Session #3: BERT

Tuesday, Sept 6

CSCI 601.771: Self-supervised Statistical Models

The New York Times

An A.I.-Generated Picture Won an Art Prize. Artists Aren't Happy.

"I won, and I didn't break any rules," the artwork's creator says.

Ice Breaker

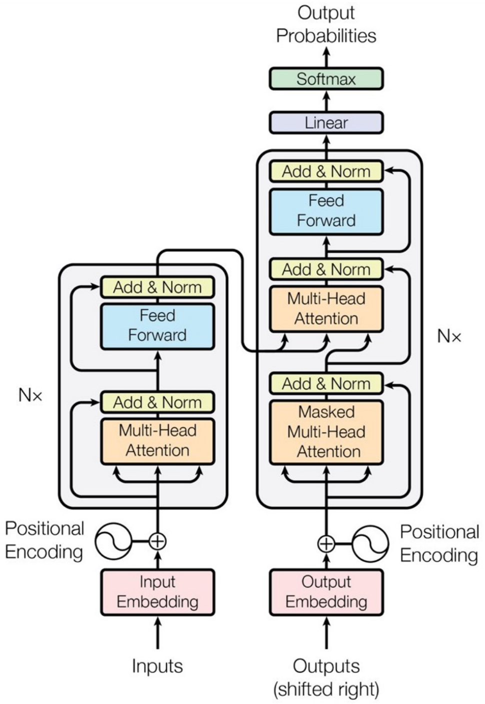
What is one question you have about this paper?

What is one thing you want to hear more about?

Recap

Transformers

- An instance of Auto-Encoder (an unsupervised learning technique)
    - Encoder: Mapping from X->Z, aiming to understand the feature representation Z
    - Decoder: Mapping from Z->X, inverse mapping for reconstruction
- An encoder-decoder architecture built with attention modules



Ashish, 2017. Attention Is All You Need

Problems and motivation

Problems:

- Previous NLP pretrained techniques are unidirectional, limit context understanding
- Bi-directional LSTM is slow and not actually bidirectional (only concat left-to-right and right-to-left, leading to lost context understanding)

Motivation:

- How to build a fast pretrained model, bidirectional, and preserve context understanding

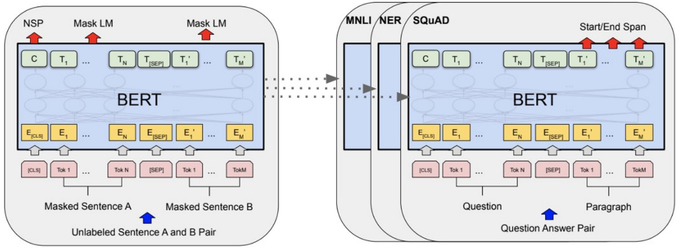
=> Jacob, 2018. Bidirectional Encoder Representation from Transformer (**BERT**):

- A stack of multiple transformer encoders
- BERT is a **fast bidirectional** model and **preserves context understanding**

Method

Overview of two steps of training BERT:

- Pre-training:
  - Goal: **Understanding** features in representation space
  - Trains model on unlabeled data over different pre-training tasks (**Self-supervised learning**)
- Fine-tuning:
  - Goal: Make pre-trained model **usable** in **downstream tasks**
  - Initialized with pre-trained model parameters
  - Fine-tuned model parameters using labeled data from downstream tasks



Pre-training Fine-Tuning

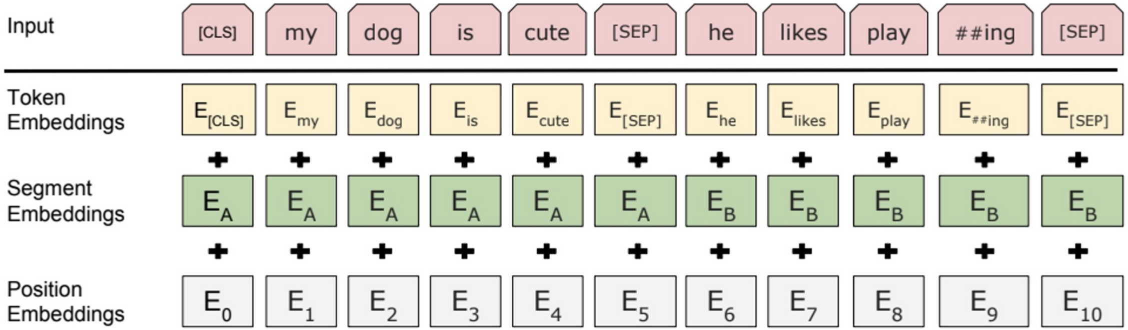
✍: Ha, Isabel

Jacob, 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Method

Input Embeddings:

- **Token** Embeddings: pre-trained token vocabs (“WordPleces”: 30K vocabs/tokens)
    - [CLS]: token beginning sentence, [SEP]: token ending sentence, [PAD]: padding token
- **Segment** Embedding: sentence number encoder to vectors
- **Position** Embedding: position of words within that sentence
- => Preserve **ordering** sentence inputs for BERT => Robust across downstream tasks



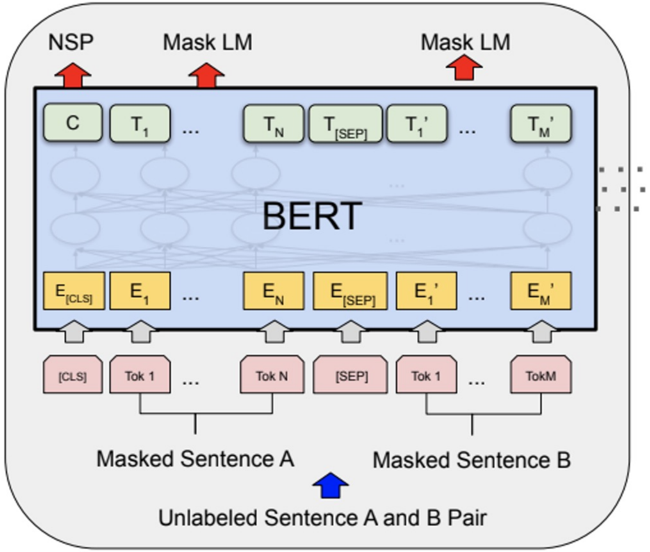
Jacob, 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

✍: Ha, Isabel

Method

Pre-training BERT:

- Task #1: **Masked Language Model**
    - Inputs: The [Mask1] Hopkins University is located in [Mask2] city (E)
    - Outputs: [Mask1] = Johns, [Mask2] = Baltimore (C, T)
    - => Helps understand bi-directional context
- Task #2: **Next Sentence Prediction**
    - Inputs:
        - A: Johns Hopkins is a university (E)
        - B: It is located in Baltimore city (E)
    - Outputs:
        - Yes: Sentence B follows sentence A (C = 1)
    - => Help understand context across different sentences
- Jointly training by soft-max layer and cross-entropy



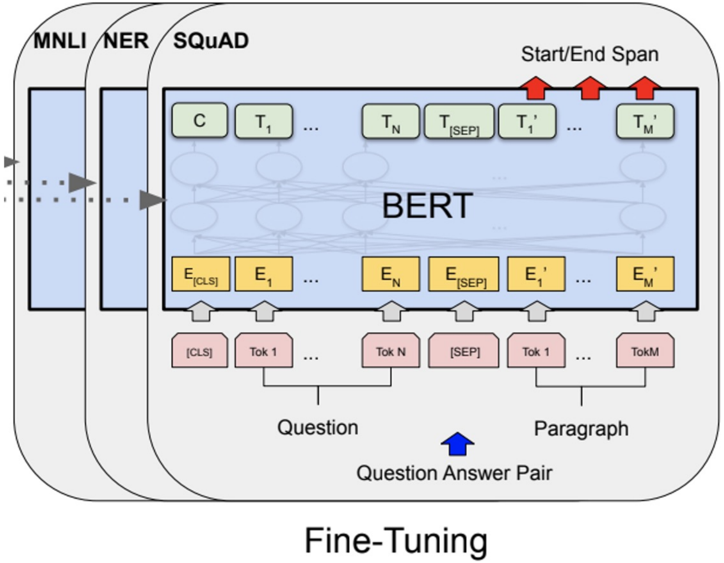
Pre-training

Jacob, 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Method

Fine-tuning BERT:

- Plugs appropriate inputs/outputs into BERT and fine-tuning all params end-to-end
- Example in Questions Answering:
    - Inputs: Question, Paragraph
    - Outputs: start and end words that encapsulate the answer



Jacob, 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

✍: Ha, Isabel

Experiments

Experimental Settings:

- Models:
    - **BERT_base** (#transformer blocks L = 12, #hidden size H = 768, #self-attention heads A = 12): 110M params
    - **BERT_large** (L =24, H = 1024, A = 16): 340M params
- Fine-tuning on 11 NLP tasks over GLUE, SQuAD v1.1, SQuAD v2.0, SWAG dataset

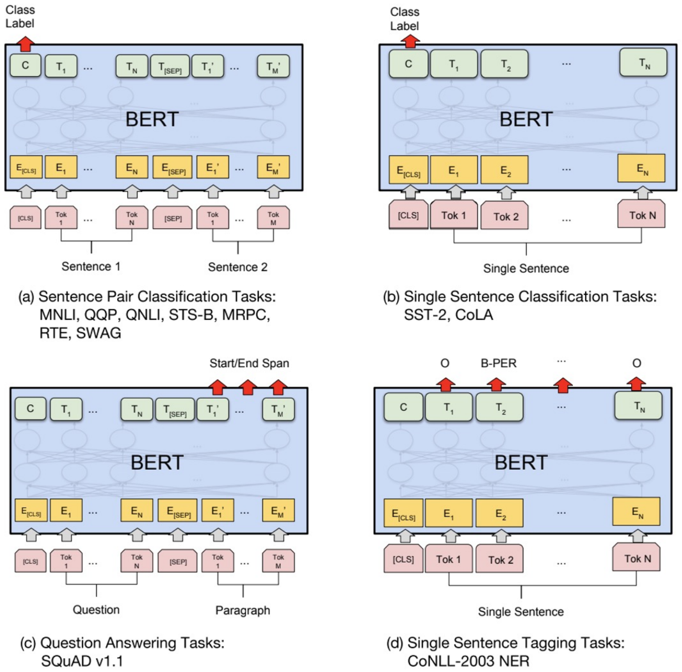


Figure 4: Illustrations of Fine-tuning BERT on Different Tasks.

Jacob, 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

✍: Ha, Isabel

Results

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**

Table 1: GLUE Test results, scored by the evaluation server (`https://gluebenchmark.com/leaderboard`).

Jacob, 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

System	Dev EM	Dev F1	Test EM	Test F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
Published				
BiDAF+ELMo (Single)	-	85.6	-	85.8
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 1.1 results. The BERT ensemble is 7x systems which use different pre-training checkpoints and fine-tuning seeds.

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

Table 3: SQuAD 2.0 results. We exclude entries that use BERT as one of their components.

System	Dev	Test
ESIM+GloVe	51.9	52.7
ESIM+ELMo	59.1	59.2
OpenAI GPT	-	78.0
$\text{BERT}_{\text{BASE}}$	81.6	-
$\text{BERT}_{\text{LARGE}}$	**86.6**	**86.3**
Human (expert)†	-	85.0
Human (5 annotations)†	-	88.0

Table 4: SWAG Dev and Test accuracies. †Human performance is measured with 100 samples, as reported in the SWAG paper.

Jacob, 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

✍: Ha, Isabel

Ablation Studies

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	-

BERT is effective for both fine-tuning and feature-based approaches

Jacob, 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Ablation Studies

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

The deeper model, the better generalization

Tasks	Dev Set MNLI-m (Acc)	QNLI (Acc)	MRPC (Acc)	SST-2 (Acc)	SQuAD (F1)
$\text{BERT}_{\text{BASE}}$	84.4	88.4	86.7	92.7	88.5
No NSP	83.9	84.9	86.5	92.6	87.9
LTR & No NSP	82.1	84.3	77.5	92.1	77.8
+ BiLSTM	82.1	84.1	75.7	91.6	84.9

Pre-training Tasks matters

Jacob, 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

✍: Ha, Isabel

Summary

- Based on Transformer, BERT is a **fast** and **bidirectional pre-trained** model for NLP tasks
- Training BERT includes 2 steps:
  - Pretraining: use **self-supervised** techniques to build good representation space
  - Fine-tuning: make use pre-trained representation for downstream tasks
- BERT archives SOTA across many tasks:
  - Proving its **context understanding** in NLP
  - Showing a good pre-trained encoder for downstream tasks

Table of contents (Reviewers)

Legends

🟢 Positive Point
🟡 Critical Point

Brief Summary of BERT

What is BERT?

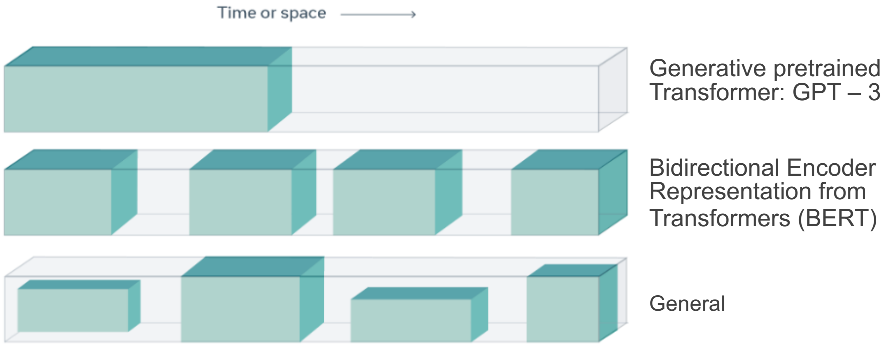
A predictive language Model that takes into account bi-directional context.

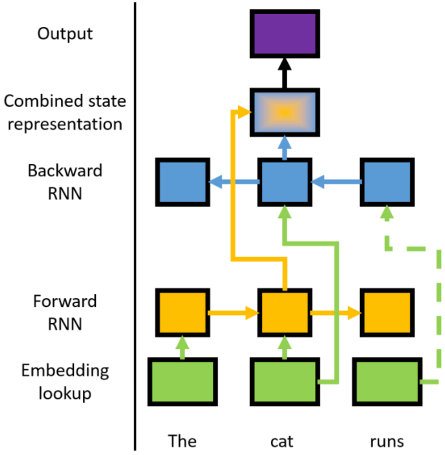
What is the motivation behind BERT?

Include deep bi-directional context in learning language.

How ?

Masked Language Modelling



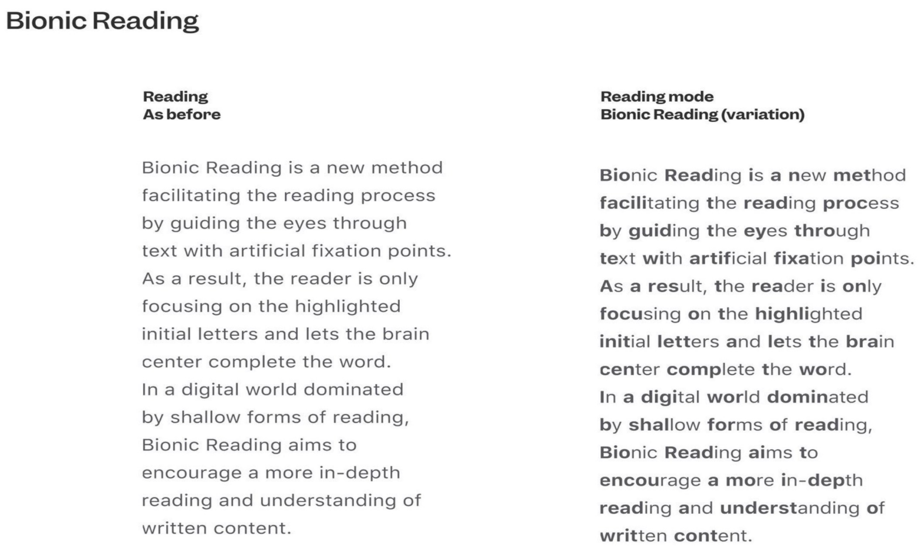


Problem in defining bi-directional context with models that are defined to predict next given past.

Reviewers #1 Comments

Background

- Humans learn/comprehend language through reading.
- The research indicates brain reads faster when pseudo masked
- UNIDIRECTIONAL!!



https://bionic-reading.com/

Comments

- 🟢 Hence, BERT is loosely doing something similar to how brain does it.
- 🟡 BUT it used LTR and RTL?
- Does our brain look at the future context while understanding language?

Just, Marcel Adam and Patricia A. Carpenter. “A theory of reading: from eye fixations to comprehension.” *Psychological review* 87 4 (1980): 329-54 .

Reviewers #1 Comments

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**

Table 1: GLUE Test results, scored by the evaluation server (https://gluebenchmark.com/leaderboard). 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.[8] BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

Subset: wnli

Split: train

sentence1 (string)	sentence2 (string)
I stuck a pin through a carrot. When I pulled the pin out, it had a hole.	The carrot had a hole.

[8] See (10) in https://gluebenchmark.com/faq.

12. I get weird results for QQP or WNLI. What gives?

QQP: There is a difference in the dev and test distributions that likely explains discrepancies observed between scores for the two. WNLI: The train/dev split for WNLI is correct, but turns out to be somewhat adversarial: when two examples contain the same sentence, that usually means they'll have opposite labels. The train and dev splits may share sentences, so if a model has overfit the training set, it may get worse than chance accuracy on WNLI on the dev set. Additionally, the test set has a different label distribution than the train and dev sets.

- The overall performance of BERT has good improvement!
- BUT curious as to why BERT and GPT never mentioned WNLI task results.
  - they claim based on the FAQs that WNLI did not perform well because of the dataset mismatch BUT they mention QQP.
  - Curious about the LM performance on the WNLI task. Is the bi-directional context confusing the model for the WNLI? (Cause in WNLI the LTR plays a major role)

Reviewers #2 Comments

- 🟡 Why not a more contextually heavy task such as the Argument Reasoning Comprehension Task(ARCT)

Unit	**Text**
Reason	Cooperating with Russia on terrorism ignores Russia's overall objectives.
Claim	Russia cannot be a partner.
Warrant0	Russia has the same objectives of the US.
Warrant1	Russia has the opposite objectives of the US.
Reason	Economic growth needs innovation.
Claim	3-D printing will change the world.
Warrant0	There is no innovation in 3-d printing since it's unsustainable.
Warrant1	There is much innovation in 3-d printing and it is sustainable.
Reason	College students have the best chance of knowing history.
Claim	College students' votes do matter in an election.
Warrant0	Knowing history doesn't mean that we will repeat it.
Warrant1	Knowing history means that we won't repeat it.

ARCT snippet

Reviewers #2 Comment

word-piece tokenizer concept

Reviewers #2 Comments

- 🟡 Over parameterized and no analysis on the inference time
- 🟡 Effects of Increase/decrease in number of attention heads and its effects on the accuracy of the NLP tasks.

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

Table 6: Ablation over BERT model size. #L = the number of layers; #H = hidden size; #A = number of attention heads. "LM (ppl)" is the masked LM perplexity of held-out training data.

Reviewers #3 Comments

🟡 Masked Language and Masking Procedure

- We may need to re-evaluate how we learn language as humans
- Mask too little and it will require more learning of the context by the model
- Mask a lot and the model will miss the context of the sentence

Reviewers #3 Comments

- Unknown effect of varied % of masking rate

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

Table 8: Ablation over different masking strategies.

Reviewers #3 Comments

🟡 Fine Tuning BERT

- Lack of specific language data set
- Not enough data analysis and diversity for a fine tuning task

Conclusion (Gist of other Comments)

🟢	🟡
High Performance.	Very compute Intensive.
Authors acknowledge the disadvantages of the pre-training approach and perform experiments on feature extraction type setting.	Not practical in real-world applications, quite costly. It is slow to train because it is big and there are a lot of parameters to update.
Does not have the problem of ”see-itself” unlike other models.	Bi-directional context is deep, hence not explicitly weighted for LTR and RTL. Given a task the model cannot explain what is more important LTR or RTL.
Deep Bi-direction better than explicit LTR and RTL context as other layers can share this information.	Defined “sentence” does not make sense linguistically and is arbitrary.

Testing a Finetuned Question-Answer BERT

Here, I tested the capabilities of a BERT model fine-tuned on SQuAD, a question-answer dataset that was mentioned in the original BERT paper.

```
context = """ The Johns Hopkins University (Johns Hopkins, Hopkins, or JHU) is a private research university in Baltimore, Maryland. Founded in 1876, Johns Hopkins is the oldest re

The university was named for its first benefactor, the American entrepreneur and Quaker philanthropist Johns Hopkins.[10] Hopkins' $7 million bequest to establish the university wa

queries = ["When was Johns Hopkins founded?",
           "Who is it named after?",
          #  "which university has the highest research expenditure?"
           "How many noble laureates"
          ]

for q in queries:
  give_an_answer(context,q)

```

```
Question: When was Johns Hopkins founded?
Prediction: 1876
Question: Who is it named after?
Prediction: johns hopkins
Question: How many noble laureates
Prediction: it consistently ranks among the most prestigious universities in the united states and the world
```

Testing a Finetuned Question-Answer BERT

Testing with a math-heavy text

```
context = """ The principal components of a collection of points in a real coordinate space are a sequence of {displaystyle p}p unit vectors, where the {displaystyle i}i-th vector

In data analysis, the first principal component of a set of {displaystyle p}p variables, presumed to be jointly normally distributed, is the derived variable formed as a linear com

queries = [
           "When is PCA used?",
           "What is PCA?",
           "Do you think PCA is useful?"
          ]

for q in queries:
  give_an_answer(context,q)
```

```
Question: When is PCA used?
Prediction: when many of the variables are highly correlated with each other
Question: What is PCA?
Prediction: principal component analysis
Question: Do you think PCA is useful?
Prediction: pca is most commonly used when many of the variables are highly correlated with each other and it is desirable to reduce their number to an independent set
```

Testing a Finetuned Question-Answer BERT

Testing with more information on the same questions

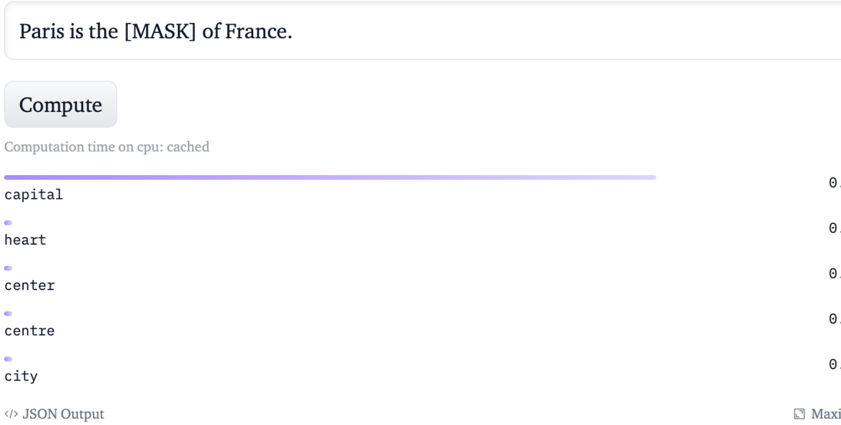
```
[22]
context = """ he principal components of a collection of points in a real coordinate space are a sequence of {\displaystyle p}p unit vectors, where the {\displaystyle i}i-th vector

In data analysis, the first principal component of a set of {\displaystyle p}p variables, presumed to be jointly normally distributed, is the derived variable formed as a linear cc

PCA is used in exploratory data analysis and for making predictive models. It is commonly used for dimensionality reduction by projecting each data point onto only the first few pr

For either objective, it can be shown that the principal components"""


queries = [
           "When is PCA used?",
           "What is PCA?",
           "Do you think PCA is useful?"
          ]


for q in queries:
  give_an_answer(context,q)
```

```
Question: When is PCA used?
Prediction: in exploratory data analysis and for making predictive models
Question: What is PCA?
Prediction: principal component analysis ( pca ) is the process of computing the principal components and using them to perform a change of basis on the data
Question: Do you think PCA is useful?
Prediction: pca is used in exploratory data analysis and for making predictive models
```

BERT AS MASKED LANGUAGE MODEL

What is a Masked Language Model?

- **The masked language model** enables bidirectional learning of text by masking (hiding) a word in a sentence.
- In this scenario, forcing BERT to use words on either side of the masked word in both directions to predict the masked word.

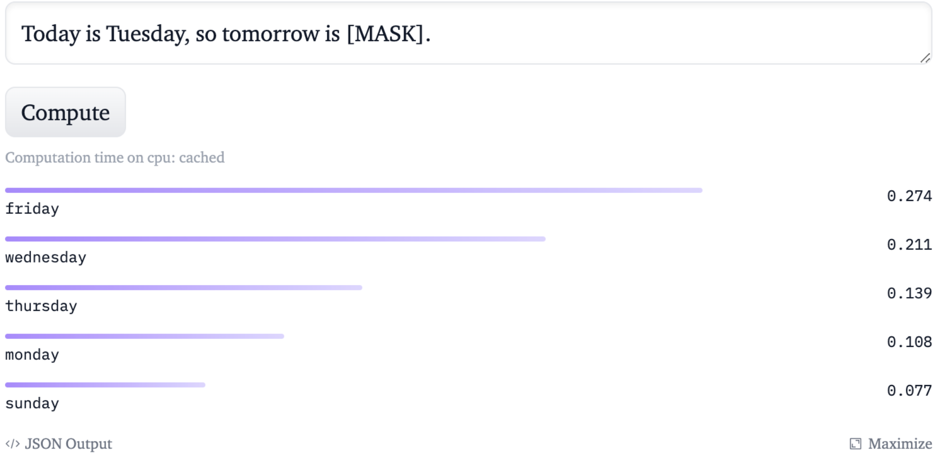


: Haoyue Guan and Ammar Ahmed Pallikonda Latheef

BERT AS MASKED LANGUAGE MODEL

What is a Masked Language Model?

- **The masked language model** enables bidirectional learning of text by masking (hiding) a word in a sentence.
- In this scenario, forcing BERT to use words on either side of the masked word in both directions to predict the masked word.



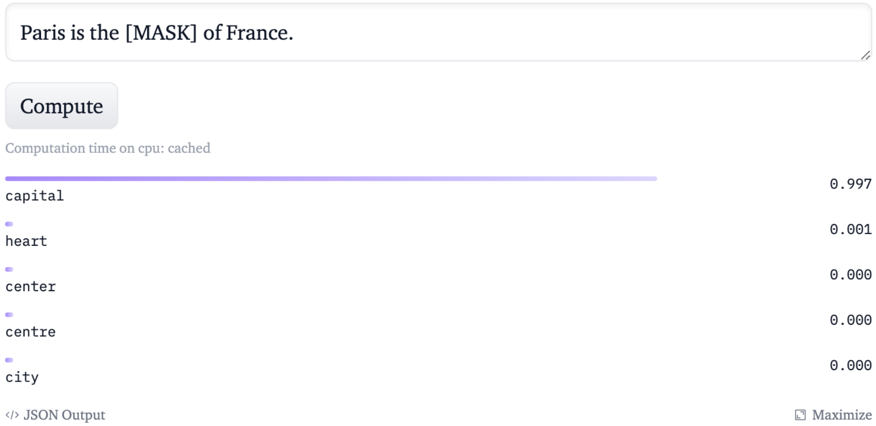
BERT AS MASKED LANGUAGE MODEL

- BERT was specifically trained on Wikipedia (~2.5B words) and Google's BooksCorpus (~800M words)
- A massive dataset of 3.3 Billion words has contributed to the training of BERT
- The prediction of the masked word lacks reasoning

BERT AS MASKED LANGUAGE MODEL

What is a Masked Language Model?

- **The masked language model** enables bidirectional learning of text by masking (hiding) a word in a sentence.
- In this scenario, forcing BERT to use words on either side of the masked word in both directions to predict the masked word.



Journey of BERT

JOHNS HOPKINS UNIVERSITY

: Ayo Ajayi and Yongrui Qi

Last week...

- What problem was RNN trying to solve?

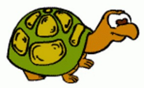
🏺: Ayo Ajayi and Yongrui Qi

Last week…

- What problem was RNN trying to solve?
  - Language Translation

Last week…

- What problem was RNN trying to solve?
  - Language Translation
- What were the issues with Recurrent Neural Networks?



: Ayo Ajayi and Yongrui Qi

Last week...

- What problem was RNN trying to solve?
  - Language Translation
- What were the issues with Recurrent Neural Networks?
  - "Recurrent computation is slow"
  - Long sequences could result in parts of the input being forgotten.

Last week...

- What problem was RNN trying to solve
  - Language Translation
- What were the issues with Recurrent Neural Networks?
  - "Recurrent computation is slow"
  - Long sequences could result in parts of the input being forgotten.

Long Short-Term Memory Networks

LSTM?

- LSTM networks are even slower to train.
  - Input data needs to be passed sequentially (one after the other). Words are passed in sequentially and are generated sequentially.
- We need input from the previous state to make any progress on the current state. **This does not make use of today's GPU designed for parallelization.**

What Inspired BERT?

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser*
Google Brain
lukaszkaiser@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Timeline

LONG SHORT-TERM MEMORY

NEURAL COMPUTATION 9(8):1735–1780, 1997

Sepp Hochreiter
Fakultät für Informatik
Technische Universität München
80290 München, Germany
hochreit@informatik.tu-muenchen.de
http://www7.informatik.tu-muenchen.de/~hochreit

Jürgen Schmidhuber
IDSIA
Corso Elvezia 36
6900 Lugano, Switzerland
juergen@idsia.ch
http://www.idsia.ch/~juergen

01 LSTM

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

02 **Transformer**

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

03 BERT

ALBERT: A LITE BERT FOR SELF-SUPERVISED LEARNING OF LANGUAGE REPRESENTATIONS

Zhenzhong Lan[1] **Mingda Chen**[2*] **Sebastian Goodman**[1] **Kevin Gimpel**[2]
Piyush Sharma[1] **Radu Soricut**[1]
[1]Google Research [2]Toyota Technological Institute at Chicago
{lanzhzh, seabass, piyushsharma, rsoricut}@google.com
{mchen, kgimpel}@ttic.edu

05 Albert

1735-1780, 1997 **2017** **2019** **2020**

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

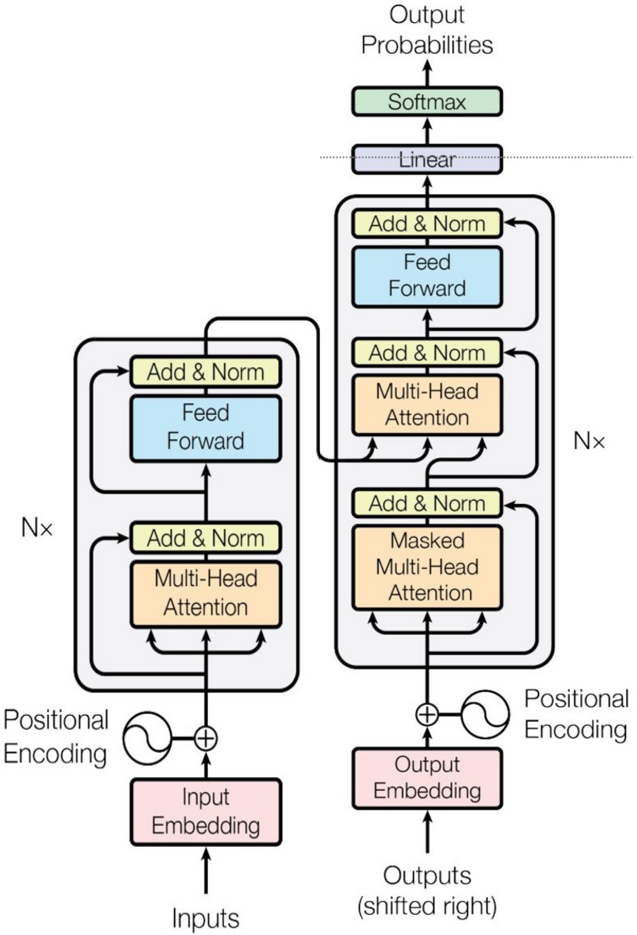
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

: Ayo Ajayi and Yongrui Qi

Transformers



Input sequence can be passed in parallel

Transformer - Input Embeddings and Positional Encoding

Group similar words closer to each other in an embedding space.

Task: Translate English to Spanish

Input: I Love this Class

Goal: Word vectors with positional information

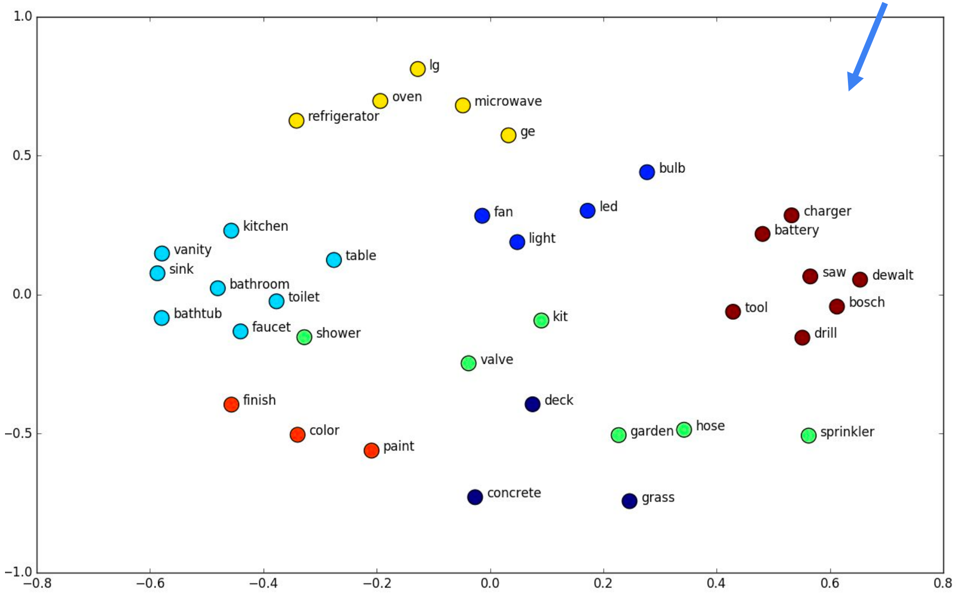


Image Source

Adapted from here

Transformer – Encoder Block

- Multi-Headed Attention Layer
  - What part of the input should I focus on?
  - **Self-attention:** How relevant is the word in the sentence relevant to other words in the same sentence
  - Example: I love this class
    - **I -> I l**ove this class. -> attention vector [0.12 0.54 0.43 0.02]
    - **Love** -> I **love** this class. -> attention vector [0.08 0.61 0.87 0.52]
      - This captures relationships between words in a sentence.
- Feed Forward Layer
  - Feed forward nets that is applied to all attention vectors. It transforms the vectors to a way that can be interpreted by either another encoder or decoder.

: Ayo Ajayi and Yongrui Qi

Goal: What is English? What is context?

Adapted from here

Decoder

- Spanish Input -> Embed + Positional Encoding
- Self-attention: Creates attention vectors for every word in the Spanish sentence.
- The attention vectors here and the vectors from the encoder are passed into another (encoder-decoder) attention block.
  - Determine how related they are to one another which is where the mapping happens.
  - Essentially, the decoder learns How to English words map to Spanish words?

Goal: What is Spanish? What is context?

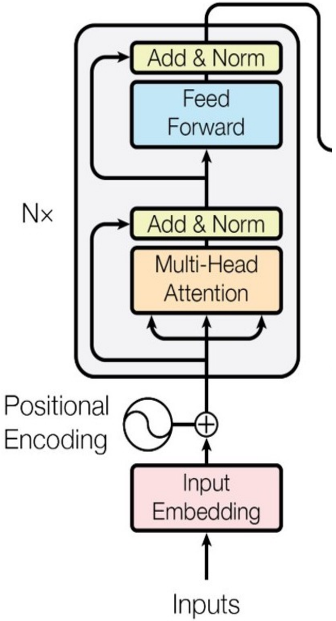
Results

- Models were trained on WMT 2014
  - English-German dataset with 4.5M sentence pairs
  - English-French dataset with 36M sentences
- Models trained on 8 Nvidia P100 GPUs
- BLEU Score:
  - Algorithm for evaluating the quality of text being translated.

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

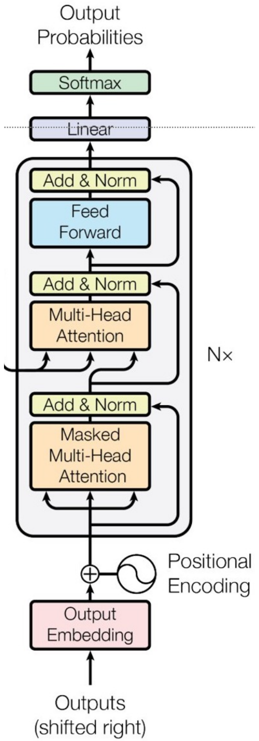
Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [15]	23.75			
Deep-Att + PosUnk [32]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [31]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [8]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [26]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [32]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [31]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [8]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	$\mathbf{3.3 \cdot 10^{18}}$	
Transformer (big)	**28.4**	**41.0**	$2.3 \cdot 10^{19}$	

Encoder



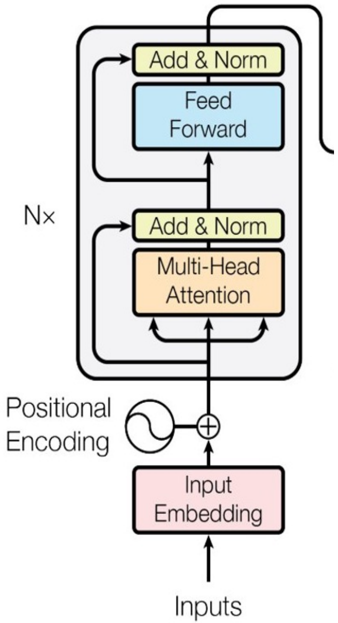
Understands language and context

Decoder

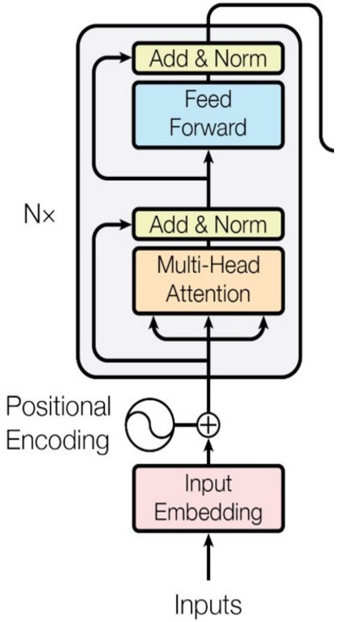


Understands language and context

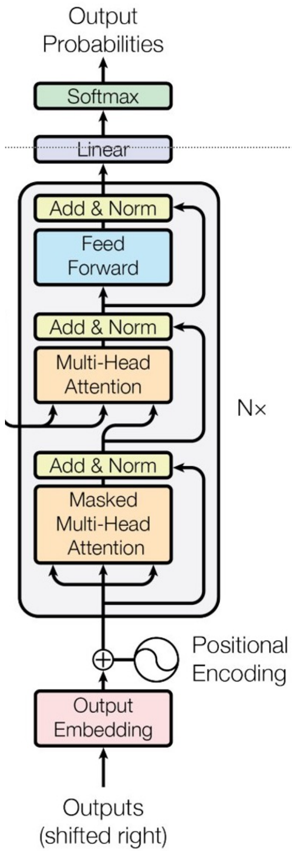
Encoder



+



Decoder



: Ayo Ajayi and Yongrui Qi

BERT - 2019

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

Bidirectional **Encoder** Representation from Transformer

ALBERT: A Lite BERT

- Why ALBERT
- How ALBERT works
- Performance ALBERT v.s. BERT

: Ayo Ajayi and Yongrui Qi

Why ALBERT

- The problems in BERT:
  - Memory limitation
    - Model parallelization ✅
    - Clever management ✅
  - Communication overhead
    - ALBERT incorporates 2 parameter reduction techniques:
      - Factorized embedding parameterization
      - Cross layer parameter sharing
  - Next Sentence Prediction (NSP) ineffectiveness
    - Self-supervised loss for sentence-order prediction (SOP)

How ALBERT works

- Factorized embedding parameterization
  - Recall BERT, XLNet, RoBERTa:
    - WordPiece Embedding Size E = Hidden Layer Size H
  - Question:
    - E: context independent
    - H: context dependent
  - Reduce Embedding Parameters
    - First project one-hot vectors into a lower dimensional embedding size E
    - Then project it into hidden space
    - O(V*H) ➡️ O(V*E+E*H)
    - E: 64, 128(best), 256, 768

How ALBERT works

- Cross-layer parameter sharing
    - Share all parameters across layers
    - Prevent the parameter from growth with the depth of network
    - Weight-sharing has an effect on stabilizing network parameters

: Ayo Ajayi and Yongrui Qi

How ALBERT works

- Inter-sentence coherence loss
  - Why NSP ineffectiveness
    - Lack of difficulty as a task
      - NSP conflates topic prediction and coherence prediction in a single task
      - Topic prediction is much easier
  - ALBERT: sentence order prediction (SOP) loss
    - Avoid topic prediction
    - Focuses on modeling inter-sentence coherence

Performance ALBERT v.s. BERT
Factorized embedding parameterization

Model		Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg	Speedup
BERT	base	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3	4.7x
	large	334M	92.2/85.5	85.0/82.2	86.6	93.0	73.9	85.2	1.0
ALBERT	base	12M	89.3/82.3	80.0/77.1	81.6	90.3	64.0	80.1	5.6x
	large	18M	90.6/83.9	82.3/79.4	83.5	91.7	68.5	82.4	1.7x
	xlarge	60M	92.5/86.1	86.1/83.1	86.4	92.4	74.8	85.5	0.6x
	xxlarge	235M	**94.1/88.3**	**88.1/85.1**	**88.0**	**95.2**	**82.3**	**88.7**	0.3x

Table 2: Dev set results for models pretrained over BOOKCORPUS and Wikipedia for 125k steps. Here and everywhere else, the Avg column is computed by averaging the scores of the downstream tasks to its left (the two numbers of F1 and EM for each SQuAD are first averaged).

Performance ALBERT v.s. BERT
Cross-layer parameter sharing

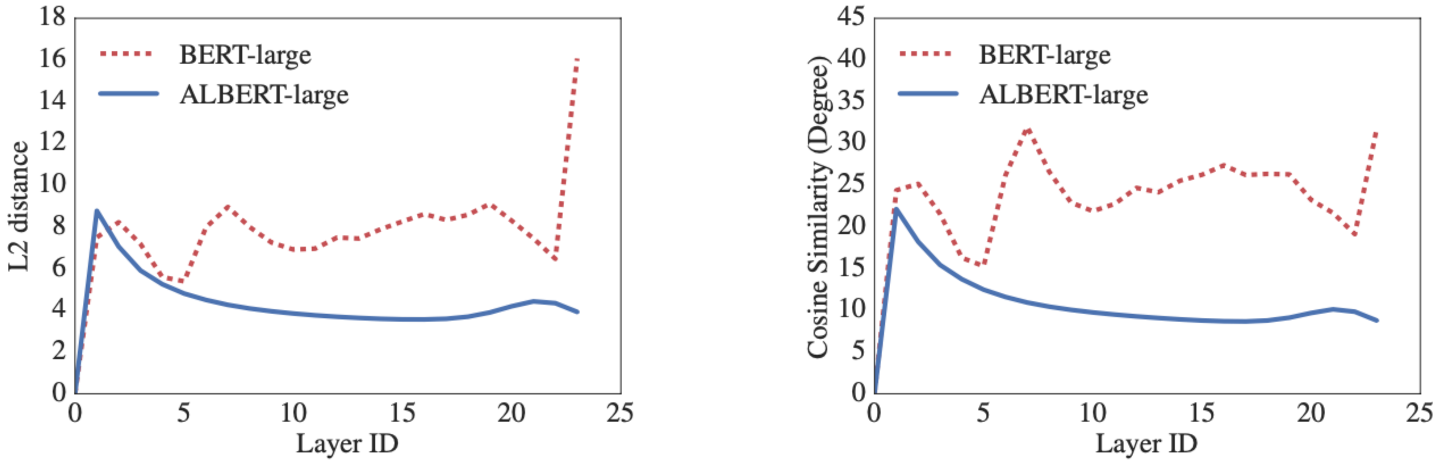


Figure 1: The L2 distances and cosine similarity (in terms of degree) of the input and output embedding of each layer for BERT-large and ALBERT-large.

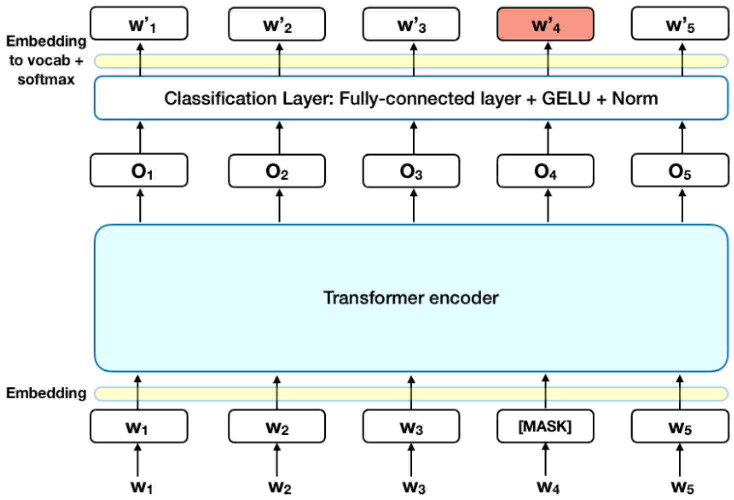
: Ayo Ajayi and Yongrui Qi

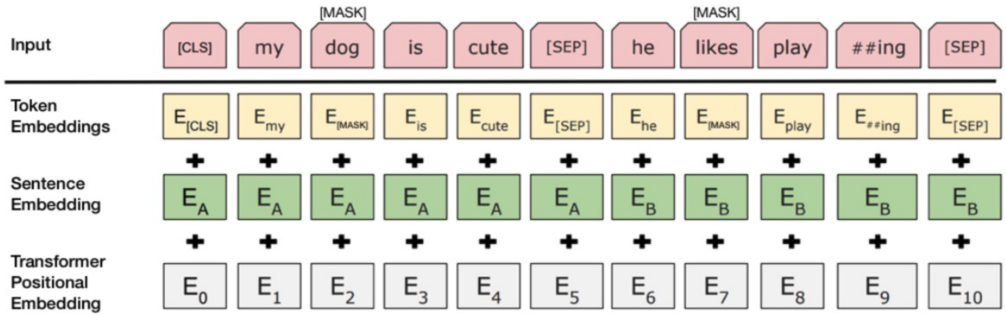
High Level BERT overview

1. BERT uses attention based mechanism to learn contextual relations within NLP

2. An encoder reads the text and works with it bidirectionally

3. Bert is trained using masked tokens for 15% of the words and via next sentence prediction


w'_1 w'_2 w'_3 w'_4 w'_5
Embedding to vocab + softmax
Classification Layer: Fully-connected layer + GELU + Norm
O_1 O_2 O_3 O_4 O_5
Transformer encoder
Embedding
w_1 w_2 w_3 [MASK] w_5
w_1 w_2 w_3 w_4 w_5


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

- 🔭: Muzzi Godil

PROBLEM

- BERT is too big

- Bert uses roughly 110 million parameters for its base form which means it take a long time to train , the larger form of BERT has 340 million parameters.

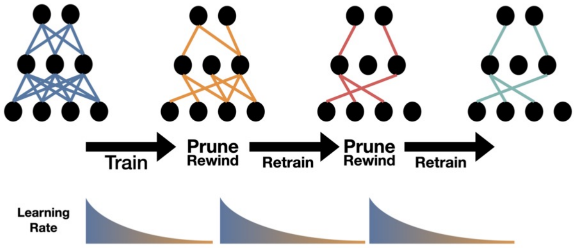
- This means a 12 layer transformer BERT takes 4 days to train !!

How do we make BERT smaller ?

1) Pruning:

This helps us identify the correct subnetwork needed

A MIT ICLR Study says rewinding and training gives great factorization results



- 🔭: Muzzi Godil

How do we make BERT smaller ?

2) Parameter sharing:

- Creates a shared feature space which enables the same feature detector to be used across everything on one plane
- To put into context for AlexNet: 105,415,600 weights vs 34,944 weights

How do we make BERT smaller ?

3) Knowledge Distillation:

- This allows efficient distilling of information from a pretrained BERT to a smaller model that can still work with great accuracy but much faster

- Student-Teacher training where a teacher network adds its error to the student's loss function, thus, helping the student network to converge to a better solution.