A Gentle Introduction to (Large) Language Models

From word2vec to ChatGPT

Jakub Koubele

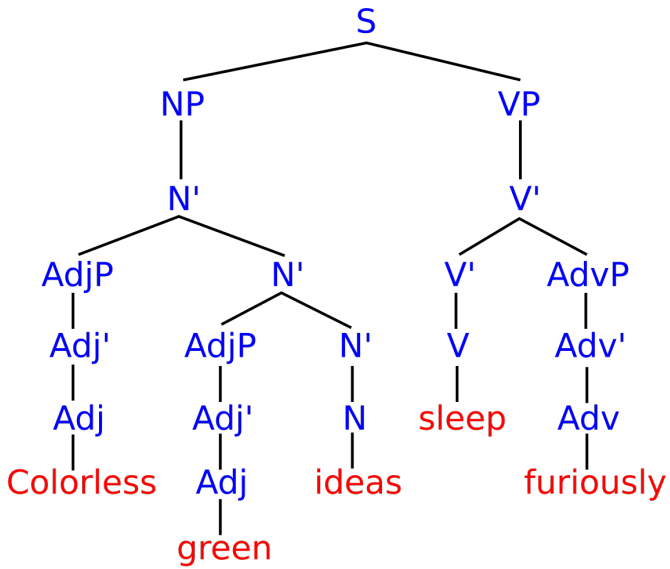
Beyer lab retreat

22nd August, 2025

What is a language, anyway?

What is a language, anyway?

In a hole in the ground there lived a hobbit. Not a nasty, dirty, wet hole, filled with the ends of worms and an oozy smell, nor yet a dry, bare, sandy hole with nothing in it to sit down on or to eat: it was a hobbit-hole, and that means comfort.




```c
float Q_rsqrt( float number )
{
    long i;
    float x2, y;
    const float threehalfs = 1.5F;

    x2 = number * 0.5F;
    y  = number;
    i  = * ( long * ) &y;                       // evil floating point bit level hacking
    i  = 0x5f3759df - ( i >> 1 );               // what the fuck?
    y  = * ( float * ) &i;
    y  = y * ( threehalfs - ( x2 * y * y ) );   // 1st iteration
//  y  = y * ( threehalfs - ( x2 * y * y ) );   // 2nd iteration, this can be removed

    return y;
}
```

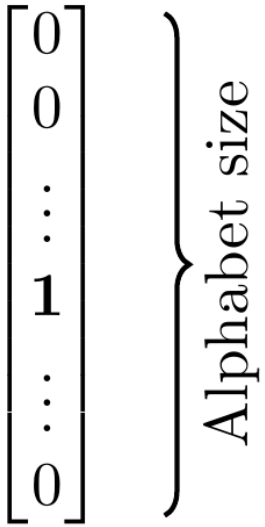
Tokenization

Tokenization

- Text = sequence of symbols.
- Alphabet = set of all symbols.

Tokenization

- Text = sequence of symbols.
- Alphabet = set of all symbols.
- Symbol can be represented by one-hot vector.



Tokenization

- Text = sequence of symbols.
- Alphabet = set of all symbols.
- Symbol can be represented by one-hot vector.
- Which alphabet we should use?
  - Words?
  - Letters?

$$\underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ \mathbf{1} \\ \vdots \\ 0 \end{bmatrix}}_{\text{Alphabet size}}$$

Tokenization

- Text = sequence of symbols.
- Alphabet = set of all symbols.
- Symbol can be represented by one-hot vector.
- Which alphabet we should use?
  - Words?
  - Letters?
- Subword tokenization (e.g., WordPiece) is commonly used.

$$\underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}}_{\text{Alphabet size}}$$

```
In : tokenizer.tokenize("Bilbo Baggins was a hobbit.")
Out : ['B', '##il', '##bo', 'Ba', '##ggins', 'was', 'a', 'ho', '##bb', '##it', '.']
```

Problems of one-hot encoding

Problems of one-hot encoding

- After tokenization, we have a sequence of tokens, represented by one-hot vectors.

$$\begin{bmatrix}0\\0\\0\\0\\\vdots\\\mathbf{1}\\\vdots\\0\end{bmatrix}, \begin{bmatrix}0\\\mathbf{1}\\0\\\vdots\\\vdots\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\\vdots\\\vdots\\\vdots\\0\\0\\0\\\mathbf{1}\end{bmatrix}, \ldots, \begin{bmatrix}0\\0\\\mathbf{1}\\\vdots\\\vdots\\0\\0\\0\end{bmatrix}$$

Problems of one-hot encoding

- After tokenization, we have a sequence of tokens, represented by one-hot vectors.

$$\begin{bmatrix}0\\0\\0\\0\\\vdots\\\mathbf{1}\\\vdots\\0\end{bmatrix}, \begin{bmatrix}0\\\mathbf{1}\\0\\\vdots\\\vdots\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\\vdots\\\vdots\\\vdots\\0\\0\\0\\\mathbf{1}\end{bmatrix}, \ldots, \begin{bmatrix}0\\0\\\mathbf{1}\\\vdots\\\vdots\\0\\0\\0\end{bmatrix}$$

- Problems:
  - High dimensionality (alphabet size usually 10k - 100k symbols).
  - Spare vectors without much information.

Distributed word representation

Distributed word representation

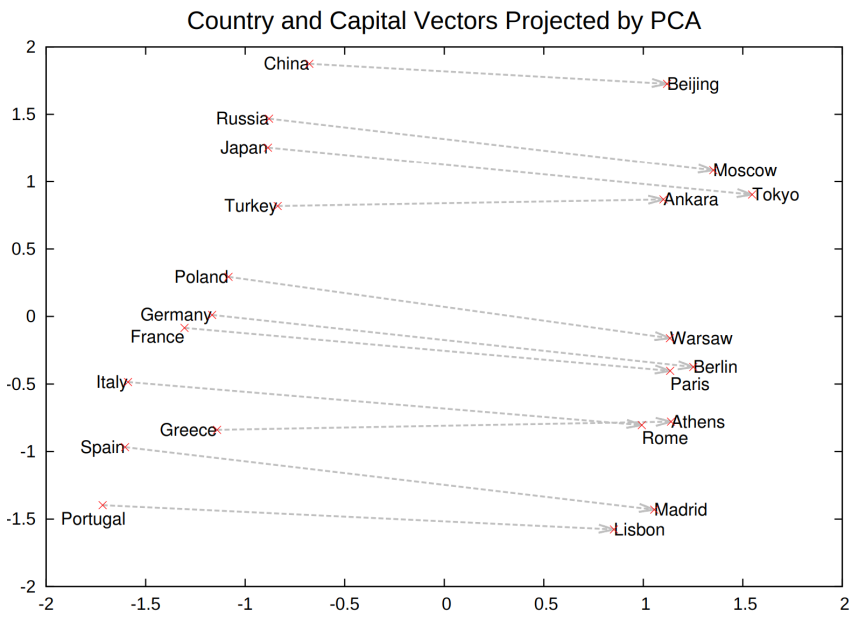
- We would like to represent the tokens (words) by vectors that:
  - Are dense, with reasonable dimensionality (e.g., 100 – 500).

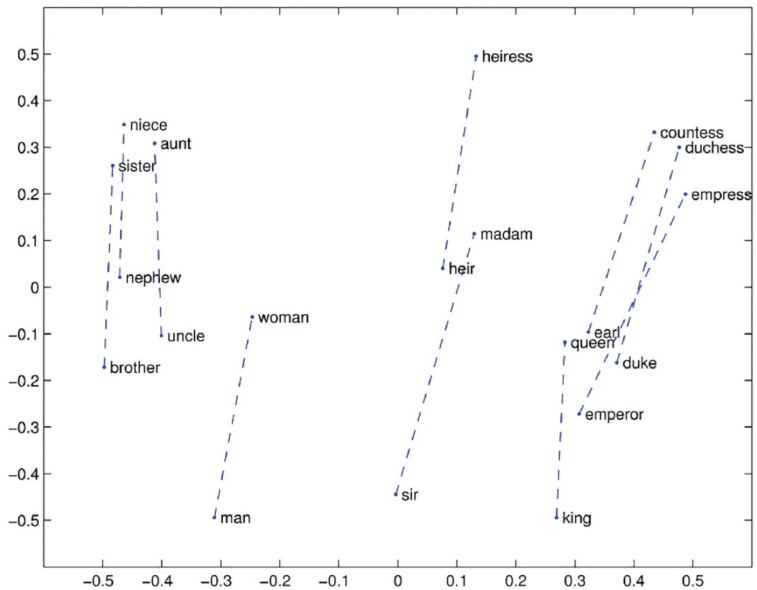
$$\begin{bmatrix} 0.72 \\ -0.14 \\ 4.35 \\ \vdots \\ -1.80 \\ 2.15 \end{bmatrix}$$

Distributed word representation

- We would like to represent the tokens (words) by vectors that:
  - Are dense, with reasonable dimensionality (e.g., 100 – 500).
  - Arithmetic of these vectors (addition, subtraction etc.) captures the semantics of words.

$$\begin{bmatrix} 0.72 \\ -0.14 \\ 4.35 \\ \vdots \\ -1.80 \\ 2.15 \end{bmatrix}$$

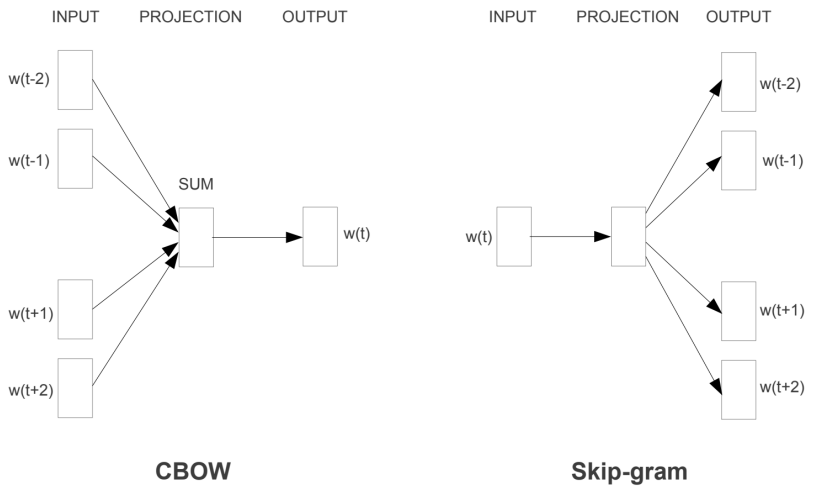




Word2vec (2013)

Word2vec (2013)

- Two similar models, predicting word(s) from vector representation of words nearby. The vectors are parameters of the model, and are learned by maximum likelihood estimation (MLE).



...government debt problems turning into ***banking*** *crises as happened in 2009...*
...saying that Europe needs unified ***banking*** *regulation to replace the hodgepodge...*
...India has just given its ***banking*** *system a shot in the arm...*

These context words will represent ***banking***

Brief intro to neural networks

Brief intro to neural networks

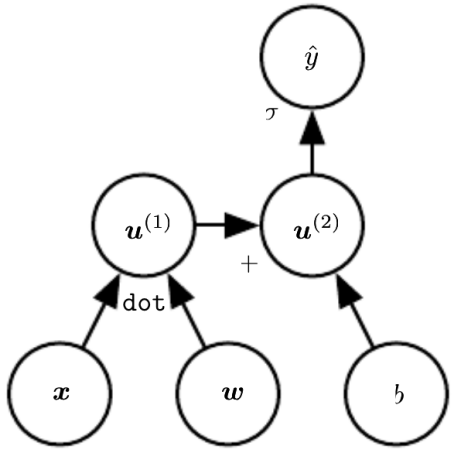
- Neural networks are **functions**, that...

Brief intro to neural networks

- Neural networks are **functions**, that...
  - Work with **tensors** (nd-arrays).

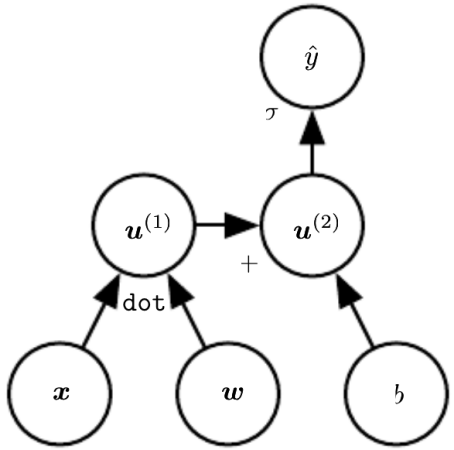
Brief intro to neural networks

- Neural networks are **functions**, that...
  - Work with **tensors** (nd-arrays).
  - Are **compound** (composite of simple operations)
    - May be represented by directed acyclic **computational graph**.



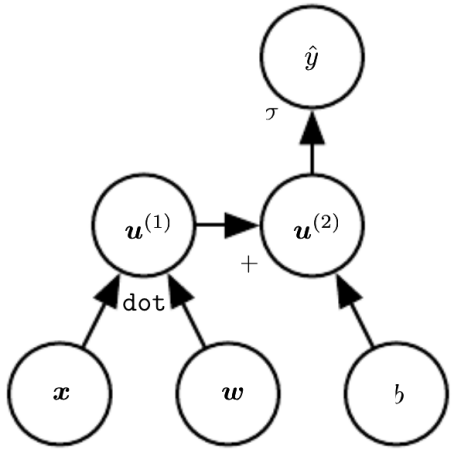
Brief intro to neural networks

- Neural networks are **functions**, that...
  - Work with **tensors** (nd-arrays).
  - Are **compound** (composite of simple operations)
    - May be represented by directed acyclic **computational graph**.
  - Are **parametrized**.



Brief intro to neural networks

- Neural networks are **functions**, that...
  - Work with **tensors** (nd-arrays).
  - Are **compound** (composite of simple operations)
    - May be represented by directed acyclic **computational graph**.
  - Are **parametrized**.
  - Are **differentiable** (loss of output with respect to the parameters).
    - Parameters are learned by gradient methods.



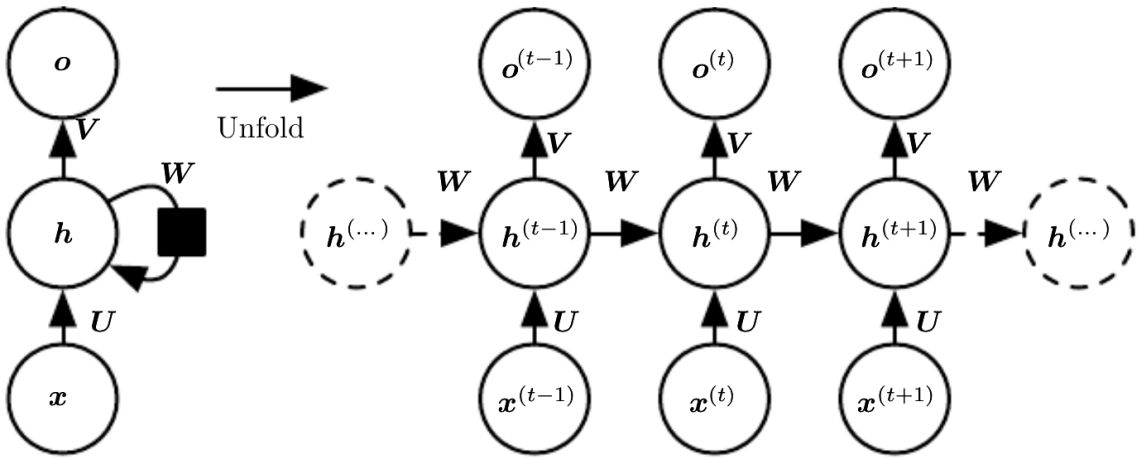
Recurrent neural networks (RNNs)

Recurrent neural networks (RNNs)

- RNNs are specialized for sequential data.

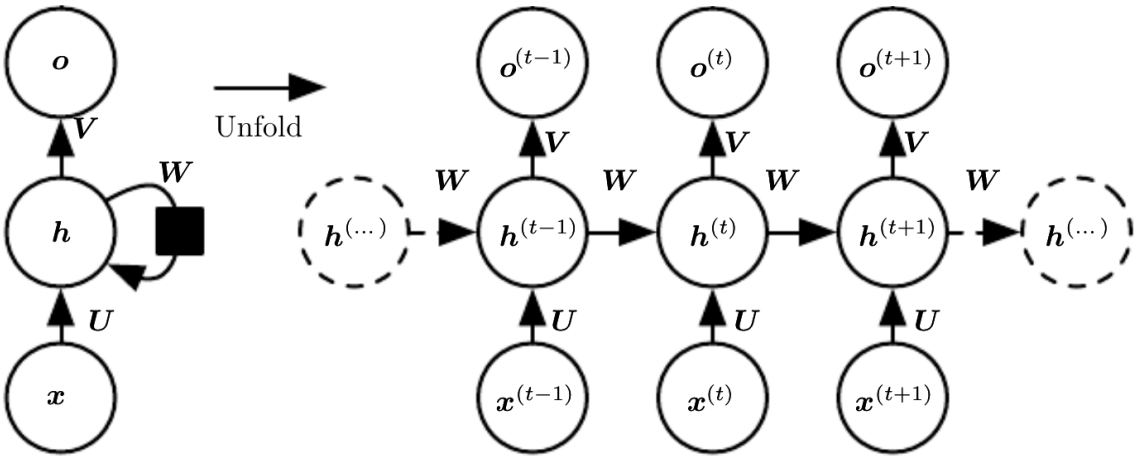
Recurrent neural networks (RNNs)

- RNNs are specialized for sequential data.
- Have **inner state** that is passed through time.
  - Output depends on the inner state.
  - Taking input updates the inner state.

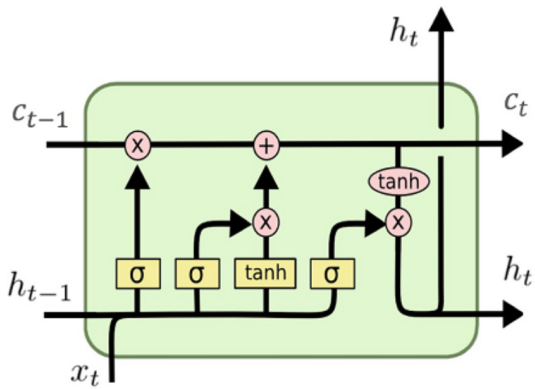


Recurrent neural networks (RNNs)

- RNNs are specialized for sequential data.
- Have **inner state** that is passed through time.
  - Output depends on the inner state.
  - Taking input updates the inner state.



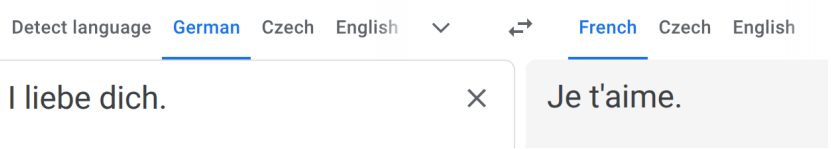
- Many different architectures, most popular one is the Long-Short Term Memory (LSTM).

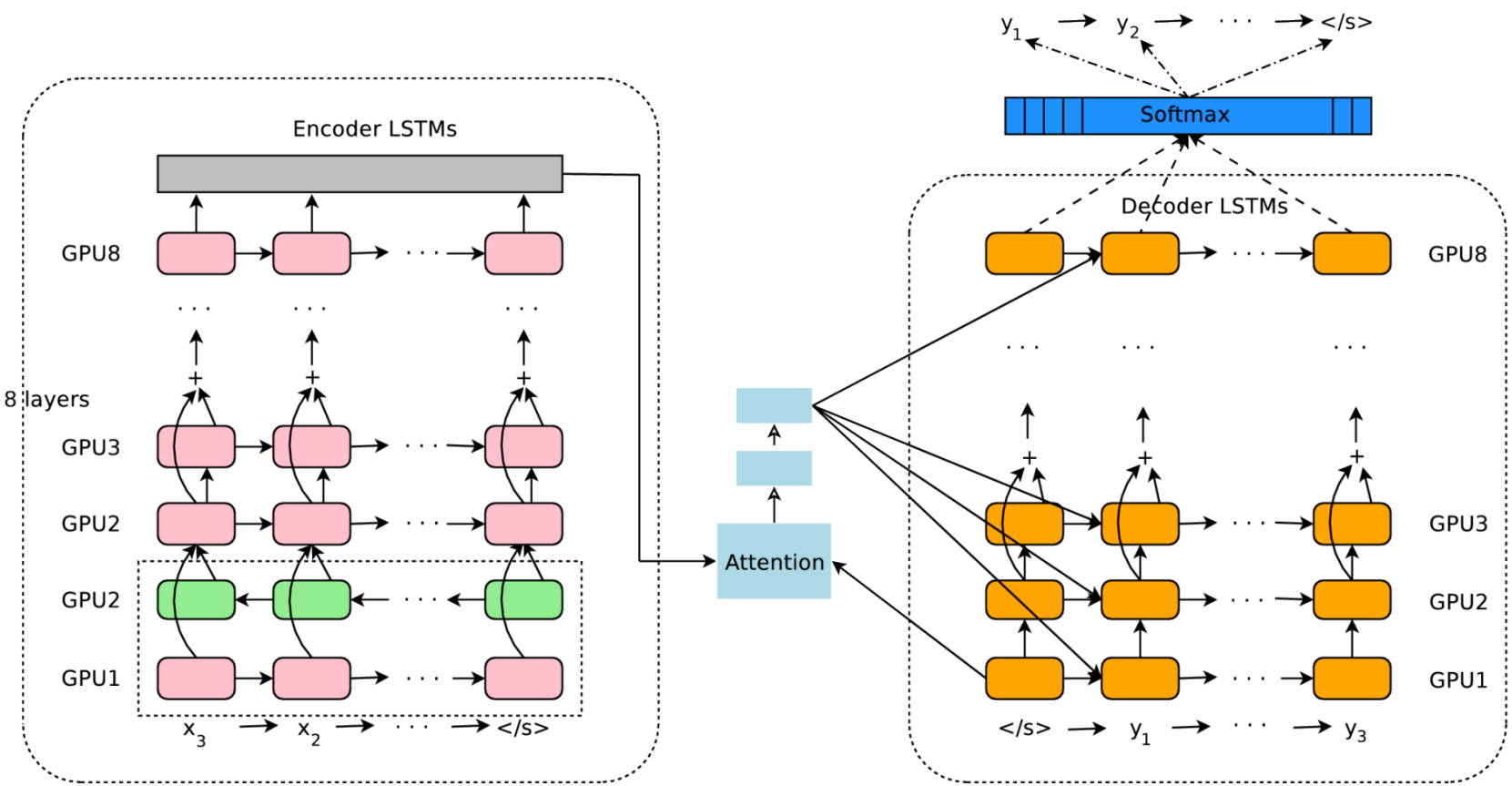


LSTM
(Long–Short Term Memory)

Google Translator (2016)

Google Translator (2016)





All you need is...

All you need is...

- Before 2017, the best performing model architectures were based on LSTM.

All you need is...

- Before 2017, the best performing model architectures were based on LSTM.
- Until...they weren't.

All you need is...

- Before 2017, the best performing model architectures were based on LSTM.
- Until...they weren't.

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

All you need is...

- Before 2017, the best performing model architectures were based on LSTM.
- Until...they weren't.

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

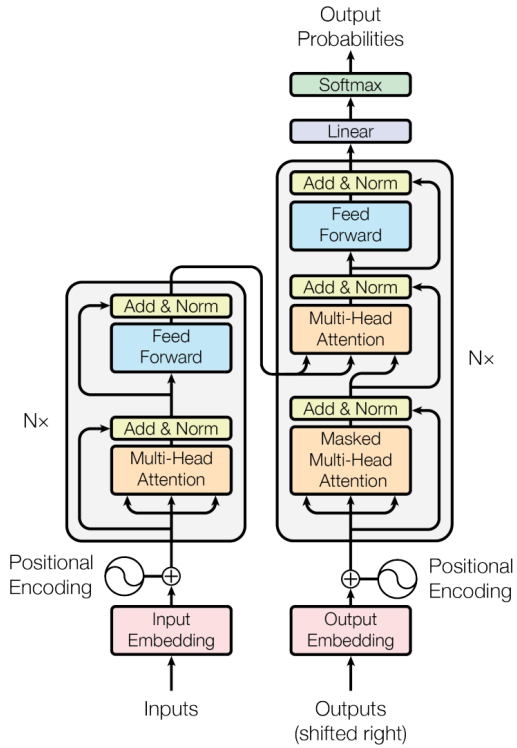
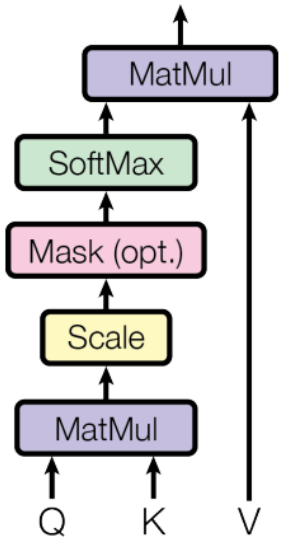


Figure 1: The Transformer - model architecture.

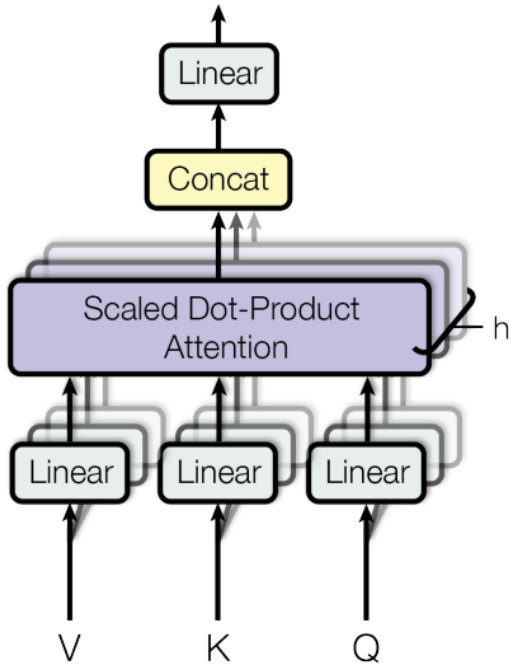
Attention layer

Attention layer

Scaled Dot-Product Attention



Multi-Head Attention



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

Attention visualization

Attention visualization

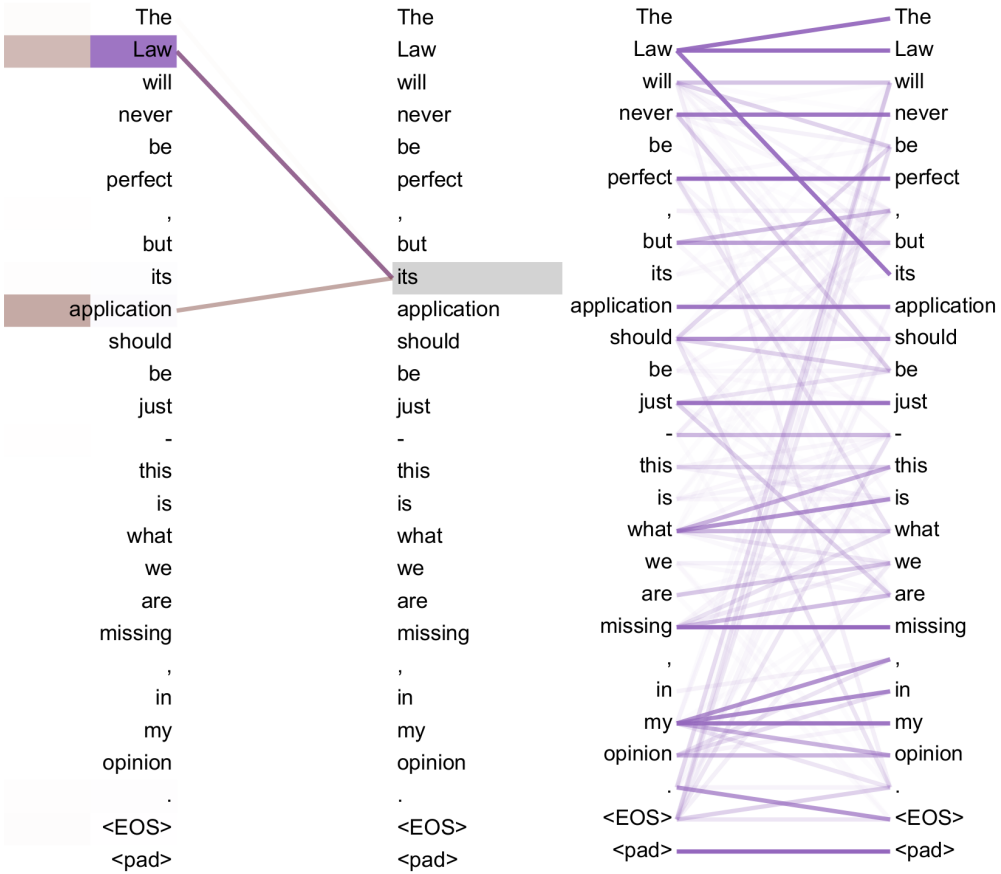


Figure 4: Two attention heads, also in layer 5 of 6, apparently involved in anaphora resolution. Top: Full attentions for head 5. Bottom: Isolated attentions from just the word 'its' for attention heads 5 and 6. Note that the attentions are very sharp for this word.

BERT (Bidirectional Encoder Representation from Transformer)

BERT (Bidirectional Encoder Representation from Transformer)

- Pre-train transformer on masked-word prediction.
  - No labeled data needed, can be done on large corpus.

BERT (Bidirectional Encoder Representation from Transformer)

- Pre-train transformer on masked-word prediction.
  - No labeled data needed, can be done on large corpus.
- Finetune on task-specific labeled data.

BERT (Bidirectional Encoder Representation from Transformer)

- Pre-train transformer on masked-word prediction.
  - No labeled data needed, can be done on large corpus.
- Finetune on task-specific labeled data.

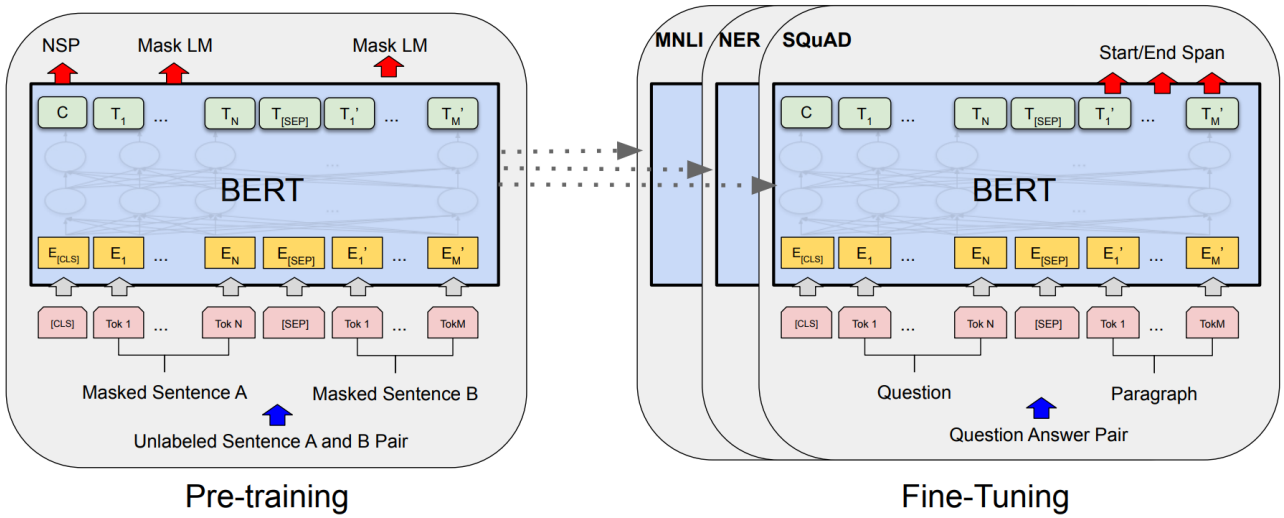


Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. `[CLS]` is a special symbol added in front of every input example, and `[SEP]` is a special separator token (e.g. separating questions/answers).

GPT (Generative Pretrained Transformer)

GPT (Generative Pretrained Transformer)

- Pretrained on next word prediction.

GPT (Generative Pretrained Transformer)

- Pretrained on next word prediction.
- Worse performance than BERT for fine-tunning, but good for generating text.

GPT (Generative Pretrained Transformer)

- Pretrained on next word prediction.
- Worse performance than BERT for fine-tunning, but good for generating text.
- Few-shot / zero-shot learning without gradient updates.

Generating text is not enough

Generating text is not enough

Prompt

Explain the moon landing to a 6 year old in a few sentences.

Completion

GPT-3

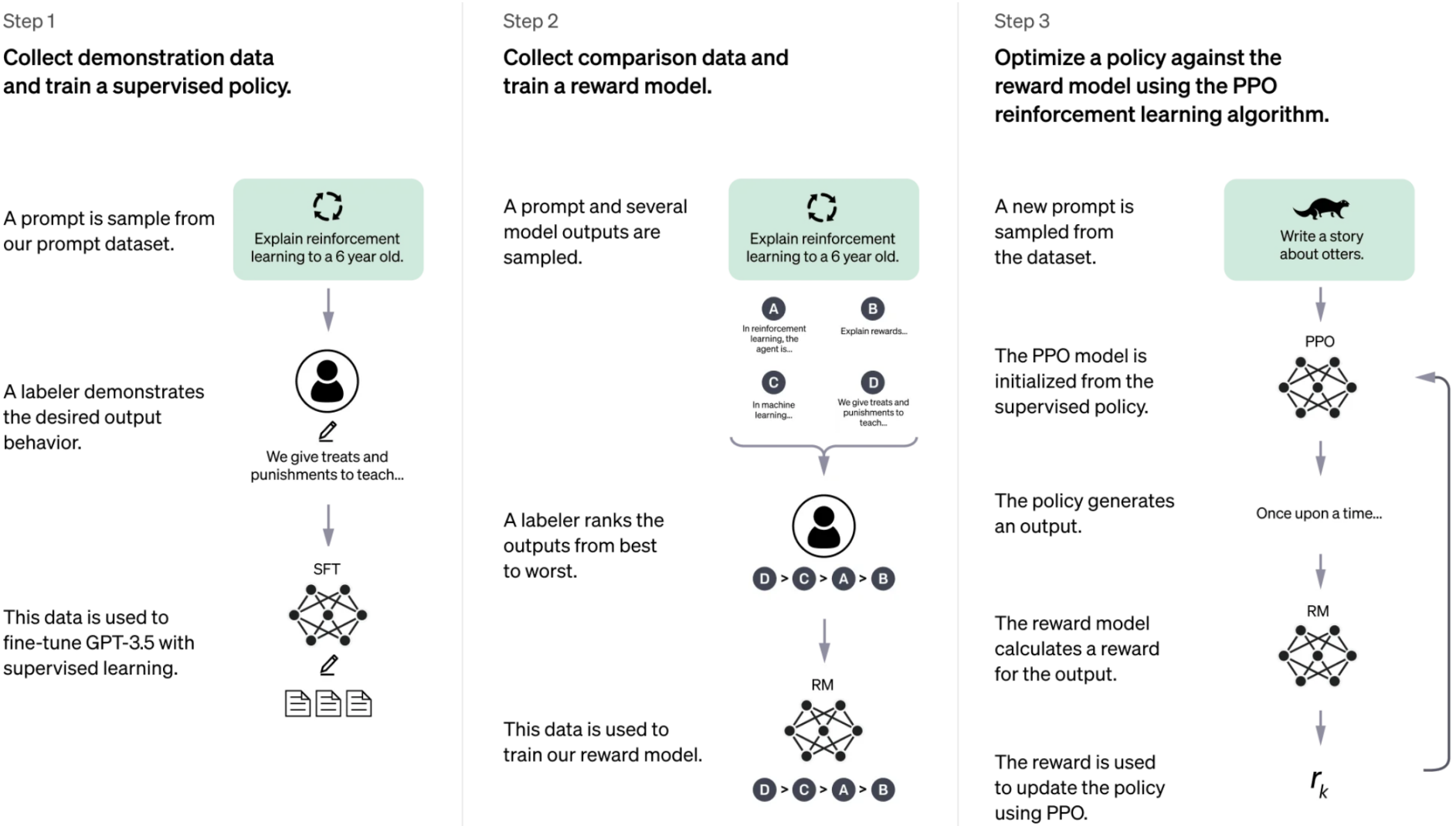
```
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

InstructGPT

```
People went to the moon, and they took pictures of what they saw, and
sent them back to the earth so we could all see them.
```

Reinforcement Learning from Human Feedback (RLHF)

Reinforcement Learning from Human Feedback (RLHF)



The End

Thanks for your attention! (It's all you need).

References (1/2)

General references

- Goodfellow, Ian, et al. Deep learning. (2016).
- Transformers library
- CS224N at Standford

Papers (1/2)

- Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." (2013).
- Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." (2013).
- Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." (2014).
- Wu, Yonghui, et al. "Google's neural machine translation system: Bridging the gap between human and machine translation." (2016).
- Vaswani, Ashish, et al. "Attention is all you need." (2017).

References (2/2)

Papers (2/2)

- Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." (2019).
- Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).
- Radford, Alec, et al. "Language models are unsupervised multitask learners." (2019)
- Brown, Tom, et al. "Language models are few-shot learners." (2020)
- Ouyang, Long, et al. "Training language models to follow instructions with human feedback." (2022).