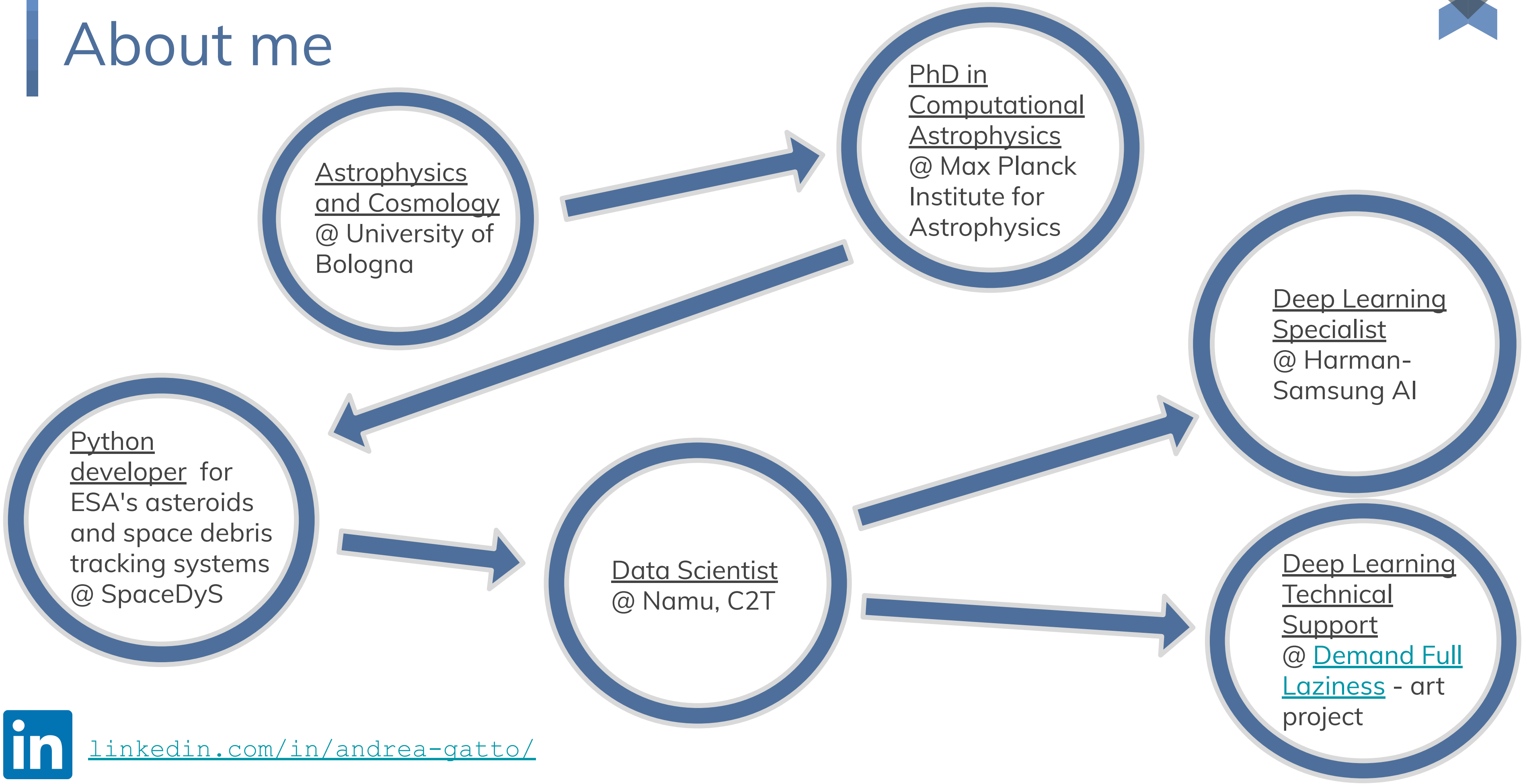
State-of-the-art Natural Language Processing with BERT

Andrea Gatto

Deep Learning Specialist at Harman-Samsung AI

Deep Learning Italia meetup - 18 Feb 2020

About me



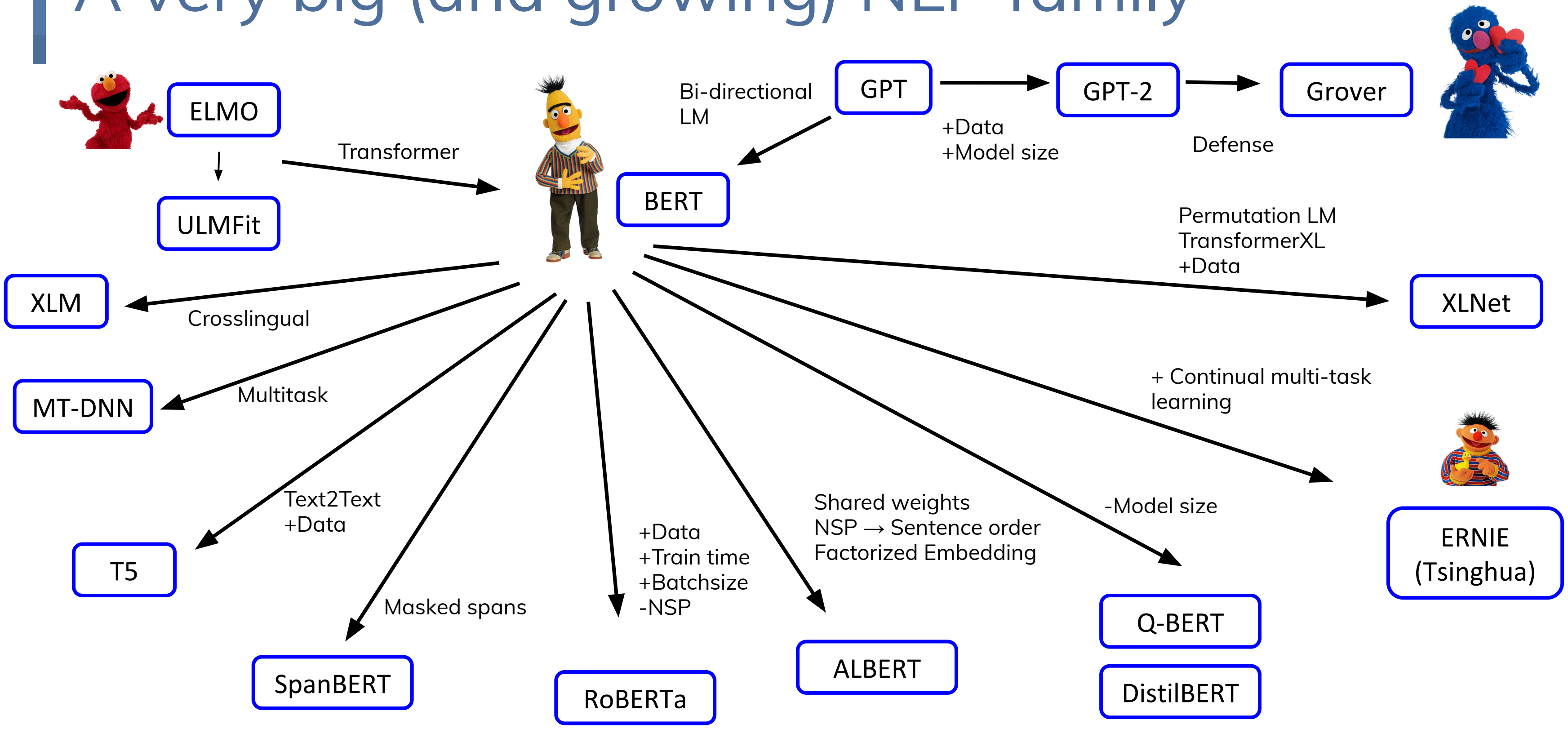
linkedin.com/in/andrea-gatto/

Outline

- What is BERT?
- Pre-training BERT
- Fine-tuning BERT for language modelling
- Downstream tasks with BERT
- Adversarial attacks and Clever Hans effect

What is **BERT**?

A very big (and growing) NLP family

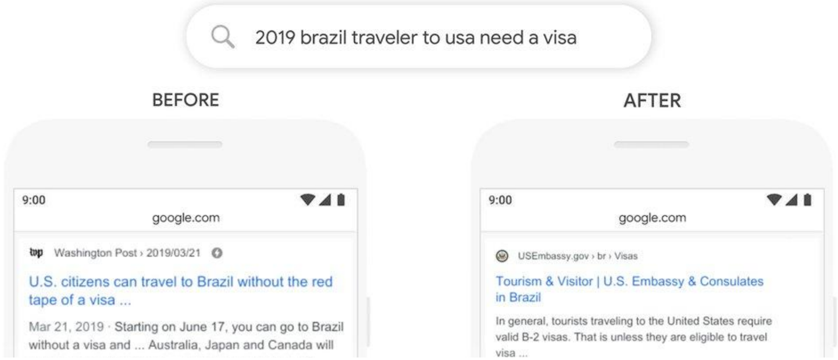


Credit: deepset.ai - Z. Zhang & X. Wang - https://docs.google.com/presentation/d/1INCmHpmFC9s5atS2w5J01wDoEweyzau3dVTktT02gOc/edit?usp=sharing

Applications

- Sentence classification
  “Hello Sir I am Nigerian Prince good business opportunity send money plz” → spam

- Understand search queries better



- Question Answering (QA)
  “Professor Plum killed Dr. Black in the dining room with the candlestick”
  Q: *Where was Dr. Black killed?* → A: *In the dining room*

- Etc...

What is BERT?

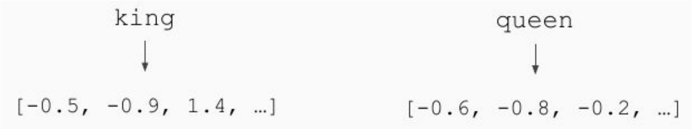
BERT =
Bidirectional Encoder Representations from Transformers

Paper by Google AI Devlin et al., Oct. 2018
Pre-training of Deep Bidirectional Transformers for Language Understanding
(arxiv/1810.04805)

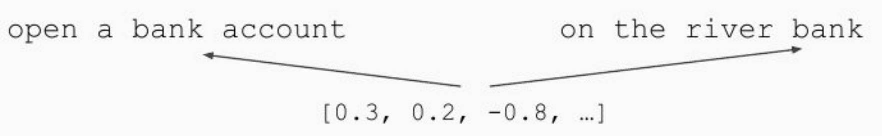
BERT(/transformer)-based architectures currently give state-of-the-art performance on many NLP tasks

Bidirectional Encoder **Representations** from Transformers

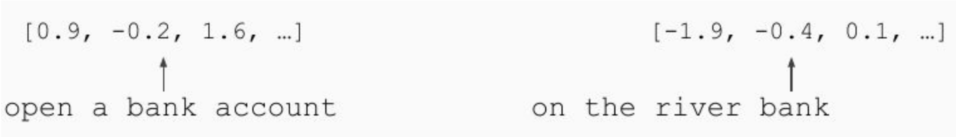
Word embeddings are the basis for NLP



Problem:
Word embeddings are applied in a context free manner



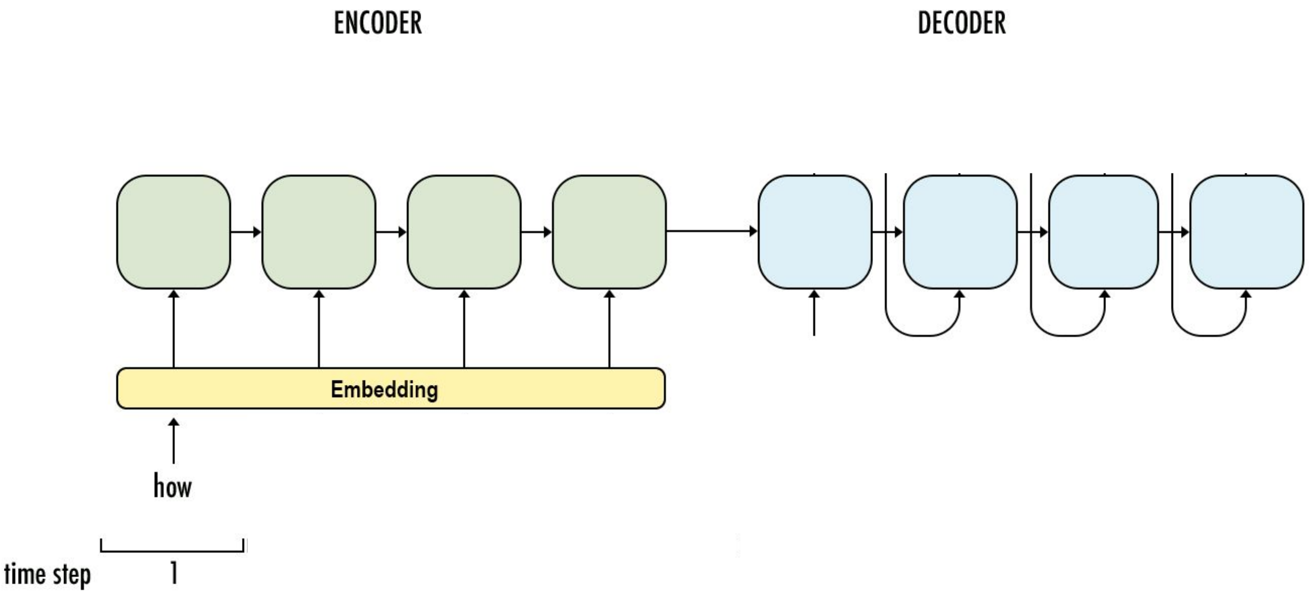
Solution:
Train contextual representations of words



Credit: https://nlp.stanford.edu/seminar/details/jdevlin.pdf

Bidirectional Encoder Representations from Transformers

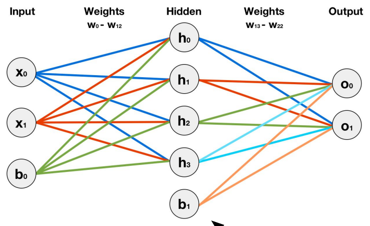
From Recurrent Neural Networks...



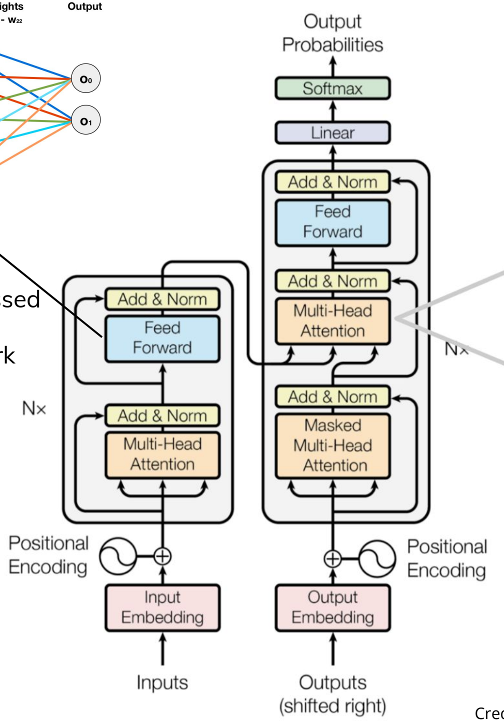
Credit: https://towardsdatascience.com/language-translation-with-rnns-d84d43b40571

Bidirectional Encoder Representations from **Transformers**

… to Transformers



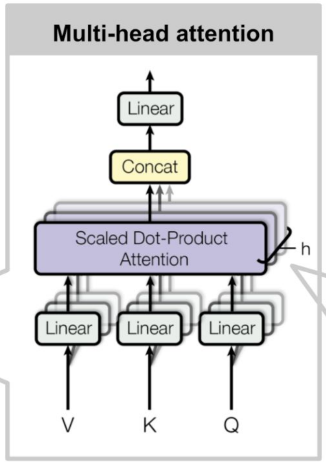
Position-wise
(= words/tokens processed independently)
feed-forward network



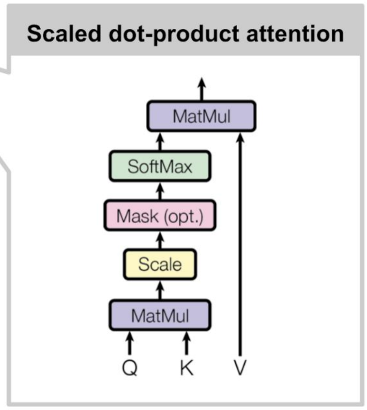
$$\mathrm{MultiHead}(Q,K,V) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_\mathrm{h})W^O$$

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$XW^K = K$$
$$XW^Q = Q$$
$$XW^V = V$$



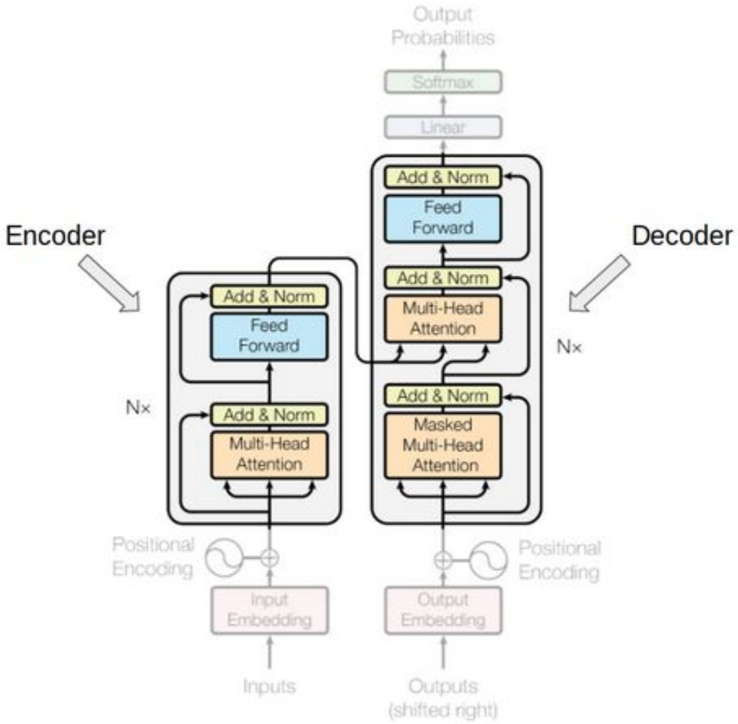
Zoom-In!



Zoom-In!

Credit: https://github.com/lilianweng/transformer-tensorflow

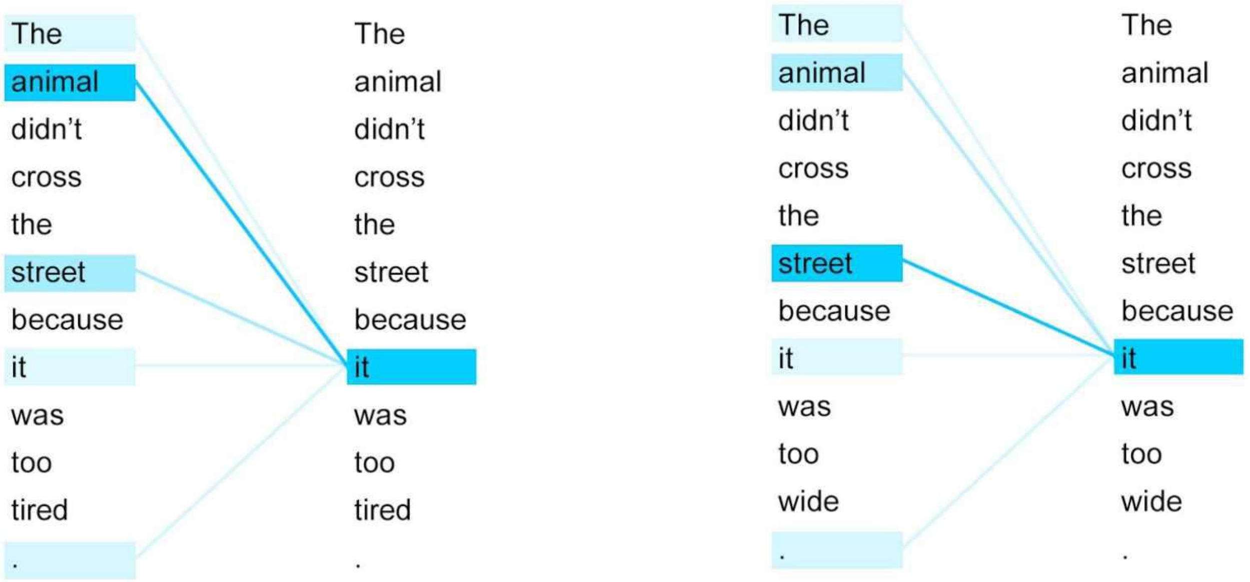
Bidirectional **Encoder** Representations from Transformers



Credit: https://www.cluzters.ai/Feed-view/1/16207/how-do-transformers-work-in-nlp-a-guide-to-the-latest-state-of-the-art-models?

Bidirectional Encoder Representations from Transformers

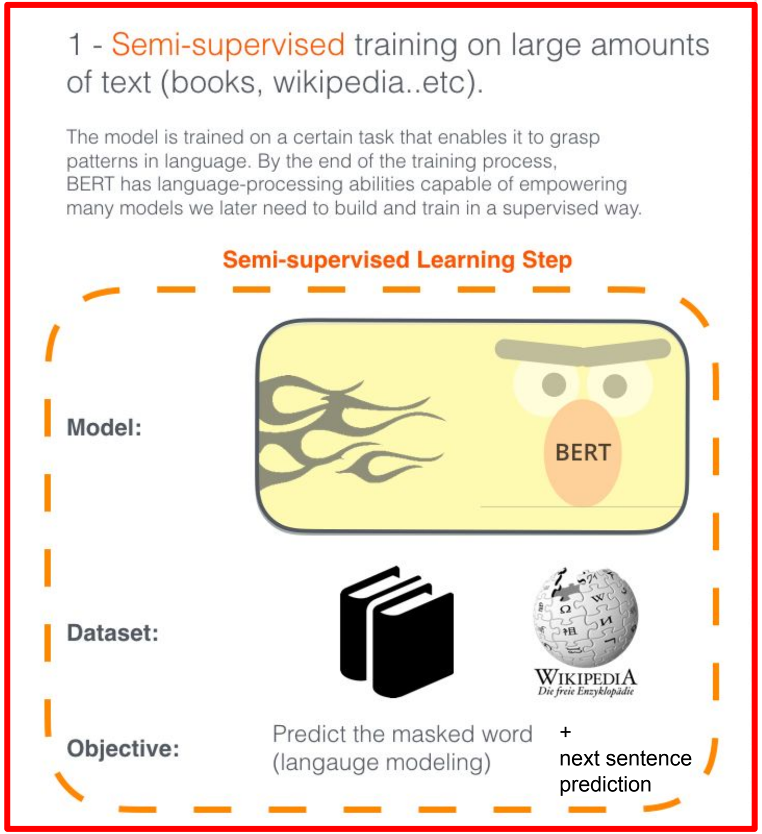
Self-Attention



$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Credit: https://www.infoq.com/presentations/nlp-practitioners/

The idea behind BERT



<u>Goal of pre-training</u>:

obtain general, contextual word (aka token aka subword) embeddings to be used (and updated) for downstream tasks

Credit: http://jalammar.github.io/illustrated-bert/

Pre-training **BERT**

Pre-training BERT

1. Build vocabulary/tokenizer and tokenize sentences

2. Masked language model task

3. Next sentence prediction task

4. Token, segment and positional encodings

5. Train

1. Build vocabulary/tokenizer and tokenize sentences

Build dictionary/tokenizer via WordPiece tokenization, then apply to corpus

Example:

```
This tokenization is really cool
```

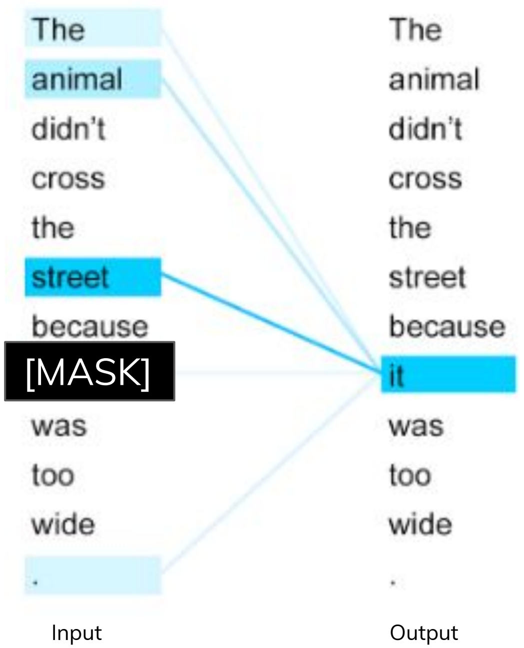
becomes

```
['[CLS]', 'This', 'token', '##ization', 'is', 'really', 'cool', '[SEP]']
```

Special tokens (see next slides)

2. Masked Language Model

Why masked language model?

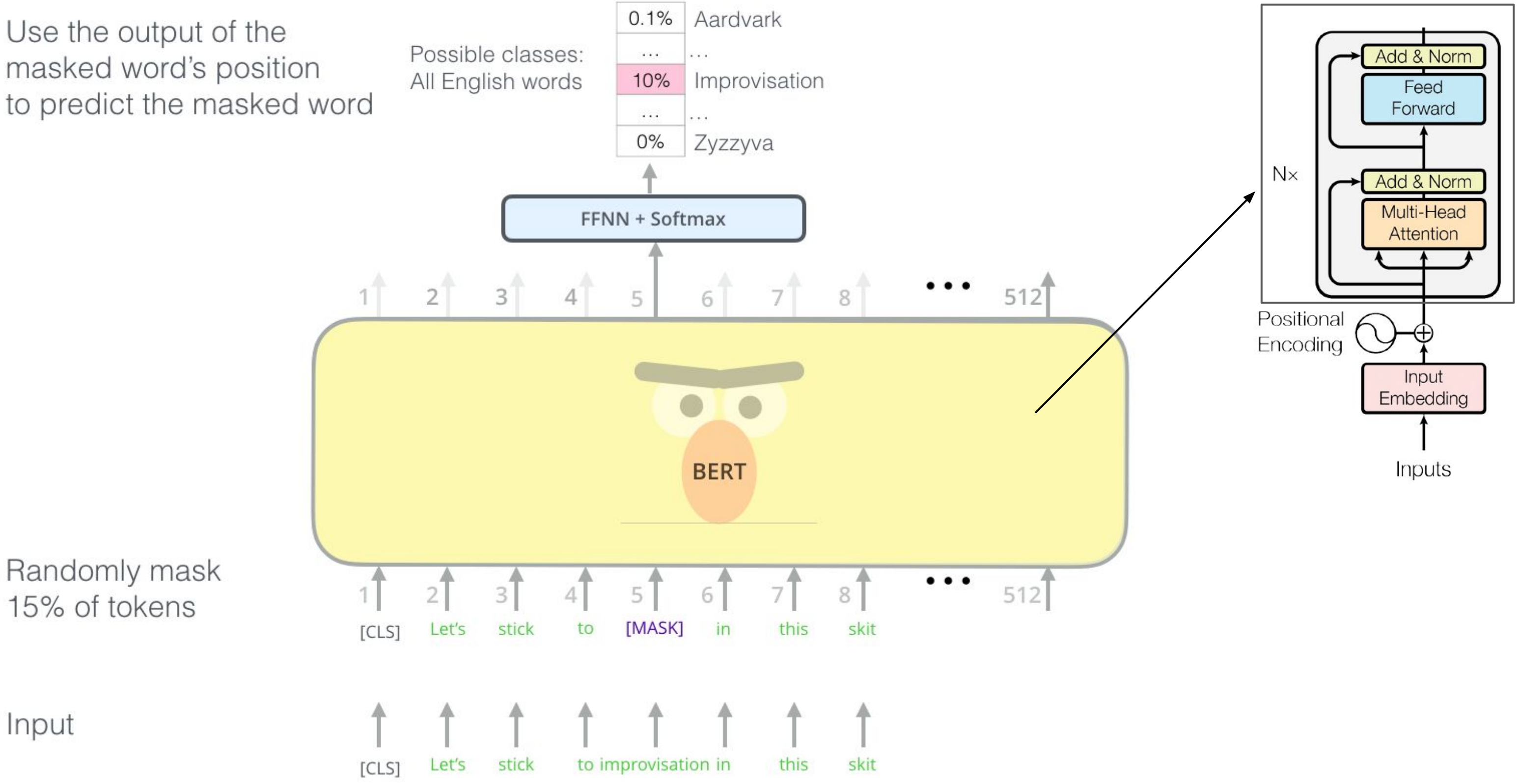


Problem:
In a bidirectional context target words/tokens can "see themselves"

Solution:
Mask out N% of input tokens, then predict them
Pick the "right" N in order to avoid

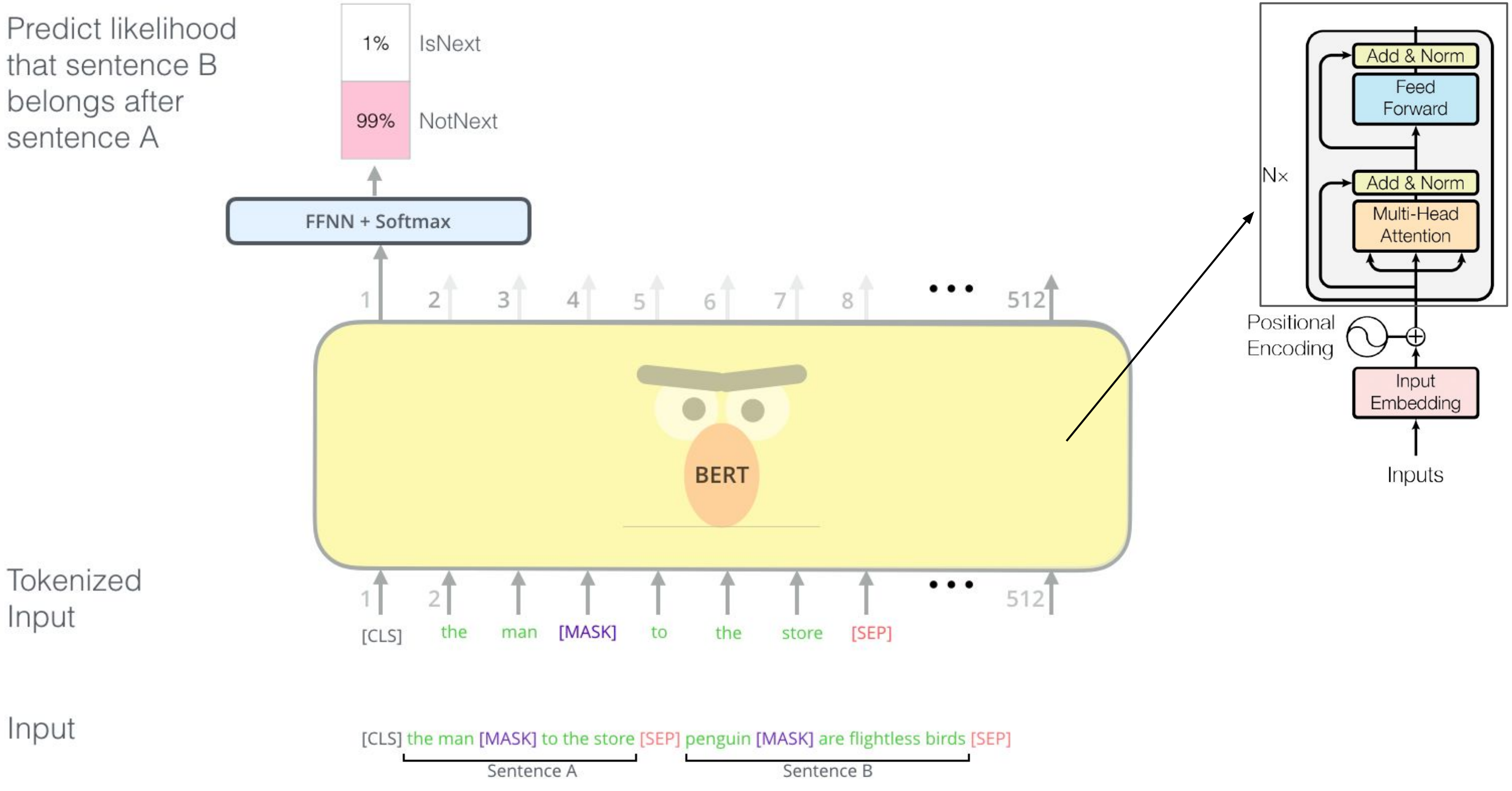
- Not enough context for N too large
- Model too expensive to train for N too low

2. Masked Language Model



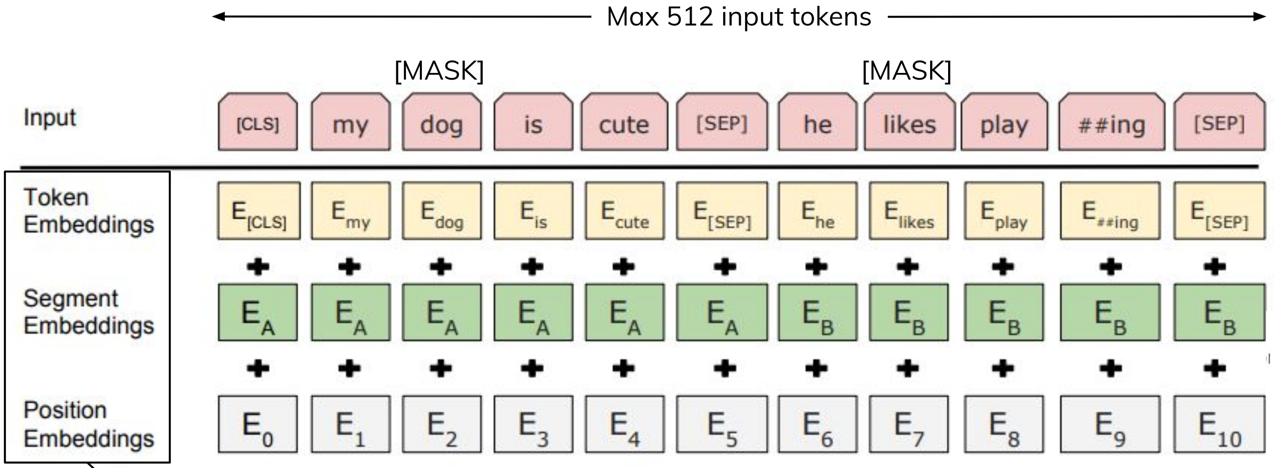
Credit: http://jalammar.github.io/illustrated-bert/

3. Next Sentence Prediction

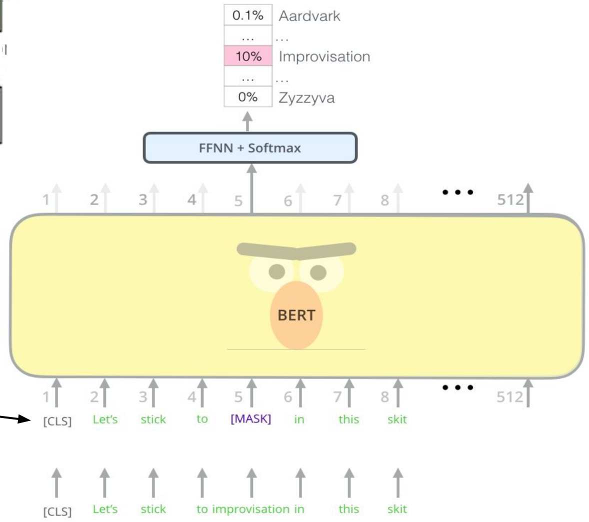


Credit: http://jalammar.github.io/illustrated-bert/

4. Input embeddings



Randomly initialized
Learned during training



Credit: BERT paper (arxiv/1810.04805) - http://jalammar.github.io/illustrated-bert/

5. Train

Train like there's no tomorrow. English model took 4 days on 4 to 16 cloud TPUs.
Produce N pre-trained models divided as:

BERT large

- encoder layers = 24
- hidden size = 1024
- self-attention heads = 16
- Parameters = 340M
- batch size = 256 sentences
- epochs = 40 (1M steps)

BERT base

- encoder layers = 12
- hidden size = 768
- self-attention heads = 12
- Parameters = 110M
- batch size = 256 sentences
- epochs = 40 (1M steps)

Examples:

BERT English (large and base)

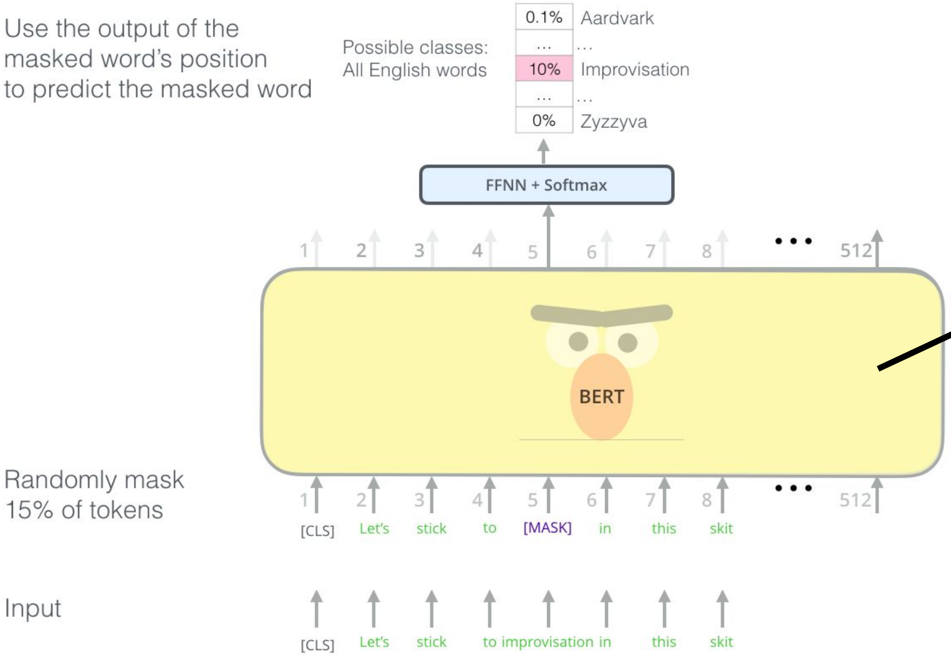
- BOOKCORPUS + English Wikipedia
- 16 GB of data
- 3.3 billion word corpus
- 30k vocabulary size

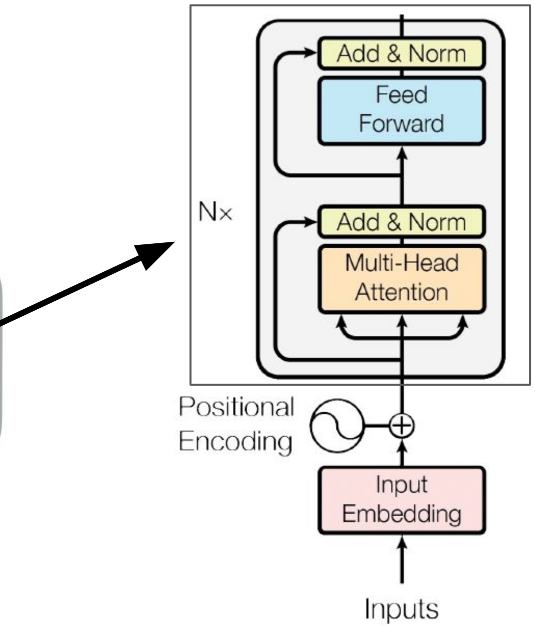
BERT Multilingual (only base)

- Top 104 Wikipedia languages
- ? GB of data
- ? billion word corpus
- 110k vocabulary size

Word/token embeddings

Finally after training we have our pre-trained model with contextual word/token embeddings!!!

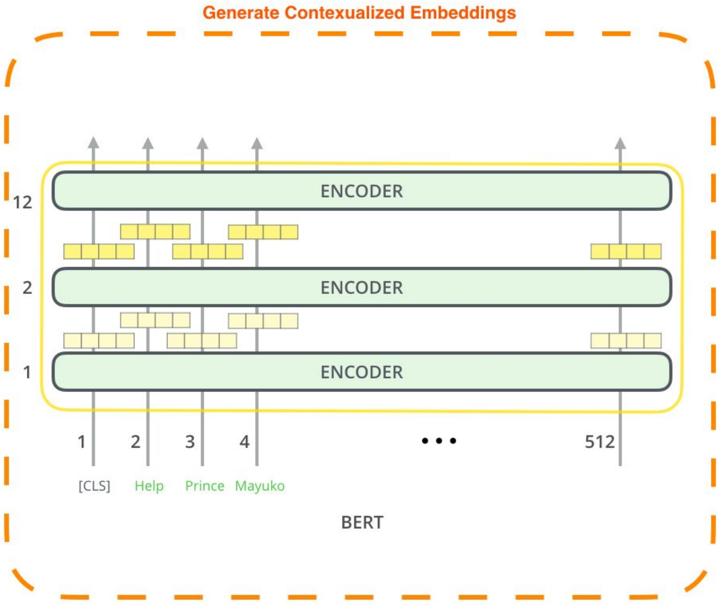




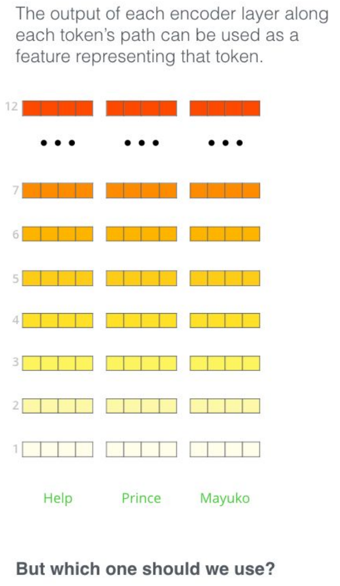
Where are they?

Word/token embeddings

Every encoder layer generates an embedding corresponding to the token + each embedding depends on the context of (words within) the sentence
More on this later when discussing feature-based approach with BERT



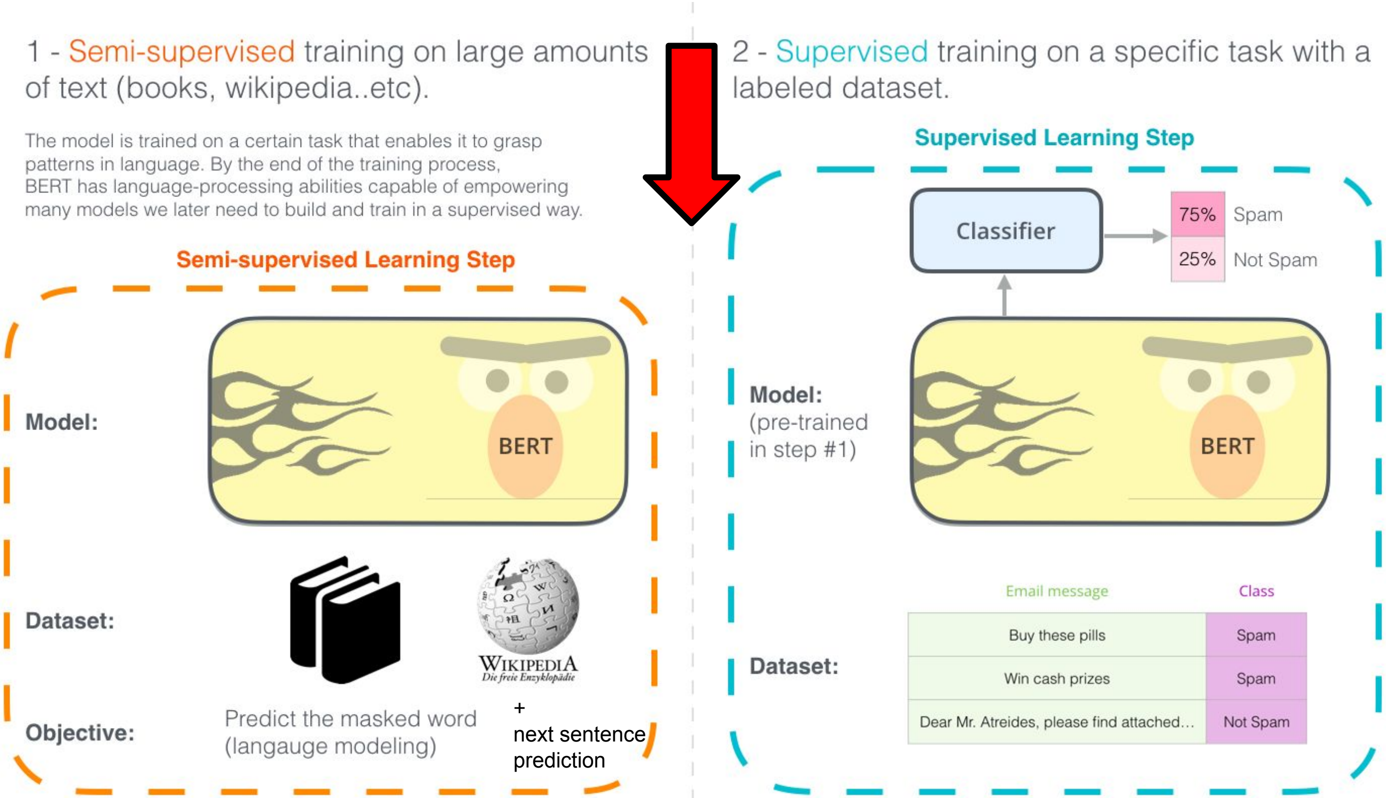
Credit: http://jalammar.github.io/illustrated-bert/



Don't even start talking about sentence embeddings!

Fine-tuning **BERT** for **language modelling**

The idea behind BERT



Credit: http://jalammar.github.io/illustrated-bert/

Adapting BERT to a specific language, e.g. multilingual to Italian - but see also monolingual Italian models AlBERTo (University of Bari), GilBERTo (Ernst & Young), UmBERTo (Musixmatch)

Adapting BERT to a certain domain

Fine-tuning BERT for language modelling

Example:

We take Italian sentences belonging to a particular domain and try to adapt the multilingual pre-trained BERT base to Italian, and to this domain as well

We use:

- cartesian grid search for hyperparameter tuning
- 10-fold cross-validation + early stopping
- dynamic masking instead of static
- no next sentence prediction task

Let's check how this model performs with respect to the pre-trained multilingual BERT

Fine-tuning BERT for language modelling

BERT base pre-trained multilingual

Loss = 4.850
Perplexity = 127.801

```
fa ##mmi *[vedere]* dove è mos ##cov ##a a mil *[##ano]* sulla map ##pa
Top 5 predicted = [##e   | ##a | ##ere | ##ei | ##cone]
Probability (%) = [72.9 | 9   | 2.4   | 1.4  | 1.1]
Top 5 predicted = [##iare | ##ano | ##anese | ##ana | mil]
Probability (%) = [14.7   | 14.3  | 8.8     | 6.1   | 5.1]
```

```
cambia il *[giorno]* *[della]* terza sve ##glia in elenco a sa ##bato
Top 5 predicted = [numero | nome | codice | colore | percorso]
Probability (%) = [15.6   | 12.7 | 3.2    | 2.3    | 2.2]
Top 5 predicted = [della | dalla | di   | alla | da]
Probability (%) = [41    | 21.4  | 11.3 | 7.3  | 3.6]
```

```
*[che]* animale è la form ##ica
Top 5 predicted = [un  | Un | '   | questo | Lo]
Probability (%) = [9.9 | 7  | 5.9 | 5      | 4.1]
```

```
da *[dove]* vien ##i tu
Top 5 predicted = [cui | '   | chi | ,   | ##v]
Probability (%) = [7.1 | 4.4 | 3.8 | 3.4 | 3]
```

Fine-tuned model

Loss = 2.121
Perplexity = 8.342

```
fa ##mmi *[vedere]* dove è mos ##cov ##a a mil *[##ano]* sulla map ##pa
Top 5 predicted = [vedere | sapere | dire | di  | da]
Probability (%) = [93.4   | 4.6    | 0.9  | 0.3 | 0.2]
Top 5 predicted = [##ano | ##anese | ##ana | ##a | ##ane]
Probability (%) = [92.9  | 3.6     | 2.7   | 0.3 | 0.1]
```

```
cambia il *[giorno]* *[della]* terza sve ##glia in elenco a sa ##bato
Top 5 predicted = [numero | tempo | giorno | calendario | nome]
Probability (%) = [53.4   | 14.8  | 11.4   | 4.6        | 2.3]
Top 5 predicted = [della | di   | da | dalla | con]
Probability (%) = [81.9  | 10.8 | 3  | 2.4   | 0.4]
```

```
*[che]* animale è la form ##ica
Top 5 predicted = [che | quale | dove | quanto | questo]
Probability (%) = [89  | 6.5   | 2.1  | 0.9    | 0.5]
```

```
da *[dove]* vien ##i tu
Top 5 predicted = [dove | onde | qui | cui | quando]
Probability (%) = [98.6 | 0.4  | 0.4 | 0.2 | 0.1]
```

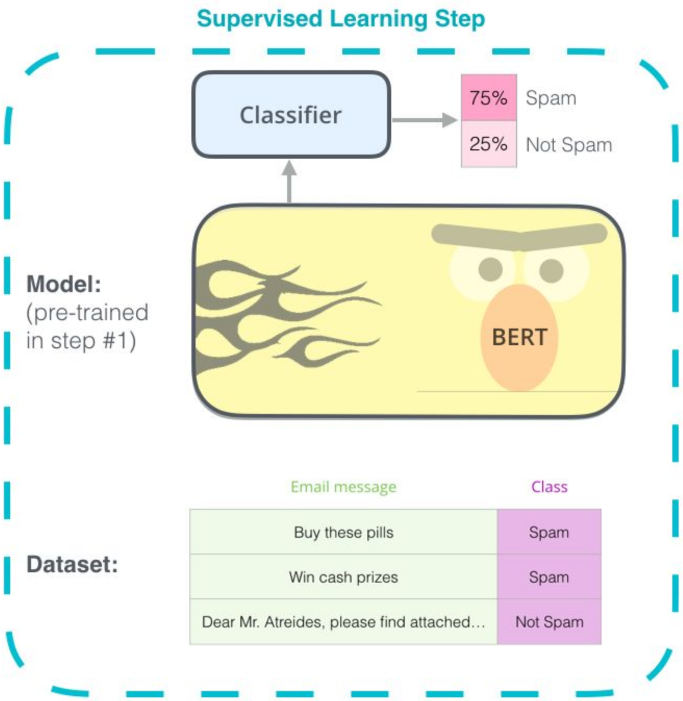
Downstream tasks with **BERT**

The idea behind BERT

<u>Goal of fine-tuning for downstream task</u>:

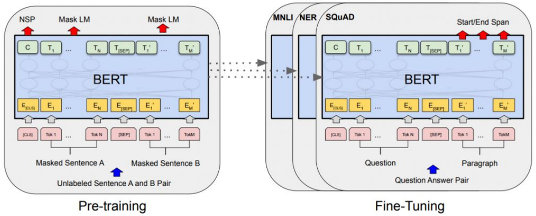
Fine-tune (train for - hopefully - much less epochs wrt pre-training) pre-trained BERT on a specific task

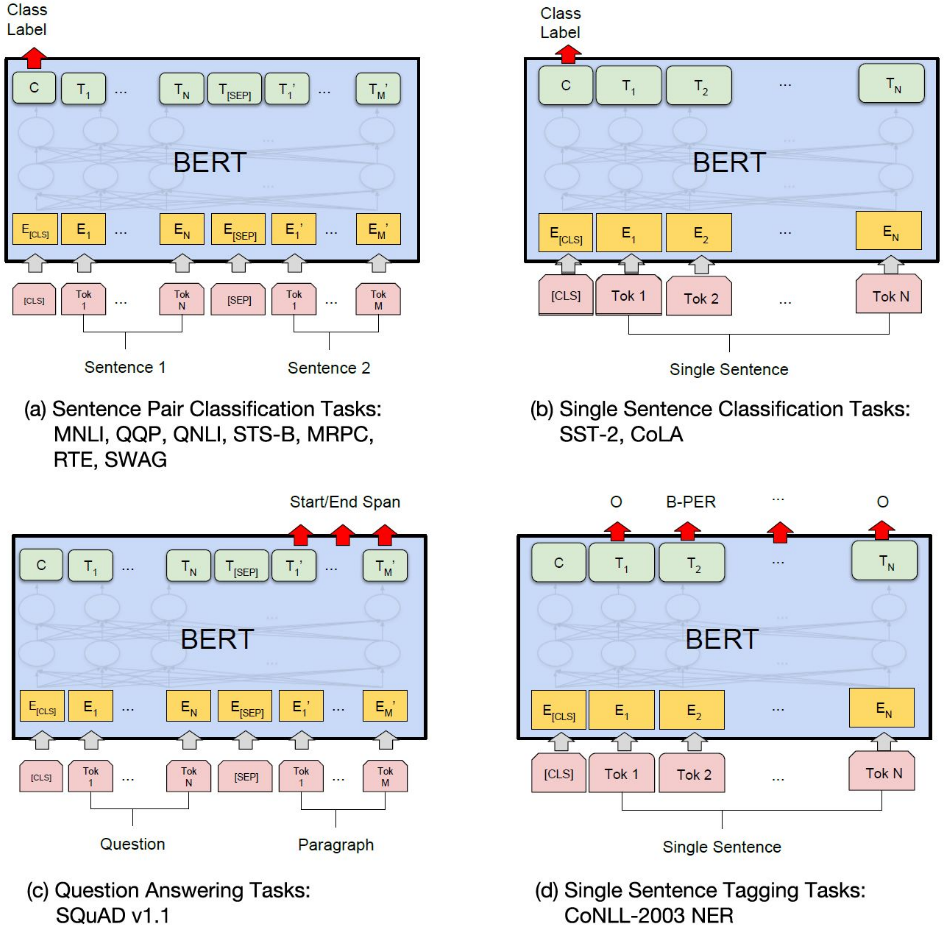
2 - Supervised training on a specific task with a labeled dataset.



Credit: http://jalammar.github.io/illustrated-bert/

Downstream tasks





Credit: BERT paper (arxiv/1810.04805)

Downstream tasks: GLUE results

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.9	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**

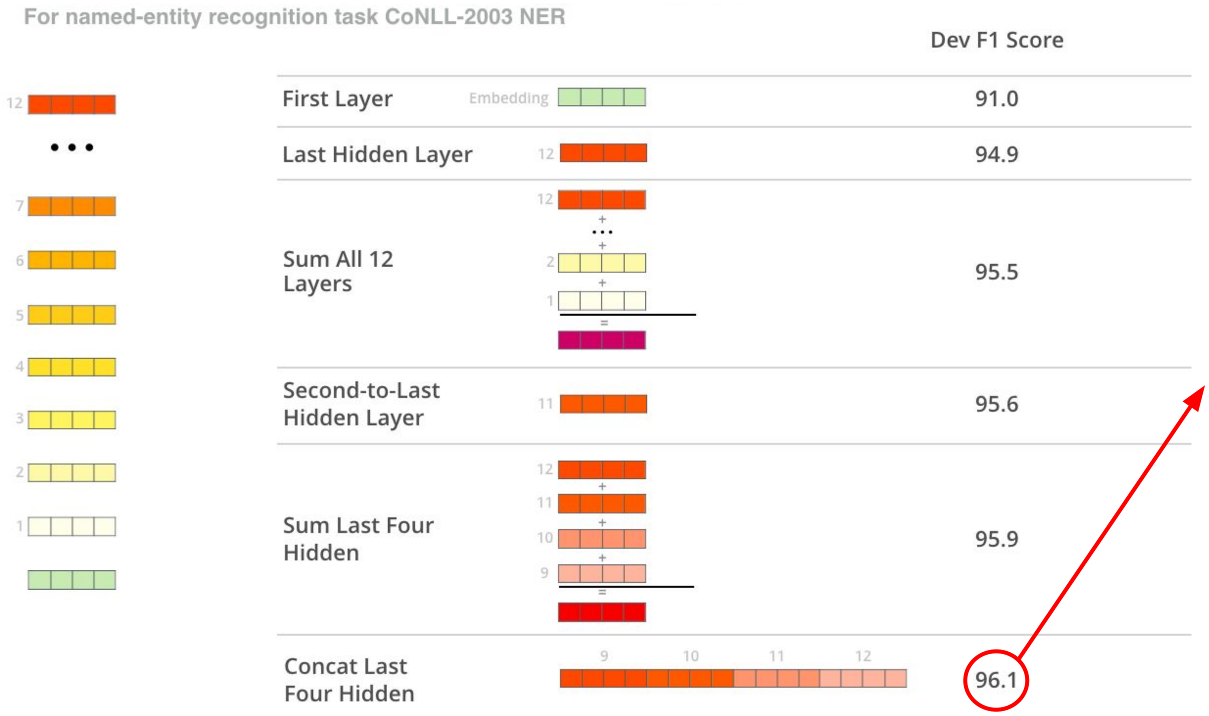
MNLI example

Premise: Hills and mountains are especially sanctified in Jainism.
Hypothesis: Jainism hates nature.
Label: Contradiction

Credit: https://nlp.stanford.edu/seminar/details/jdevlin.pdf

Feature-based approach with BERT

Instead of fine-tuning on downstream task, one can extract fixed features from pre-trained models and feed them to something different, e.g. to a BiLSTM



Only 0.3 F1 behind fine-tuning the entire model

NB: task-specific results

Credit: http://jalammar.github.io/illustrated-bert/

Adversarial attacks and Clever Hans effect

Adversarial attacks

Example: adversarial attack on Bi-directional Attention Flow (BiDAF) network QA model (from Jia & Liang, 2017)

Article: Super Bowl 50
Paragraph: *"Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager.*

Question: *"What is the name of the quarterback who was 38 in Super Bowl XXXIII?"*
Original Prediction: John Elway

Without the adversarial distracting sentence the model gets the correct answer.

When adding the blue sentence the model returns the wrong answer!

Hsieh et al., 2019: BERT and self-attentive architectures suffer from adversarial attacks but they are more robust than LSTM (although it depends on the adversarial attacks generation schemes)

Clever Hans effect

Niven & Kao, 2019: BERT on Argument Reasoning Comprehension Task

Claim	Take the umbrella
Reason	It's raining outside
Warrant	Being wet outside is bad for you
Alternative	Being wet outside is good for you

Given Claim and Reason pick Warrant over Alternative

$$\textbf{Reason (and since) Warrant} \rightarrow \textbf{Claim}$$
$$\textbf{Reason (but since) Alternative} \rightarrow \neg\,\textbf{Claim}$$

Result: BERT close to human performance!
~~"SOTA results! We are the best! Accept our paper!"~~

What is going on? What has BERT learned about argument comprehension?

It turns out that BERT didn't learn how to "understand". It simply took shortcuts predicting the correct label based on presence/absence of words like "not", "is", "do", etc.
Transforming the dataset into an adversarial dataset - where these shortcuts are not present - returns random performance

The end

Thank you!

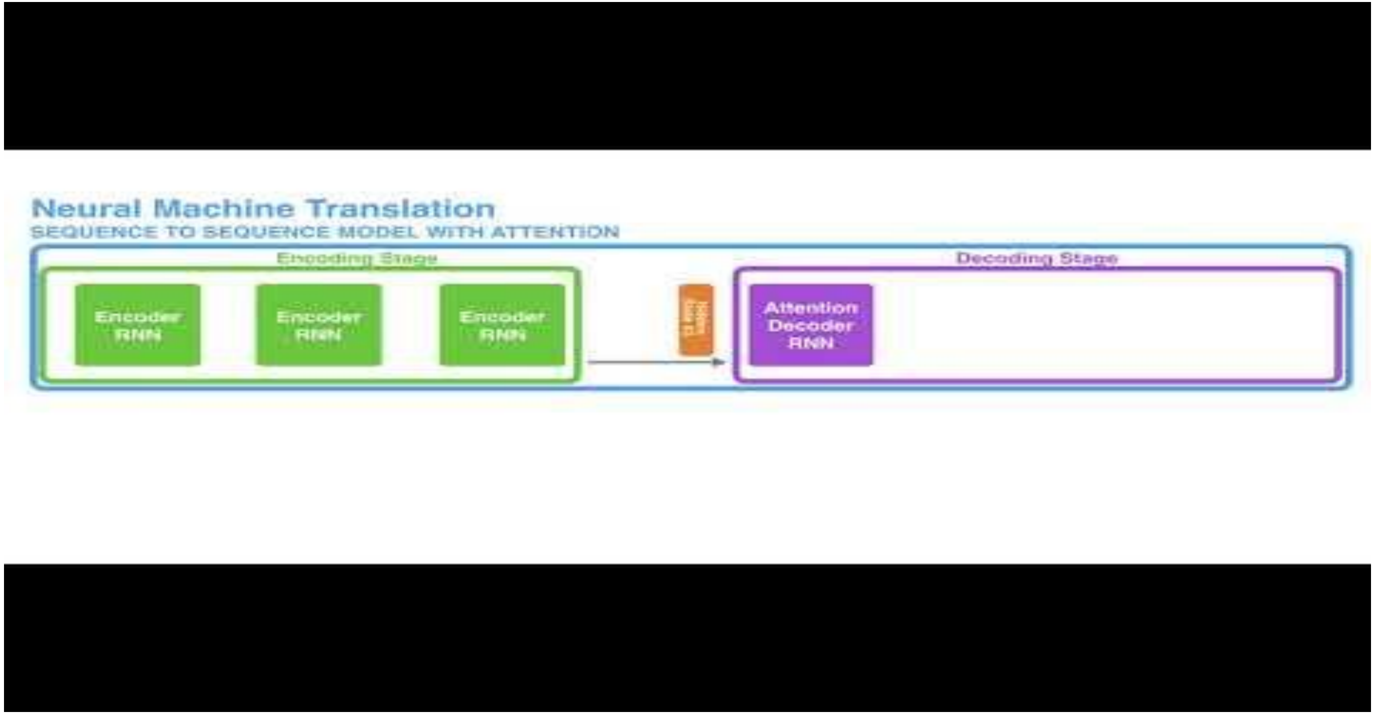
Some interesting links:

- https://nlp.stanford.edu/seminar/details/jdevlin.pdf
- https://towardsdatascience.com/transformers-141e32e69591
- https://medium.com/@mromerocalvo/dissecting-bert-part1-6dcf5360b07f
- https://lilianweng.github.io/lil-log/2019/01/31/generalized-language-models.html
- http://jalammar.github.io/illustrated-bert/
- http://exbert.net/
- https://mlexplained.com/2019/11/06/a-deep-dive-into-the-wonderful-world-of-preprocessing-in-nlp/
- https://thegradient.pub/nlps-clever-hans-moment-has-arrived/

Bonus slides

Bidirectional Encoder Representations from **Transformers**

From Recurrent Neural Networks…



Credit: https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

Multi-Head Self-Attention

$$\mathrm{MultiHead}(Q,K,V) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)W^O \qquad \mathrm{Attention}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$XW^K = K \qquad XW^Q = Q \qquad XW^V = V$$

$$\begin{array}{cc|c} & & K \\ & & \begin{array}{cccccc} \textit{Hello} & , & \textit{how} & \textit{are} & \textit{you} & ? \end{array} \\ Q & \begin{array}{c} \textit{Hello} \\ , \\ \textit{how} \\ \textit{are} \\ \textit{you} \\ ? \end{array} & \begin{pmatrix} 78.49 & 43.29 & 1.2 & 41.74 & 91.43 & 74.47 \\ 95.84 & 28.78 & 57.13 & 68.20 & -60.94 & 26.85 \\ -95.69 & -52.16 & 17.00 & 45.71 & 48.49 & 64.35 \\ -69.92 & 85.16 & 94.94 & 91.04 & -92.83 & 77.49 \\ 65.85 & 55.85 & 62.54 & -97.46 & 76.38 & 13.20 \\ -30.05 & -4.52 & 76.02 & 42.35 & 15.29 & 63.61 \end{pmatrix} \end{array} \xrightarrow[\text{element-wise scaling}]{\text{row-wise softmax}} \begin{array}{c|c} & \begin{array}{cccccc} \textit{Hello} & , & \textit{how} & \textit{are} & \textit{you} & ? \end{array} \\ \begin{array}{c} \textit{Hello} \\ , \\ \textit{how} \\ \textit{are} \\ \textit{you} \\ ? \end{array} & \begin{pmatrix} 0.1 & 0 & 0.06 & 0.1 & 0.6 & 0.14 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix} \end{array}$$

NB: not symmetric

NB: these have multiple columns

$$\begin{array}{cc|c} & & K \\ & & \begin{array}{cccccc} \textit{Hello} & , & \textit{how} & \textit{are} & \textit{you} & ? \end{array} \\ Q & \begin{array}{c} \textit{Hello} \\ , \\ \textit{how} \\ \textit{are} \\ \textit{you} \\ ? \end{array} & \begin{pmatrix} 0.1 & 0 & 0.06 & 0.1 & 0.6 & 0.14 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix} \end{array} \begin{array}{c} V \\ \begin{pmatrix} v_{Hello} \\ v_{,} \\ v_{how} \\ v_{are} \\ v_{you} \\ v_{?} \end{pmatrix} \end{array} = \begin{array}{c} \textit{Hello} \\ , \\ \textit{how} \\ \textit{are} \\ \textit{you} \\ ? \end{array} \begin{pmatrix} 0.1v_{Hello} + 0v_{,} + 0.06v_{how} + 0.1v_{are} + 0.6v_{you} + 0.14v_{?} \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ \cdots\cdots\cdots\cdots\cdots\cdots \end{pmatrix}$$

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

1. Build vocabulary/tokenizer

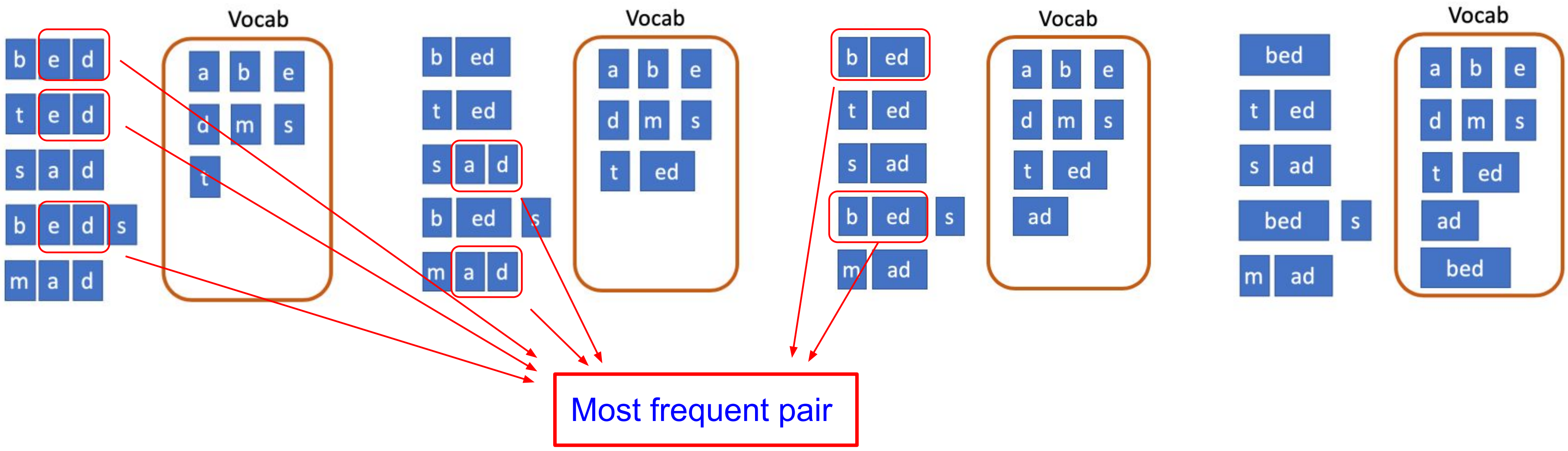
BERT uses WordPiece tokenization
How does it work? Let’s talk about Byte Pair Encoding (BPE) tokenization first

BPE:

1. Get corpus
2. Set vocabulary maximum size
3. Split text at character level (tokens = characters)
4. Add tokens to vocabulary and tokenize text based on vocabulary
5. Merge token pair with highest frequency within text
6. End if vocabulary maximum size is reached, otherwise go to 4.

1. Build vocabulary/tokenizer

BPE example:
Corpus: “bed”, “ted”, “sad”, “beds”, “mad”
Vocabulary size: 10



Credit: https://mlexplained.com/2019/11/06/a-deep-dive-into-the-wonderful-world-of-preprocessing-in-nlp/

1. Build vocabulary/tokenizer

WordPiece tokenization works like BPE but instead of picking the most frequent token pair it builds an n-gram Language Model and selects the pair that minimizes the cross-entropy loss/perplexity

WordPiece:

1. Get corpus
2. Set vocabulary maximum size
3. Split text at character level (tokens = characters)
4. Add tokens to vocabulary and tokenize text based on vocabulary
5. Build an n-gram language model on the corpus based on new vocabulary
6. Merge token pair that minimizes loss of language model
7. End if vocabulary maximum size is reached, otherwise go to 4.

3. Masked Language Model

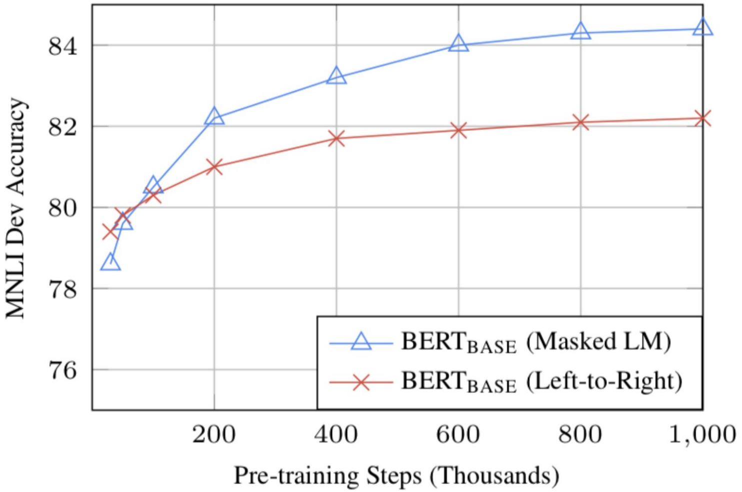
Masked Language Modelling allows deep bidirectionality but:

1) It creates a mismatch between pre-training and downstream tasks since [MASK] token is never seen during fine-tuning for downstream tasks

2) It doesn't take into account dependence between masked tokens (all masked tokens within a sentence are predicted simultaneously → autoencoding vs autoregressive approach)

3) It doesn't process all possible word-context combinations

4) It predicts only 15% of tokens → slower training times



3. Masked Language Model

Problem: Masked Language Model creates a mismatch between pre-training and downstream tasks since [MASK] token is never seen during fine-tuning for downstream tasks

Solution: out of the 15% selected tokens, don't use [MASK] 100% of the time. Instead:

- 80% of the time use [MASK] token

```
['[CLS]', 'The', 'fox', 'is', 'brown', '[SEP]']
['[CLS]', 'The', 'fox', '[MASK]', 'brown', '[SEP]']
```

- 10% of the time use random token

```
['[CLS]', 'The', 'fox', 'is', 'brown', '[SEP]']
['[CLS]', 'The', 'fox', 'chair', 'brown', '[SEP]']
```

- 10% of the time use original token

```
['[CLS]', 'The', 'fox', 'is', 'brown', '[SEP]']
['[CLS]', 'The', 'fox', 'is', 'brown', '[SEP]']
```

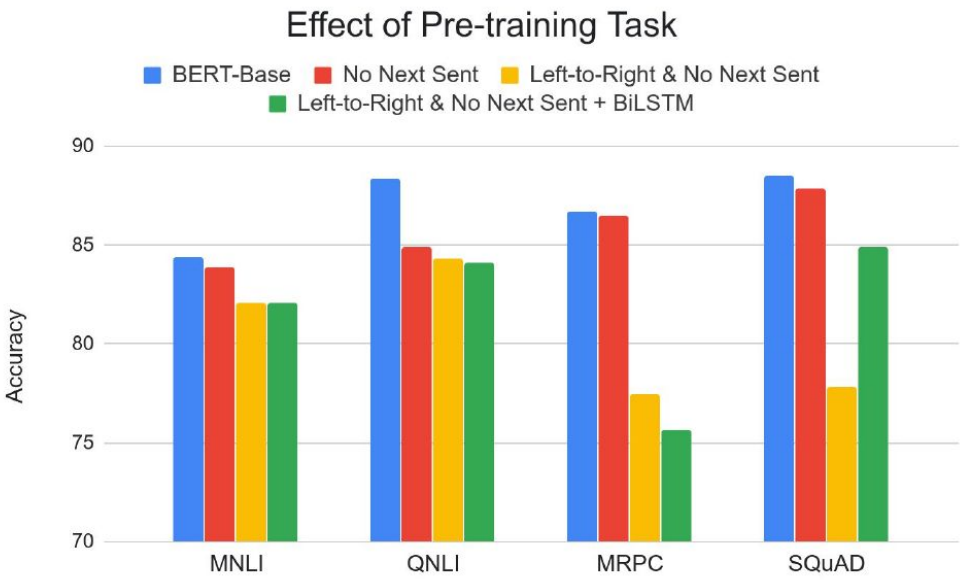
Pre-training bonus: meet RoBERTa

RoBERTa = A Robustly Optimized BERT Pretraining Approach (arxiv/1907.11692, Jul. 2019)
Authors state that BERT was significantly undertrained. They changed the pre-training procedure by:

- Training the model longer
- Increasing the batch size (256 -> 8000 sentences per batch)
- Increasing data 16 GB -> 160 GB of text (adding CC-NEWS+OPENWEBTEXT+STORIES)
- Removing next sentence prediction task, hence being able to train on longer sequences (since T_{max} = 512 is the same as BERT but without two contiguous sentences)
- Switching from static to dynamic masking for masked language model task
- Using a larger vocabulary of 50k subword units as opposed to BERT 30k units

Model	data	bsz	steps	SQuAD (v1.1/2.0)	MNLI-m	SST-2
RoBERTa						
with BOOKS + WIKI	16GB	8K	100K	93.6/87.3	89.0	95.3
+ additional data (§3.2)	160GB	8K	100K	94.0/87.7	89.3	95.6
+ pretrain longer	160GB	8K	300K	94.4/88.7	90.0	96.1
+ pretrain even longer	160GB	8K	500K	**94.6/89.4**	**90.2**	**96.4**
$\text{BERT}_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	90.9/81.8	86.6	93.7
$\text{XLNet}_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	94.0/87.8	88.4	94.4
+ additional data	126GB	2K	500K	94.5/88.8	89.8	95.6

Downstream tasks: effect of pre-training



- 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 BiLSTM

Credit: https://nlp.stanford.edu/seminar/details/jdevlin.pdf