COMP5046
Natural Language Processing

Lecture 12: Pretrained Model

Dr. Caren Han

Semester 1, 2022
School of Computer Science,
University of Sydney

0 LECTURE PLAN

THE UNIVERSITY OF SYDNEY

Lecture 12: Pretrained Model

1. The Rise of the Pre-trained Model
2. BERT
3. Post BERT
4. Multimodal Pretrained Model

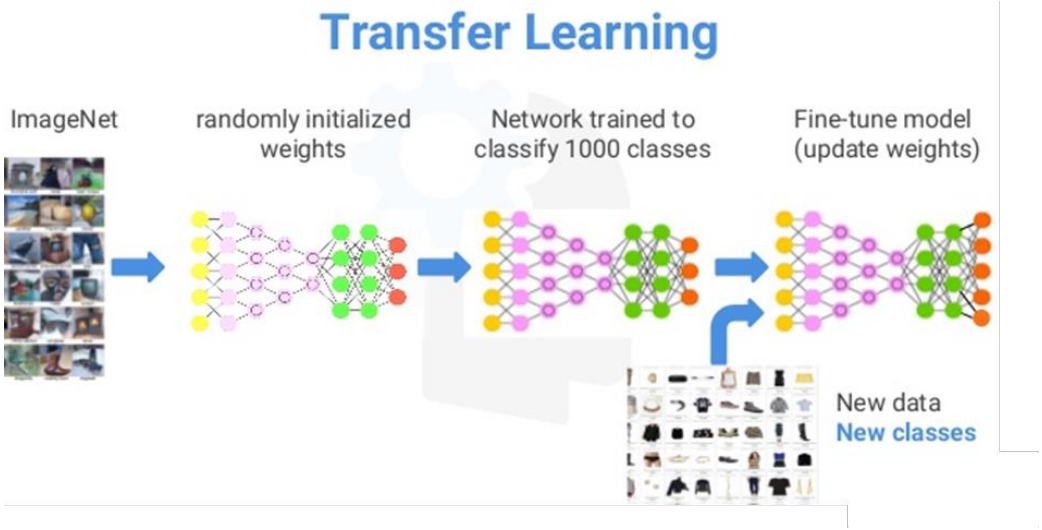
1 The Rise of the Pre-trained Model

Pre-training and Transfer Learning

solve one problem and apply it to related problem

In computer vision, prove the value of transfer learning

- *pre-training a neural network on a known task (i.e. ImageNet)*
- *performing fine-tuning*
- *using the trained neural network as the basis of a new purpose-specific model.*



The Rise of the Pre-trained Model

Pre-training and Transfer Learning in NLP

Popular Pre-trained Model in NLP

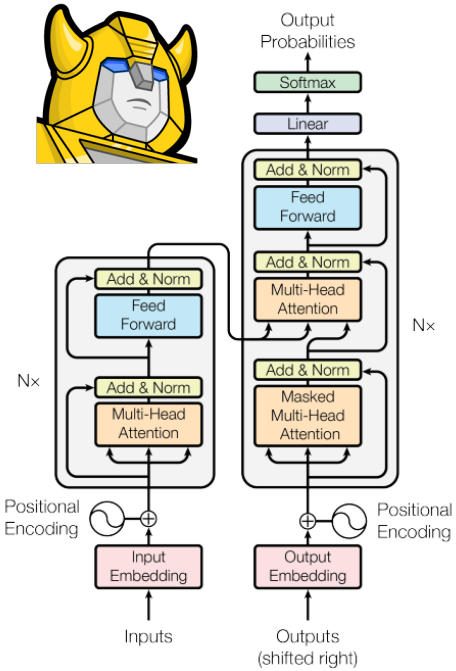
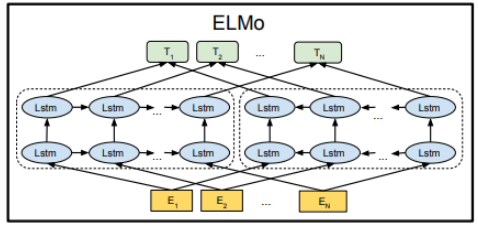
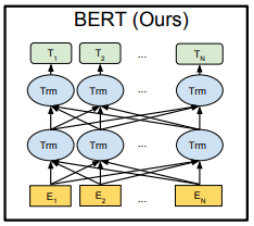


Figure 1: The Transformer - model architecture.



(Peters et al, 2018)



(Devlin et al, 2018)

Using Contextual word representations

The Rise of the Pre-trained Model

Before we started…

Word Structure and subword models

We assume a fixed vocab of tens of thousands of words, built from the training set.

All novel words seen at test time are mapped to a single UNK.

unknown

	Word	Vocab mapping
Common words	computer	desk (index)
	play	cute (index)
Variations	cooooooooooool	UNK (index)
misspellings	laern	UNK (index)
novel items	Transformerify	UNK (index)

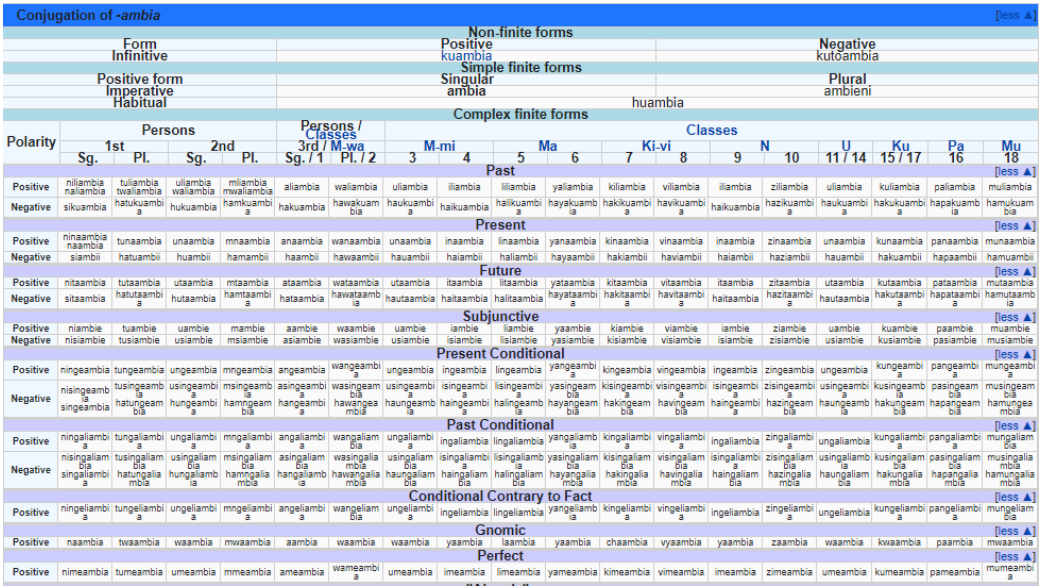
The Rise of the Pre-trained Model

Before we started…

Word Structure and subword models

Many languages exhibit complex morphology, or word structure.

- *The effect is more word types, each occurring fewer times.*



Conjugation of *-ambia* [less ▲]

Non-finite forms																			
Form	Positive											Negative							
Infinitive	kuambia											kutoambia							
Simple finite forms																			
Positive form	Singular											Plural							
Imperative	ambia											ambieni							
Habitual	huambia																		
Complex finite forms																			
Polarity	Persons				Persons / Classes		Classes												
	1st		2nd		3rd / M-wa		M-mi		Ma		Ki-vi		N		U		Ku	Pa	Mu
	Sg.	Pl.	Sg.	Pl.	Sg. / 1	Pl. / 2	3	4	5	6	7	8	9	10	11 / 14	15 / 17	16	18	
Past																			[less ▲]
Positive	niliambia naliambia	tuliambia twaliambia	uliambia waliambia	mliambia mwaliambia	aliambia	waliambia	uliambia	iliambia	liliambia	yaliambia	kiliambia	viliambia	iliambia	ziliambia	uliambia	kuliambia	paliambia	muliambia	
Negative	sikuambia	hatukuambi a	hukuambia	hamkuambi a	hakuambia	hawakuam bia	haukuambi a	haikuambia	halikuambi a	hayakuamb ia	hakikuambi a	havikuambi a	haikuambia	hazikuambi a	haukuambi a	hakukuambi a	hapakuamb ia	hamukuam bia	
Present																			[less ▲]
Positive	ninaambia naambia	tunaambia	unaambia	mnaambia	anaambia	wanaambia	unaambia	inaambia	linaambia	yanaambia	kinaambia	vinaambia	inaambia	zinaambia	unaambia	kunaambia	panaambia	munaambia	
Negative	siambii	hatuambii	huambii	hamambii	haambii	hawaambii	hauambii	haiambii	haliambii	hayaambii	hakiambii	haviambii	haiambii	haziambii	hauambii	hakuambii	hapaambii	hamuambii	
Future																			[less ▲]
Positive	nitaambia	tutaambia	utaambia	mtaambia	ataambia	watataambia	utaambia	itaambia	litaambia	yataambia	kitaambia	vitaambia	itaambia	zitaambia	utaambia	kutaambia	pataambia	mutaambia	
Negative	sitaambia	hatutaambi a	hutaambia	hamtaambi a	hataambia	hawataamb ia	hautaambia	haitaambia	halitaambia	hayataambi a	hakitaambi a	havitaambi a	haitaambia	hazitaambi a	hautaambia	hakutaambi a	hapataambi a	hamutaamb ia	
Subjunctive																			[less ▲]
Positive	niambie	tuambie	uambie	mambie	aambie	waambie	uambie	iambie	liambie	yaambie	kiambie	viambie	iambie	ziambie	uambie	kuambie	paambie	muambie	
Negative	nisiambie	tusiambie	usiambie	msiambie	asiambie	wasiambie	usiambie	isiambie	lisiambie	yasiambie	kisiambie	visiambie	isiambie	zisiambie	usiambie	kusiambie	pasiambie	musiambie	
Present Conditional																			[less ▲]
Positive	ningeambia	tungeambia	ungeambia	mngeambia	angeambia	wangeambi a	ungeambia	ingeambia	lingeambia	yangeambi a	kingeambia	vingeambia	ingeambia	zingeambia	ungeambia	kungeambi a	pangeambi a	mungeambi a	
Negative	nisingeamb ia singeambia	tusingeamb a hatungeam bia	usingeambi a hungeambi a	msingeamb a hamngeam bia	asingeambi a hangeambi a	wasingeam bia hawangea mbia	usingeambi a haungeamb ia	isingeambi a haingeambi a	lisingeambi a halingeamb ia	yasingeam bia hayangeam bia	kisingeambi a hakingeam bia	visingeambi a havingeam bia	isingeambi a haingeambi a	zisingeambi a hazingeam bia	usingeambi a haungeamb ia	kusingeamb a hakungeam bia	pasingeam bia hapangeam bia	musingeam bia hamungea mbia	
Past Conditional																			[less ▲]
Positive	ningaliambi a	tungaliambi a	ungaliambi a	mngaliambi a	angaliambi a	wangaliam bia	ungaliambi a	ingaliambia	lingaliambia	yangaliambi a	kingaliambi a	vingaliambi a	ingaliambia	zingaliambi a	ungaliambia	kungaliambi a	pangaliambi a	mungaliam bia	
Negative	nisingaliam bia singaliambi a	tusingaliam bia hatungalia mbia	usingaliam bia hungaliamb ia	msingaliam bia hamngalia mbia	asingaliam bia hangaliamb ia	wasingalia mbia hawangalia mbia	usingaliam bia haungaliam bia	isingaliambi a haingaliam bia	lisingaliamb ia halingaliam bia	yasingaliam bia hayangalia mbia	kisingaliam bia hakingalia mbia	visingaliam bia havingalia mbia	isingaliambi a haingaliam bia	zisingaliam bia hazingalia mbia	usingaliamb ia haungaliam bia	kusingaliam bia hakungalia mbia	pasingaliam bia hapangalia mbia	musingalia mbia hamungalia mbia	
Conditional Contrary to Fact																			[less ▲]
Positive	ningeliambi a	tungeliambi a	ungeliambi a	mngeliambi a	angeliambi a	wangeliam bia	ungeliambi a	ingeliambia	lingeliambia	yangeliamb ia	kingeliambi a	vingeliambi a	ingeliambia	zingeliambi a	ungeliambia	kungeliambi a	pangeliambi a	mungeliam bia	
Gnomic																			[less ▲]
Positive	naambia	twaambia	waambia	mwaambia	aambia	waambia	waambia	yaambia	laambia	yaambia	chaambia	vyaambia	yaambia	zaambia	waambia	kwaambia	paambia	mwaambia	
Perfect																			[less ▲]
Positive	nimeambia	tumeambia	umeambia	mmeambia	ameambia	wameambi a	umeambia	imeambia	limeambia	yameambia	kimeambia	vimeambia	imeambia	zimeambia	umeambia	kumeambia	pameambia	mumeambi a	

1 The Rise of the Pre-trained Model

THE UNIVERSITY OF SYDNEY

Before we started…

The byte-pair encoding algorithm

Subword modeling in NLP encompasses a wide range of methods for reasoning about structure below the word level. (Parts of words, characters, bytes.)

Byte-pair encoding is a simple, effective strategy for defining a subword vocabulary.

1. *Start with a vocabulary containing only characters and an "end-of-word" symbol.*
2. *Using a corpus of text, find the most common adjacent characters "a,b"; add "ab" as a subword.*
3. *Replace instances of the character pair with the new subword; repeat until desired vocab size.*

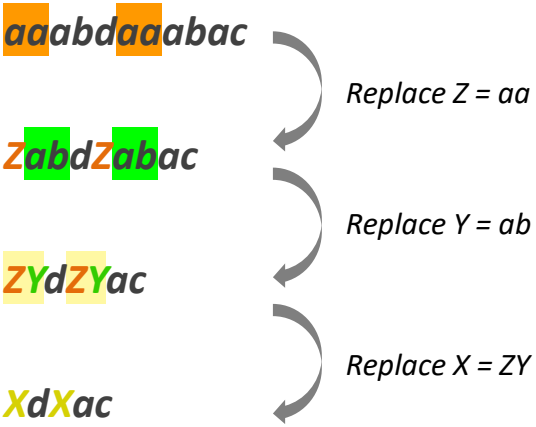
The Rise of the Pre-trained Model

The byte-pair encoding algorithm (1994)

The byte-pair encoding algorithm: How it works?

A simple form of data compression in which the most common pair of consecutive bytes of data is replaced with a byte that does not occur within that data

aaabdaaabac



Byte Pair	Replacement
ZY	X
ab	Y
aa	Z

1 The Rise of the Pre-trained Model

THE UNIVERSITY OF SYDNEY

The byte-pair encoding algorithm in NLP

Traditional Word Encoding in NLP

Dictionary

#vocab: occurance
low: 5,
lower: 2,
newest: 6,
widest: 3

Vocabulary

low, lower, newest, widest

What if we have the word 'lowest' in the test set?

OOV Issue

1 The Rise of the Pre-trained Model

The byte-pair encoding algorithm in NLP

Subword Segmentation

Character/unicode → Vocabulary (Bottom up style)

→ The most common pair of consecutive bytes of data is replaced with a byte

make as character

Dictionary

#vocab: occurance
l o w: 5,
l o w e r: 2,
n e w e s t: 6,
w i d e s t: 3

Character-based segmentation

Vocabulary

l, o, w, e, r, n, w, s, t, i, d

The Rise of the Pre-trained Model

The byte-pair encoding algorithm in NLP

Subword Segmentation

Character/unicode → Vocabulary (Bottom up style)

→ The most common pair of consecutive bytes of data is replaced with a byte

Dictionary

#vocab: occurance
l o w: 5,
l o w e r: 2,
n e w e s t: 6,
w i d e s t: 3

Character-based segmentation

Vocabulary

l, o, w, e, r, n, w, s, t, i, d

1 The Rise of the Pre-trained Model

The byte-pair encoding algorithm in NLP

Subword Segmentation – 1st round

Character/unicode → Vocabulary (Bottom up style)

→ The most common pair of consecutive bytes of data is replaced with a byte

Dictionary

#vocab: occurance
l o w: 5,
l o w e r: 2,
n e w e s t: 6,
w i d e s t: 3

Character-based segmentation

Vocabulary

l, o, w, e, r, n, w, s, t, i, d

es 9 times

Replace es = e s

Dictionary

#vocab: occurance
l o w: 5,
l o w e r: 2,
n e w es t: 6,
w i d es t: 3

Vocabulary

l, o, w, e, r, n, w, s, t, i, d, es

The Rise of the Pre-trained Model

The byte-pair encoding algorithm in NLP

Subword Segmentation – 2nd round

Character/unicode → Vocabulary (Bottom up style)

→ The most common pair of consecutive bytes of data is replaced with a byte

Dictionary

#vocab: occurance
l o w: 5,
l o w e r: 2,
n e w es t: 6,
w i d es t: 3

Character-based segmentation

Vocabulary

l, o, w, e, r, n, w, s, t, i, d, es

Replace est = es t

Dictionary

#vocab: occurance
l o w: 5,
l o w e r: 2,
n e w est: 6,
w i d est: 3

Vocabulary

l, o, w, e, r, n, w, s, t, i, d, es, est

The Rise of the Pre-trained Model

The byte-pair encoding algorithm in NLP

Subword Segmentation – 3rd round

Character/unicode → Vocabulary (Bottom up style)

→ The most common pair of consecutive bytes of data is replaced with a byte

Dictionary

#vocab: occurance
l o w: 5,
l o w e r: 2,
n e w est: 6,
w i d est: 3

Character-based segmentation

Vocabulary

l, o, w, e, r, n, w, s, t, i, d, es, est

Replace lo = l o

Dictionary

#vocab: occurance
lo w: 5,
lo w e r: 2,
n e w est: 6,
w i d est: 3

Vocabulary

l, o, w, e, r, n, w, s, t, i, d, es, est, lo

The Rise of the Pre-trained Model

The byte-pair encoding algorithm in NLP

Subword Segmentation – 3rd round

Character/unicode → Vocabulary (Bottom up style)

→ The most common pair of consecutive bytes of data is replaced with a byte

Dictionary

#vocab: occurance
l o w: 5,
l o w e r: 2,
n e w est: 6,
w i d est: 3

Character-based segmentation

Vocabulary

l, o, w, e, r, n, w, s, t, i, d, es, est

Replace lo = l o

Dictionary

#vocab: occurance
lo w: 5,
lo w e r: 2,
n e w est: 6,
w i d est: 3

Vocabulary

l, o, w, e, r, n, w, s, t, i, d, es, est, lo

Repeat this process until 10th round

1 The Rise of the Pre-trained Model

The byte-pair encoding algorithm in NLP

Subword Segmentation – Final (after 10th round)

Character/unicode → Vocabulary (Bottom up style)

→ The most common pair of consecutive bytes of data is replaced with a byte

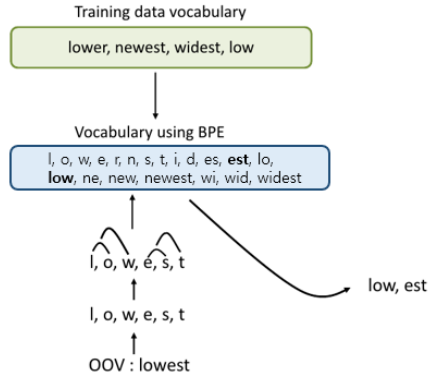
Dictionary

#vocab: occurance
low: 5,
low e r: 2,
newest: 6,
widest: 3

Vocabulary

l, o, w, e, r, n, w, s, t, i, d, es, est, lo, low, ne, new, newest, wi, wid, widest

What if we have the word 'lowest' in the test set?



Repeat this process until 10th round

1 The Rise of the Pre-trained Model

THE UNIVERSITY OF SYDNEY

Before we started…

Word Structure and subword models

Common words end up being a part of the subword vocabulary, while rarer words are split into (sometimes intuitive, sometimes not) components.

In the worst case, words are split into as many subwords as they have characters

	Word	Vocab mapping
Common words	computer	Computer
	play	play
Variations	cooooooool	coo##ooo#ool
misspellings	laern	la##ern##
novel items	Transformerify	Transformer##ify

1 The Rise of the Pre-trained Model

THE UNIVERSITY OF SYDNEY

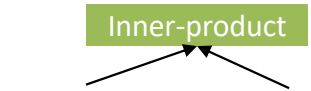
Pre-training and Transfer Learning in NLP

Popular Pre-trained Model: Word Embeddings

Word embeddings (e.g. word2vec) are the basis of deep learning for NLP

king *[-0.5, -0.9, 1.4, …]* ***queen*** *[0.7, 0.2, -0.5, 1.1, …]*

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



The king wore a crown

The queen wore a crown

1 The Rise of the Pre-trained Model

THE UNIVERSITY OF SYDNEY

Pre-training and Transfer Learning in NLP

Popular Pre-trained Model: Contextual Representations

Word embeddings (i.e. word2vec, fastText, GloVe) are applied in a context free manner

Step up to the ***bat*** — ***bat*** *[0.7, 0.2, -0.5, 1.1, …]*

A vampire ***bat*** —— ***bat*** *[0.7, 0.2, -0.5, 1.1, …]*

same embedding

Need to train ***contextual representation*** *on text corpus*

word same but context different ⇓ embedding change

Step up to the ***bat*** — ***bat*** *[1.1, -0.7, 0.8, 2.1, …]*

A vampire ***bat*** —— ***bat*** *[0.3, 0.5, -0.9, 1.3, …]*

based on domain / context

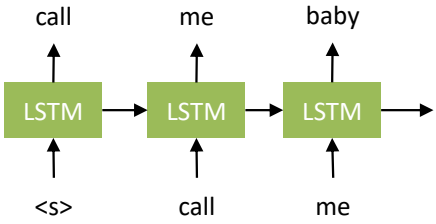
1 The Rise of the Pre-trained Model

Pre-training and Transfer Learning in NLP

Early-Stage Pretraining in NLP

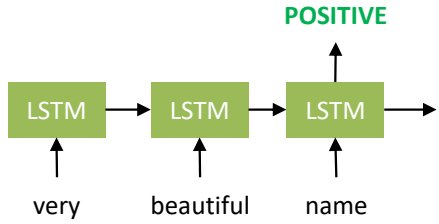
Semi-supervised Sequence Learning (Dai and Le, 2015)

Train LSTM Language Model



learn the pattern

Fine-tune on Classification Task
(e.g. sentiment analysis)



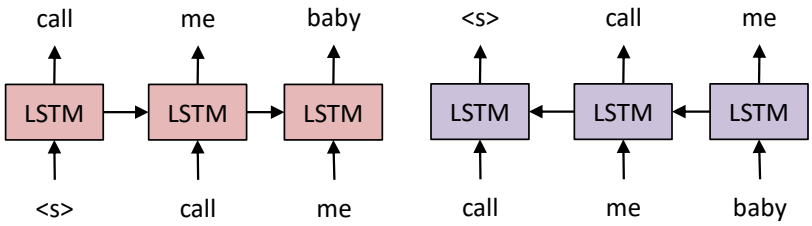
BiLSTM + new input

1 The Rise of the Pre-trained Model

Pre-training and Transfer Learning in NLP

ELMO: Deep Contextual Word Embeddings *(Peters et al., 2018)*

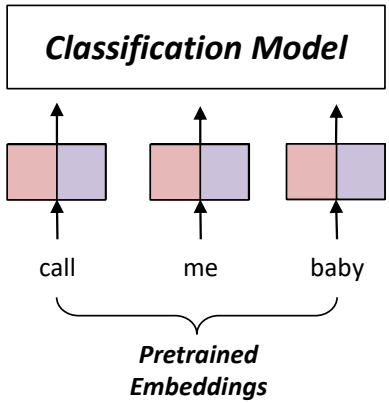
Train Separate Left-to-Right and Right-to-Left Language Models



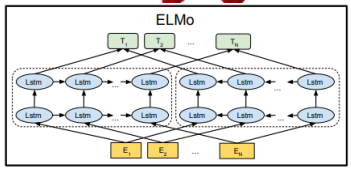
train both directions

difference between ELMO & biLSTM?

Apply as "Pretrained Embeddings"



1 The Rise of the Pre-trained Model



Pre-training and Transfer Learning in NLP

ELMo: Deep Contextual Word Embeddings (2017)

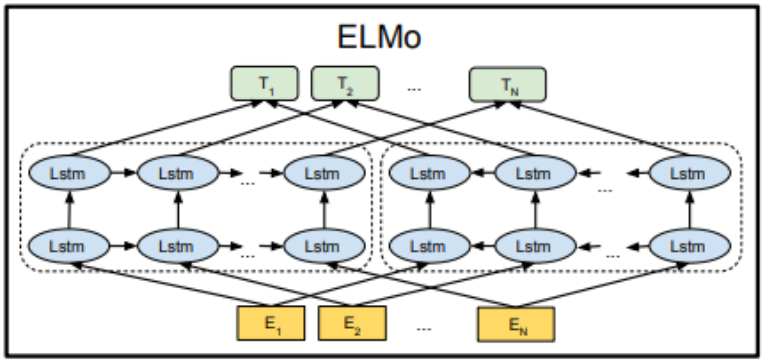
*ELMo provided a **significant step towards pre-training in the context of NLP**. Let's dig in what the ELMo's big secret is!*

1 The Rise of the Pre-trained Model

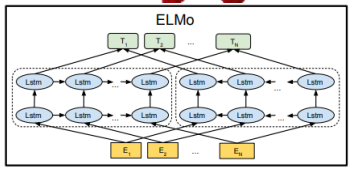
Pre-training and Transfer Learning in NLP

ELMo: Deep Contextual Word Embeddings (2017)

ELMo gained its language understanding from being trained to predict the next word in a sequence of words, Language Modeling Tasks. This is convenient because we have vast amounts of text data that such a model can learn from without needing labels.



1 The Rise of the Pre-trained Model

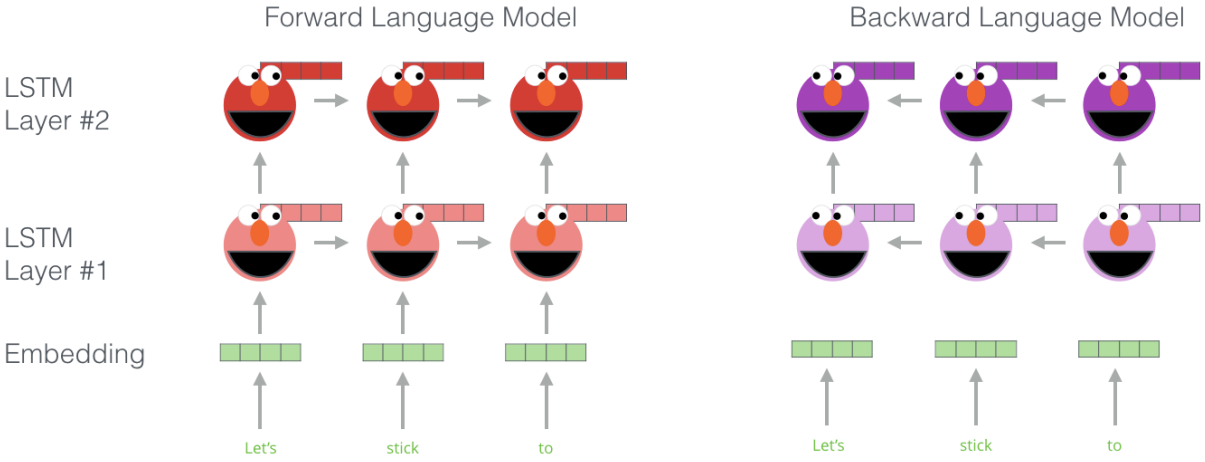


Pre-training and Transfer Learning in NLP

ELMo: Deep Contextual Word Embeddings (2017)

We can see the hidden state of each unrolled-LSTM step peaking out from behind ELMo's head. Those come in handy in the embedding process after this pre-training is done.

ELMo goes a step further and trains a bi-directional LSTM – so that its language model doesn't only have a sense of the next word, but also the previous word.



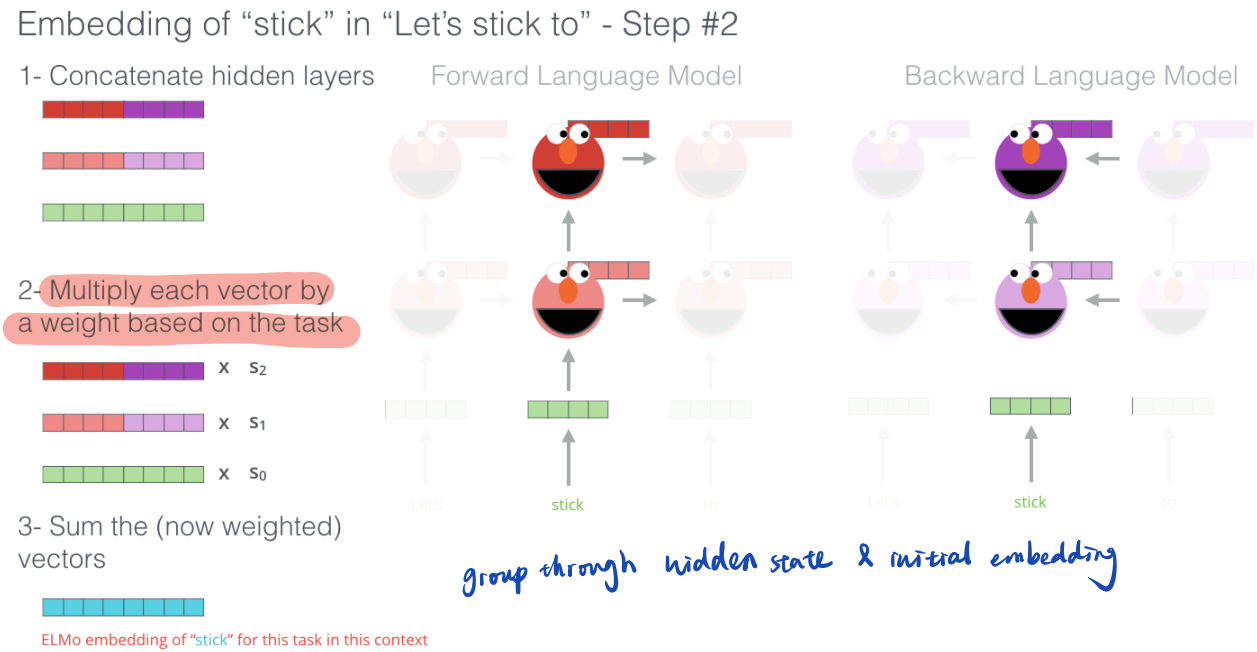
The Rise of the Pre-trained Model

Pre-training and Transfer Learning in NLP

weak bidirectional

ELMo: Deep Contextual Word Embeddings (2017)

ELMo comes with the contextualized embedding through grouping together the hidden states (and initial embedding) in a certain way (concatenation followed by weighted summation).



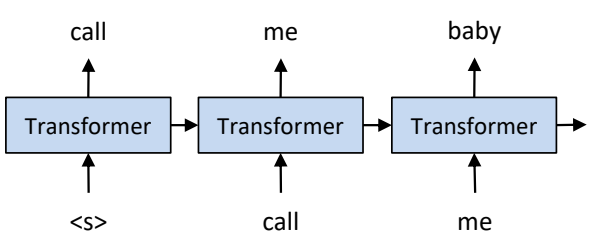
1 The Rise of the Pre-trained Model

Pre-training and Transfer Learning in NLP

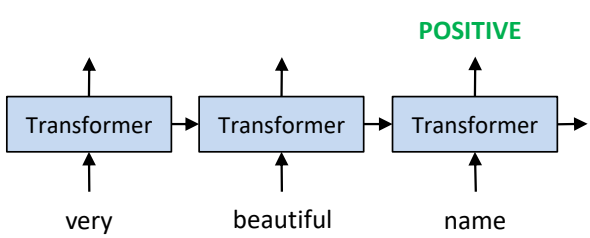
Early-Stage Pretraining in NLP

*Improving Language Understanding by **G**enerative **P**re-**T**raining (2018)*

Train Deep (12 layer) Transformer Language Model



Fine-tune on Classification Task
(e.g. sentiment analysis)



The Rise of the Pre-trained Model

Transformer (Recap)

1. Encoder

A stack of N=6 identical layers.

Each layer with two sub-layers:

1.Multi-head self-attention mechanism

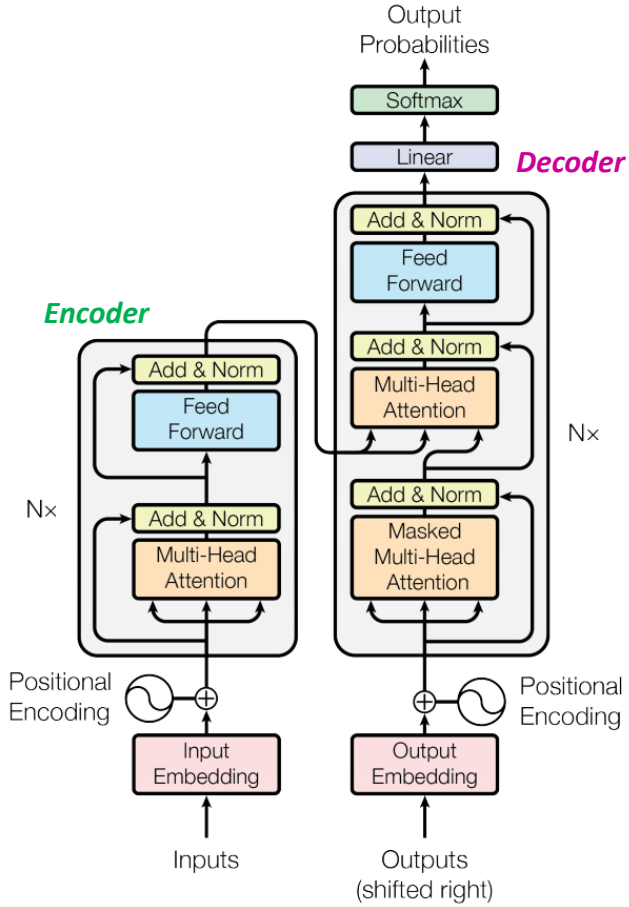
2.Position-wise fully connected feed-forward network

2. Decoder

A stack of N=6 identical layers.

Each layer with three sub-layers:

1. *Multi-head self-attention mechanism*
2. *Position-wise fully connected feed-forward network*
3. *Masked Multi-head self-attention*



The transformer – model architecture

The Rise of the Pre-trained Model

Transformer (Recap)

Multi-head Attention

- *Models context*

Feed-forward layers

- *Computes non-linear hierarchical features*

Layer norm and residuals

- *Makes training deep networks healthy*

Positional Embeddings

- *Allows model to learn relative positioning*

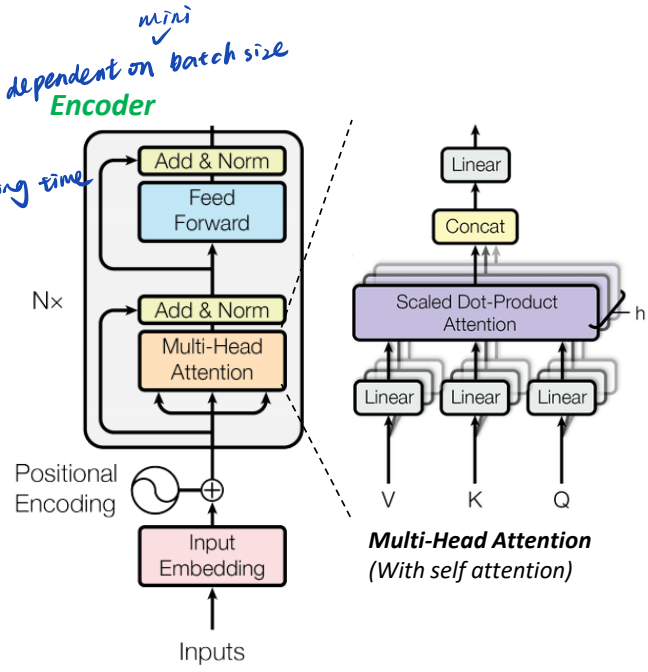
use input & output why ?

batch norm dependent on mini batch size

layer norm reduce training time

stabilizing

transformer doesn't know about sequence order



Multi-Head Attention
(With self attention)

The transformer – Encoder

1 The Rise of the Pre-trained Model

Transformer (Recap)

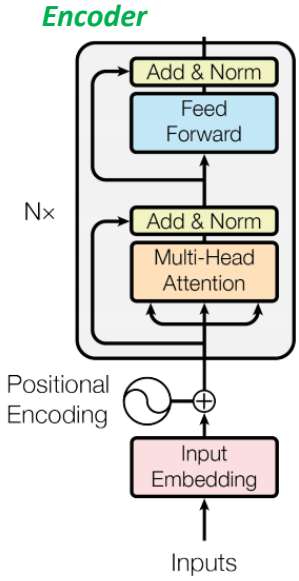
Multi-head Attention
- *Models context*

Feed-forward layers
- *Computes non-linear hierarchical features*

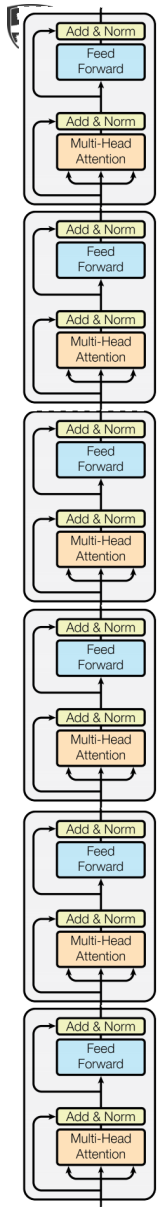
Layer norm and residuals
- *Makes training deep networks healthy*

Positional Embeddings
- *Allows model to learn relative positioning*

Blocks are repeated 6 or more times (in vertical stack)



The transformer – Encoder

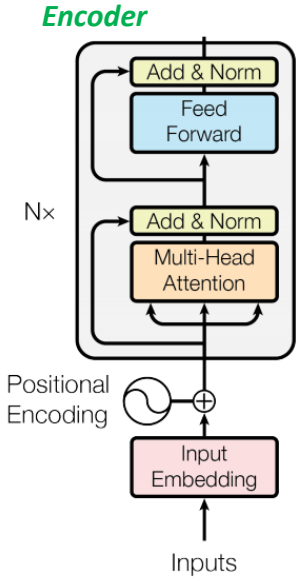


1 The Rise of the Pre-trained Model

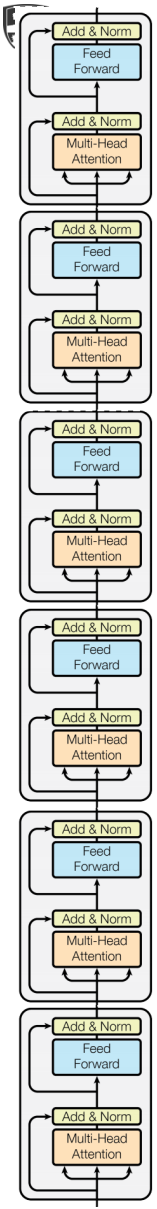
Transformer VS LSTM

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.***



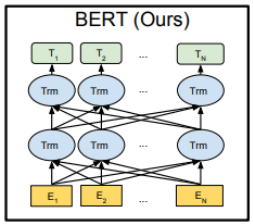
The transformer – Encoder



2 BERT

Google

BERT



BERT

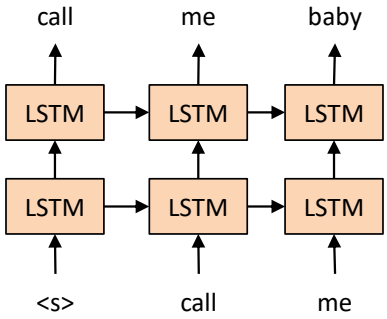
Problem with Previous Approaches

Problem: *Language models only use left context or right context, but language understanding is bidirectional*

ELMO
left style
right style
↓
concat layer
weak bidirectional

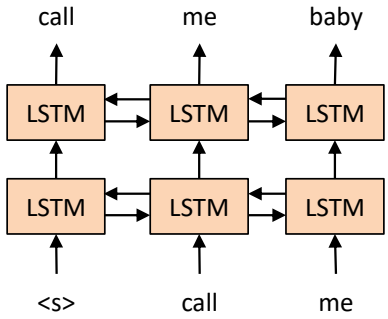
Unidirectional context

Build representation incrementally



Bidirectional context

Word can "see themselves"



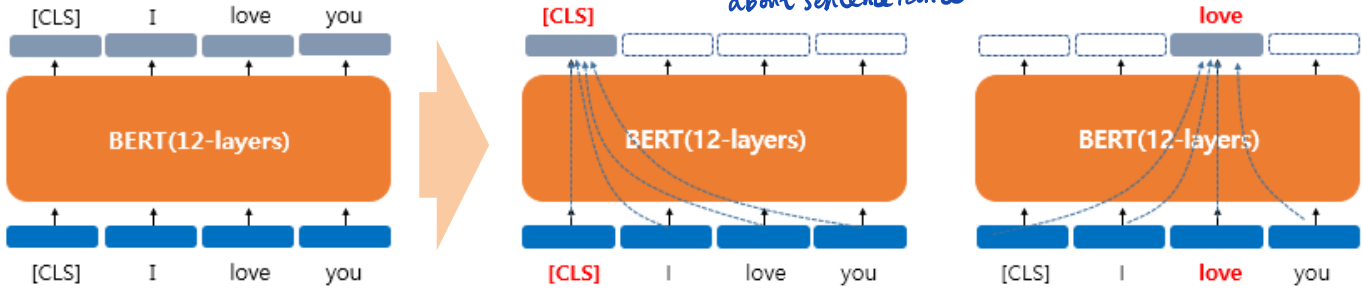
2 BERT

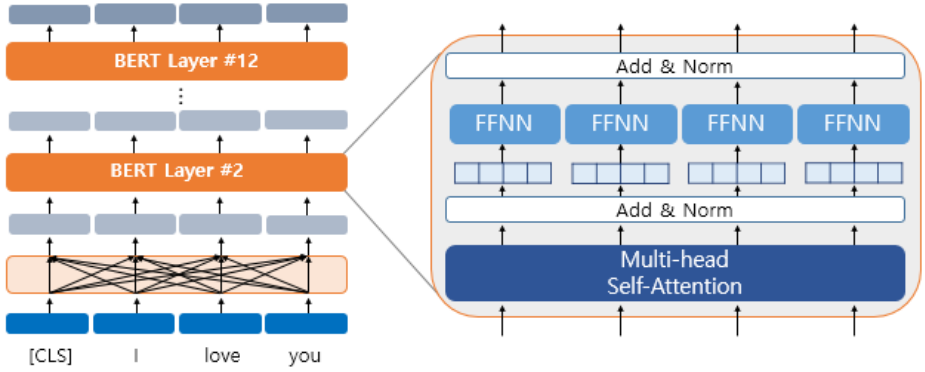
How the BERT is working (Brief)

for sentence-based classification

understand by going through all context

context vector: understand about sentence/context





Pre-training and Transfer Learning in NLP

BERT: Input Representation

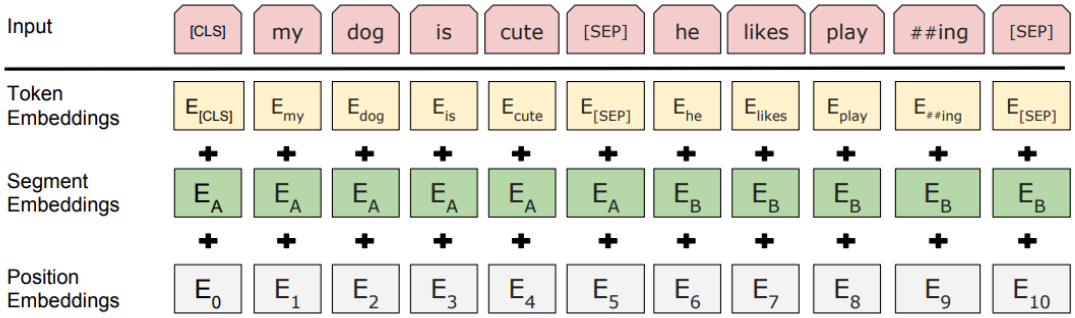


Figure 2: BERT input representation. The input embeddings is the sum of the token embeddings, the segmentation embeddings and the position embeddings.

THE UNIVERSITY OF SYDNEY

Word-piece Token Embedding

Word Structure and subword models

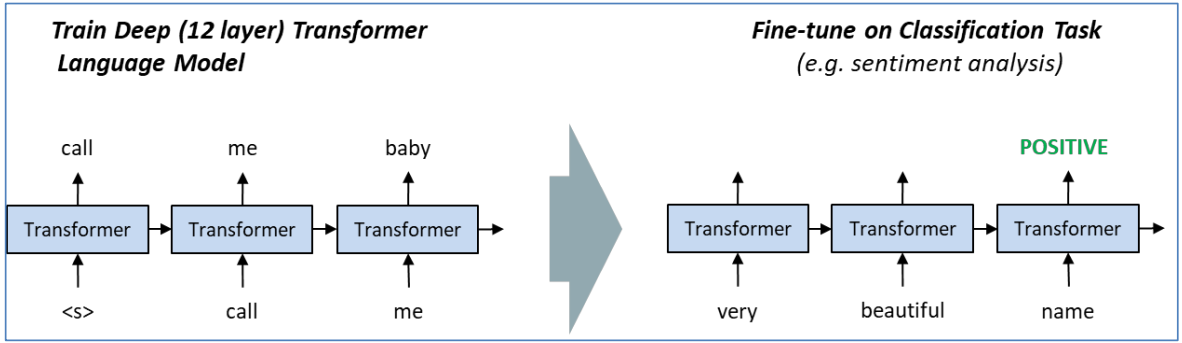
Common words end up being a part of the subword vocabulary, while rarer words are split into (sometimes intuitive, sometimes not) components.

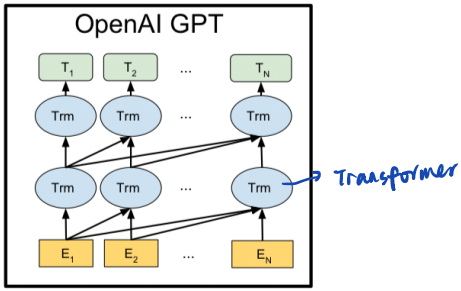
In the worst case, words are split into as many subwords as they have characters

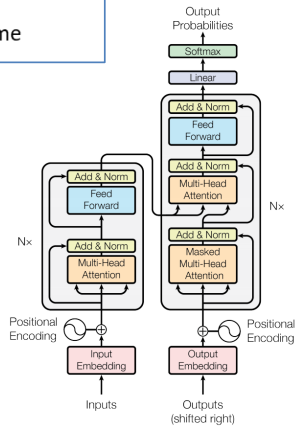
	Word	Vocab mapping
Common words	computer	Computer
	play	play
Variations	cooooooool	coo##ooo#ool
misspellings	laern	la##ern##
novel items	Transformerify	Transformer##ify

Pretraining for BERT

Remember GPT? Pretraining the Language Model







2 BERT

THE UNIVERSITY OF SYDNEY

Pretraining: Masked Language Model (LM)

Mask out k% of the input words, and then predict the masked words (use k=15%)

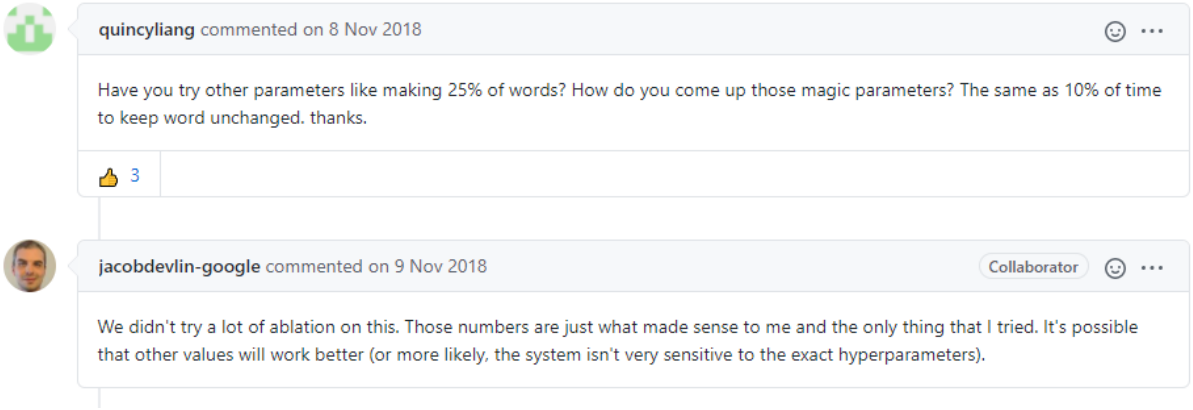
Too little masking: too expensive to train

Too much masking: not enough context

1 mask → train model for just 1 prediction

store gallon

The man went to the [MASK] to buy a [MASK] of milk



quincyliang commented on 8 Nov 2018

Have you try other parameters like making 25% of words? How do you come up those magic parameters? The same as 10% of time to keep word unchanged. thanks.

3

jacobdevlin-google commented on 9 Nov 2018 Collaborator

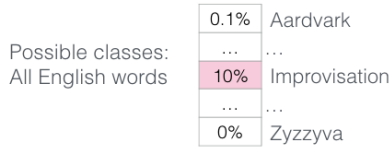
We didn't try a lot of ablation on this. Those numbers are just what made sense to me and the only thing that I tried. It's possible that other values will work better (or more likely, the system isn't very sensitive to the exact hyperparameters).

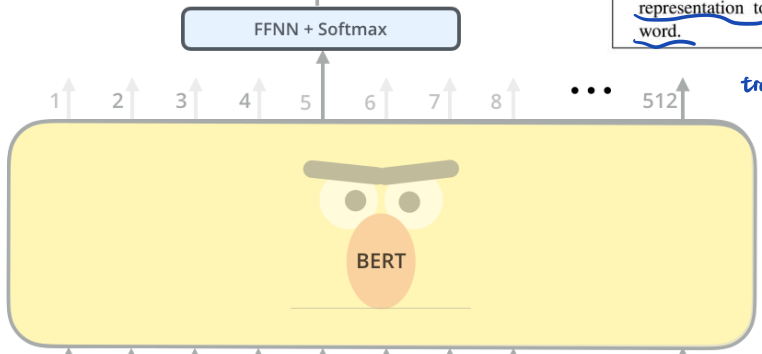
2 BERT

Pre-training and Transfer Learning in NLP

BERT: Masked language Model

Use the output of the masked word's position to predict the masked word





predict the masked

Randomly mask 15% of tokens

Input

With those 15%

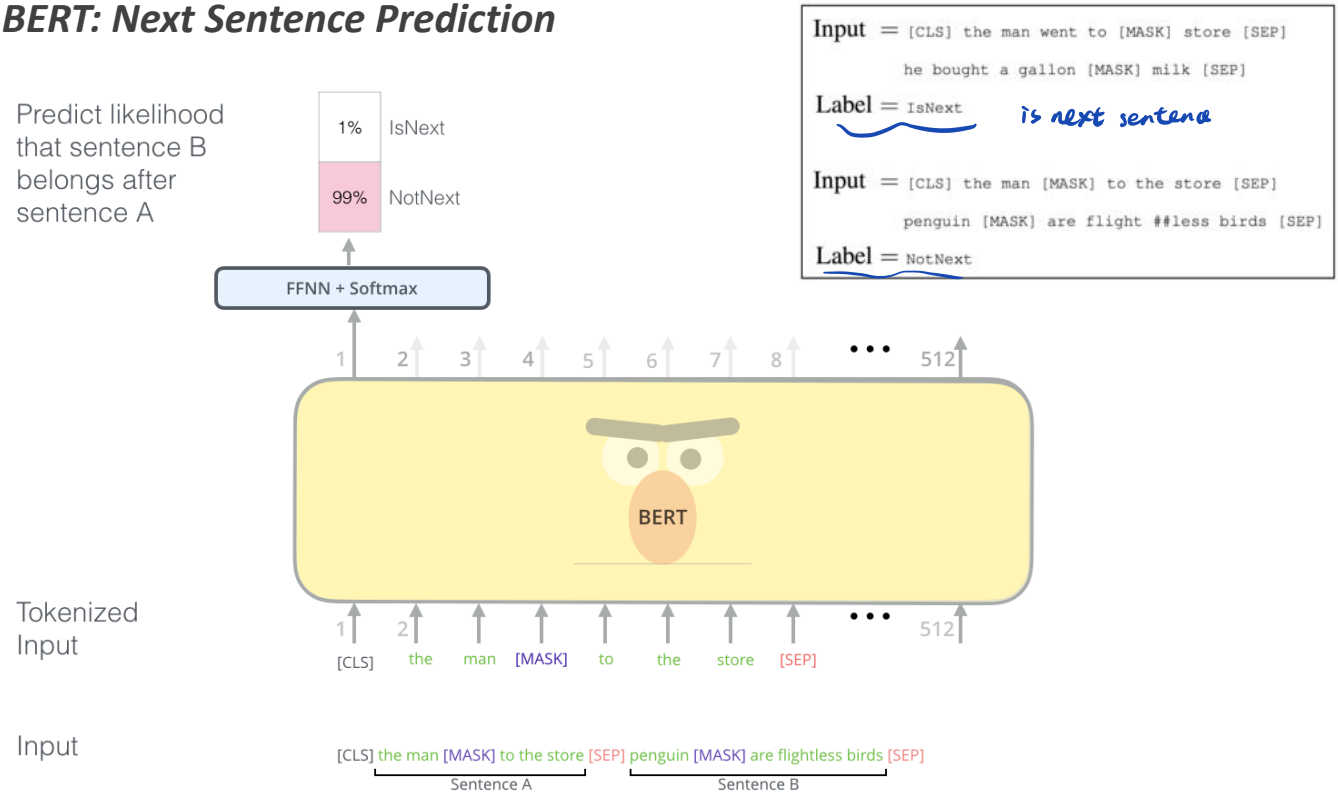
- Rather than *always* replacing the chosen words with `[MASK]`, the data generator will do the following:
- 80% of the time: Replace the word with the `[MASK]` token, e.g., `my dog is hairy → my dog is [MASK]`
- 10% of the time: Replace the word with a random word, e.g., `my dog is hairy → my dog is 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.

transformer doesn't know which word has to be predicted or which is the random one.

2 BERT

Pre-training and Transfer Learning in NLP

BERT: Next Sentence Prediction



Pre-training and Transfer Learning in NLP

BERT: Input Representation

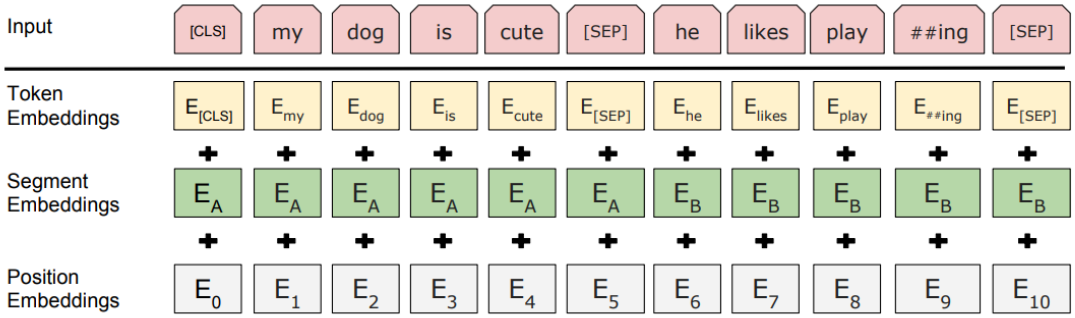


Figure 2: BERT input representation. The input embeddings is the sum of the token embeddings, the segmentation embeddings and the position embeddings.

BERT

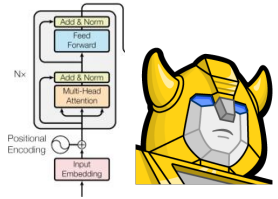
Model Details

- ***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*

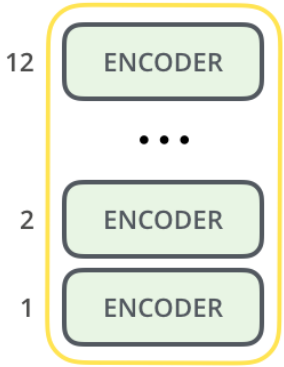
- *Trained on 4x4 or 8x8 TPU slice for 4 days*

2 BERT

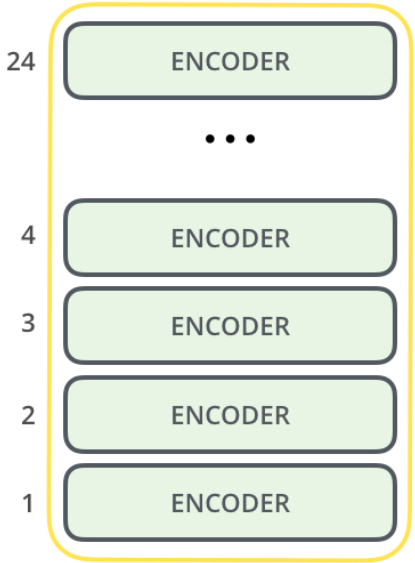


Pre-training and Transfer Learning in NLP

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



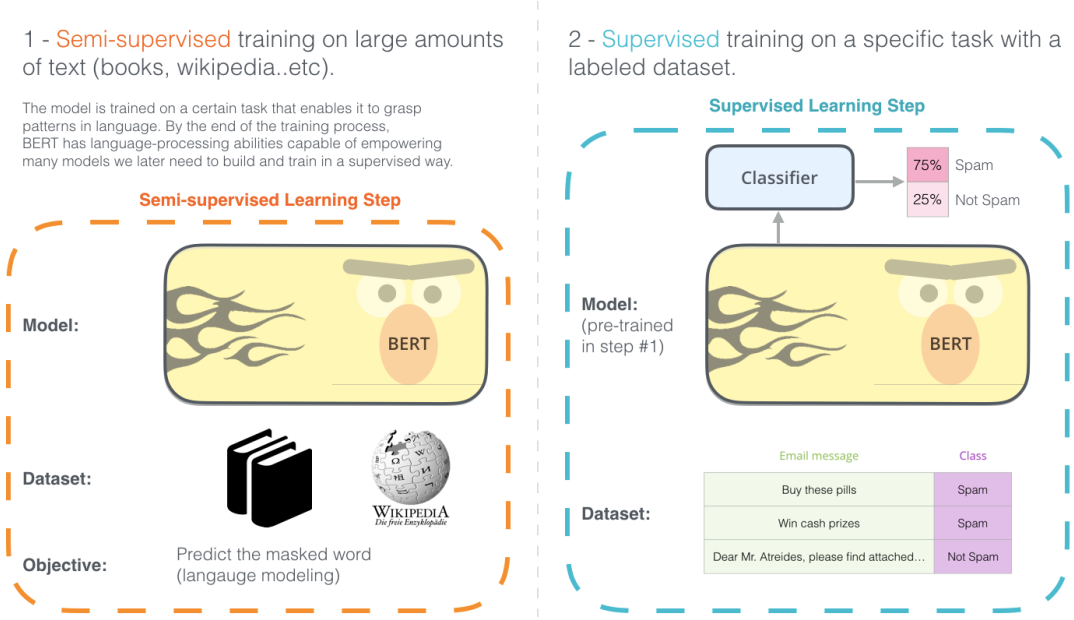
$BERT_{BASE}$



$BERT_{LARGE}$

Pre-training and Transfer Learning in NLP

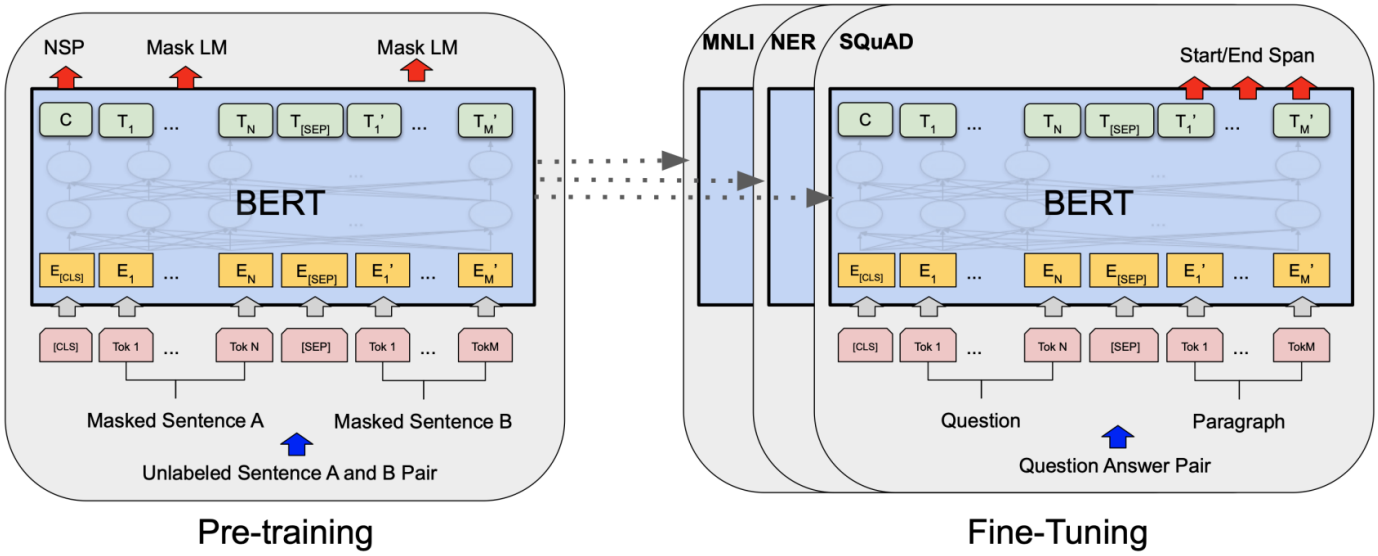
The two steps of how BERT is developed. Download the model pre-trained in step 1 (trained on un-annotated data), and only worry about fine-tuning it for step 2



Pretraining ***Fine-Tuning***

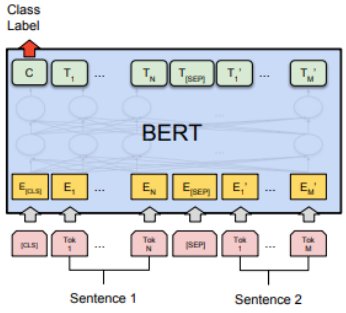
BERT

Fine-Tuning Procedure

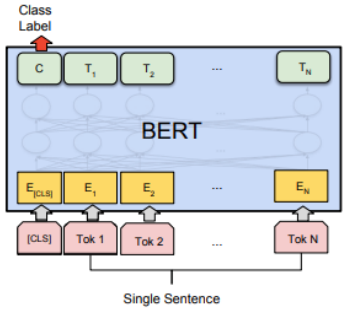


Fine-Tuning Procedure

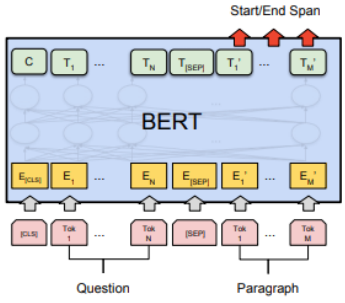
The following shows a number of ways to use BERT for different tasks.



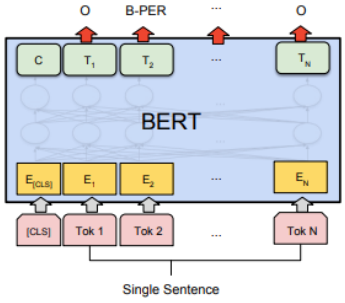
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



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



(c) Question Answering Tasks: SQuAD v1.1



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

BERT

Accuracy… Performance

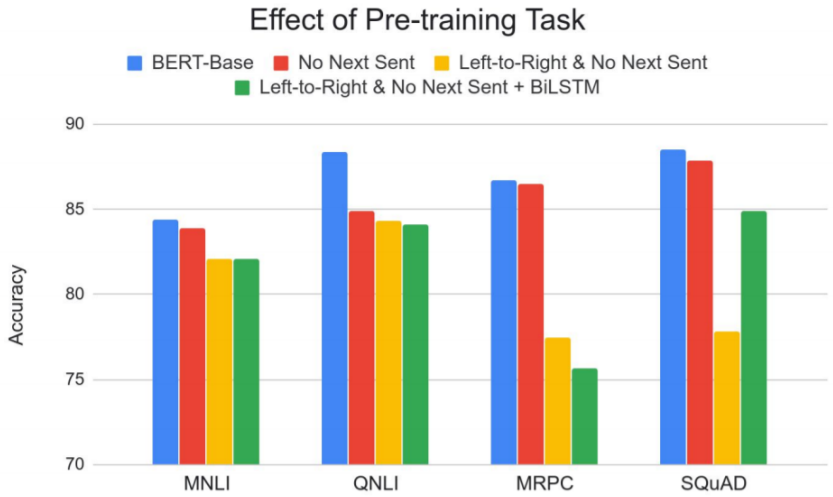
System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
$\text{BERT}_{\text{BASE}}$	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/.

https://rajpurkar.github.io/SQuAD-explorer/

Effect of Pre-training Task

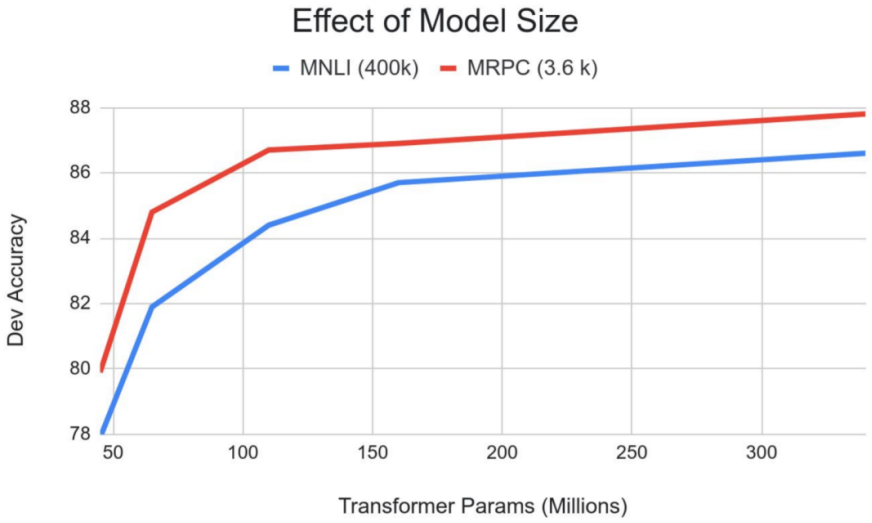
- *Masked LM (compared to left-to-right LM) is very important on some tasks, Next Sentence Prediction is important on other tasks.*
- *Left-to-right model does very poorly on word-level task (SQuAD), although this is mitigated by Bi-LSTM*



BERT

Effect of Model Size

- *Big models help a lot*
- *Going from 110M → 340M params helps even on datasets with 3,600 labelled examples*
- *Improvements have not asymptoted*

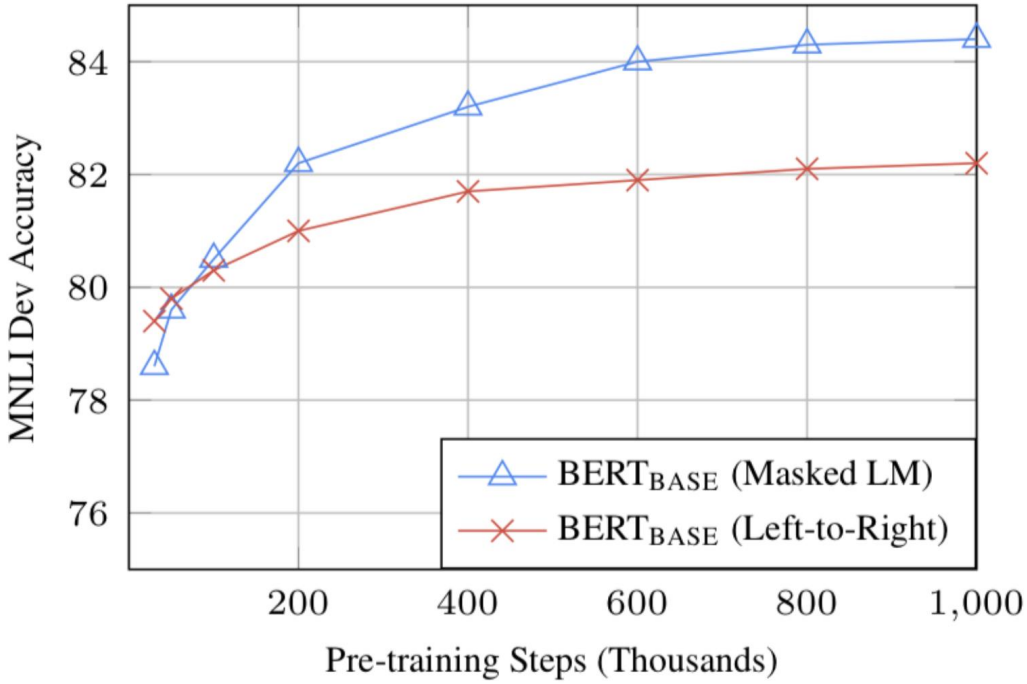


THE UNIVERSITY OF SYDNEY

Resource! Resource!

TPUS… and Resources..

***BERT-Base**: 4 Cloud TPUs (16 TPU chips total) in 4 days*
***BERT-Large**: 16 Cloud TPUs (64 TPU chips total)*



BERT

Questions

- *Why did no one think of this before?*
- *Better Question: Why wasn't contextual pre-training popular before 2018 with ELMo?*
- *Good Results on pre-training is > 1,000 x to 100,000 more expensive than supervised training.*

2 BERT

THE UNIVERSITY OF SYDNEY

Questions

- *The model must be learning more than "contextual embeddings"*
- *Alternate interpretation: Predicting missing words (or next words) requires learning many types of language understanding features*
  - *Syntax, semantics, pragmatics, coreference, etc.*
- *Implication: Pre-trained model is much bigger than it needs to be to solve specific task*
- *Task-specific model distillation words very well*

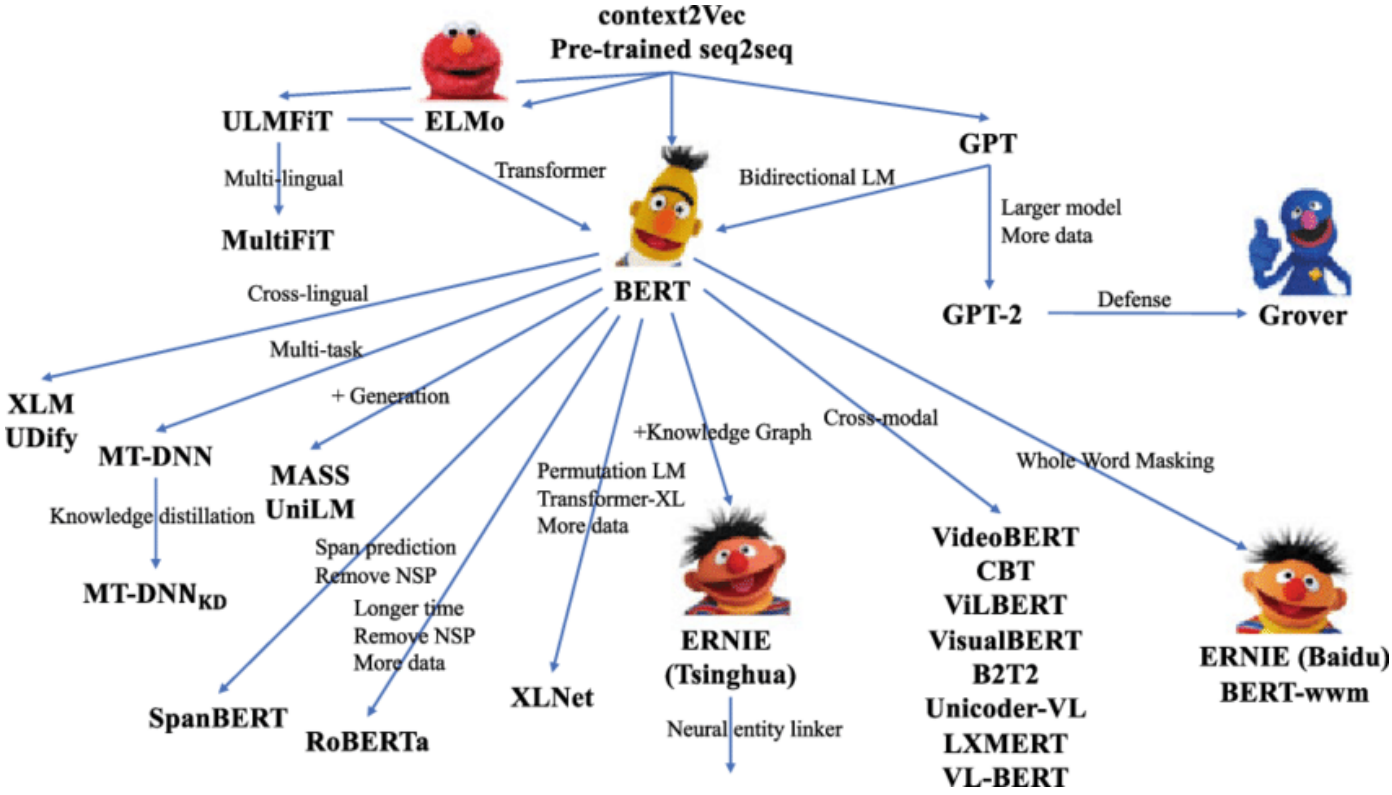
More advanced pre-trained model

	BERT	RoBERTa	DistilBERT	XLNet
Size (millions)	**Base**: 110 **Large**: 340	**Base**: 110 **Large**: 340	**Base:** 66	**Base**: ~110 **Large**: ~340
Training Time	**Base**: 8 x V100 x 12 days* **Large:** 64 TPU Chips x 4 days (or 280 x V100 x 1 days*)	**Large:** 1024 x V100 x 1 day; 4-5 times more than BERT.	**Base**: 8 x V100 x 3.5 days; 4 times less than BERT.	**Large:** 512 TPU Chips x 2.5 days; 5 times more than BERT.
Performance	Outperforms state-of-the-art in Oct 2018	2-20% improvement over BERT	3% degradation from BERT	2-15% improvement over BERT
Data	16 GB BERT data (Books Corpus + Wikipedia). 3.3 Billion words.	160 GB (16 GB BERT data + 144 GB additional)	16 GB BERT data. 3.3 Billion words.	**Base**: 16 GB BERT data **Large**: 113 GB (16 GB BERT data + 97 GB additional). 33 Billion words.
Method	BERT (Bidirectional Transformer with MLM and NSP)	BERT without NSP**	BERT Distillation	Bidirectional Transformer with Permutation based modeling

3 Post BERT

3 Post BERT

Pretrained Model Map



Post BERT

RoBERTa

*A **R**obustly **O**ptimized **BERT** Pretraining **A**pproach (Liu et al, University of Washington and Facebook, 2019)*

- *Trained BERT for more epochs and/or on more data*
  - *Showed that more epochs alone helps, even on same data*
  - *More data also helps*

- *Improved masking and pre-training data slightly*

	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	WNLI	Avg
Single-task single models on dev										
$\text{BERT}_{\text{LARGE}}$	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-	-
$\text{XLNet}_{\text{LARGE}}$	89.8/-	93.9	91.8	83.8	95.6	89.2	63.6	91.8	-	-
RoBERTa	**90.2/90.2**	**94.7**	**92.2**	**86.6**	**96.4**	**90.9**	**68.0**	**92.4**	**91.3**	-

3 Post BERT

THE UNIVERSITY OF SYDNEY

XLNet

Generalized Autoregressive Pretraining *for Language Understanding (Yang et al, CMU and Google, 2019)*

Innovation 1: Relative Position Embedding

- *Sentence: Caren ate a hot dog*
- *Absolute Attention: "How much should dog attend to hot (in any position), and how much should dog in position 4 attend to the word in position 3?"*
- *Relative Attention: "How much should dog attend to hot (in any position) and how much should dog attend to the previous word?"*

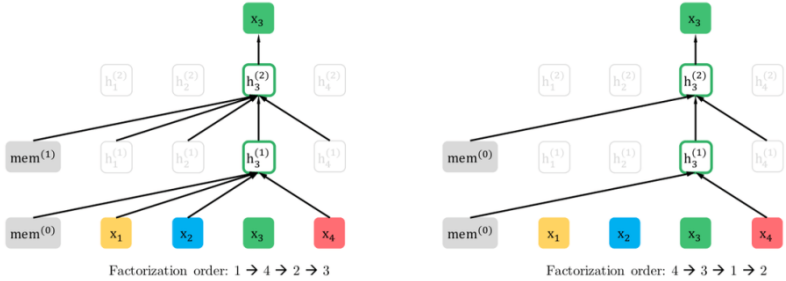
3 Post BERT

XLNet

Generalized Autoregressive Pretraining for Language Understanding (Yang et al, CMU and Google, 2019)

Innovation 2: Permutation Language Modeling

- *In a left-to-right language model, every word is predicted based on all of the words to its left*
- *Instead: Randomly permute the order for every training sentence*
- *Equivalent to masking, but many more predictions per sentence*
- *Can be done efficiently with Transformers*



Post BERT

XLNet

Generalized Autoregressive Pretraining for Language Understanding (Yang et al, CMU and Google, 2019)

- *Also used more data and bigger models, but showed that innovations improved on BERT even with same data and model size*
- *XLNet results:*

Model	MNLI	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS-B
Single-task single models on dev								
BERT [2]	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0
RoBERTa [21]	90.2/90.2	94.7	92.2	**86.6**	96.4	**90.9**	68.0	92.4
XLNet	**90.8/90.8**	**94.9**	**92.3**	85.9	**97.0**	90.8	**69.0**	**92.5**

3 Post BERT

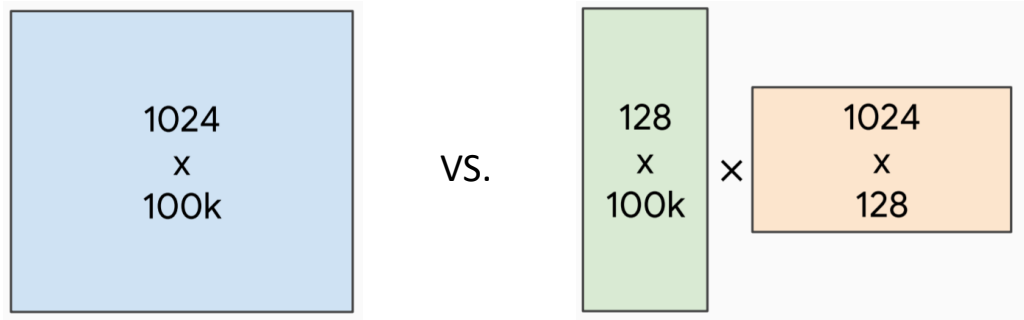
sentence order prediction
not only the next sentence
but also the relations between two sentence
(before or after on other?)

ALBERT

A Lite BERT for Self-supervised Learning of Language Representations (Lan et al, Google and TTI Chicago, 2019)

Innovation 1: Factorized Embedding Parameterisation

- *Use small embedding size (e.g., 128) and then project it to Transformer hidden size (e.g., 1024) with parameter matrix*



Post BERT

ALBERT

Innovation #2: Cross-layer parameter sharing

- *Share all parameters between Transformer layers*

Results:

Models	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS
Single-task single models on dev								
BERT-large	86.6	92.3	91.3	70.4	93.2	88.0	60.6	90.0
XLNet-large	89.8	93.9	91.8	83.8	95.6	89.2	63.6	91.8
RoBERTa-large	90.2	94.7	**92.2**	86.6	96.4	**90.9**	68.0	92.4
ALBERT (1M)	90.4	95.2	92.0	88.1	96.8	90.2	68.7	92.7
ALBERT (1.5M)	**90.8**	**95.3**	**92.2**	**89.2**	**96.9**	**90.9**	**71.4**	**93.0**

ALBERT is light in terms of parameters, not speed

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

3 Post BERT

THE UNIVERSITY OF SYDNEY

T5

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer (Raffel et al, Google, 2019)

Ablated many aspects of pre-training:

- *Model Size*
- *Amount of Training data*
- *Domain/cleanness of training data*
- *Pre-training objective details (e.g. span length of masked text)*
- *Ensembling*
- *Finetuning recipe (e.g. only allowing certain layers to finetune)*
- *Multi-task training*

3 Post BERT

THE UNIVERSITY OF SYDNEY

T5

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer (Raffel et al, Google, 2019)

Conclusions:

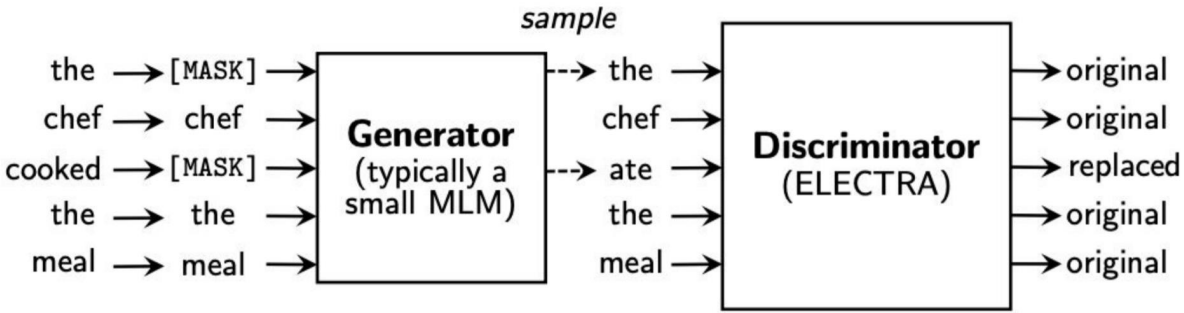
- *Scaling up model size and amount of training data helps a lot*
- *Best model is 11B parameters (BERT-Large is 330M), trained on 120B words of cleaned common crawl text*
- *Exact masking/corruptions strategy does not matter that much*
- *Mostly negative results for better finetuning and multi-task strategies*

Post BERT

ELECTRA

Pre-training Text Encoders as Discriminators Rather Than Generators (Clark et al, 2020)

Train model to discriminate locally plausible text from real text:



Post BERT

ELECTRA

Pre-training Text Encoders as Discriminators Rather Than Generators (Clark et al, 2020)

Difficult to match SOTA results with less compute

Model	Train FLOPs	Params	SQuAD 1.1		SQuAD 2.0	
			EM	F1	EM	F1
BERT-Base	6.4e19 (0.09x)	110M	80.8	88.5	–	–
BERT	1.9e20 (0.27x)	335M	84.1	90.9	79.0	81.8
SpanBERT	7.1e20 (1x)	335M	88.8	94.6	85.7	88.7
XLNet-Base	6.6e19 (0.09x)	117M	81.3	–	78.5	–
XLNet	3.9e21 (5.4x)	360M	**89.7**	**95.1**	87.9	**90.6**
RoBERTa-100K	6.4e20 (0.90x)	356M	–	94.0	–	87.7
RoBERTa-500K	3.2e21 (4.5x)	356M	88.9	94.6	86.5	89.4
ALBERT	3.1e22 (44x)	235M	89.3	94.8	87.4	90.2
BERT (ours)	7.1e20 (1x)	335M	88.0	93.7	84.7	87.5
ELECTRA-Base	6.4e19 (0.09x)	110M	84.5	90.8	80.5	83.3
ELECTRA-400K	7.1e20 (1x)	335M	88.7	94.2	86.9	89.6
ELECTRA-1.75M	3.1e21 (4.4x)	335M	**89.7**	94.9	**88.1**	**90.6**

3 Post BERT

THE UNIVERSITY OF SYDNEY

LongFormer

The Long-Document Transformer (Peters et al., 2020)

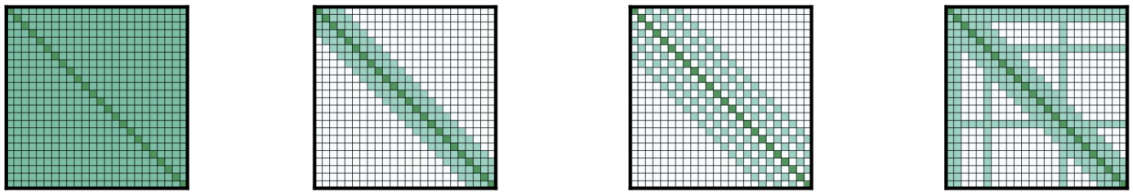
Why?

- *Traditional Transformer-based models are unable to process long sequences due to their self-attention operation, which scales quadratically with the sequence length*
- *To address this, Longformer uses an attention pattern that scales linearly with sequence length, making it easy to process documents of thousands of tokens or longer.*

Post BERT

LongFormer

The Long-Document Transformer (Peters et al., 2020)



(a) Full n^2 attention (b) Sliding window attention (c) Dilated sliding window (d) Global+sliding window

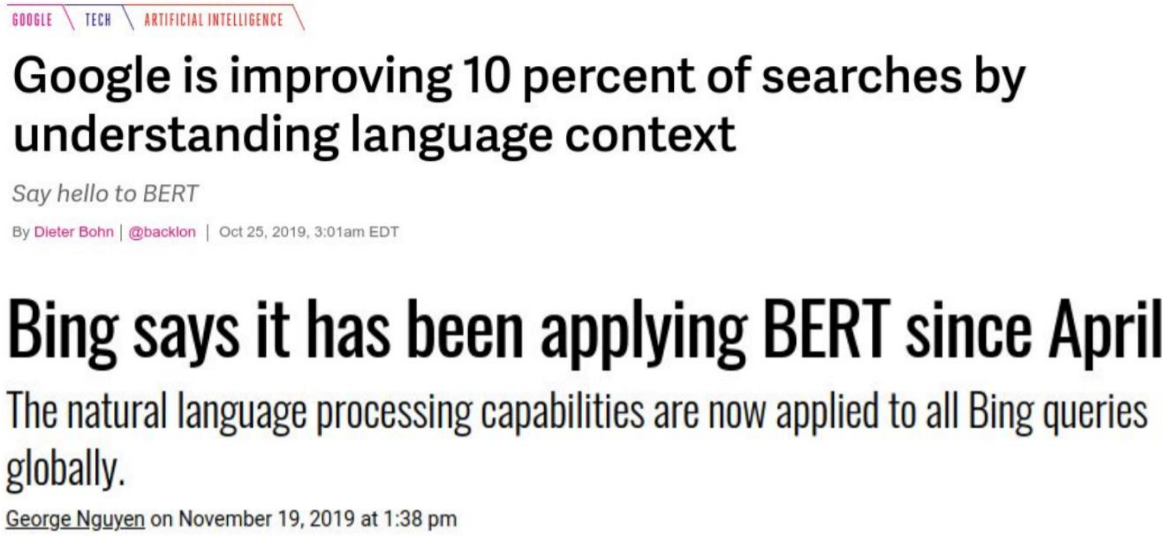
Figure 2: Comparing the full self-attention pattern and the configuration of attention patterns in our Longformer.

3 Post BERT

THE UNIVERSITY OF SYDNEY

Applying Models to Production Services

- *BERT and other pre-trained language models are extremely large and expensive*
- *How are companies applying them to low-latency production services?*



GOOGLE \ TECH \ ARTIFICIAL INTELLIGENCE

Google is improving 10 percent of searches by understanding language context

Say hello to BERT

By Dieter Bohn | @backlon | Oct 25, 2019, 3:01am EDT

Bing says it has been applying BERT since April

The natural language processing capabilities are now applied to all Bing queries globally.

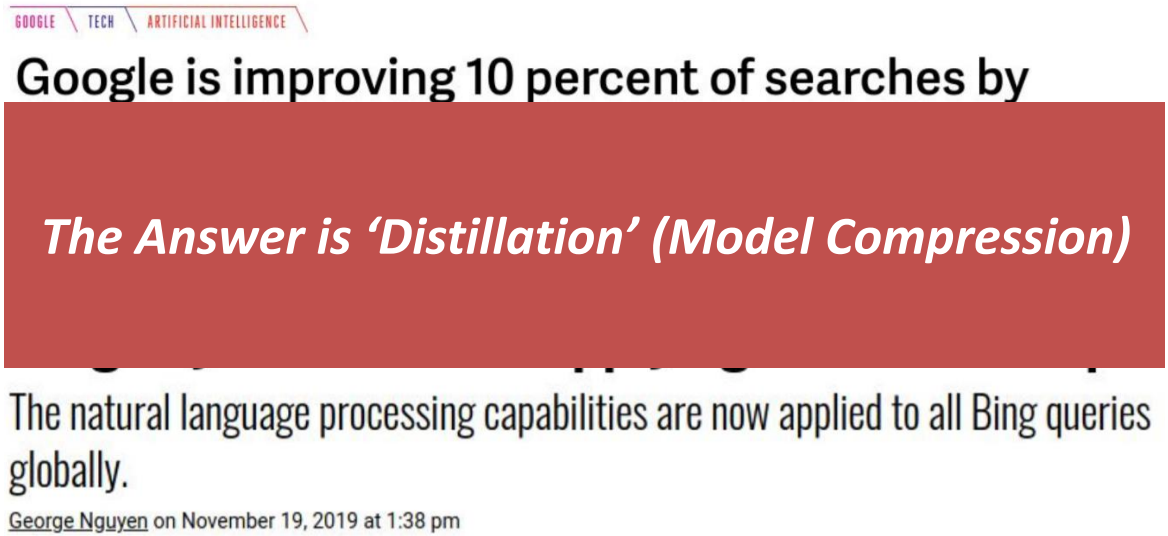
George Nguyen on November 19, 2019 at 1:38 pm

3 Post BERT

THE UNIVERSITY OF SYDNEY

Applying Models to Production Services

- *BERT and other pre-trained language models are extremely large and expensive*
- *How are companies applying them to low-latency production services?*



3 Post BERT

THE UNIVERSITY OF SYDNEY

Distillation (Model Compression)

The idea has been around for a long time (from 2006)

- *Model Compression (Bucila et al. 2006)*
- *Distilling the Knowledge in a Neural Network (Hinton et al., 2015)*

4 Multimodal Pretrained Model

Multimodal Pretrained Model

Video Representation Learning

Supervised Learning*: *Large labelled data with CNN

- Expensive to collect labelled data
- Small corresponding label vocabs: not able to represent the nuances of actions (e.g. difference “sipping” - “drinking” - “gulping”)
- Represent short video clips (a few seconds long)

Unsupervised Learning*: *Learning density models from video

- Single static stochastic variable, decoded into a sequence learning using RNN (VAE-style loss or GAN-style loss)
- Temporal stochastic variable (SV2P/SVCLP) or GAN-based (SAVP/MoCoGAN)
- What if not using explicit stochastic latent variable

Multimodal Pretrained Model

VideoBERT (ICCV 2019)

A Joint Model for Video and Language Representation Learning

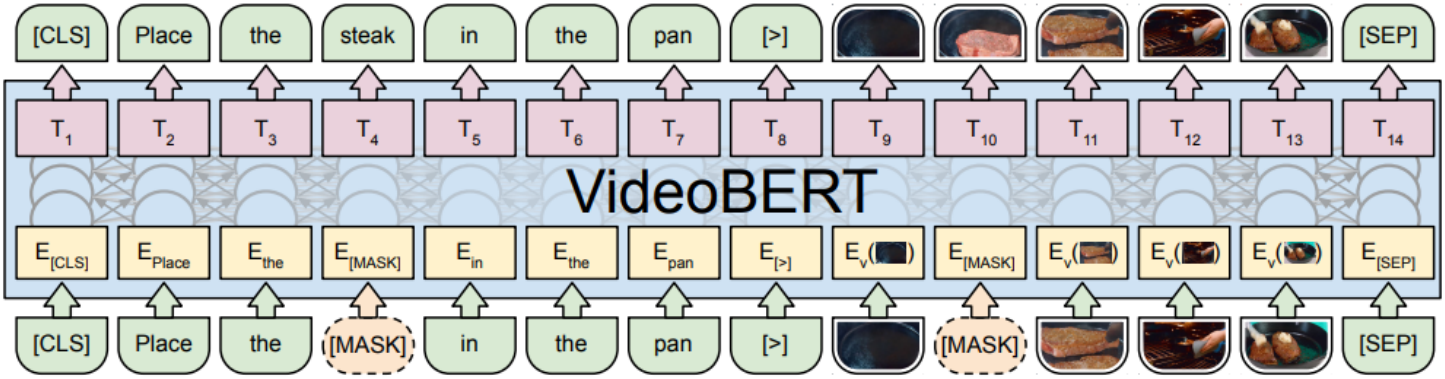
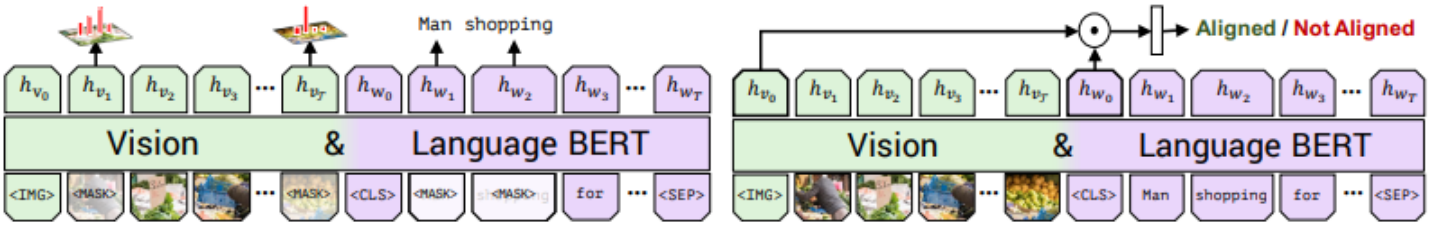


Figure 3: Illustration of VideoBERT in the context of a video and text masked token prediction, or *cloze*, task. This task also allows for training with text-only and video-only data, and VideoBERT can furthermore be trained using a linguistic-visual alignment classification objective (not shown here, see text for details).

Multimodal Pretrained Model

ViLBERT (NIPS 2019)

A Joint Model for Video and Language Representation Learning



(a) Masked multi-modal learning (b) Multi-modal alignment prediction

Figure 3: We train ViLBERT on the Conceptual Captions [24] dataset under two training tasks to learn visual grounding. In masked multi-modal learning, the model must reconstruct image region categories or words for masked inputs given the observed inputs. In multi-modal alignment prediction, the model must predict whether or not the caption describes the image content.

Multimodal Pretrained Model

ViLBERT (NIPS 2019)

Results across all transfer tasks

- *improves performance over a single-stream model*
- *result in improved visiolinguistic representations*
- *Finetuning from ViLBERT is a powerful strategy for vision and language tasks*

Table 1: Transfer task results for our ViLBERT model compared with existing state-of-the-art and sensible architectural ablations. † indicates models without pretraining on Conceptual Captions. For VCR and VQA which have private test sets, we report test results (in parentheses) only for our full model. Our full ViLBERT model outperforms task-specific state-of-the-art models across all tasks.

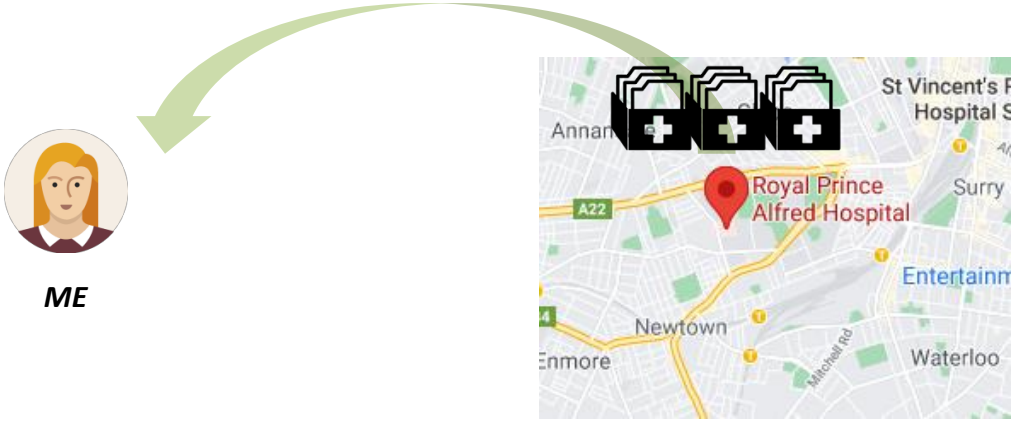
		VQA [3]	VCR [25]			RefCOCO+ [32]			Image Retrieval [26]			ZS Image Retrieval		
	Method	test-dev (test-std)	Q→A	QA→R	Q→AR	val	testA	testB	R1	R5	R10	R1	R5	R10
SOTA	DFAF [36]	70.22 (70.34)	-	-	-	-	-	-	-	-	-	-	-	-
SOTA	R2C [25]	-	63.8 (65.1)	67.2 (67.3)	43.1 (44.0)	-	-	-	-	-	-	-	-	-
SOTA	MAttNet [33]	-	-	-	-	65.33	71.62	56.02	-	-	-	-	-	-
SOTA	SCAN [35]	-	-	-	-	-	-	-	48.60	77.70	85.20	-	-	-
Ours	Single-Stream†	65.90	68.15	68.89	47.27	65.64	72.02	56.04	-	-	-	-	-	-
Ours	Single-Stream	68.85	71.09	73.93	52.73	69.21	75.32	61.02	-	-	-	-	-	-
Ours	ViLBERT†	68.93	69.26	71.01	49.48	68.61	75.97	58.44	45.50	76.78	85.02	0.00	0.00	0.00
Ours	ViLBERT	**70.55 (70.92)**	**72.42 (73.3)**	**74.47 (74.6)**	**54.04 (54.8)**	**72.34**	**78.52**	**62.61**	**58.20**	**84.90**	**91.52**	**31.86**	**61.12**	**72.80**

5 Before we finish this lecture…

0 Before we finish the lecture…

The current ML/DL-based NLP trends

1) *Assume that we collect data from Royal Prince Alfred Hospital*



0 Before we finish the lecture…

The current ML/DL-based NLP trends

1) *Assume that we collect data from Royal Prince Alfred Hospital*

ME

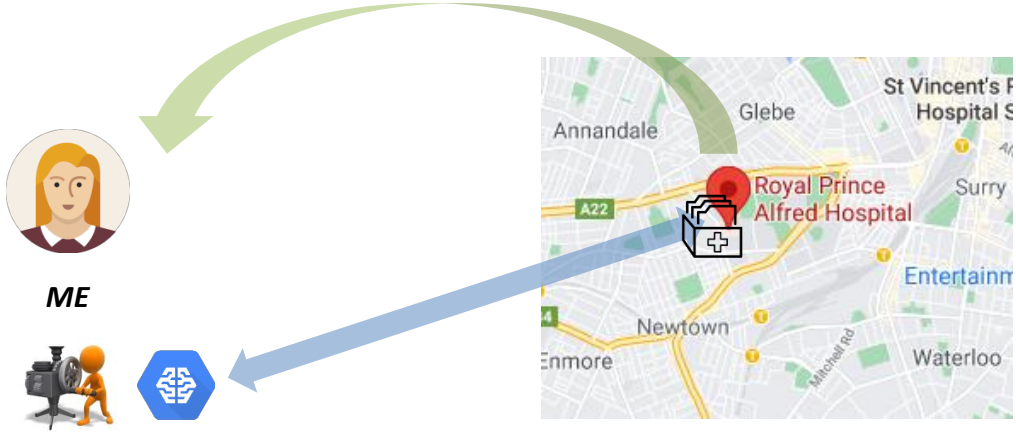
Training with RNN, Transformer, or even pretrained model…

2) Train *and Test on data from the same hospital*

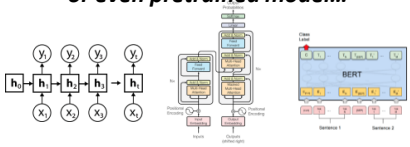
0 Before we finish the lecture…

The current ML/DL-based NLP trends

1) *Assume that we collect data from Royal Prince Alfred Hospital*



Training with RNN, Transformer, or even pretrained model…



2) *Train and* ***Test*** *on data from the same hospital*

0 Before we finish the lecture…

The current ML/DL-based NLP trends

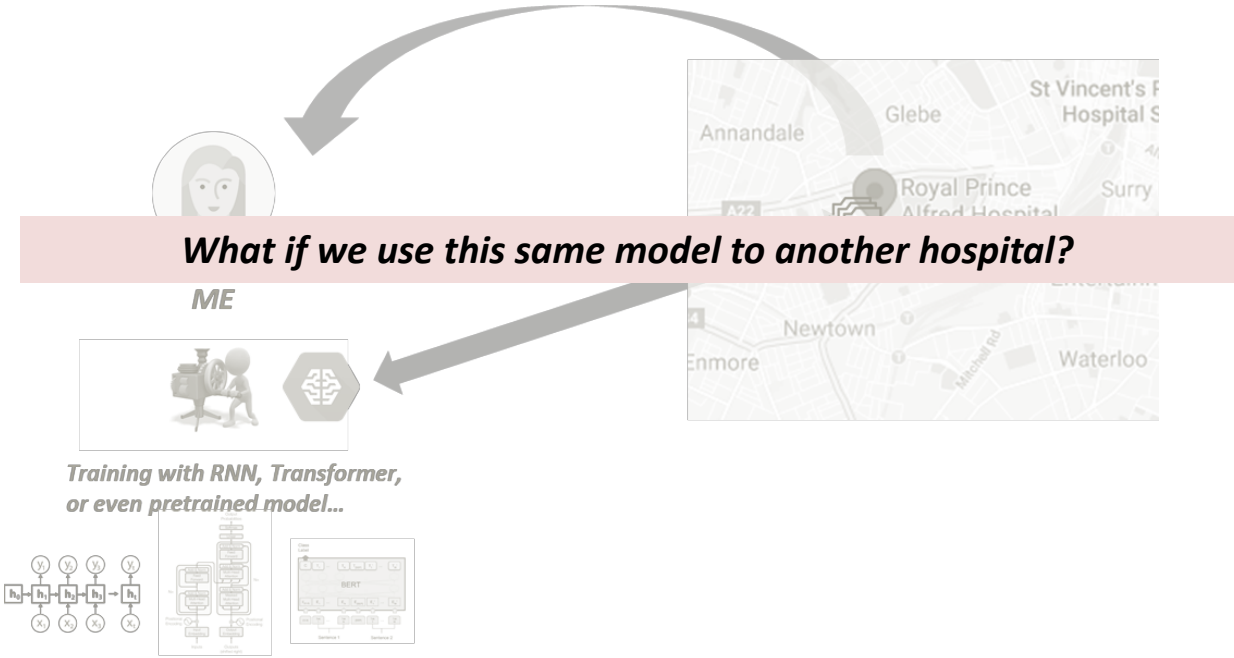
1) Assume that we collect data from Royal Prince Alfred Hospital

*2) Train and **Test** on data from the same hospital*

0 Before we finish the lecture…

The current ML/DL-based NLP trends

1) Assume that we collect data from Royal Prince Alfred Hospital



*2) Train and **Test** on data from the same hospital*

0 Before we finish the lecture…

The current ML/DL-based NLP trends

Assume you take that same DL-based NLP model, to St Vincent's Private Hospital, *with an older testing machine, and the technician there uses a slightly different testing protocol*

ME

0 Before we finish the lecture…

The current ML/DL-based NLP trends

Assume you take that same DL-based NLP model, to St Vincent's Private Hospital, *with an older testing machine, and the technician there uses a slightly different testing protocol*

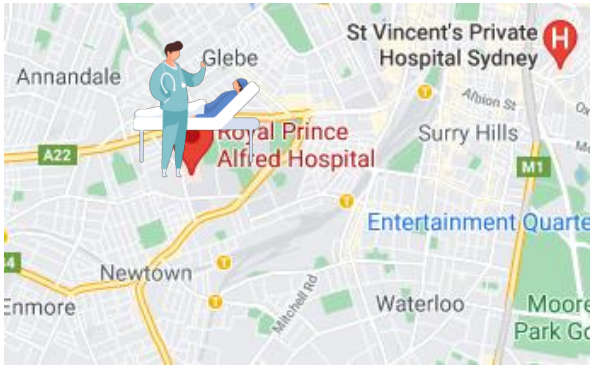
ME

Data drifts to cause the performance of DL-based NLP model to degrade significantly

0 Before we finish the lecture…

The current ML/DL-based NLP trends

In contrast, the Doctor just can walk down the street and diagnose the patient



0 Before we finish the lecture…

The current ML/DL-based NLP trends

When a system is not performing well,
many teams instinctually try to improve the code
(try different model/component or change hyperparameters)

However, for many practical applications,
it is more effective instead to focus on improving the data

0 Before we finish the lecture…

The current ML/DL-based NLP trends

When a system is not performing well,
many teams instinctually try to improve the code
(try different model/component or change hyperparameters)

However, for many practical applications,
it is more effective instead to focus on improving the data

> ***Everyone jokes about ML/DL is 80% data preparation, but no one seems to care***
>
> ***Prof. Andrew Ng (March, 2021)***

0 Before we finish the lecture…

The current ML/DL-based NLP trends

There is unprecedented competition around beating the benchmarks. If Google has BERT then OpenAI has GPT-3.

However, these fancy models take up only 20% of business problems. What differentiates a good deployment is the quality of data.

"Data Dispersion"

Reference

Reference for this lecture

- Deng, L., & Liu, Y. (Eds.). (2018). Deep Learning in Natural Language Processing. Springer.
- Rao, D., & McMahan, B. (2019). Natural Language Processing with PyTorch: Build Intelligent Language Applications Using Deep Learning. " O'Reilly Media, Inc.".
- Manning, C. D., Manning, C. D., & Schütze, H. (1999). Foundations of statistical natural language processing. MIT press.
- Manning, C 2018, Natural Language Processing with Deep Learning, lecture notes, Stanford University

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.
- Miller, A., Fisch, A., Dodge, J., Karimi, A. H., Bordes, A., & Weston, J. (2016). Key-value memory networks for directly reading documents. arXiv preprint arXiv:1606.03126.

- Drawings
- http://jalammar.github.io/illustrated-bert/
- http://jalammar.github.io/illustrated-transformer/