BERT – Bird's Eye View

PART1

Learning Notes of Senthil Kumar

Agenda

Bird’s eye view (PART1)

About Language Models

What is BERT? – A SOTA LM

What does BERT do? – Multi-task Learning

How is BERT built? – BERT Architecture

BERT Fine-tuning and Results

What is powering NLP's Growth?

Two important trends that power recent NLP growth

- Better **representation** of the text data (with no supervision)
  - By grasping the 'context' better
    (e.g. terms used language models, word embeddings, contextualized word embeddings)
  - By enabling 'transfer learning'
    (e.g.: term used pre-trained language models)

- **Deep Learning Models** that use the better representations to solve real-world problems like Text Classification, Semantic Role Labelling, Translation, Image Captioning, etc.,

Important Terms

Language Model:
A **language model (LM)** is a model that **generates a probability distribution** over sequences of words
Language Model - A model to predict the next word in a sequence of words

****

Encoder:
Converts variable length text (any sequence information) into a fixed length vector

****

Transformer:
- A sequence model forgoes the recurrent structure of RNN to adopt attention-based approach
- Transformer vs Seq2Seq Model
- { Encoder + (attention-mechanism) + Decoder } vs { Encoder + Decoder }

****

Recurrent Structure: Processes all input elements sequentially
Attention-based Approach: Process all input elements SIMULTANEOUSLY

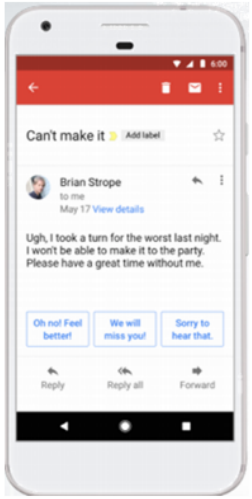
Intro – Count-based N-gram LM (1/5)

- Task of **predicting the next word** given the previous words
- A language model can assign **probability to each possible next word**. And also, help in assigning a probability to an entire sentence.
- Arguably, the **simplest and most critical language processing task** with concrete practical applications
- Language modelling is a form of unsupervised, predictive learning --> no labeled text needed

Applications:

Predicting upcoming words or estimating probability of a phrase or sentence is useful in noisy, ambiguous text data sources

a. **Speech Recognition** - E.g.: P("recognize speech") >> P("wreck a nice beach")
b. **Spelling Correction** – E.g.: P("I have a gub") << P("I have a gun")
c. **Machine Translation** – E.g.: P("strong winds") > P("large winds")
d. **Optical Character Recognition/ Handwriting Recognition**
e. **Autoreply Suggestions**. E.g.: **Intelligent keyboards, auto email reply**
f. **Text Classification**



Sample Corpus:

This is **the house** that Jack built.
This is **the** malt
That lay in **the house** that Jack built.
This is the rat.
That ate **the** malt
That lay in **the house** that Jack built.
This is the cat,
That killed **the** rat,
That ate **the** malt
That lay in **the house** that Jack built.

P (house | the) = count (the house) / count (the)

Aside from the intuitive way we would calculate, this is an example of **Conditional Probability**!

$P(B \mid A) = P(A, B) / P(A)$

Technically, what we have computed above is **a Bigram Language Model**

$$Bigram\ model: \ p(w_t \mid w_{t-1})$$

Bi-gram LM
(generally, an N-gram LM)

Markov Bi-gram LM

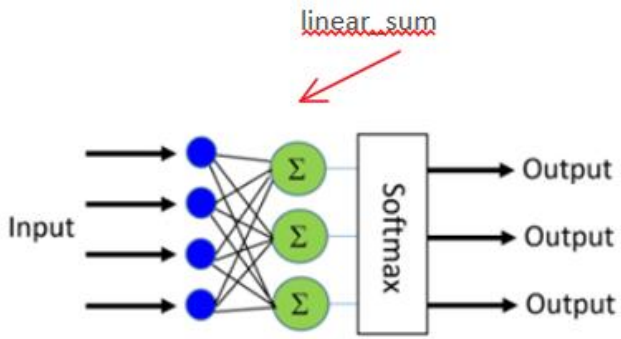
Hence the probability of occurrence of the 5-word sentence is:

$$p(A,B,C,D,E) = p(E \mid D)p(D \mid C)p(C \mid B)p(B \mid A)p(A)$$

Sources:
NLP course on Coursera – National Higher School of Economics
Advanced NLP and Deep Learning course on Udemy (by LazyProgrammer)

Intro – Context-Prediction based LMs (2/5)

Logistic Bigram Model

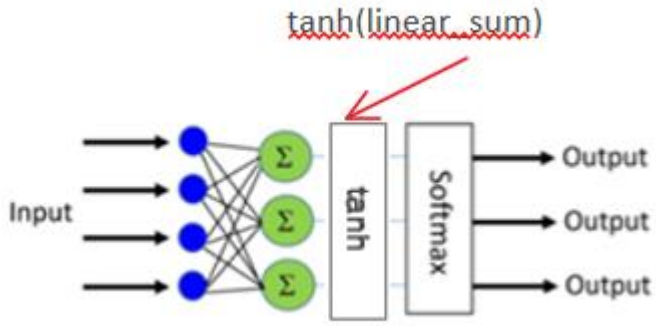


$$P(y|x) = softmax(\mathrm{W}^{\mathrm{T}}\,\mathrm{x})$$

$$P(y=j|x) = \frac{exp\,(w_j{}^T x + bj)}{\sum_{k\,\in\,K} exp\,(w_k{}^T x + bk)}$$

Feed Forward *Neural Network LM*

proposed by Bengio in 2001



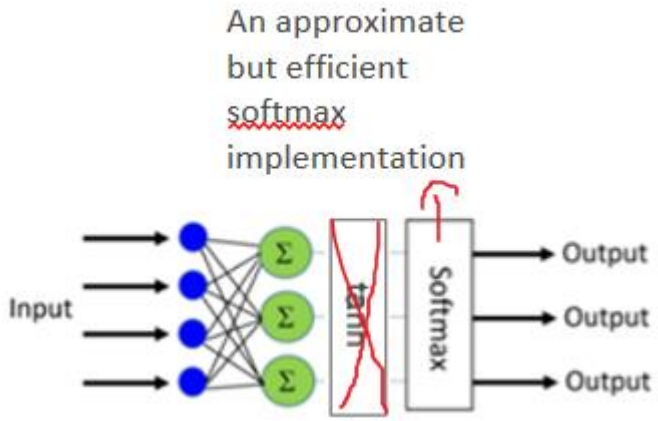
$$P(y|x) = softmax(\mathrm{tanh}(\mathrm{W}^{\mathrm{T}}\,\mathrm{x}))$$

$$P(y=j|x) =$$

$$\frac{exp\,(\tanh(wjT\,x + bj))}{\sum_{k\,\in\,K}\,exp\,(tanh(w_k{}^T x + bk))}$$

Word2Vec

proposed by Mikilov in 2001



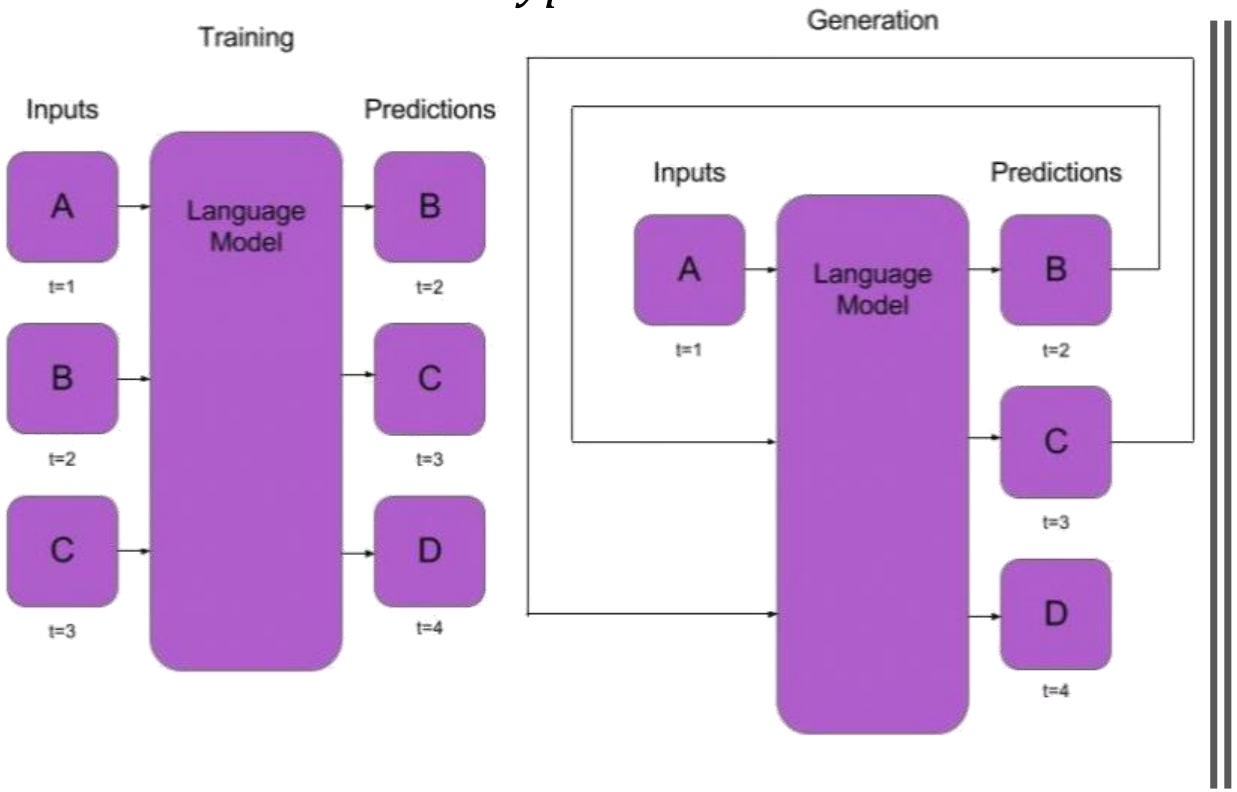
$$P(y|x) = neg\_sampling\_softmax(\mathrm{W}^{\mathrm{T}}\,\mathrm{x})$$

Sources:

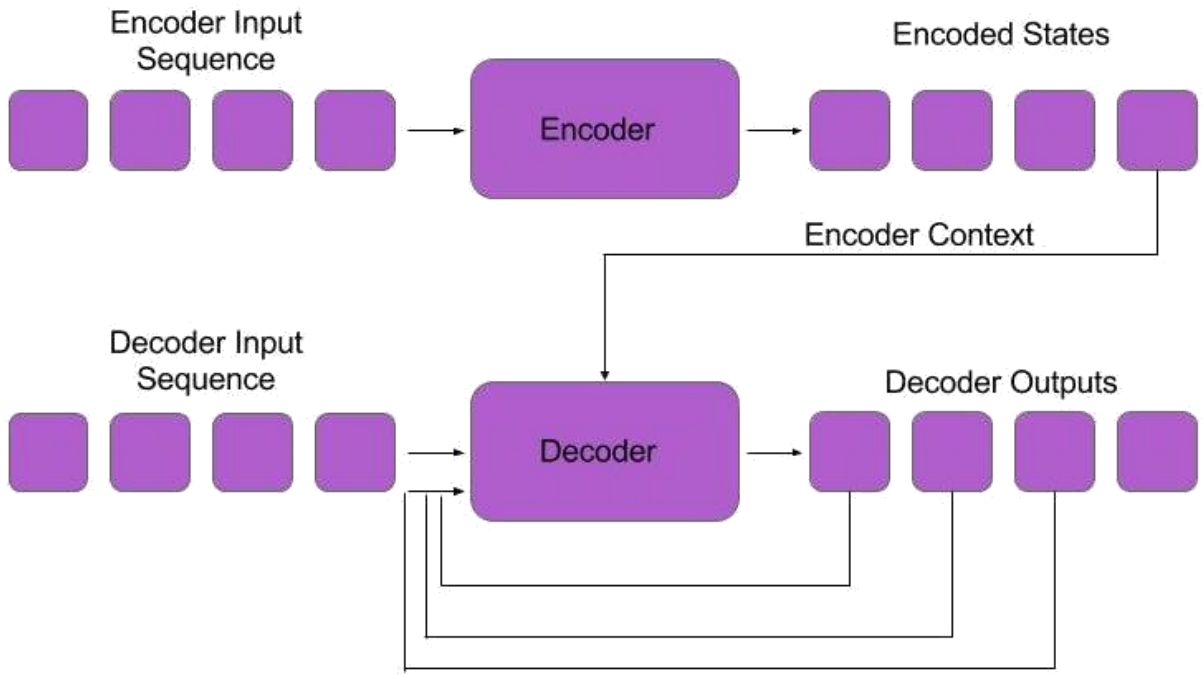
Advanced NLP and Deep Learning course on Udemy (by LazyProgrammer); Idea: http://www.marekrei.com/blog/dont-count-predict/

Higher Form of LMs(3/5)

Typical LM



Seq2Seq – A higher form of LM

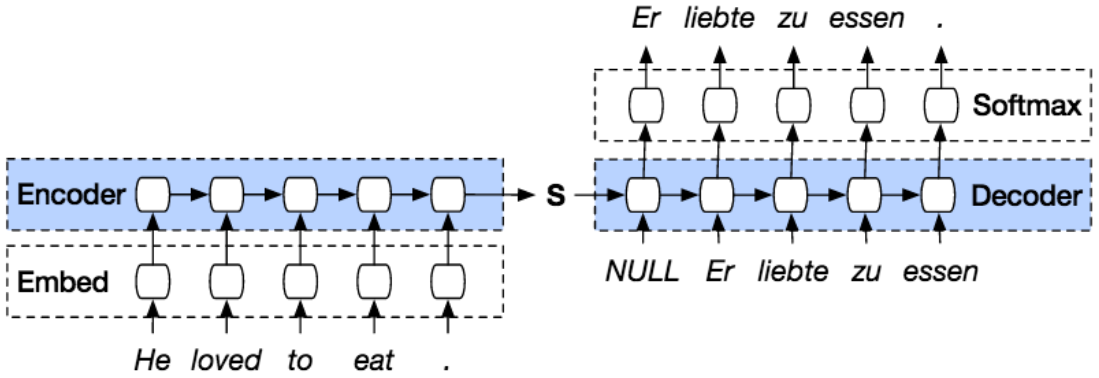


Sequence to sequence models build on top of language models by adding an encoder step and a decoder step. the decoder model sees an encoded representation of the input sequence as well as the translation sequence, it can make more intelligent predictions about future words based on the current word

Source:
https://indico.io/blog/sequence-modeling-neuralnets-part1/

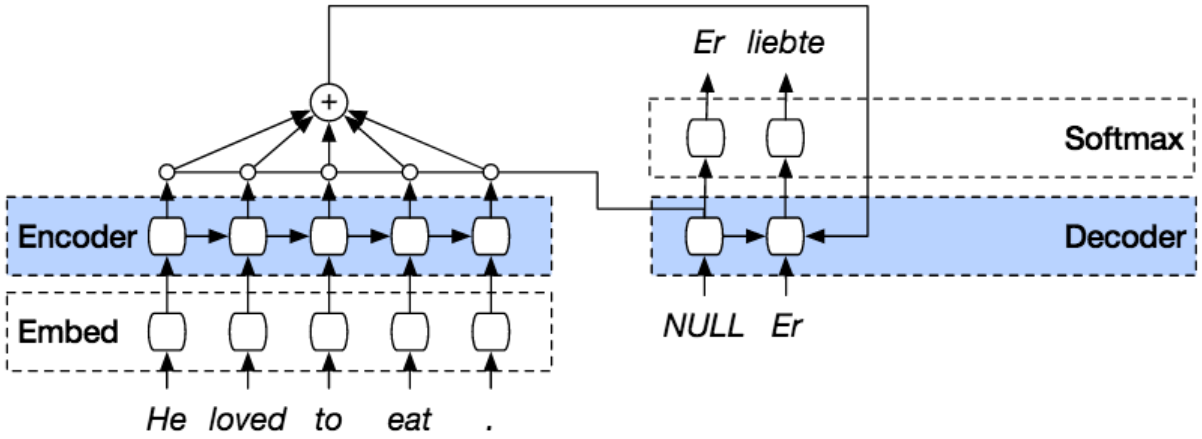
Higher Form of LMs(4/5)

Typical Seq2Seq Model



{ Encoder + Decoder }
Recurrent Structure: Processes all input elements sequentially

Transformer Model



{ Encoder + (attention-mechanism) + Decoder }
Attention-based Approach: Process all input elements SIMULTANEOUSLY

Sources:
https://towardsdatascience.com/nlp-sequence-to-sequence-networks-part-2-seq2seq-model-encoderdecoder-model-6c22e29fd7e1

Pre-trained Language Models (5/5)

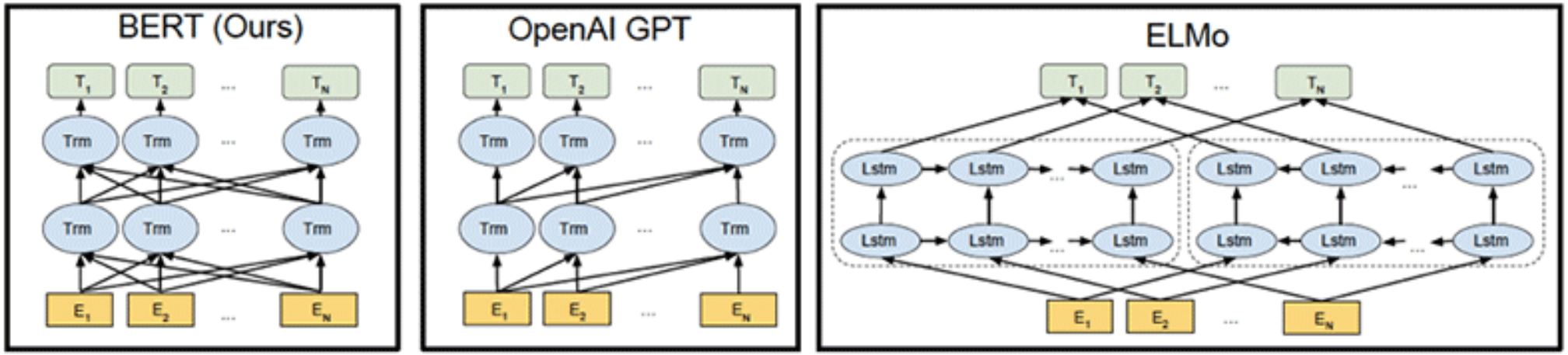
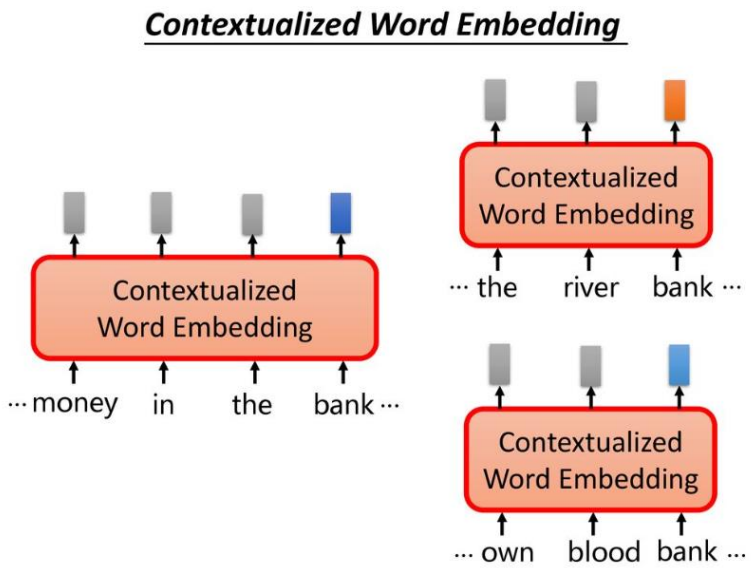


Figure 1: Differences in pre-training model architectures. BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTM to generate features for downstream tasks. Among three, only BERT representations are jointly conditioned on both left and right context in all layers.

Sources: BERT Paper - https://arxiv.org/pdf/1810.04805.pdf



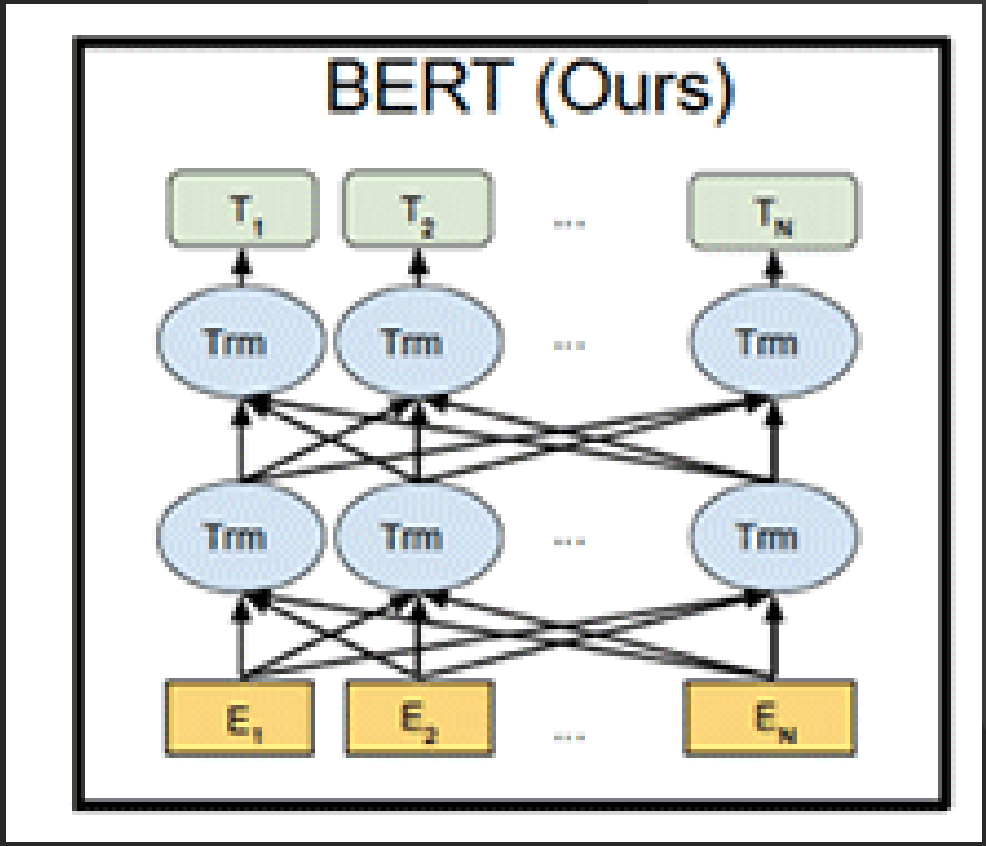
What is BERT

- Bidirectional Encoder Representations from Transformers
- SOTA Language Model
- ULMFiT, ELMo, BERT, XLNet -- NLP's ImageNet Moment – the era of contextualized word embedding and Transfer Learning

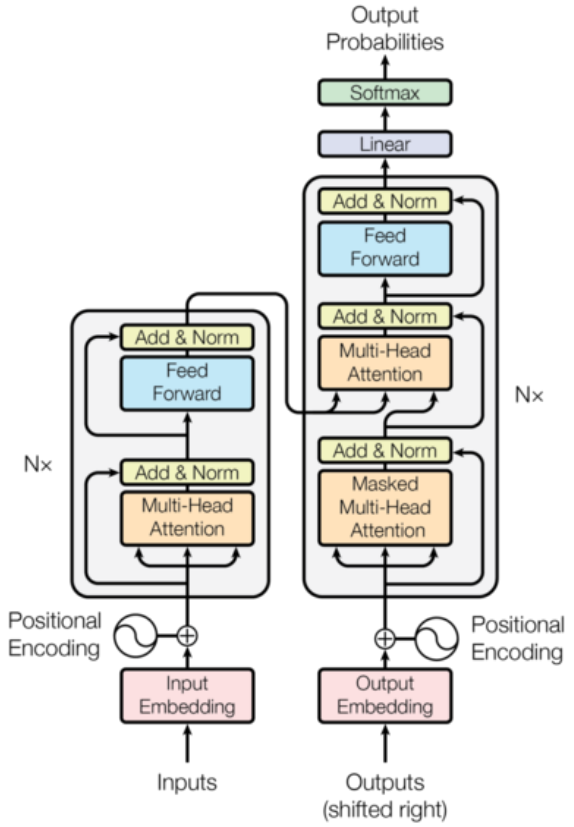
Sources:
https://slideplayer.com/slide/17025344/

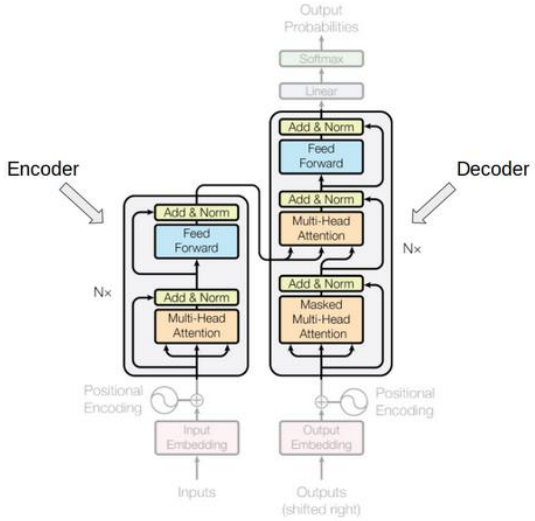
BERT is Bidirectional

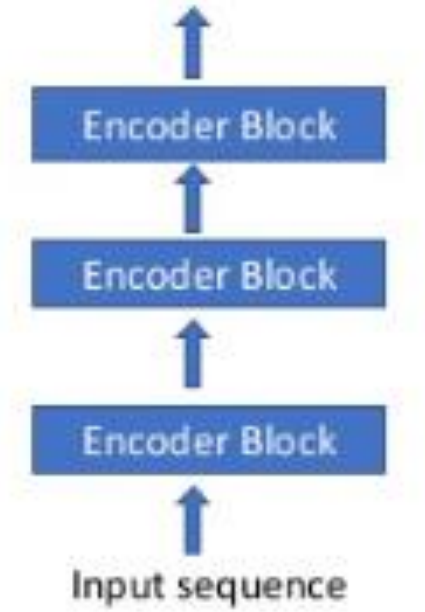
- BERT is bidirectional – reads text from both directions, left as well as from right to gain better understanding of the text
- For example, the above sentence is read from BERT is bidirectional … as well as … understanding of the text
- BERT is bidirectional because it uses Transformer Encoder (which is made bidirectional by the self attention mechanism)
- OpenAI Generative Pre-trained Transformer is uni-directional because it employs Transformer Decoder (the masked self-attention makes it only left to right)



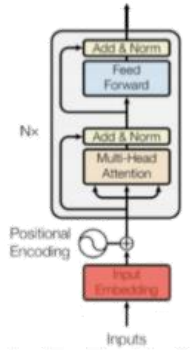
Sources:
https://github.com/google-research/bert/issues/319
https://github.com/google-research/bert/issues/83







Inside an Encoder Block



In BERT experiments, the numbe of blocks N was chosen to be 12 and 24.
Blocks do not share weights with each other

BERT uses Encoder portion of Transformer

Sources: Attention is all you Need: https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

BERT Architecture

- `BERT-Large, Uncased (Whole Word Masking)` : 24-layer, 1024-hidden, 16-heads, 340M parameters
- `BERT-Large, Cased (Whole Word Masking)` : 24-layer, 1024-hidden, 16-heads, 340M parameters
- `BERT-Base, Uncased` : 12-layer, 768-hidden, 12-heads, 110M parameters
- `BERT-Large, Uncased` : 24-layer, 1024-hidden, 16-heads, 340M parameters
- `BERT-Base, Cased` : 12-layer, 768-hidden, 12-heads , 110M parameters
- `BERT-Large, Cased` : 24-layer, 1024-hidden, 16-heads, 340M parameters
- `BERT-Base, Multilingual Cased (New, recommended)` : 104 languages, 12-layer, 768-hidden, 12-heads, 110M parameters
- `BERT-Base, Multilingual Uncased (Orig, not recommended)` **(Not recommended, use** `Multilingual Cased` **instead)**: 102 languages, 12-layer, 768-hidden, 12-heads, 110M parameters
- `BERT-Base, Chinese` : Chinese Simplified and Traditional, 12-layer, 768-hidden, 12-heads, 110M parameters

Source: BERT GitHub Repository: https://github.com/google-research/bert

Pretrained Tasks

Multi-task Learning

Sources:
Idea of Multi-task learning – Karate Kid reference:
https://towardsdatascience.com/whats-new-in-deep-learning-research-how-deep-builds-multi-task-reinforcement-learning-algorithms-f0d868684c45

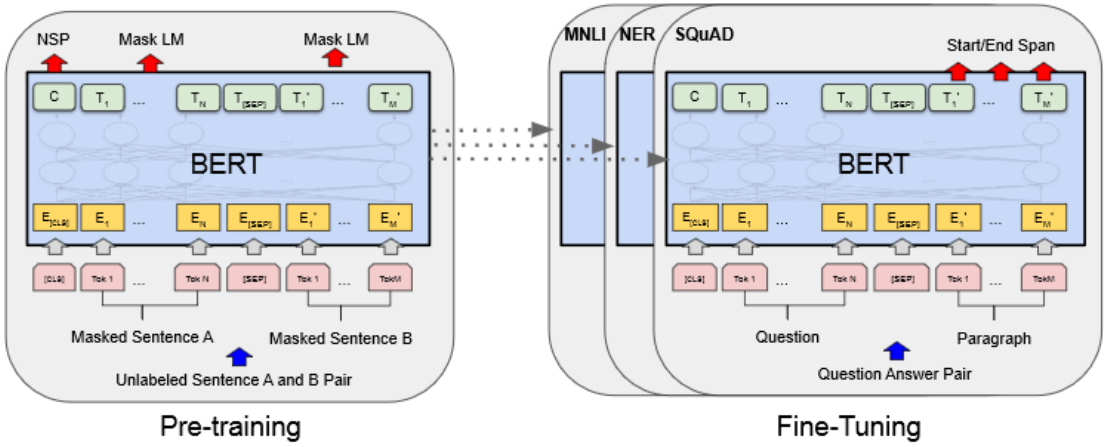
BERT's core methodology:

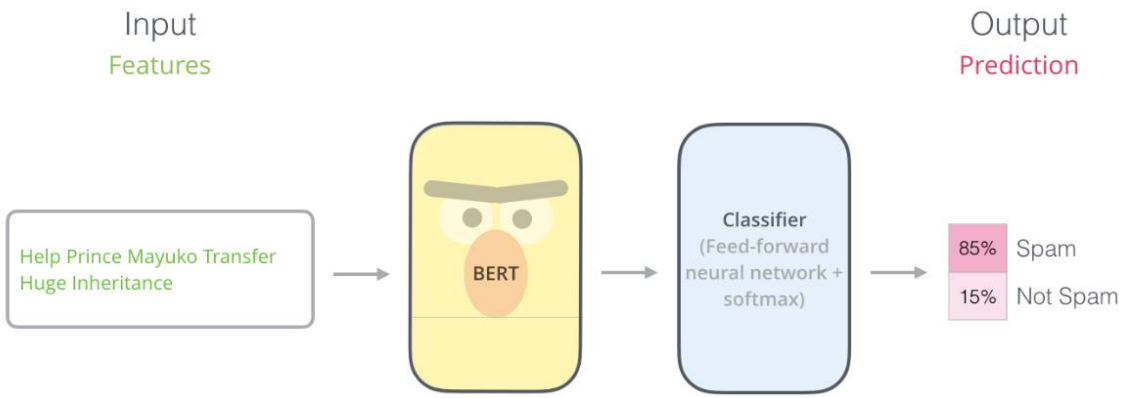
- Applying the bidirectional training of Transformer to do Language Modeling and Next Sentence Prediction

BERT is originally pretrained to do 2 tasks:

- Masked Language Model technique allows bidirectional training for next word prediction
- Next Sentence Prediction: BERT to predict the next sentence

Pretrained Tasks - Finetuning

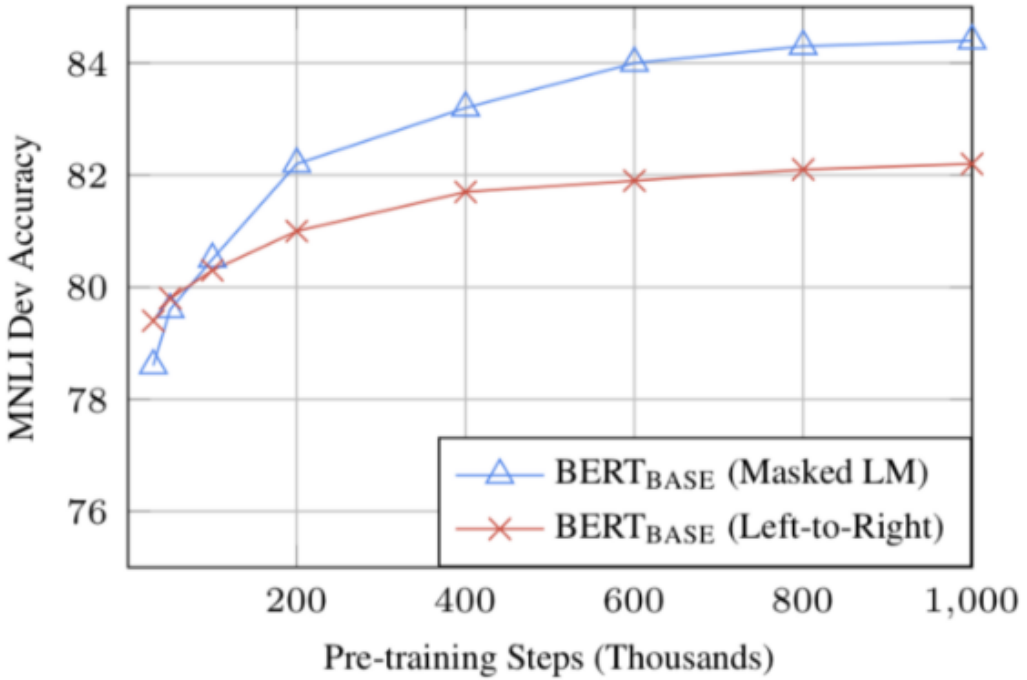




Sources:
BERT Paper - https://arxiv.org/pdf/1810.04805.pdf
http://jalammar.github.io/illustrated-bert/

Task1 - Masked Language Model

- 15% of the words are masked at random
  - and the task is to predict the masked words based on its left and right context
- Not all tokens were masked in the same way (example sentence “My dog is hairy”)
  - 80% were replaced by the <MASK> token: “My dog is <MASK>”
  - 10% were replaced by a random token: “My dog is apple”
  - 10% were left intact: “My dog is hairy”



- BERT (Masked LM technique) converges slower than left to right predictions. But after a small number of training steps, its accuracy beats the unidirectional model
- Multi-genre Natural Language Inference is a text classification task
- MNLI is a large-scale, crowdsourced entailment classification task (Williams et al., 2018). Given a pair of sentences, the goal is to predict whether the second sentence is an entailment, contradiction, or neutral with respect to the first one.

Source: https://medium.com/dissecting-bert/dissecting-bert-part-1-d3c3d495cdb3

Task2 - Next Sentence Prediction

Input = [CLS] the man went to [MASK] store [SEP]

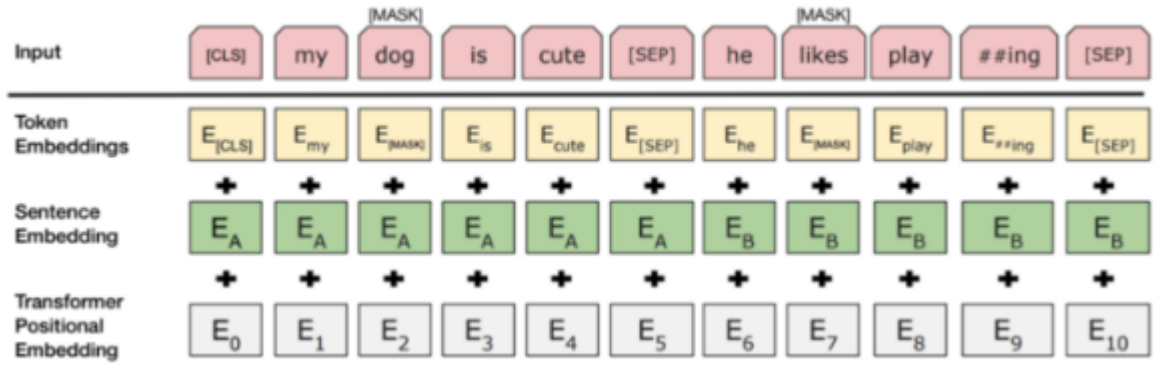
he bought a gallon [MASK] milk [SEP]

Label = IsNext

Input = [CLS] the man [MASK] to the store [SEP]

penguin [MASK] are flight ##less birds [SEP]

Label = NotNext



BERT to predict the next sentence

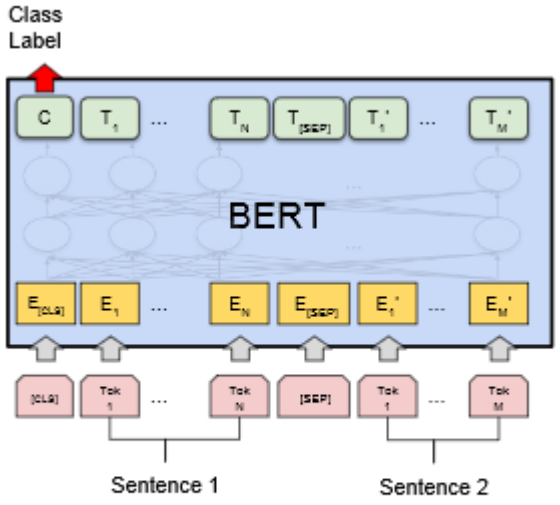
BERT <-- two types of input pairs of sentences are fed to the model

i.50% of the sentence pairs are in the right order

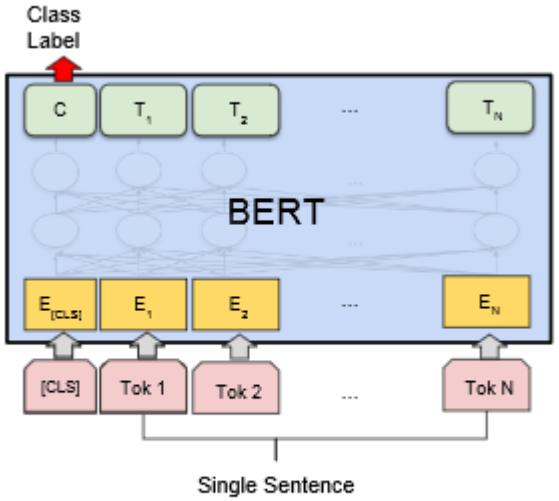
ii.In the other 50%, the second sentence in the pair is randomly chosen

Source: https://medium.com/dissecting-bert/dissecting-bert-part-1-d3c3d495cdb3

BERT Fine-tuning - Classification



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



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

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	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Table 1: GLUE Test results, scored by the evaluation server (https://gluebenchmark.com/leaderboard). The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.[8] BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

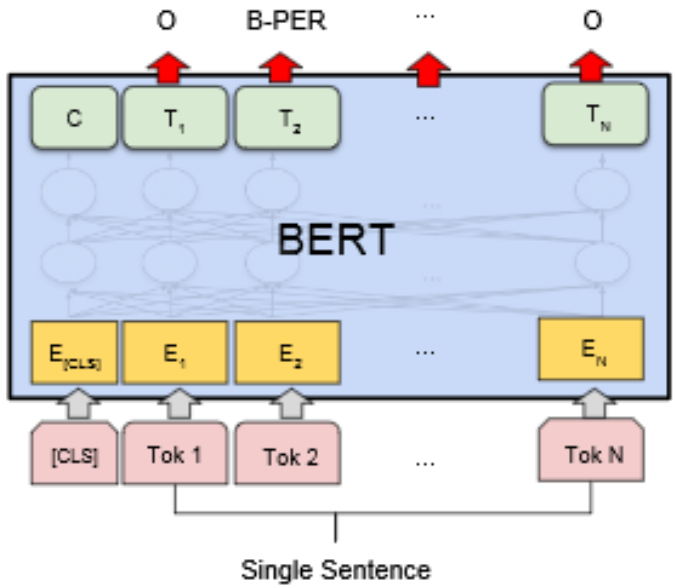
- The General Language Understanding Evaluation(GLUE) benchmark is a col-lection of diverse natural language understandingtasks.

- Mutigenre Natural Language Inference

- Stanford Sentiment Treebank 2
- Corpus of Linguistic Acceptability

Sources: BERT Paper - https://arxiv.org/pdf/1810.04805.pdf

BERT Fine-tuning – Semantic Role Labeling



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

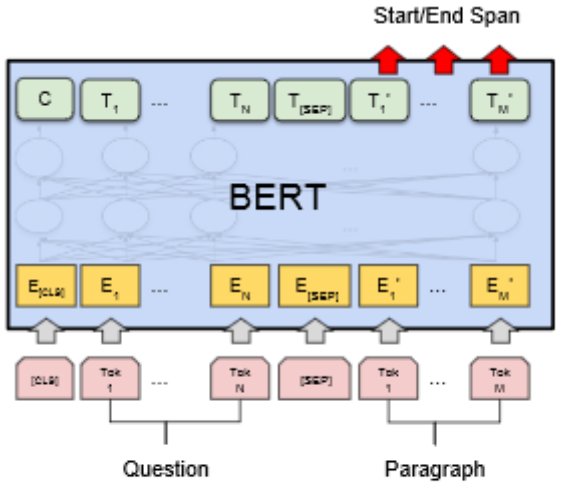
- Conference on Natural Language Learning 2003 NER Task

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

Table 7: CoNLL-2003 Named Entity Recognition results. Hyperparameters were selected using the Dev set. The reported Dev and Test scores are averaged over 5 random restarts using those hyperparameters.

Sources: BERT Paper - https://arxiv.org/pdf/1810.04805.pdf

BERT Fine-tuning – Q&A



(c) Question Answering Tasks: SQuAD v1.1

TheStanfordQuestionAnsweringDataset

System	Dev EM	Dev F1	Test EM	Test F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
Published				
BiDAF+ELMo (Single)	-	85.6	-	85.8
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$\text{BERT}_{\text{BASE}}$ (Single)	80.8	88.5	-	-
$\text{BERT}_{\text{LARGE}}$ (Single)	84.1	90.9	-	-
$\text{BERT}_{\text{LARGE}}$ (Ensemble)	85.8	91.8	-	-
$\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$\text{BERT}_{\text{LARGE}}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

Table 2: SQuAD 1.1 results. The BERT ensemble is 7x systems which use different pre-training checkpoints and fine-tuning seeds.

Sources: BERT Paper - https://arxiv.org/pdf/1810.04805.pdf

Thank You