HOCHSCHULE FÜR TECHNIK UND WIRTSCHAFT DRESDEN
UNIVERSITY OF APPLIED SCIENCES

Deep NLP 2: Word Vectors and Transfer Learning

Oliver Guhr

Quiz Time!

How is the pace of this course?

(to slow, just right, to fast)

A RNN is a network with?

A) holes
B) convolutions
C) skip connections
D) loops

A RNN works by:

A) passing information from one layer to the other

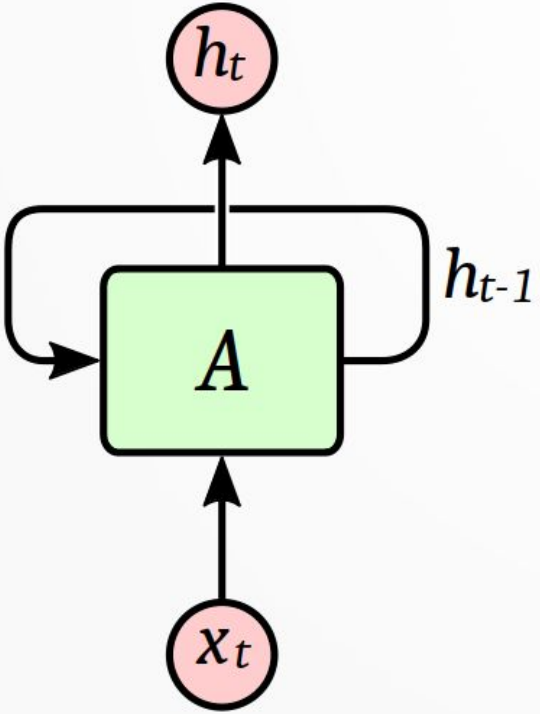
B) passing information from previous time steps

The vanishing gradient problem can be solved by:

A) unfolding the network
B) using backpropagation through time
C) clipping gradients
D) limiting the number of time steps

Recap

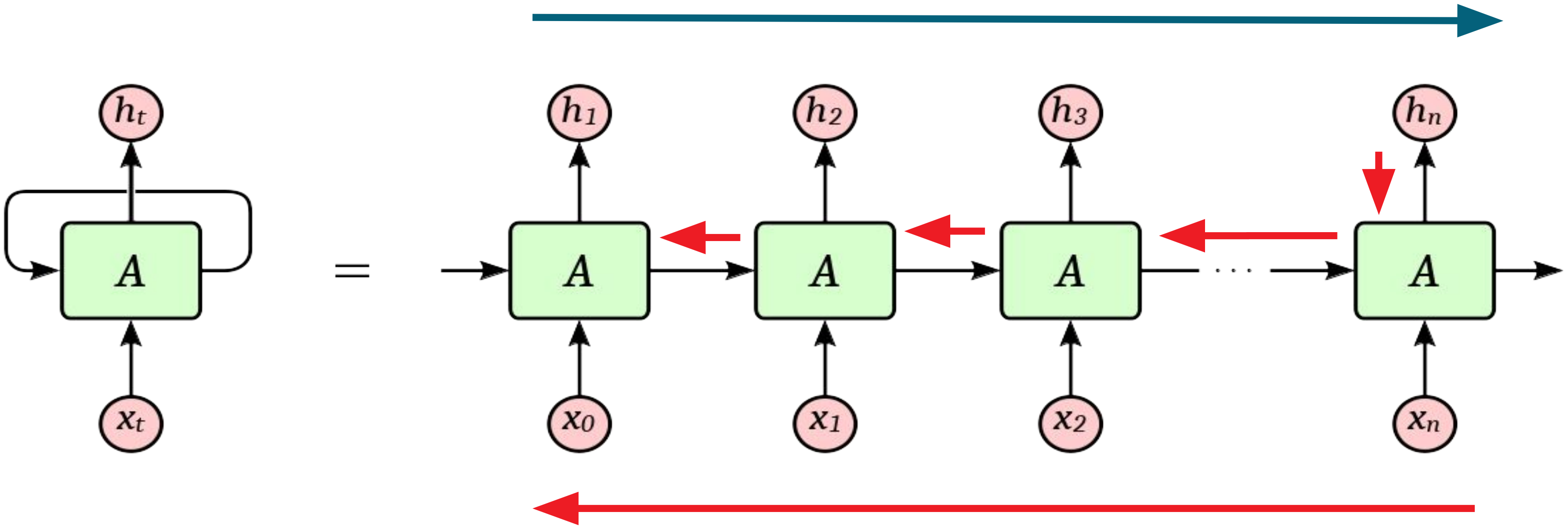
Recurrent Neural Networks



$$\underbrace{h_t}_{\text{new state}} = \underbrace{A}_{\text{network function}}\left(\underbrace{h_{t-1}}_{\text{previous state}}, \underbrace{x_t}_{\text{input vector}}\right)$$

Training RNNs

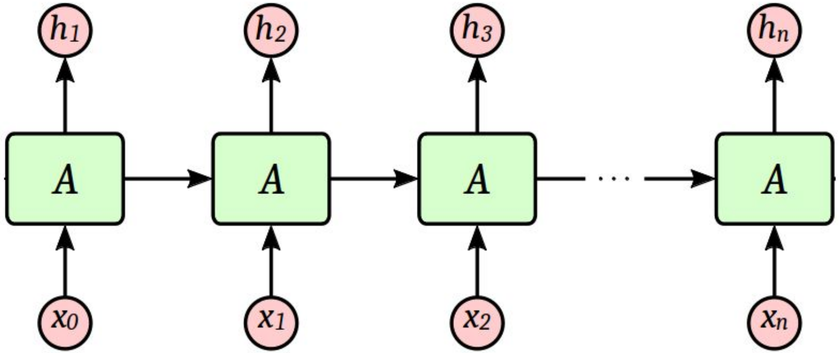
1. forward-propagate the inputs over the unfolded network



2. back-propagate the error, back across the unfolded network

3. sum the weight changes and update all weights

The vanishing gradient problem



$$h_1 = \tanh(W_{hh} h_0 + W_{xh} x_1)$$

$$h_2 = \tanh(W_{hh}(\tanh(W_{hh} h_0 + W_{xh} x_1)) + W_{xh} x_2)$$

$$h_3 = \tanh(W_{hh}(\tanh(W_{hh}(\tanh(W_{hh} h_0 + W_{xh} x_1)) + W_{xh} x_2)) + W_{xh} x_3)$$

$$h_4 = \underline{\tanh(W_{hh}(\tanh(W_{hh}(\tanh(W_{hh}(\tanh(W_{hh}}\, h_0 + W_{xh} x_1)) + W_{xh} x_2)) + W_{xh} x_3))$$

Backpropagating this recursive function leads to exploding or vanishing gradients.

One Hot Encoding

Let’s encode the word „hello“

h	0	0	0	1
e	0	0	1	0
l	0	1	0	0
o	1	0	0	0

$$v^{h} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad v^{e} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \dots \longrightarrow \quad V^{hello} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

How can neural networks process texts?

One Hot Encoding

Let's encode the word „hello“

h	0	0	0	1
e	0	0	1	0
l	0	1	0	0
o	1	0	0	0

$$v^h = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad v^e = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \ldots \longrightarrow \quad V^{hello} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

One Hot Encoding

Let’s encode the word „hello“

h	0	0	0	1
e	0	0	1	0
l	0	1	0	0
o	1	0	0	0

Let’s encode a sentence!

hello	0	0	0	1
deep	0	0	1	0
learning	0	1	0	0
students	1	0	0	0

Bag-Of-Words (BOW)

- You can‘t encode words that are not in your vocabulary.
- Size of the matrix is n x n, where n is the size of your vocabulary
- The German language has an estimated number of 5,3 million words[1]. We can‘t handle such matrices.
- Since they are sparse matrices most of the entries will be zero. (inefficient)

[1] Wolfgang Klein, Page 34 http://pubman.mpdl.mpg.de/pubman/item/escidoc:1850493:4/component/escidoc: 1850492/ReichtumundArmut.pdf

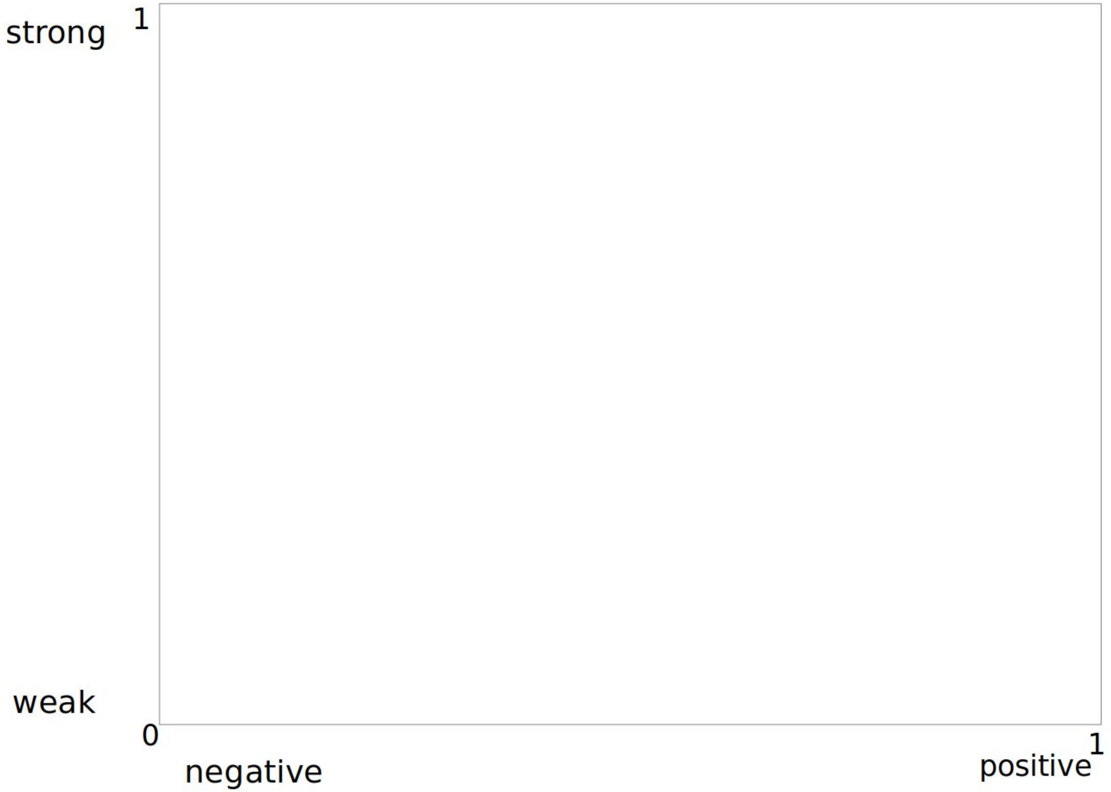
How can we efficiently encode words?

What is a vector again?

- A sequence of numbers that is used to identify a point in space is called a vector.

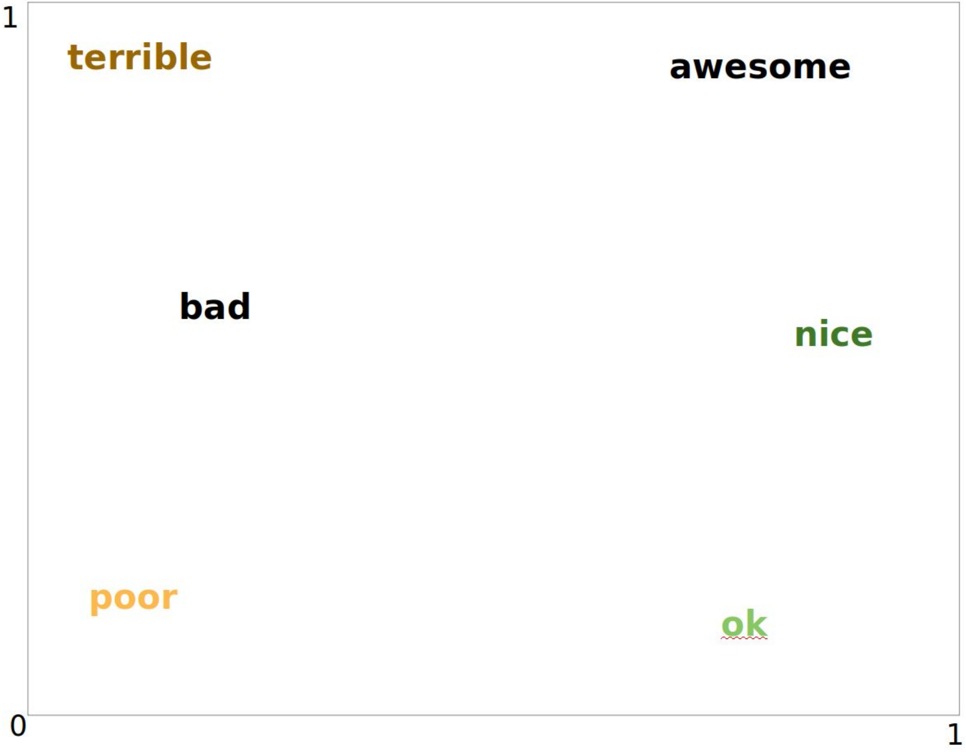
- A list of vectors that belong to the same data set, is called a vector space.

Word Vectors



Word Vectors

Word Vectors



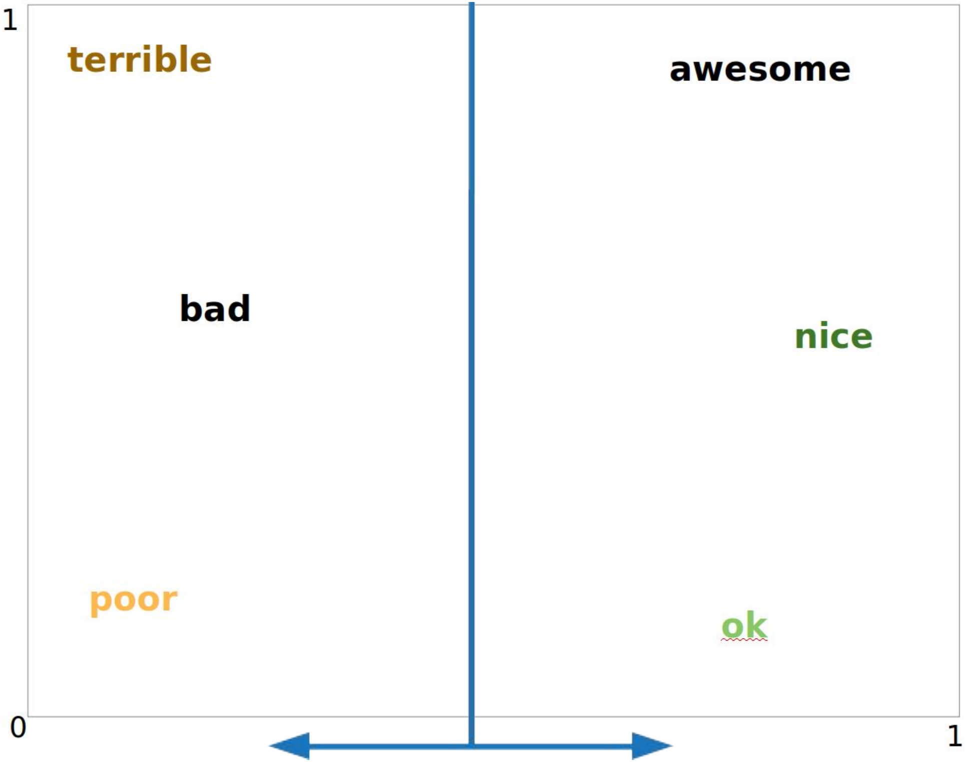
$$v_{ok} = [0.75, 0.15]$$

$$v_{nice} = [0.85, 0.50]$$

$$v_{poor} = [0.15, 0.18]$$

$$v_{terrible} = [0.10, 0.91]$$

Word Vectors - Klassifikation



$v_{ok} = [0.75, 0.15]$

$v_{nice} = [0.85, 0.50]$

$v_{poor} = [0.15, 0.18]$

$v_{terrible} = [0.10, 0.91]$

Distance and Similarity

Since our words are now vectors, we can use the euclidean distance to calculate similarity of two words.

$$||v_{nice} - v_{ok}|| = 0.364$$

$$||v_{terrible} - v_{ok}|| = 1$$

How can we create these “maps”?

What is Unsupervised Learning

- Idea: Use existing texts to train a model.
- Formal: “type of algorithm that learns patterns from unlabeled data”
- Goal: Teach the model “how language works”
- find a task that you can apply on unlabeled text
- one of the first successful models was Word2Vec
  - Efficient Estimation of Word Representations in Vector Space by Mikolov et al. 2013

Distributional Hypothesis

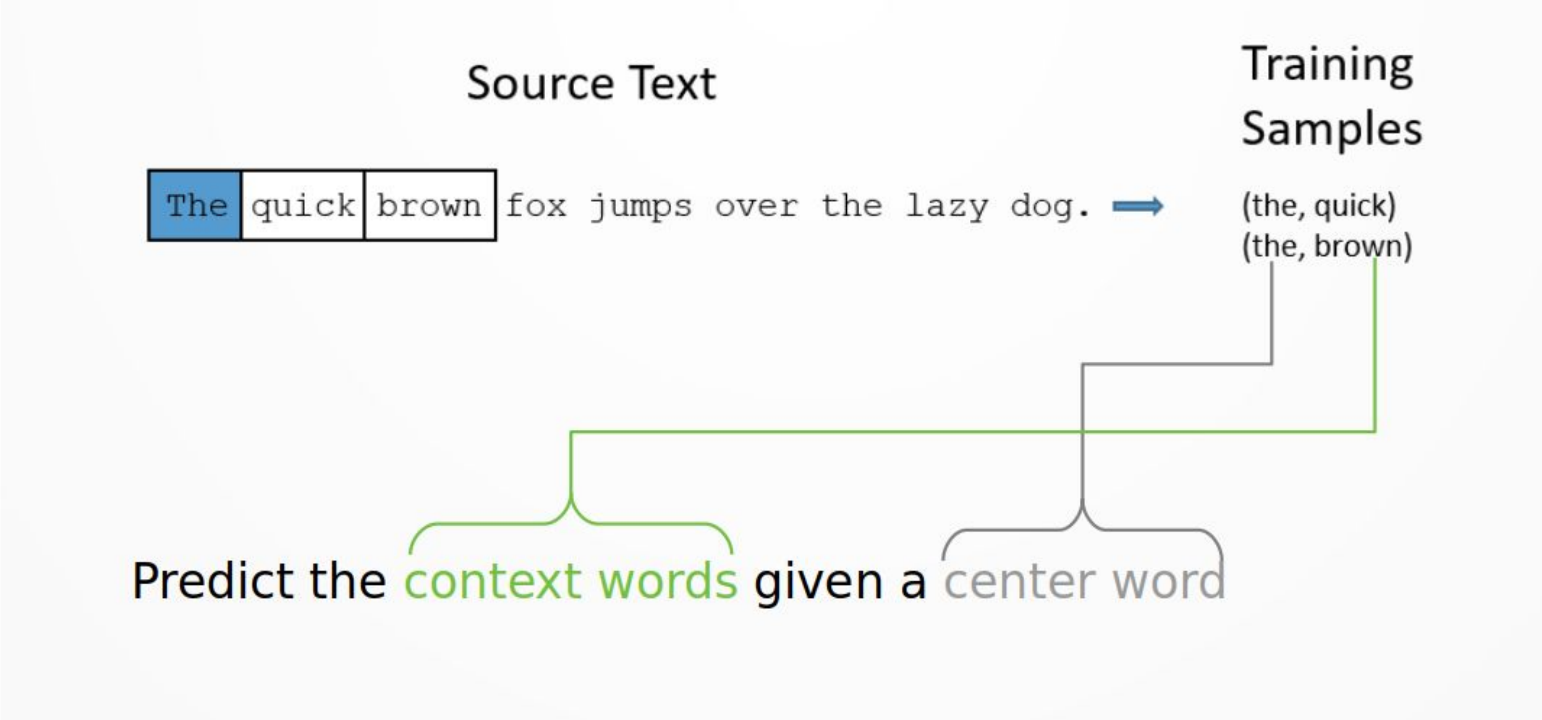
Words that occur in the same contexts tend to have similar meanings.
Harris (1954)

A word is characterized by the company it keeps.
Firth (1957)

Skip Gram

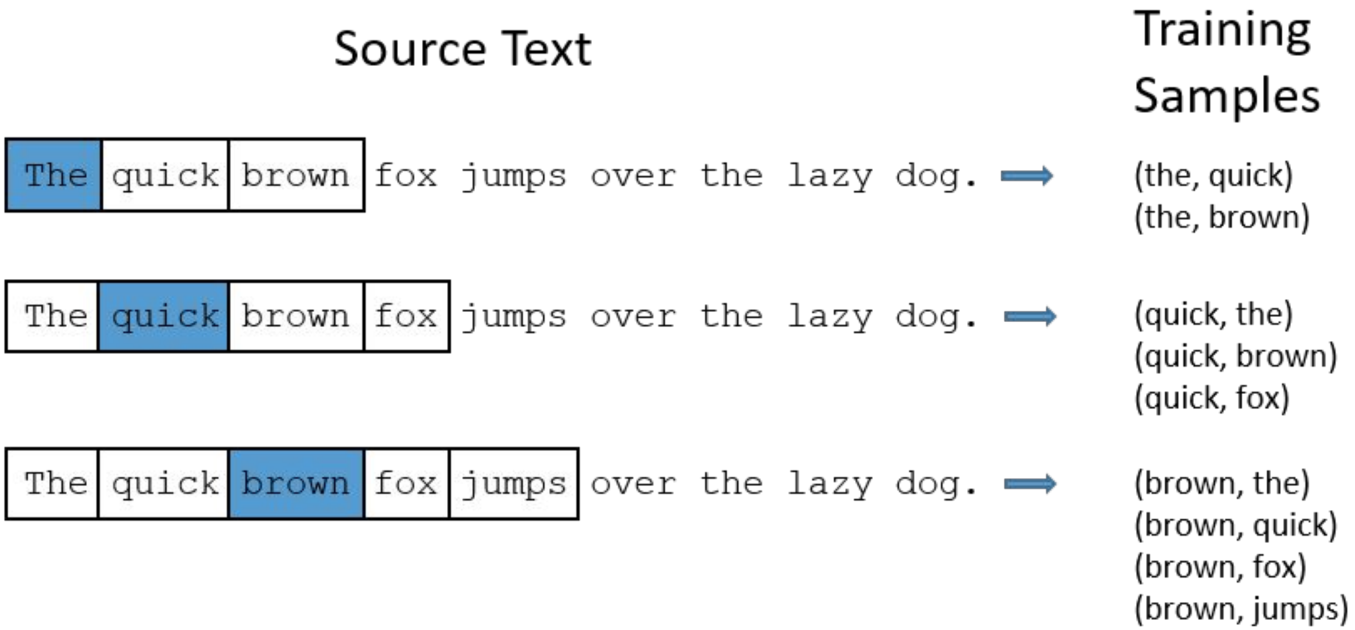
Efficient Estimation of Word Representations in Vector Space, Mikolov et al., 2013

Skip-Gram



McCormick, Word2Vec Tutorial, http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

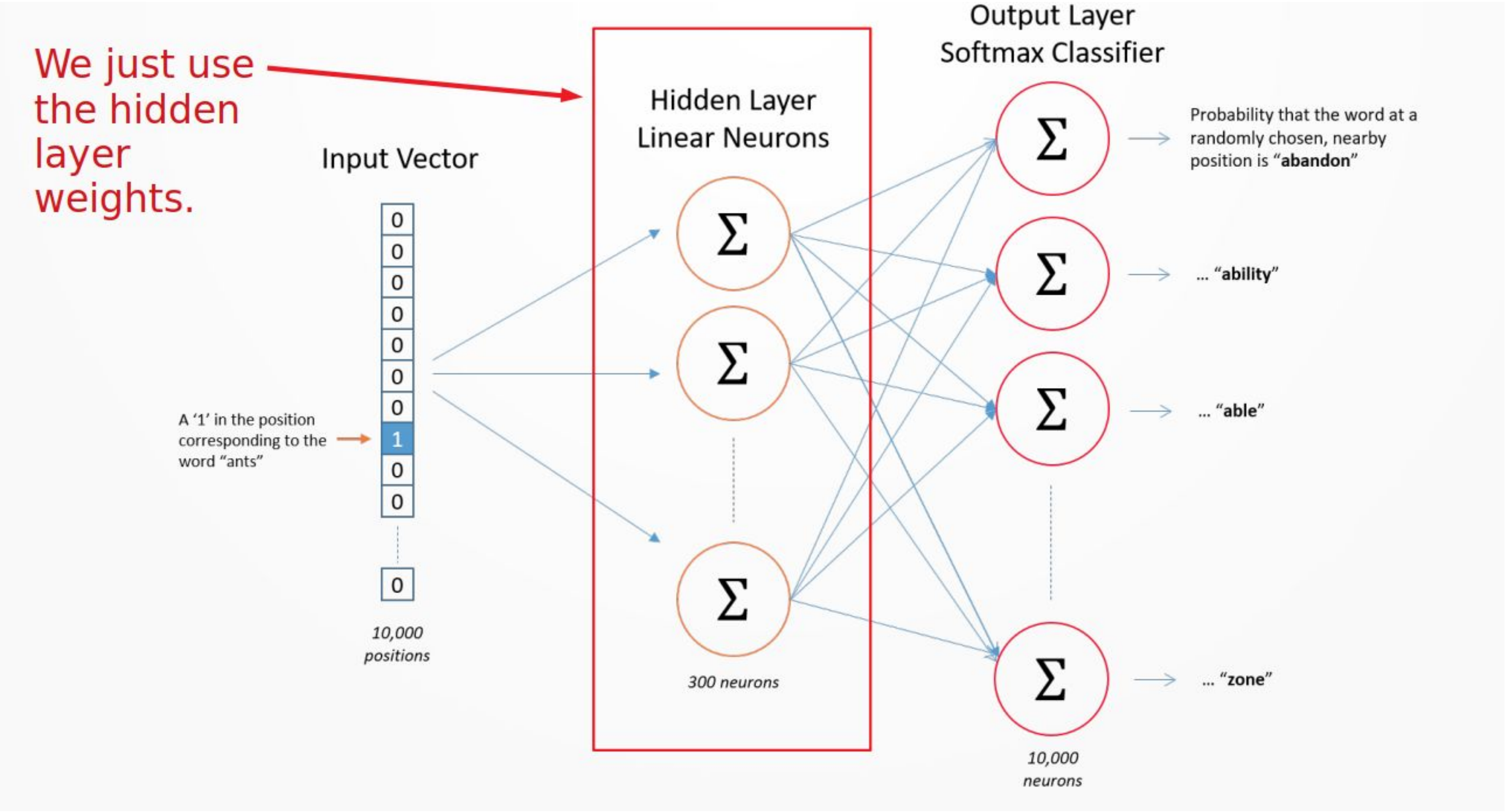
Skip-Gram



We are using a window size of 2. Meaning: two words behind and 2 words ahead of the center word.

McCormick, Word2Vec Tutorial, http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

Skip-Gram



McCormick, Word2Vec Tutorial, http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

Skip-Gram

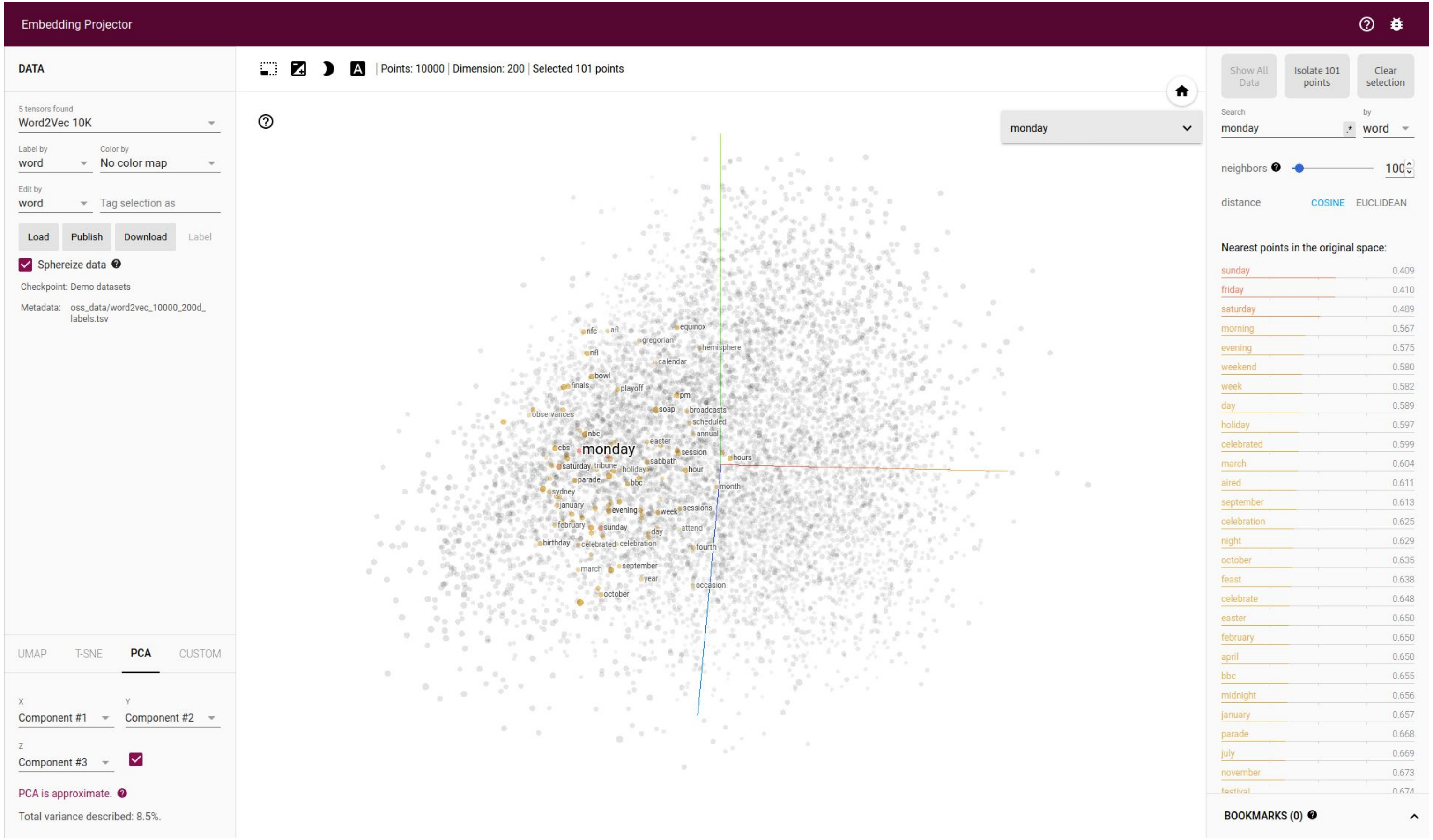
- For words with similar contexts, our model needs to produce a similar result. This will motivate the model to learn similar weights, that we use as words vectors.

- This way we „compressed“ our 1 x 10000 sparse one-hot vector to a 1 x 300 dense vector.

- **<u>We can now reuse this pretrained vectors in other models.</u>**

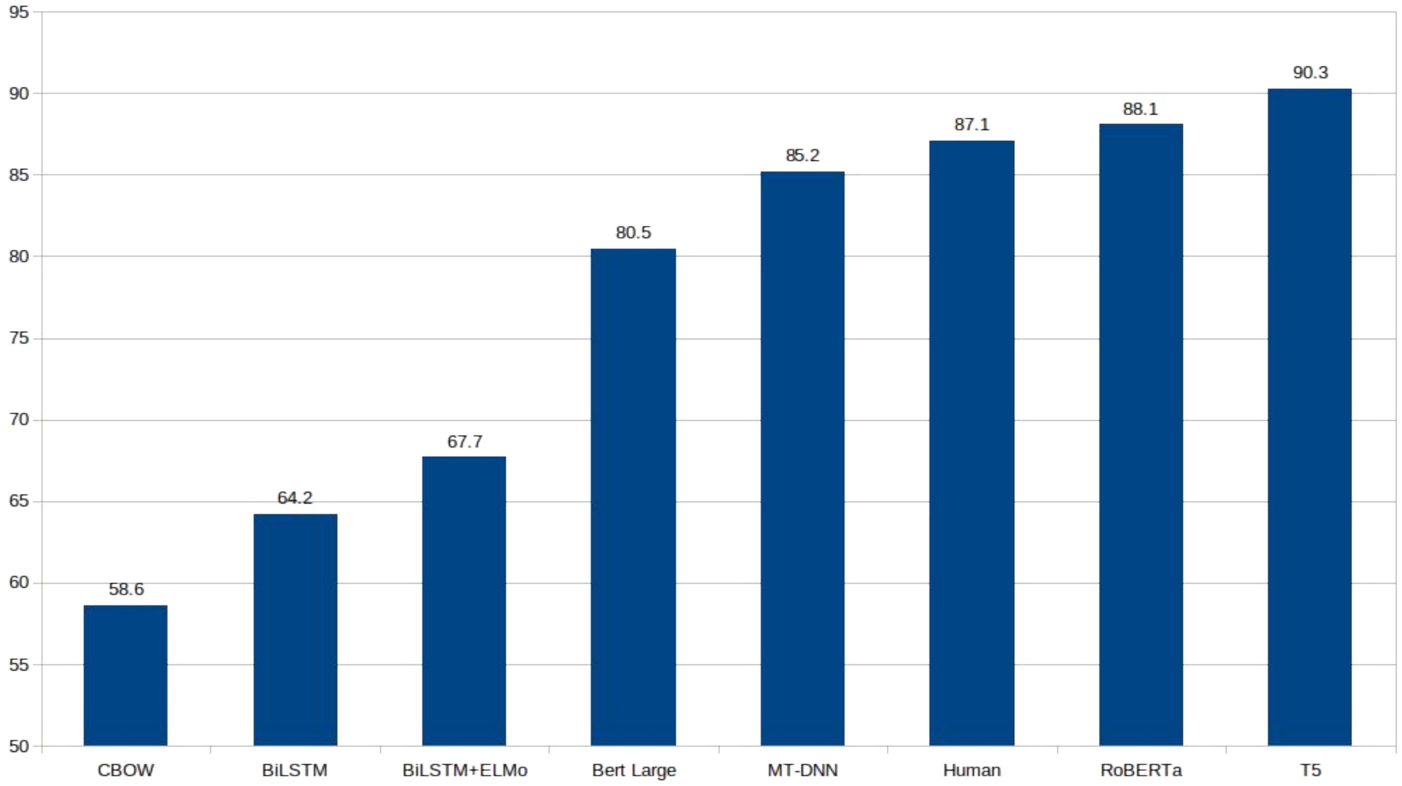
- More details on this:
  - http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

Word Vectors



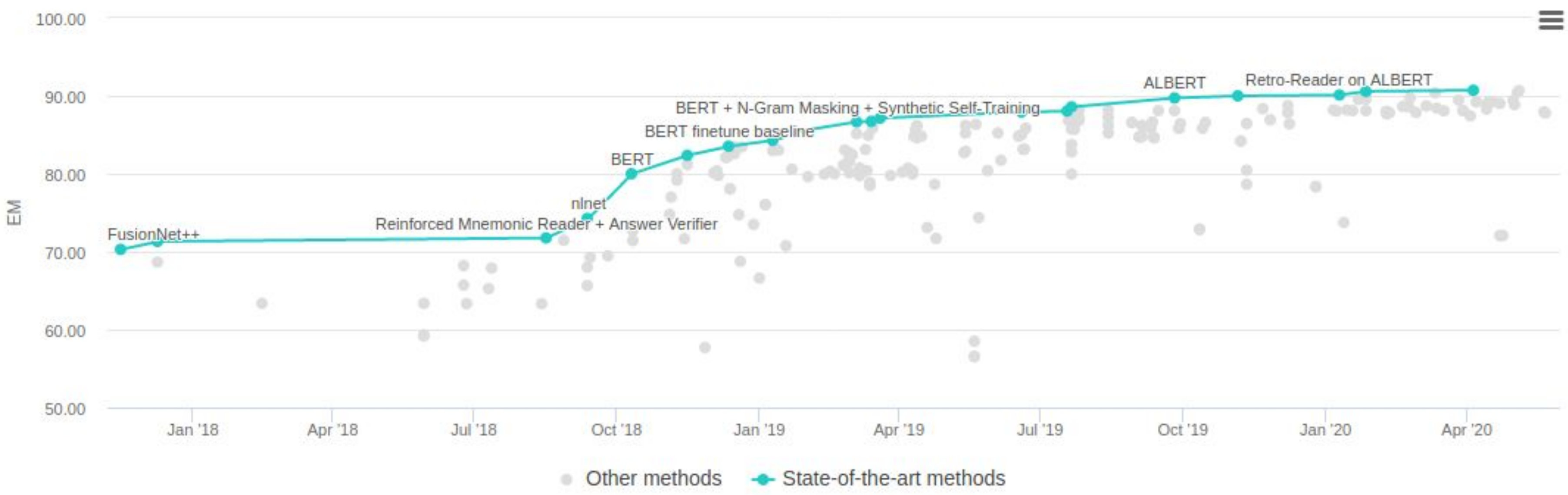
Deep Natural Language Processing

GLUE Benchmark



GLUE Leaderboard: https://gluebenchmark.com/leaderboard

Question Answering SQUAD 2.0



Source: https://paperswithcode.com/sota/question-answering-on-squad20

Deep Language Models

In 2018 several Ideas led to new models

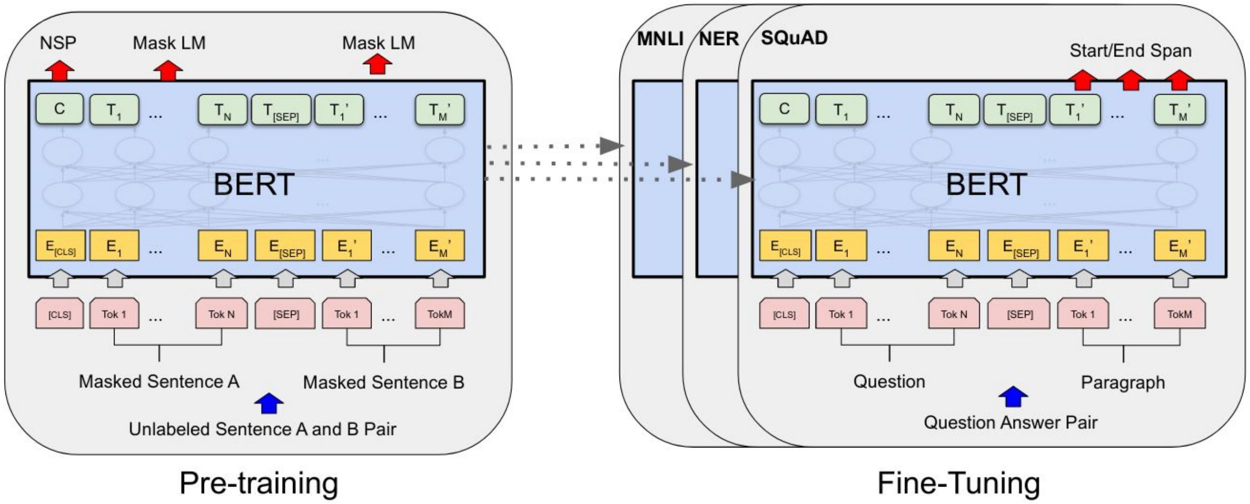
- Semi-supervised Sequence Learning Andrew Dai, Quoc Le
- ELMo Peters et al.
- ULMFiT Howard, Ruder
- OpenAI Transformer Radford, Narasimhan, Salimans, Sutskever
- Transformer Vaswani et al.
- GTP / GTP2 Radford et al.

BERT

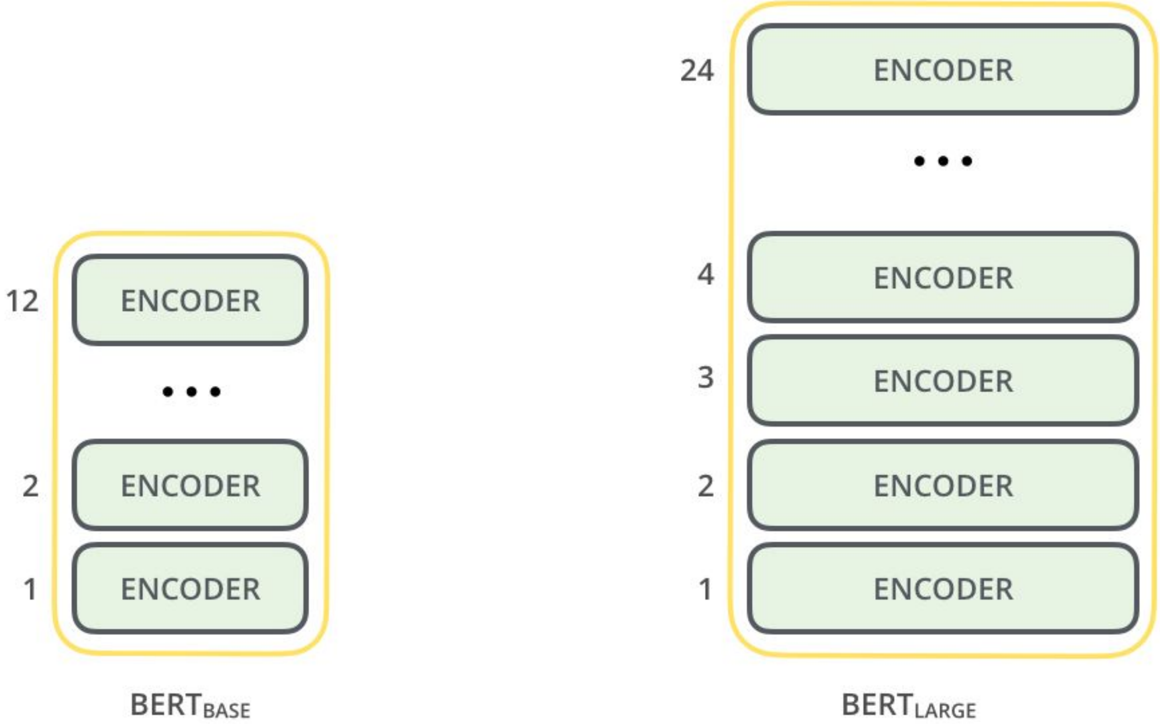
Bidirectional
Encoder
Representations from
Transformers

Bert

- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
- Paper by Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova
- Published in 2018
- improved the state-of-the-art in most important benchmarks



Bert


12
ENCODER
2
ENCODER
1
ENCODER
BERTBASE
24
ENCODER
4
ENCODER
3
ENCODER
2
ENCODER
1
ENCODER
BERTLARGE

(some) Bert Models

- English
  - 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
- Multi Language
  - 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
- Chinese
  - BERT-Base, Chinese: Chinese Simplified and Traditional, 12-layer, 768-hidden, 12-heads, 110M parameters

Transfer learning with texts

- Training Process
  - Pre-train a model on plain text
  - Choose a task specific labeled data set
  - Retrain the model with this data set

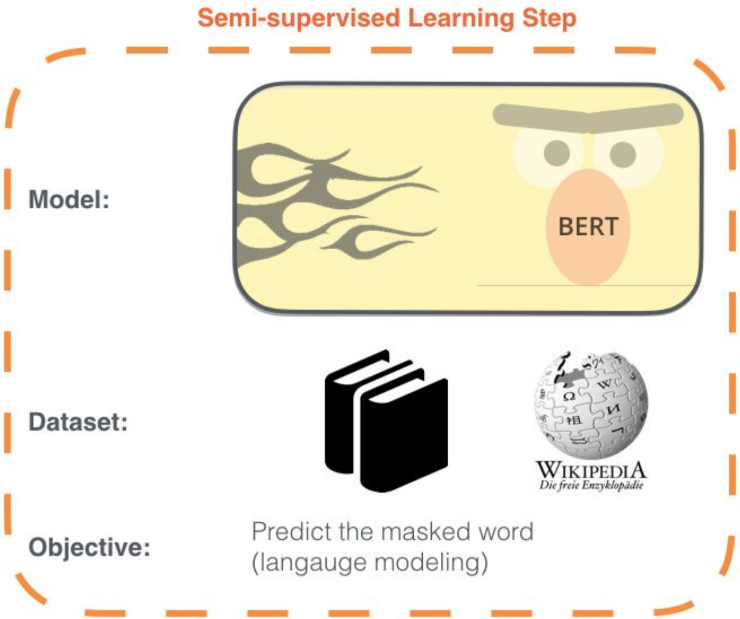
- Use the same pre-trained model for all tasks
  - Classification
  - Named Entity Recognition
  - Question Answering etc.

Step One: Pretraining

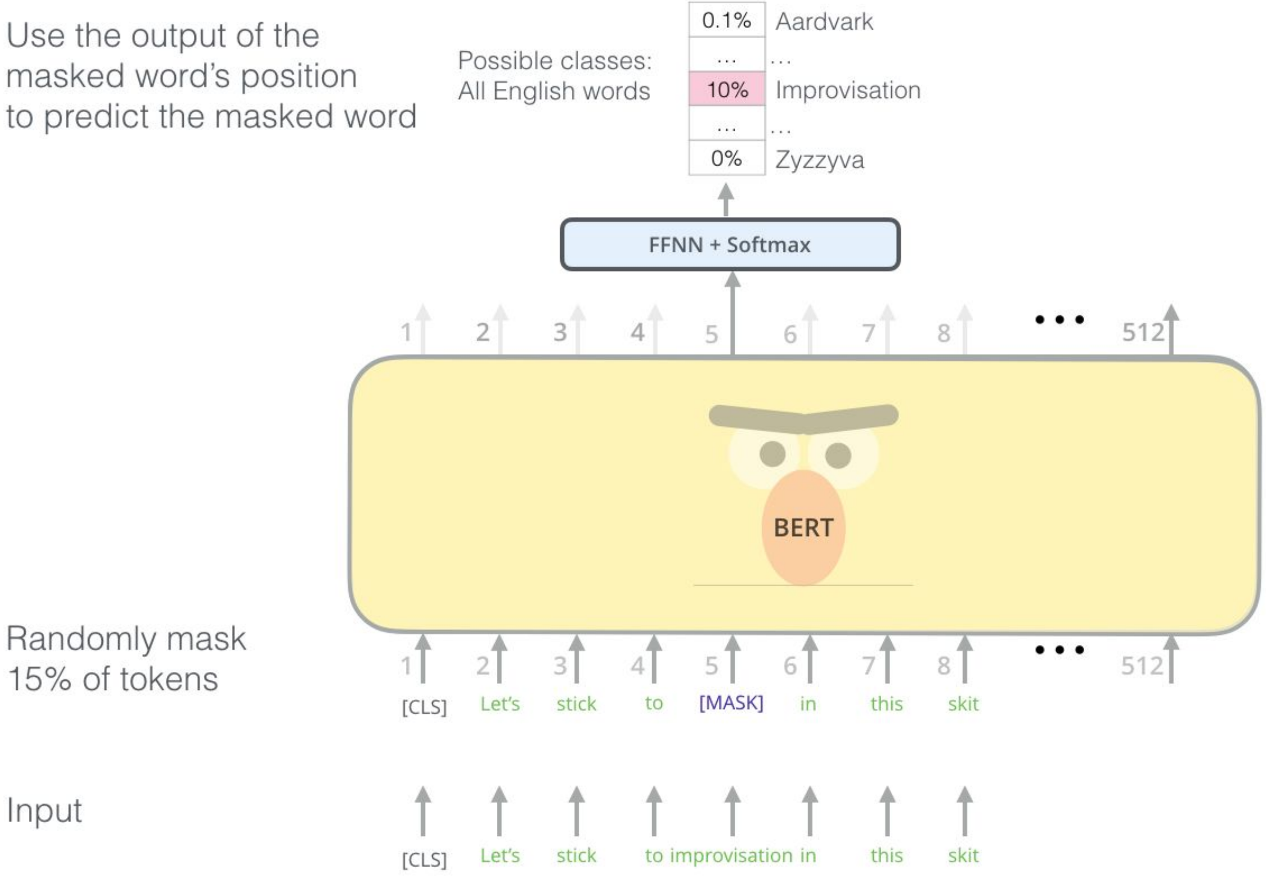
Bert

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

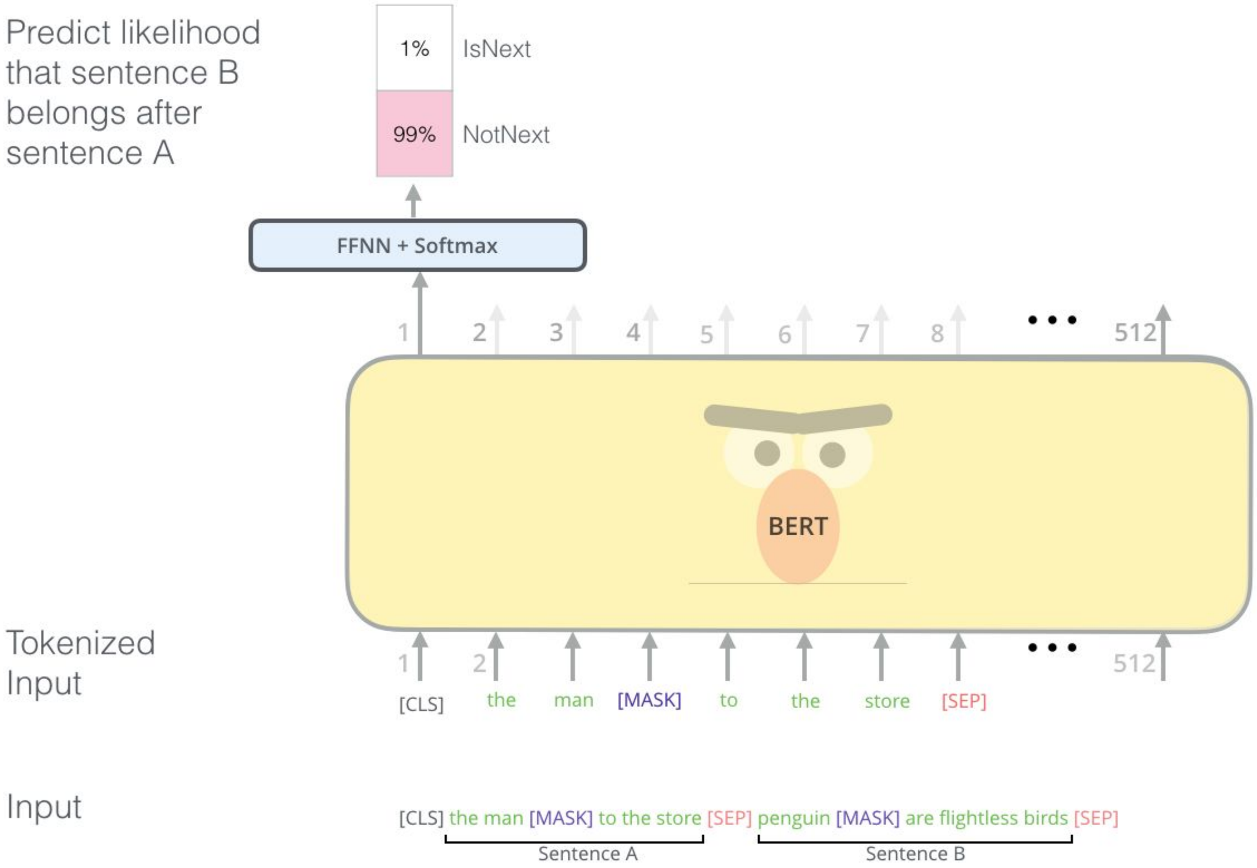
The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.



Task One: Mask Words



Task Two: Next Sentence Prediction



Semi-supervised training

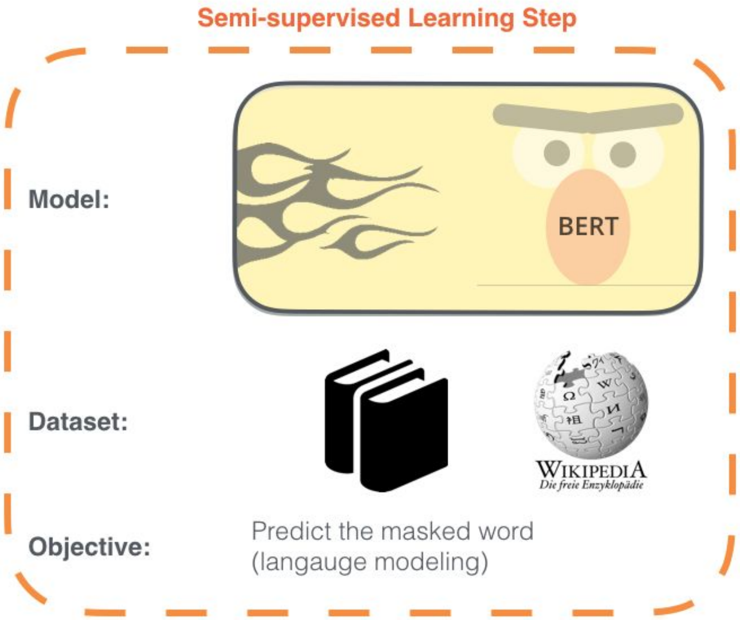
- pre trained models are also called **language models**
- To create them, Bert used two methods:
  - Task One: Mask Words
  - Task Two: Next Sentence Prediction
- Pre Training takes 14 days on a TPUv2 (500$)
- Fine-tuning a model with 1GB of text takes several hours on a single GPU (1080 / 2080)

Step Two: Fine Tuning

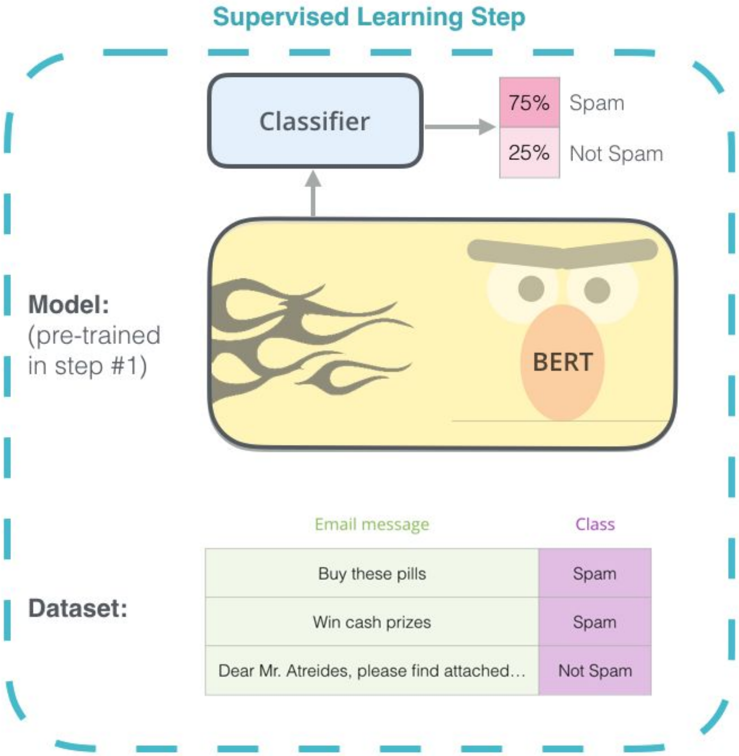
Bert

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

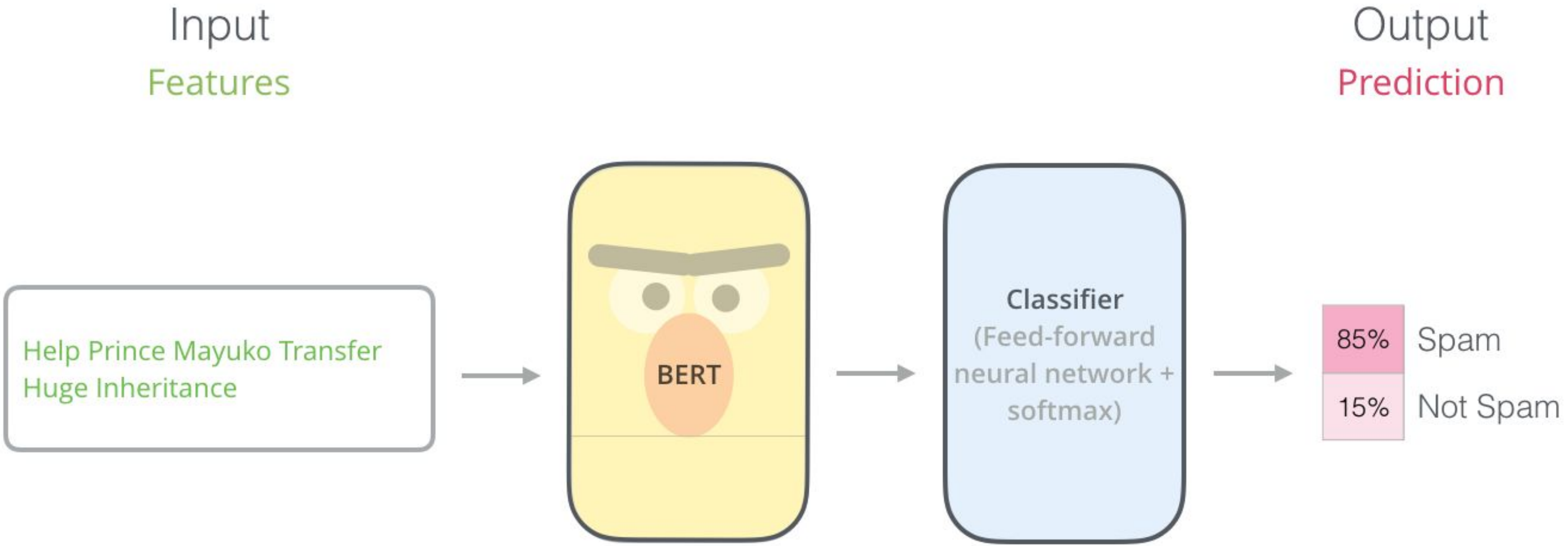
The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.



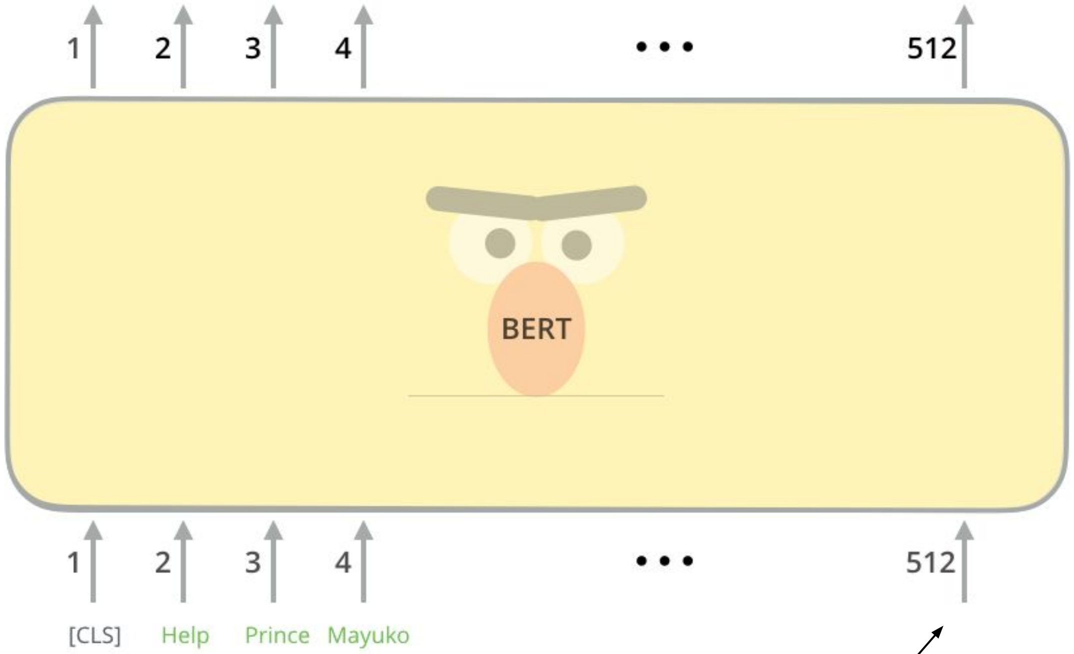
2 - Supervised training on a specific task with a labeled dataset.



Text Classification



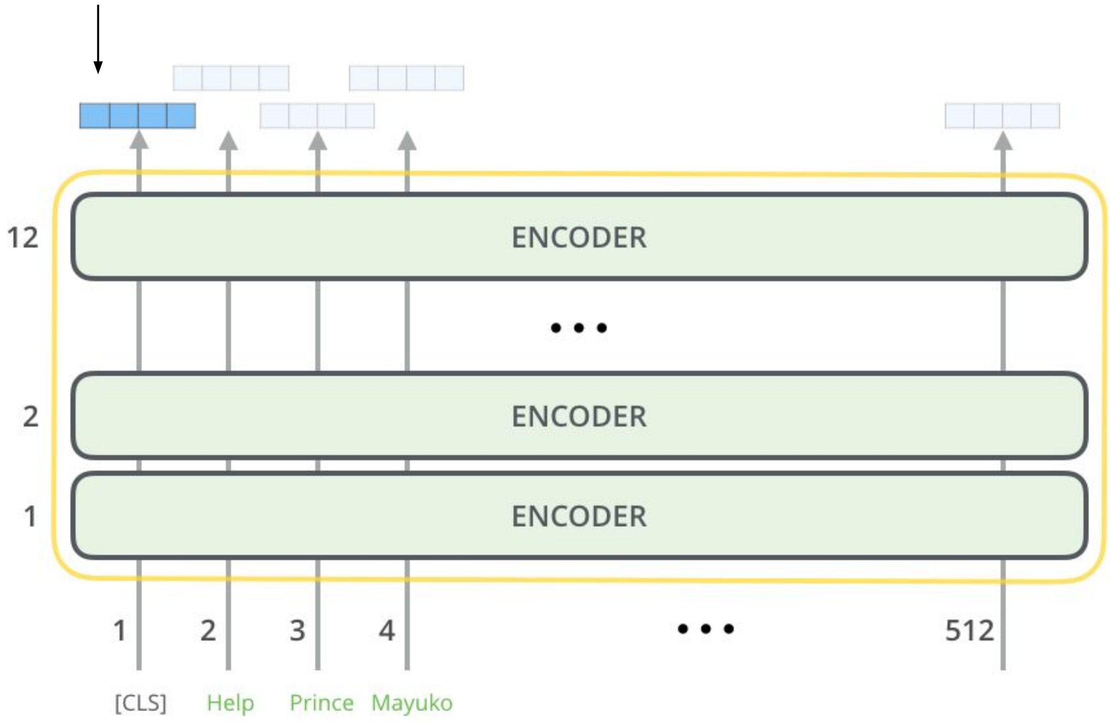
Bert



This Bert model can process sequences up to 512 tokens.

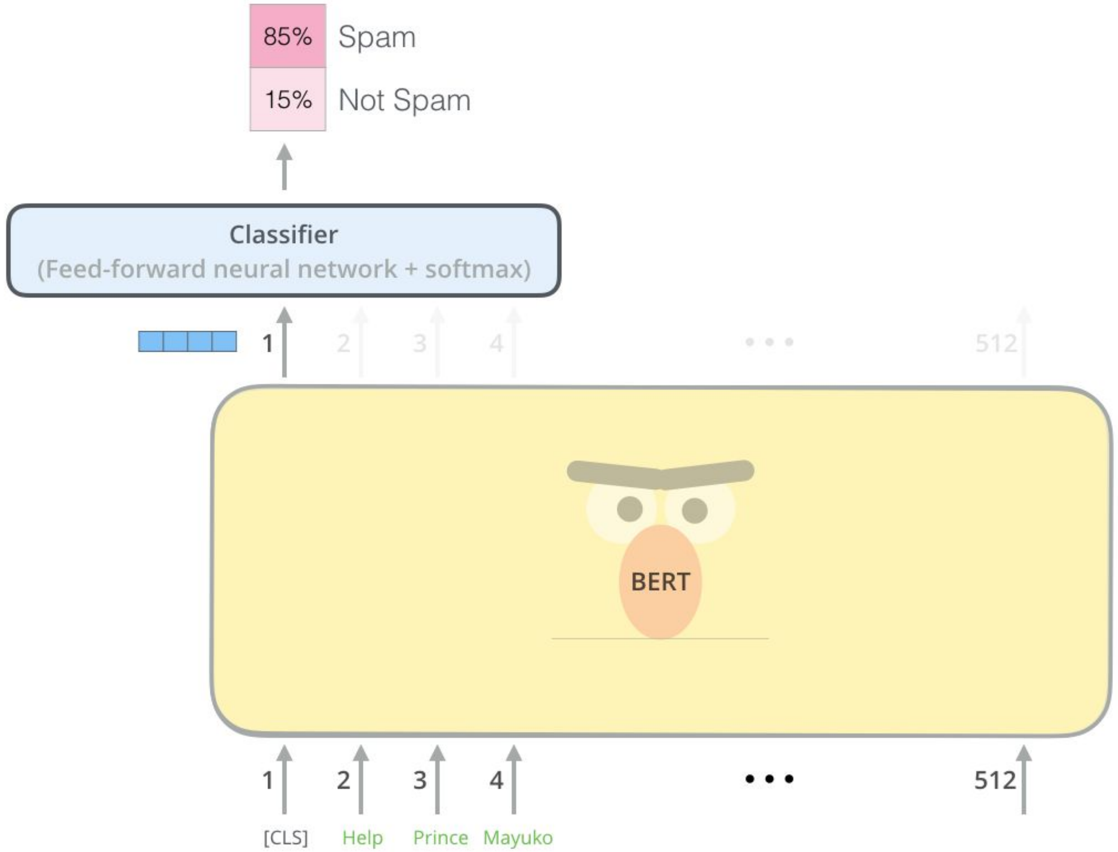
Bert

Each token generates a vector with the length of the hidden size.

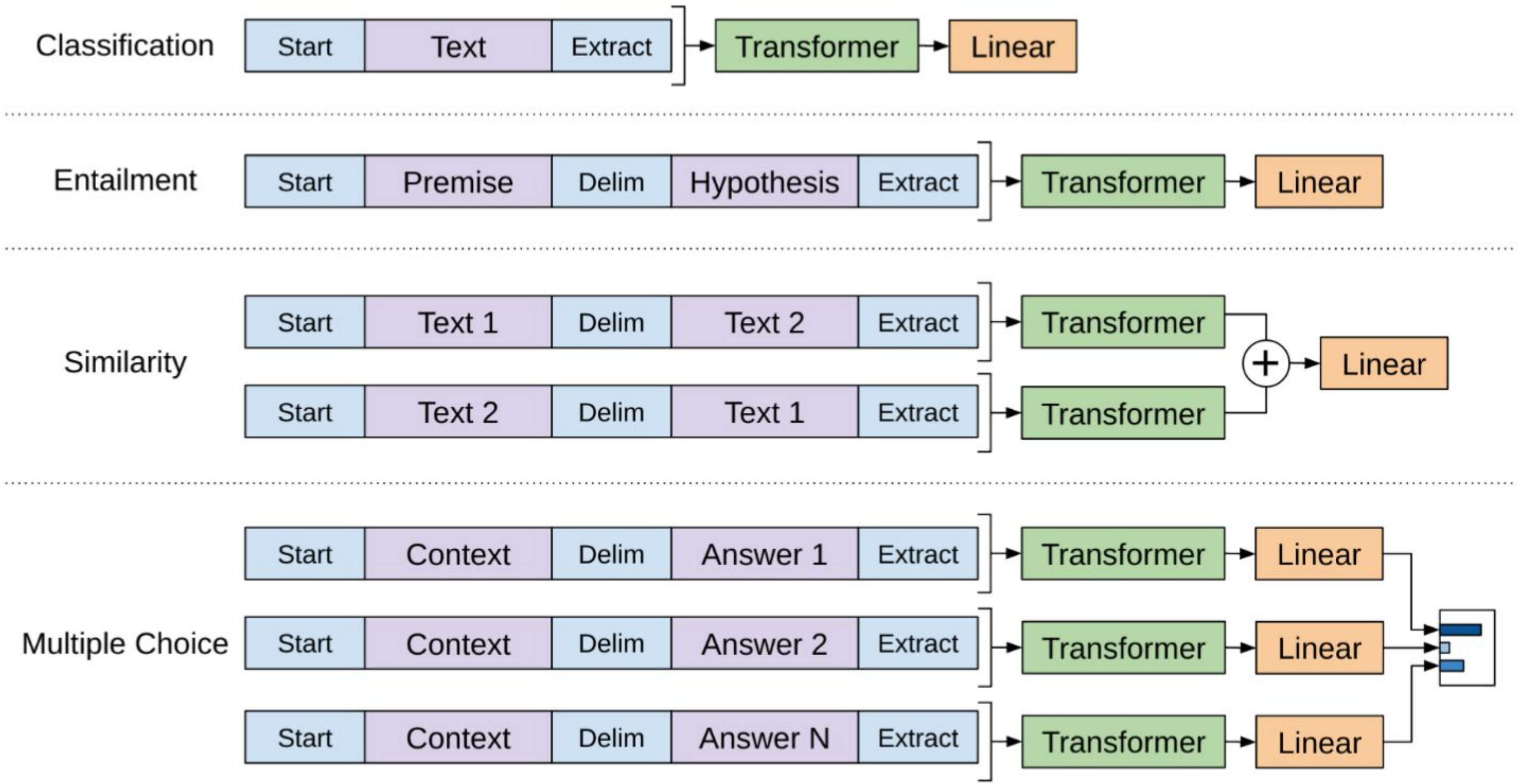


BERT

Bert Classification

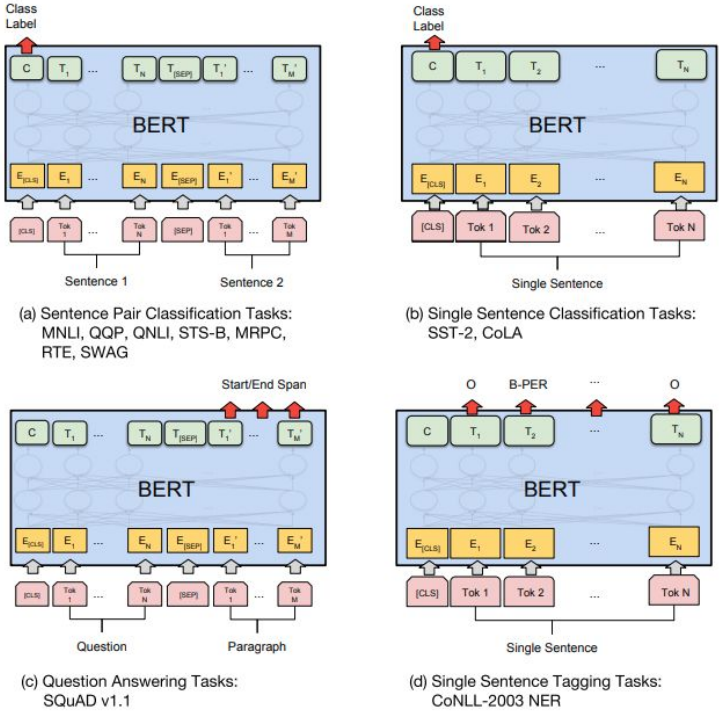


Task specific training



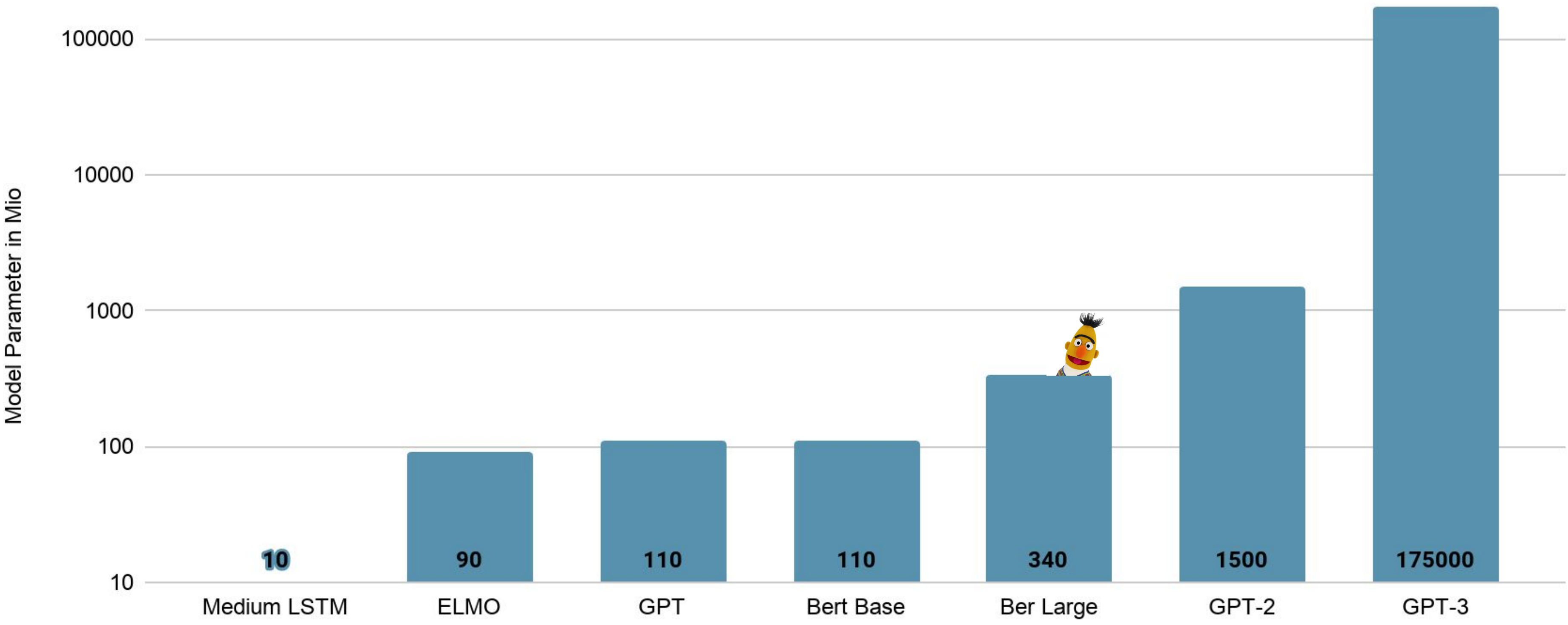
Improving Language Understanding by Generative Pre-Training, Radford et. al

Task specific training

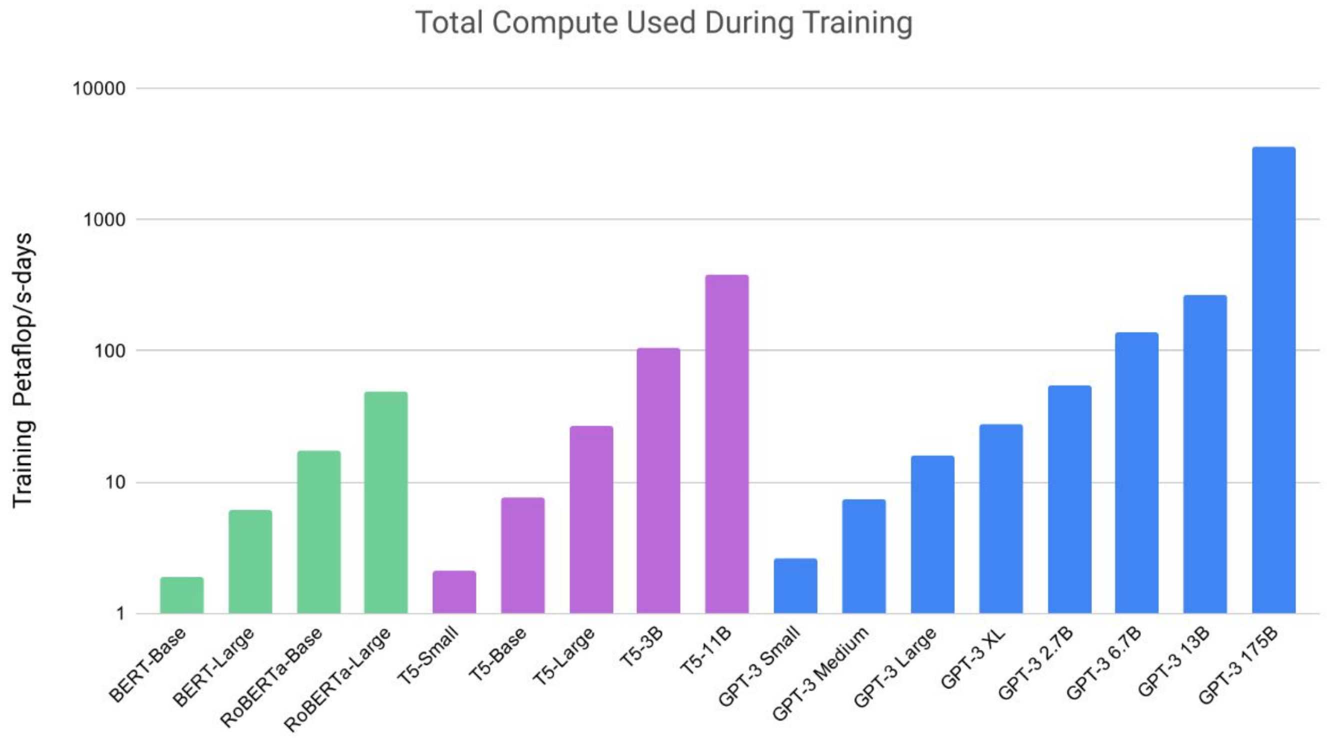


Improving Language Understanding by Generative Pre-Training, Radford et. al

How deep are these models?



How long does it take to train such a model?



[Language Models are Few-Shot Learners, Brown et al.]

Transformer

- Paper
  - Attention is all you need. Vaswani et al.
  - BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding Devlin et al.
  - Reformer: The Efficient Transformer Kitev et al.

- Good Read
  - Jay Alammars The Illustrated Transformer
  - Jay Alammars The Illustrated BERT

- Conference Talk:
  - Attention is all you need attentional neural network models by Łukasz Kaiser

Identify offensive language using Transformers

Tutorial

- shared task on the identification of offensive language from GermEval 2018
- Project Page
  - https://projects.fzai.h-da.de/iggsa/projekt/
- Dataset
  - https://github.com/uds-lsv/GermEval-2018-Data

The Data

- The task is to decide whether a message includes some form of offensive language or or not.

- **OFFENSE**
  - Juhu, das morgige Wetter passt zum Tag SCHEIßWETTER
  - @KarlLagerfeld ist in meinen Augen strunzdumm wie ein Knäckebrot.

- **OTHER**
  - @Sakoelabo @Padit1337 @SawsanChebli Nicht alle Staatssekretäre kann man ernst nehmen.
  - Die Türkei führt einen Angriffskrieg und die @spdde inkl. @sigmargabriel rüstet noch ihre Panzer auf.

Your Task

- Get familiar with the training script
- Improve the training script to beat the top score from 2018

- The top team from TU Wien scored in 2018
  - Accuracy 79,53%
  - F1 76,77%