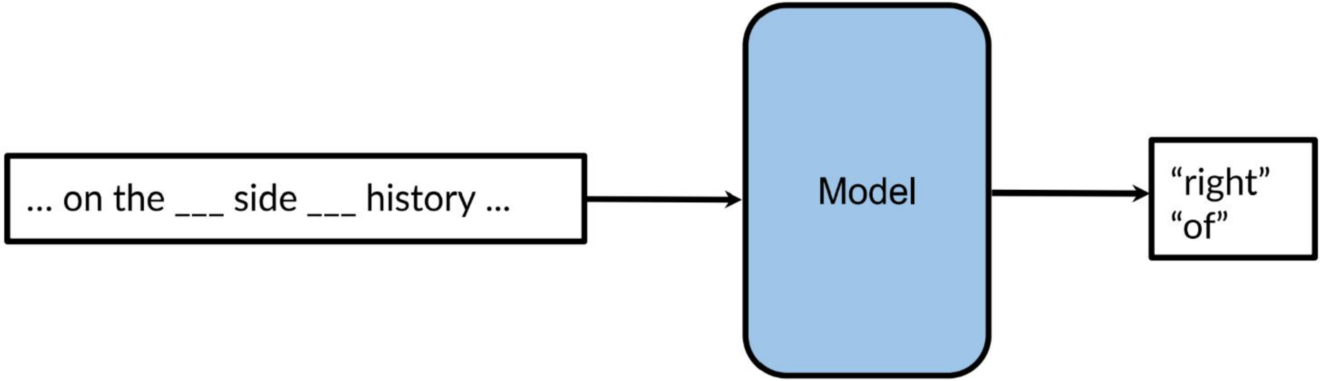
BERT



The legislators believed that they were on the _____ side of history, so they changed the law.

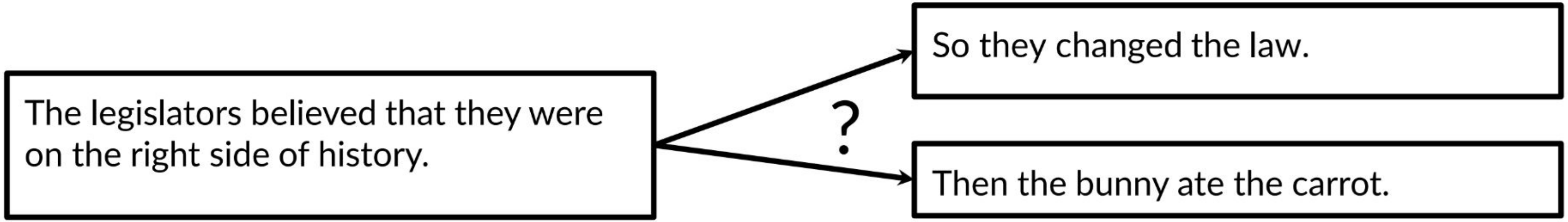
Bi-directional

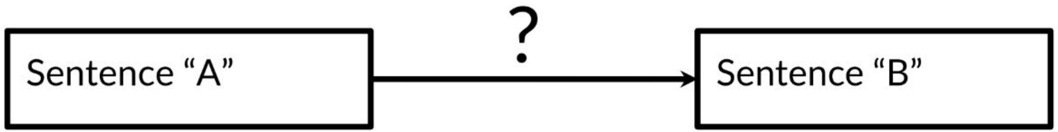
Transformer + Bi-directional Context



Multi-Mask Language Modeling

BERT: Words to Sentences

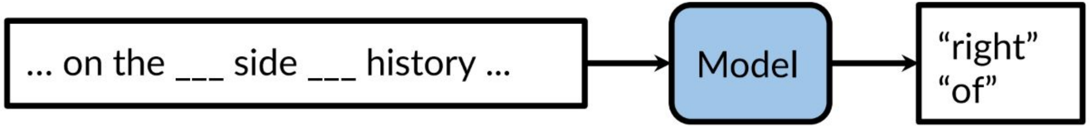




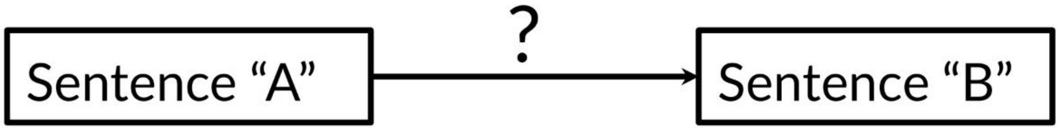
Next Sentence Prediction

BERT Pre-training Tasks

Multi-Mask Language Modeling



Next Sentence Prediction



Bidirectional Encoder Representations from Transformers (BERT)

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

[1810.04805.pdf (arxiv.org)](https://arxiv.org/pdf/1810.04805.pdf)

BERT

- A multi layer bidirectional transformer
- Positional embeddings
- BERT_base:

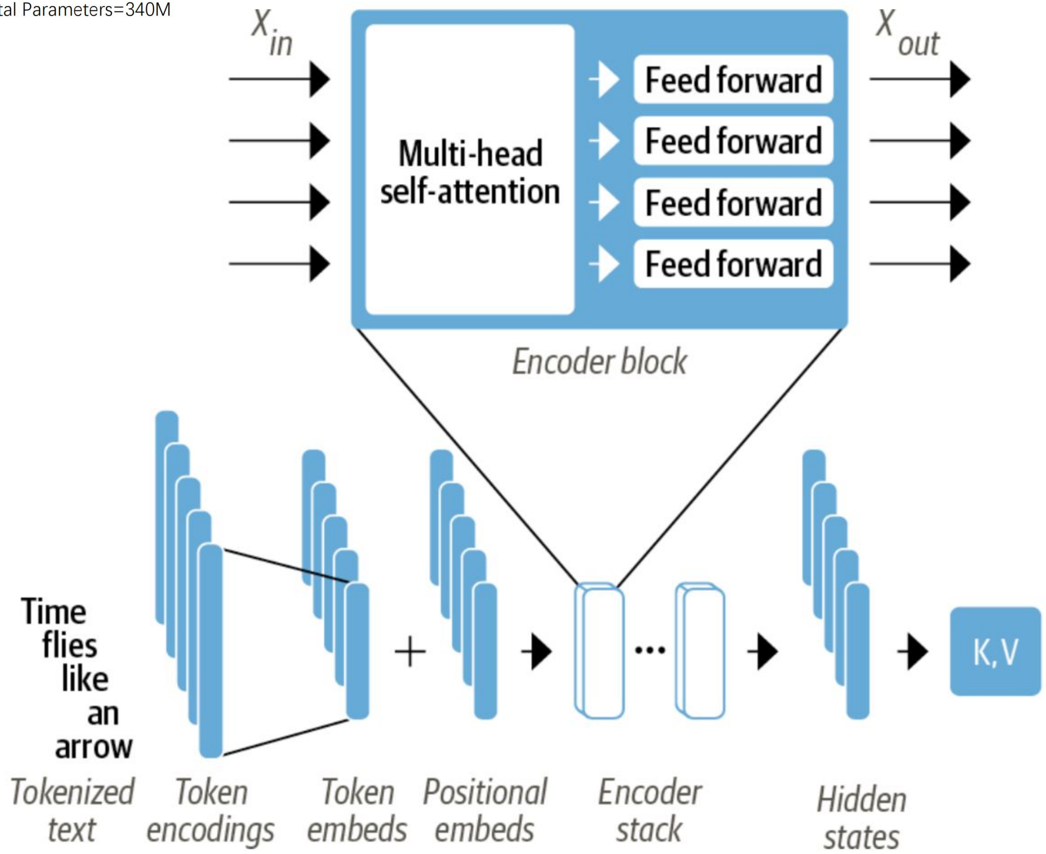
  12 layers (12 transformer blocks)

  12 attentions heads

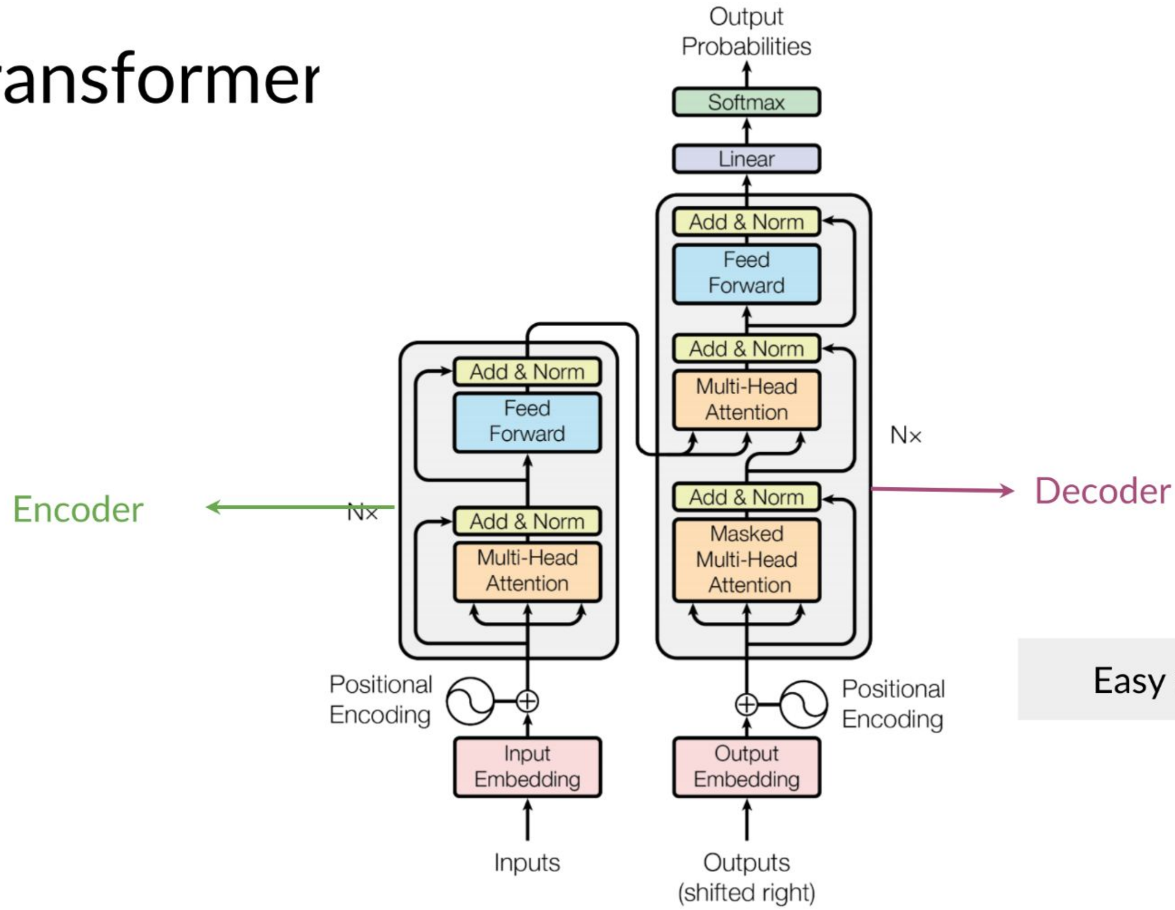
  110 million parameters

- $\mathrm{BERT_{BASE}}$: $\mathrm{N} = 6$, $d_{\mathrm{model}} = 512$, $h = 12$, Total Parameters=110M
- 4 cloud TPUs in Pod configuration (16 TPU chips total)

- $\mathrm{BERT_{LARGE}}$: $\mathrm{N} = 24$, $d_{\mathrm{model}} = 1024$, $h = 16$, Total Parameters=340M
- 16 Cloud TPUs (64 TPU chips total)
- Each pretraining took 4 days to complete.



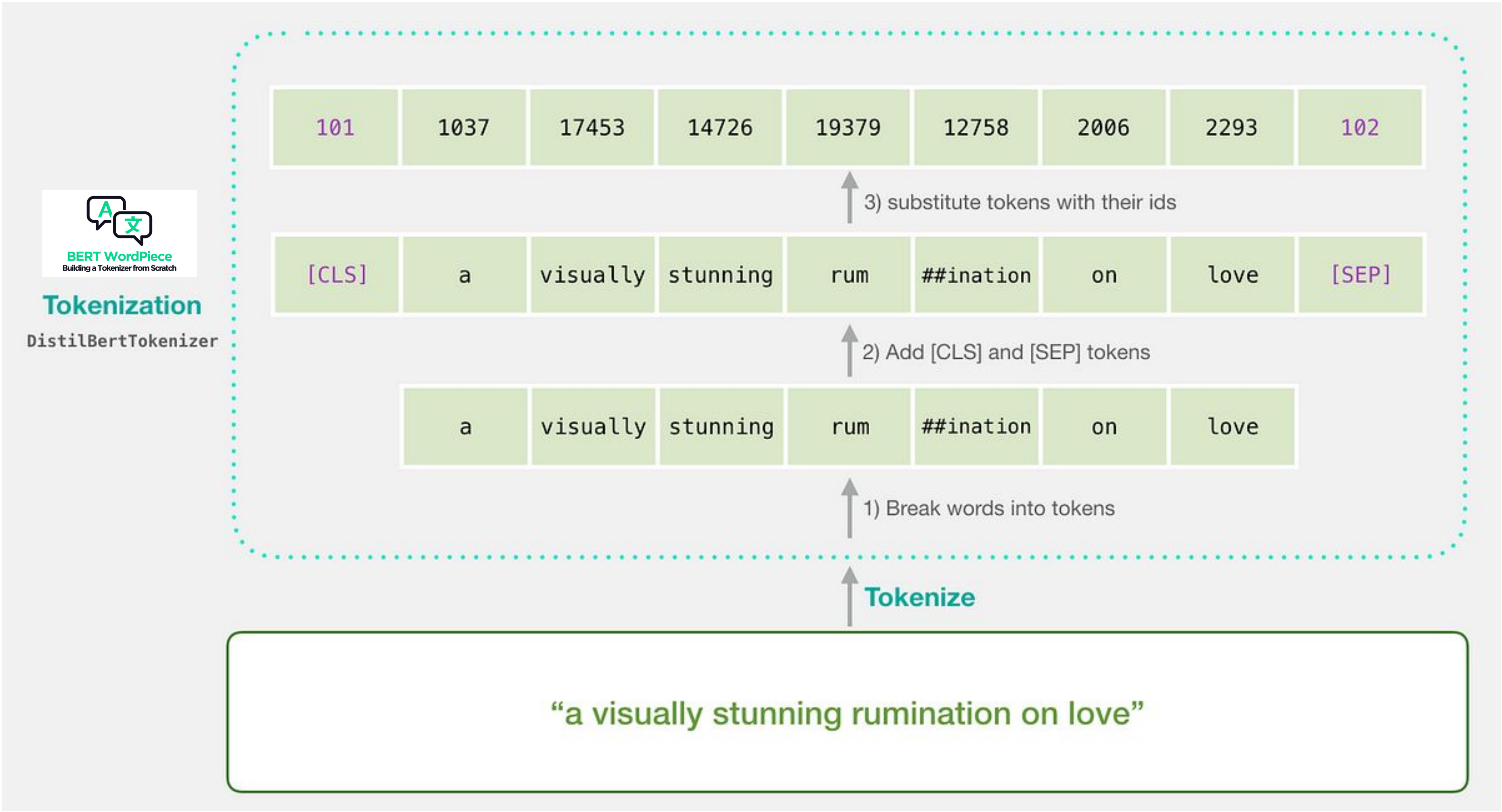
The Transformer



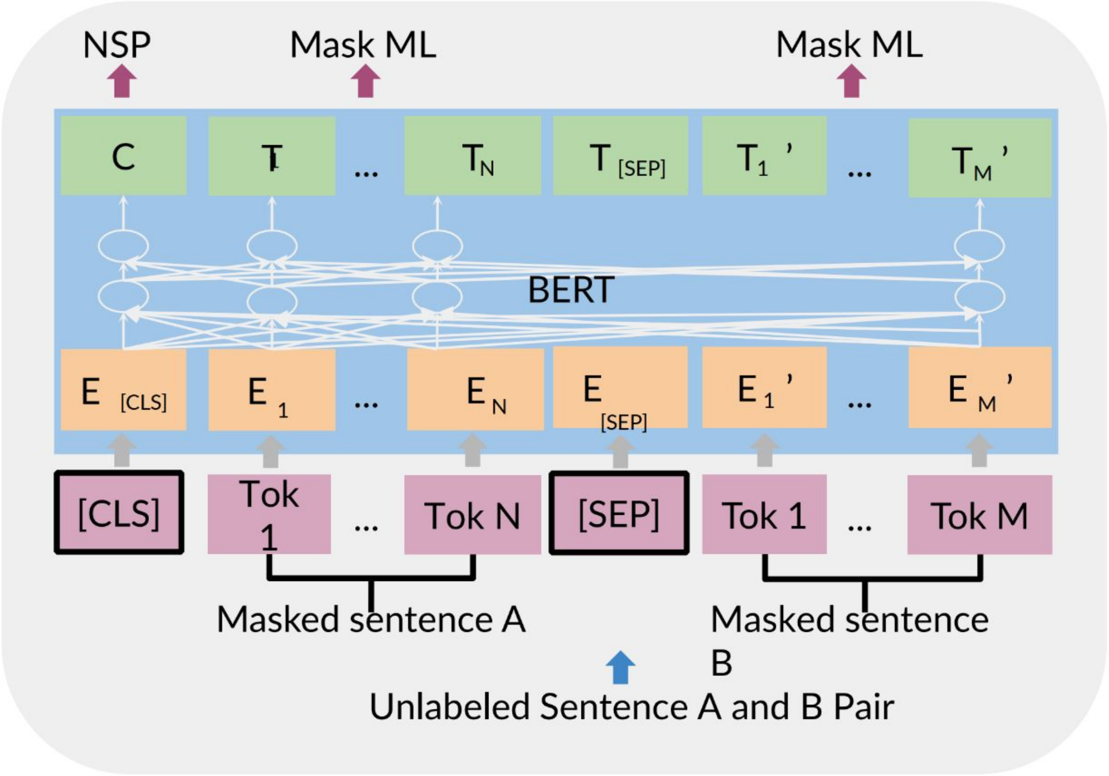
Easy to parallelize!

Formalizing the input

Input	[CLS]	my	dog	is	cute	[SEP]	he	likes	play	##ing	[SEP]
Token Embeddings	$E_{[CLS]}$	E_{my}	E_{dog}	E_{is}	E_{cute}	$E_{[SEP]}$	E_{he}	E_{likes}	E_{play}	$E_{\#\#ing}$	$E_{[SEP]}$
	+	+	+	+	+	+	+	+	+	+	+
Segment Embeddings	E_A	E_A	E_A	E_A	E_A	E_A	E_B	E_B	E_B	E_B	E_B
	+	+	+	+	+	+	+	+	+	+	+
Position Embeddings	E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}


BERT WordPiece
Building a Tokenizer from Scratch
Tokenization
DistilBertTokenizer
101 1037 17453 14726 19379 12758 2006 2293 102
3) substitute tokens with their ids
[CLS] a visually stunning rum ##ination on love [SEP]
2) Add [CLS] and [SEP] tokens
a visually stunning rum ##ination on love
1) Break words into tokens
Tokenize
"a visually stunning rumination on love"

Visualizing the output



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

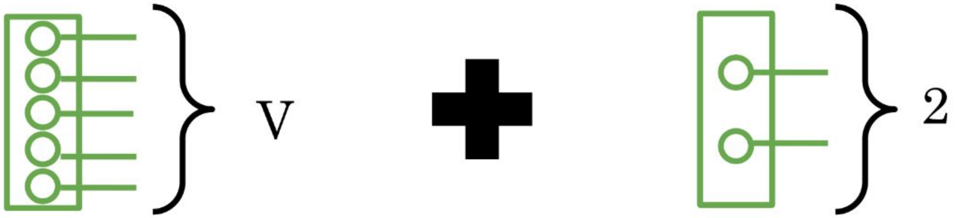
BERT Objective

Objective 1:
Multi-Mask LM

Loss: Cross Entropy Loss

Objective 2:
Next Sentence Prediction

Loss: Binary Loss



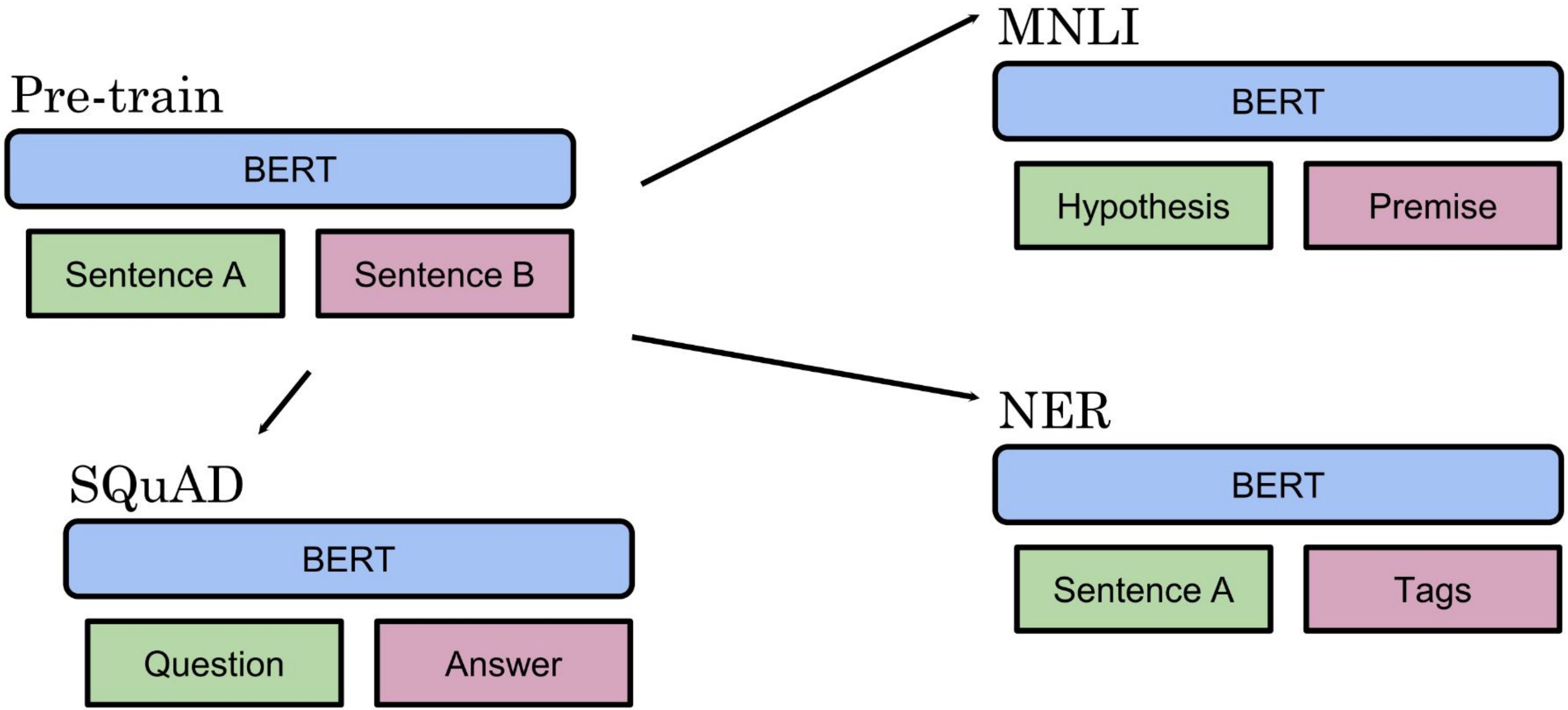
- 80% of the time: Replace the word with the `[MASK]` token, e.g., `my dog is hairy` → `my dog is [MASK]`

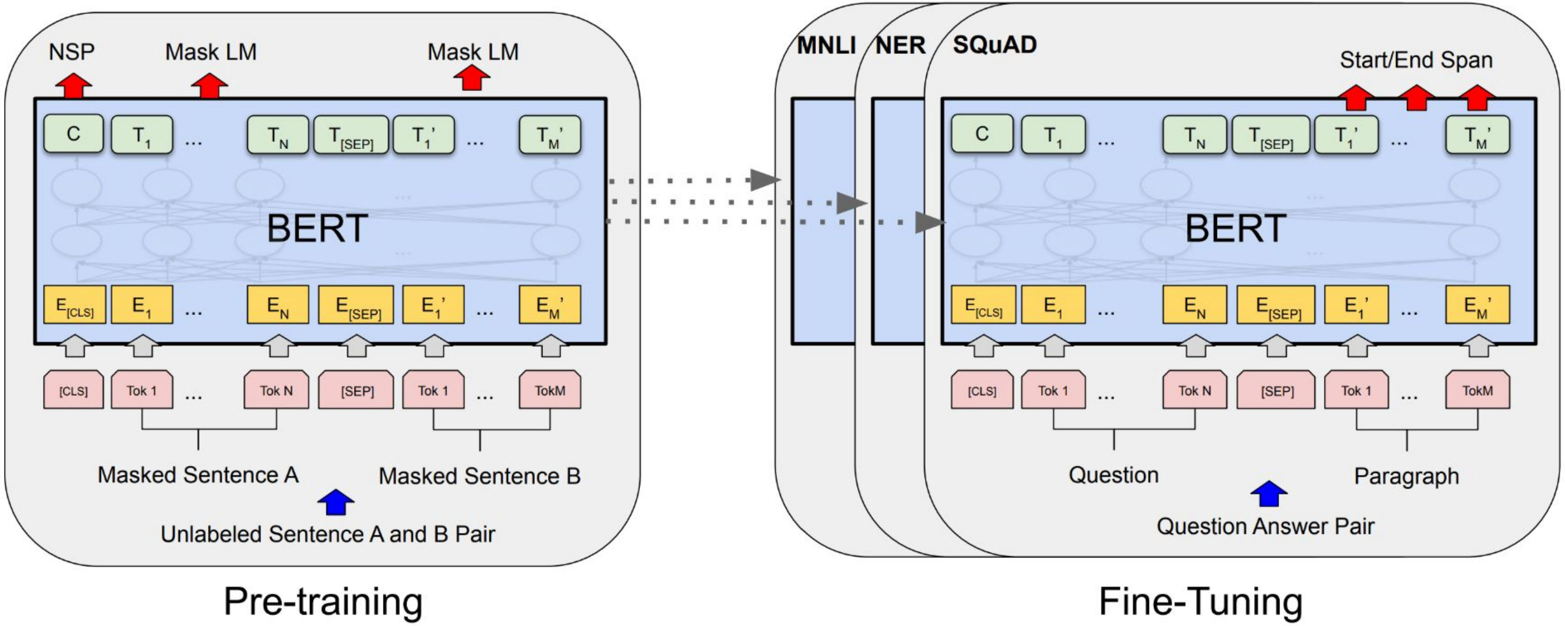
- 10% of the time: Replace the word with a random word, e.g., `my dog is hairy` → `my dog is apple`

- 10% of the time: Keep the word unchanged, e.g., `my dog is hairy` → `my dog is hairy`. The purpose of this is to bias the representation towards the actual observed word.

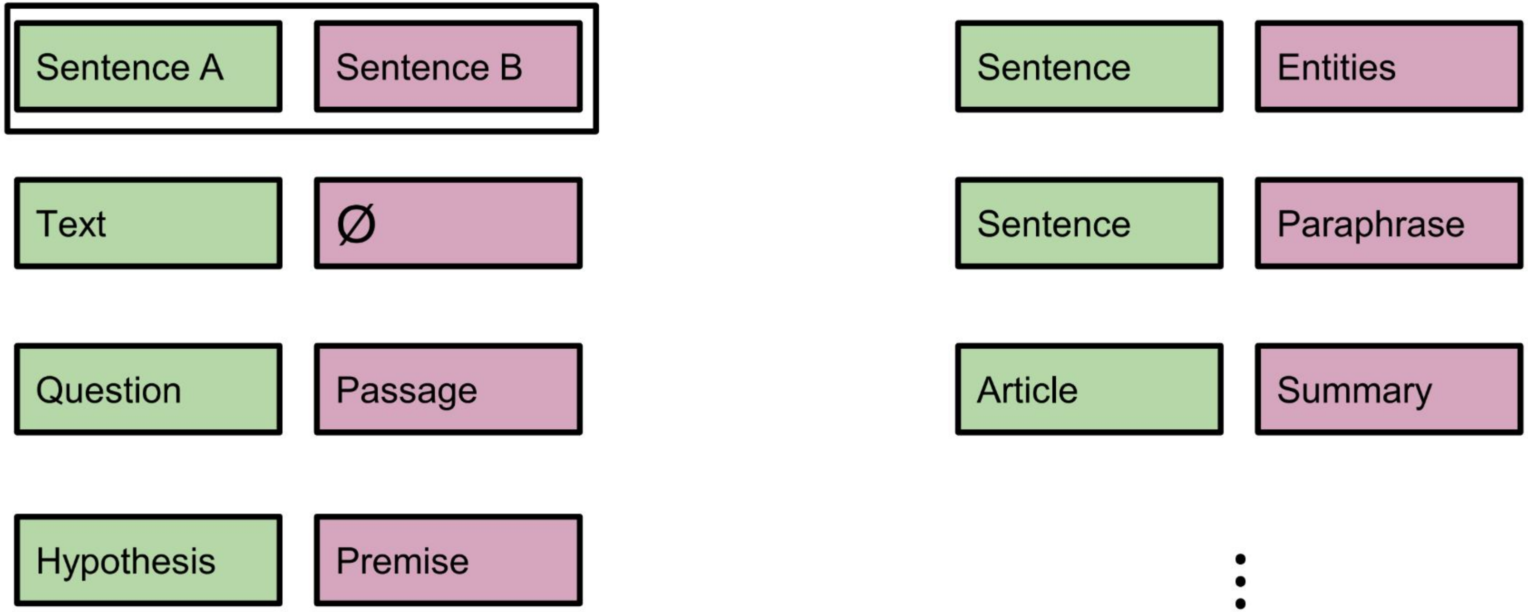
Fine-tuning BERT

Fine-tuning BERT: Outline




NSP
Mask LM
Mask LM
C
T_1
...
T_N
$T_{[SEP]}$
T_1'
T_M'
BERT
$E_{[CLS]}$
E_1
E_N
$E_{[SEP]}$
E_1'
E_M'
[CLS]
Tok 1
Tok N
[SEP]
Tok 1
TokM
Masked Sentence A
Masked Sentence B
Unlabeled Sentence A and B Pair
Pre-training
MNLI
NER
SQuAD
Start/End Span
C
T_1
T_N
$T_{[SEP]}$
T_1'
T_M'
BERT
$E_{[CLS]}$
E_1
E_N
$E_{[SEP]}$
E_1'
E_M'
[CLS]
Tok 1
Tok N
[SEP]
Tok 1
TokM
Question
Paragraph
Question Answer Pair
Fine-Tuning

Summary



Sentence A
Sentence B
Text
Ø
Question
Passage
Hypothesis
Premise
Sentence
Entities
Sentence
Paraphrase
Article
Summary

Fine-tuning with BERT

- Context vector C: Take the final hidden state corresponding to the first token in the input: [CLS].
- Transform to a probability distribution of the class labels:

$$P = \mathrm{softmax}(CW^T)$$

- **Batch size**: 16, 32
- **Learning rate (Adam)**: 5e-5, 3e-5, 2e-5
- **Number of epochs**: 3, 4

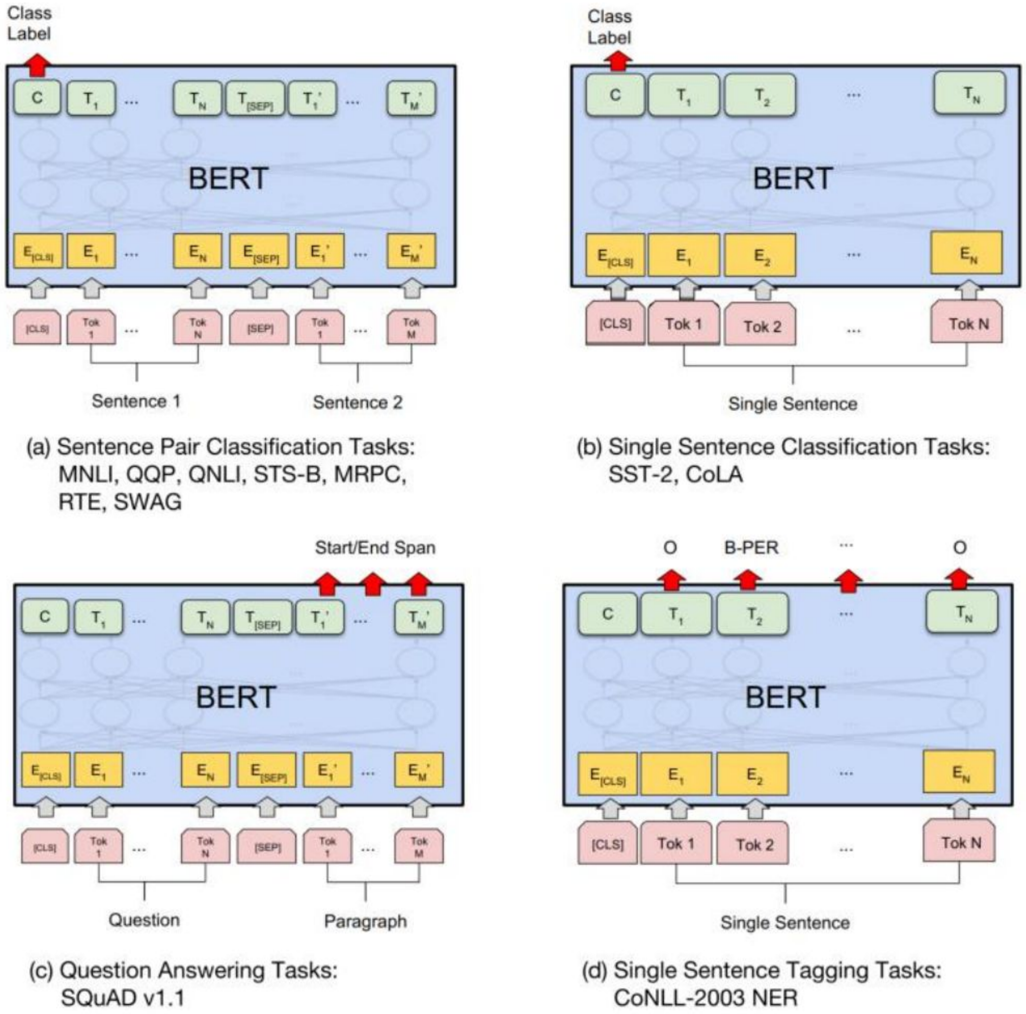


Figure in (Devlin *et al.*, 2018)

Evaluation for BERT: GLUE

- General Language Understanding Evaluation (**GLUE**) benchmark: Standard split of data to train, validation, test, where labels for the test set is only held in the server.

- Sentence pair tasks
  - **MNLI**, Multi-Genre Natural Language Inference
  - **QQP**, Quora Question Pairs
  - **QNLI,** Question Natural Language Inference
  - **STS-B** The Semantic Textual Similarity Benchmark
  - **MRPC** Microsoft Research Paraphrase Corpus
  - **RTE** Recognizing Textual Entailment
  - **WNLI** Winograd NLI is a small natural language inference dataset
- Single sentence classification
  - **SST-2** The Stanford Sentiment Treebank
  - **CoLA** The Corpus of Linguistic Acceptability

Dataset: [Hugging Face – The AI community building the future.]

Evaluation for BERT: GLUE

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	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**91.1**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**81.9**

Table 1: GLUE Test results, scored by the GLUE evaluation server. The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set. OpenAI GPT = (L=12, H=768, A=12); $\text{BERT}_{\text{BASE}}$ = (L=12, H=768, A=12); $\text{BERT}_{\text{LARGE}}$ = (L=24, H=1024, A=16). BERT and OpenAI GPT are single-model, single task. All results obtained from https://gluebenchmark.com/leaderboard and https://blog.openai.com/language-unsupervised/.

Evaluation on SQUAD

- The Standford Question Answering Dataset (SQuAD) is a collection of 100k crowdsourced question/answer pairs.

- **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
```

System	Dev EM	Dev F1	Test EM	Test F1
Leaderboard (Oct 8th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
#1 Single - nlnet	-	-	83.5	90.1
#2 Single - QANet	-	-	82.5	89.3
Published				
BiDAF+ELMo (Single)	-	85.8	-	-
R.M. Reader (Single)	78.9	86.3	79.5	86.6
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$\text{BERT}_{\text{BASE}}$ (Single)	80.8	88.5	-	-
$\text{BERT}_{\text{LARGE}}$ (Single)	84.1	90.9	-	-
$\text{BERT}_{\text{LARGE}}$ (Ensemble)	85.8	91.8	-	-
$\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$\text{BERT}_{\text{LARGE}}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

Table 2: SQuAD results. The BERT ensemble is 7x systems which use different pre-training checkpoints and fine-tuning seeds.

Table in (Devlin *et al.*, 2018)

Evaluation on Named Entity Recognition

- The CoNLL 2003 Named Entity Recognition (NER) dataset. This dataset consists of 200k training words which have been annotated as **Person**, **Organization**, **Location**, **Miscellaneous**, or **Other** (non-named entity).

```
Jim    Hen    ##son was a puppet ##eer
I-PER I-PER X      O   O O      X
```

System	Dev F1	Test F1
ELMo+BiLSTM+CRF	95.7	92.2
CVT+Multi (Clark et al., 2018)	-	92.6
BERT$_{\text{BASE}}$	96.4	92.4
BERT$_{\text{LARGE}}$	**96.6**	**92.8**

Table 3: CoNLL-2003 Named Entity Recognition results. The hyperparameters were selected using the Dev set, and the reported Dev and Test scores are averaged over 5 random restarts using those hyperparameters.

Table in (Devlin *et al.*, 2018)