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

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova

Google AI Language

2019 NAACL-HLT (Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies) Best Long Paper

Present@2019/04/26 陳廷易


ELMO
ERNIE
BERT

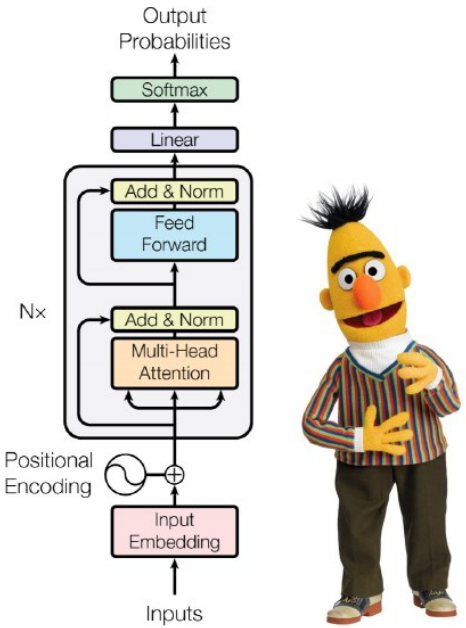
Sesame Street Language Model

- ELMo(**NAACL18** Allenai & University of Washington): **E**mbeddings from **L**anguage **Mo**dels
- BERT(**NAACL19** Google): **B**idirectional **E**ncoder **R**epresentations from **T**ransformers
- ERNIE(Baidu): **E**nhanced **R**epresentation through k**N**owledge **I**nt**E**gration

Outline

- Introduction & Contribution
- Background & Related Work
  - Transformer
  - ELMo、OpenAI GPT
- Pre-Training Tasks
  - Masked Language Model (MLM)
  - Next Sentence Prediction (NSP)
- Input Representation
  - Token Embeddings + Segment Embeddings + Position Embeddings
- Experiments
  - 11 NLP tasks fine-tuning
- Ablation Studies
  - Pre-training tasks
  - Model size
- Feature-based approach
- Conclusion

Introduction



- language representation model
- Idea: contextualized word representations
- Learn word vectors using long contexts using Transformer instead of LSTM

Related Work

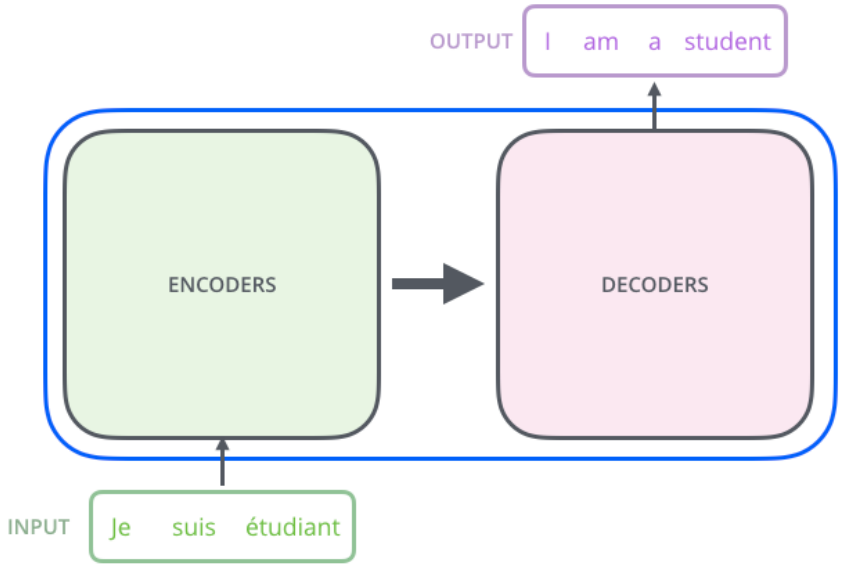
Transformer

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information ProcessingSystems, pages 6000-6010.

Why Transformer?

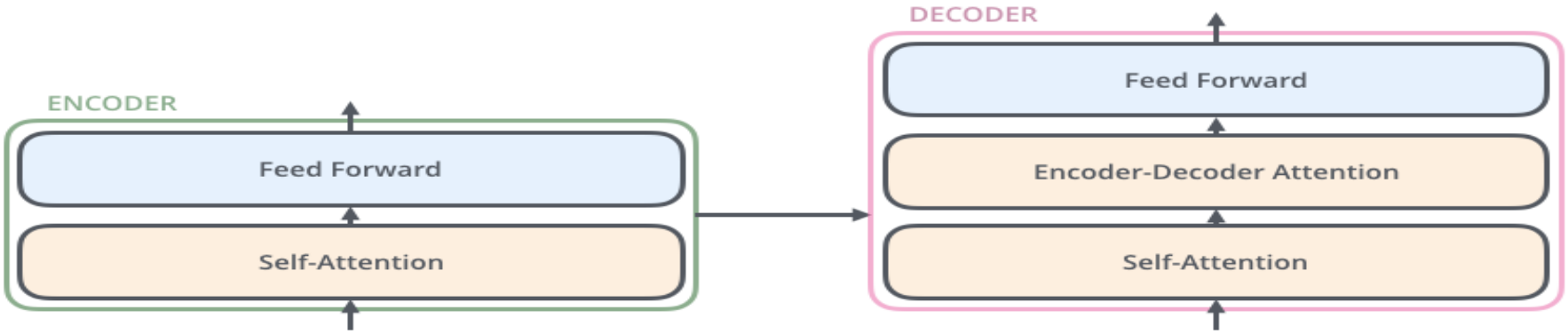
- Uses attention to boost the speed with which these models can be trained.
- Outperforms the Google Neural Machine Translation model in specific tasks
  - The Transformer lends itself to parallelization

Transformer–High-Level Look

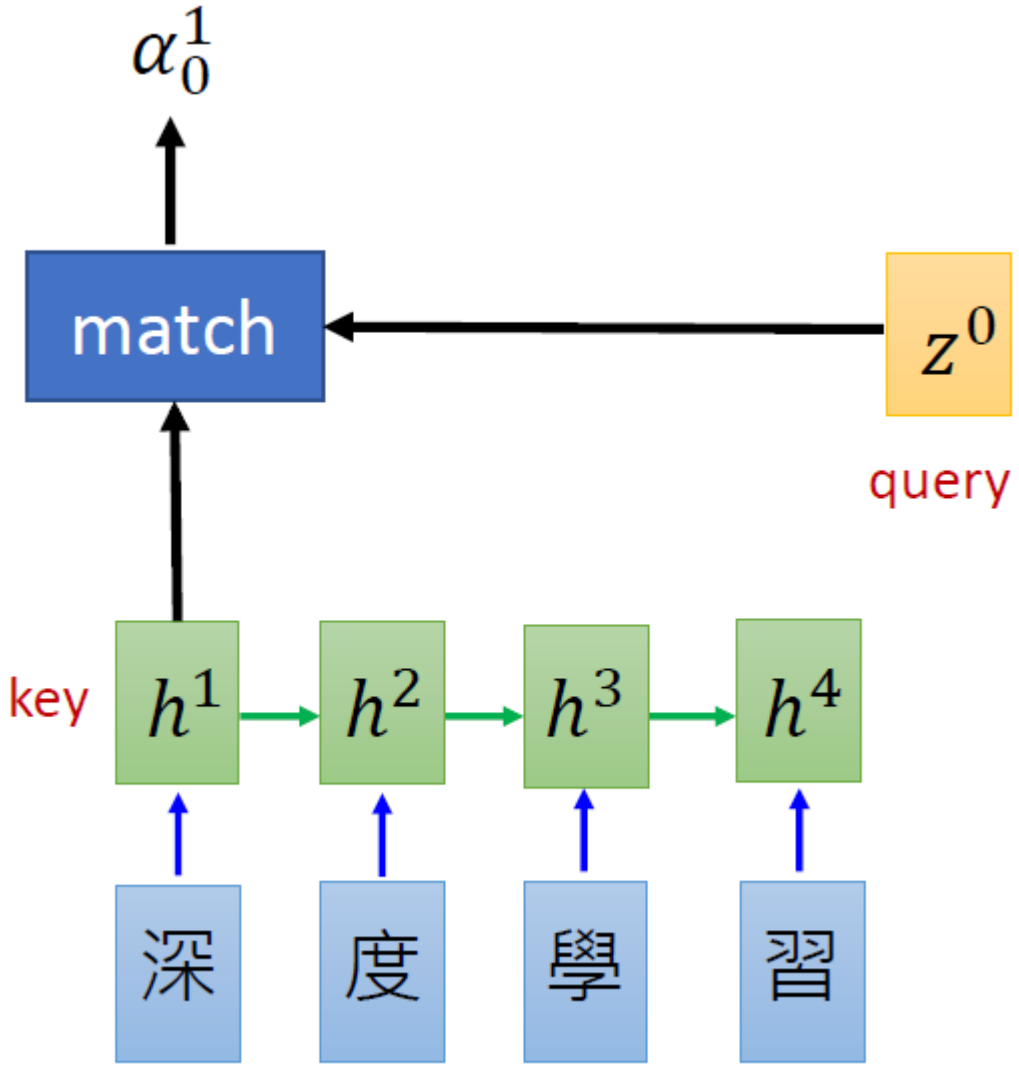


What is Attention

- helped improve the performance of neural machine translation applications
  - highly improved the quality of machine translation systems
- allows the model to focus on the relevant parts of the input sequence as needed
  - to deal with long sentences

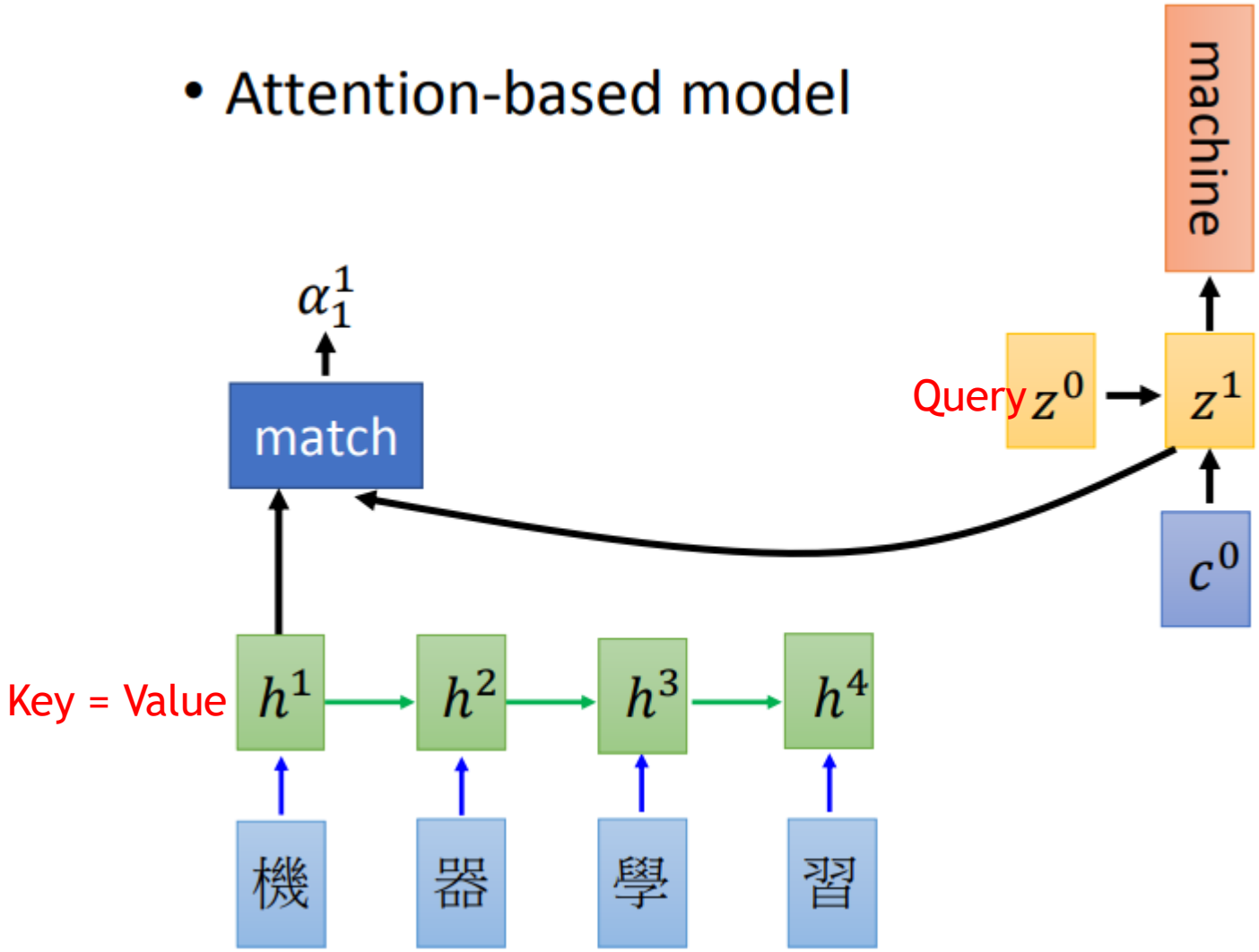


Attention

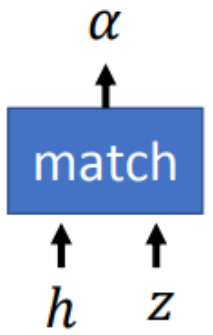


Encoder-Decoder Attention

- Attention-based model



Jointly learned with other part of the network

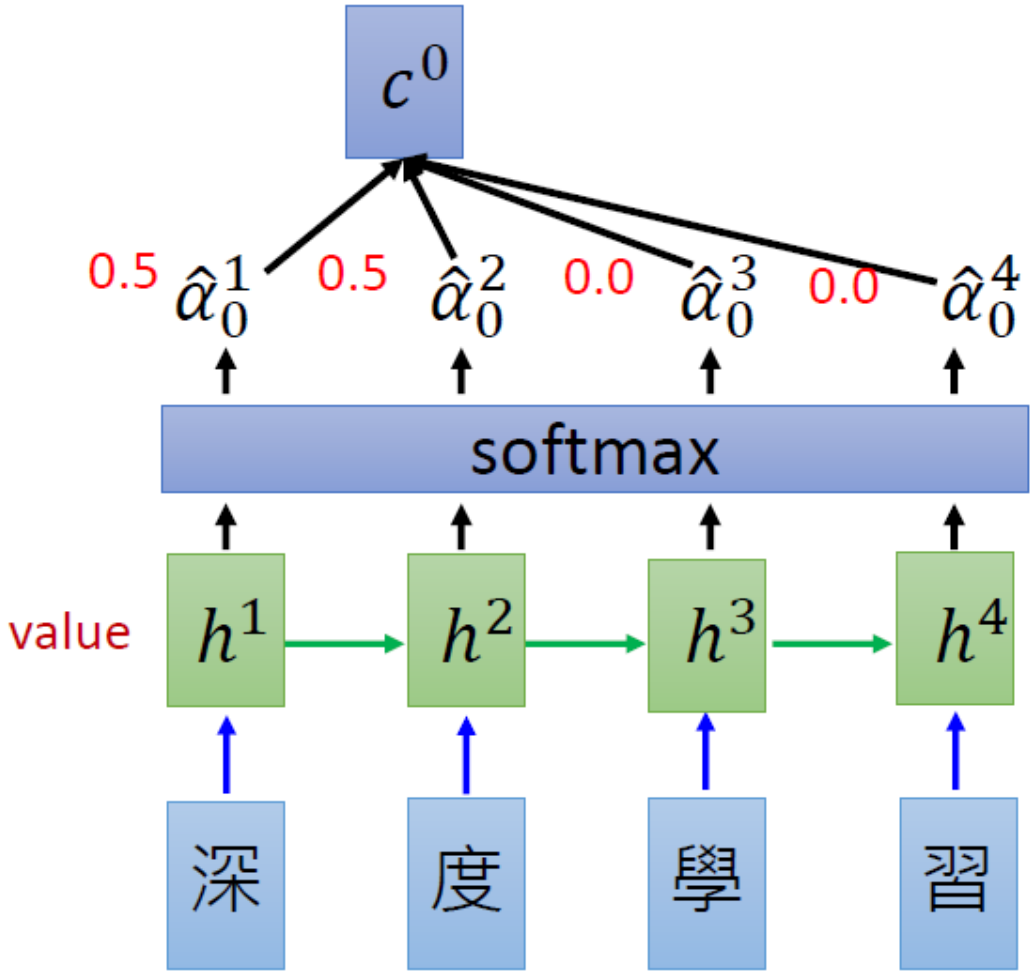


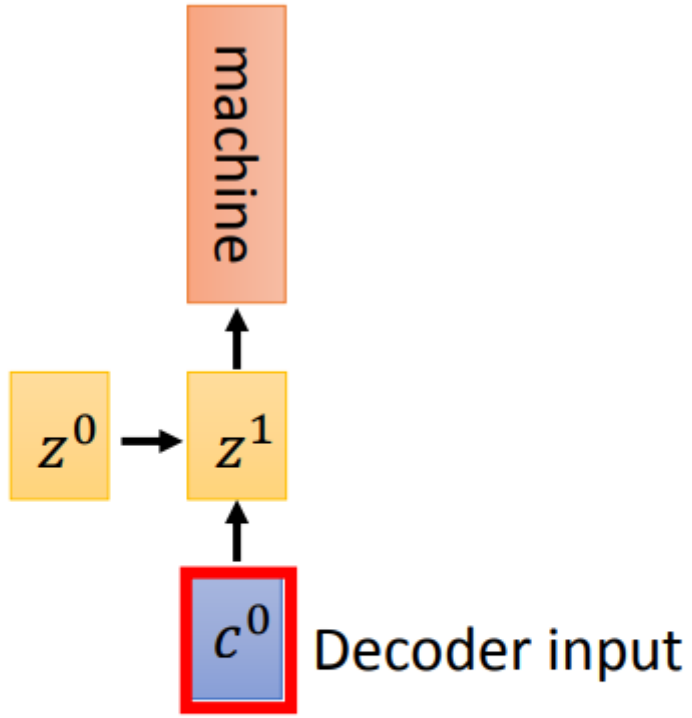
What is match ?

Design by yourself

- Cosine similarity of z and h
- Small NN whose input is z and h, output a scalar
- $\alpha = h^T W z$

Encoder-Decoder Attention

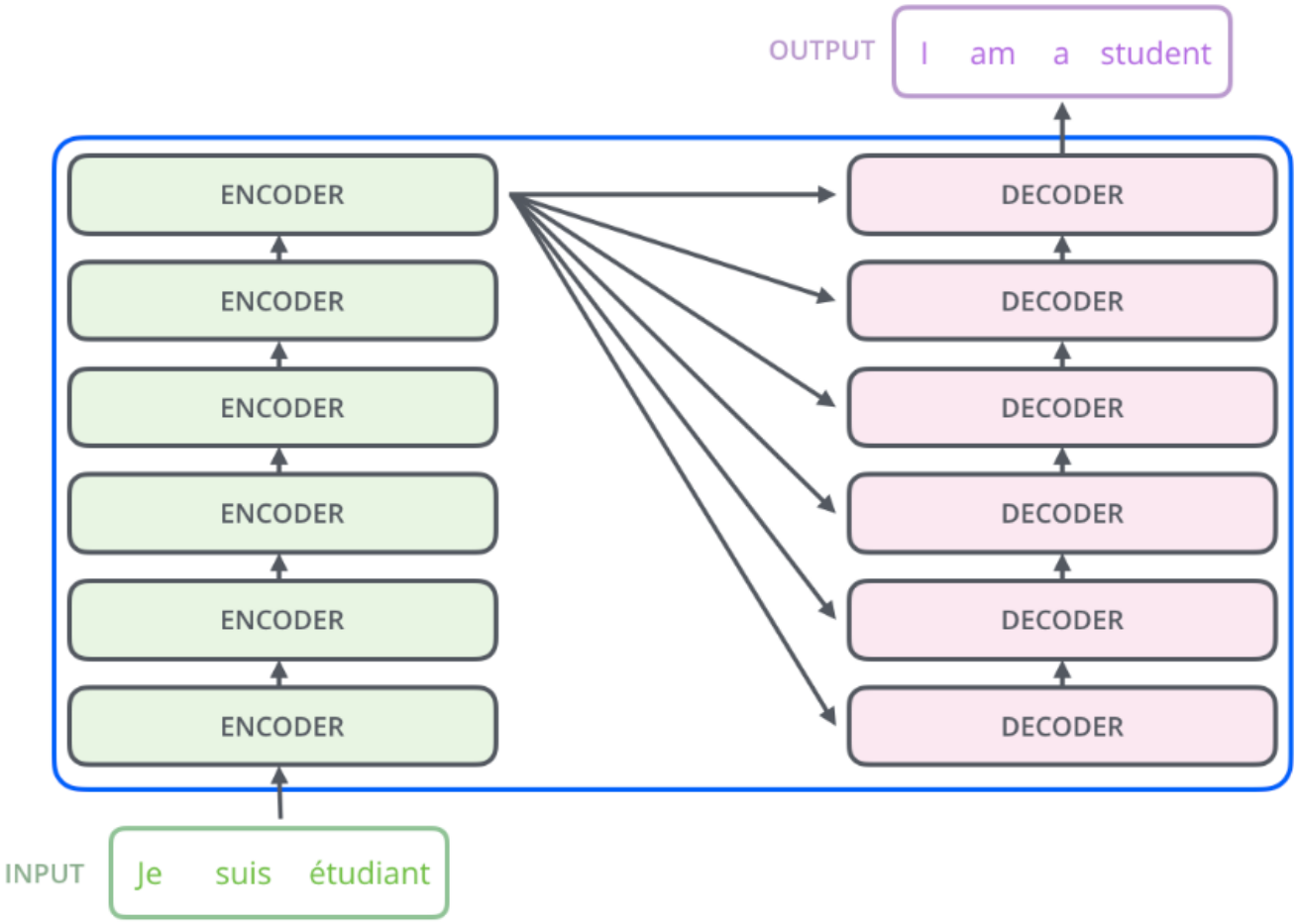


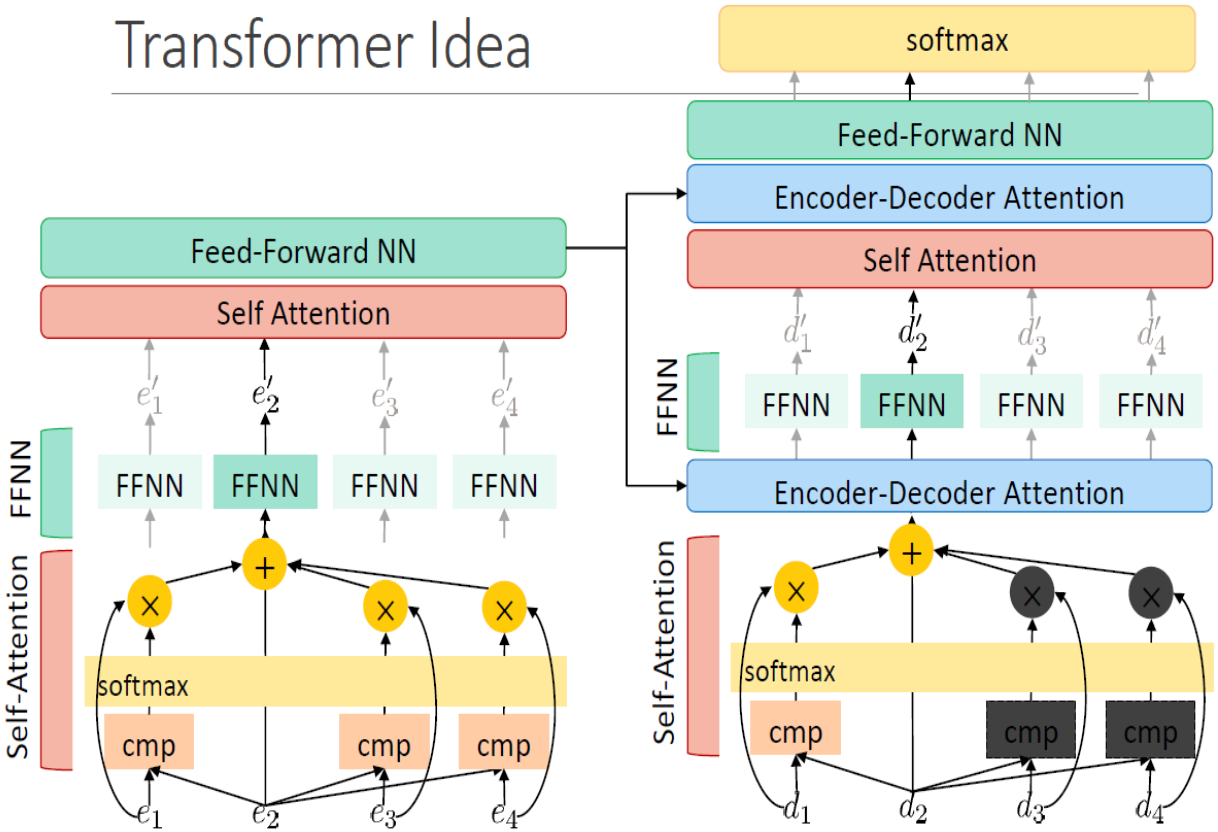


$$c^0 = \sum \hat{\alpha}_0^i h^i$$

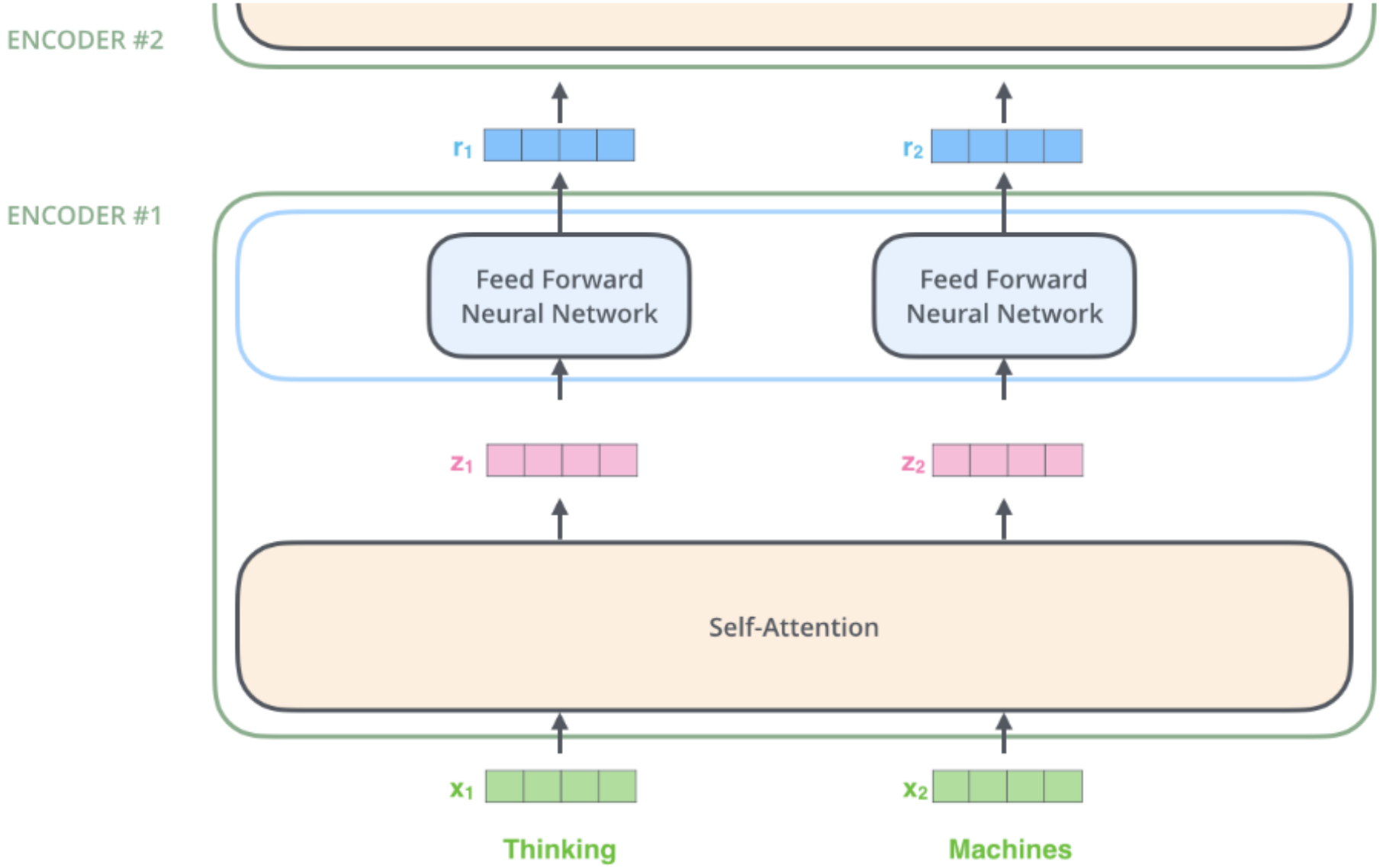
$$= 0.5h^1 + 0.5h^2$$

Transformer–High-Level Look



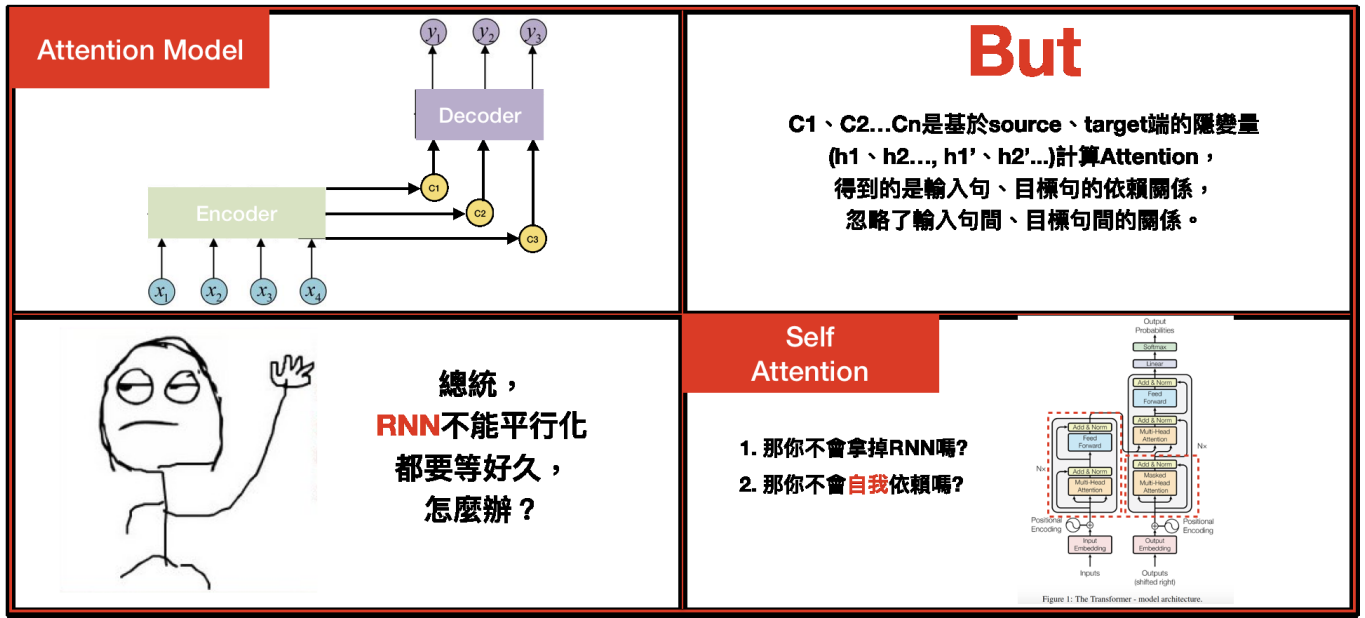


Transformer–High-Level Look

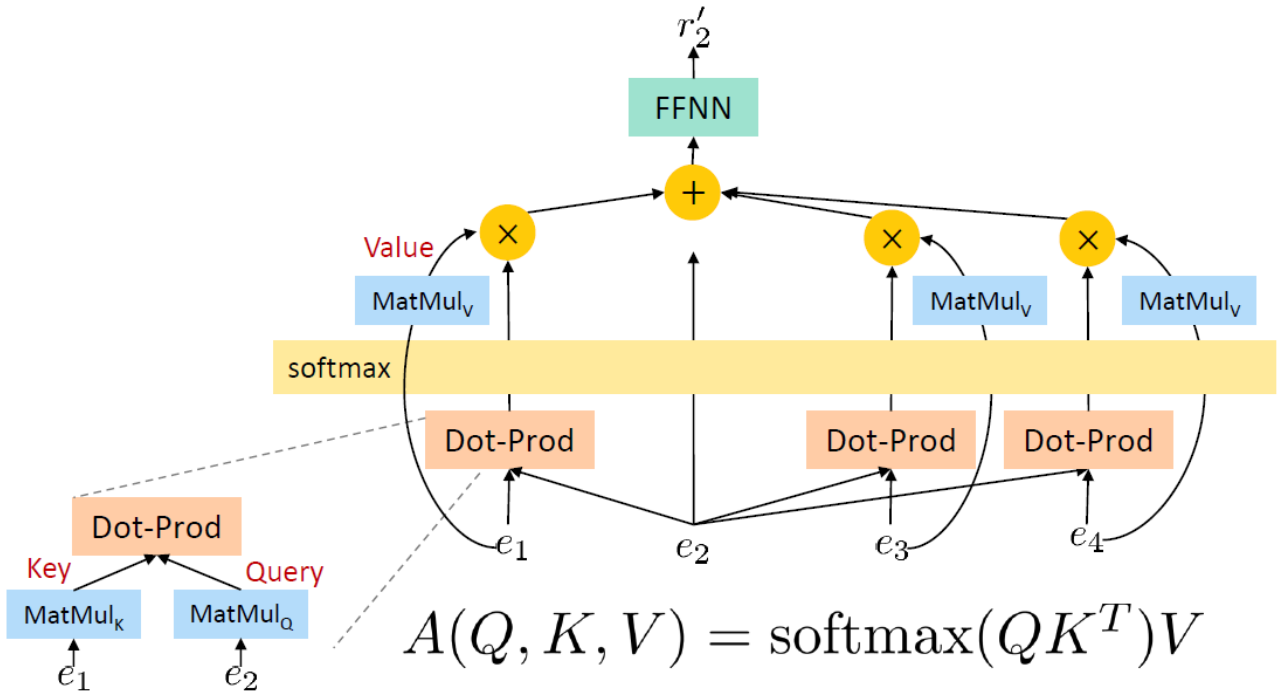


What is Self-Attention?

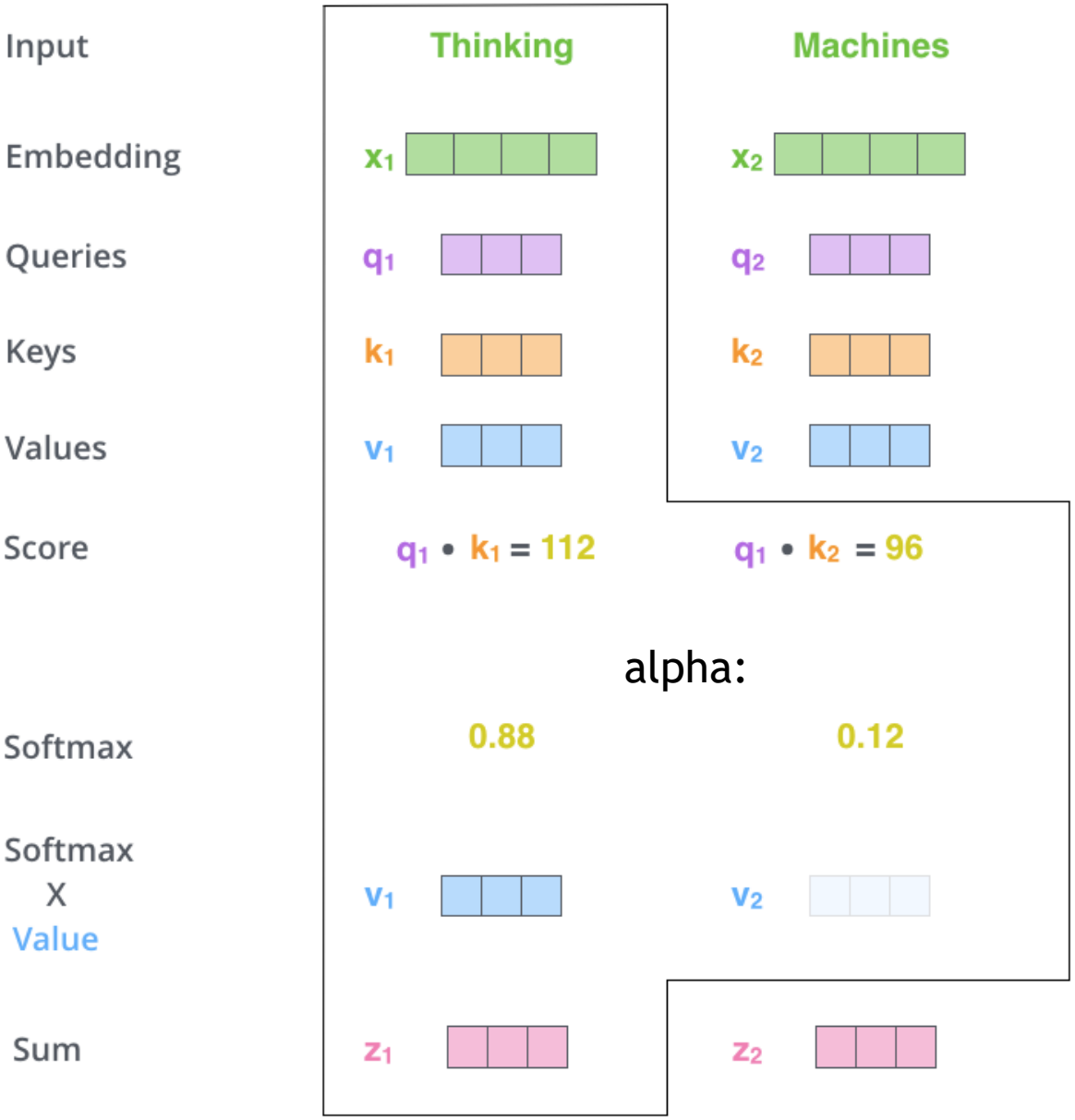
- As the model processes each word (each position in the input sequence), self attention allows it to look at other positions in the input sequence for clues that can help lead to a better encoding for this word.
- Self-attention is the method the Transformer uses to bake the “understanding” of other relevant words into the one we’re currently processing



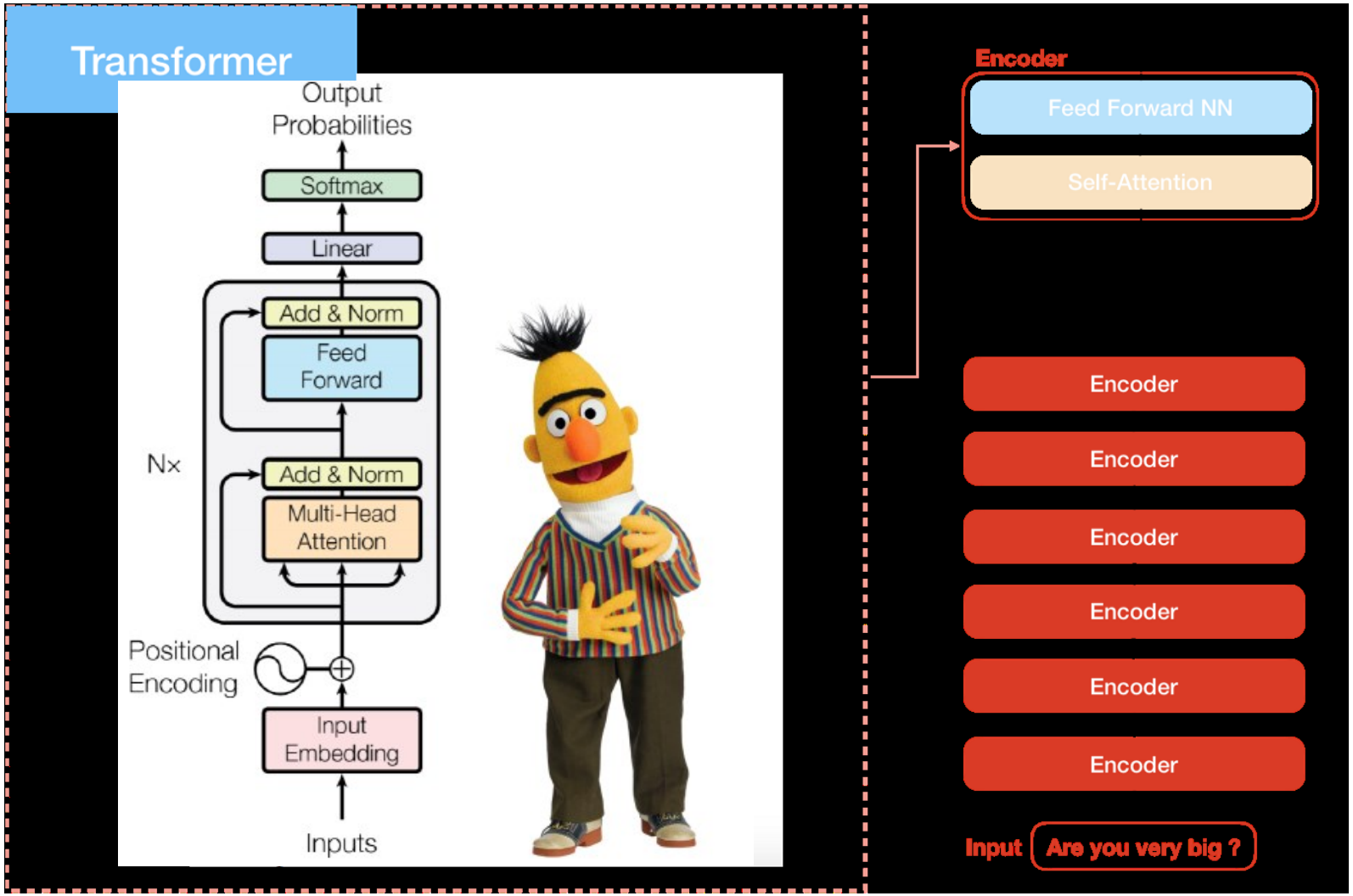
Self-Attention



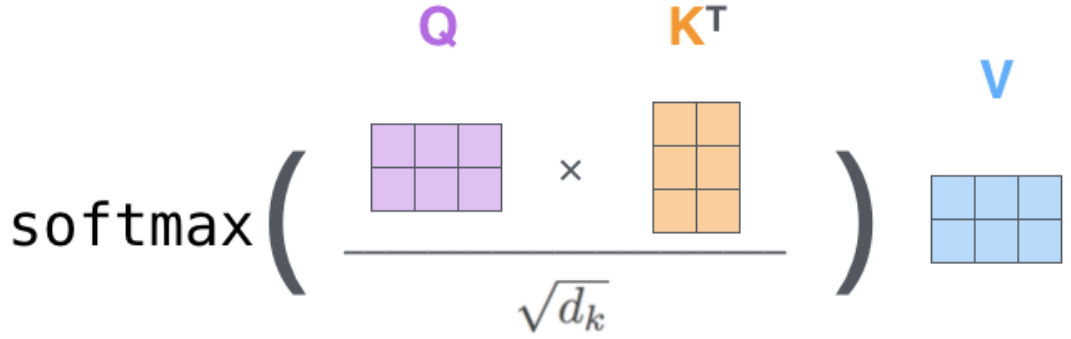
$$A(Q, K, V) = \text{softmax}(QK^T)V$$

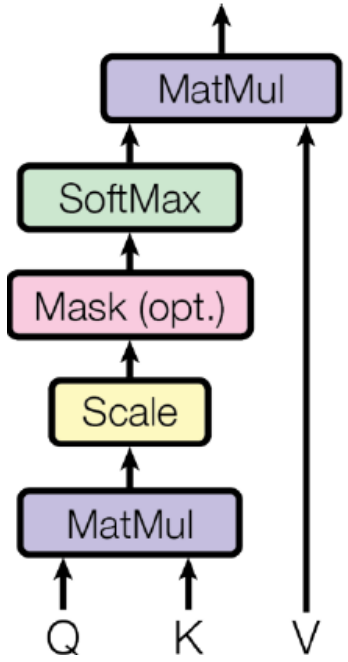


BERT = Transformer Encoder



Self-Attention Matrix Calculation

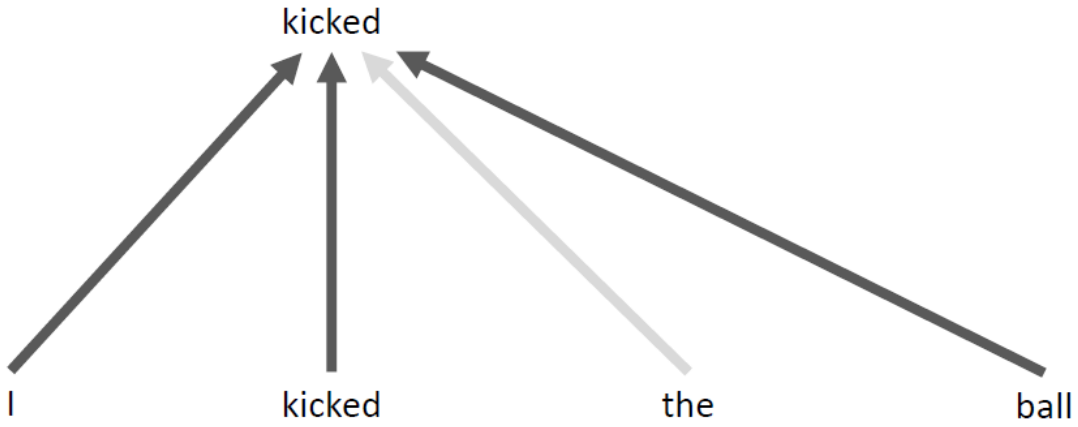


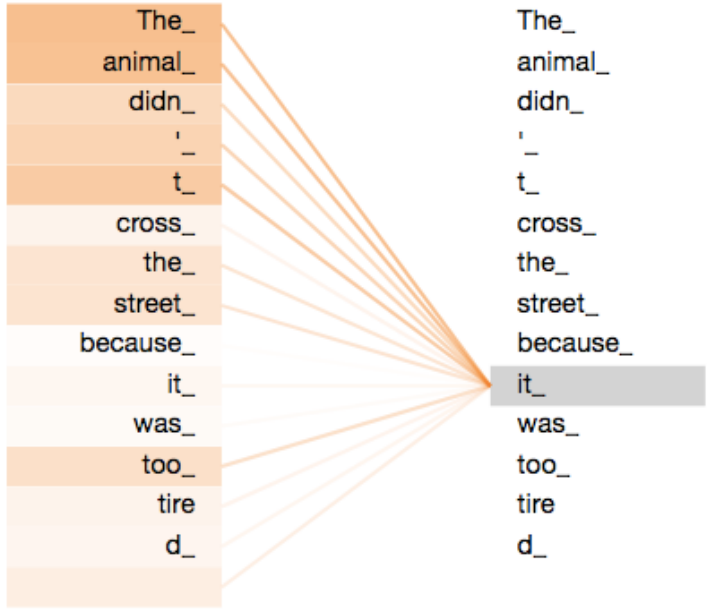


$$A(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Why Multi-head Attention?

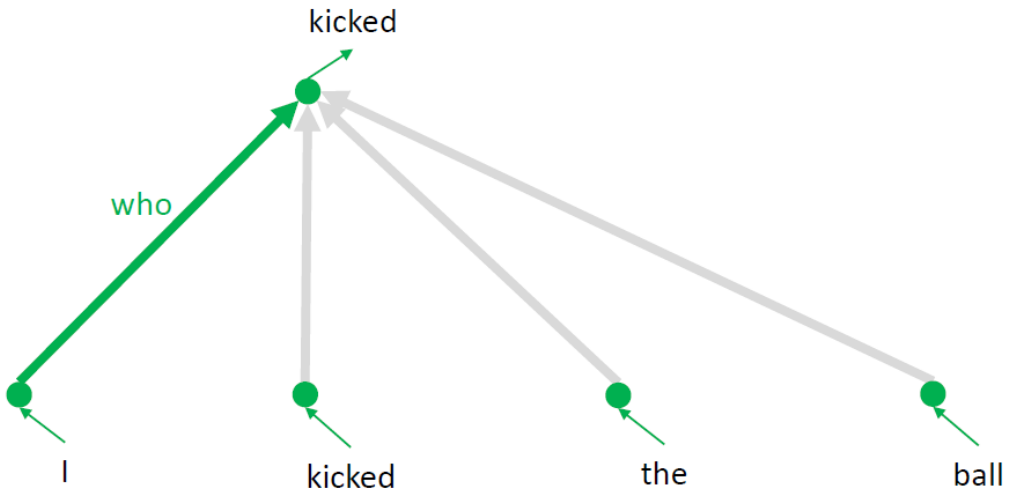
- expands the model's ability to focus on different positions.
- It gives the attention layer multiple "representation subspaces"
  - multiple sets of Query/Key/Value weight matrices (for each encoder/decoder)
  - Each of these sets is randomly initialized
  - after training, each set is used to project the input embeddings into a different representation subspace



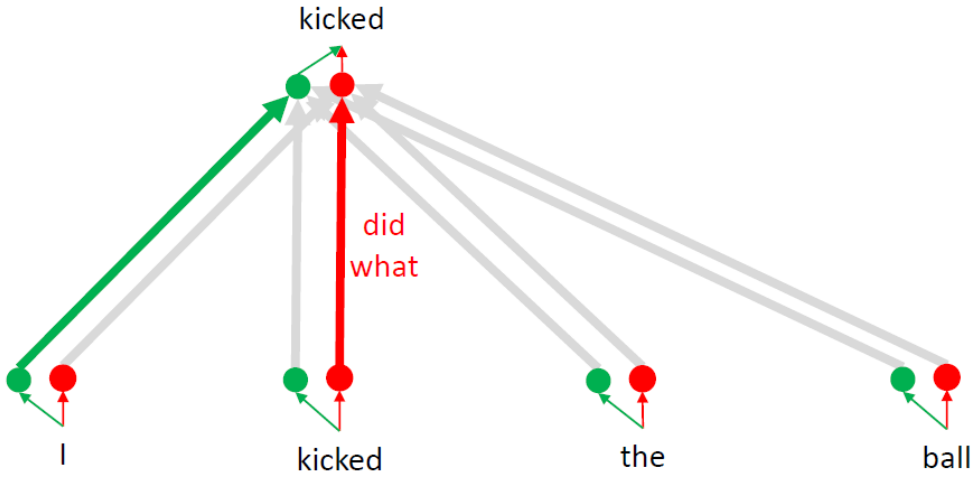


Different attention head represent different meaning

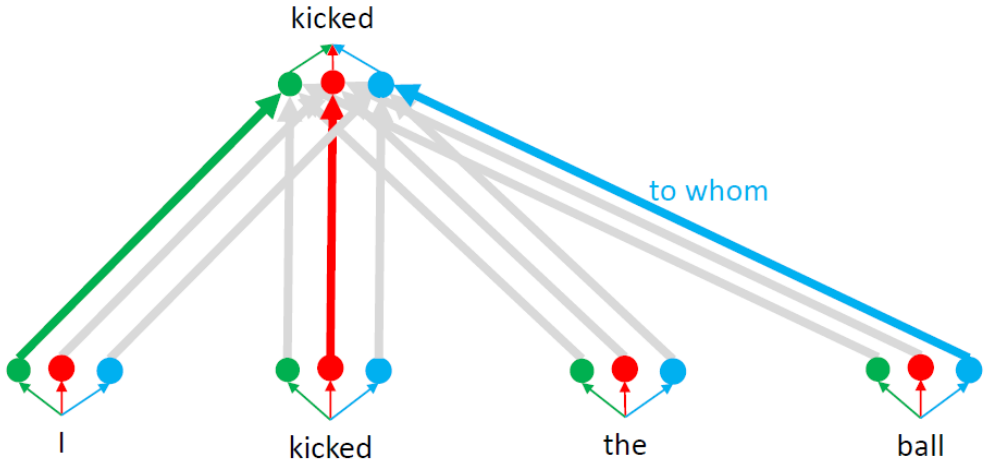
Attention Head: who

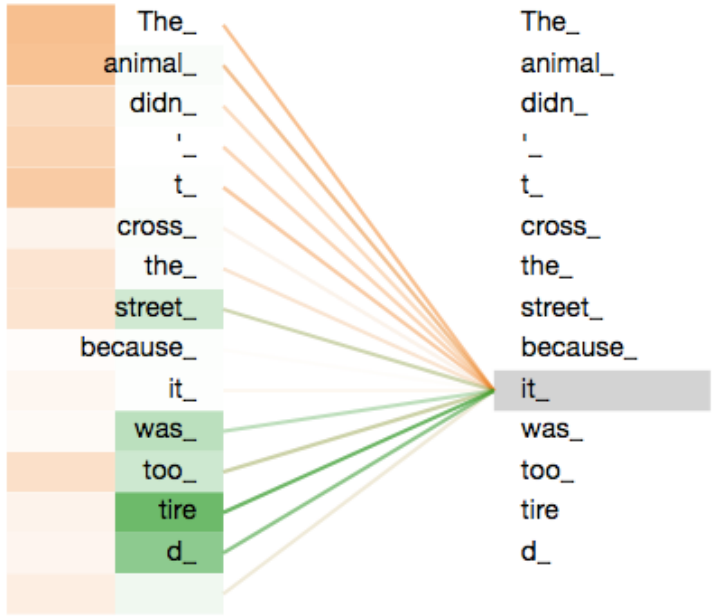


Attention Head: did what

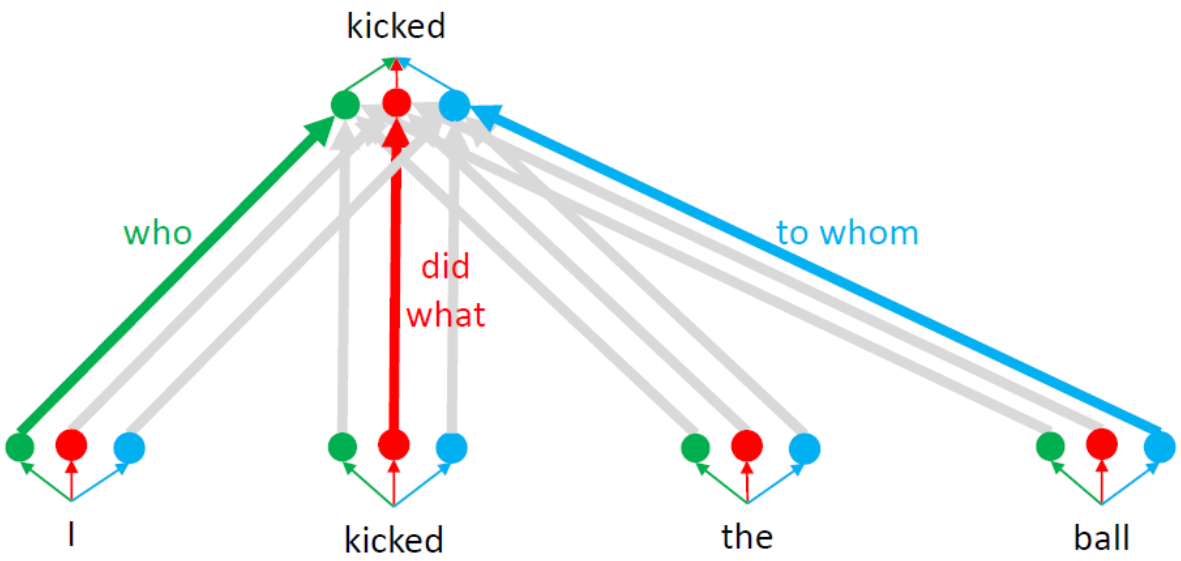


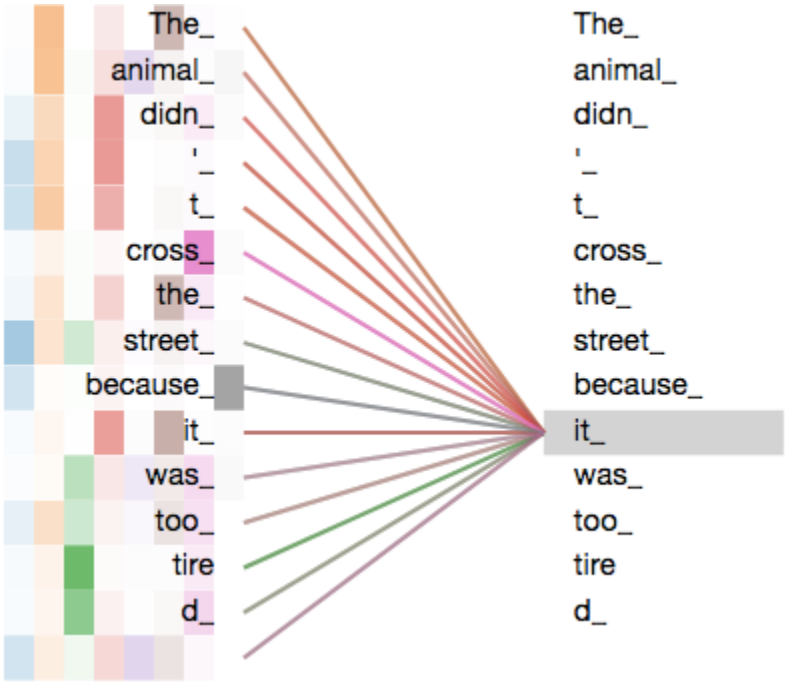
Attention Head: to whom



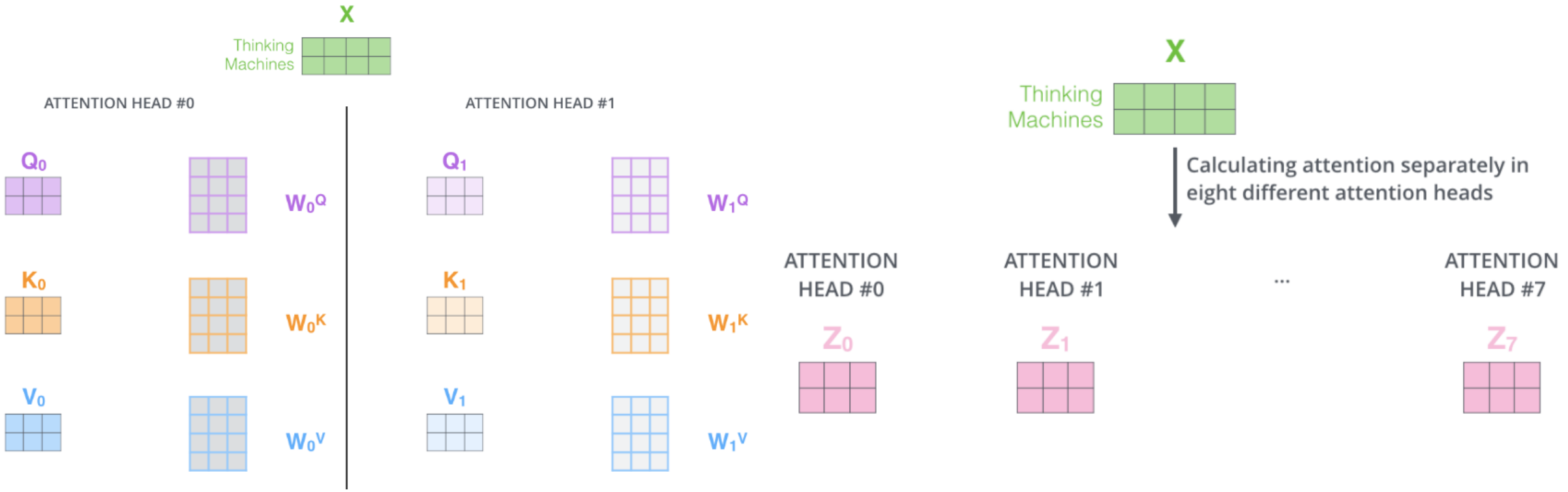


Multi-Head Attention





Multi-Head Attention Matrix Calculation

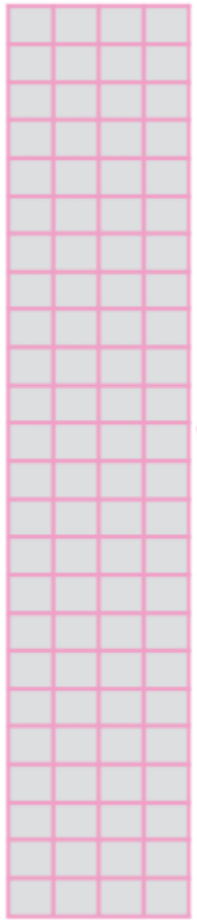


Multi-Head Attention Matrix Calculation

1) Concatenate all the attention heads

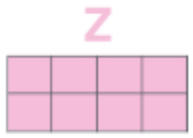
2) Multiply with a weight matrix W^O that was trained jointly with the model

X



3) The result would be the Z matrix that captures information from all the attention heads. We can send this forward to the FFNN

=



Multi-Head Attention Matrix Calculation

1) This is our input sentence*

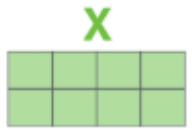
2) We embed each word*

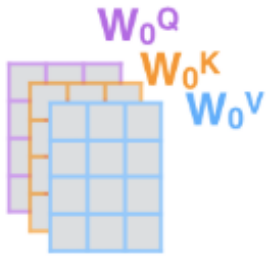
3) Split into 8 heads. We multiply X or R with weight matrices

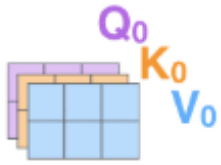
4) Calculate attention using the resulting Q/K/V matrices

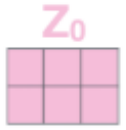
5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer

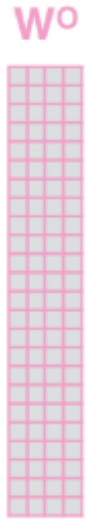
Thinking
Machines



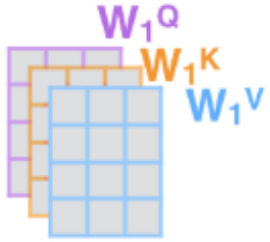


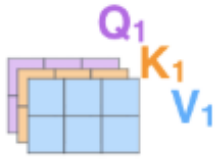


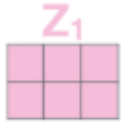




* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one



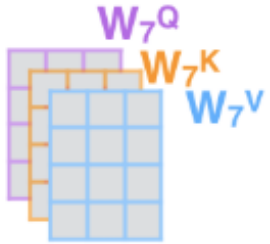


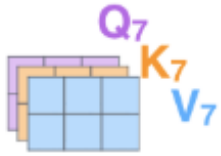


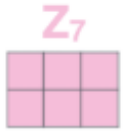
...

...

...







Layer Normalization & Residual Connections

Transformer

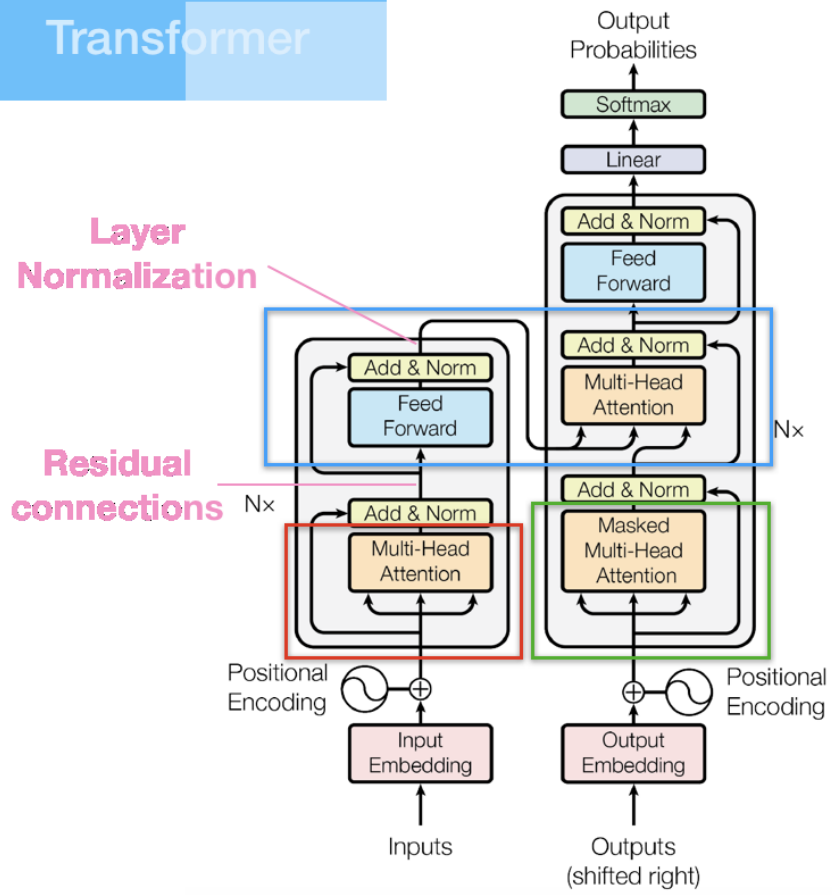
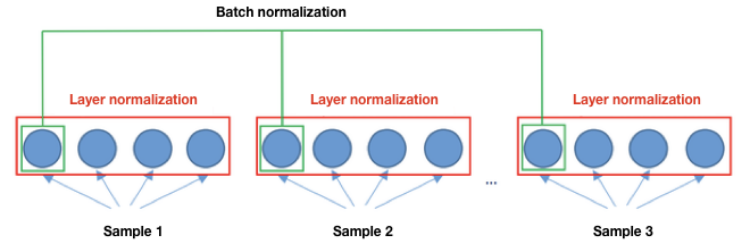


Figure 1: The Transformer - model architecture.

Layer Normalization

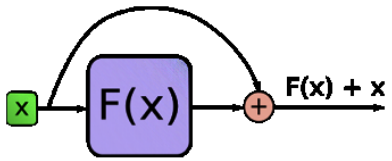


優點： 針對單樣本，不依賴其他數據

避免Interval covariate shift (ICS)

Residual Connections

$$H(x) = g(x) = x + F(x)$$

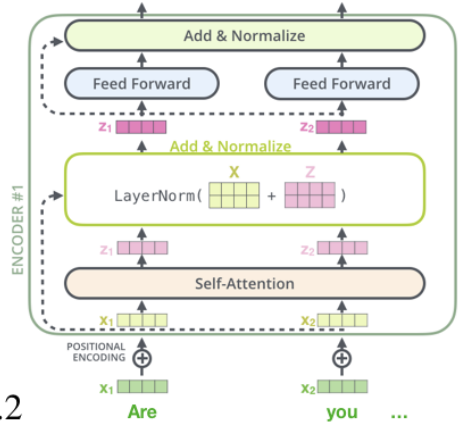


優點： 僅需擬合 $F(x)$，∵ x=輸入

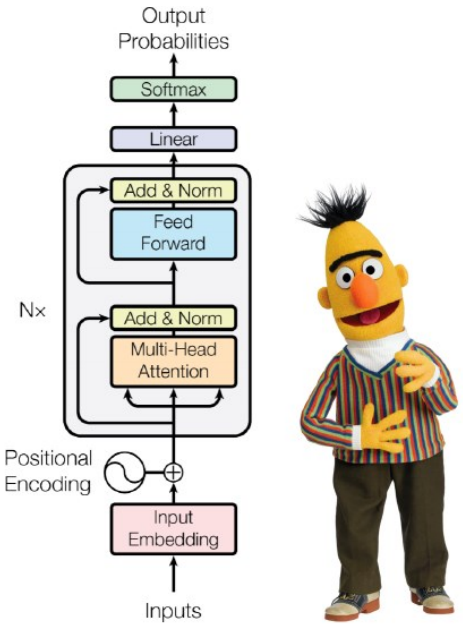
可注意到微小變化

$$Ex: x = 2.9 \begin{cases} H_1(x) = 3,\ F(x) = 0.1 \\ H_2(x) = 3.1,\ F(x) = 0.2 \end{cases}$$

$$Plain\ layer = 3.3\,\%,\ Residual\ block = 100\,\%$$



BERT Introduction



- language representation model
- Idea: contextualized word representations
- Learn word vectors using long contexts using Transformer instead of LSTM
- pre-train deep bidirectional representations by jointly conditioning on both left and right context in all layers
  - Deep bidirectional Transformer model
  - **bidirectional** self-attention (**Transformer encoder**)
- pre-trained BERT representations can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks

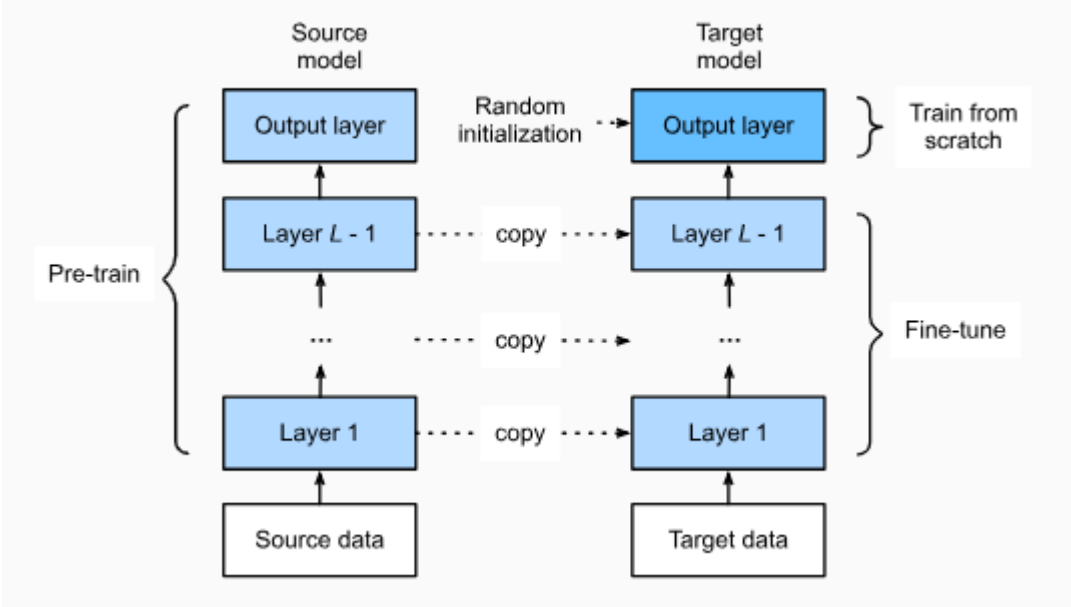
Pre-trained

- 預訓練 (訓練好的模型參數)
- 之前被訓練好的model, 比如很大、很耗時間的model, 又不想從頭training一遍。可以直接使用他人訓練好的model，裡面保存的都是每一層的parameter情況
- 有了這樣的model之後，就可以直接拿來做testing，或訓練其他down-stream task

Model	Size	Parameters	Depth
Xception	88 MB	22,910,480	126
VGG16	528 MB	138,357,544	23
VGG19	549 MB	143,667,240	26
ResNet50	99 MB	25,636,712	168
InceptionV3	92 MB	23,851,784	159
InceptionResNetV2	215 MB	55,873,736	572
MobileNet	17 MB	4,253,864	88
DenseNet121	33 MB	8,062,504	121
DenseNet169	57 MB	14,307,880	169
DenseNet201	80 MB	20,242,984	201

Fine-Tune (Transfer learning)

- 微調 (few parameters need to be learned from scratch)
- 原本預訓練模型已經具備提取淺層基礎特徵的能力跟深層抽取抽象特徵的能力
- 為了節省計算時間跟運算資源，也為了防止模型難以收斂或參數訓練不好而過度擬合
- 透過有較小的labeled data可以做到快速訓練，也可以加入其他層以符合所需要的task加速收斂



Transfer Learning

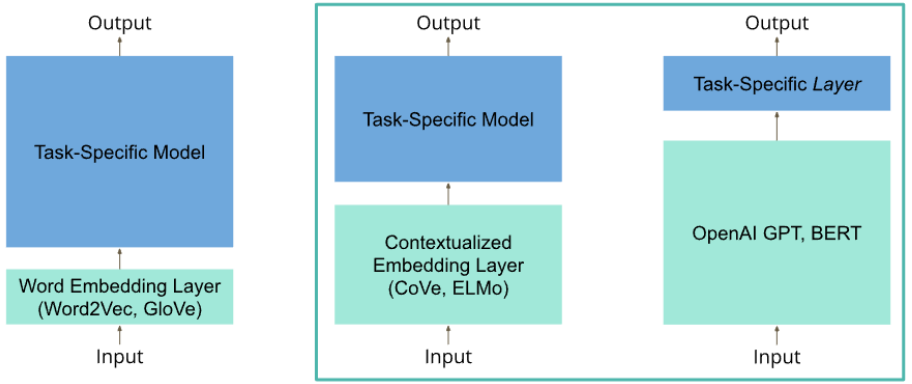
- Pre-trained + Fine-tuned
- unsupervised pre-training is that there is a nearly unlimited amount of data available
- 利用少量的標記資料在進行fine-tune可以獲得很好的成效
  - EX: natural language inference、machine translation
- CV領域也證明利用ImageNet dataset進行pre-train過的在去fine-tune表現會比較好

ImageNet

- 目前世界上圖像識別最大的資料庫，Stanford，模擬人類的識別系統，目的是為了從圖片識別物體
- 擁有超過1400萬的圖像URL被ImageNet Community進行手動標註，具有兩萬多個類別
- **ImageNet Large Scale Visual Recognition Challenge**
  - 訓練集包含1281167張圖片，驗證集包含50000張圖片，測試集為100000張圖片
  - 影像分類的任務是要判斷圖片中物體在1000個分類中所屬的類別
  - 對於每張圖給出5次猜測結果，只要5次中有一次命中真實類別就算正確分類，最後統計沒有命中的錯誤率 (top-5)

Strategies for applying pre-trained language representations

- feature-based approach
  - ELMo (Peters et al., 2018)
  - uses tasks-specific architectures that include the pre-trained representations as additional features
- fine-tuning approach
  - Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018)
  - minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning the pretrained parameters



Related Work

OpenAI GPT

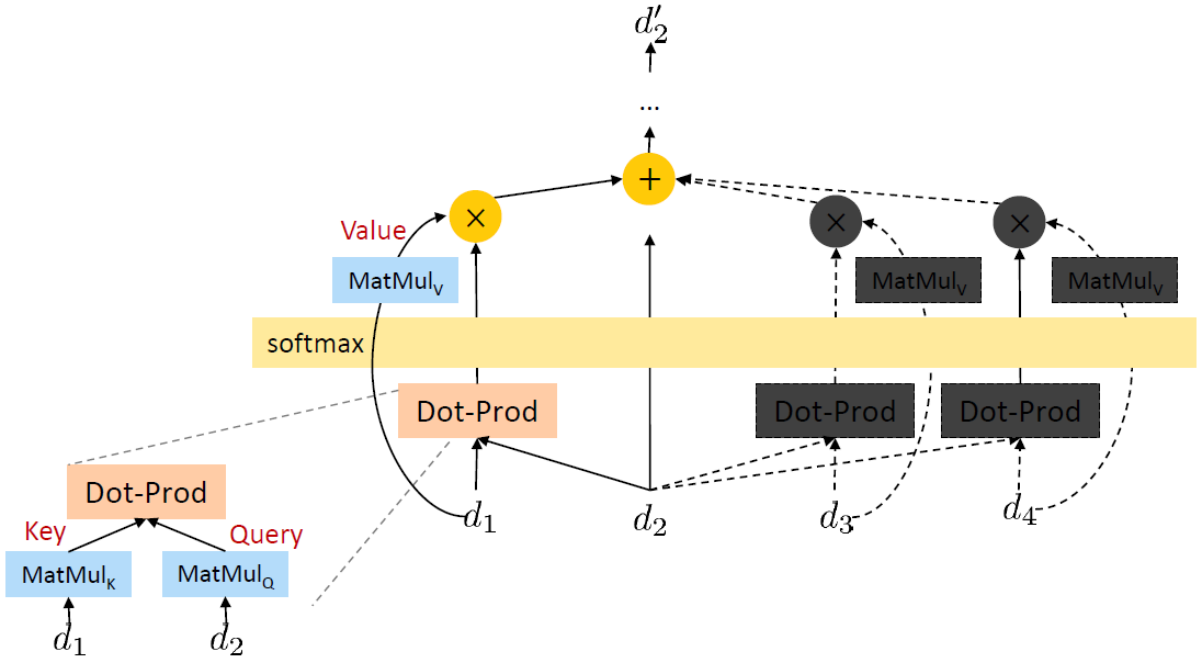
Alec Radford, Karthik Narasimhan, Tim Salimans, andIlya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report,OpenAI.

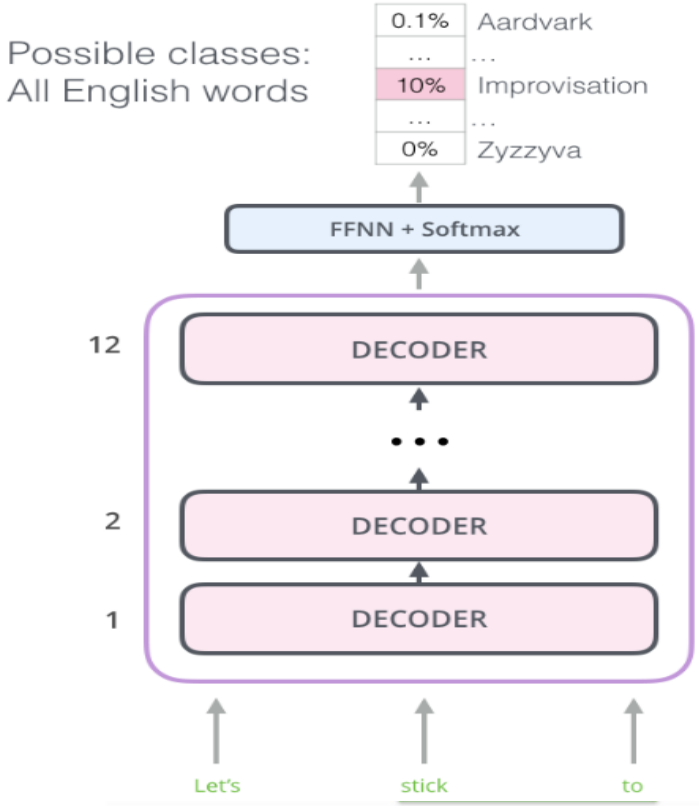
OpenAI GPT

- left-to-right architecture (LTR Transformer decoder)
  - constrained self-attention where every token can only attend to context to its left
  - Transformer decoder=>text generation
- every token can only attended to previous tokens in the self-attention layers of the Transformer
- BERT作者認為當利用LTR的傳統模型架構來訓練token-level task時表現通常不太好
  - 例如: SQuAD question answering, 如果上下前後文都考慮其內容的話表現才會比較好

Decoder Self-Attention

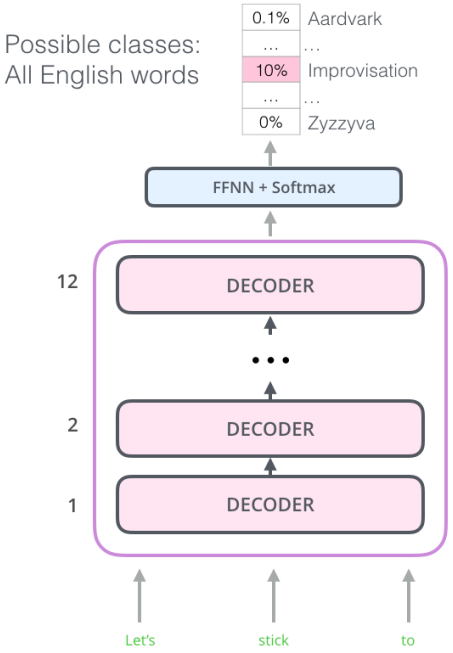
- In the decoder, the self-attention layer is only allowed to attend to earlier positions in the output sequence.
  - It's a natural choice for language modeling (predicting the next word) since it's built to mask future tokens – a valuable feature when it's generating a translation word by word
  - Masking future positions before the Softmax step

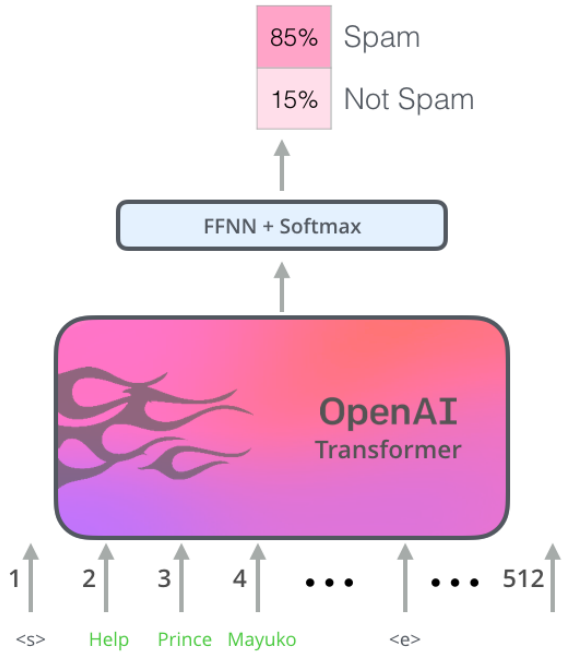




OpenAI Transformer

- train the model on the same language modeling task
  - predict the next word using massive (unlabeled) datasets
- 當pre-trained完成以後，可以用至downstream task，如sentence classification (垃圾郵件分類器)





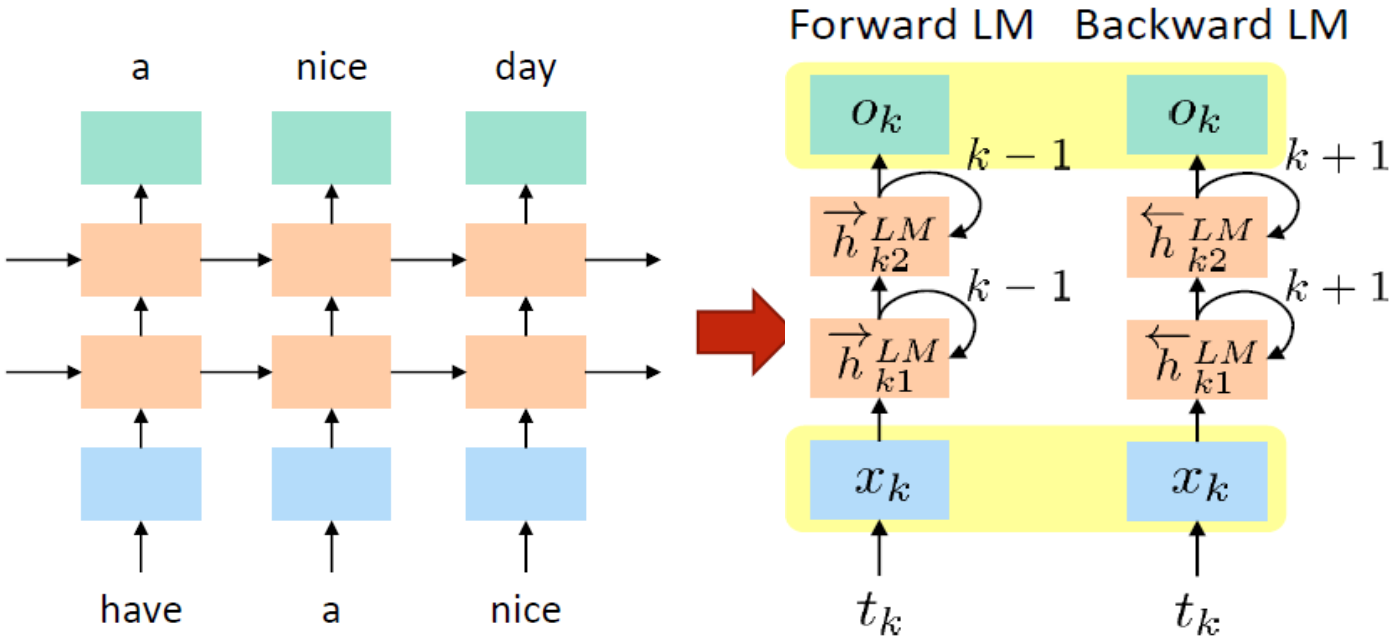
Related Work

ELMo

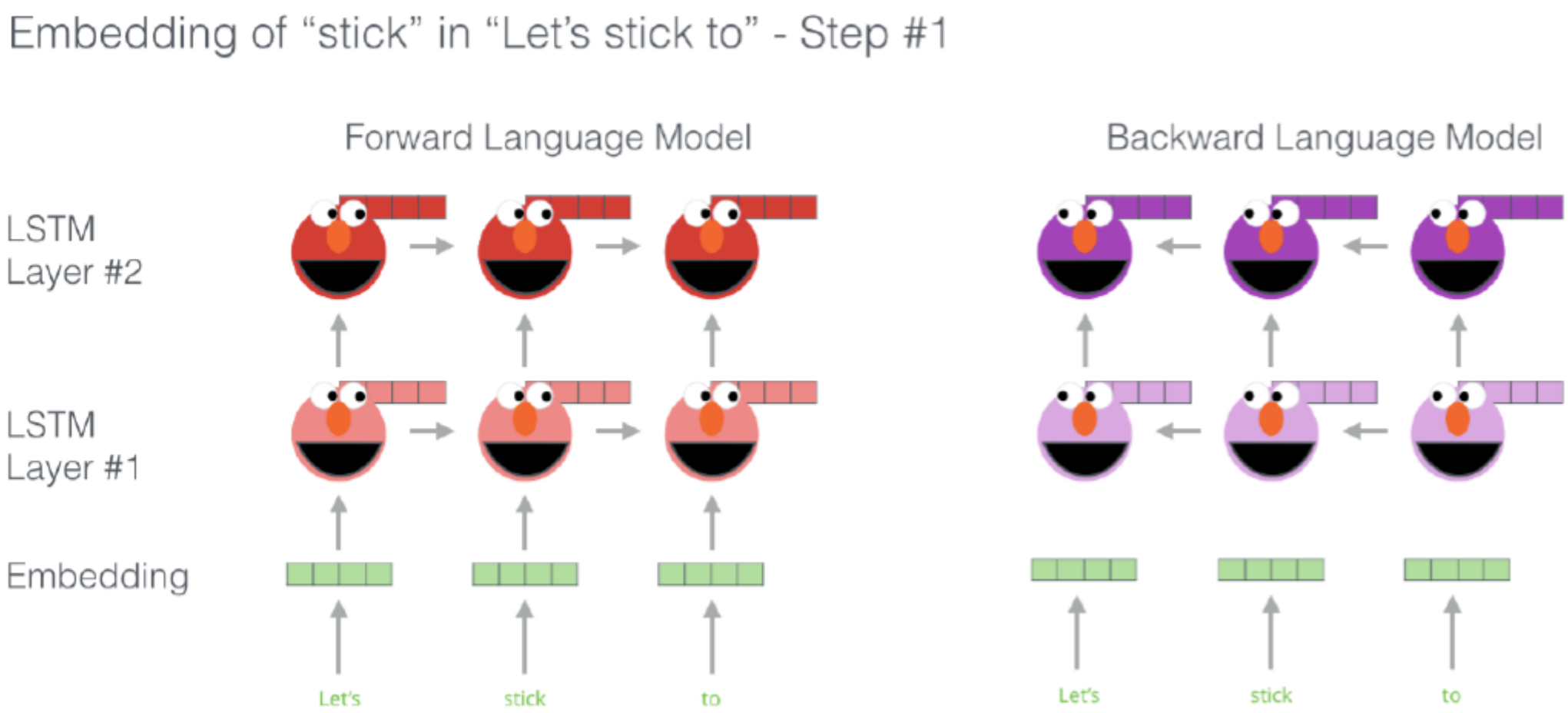
Matthew Peters, Mark Neumann, Mohit Iyyer, MattGardner, Christopher Clark, Kenton Lee, and LukeZettlemoyer. 2018. Deep contextualized word representations. In NAACL.

ELMo: Embeddings from Language Models

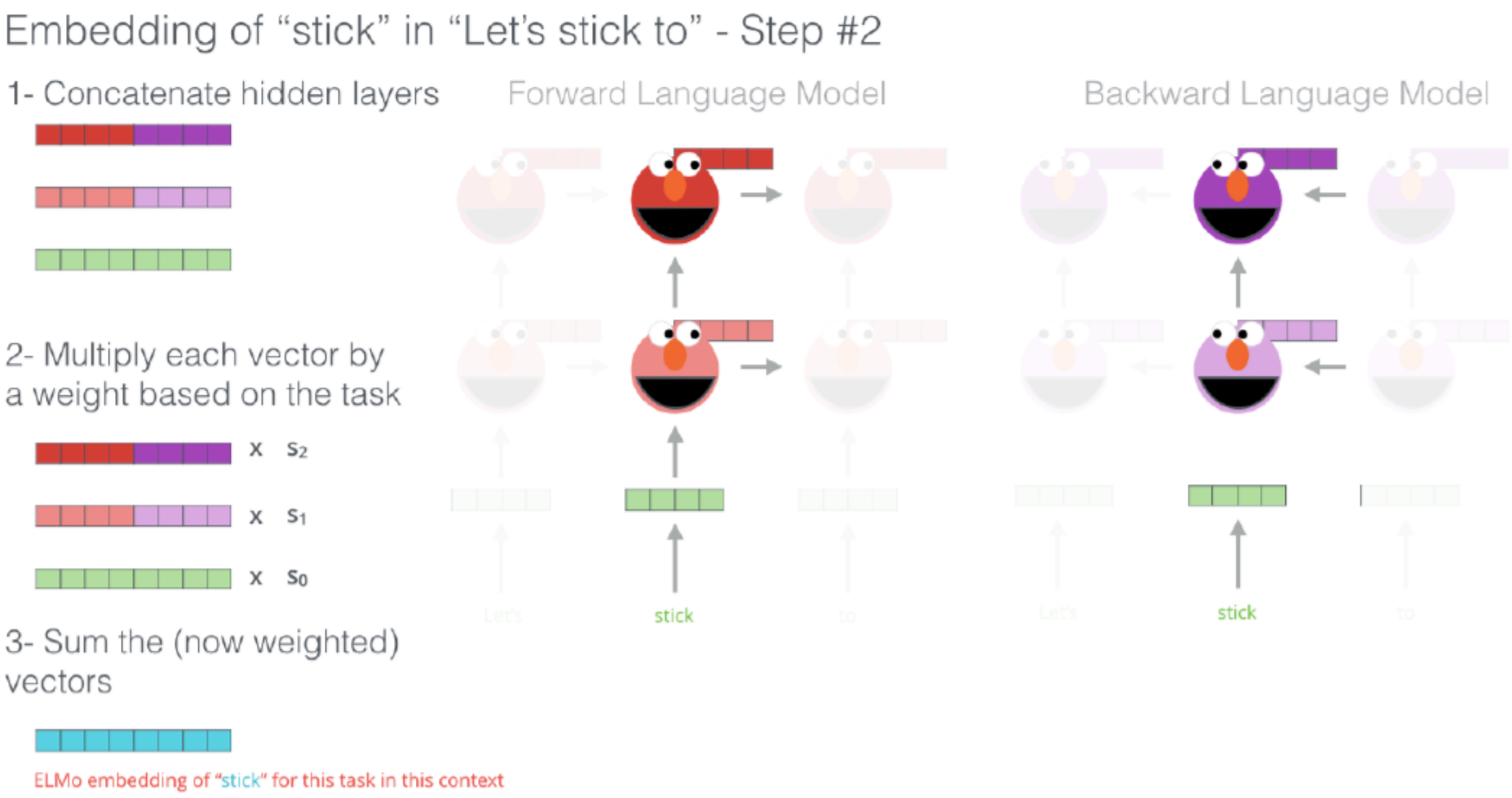
- A bidirectional language model
  - extract context sensitive features from a language model
- The contextualized representation is the concatenation of the output of the forward and backward LSTM
  - concatenation of independently trained left-to-right(LTR) and right-to-left(RTL) LSTM to generate features for downstream tasks



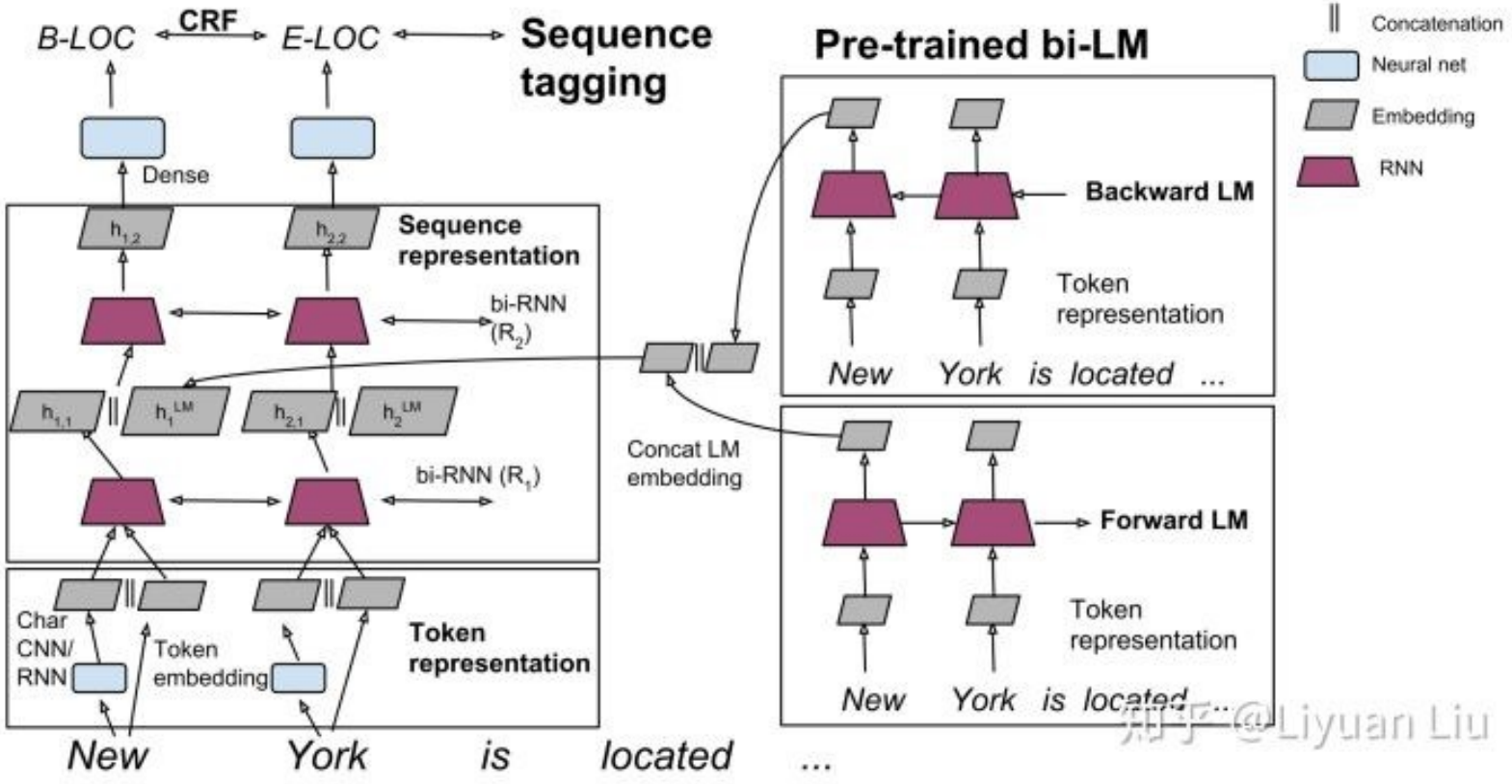
ELMo: Embeddings from Language Models



ELMo: Embeddings from Language Models

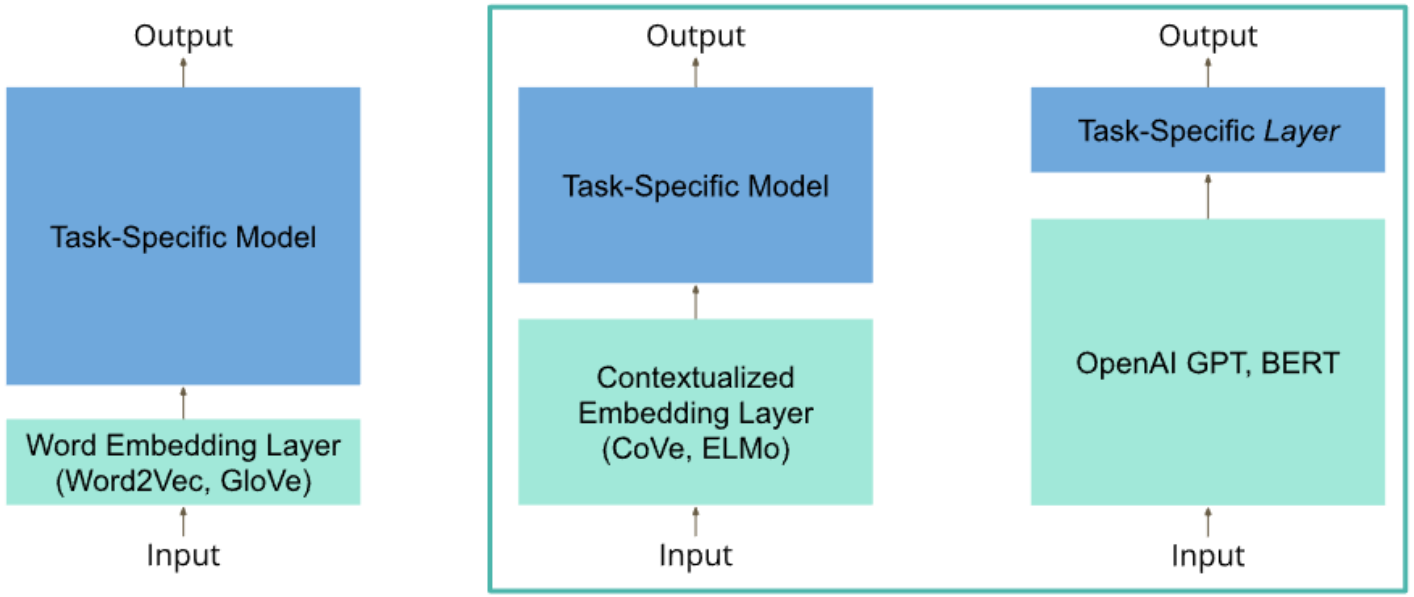


ELMo: Embeddings from Language Models



Strategies for applying pre-trained language representations

- feature-based approach
  - ELMo (Peters et al., 2018)
  - uses tasks-specific architectures that include the pre-trained representations as additional features
- fine-tuning approach
  - Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018)
  - minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning the pretrained parameters



BERT Contribution

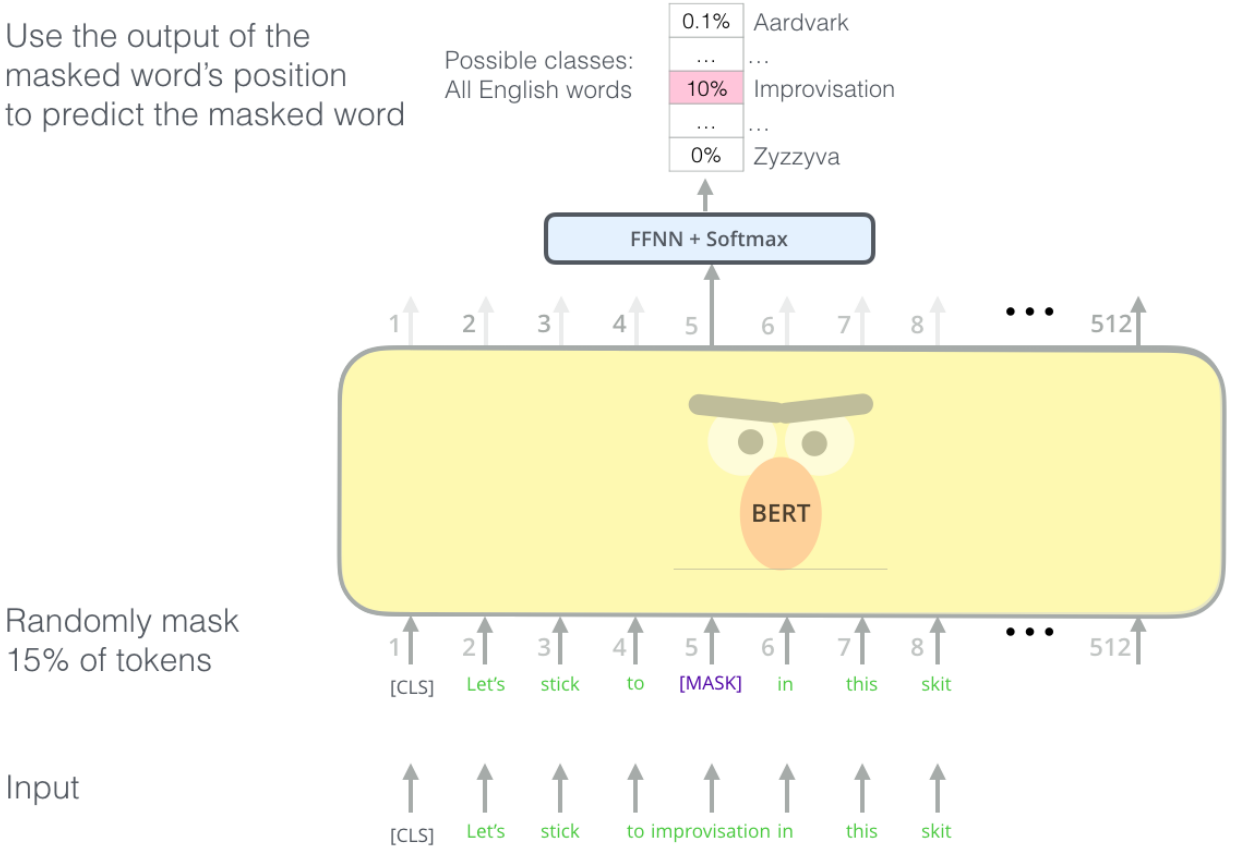
- BERT is a multi-layer bidirectional Transformer encoder
- It demonstrate the importance of bidirectional pre-training for language representations
  - masked language models(MLM) to enable pre-trained deep bidirectional representations
- It is the first fine-tuning based representation model that achieves state-of-the-art performance on a large suite of sentence-level and token-level tasks
  - outperforming many systems with task-specific architectures

Pre-training Tasks

MLM & NSP

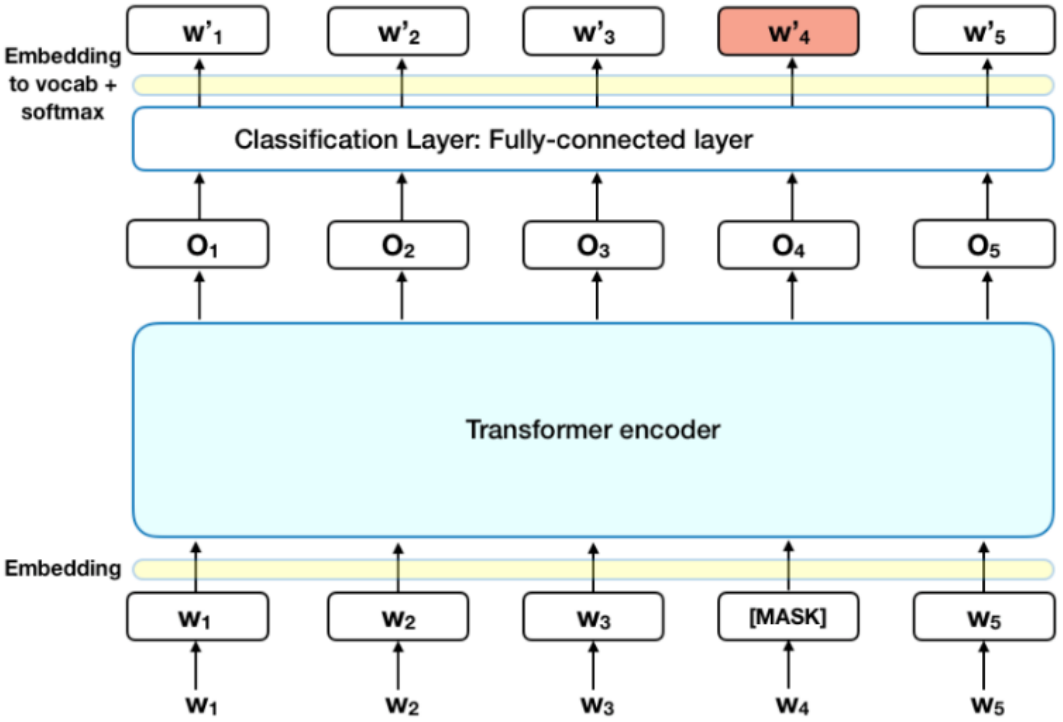
Task#1 – Masked Language Model (MLM)

- masking some percentage of the input tokens at random, and then predicting only those masked tokens
- 最後mask tokens的hidden vectors會輸出至vocabulary size的softmax去預測



Task#1 – Masked Language Model (MLM)

- Randomly masks some of the tokens from the input
  - Randomly mask 15% of tokens
  - < 15%: too expensive to train
  - > 15%: not enough context
- 依據上下文(context)來預測原本的字彙應該要是什麼
  - 融合左邊跟右邊的context



Task#1 – Masked Language Model (MLM)

- Motivation: a mismatch between pre-training and finetuning,
  - since the [MASK] token is never seen during fine-tuning
- Approach: do not always replace “masked” words with the actual [MASK] token
- 80%的時間會用[MASK] token來取代=>my dog is hairy → my dog is [MASK]
- 10%的時間會隨機取代成別的字=>my dog is hairy → my dog is apple
  - 因為隨機的替換成別的字詞只有1.5%的機率(15%的字詞*10%的隨機)，不會傷害模型對語言的理解能力
- 10%的時間會保留原本的字詞=>my dog is hairy → my dog is hairy
  - 為了讓所學到的表示法和實際觀察到的字詞不會有太大的gap

Task#2 – Next Sentence Prediction (NSP)

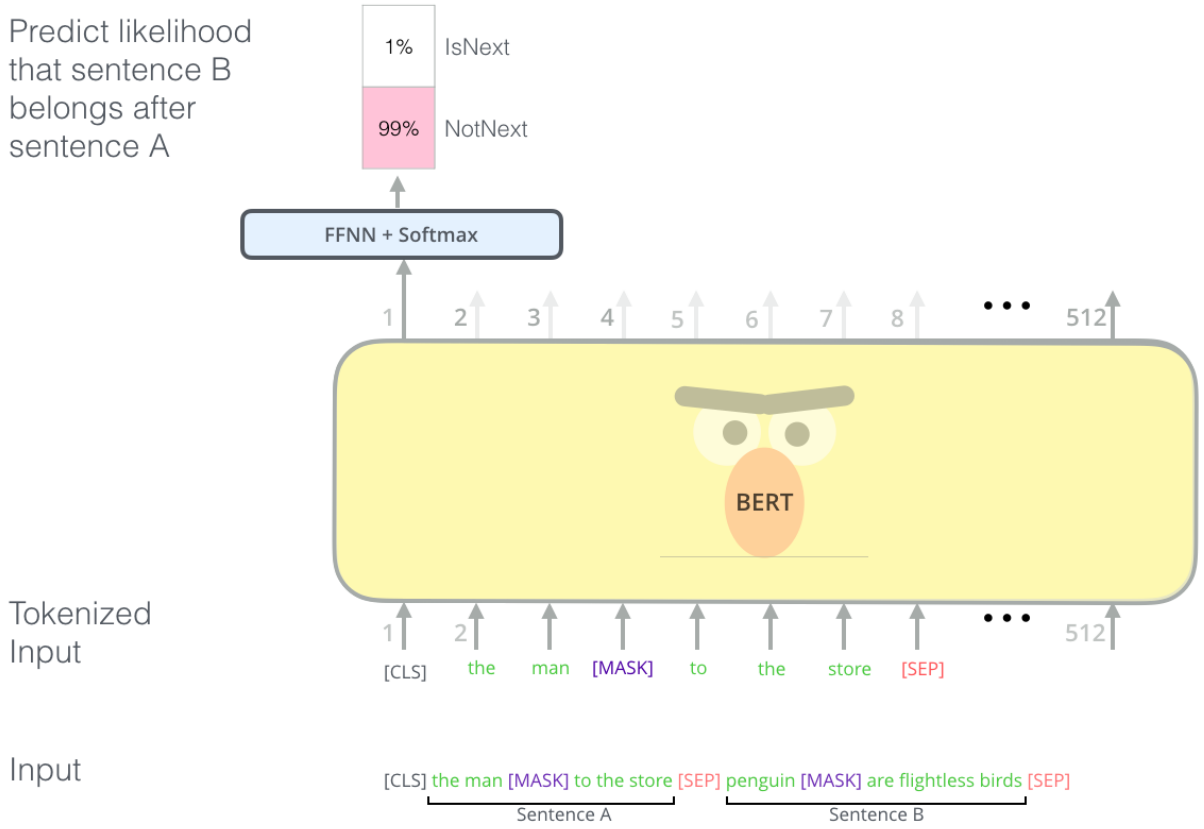
- To model relationships between sentences
- 很多downstream task例如Question Answering (QA)、Natural Language Inference (NLI)是基於理解兩句話的關係的(inter-sentence relationship)，因此希望可以把相鄰上下兩句話的關係捕捉起來

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

Input = [CLS] the man [MASK] to the store [SEP]
penguin [MASK] are flight ##less birds [SEP]
Label = NotNext

Task#2 – Next Sentence Prediction (NSP)

- jointly pre-trains text-pair representations
- 有50%的時間兩句話會是下一句關係，有50%時間會隨機挑另外一句話當成下一句

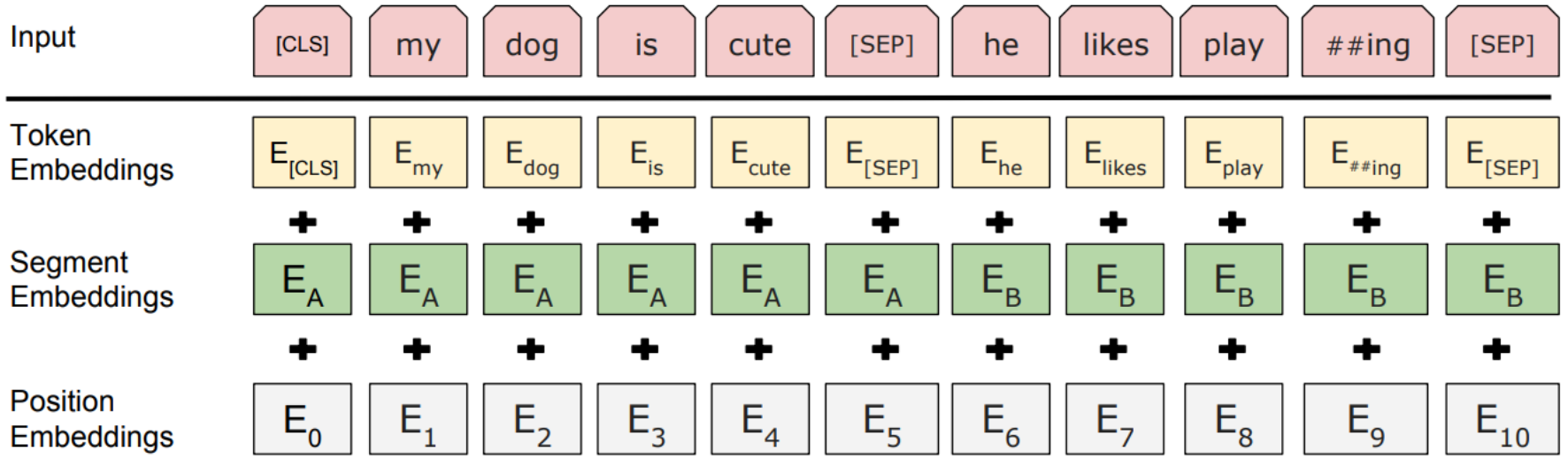


Input Representation

Embeddings

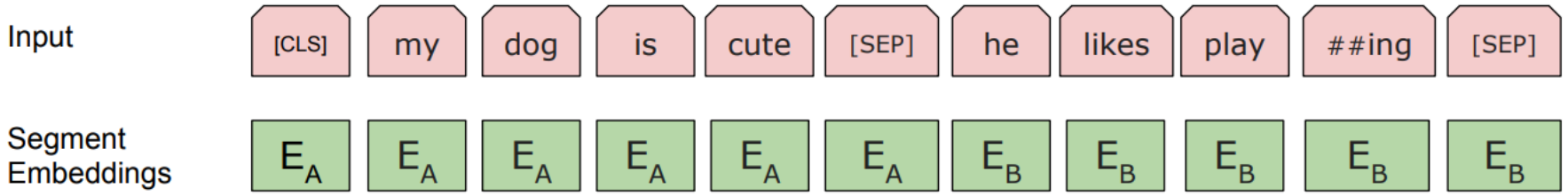
Input Representation

- Input embeddings contain :
  - Word-level token embeddings
  - Sentence-level segment embeddings
  - Position embeddings



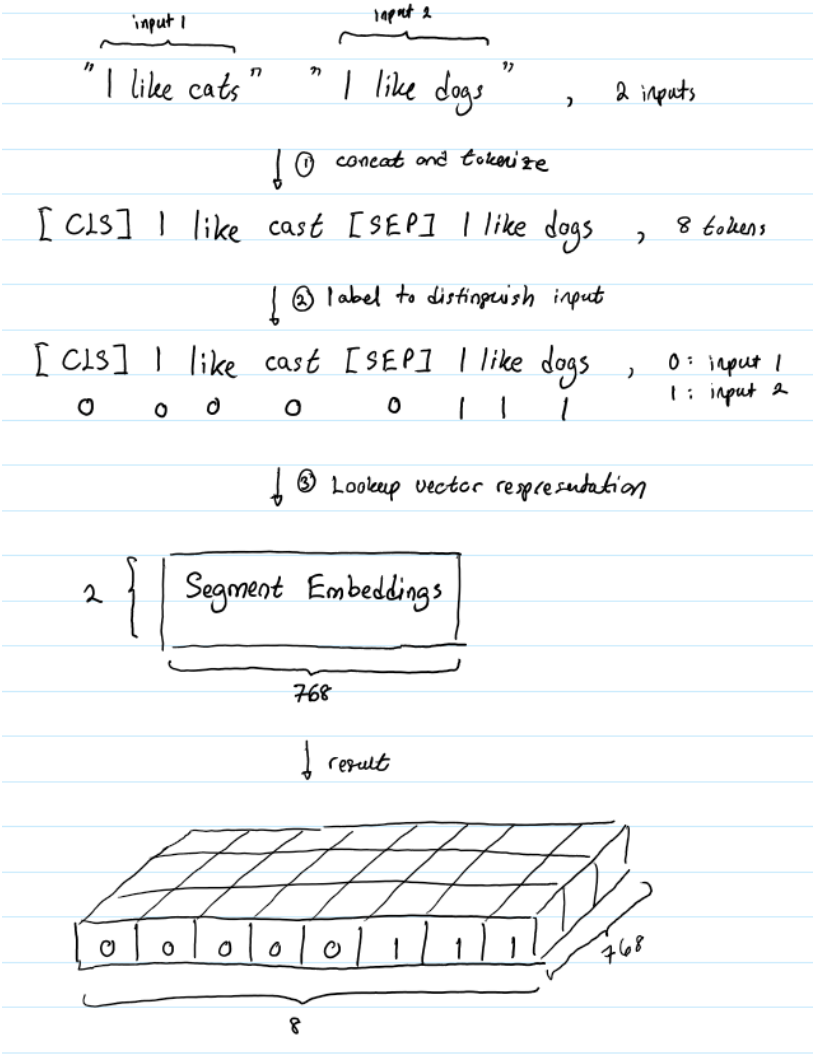
Input Representation

- First token of every sequence is always the special classification embedding ([CLS])
  - output of Transformer corresponding to this token is used as the aggregate sequence representation for classification tasks
  - for nonclassification tasks, this vector is ignored
- Differentiate the sentences with a special token ([SEP])
  - learned sentence A embedding to every token of the first sentence and a sentence B embedding to every token of the second sentence
  - for single-sentence inputs we only use the sentence A embeddings



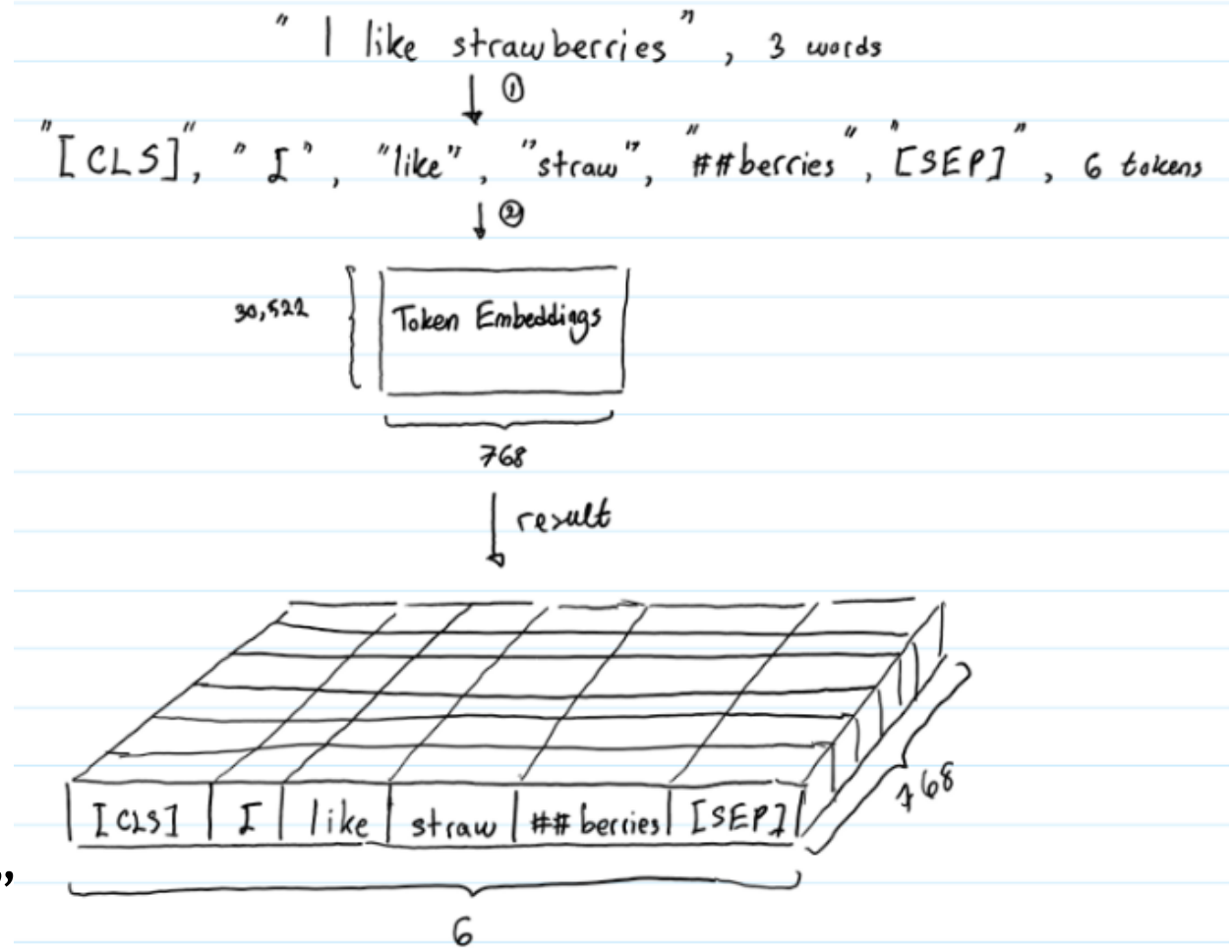
Segment Embedding

- To distinguishes the inputs in a given pair
- The Segment Embeddings layer only has 2 vector representations.
  - The first vector (index 0) is assigned to all tokens that belong to input 1
  - the last vector (index 1) is assigned to all tokens that belong to input 2



WordPiece Embedding

- to transform words into vector representations of fixed dimension (768-dimensional vector)
  - tokenization method that aims to achieve a balance between vocabulary size and out-of-vocab words
- Tokenized and add extra tokens [CLS] at the start and [SEP] at end
  - to separate a pair of input texts respectively
- enables BERT to only store 30,000 “words” in its vocabulary
  - EX: strawberries = “straw” + “##berries”

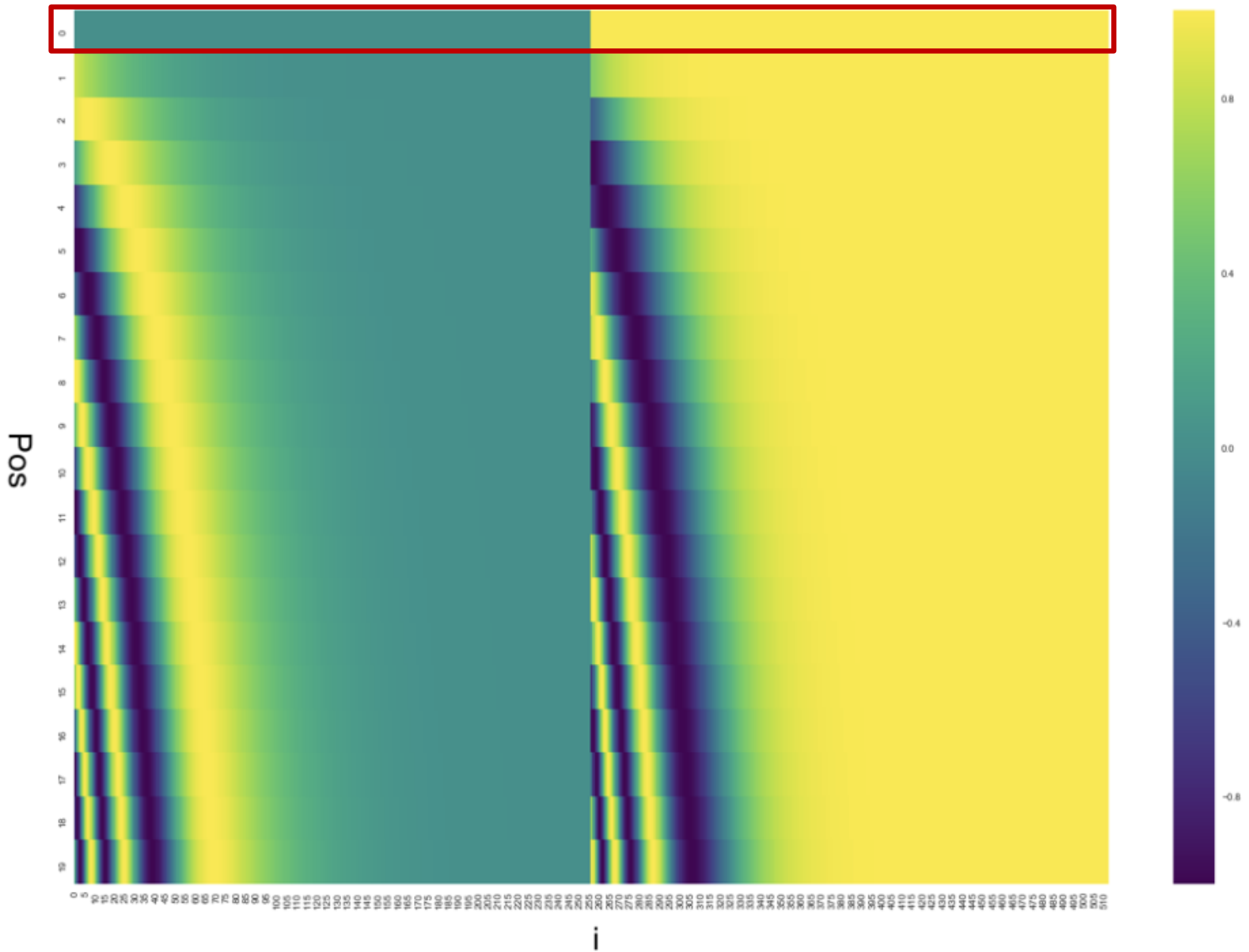


Positional Embeddings

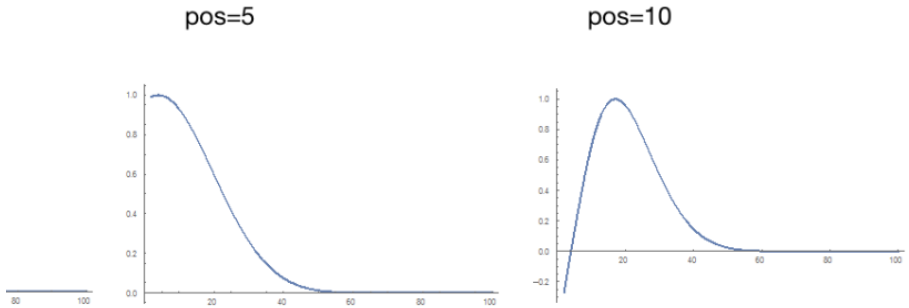
- To understand that given an input text
  - EX: *I think, therefore I am*
  - the first “I” should not have the same vector representation as the second “I”
- The authors incorporated the sequential nature of the input sequences by having BERT learn a vector representation for each position
- Input like “Hello world” and “Hi there”, both “Hello” and “Hi” will have identical position embeddings since they are the first word in the input sequence. Similarly, both “world” and “there” will have the same position embedding.

Positional Encoding Visualization

$d_{model} = 512,\ pos = [0,\ 20),\ i = [0,\ d_{model}) :$



First row is the vector representation of any word in the first position, the second row is the vector representation of any word in the second position, etc.



問題

Multi-head不能捕捉序列順序，
如果打亂句子的詞序，
Attention結果一樣

解法 Positional Encoding

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

$if\ shape\ of\ enc, dec\ = [T, d_{model}]$ pos: 詞的位置

$\rightarrow then\ pos \in [0, T), i \in [0, d_{model})$ i: d_{model} 的第i個元素

優點： 位置相隔的詞，其位置關係是線性關係
→ 位置 i+k 可表示為 i 向量的線性變換

$$sin(\alpha+\beta) = sin\alpha cos\beta + cos\alpha sin\beta$$

$$cos(\alpha+\beta) = cos\alpha cos\beta - sin\alpha sin\beta$$

∵ 三角函數週期性

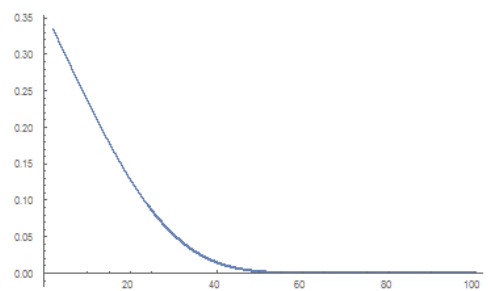
∴ **學到絕對、相對位置關係**

最後 Word embedding + positional encoding

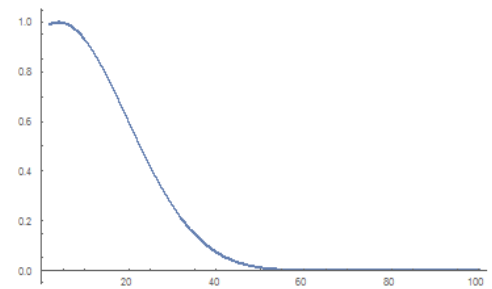
(sum、concat皆可)

$if\ d_{model} = 512:$

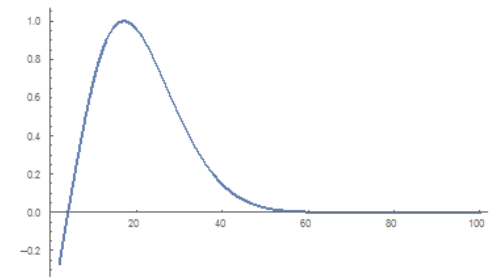
pos=1



pos=5



pos=10

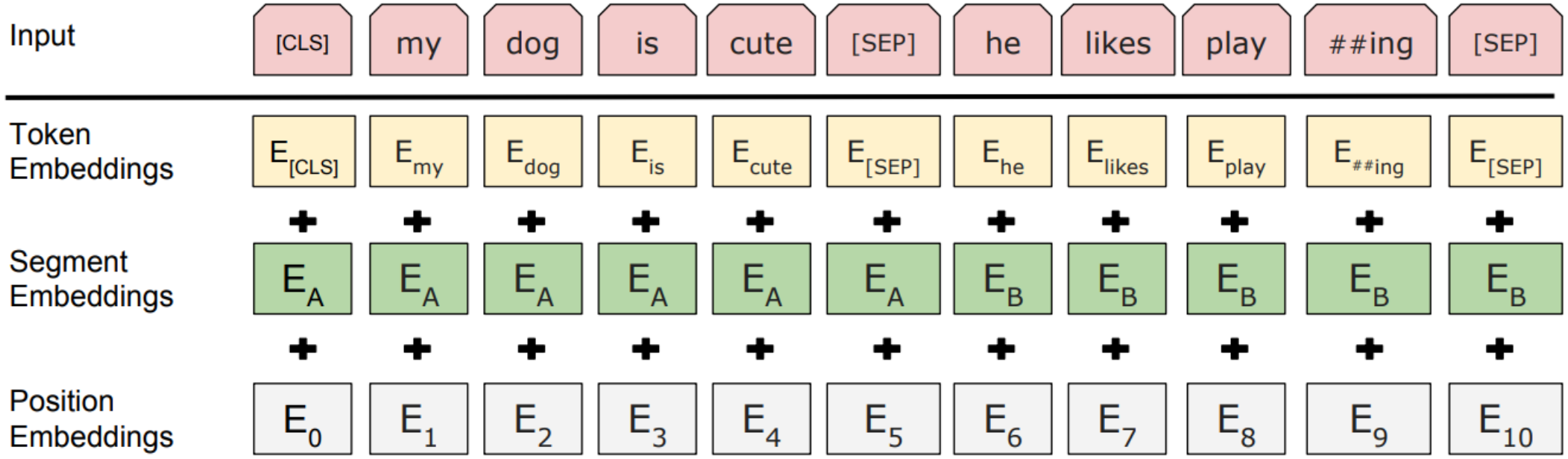


⟶ **三張圖很像進行平移縮放**

Combining Representations

- input sequence of length = n
- Token Embeddings with shape (1, n, 768) which are just vector representations of words
- Segment Embeddings with shape (1, n, 768) which are vector representations to help BERT distinguish between paired input sequences
- Position Embeddings with shape (1, n, 768) to let BERT know that the inputs its being fed with have a temporal property
- summed element-wise to produce a single representation with shape (1, n, 768)
  - input representation that is passed to BERT's Encoder layer

Combining Representations



- Summed element-wise to produce a single representation
  - input representation that is passed to BERT's Encoder layer

Pre-Training BERT

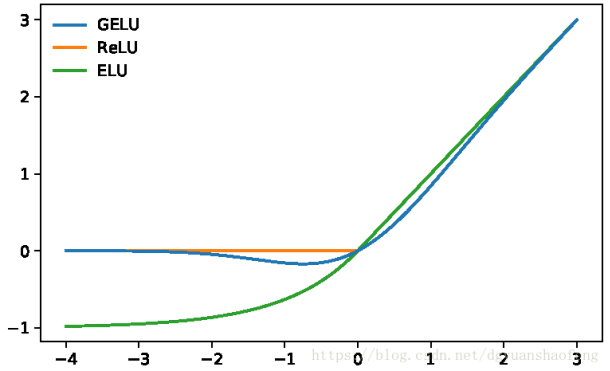
- Dataset: BooksCorpus & English Wikipedia

Dataset

- BooksCorpus
  - 未公開的dataset (含有800M words)
- Wikipedia
  - Google從English Wikipedia 所爬的 (含有2500M words)
  - extract only the text passages and ignore lists, tables, and headers

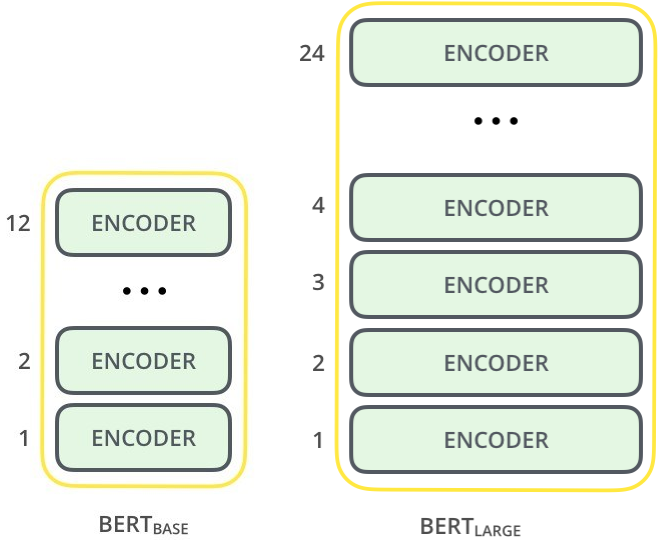
Pre-Training BERT

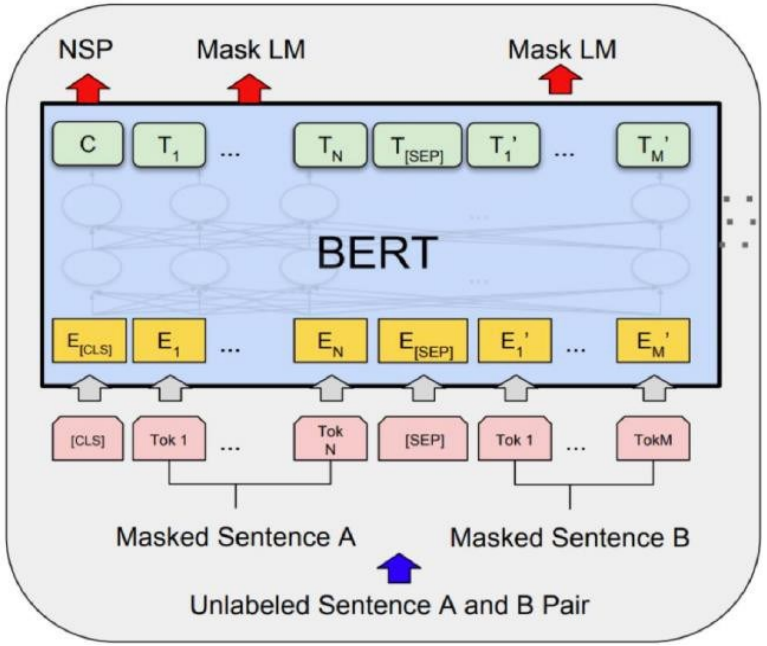
- 1. 從corpus當中取兩個句子
  - 第一個句子會得到A embedding，第二個句子會得到B embedding
- 2. 50%的時間B確實接在A後面，另外50%時間隨機選一個句子接在A後面 (next sentence prediction)
  - combined length is ≤ 512 tokens
- 3. WordPiece tokenization with a uniform masking rate of 15% (LM masking)
  - no special consideration given to partial word pieces
- Training Hyperparameters
  - batch size = 256 (sequences) (*512 tokens=12800tokens/batch)
  - 約40 epochs
  - Optmizer = Adam
  - Dropout = 0.1 over all layers
  - loss = sum of the mean masked LM likelihood (字典categorical cross-entropy) + mean next sentence prediction likelihood (binary cross-entropy)
  - activation function = gelu
  - corpus: 3.3 billion words
  - Bert-base: 16*TPU 4-days (55*1080TI) (40*2080TI)
  - Bert-Large: 64*TPU 4-days (215*1080TI) (160*2080TI)



Pre-Training BERT

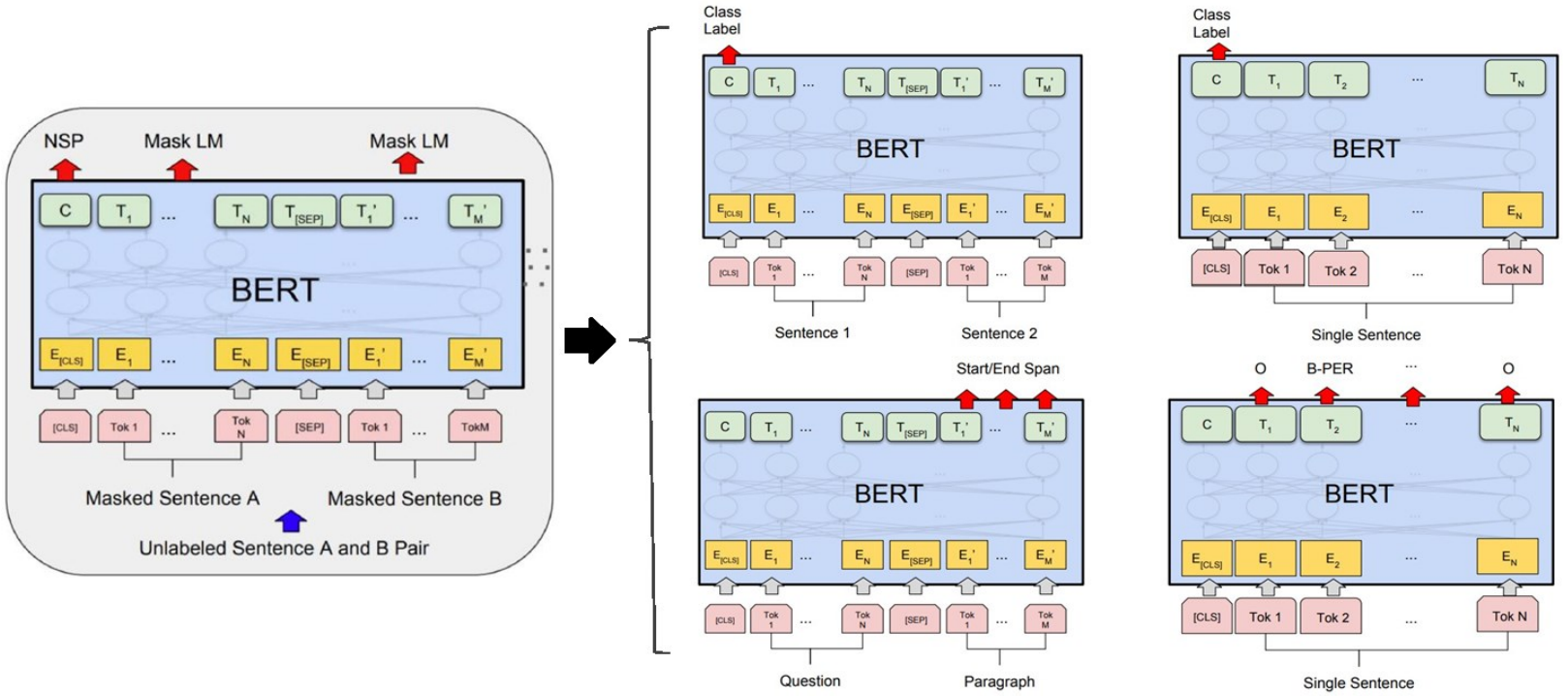
- Training data: Wikipedia + BookCorpus
- 2 BERT models
  - BERT-Base: 12-layer, 768-hidden
  - BERT-Large: 24-layer, 1024-hidden





Fine-Tune BERT

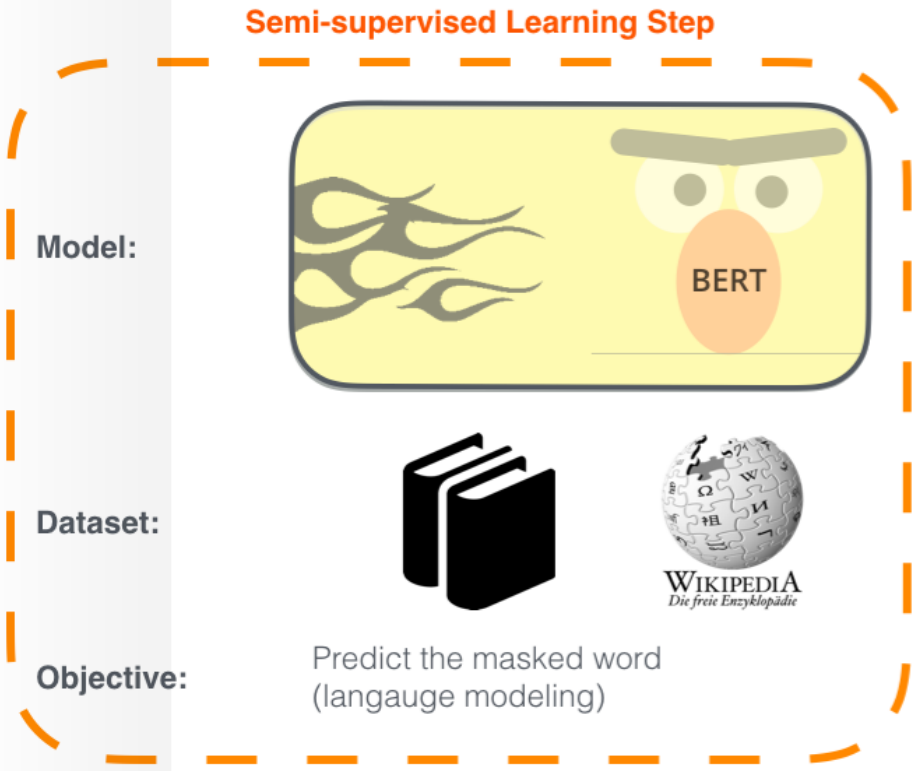
- Idea: simply learn a classifier/tagger built on the top layer for each target task (BERT with one additional output layer)
  - minimal number of parameters need to be learned from scratch
  - sequence-level tasks / token-level tasks



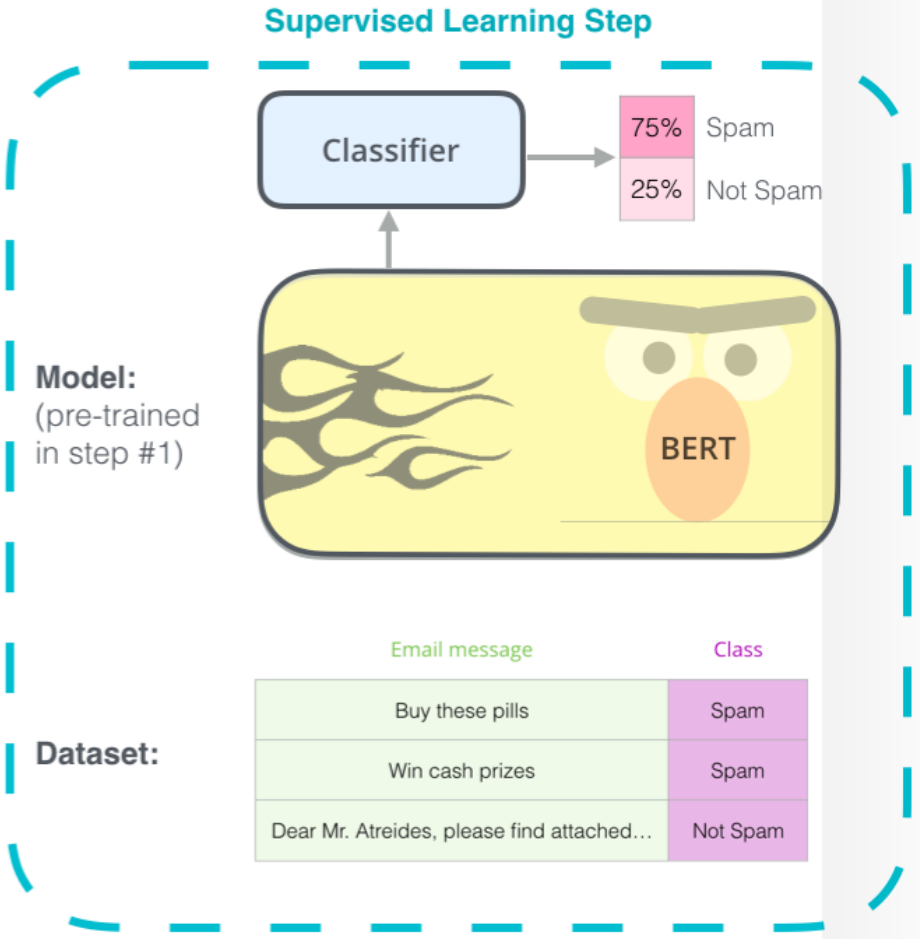
BERT Pre-Train & Fine-Tune Overview

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

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



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



Experiments

GLUE

GLUE Dataset

- General Language Understanding Evaluation Benchmark
- Natural language understanding(NLU) task
- 嚴格地切分Train, Valid, Test dataset，並建立Test set伺服器(一定要來這邊上傳)，防止有人刻意過度overfitting

MNLI

- Multi-Genre Natural Language Inference
- Given a pair of sentences, the goal is to predict whether the second sentence is an entailment(支持), contradiction(反對), or neutral(沒關係) with respect to the first one

Premise	Label	Hypothesis
Fiction		
The Old One always comforted Ca'daan, except today.	*neutral*	Ca'daan knew the Old One very well.
Letters		
Your gift is appreciated by each and every student who will benefit from your generosity.	*neutral*	Hundreds of students will benefit from your generosity.
Telephone Speech		
yes now you know if if everybody like in August when everybody's on vacation or something we can dress a little more casual or	*contradiction*	August is a black out month for vacations in the company.
9/11 Report		
At the other end of Pennsylvania Avenue, people began to line up for a White House tour.	*entailment*	People formed a line at the end of Pennsylvania Avenue.

QNLI

- Question Natural Language Inference (Stanford Question Answering Dataset)
- binary classification task
  - positive examples: (question, sentence) pairs which do contain the correct answer
  - negative examples: (question, sentence) from the same paragraph which do not contain the answer

> *"What would a teacher do for someone who is cocky?"*
>
> *"The function of the teacher is to pressure the lazy, inspire the bored, deflate the cocky, encourage the timid, detect and correct individual flaws, and broaden the viewpoint of all."*

> *"How many people were lost in Algiers during 1620-21?"*
>
> *"Plague was present in at least one location in the Islamic world virtually every year between 1500 and 1850."*

MRPC

- Microsoft Research Paraphrase Corpus
- Sentence pairs automatically extracted from online news sources (with human annotations)
- To predict the sentences in the pair are semantically equivalent

> *"The decision to issue new guidance has been prompted by intelligence passed to Britain by the FBI in a secret briefing in late July ."*
>
> *"Scotland Yard 's decision to issue new guidance has been prompted by new intelligence passed to Britain by the FBI in late July ."*

> *"The company 's operating loss rose 59 percent to $ 73 million , from $ 46 million a year earlier ."*
>
> *"Operating revenue fell 4.5 percent to $ 2.3 billion from a year earlier ."*

Other GLUE tasks

- QQP (Quora Question Pairs)
  - binary classification task
  - 判斷兩個問句是否在語意上是相等的
- SST-2 (Stanford Sentiment Treebank)
  - binary single-sentence classification task
  - sentences extracted from movie reviews with human annotations of their sentiment
- CoLA (Corpus of Linguistic Acceptability)
  - a binary single-sentence classification task
  - predict whether an English sentence is linguistically “acceptable” or not

Other GLUE tasks

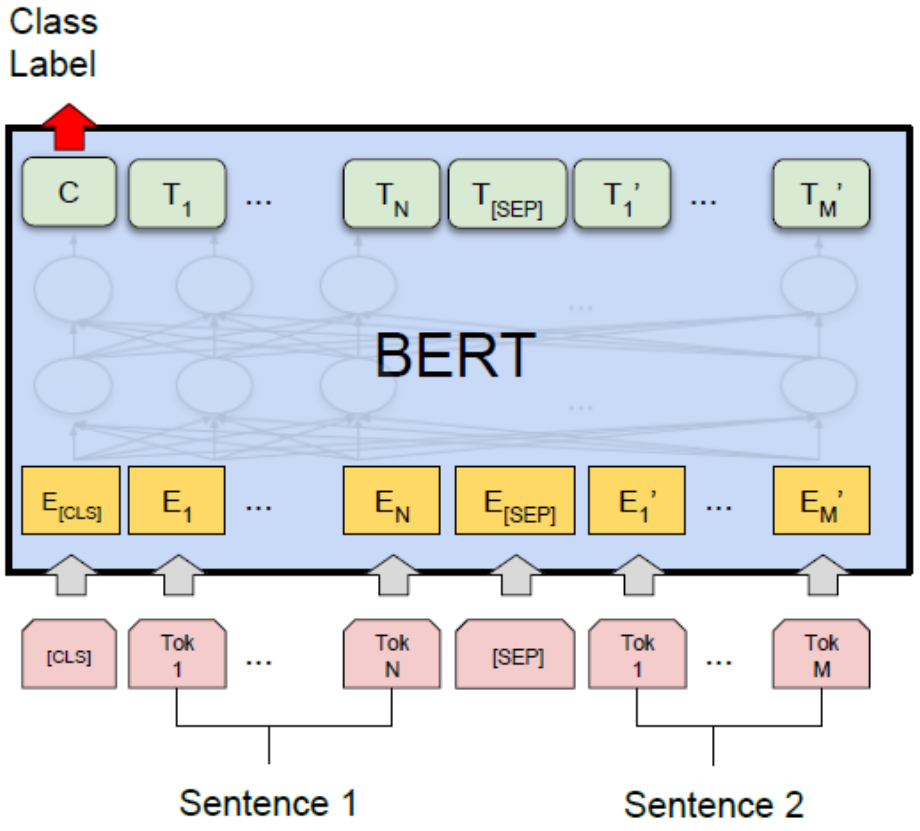
- STS-B (Semantic Textual Similarity Benchmark)
  - sentence pairs drawn from news headlines and other sources
  - annotated with a score from 1 to 5 denoting how similar the two sentences are in terms of semantic meaning
- RTE (Recognizing Textual Entailment)
  - binary entailment task (類似MNLI但是training data少很多)
- WNLI (Winograd NLI)
  - small natural language inference dataset

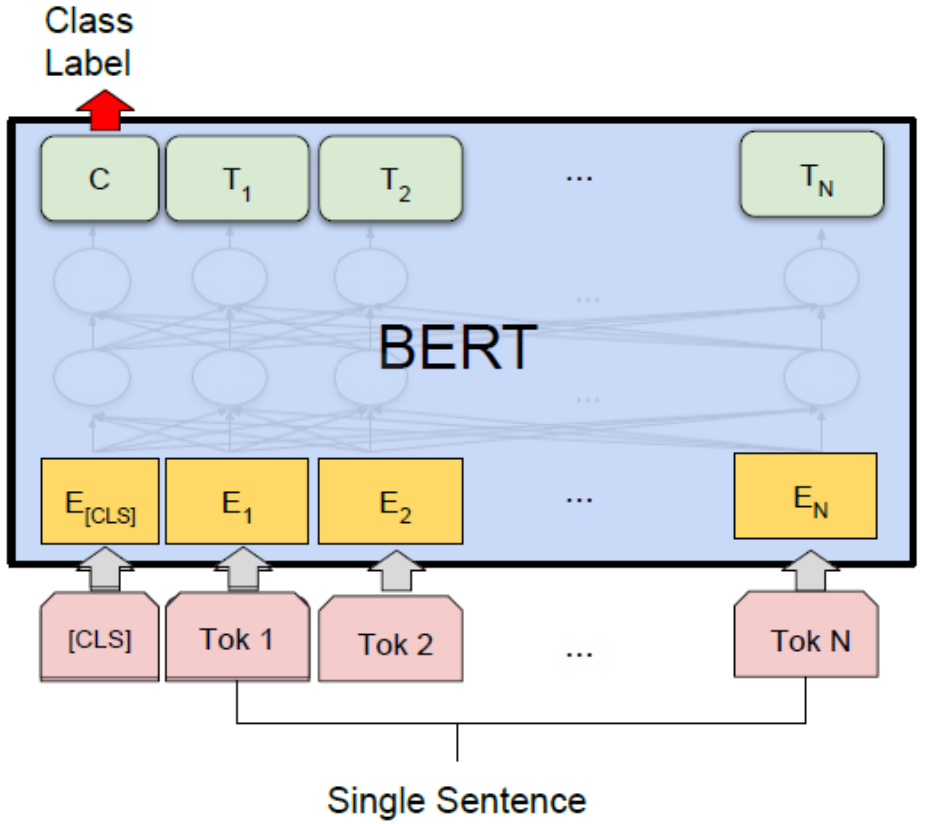
"I put the cake away in the refrigerator. It has a lot of butter in it."

"The cake has a lot of butter in it."

Fine-Tune BERT in sequence-level

- Take the final hidden state (i.e., the output of the Transformer [CLS]) 作為aggregate representation





GLUE Results

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**

- BERT_BASE跟BERT_LARGE都比現有的所有系統表現還好上不少
  - 相較於當時的state-of-the-art(SOTA)有分別4.4%跟6.7%的improvement
  - BERT_LARGE比BERT_BASE在各項表現都還要好，即便是比較小的dataset仍是如此

Experiments

SQuAD

SQuAD v1.1

- Standford Question Answering Dataset
  - crowdsourced question/answer pair
- Given a question and a paragraph from Wikipedia containing the answer
  - predict the answer text span in the paragraph

- Input Question:

  Where do water droplets collide with ice crystals to form precipitation?

- Input Paragraph:

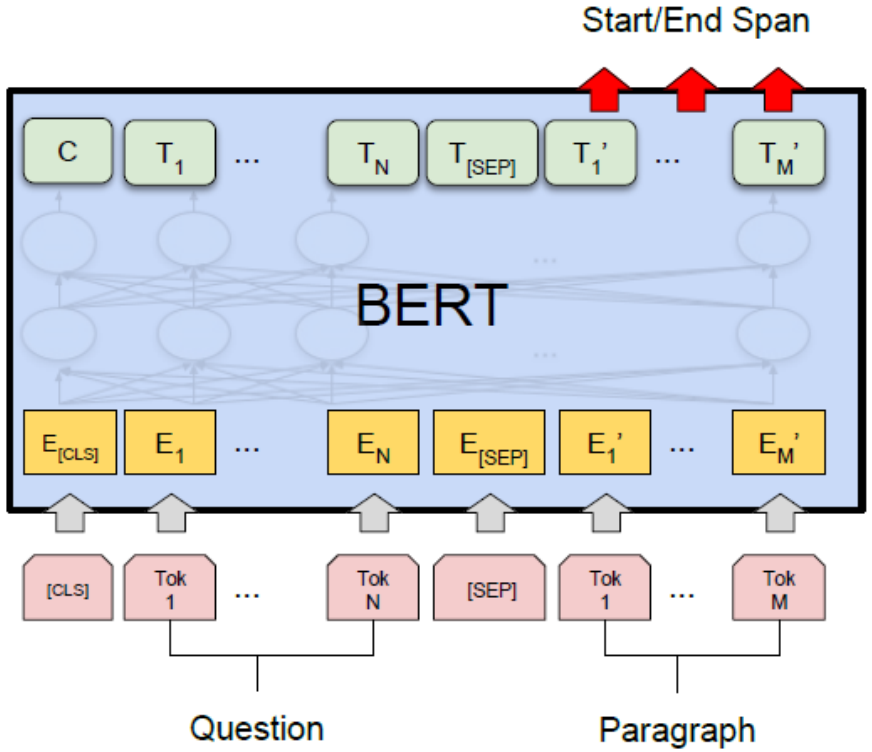
  ... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. ...

- Output Answer:

  within a cloud

Fine-Tune BERT for SQuAD

- Represent the input question and paragraph as a packed sequence
  - question using the A embedding and the paragraph using the B embedding
- 輸出paragraph的哪一個token是start span，哪一個token是end span
  - 因為兩者是獨立判斷的，因此限制end token要在start token後面



SQuAD Results

- EM = Exact Match
  - measures the percentage of predictions that match any one of the ground truth answers exactly
- F1 score
  - measures the average overlap between the prediction and ground truth answer

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

Experiments

NER

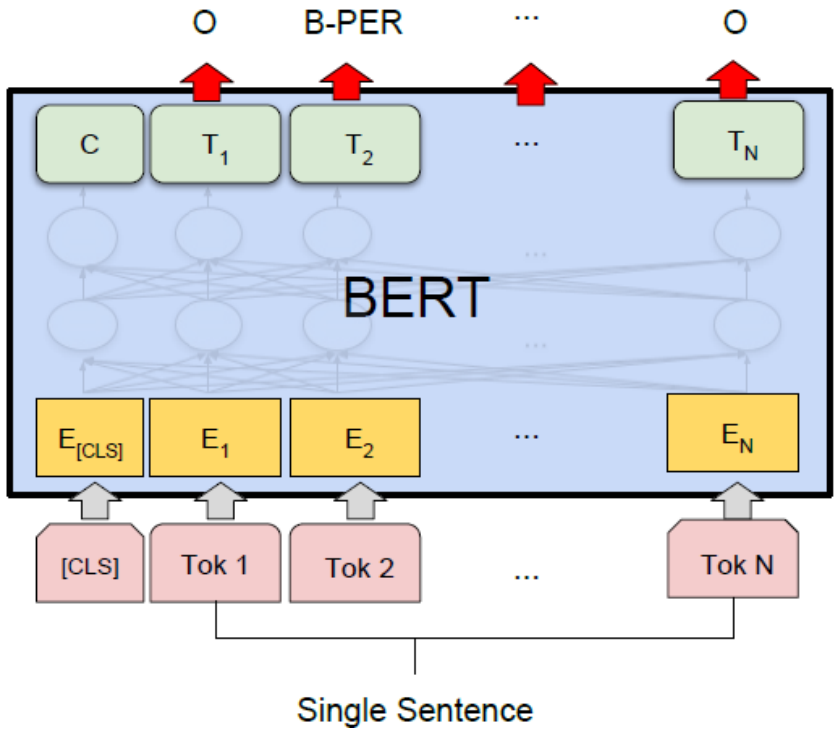
CoNLL 2003 NER

- Named Entity Recognition (NER) dataset
- Label = Person, Organization, Location, Miscellaneous(混合), or Other (non-named entity)
- 為了可以跟WordPiece tokenizer相容，利用的是第一個sub-token來進行分類
  - 其他##開頭的subtoken不進行predict
  - 採用的是cased model (區分大小寫，其他task都是uncased)

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

Fine-Tune BERT for NER

- 對每一個token的final hidden representation (vector)進行NER label 分類
- 不考慮周圍的字詞關係，每個token獨立判斷



NER Results

Model	Description	CONLL 2003 F1
TagLM (Peters+, 2017)	LSTM BiLM in BLSTM Tagger	91.93
ELMo (Peters+, 2018)	ELMo in BLSTM	92.22
BERT-Base (Devlin+, 2019)	Transformer bidi LM + fine tune	92.4
CVT Clark (Clark et al., 2018)	Cross-view training + multitask learn	92.61
BERT-Large (Devlin+, 2019)	Transformer bidi LM + fine tune	92.8

Experiments

SWAG

SWAG

- Situations With Adversarial Generations
- 113k sentence-pair completion examples
  - evaluate grounded commonsense inference
- 第一句話當成 sentence A，把可能的第二句話當成 sentence B，最後會得到一個 aggregate representation
  - 因為有四個選項所以會有四個final vector，這四個 C最後丟進去clf softmax
  - fine-tune 3 epochs lr=2e-5 batch_size=16

Ablation Studies

Pre-Training Tasks

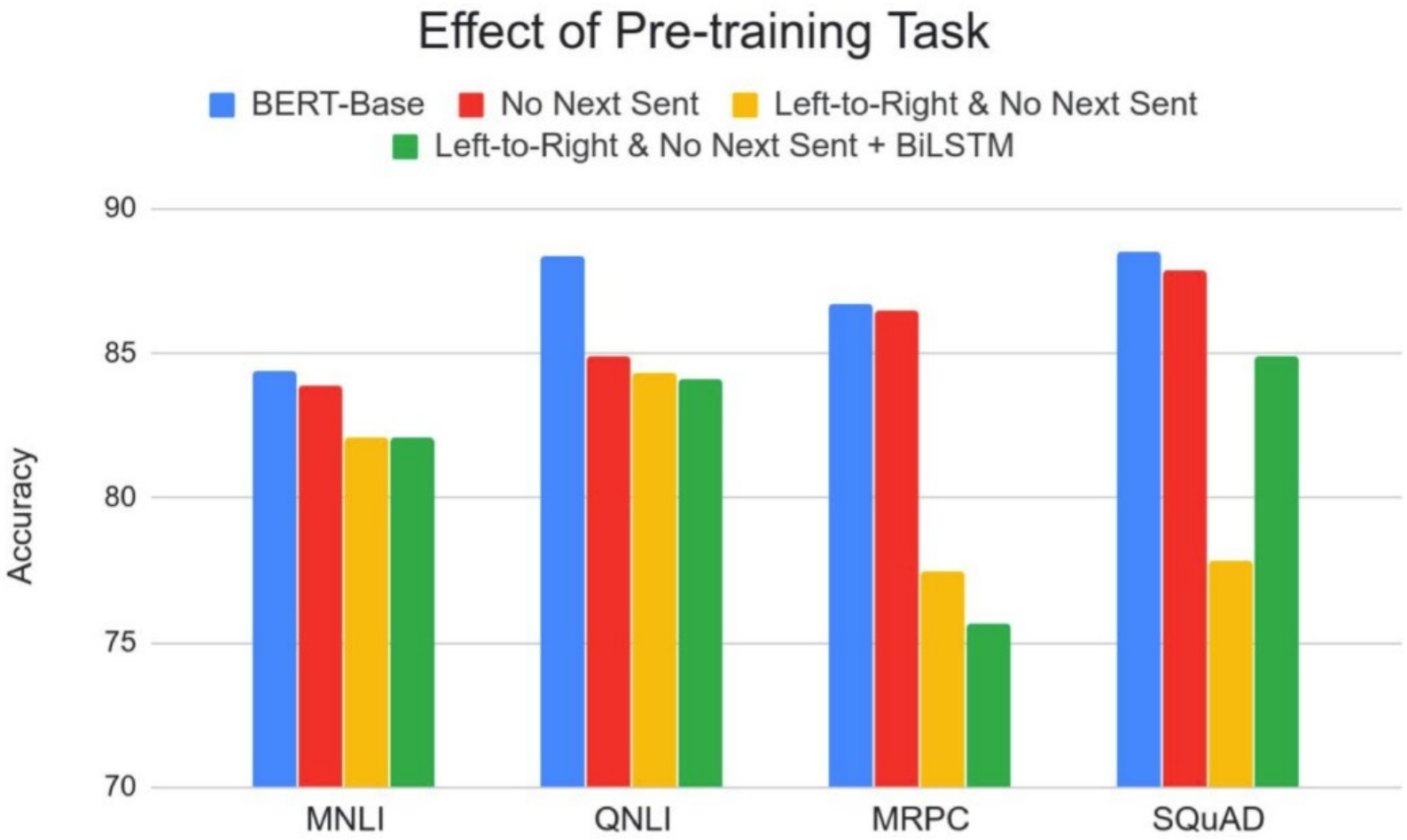
Only MLM, w/o NSP

- 僅使用MLM(Masked Language Model)不使用NSP(Next Sentence Prediction)
  - 對QNLI、MNLI、SQuAD的performance有較大的傷害
  - 作者認為是因為少了前一句話的context經驗幫忙而影響

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

Only LTR, w/o NSP & MLM

- 仿造Op
  用BERT
  - 各項ta
  - MRPC
- 為了證明
  大的進步



並且改

得了很

Ablation Studies

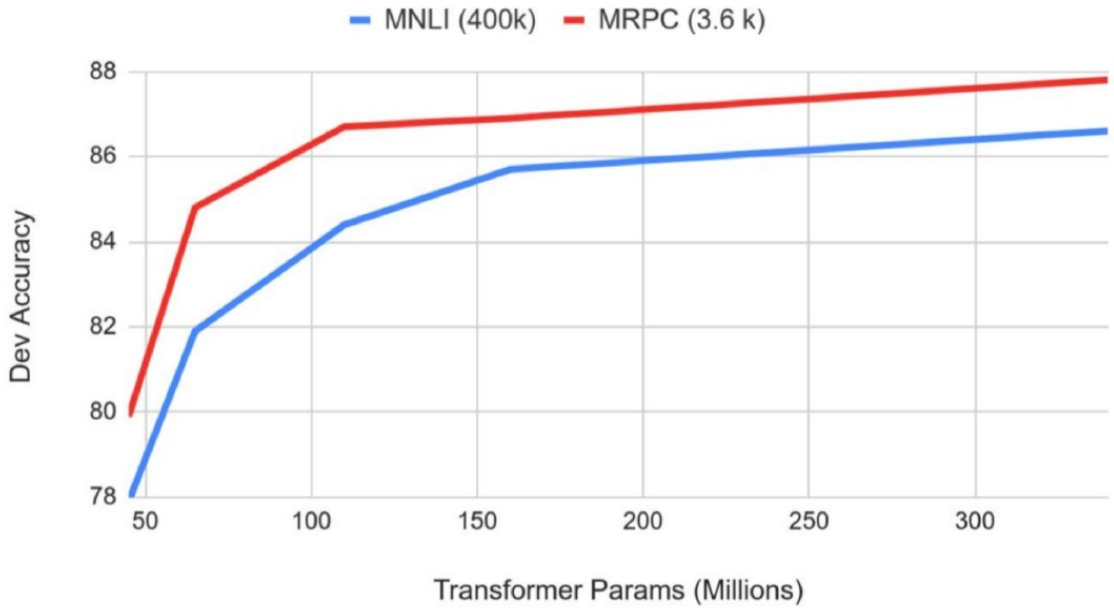
Model Size

Effect of BERT Model Size

- 探討模型大小對fine-tuning task的accuracy影響
  - 利用相同的超參數跟訓練過程
  - 以GLUE tasks來做實驗
- 考量不同的layers(3層、6層、12層)、hidden size(768、1024)
  - model越大越複雜，模型參數量越多

BERT Results with Different Model Sizes

- larger model有越好的accuracy improvement
- 即便是對MRPC的小dataset(僅有3600筆labeled data)，依然是越大的model表現越好，沒有overfitting的現象
  - 作者認為自己是第一篇ML model發表說大model size但是只有小dataset卻可以依然保有很卓越進步的 (只要該model 有足夠的pre-trained即可)



Feature-based Approach with BERT

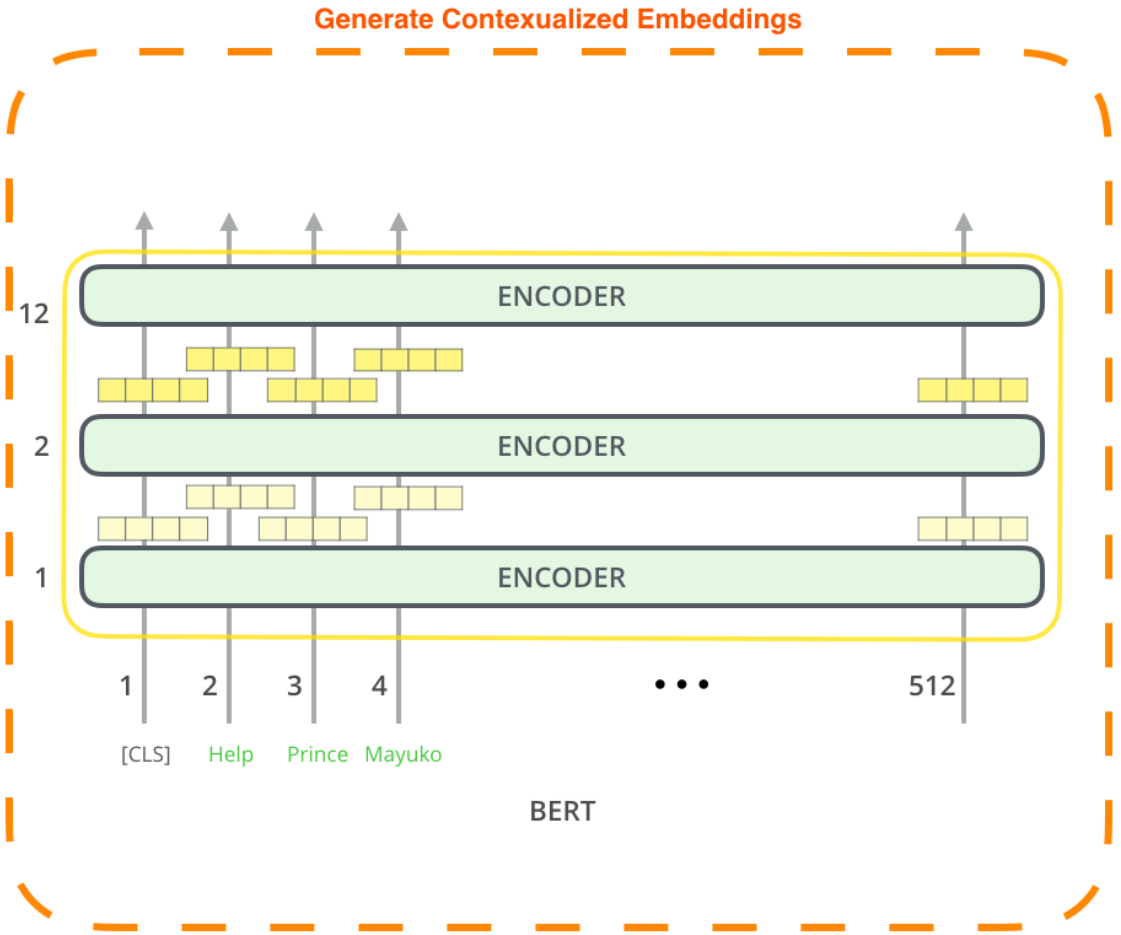
BERT for Contextualized Word Embeddings

BERT for feature extraction

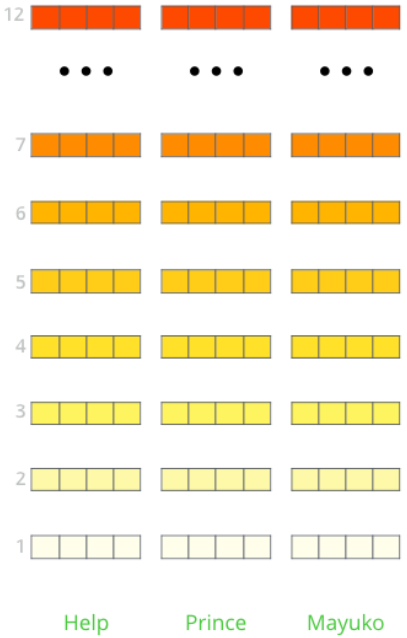
- Use the pre-trained BERT to create contextualized embeddings, and feed these embeddings to existing model優勢：
  - 作者認為並非所有的NLP task都可以表示成Transformer encoder input的樣子，因此可能會額外設計模型的架構
  - 只要先行運算過昂貴的pre-training 得到data的representation後，就可以依據這個表示做為基礎來訓練其他的模型，可以獲得很大的運算優勢

BERT for feature extraction

- Use pre-trained BERT to get contextualized embeddings and feed them into the task-specific models



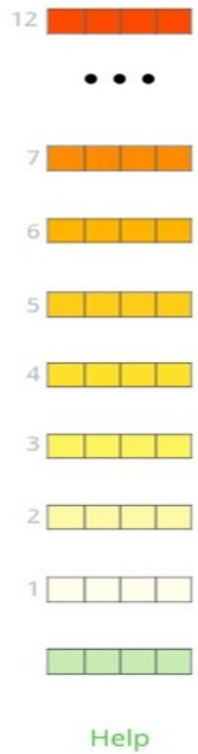
The output of each encoder layer along each token's path can be used as a feature representing that token.



But which one should we use?

BERT Embeddings Results on NER

What is the best contextualized embedding for "Help" in that context?
For named-entity recognition task CoNLL-2003 NER



		Dev F1 Score
First Layer	Embedding	91.0
Last Hidden Layer	12	94.9
Sum All 12 Layers	12 + ... + 2 + 1 =	95.5
Second-to-Last Hidden Layer	11	95.6
Sum Last Four Hidden	12 + 11 + 10 + 9 =	95.9
Concat Last Four Hidden	9 10 11 12	96.1

fine-tune = **96.4**

Conclusion

Remarks

Conclusion

- BERT – a general approach for learning contextual representations from Transformers and benefiting language understanding
- Contextualized embeddings learned from masked LM via Transformers provide informative cues for **transfer learning (pre-train + fine-tune)**
- Recent empirical improvements due to transfer learning with language models have demonstrated that rich, unsupervised pre-training is an integral part of many language understanding systems
- 即便對資料量很少的task仍可以受益於very deep bidirectional architectures，使BERT能成功地處理非常廣泛的NLP tasks，在某些task的表現甚至超越人類
- 未來希望可以證明BERT得以抓到語言現象(linguistic phenomena)

Thank You

GitHub: tychen5

LinkedIn: tychen5

SITCON (2019)