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

Google AI Language, 2019

Presented by
Seyoung Kim

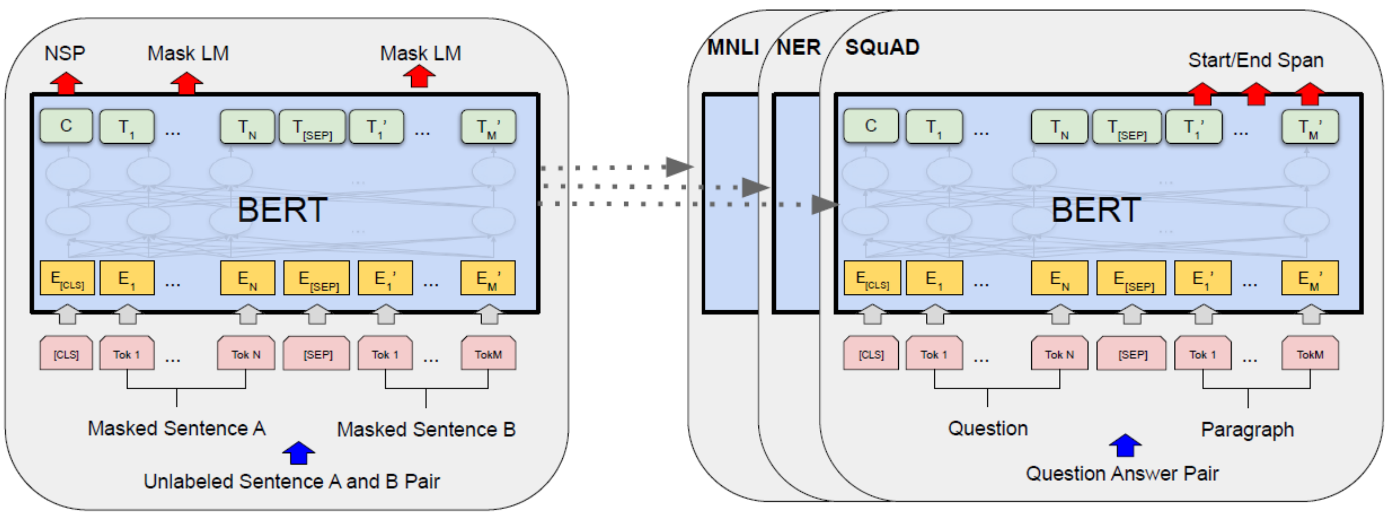
M.IN.D Lab

Table of Contents

Introduction

BERT

- **B**idirectional **E**ncoder **R**epresentations from **T**ransformers
- **Deep Bidirectional** Transformers for language understanding
  - Jointly conditioning on left & right context in all layers



Pre-training

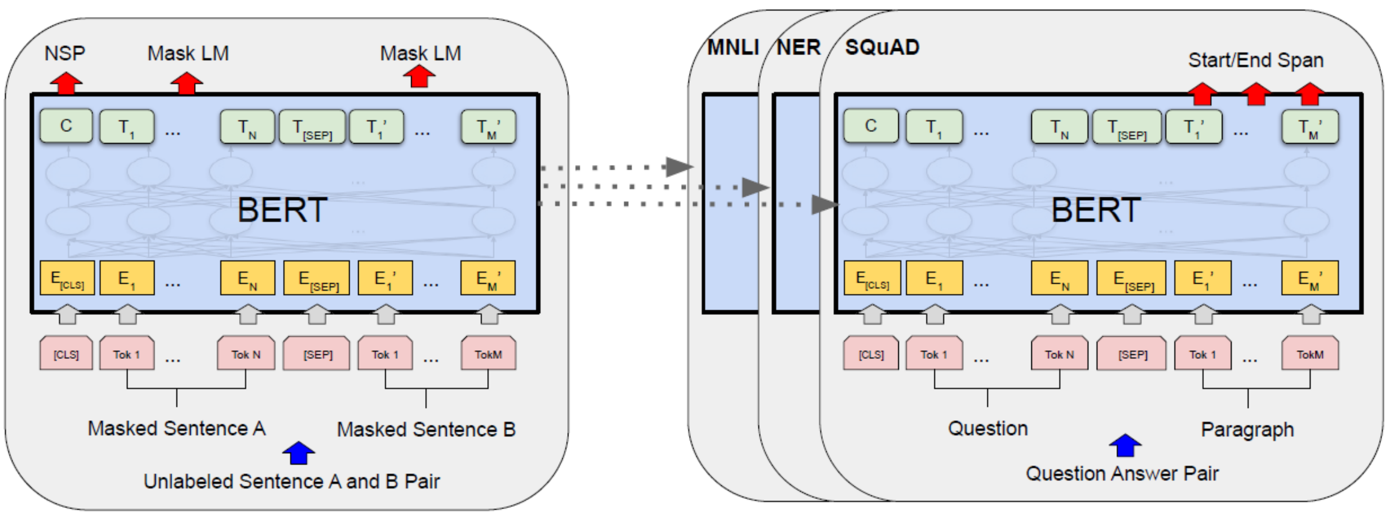
Fine-Tuning

M.IN.D Lab

BERT

- Adaptive to many downstream tasks regardless of input type
  - Single sentence
  - <Sentence, Sentence> pair
- Need only one additional output layer, fine-tuned end-to-end

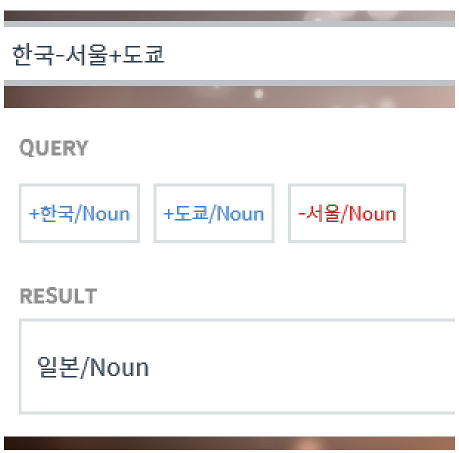
  → Minimize the # of params need to be learned from scratch

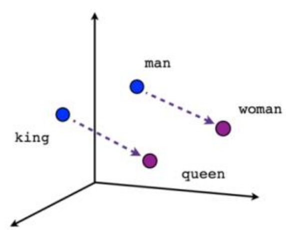


Pre-training Fine-Tuning

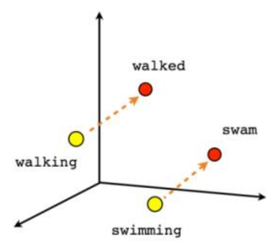
Word Embedding in NLP

- Word Embedding
  - The basis of deep learning for NLP
  - Representing words in the vector space
- Word2Vec, GloVe
  - Pre-trained on text corpus from co-occurrence statistics

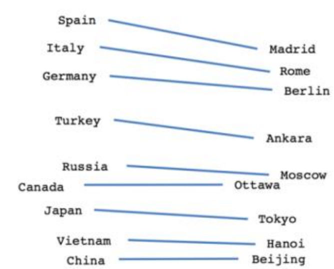




Male-Female



Verb tense



Country-Capital

Word Embedding in NLP

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

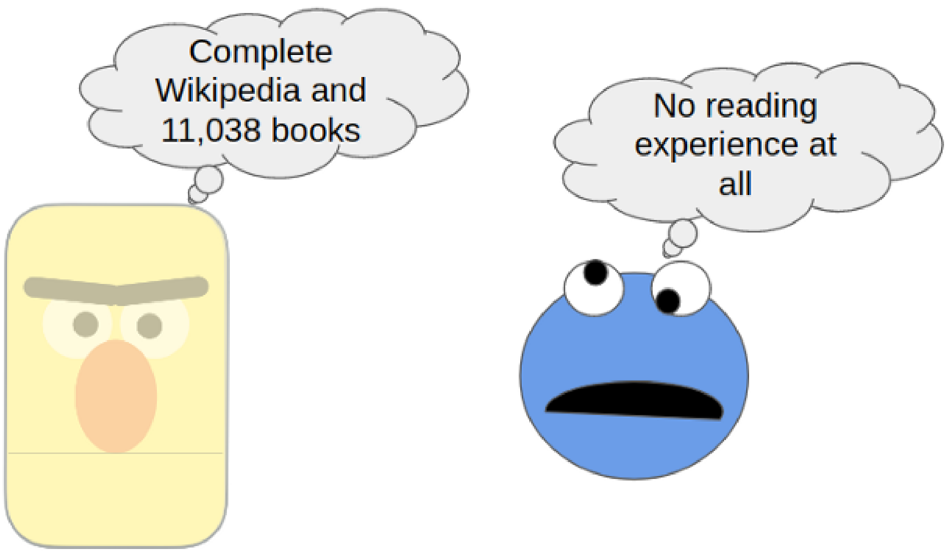
```
open a bank account                on the river bank

              [0.3, 0.2, -0.8, …]
```

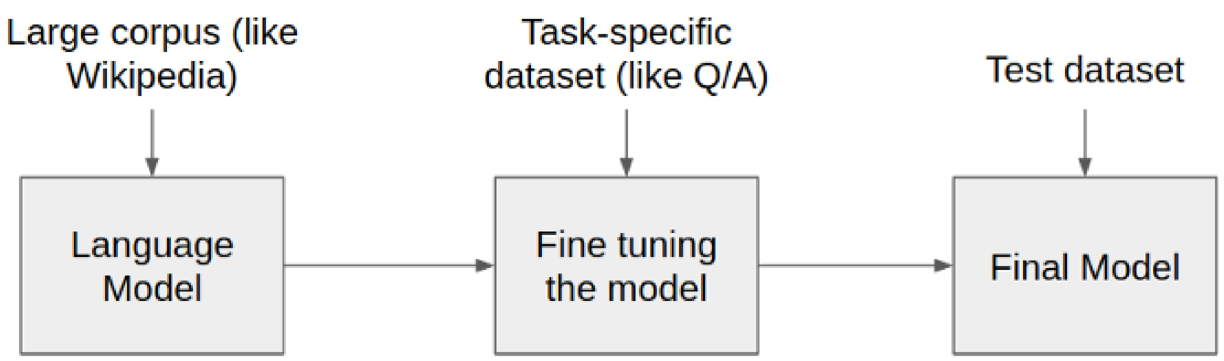
- Solution
  - Train contextual representations on text corpus

    → **Make pre-trained Language Model!**

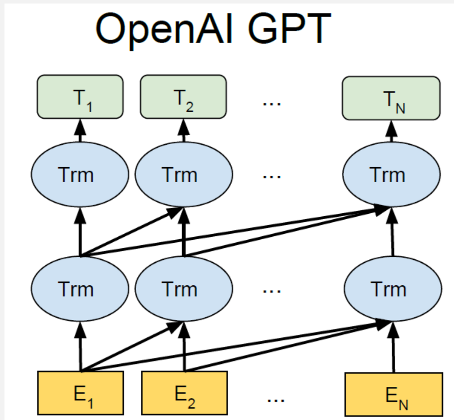
```
[0.9, -0.2, 1.6, …]                   [-1.9, -0.4, 0.1, …]

open a bank account                on the river bank
```

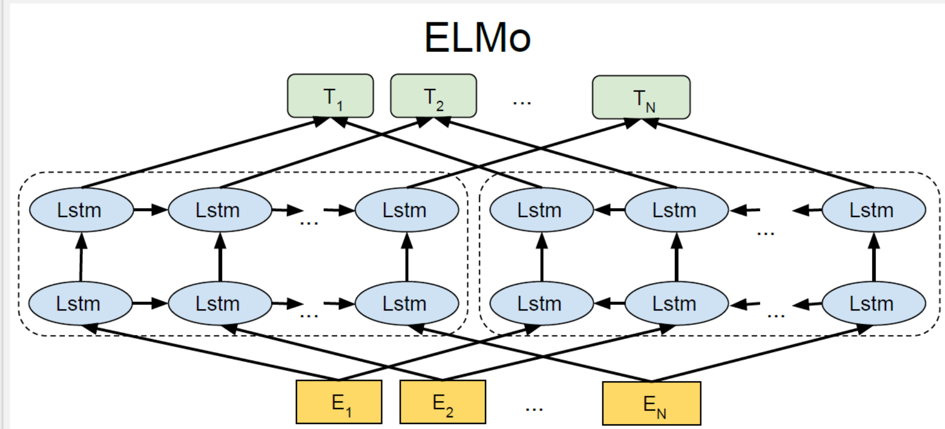
Pre-trained Language Model





Previous Works (Pre-train Model)





Model Structure	Left-to-right model	Shallow concat of left-to-right & right-to-left model
Block	Transformer's Decoder	LSTM
Applying strategy	Fine-tuning	Feature-based
Pre-train task	Next word prediction	Next word prediction

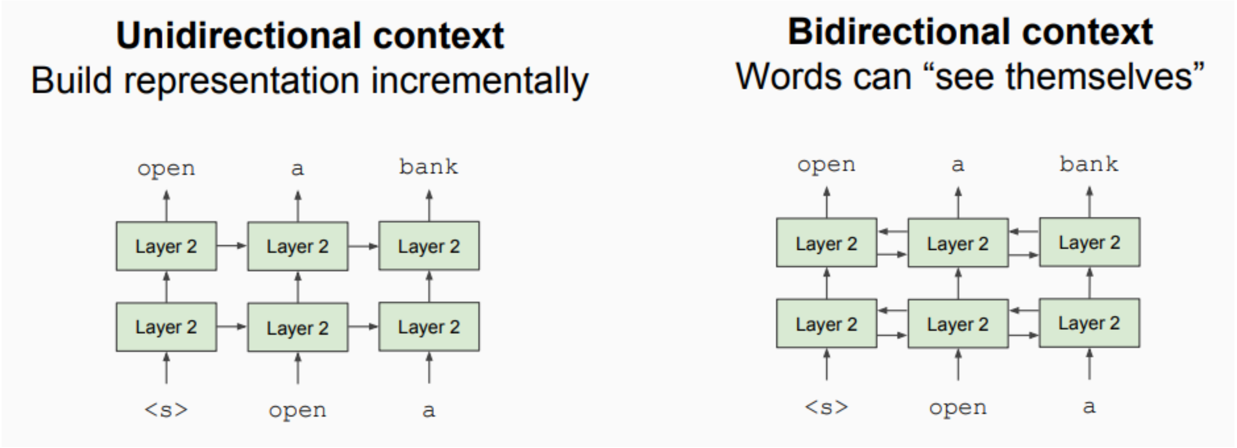
Limitations of previous works

- Unidirectional structure
  - Cannot learn context from both direction
    → Harmful for both sentence-level, token-level tasks

I need a fan to cool my heat!

Limitations of previous works

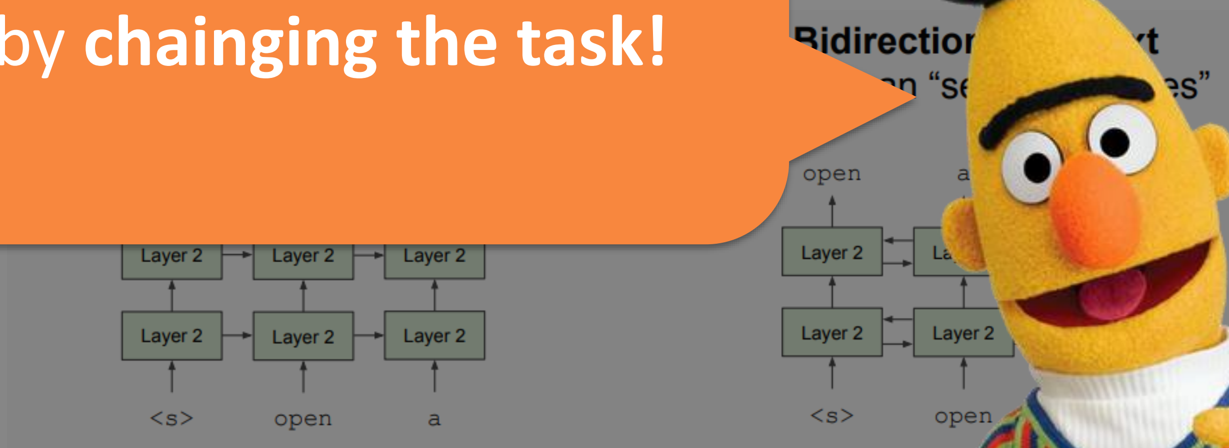
- Why are traditional LMs unidirectional?
  - Pre-train task: *Predict next word*
    → If we use bidirectional structure, words can "see themselves"
    → *CHEATING*, not *LEARNING*



Limitations of previous works

- Why are traditional LMs unidirectional?
  - Pre-train task: *Predict next word*
    - …, words can "see themselves"

Overcome this limitation by **chainging the task!**



BERT

Model Architecture

- Multi-layer bidirectional Transformer Encoder
  - Use Transformer encoder as a basic block (bidirectional)

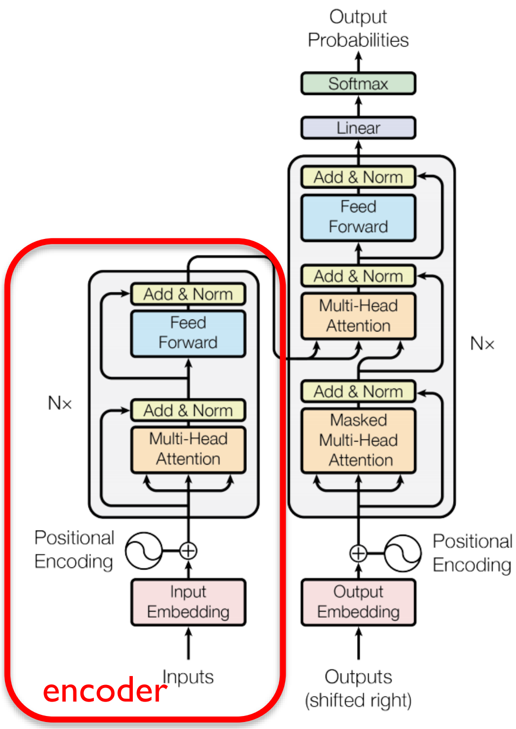
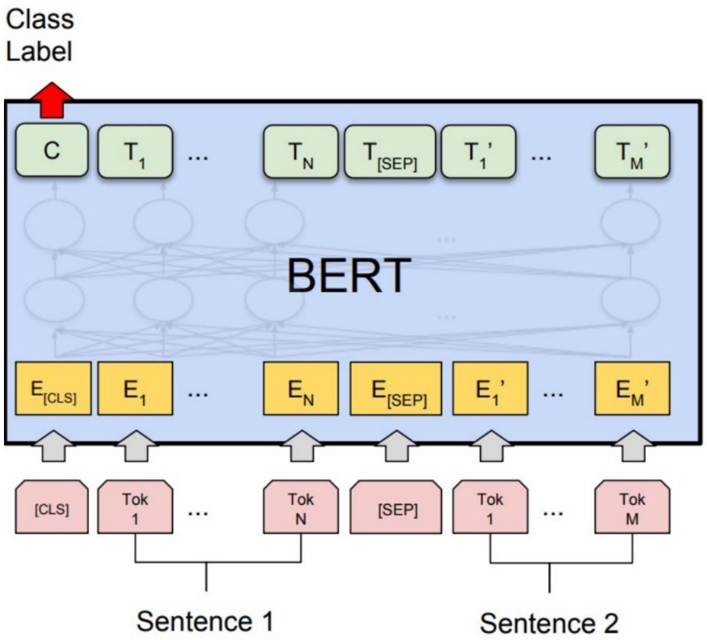


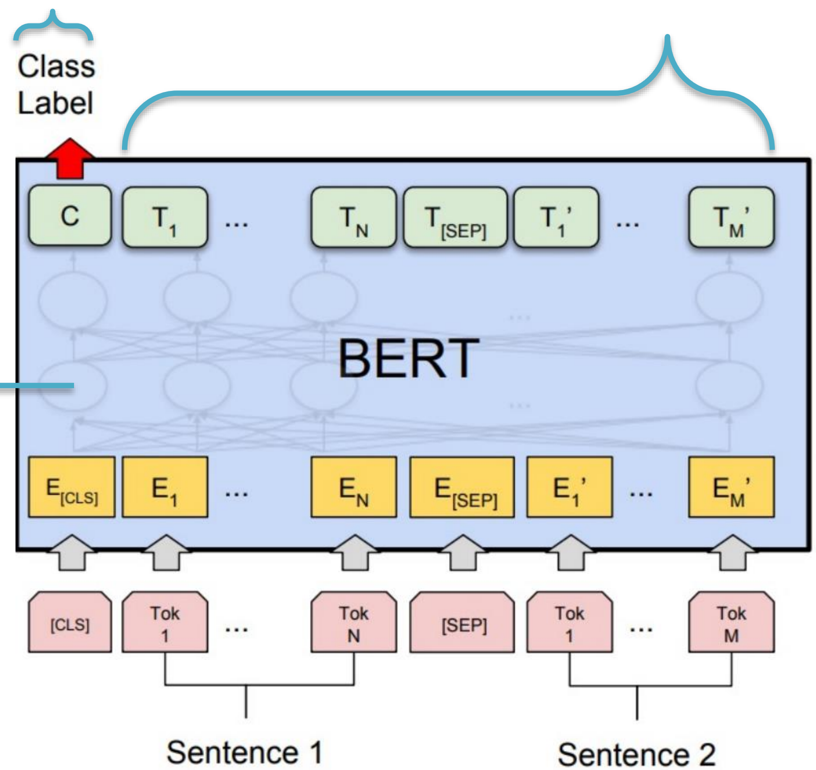
Figure 1: The Transformer - model architecture.



Model Architecture

Final hidden state
Sentence information embedded

Final hidden vector
Contextual Representation of each token



$C \in \mathbb{R}^H$, $T_i \in \mathbb{R}^H$ (H=768)

Encoder Layer
(A=12)

Multi-layer
Transformer encoder (L=12)

Input embedding

$L = number\ of\ layers$
$H = hidden\ size$
$A = number\ of\ self-attention\ heads$

M.IN.D Lab

Input Representation

- Input representation is able to represent both a single sentence and a pair of sentences
  - To make BERT handle variety of downstream tasks
    - **Sentence**: arbitrary span of contiguous text, rather than a linguistic sentence
    - **Sequence**: the input token sequences to BERT
      → may be a single sentence or two sentences

An aim is a goal or objective to achieve in life. In order to succeed in life, one must have a goal. My aim in life is to be a teacher.

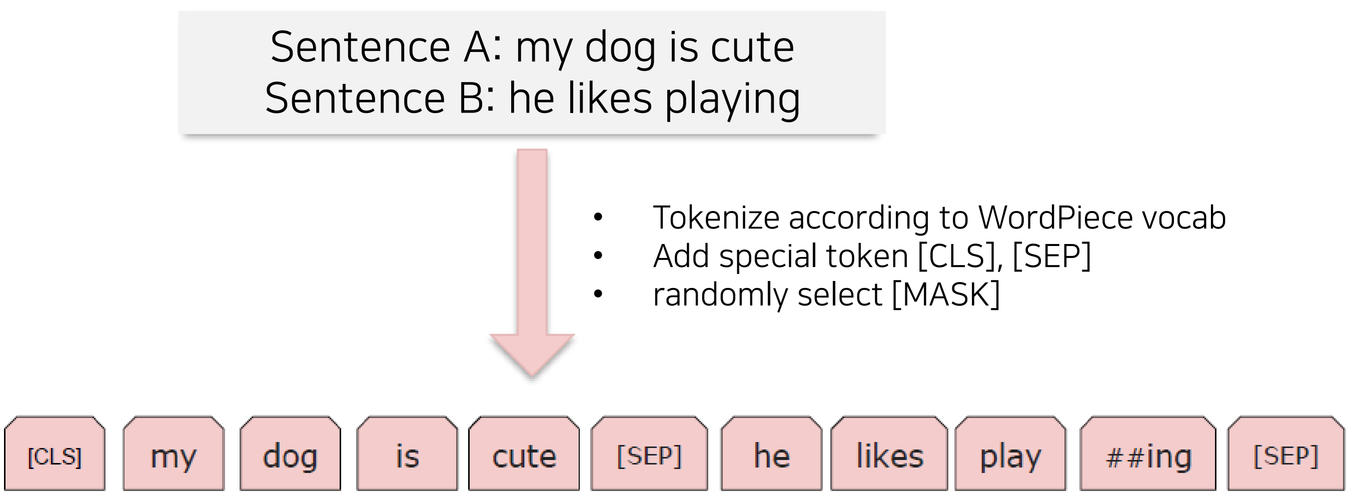
Sometimes we come across some forgetful persons in our surroundings. And some geniuses are also forgetful to some extent.

objective to achieve in life. In order to succeed in life, one must [SEP] sometimes we come across some forgetful persons

M.IN.D Lab

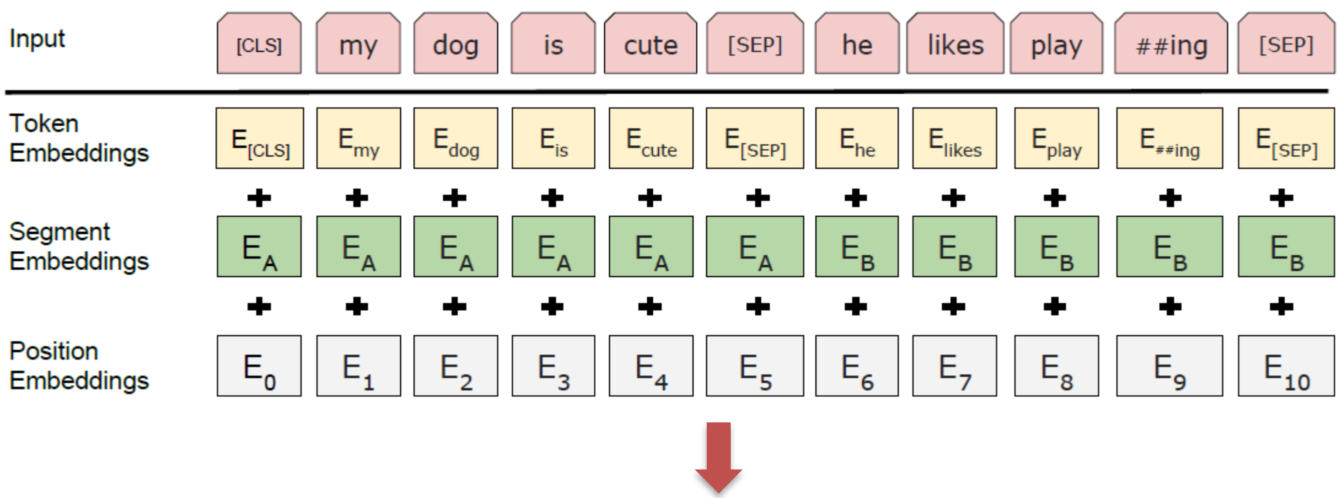
Input Representation

- Use special classification token
  - [CLS] – first token of every sequence
  - [SEP] – separation between two sentences
  - [MASK] – replaced for randomly selected token (for MLM)
- Use 30K WordPiece Tokenizer



Input Representation

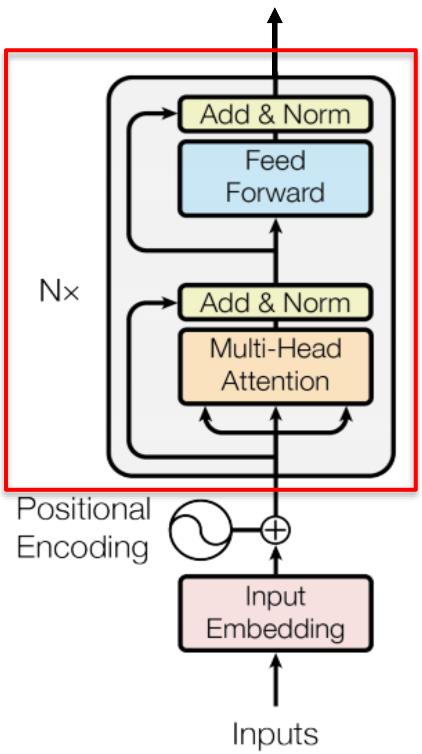
- Token embedding
  – Pre-trained WordPiece token embedding
- Segment embedding
  – *Learned* embedding indicating where it belongs
- Position embedding
  – Represent relative position of each token
  – Based on Sine, Cosine



Input Embedding

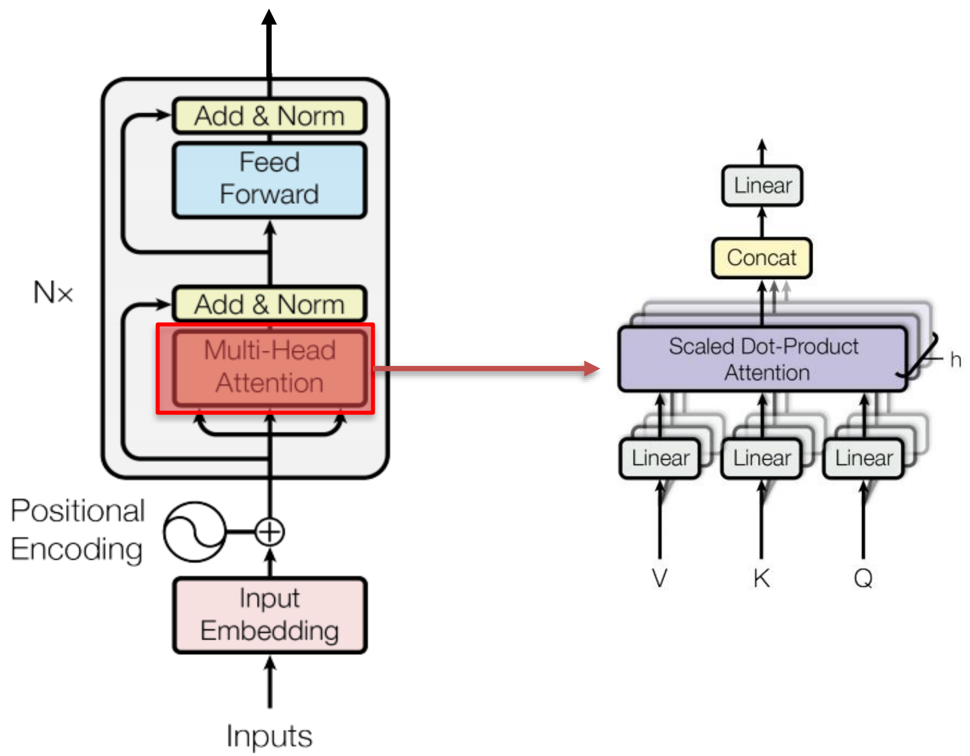
Model Architecture

- Encoder Block
  - Construct the meaning of the entire input sequence repeatedly (as many as L times)



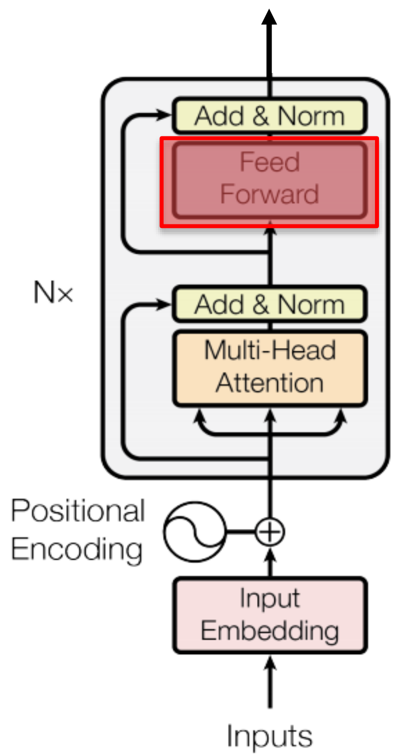
Model Architecture

- Multi-head attention
  - Compute attention H times with different weights
  - Concatenate results



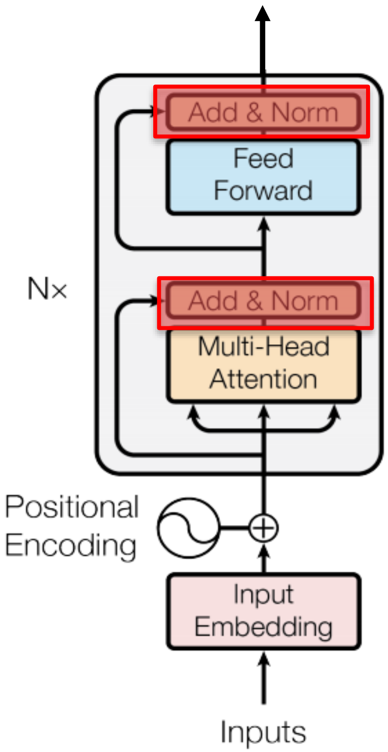
Model Architecture

- Position-wise Feed-forward Network
  - Two linear transformation
  - Apply GELU activation



Model Architecture

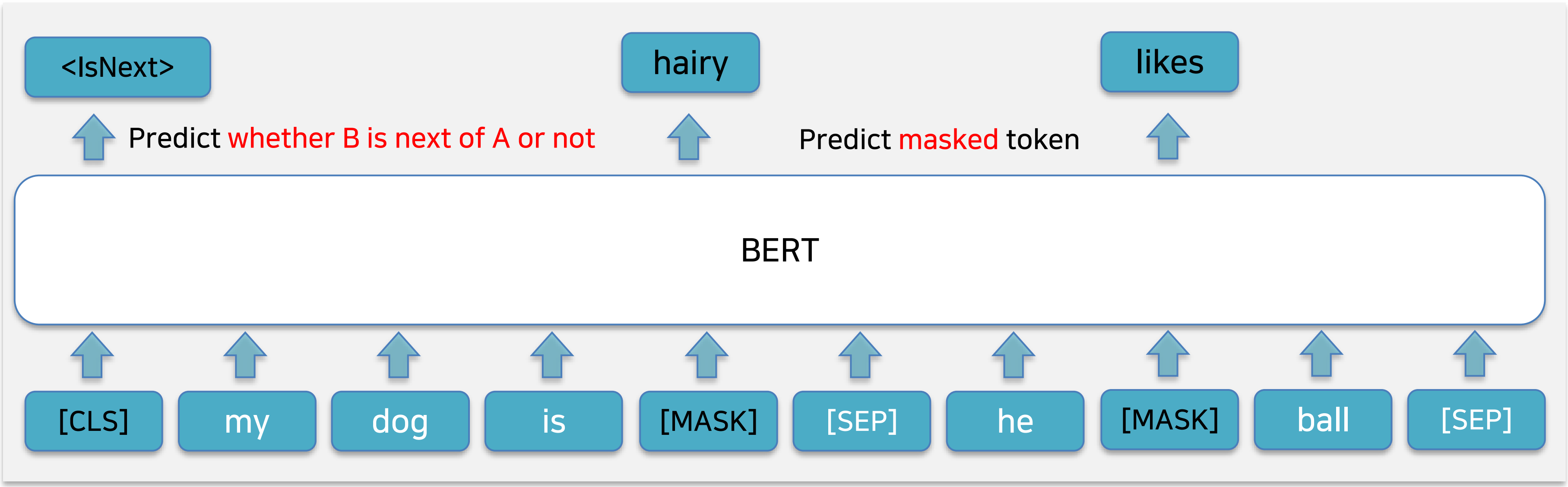
- Dropout, Add & Norm
  - Dropout FFN/Multi-head attention output with 10% prob
  - Add original representation
    - *Learn relationship with the rest of the tokens, but don't forget what we already learned!*
  - Apply LayerNorm
    - Improve the stability of network



M.IN.D Lab

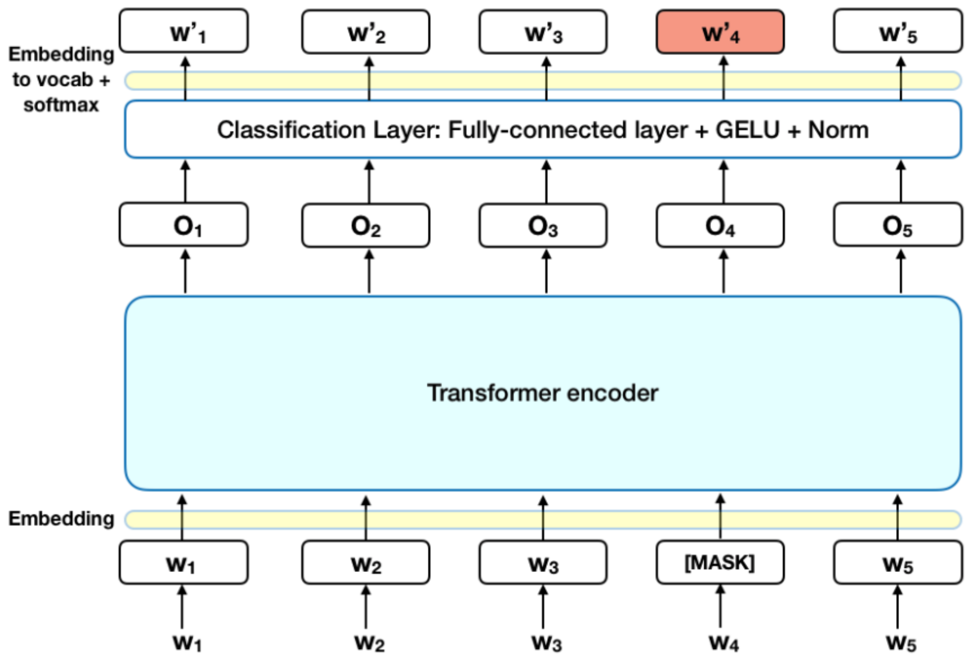
Pre-training BERT

- Pre-trained with two unsupervised task
  - Masked Language Model (MLM)
  - Next Sentence Prediction (NSP)
- Loss
  - *Mean MLM likelihood + Mean NSP likelihood*



Pre-training BERT

- Masked Language Model
  - Mask out *k*% of the input words (*k*=15%)
  - Predict the masked words



https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

Pre-training BERT

- But...
  - [Mask] token never seen at fine-tuning
- Solution
  - For selected 15% of words to predict,
    - 80%: Replace with [MASK]

      men went to the store → men went to the [MASK]
    - 10%: Replace with random token

      men went to the store → men went to the zoo
    - 10%: Left intact

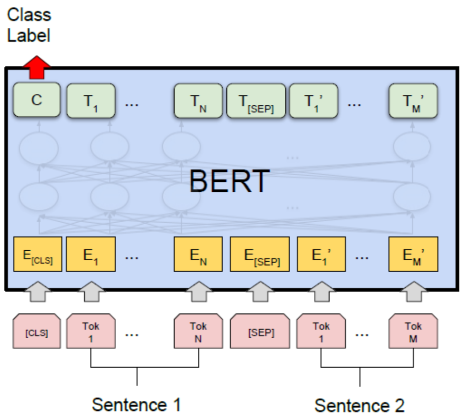
      men went to the store → men went to the store

Pre-training BERT

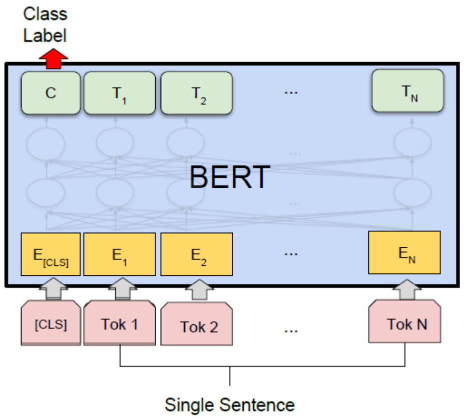
- Next Sentence Prediction (NSP)
  - Learn relationships between sentences
    → Beneficial to QA, NLI task
  - Predict whether sentence *B* is next sentence of sentence *A*
  - Selecting sentence *B*
    - 50%: Actual next sentence
    - 50%: Random sentence from corpus

```
Input = [CLS] the man [MASK] to the store [SEP]
        he bought a [MASK] of milk [SEP]
Label = IsNext

Input = [CLS] the man went to the [MASK] [SEP]
        chicken [MASK] are flight ##less birds [SEP]
Label = NotNext
```

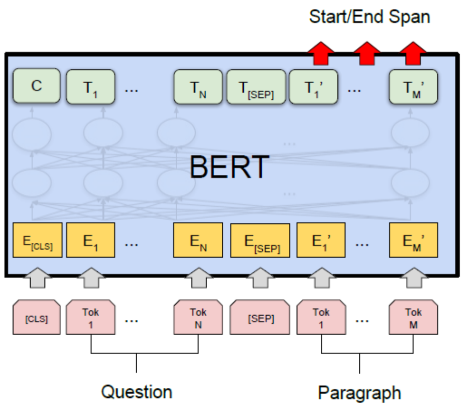
Fine-tuning BERT



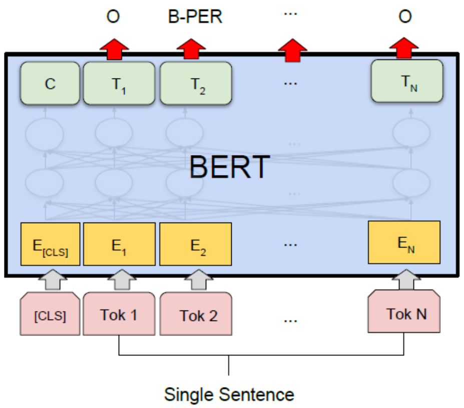
(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG



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



(c) Question Answering Tasks:
SQuAD v1.1



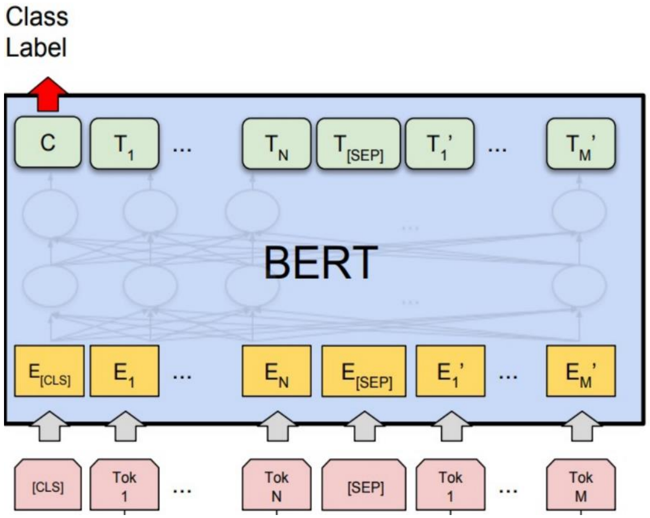
(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

Experiments

Model Architecture

	$BERT_{BASE}$	$BERT_{LARGE}$
L	12	24
A	12	16
H	768	1024
Total Params	110M	340M

$L = number\ of\ layers\ (Transformer\ blocks)$
$H = hidden\ size$
$A = number\ of\ self-attention\ heads$



Pre-training BERT

- Making input sequence
  - Data: BookCorpus (800M words) + WikiPedia (2,500M words)
  - Tokenized using 37,000 WordPiece tokens
  - Get 2 sentences
    - Combined length ≤ 512 tokens
    - 50%: random sentence, 50%: next sentence
  - batchsize=256 (256*512=128,000 tokens/batch)
- Training
  - 40 epochs
  - *gelu* activation
  - 4 days to complete with 16TPU($\mathbf{BERT_{BASE}}$), 64TPU($\mathbf{BERT_{LARGE}}$)
  - To speed up the training...
    - 90% of the steps: sequence with 128 tokens
    - 10% of the steps: sequence with 512 tokens

GLUE

		F1				Spearman correlations	F1		
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
$\mathrm{BERT}_{\mathrm{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\mathrm{BERT}_{\mathrm{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

* Accuracy reported if not specified.

- GLUE - The General Language Understanding Evaluation benchmark
  - Collection of diverse natural language understanding tasks
- $\mathrm{BERT}_{\mathrm{BASE}}$, same size with GPT, shows better results
- $\mathrm{BERT}_{\mathrm{LARGE}}$ beats $\mathrm{BERT}_{\mathrm{BASE}}$, and records state-of-the-art

SQUAD

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**

v1.1

System	Dev EM	Dev F1	Test EM	Test F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	86.3	89.0	86.9	89.5
#1 Single - MIR-MRC (F-Net)	-	-	74.8	78.0
#2 Single - nlnet	-	-	74.2	77.1
Published				
unet (Ensemble)	-	-	71.4	74.9
SLQA+ (Single)	-		71.4	74.4
Ours				
$\text{BERT}_{\text{LARGE}}$ (Single)	78.7	81.9	80.0	83.1

v2.0

- SQuAD - The Stanford Question Answering Dataset
  - Collection of 100K <question, answer> pairs
  - Given the question and paragraph that contains answer, model predict the answer text span in the paragraph (v2.0 has "no answer" label)
- Ensemble of $\text{BERT}_{\text{LARGE}}$ with data augmentation records state-of-the-art

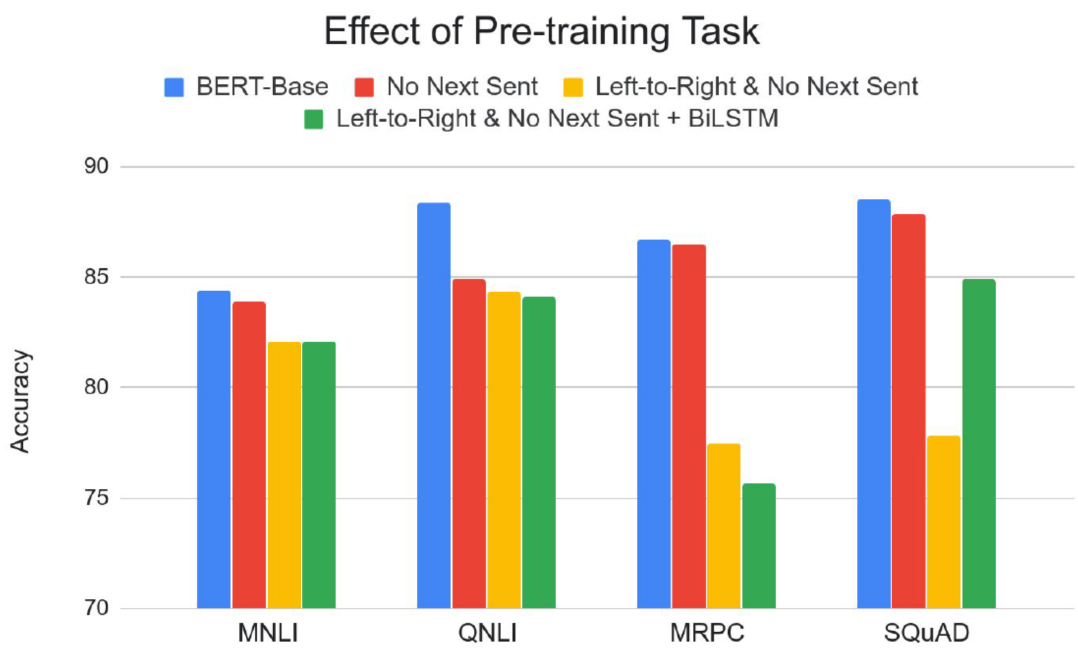
SWAG

System	Dev	Test
ESIM+GloVe	51.9	52.7
ESIM+ELMo	59.1	59.2
$\text{BERT}_{\text{BASE}}$	81.6	-
$\text{BERT}_{\text{LARGE}}$	**86.6**	**86.3**
Human (expert)†	-	85.0
Human (5 annotations)†	-	88.0

- SWAG – The Situations With Adversarial Generations (SWAG)
  - Given a sentence, choose the most plausible continuation among four choices
  - 113k sentence-pair completion examples that evaluate grounded commonsense inference
- $\text{BERT}_{\text{LARGE}}$ records state-of-the-art, outperforming human!

Ablation Study

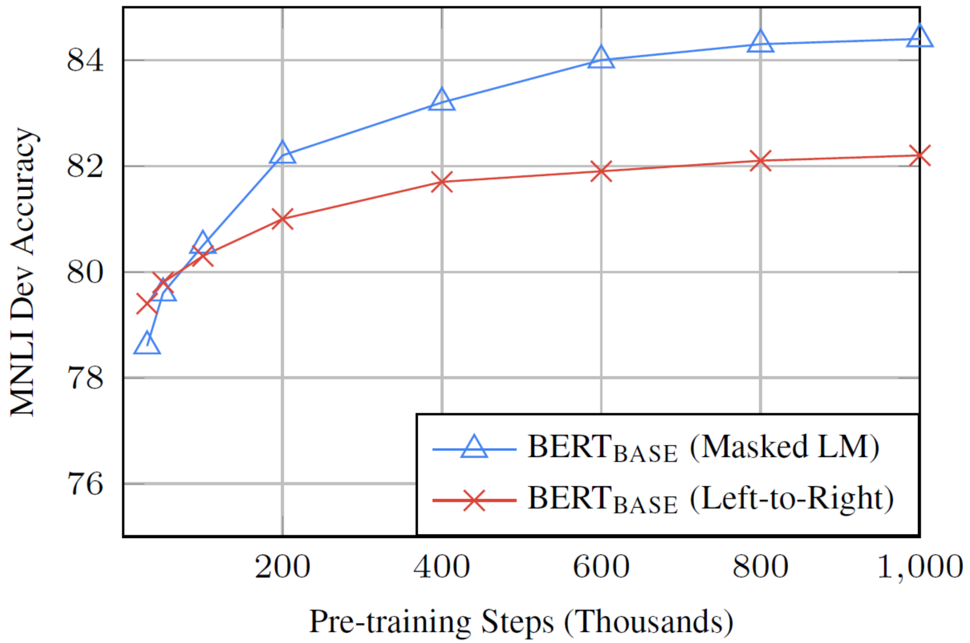
Effect of Pre-training Task



- Left-to-Right (LTR) model performs worse than MLM on all task
- Adding random initialized BiLSTM on top
  - Results better, but still far worse than MLM

Effect of Directionality & Training Time

Ablation Study



- MLM takes longer to converge - because MLM predict 15% instead of 100%
- But absolute results are much better

Effect of Model Size

Bigger model

Hyperparams				Dev Set Accuracy		
#L	#H	#A	LM (ppl)	MNLI-m	MRPC	SST-2
3	768	12	5.84	77.9	79.8	88.4
6	768	3	5.24	80.6	82.2	90.7
6	768	12	4.68	81.9	84.8	91.3
12	768	12	3.99	84.4	86.7	92.9
12	1024	16	3.54	85.7	86.9	93.3
24	1024	16	3.23	86.6	87.8	93.7

Better result

- Bigger is better even on small task (If the model is sufficiently pre-trained)
- Previous feature-based bi-LSTM study showed big model is not always good
- Authors hypothesize that when the model is directly fine-tuned on the downstream task w/ small additional params → model can benefit from large pre-trained representations

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional LSTM. In CoNLL.

Feature-based Approach

System	Dev F1	Test F1
ELMo (Peters et al., 2018a)	95.7	92.2
CVT (Clark et al., 2018)	-	92.6
CSE (Akbik et al., 2018)	-	**93.1**
Fine-tuning approach		
$\text{BERT}_{\text{LARGE}}$	96.6	92.8
$\text{BERT}_{\text{BASE}}$	96.4	92.4
Feature-based approach ($\text{BERT}_{\text{BASE}}$)		
Embeddings	91.0	-
Second-to-Last Hidden	95.6	-
Last Hidden	94.9	-
Weighted Sum Last Four Hidden	95.9	-
Concat Last Four Hidden	96.1	-
Weighted Sum All 12 Layers	95.5	-

- Tested on CoNLL-2033 NER dataset
- Feature-based approach
  - Use contextual embeddings as input of 2-layer 768-dim BiLSTM
  - Append classification layer
- Both feature-based approach & fine-tuning approach works well!

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional LSTM. In CoNLL.

Effect of Different Masking Strategy

Masking Rates			Dev Set Results		
MASK	SAME	RND	MNLI Fine-tune	NER Fine-tune	NER Feature-based
80%	10%	10%	84.2	95.4	94.9
100%	0%	0%	84.3	94.9	94.0
80%	0%	20%	84.1	95.2	94.6
80%	20%	0%	84.4	95.2	94.7
0%	20%	80%	83.7	94.8	94.6
0%	0%	100%	83.6	94.9	94.6

- Fine-tuning is robust to different masking strategy
- Random words 100% of time: degrades performance
- Masking 100% of time: problematic when applying feature-based approach

Conclusion

Contribution

- Demonstrate the importance of bidirectional pre-training for language representations
- Achieve SOTA performance on -
  - 11 NLP tasks
  - Both *sentence-level* and *token-level* tasks
  - Outperform many task-specific architecture
- Demonstrate pre-trained representations reduce the need for heavily-engineered task-specific architectures

Appendix

History of Natural Language Processing

2001	Neural language models	
2008	Multi-task learning	
2013	Word embeddings	Word2Vec, GloVe
2013	Neural networks for NLP	RNN, LSTM
2014	Sequence-to-sequence models	
2015	Attention	Transformer
2015	Memory-based networks	
2018	Pre-trained language models	ELMo, GPT, BERT

WordPiece Tokenizer

- Make vocabulary set by merging subword from letter
- Based on likelihood

Applying pre-trained model

- Strategies for applying pre-trained model to downstream tasks
  - Feature based (**ELMo**)
    - Use ELMo's output token as an embedding vector of additional task-specific architectures
  - Fine-tuning based (**GPT**)
    - Introduces minimal task-specific params
    - Trained on the downstream tasks by fine-tuning *all* pre-trained params

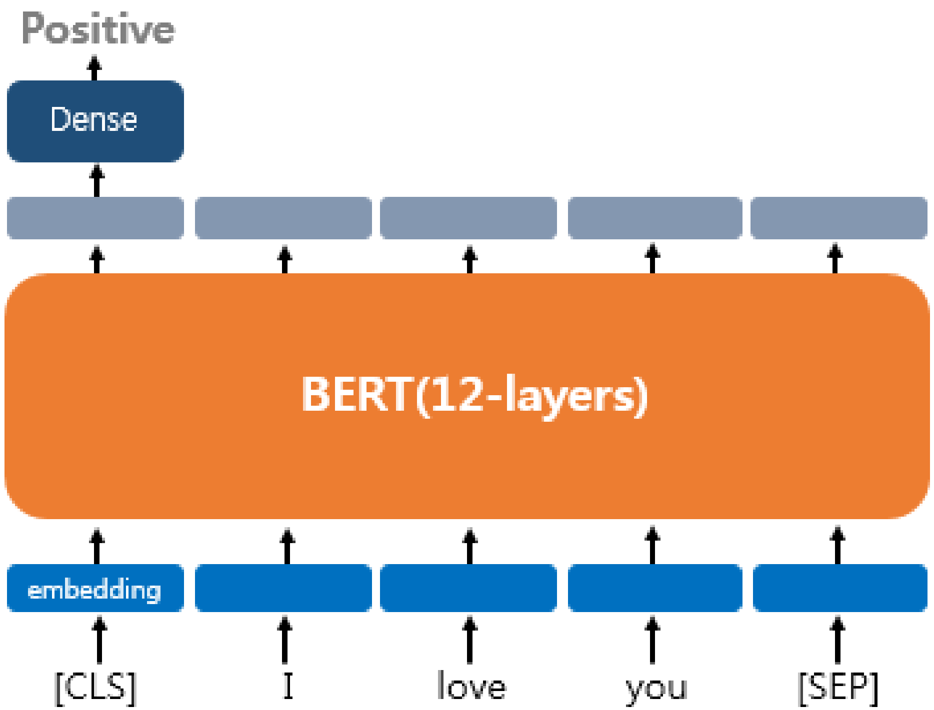
Positional Embedding

$$p_{i,j} = \begin{cases} \sin\left(\dfrac{i}{10000^{\frac{j}{d_{emb\_dim}}}}\right) & \text{if } j \text{ is even} \\ \cos\left(\dfrac{i}{10000^{\frac{j-1}{d_{emb\_dim}}}}\right) & \text{if } j \text{ is odd} \end{cases}$$

$$\begin{array}{c} \\ \begin{array}{c} Hello \\ , \\ how \\ are \\ you \\ ? \end{array} \end{array} \begin{array}{c} < \quad - \quad d_{emb\_dim} \quad - \quad > \\ \begin{pmatrix} \sin\left(\frac{0}{10000^{\frac{0}{emb_{dim}}}}\right) & \cos\left(\frac{0}{10000^{\frac{0}{emb_{dim}}}}\right) & \sin\left(\frac{0}{10000^{\frac{2}{emb_{dim}}}}\right) & \cos\left(\frac{0}{10000^{\frac{2}{emb_{dim}}}}\right) & \cdots \\ \sin\left(\frac{1}{10000^{\frac{0}{emb_{dim}}}}\right) & \cos\left(\frac{1}{10000^{\frac{0}{emb_{dim}}}}\right) & \sin\left(\frac{1}{10000^{\frac{2}{emb_{dim}}}}\right) & \cos\left(\frac{1}{10000^{\frac{2}{emb_{dim}}}}\right) & \cdots \\ \sin\left(\frac{2}{10000^{\frac{0}{emb_{dim}}}}\right) & \cos\left(\frac{2}{10000^{\frac{0}{emb_{dim}}}}\right) & \sin\left(\frac{2}{10000^{\frac{2}{emb_{dim}}}}\right) & \cos\left(\frac{2}{10000^{\frac{2}{emb_{dim}}}}\right) & \cdots \\ \sin\left(\frac{3}{10000^{\frac{0}{emb_{dim}}}}\right) & \cos\left(\frac{3}{10000^{\frac{0}{emb_{dim}}}}\right) & \sin\left(\frac{3}{10000^{\frac{2}{emb_{dim}}}}\right) & \cos\left(\frac{3}{10000^{\frac{2}{emb_{dim}}}}\right) & \cdots \\ \sin\left(\frac{4}{10000^{\frac{0}{emb_{dim}}}}\right) & \cos\left(\frac{4}{10000^{\frac{0}{emb_{dim}}}}\right) & \sin\left(\frac{4}{10000^{\frac{2}{emb_{dim}}}}\right) & \cos\left(\frac{4}{10000^{\frac{2}{emb_{dim}}}}\right) & \cdots \\ \sin\left(\frac{5}{10000^{\frac{0}{d_{emb_{dim}}}}}\right) & \cos\left(\frac{5}{10000^{\frac{0}{emb_{dim}}}}\right) & \sin\left(\frac{5}{10000^{\frac{2}{emb_{dim}}}}\right) & \cos\left(\frac{5}{10000^{\frac{2}{emb_{dim}}}}\right) & \cdots \end{pmatrix} \end{array}$$

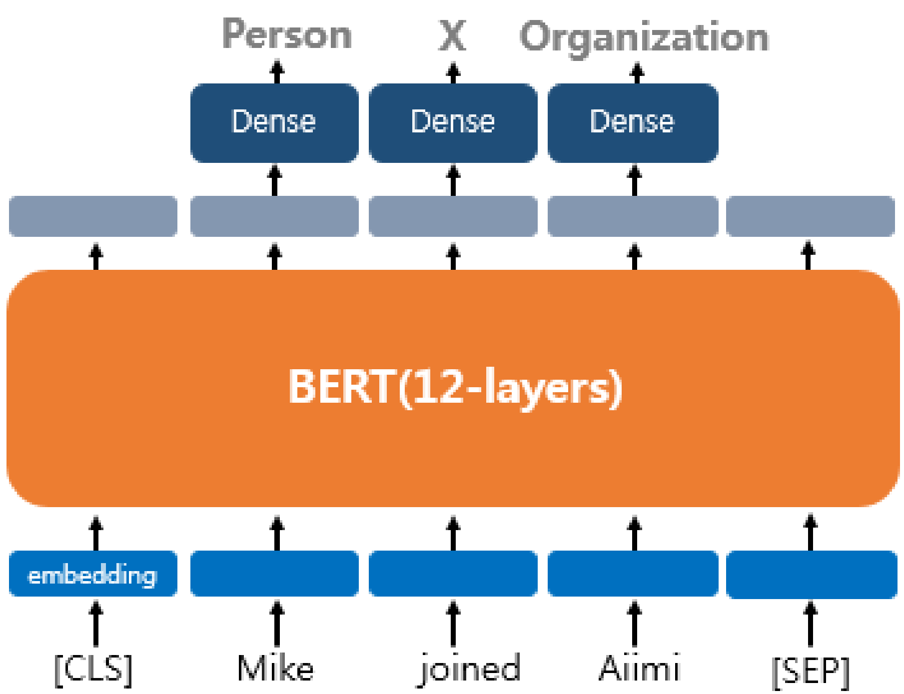
Fine-tuning BERT

- Single Text Classification
  - SST-2, CoLA



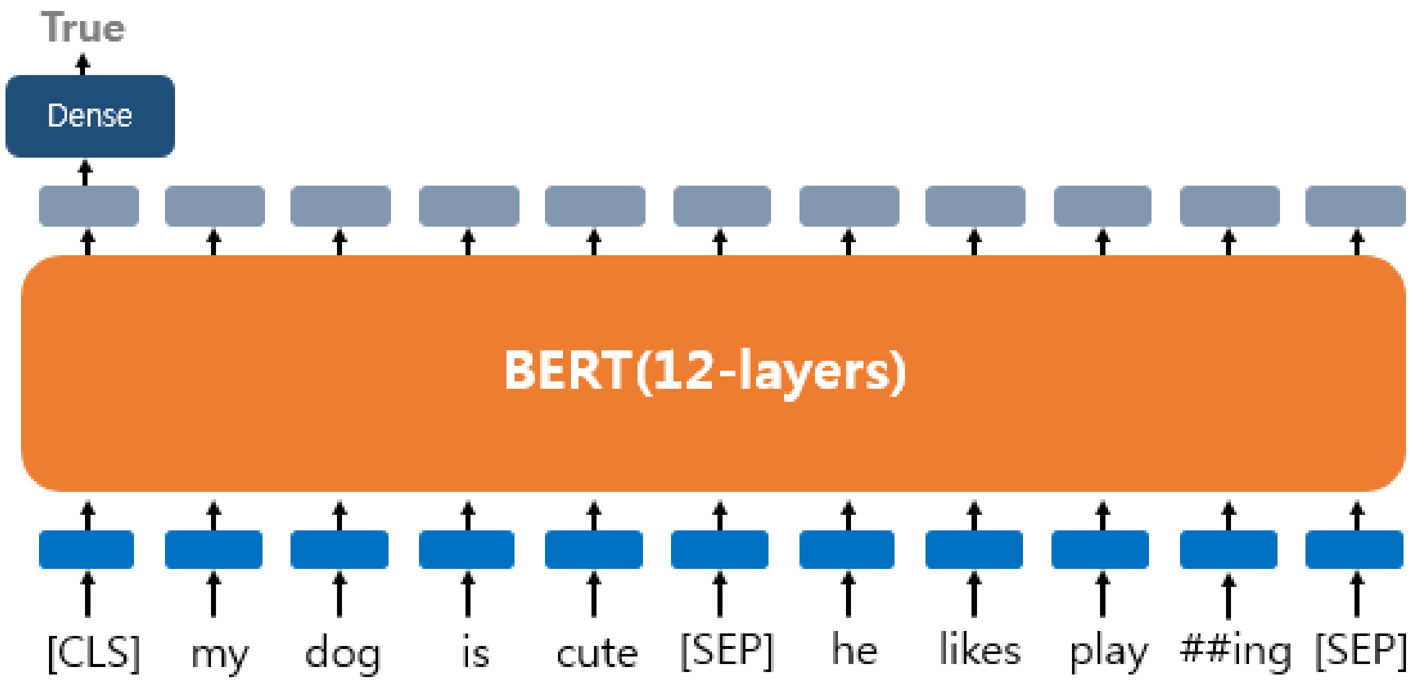
Fine-tuning BERT

- Single Text Tagging
  - CoNLL-2033 NER



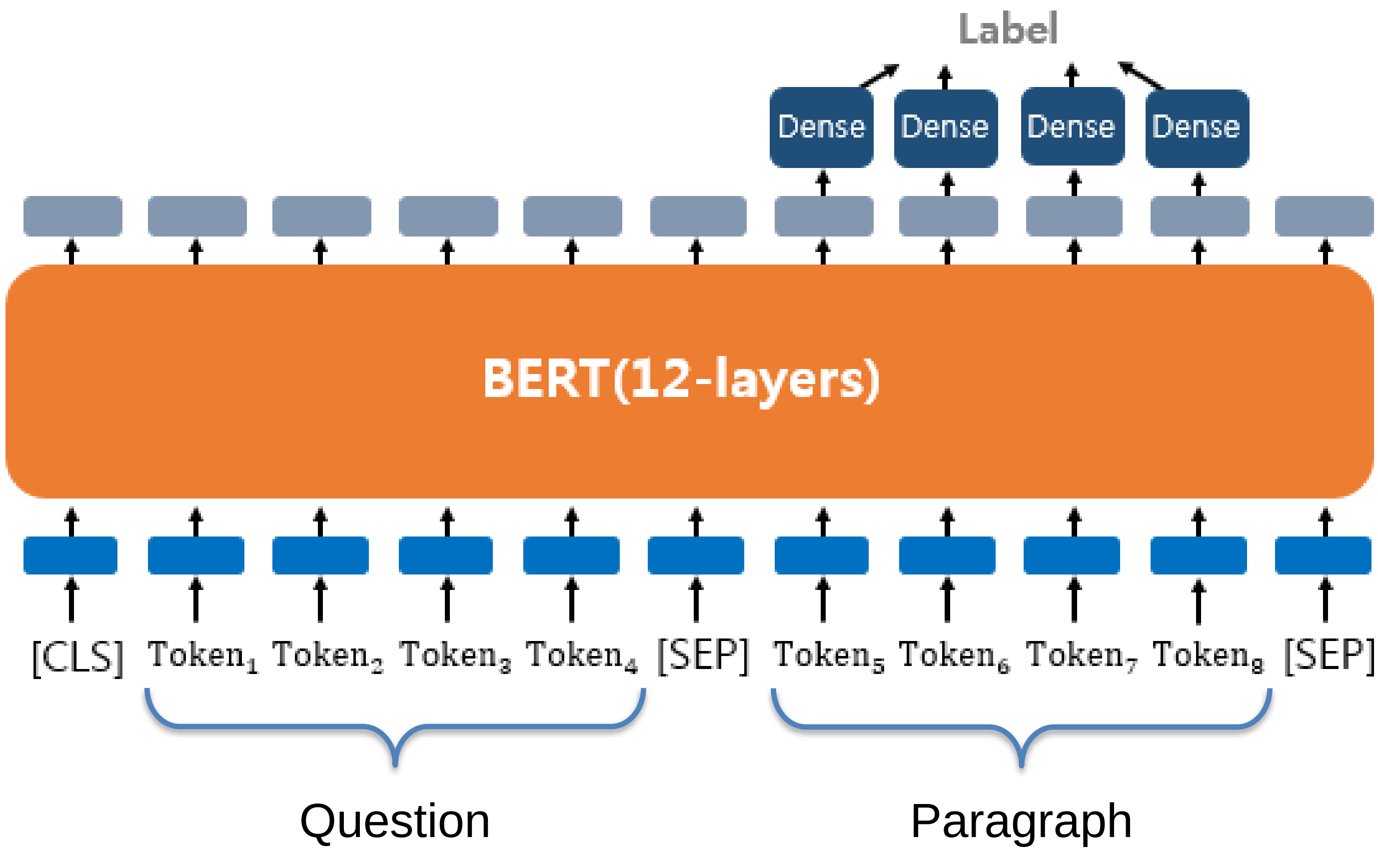
Fine-tuning BERT

- Text Pair Classification or Regression
  - MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



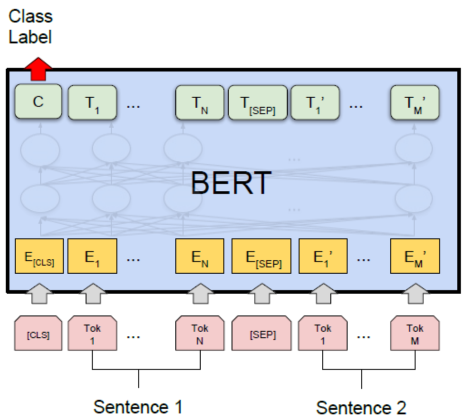
Fine-tuning BERT

- Question Answering
  - SQuAD v1.1



M.IN.D Lab

Fine-tuning BERT: GLUE



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

- Introduce classification layer

- Final hidden vector $C \in \mathbb{R}^H$
- Weights $W \in \mathbb{R}^{K \times H}$, where K is number of labels

- Compute standard CE loss

$$\text{i.e., } \log(\text{softmax}(CW^T)).$$

Fine-tuning BERT: SQuAD v1.1

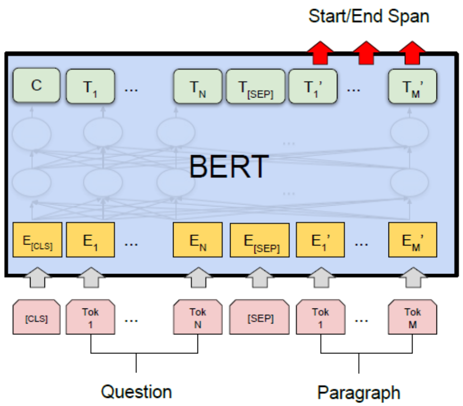
- Introduce start vector $S \in \mathbb{R}^H$, end vector $E \in \mathbb{R}^H$

- Probabilty of word i being the start of the answer span is computed as a dot product between T_i and S
- Softmax over all of the words in paragraph

$$P_i = \frac{e^{S \cdot T_i}}{\sum_j e^{S \cdot T_j}}$$

- The score of candidate span from i to j defined as $S \cdot T_i + E \cdot T_j$
- Find maximum scoring span

- Sum of log-likelihoods of correct start&end position



(c) Question Answering Tasks:
SQuAD v1.1

References

- BERT
  - Blogs
    - https://medium.com/dissecting-bert
    - https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270
    - https://docs.likejazz.com/bert/
    - https://wikidocs.net/115055
    - https://eatchu.tistory.com/31
  - Lectures
    - Stanford CS224N – BERT and other pre-trained LMs
      https://youtu.be/knTc-NQSjKA
    - https://youtu.be/30SvdoA6ApE
    - https://youtu.be/IwtexRHoWG0
  - Code
    - https://github.com/codertimo/BERT-pytorch
- Transformer
  - http://nlp.seas.harvard.edu/2018/04/03/attention.html
- NLP Tasks
  - https://huffon.github.io/2019/11/16/glue/