



ALICE was beginning to get very tired of sitting by her sister upon the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, "and what is the use of a book," thought Alice, "without pictures or conversations?" So she was very glad to find herself once more in her own mind, (as well as she could be, in such a state) and whether the pleasure of making a daisy-chain was worth the trouble of getting up and down among the waistises, when suddenly a white rabbit with pink whiskers ran close by her. There was nothing very remarkable in that; nor did Alice think it very remarkable out of the way to hear the Rabbit say to itself, "Oh dear ! Oh dear ! I shall be too late ! " When she thought it over afterwards, it occurred to her that she ought to have wondered at this, but it seemed quite natural; but when the Rabbit actually took a watch out of its waistcoat-pocket, and looked at it, and then hurried on, Alice started to her feet, for it flashed across her mind that she had never before seen a rabbit with either a waistcoat-pocket or a watch to take out of it.

Miner Image source: <https://freesvg.org/miner-1574424884>

## Doctoral Course: The Rise (and *Rise*) of Transformers

Personally I'm yet to find  
any serious limitations  
for this architecture ...

# Lecture: Text Transformers

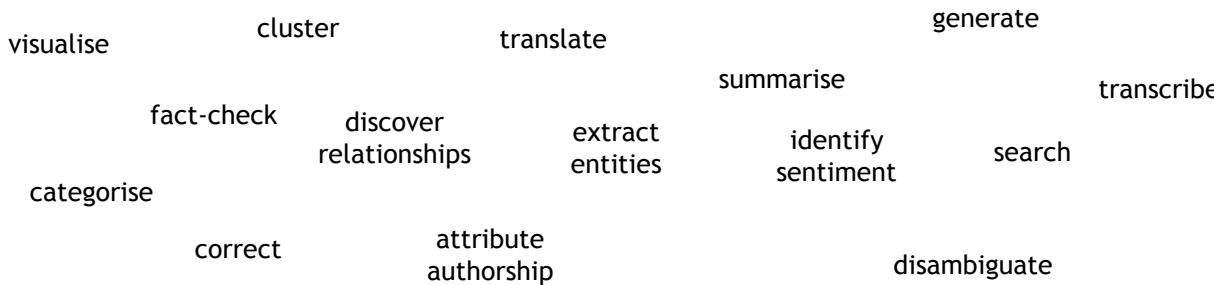
Mark Carman

# What is today's session about?

Mark Carman  
25.01.2023

## Natural Language Processing (NLP) / Text Mining

- process of **working with** & **extracting useful knowledge** from **text data**
- lots of different things we can do with text data:



ALICE was beginning to get very tired of sitting by her sister on the bank, and of having nowhere to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, " and what is the use of a book," thought Alice, " without pictures or conversations ?" So she was considering in her own mind, (as well as she could, for the hot day made her feel very sleepy and stupid,) whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a white rabbit with pink ears ran close by her. There was nothing so very remarkable in that; nor did Alice think it so very much out of the way to hear the Rabbit say to itself, " Oh dear ! I shall be too late !" (when she thought it over afterwards, it occurred to her that she ought to have wondered at this, but at the time it all seemed quite natural); but when the Rabbit actually took a watch out of its waistcoat-pocket, and looked at it, and then hurried on, Alice started to her feet, for it flashed across her mind that she had never before seen a rabbit with either a waistcoat-pocket or a watch to take out of it, and

Miner Image source: <https://freesvg.org/images/1574024894>

## with **Transformers**

- machine learning models using deep self-attention neural networks
- which have revolutionised performance on **all these tasks** over last few years

## Who am I?

- Mark Carman, Politecnico di Milano
- Background:
  - Information Retrieval & *statistical* Natural Language Processing\*\*
  - Machine Learning & Data Science
- Applications:
  - *Personalisation & Recommendation, Web Search, Social Media Analysis, Digital Forensics, Bioinformatics, ...*
- Teaching:
  - Data Science and Artificial Intelligence
  - classes are MUCH more fun when they're interactive, so please **help me out** by asking **lots of questions!**



🤔 Let's hope his teaching is better than his cooking ... 😅😅

\*\* Favourite NLP quote:  
“Every time I fire a linguist, the performance of  
the speech recognizer goes up” Frederick Jelinek

## Who are you?

- How familiar are you with Machine Learning?
- What about text processing?
- And Deep Learning?



CC, source: [https://commons.wikimedia.org/wiki/File:5th\\_Floor\\_Lecture\\_Hall\\_\(cartoon\).jpg](https://commons.wikimedia.org/wiki/File:5th_Floor_Lecture_Hall_(cartoon).jpg)

# 🔍 Today's Lecture ...

Will be about the **NLP / Text Mining view** on **Transformers**

- Not much maths 😢
  - aimed more toward those with less of a deep learning background
  - but lots of useful information for those with substantial DL experience
- Transformers come from NLP, so Matteo introduced many concepts already 😠
  - lecture will necessarily contain some repetition of concepts explained on Monday
  - but viewing from different perspective always useful for understanding (**repetita iuvant**)



For **maths/deep learning lovers** out there, I have 3 questions for you to think about during the lecture:

1. Do we really need to linearly transform the key values or is the query enough?
2. Where are most of the parameters in a Transformer located?
3. How does the number of parameters grow with:
  - i. dimension of the embedding
  - ii. number of heads
  - iii. depth (number of transformer blocks)
  - iv. sequence length
  - v. vocabulary size



# 🔍 Text generation has become so good ...

👤 Mark Carman  
📅 25.01.2023

So good in fact, that I considered using it to generate a recent talk ... 😜🤣

## Workshop Title: How Deep Neural Techniques have revolutionised Information Retrieval and Text Mining

**Abstract:** The world of information retrieval and text mining is a global one, but the applications of deep learning have become highly specific and specialized. There are huge opportunities in this field. We examine how deep learning have revolutionized the problem of extracting and processing text. We demonstrate the importance of two major breakthroughs in the field: 1) The first, deep convolutional neural network (DNN), which can find hidden layer effects using very sparse input and a few simple operators. It is widely used for word embeddings and natural language processing. 2) The second, reinforcement learning, which can learn the hidden layer effect using a small amount of input. This technique is also widely used for image and speech recognition.

## Workshop Title: How Deep Neural Techniques have revolutionised Information Retrieval and Text Mining

**Abstract:** Over the last few years, deep neural architectures have rewritten the rulebook in terms of the performance that can be achieved across a multitude of text processing tasks from sentiment analysis and sarcasm detection, to machine translation, web search, question answering, and dialog generation. In this workshop I will explain the language modelling technology behind these advances, discussing its evolution from shallow embeddings to modern transformer models composed of ever deeper self-attention networks. I will describe numerous applications of these deep models in information retrieval and text mining and then look to the future, to applications that seamlessly combine information across text and image modalities.

- One of these texts was **generated automatically** by conditioning on the title
- The other is my abstract. Can you tell which is which?
- If you guessed that the first abstract was the automatically generated one
  - then you were right ;-)
  - try the same GPT-2 based text generator here: <https://transformer.huggingface.co/>

## 1. Brief history of language models

- What is language modelling and why should I care about it?
- Markov models, word embeddings, recurrent neural networks, attention

## 2. Transformers for text

- deep learning, tokenization, why self-attention works
- BERT vs GPT-2, how to use them
- applications Text classification, translation, summarisation, etc.

## 3. research applications

Nutrition information			
Typical values	Per 100g	Per 1/4 pot	% based on GDA for women
Energy	256 kJ <b>61 kcal</b>	320 kJ <b>76 kcal</b>	3.8%
Protein	<b>4.9g</b>	<b>6.1g</b>	13.6%
Carbohydrate	<b>6.9g</b>	<b>8.6g</b>	3.7%
of which sugars	<b>6.9g</b>	<b>8.6g</b>	9.6%
of which starch	nil	nil	-
Fat	<b>1.5g</b>	<b>1.9g</b>	2.7%
of which saturates	<b>0.9g</b>	<b>1.1g</b>	5.5%
mono-unsaturates	0.4g	0.5g	-
polyunsaturates	nil	nil	-
Fibre	nil	nil	nil
Salt	<b>0.2g</b>	<b>0.3g</b>	5.0%
of which sodium	trace	0.1g	42%
Vitamins & minerals			
Calcium	<b>168mg</b>	<b>210mg</b>	26%

Nutrition Information UK Label Yoghurt by Samatarou (CC0 1.0)

\*\*all part of a balanced  
deep learning diet

Plus: hands on  
training of models  
in Google colab!

# What is Language Modeling and why should I care about it?



Source: <https://pixabay.com/photos/bored-female-girl-people-school-16811/>

# 🔍 A language what?

According to [Wikipedia](#):

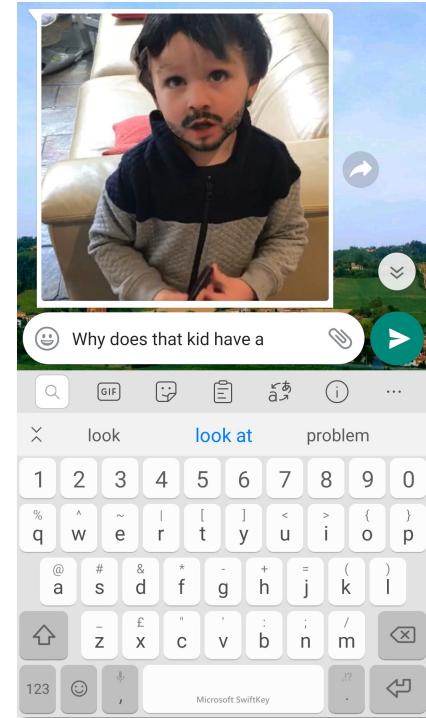
- A **statistical language model** is a **probability distribution** over **sequences of words**

If we have a distribution over word sequences,

- we can **condition** the next word on the previous content,
- and **sample new sequences** from it

In other words a **language model**

- is general-purpose random **text generator**



Way back in 2019 there was a lot of buzz about language models when **GPT-2** was released

- Could a **language model** really be that dangerous?



**Support The Guardian**  
Available for everyone, funded by readers  
[Contribute →](#) [Subscribe →](#)

Sign in

# The Guardian

News Opinion Sport Culture Lifestyle

World UK Science Cities Global development Football Tech Business More

Artificial intelligence (AI)

## New AI fake text generator may be too dangerous to release, say creators

The Elon Musk-backed no release research publicly f

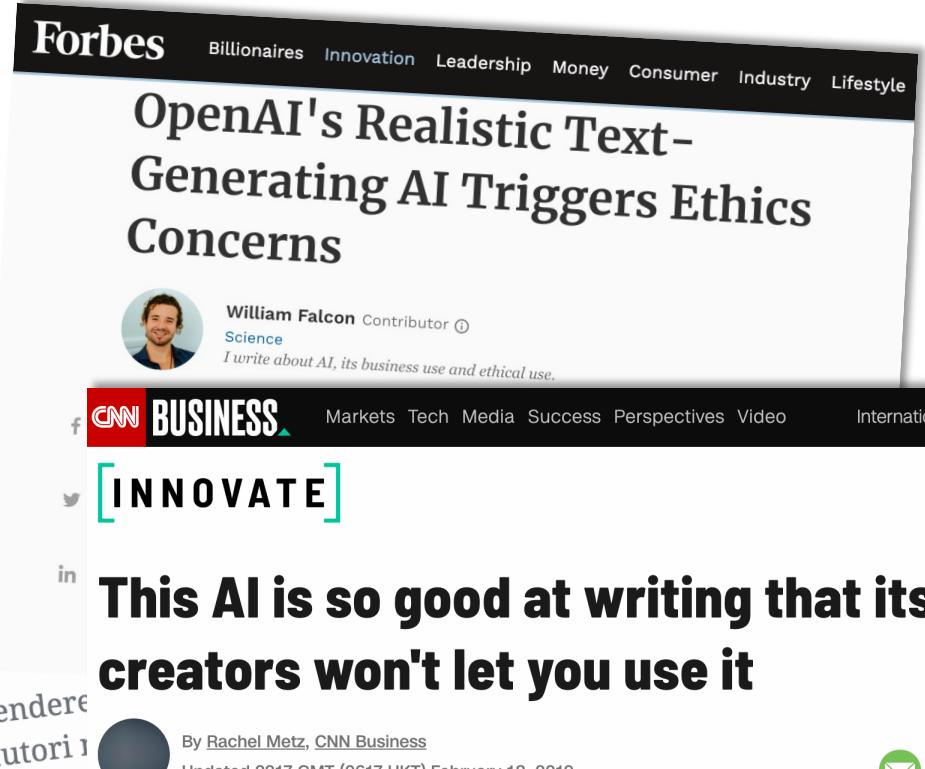
WIRED IT Sezioni ▾ Wired Next Fest ▾

HOT TOPIC GOVERNO ELEZIONI EUROPEE CYBERSECURITY TRAILER TRUMP

# Open Ai, l'intelligenza artificiale di Elon Musk troppo pericolosa per essere resa pubblica

La nuova tecnologia è in grado di comprendere e generare testi in vari stili. Gli autori i

POLITECNICO MILANO 1863



Forbes Billionaires Innovation Leadership Money Consumer Industry Lifestyle

## OpenAI's Realistic Text-Generating AI Triggers Ethics Concerns



William Falcon Contributor ⓘ  
Science  
I write about AI, its business use and ethical use.

f CNN BUSINESS Markets Tech Media Success Perspectives Video International Edition

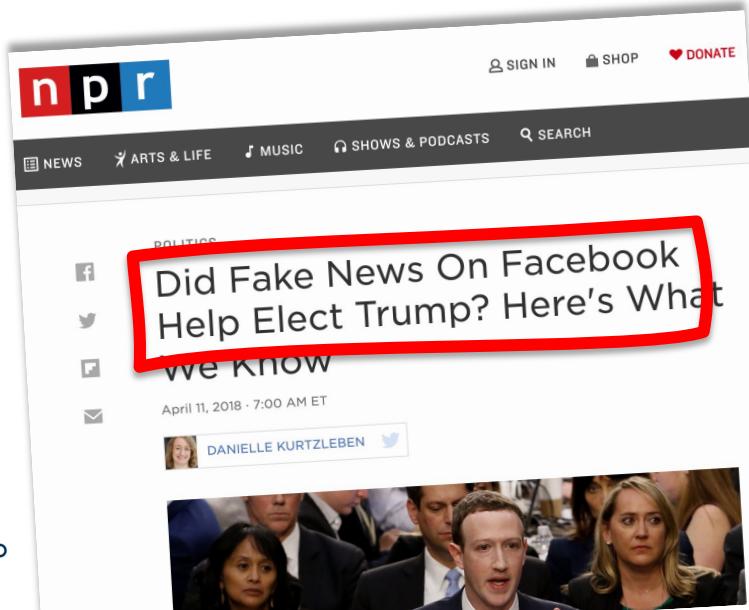
in [INNOVATE]

## This AI is so good at writing that its creators won't let you use it

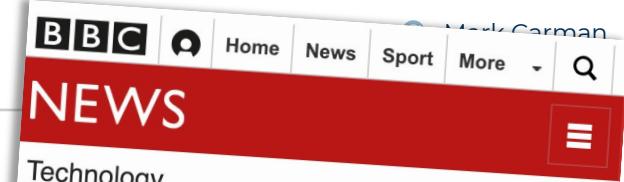
By Rachel Metz, CNN Business  
Updated 2317 GMT (0617 HKT) February 18, 2019

# Impact ...

- Language modeling may be a **lucrative business model** ...
- and in geopolitics:  
*the **automated pen** may be mightier than the sword*



The screenshot shows a news article from NPR. The title is "Did Fake News On Facebook Help Elect Trump? Here's What we KNOW". The text discusses the role of fake news on Facebook during the 2016 US election. A red box highlights the title.



The screenshot shows a BBC News article titled "Prices for fake news campaigns revealed". It includes a date (15 June 2017) and social media sharing icons. A red box highlights the headline.

Technology

Prices for fake news campaigns revealed

15 June 2017



A callout box highlights a section of the BBC article. It features a purple background with white text and a teal circular border. The text reads: "Tips for spotting false news. It's possible to spot false news. As we work to limit the spread, check out a few ways to identify whether a story is genuine." Below this, it says "Facebook ran adverts telling people how to spot fake news".

Mounting a year-long fake news campaign can cost about \$400,000 (£315,000), suggests a report.

The Trend Micro report draws on price lists found on sites that run the misinformation campaigns.

Costs cover setting up fake social media profiles

# brief history of Language Models

Markov models

word embeddings

recurrent neural networks

attention

model **statistical regularities** in text:

- to predict next word in sentence

example of a regularity?

- next token is often a repeat of a previous word from text

If we can predict next word:

- iterating forward allows us to predict entire sentences

## CHAPTER I. Down the Rabbit-Hole

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, 'and what is the use of a book,' thought Alice 'without pictures or conversations?'

So she was considering in her own mind (as well as she could, for the hot day made her feel very sleepy and stupid), whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a White Rabbit with pink eyes ran close by her.



Source: [https://commons.wikimedia.org/wiki/File:John\\_Tenniel\\_-\\_Illustration\\_from\\_The\\_Nursery\\_Alice\\_\(1890\)\\_-\\_C03757\\_02.jpg](https://commons.wikimedia.org/wiki/File:John_Tenniel_-_Illustration_from_The_Nursery_Alice_(1890)_-_C03757_02.jpg)

# brief history of Language Models

> Markov models

word embeddings

recurrent neural networks

attention

simplest models count **n-grams** in large corpus

- n-gram = sequence of  $n$  words
- longer n-grams give better predictions

Query	Google Hits
play station	16100000
play sport.	2680000
play gym	2430000
...	
play	Query Google Hits
to play sport	403000
to play croquet	119000
...	
to pla	Query Google Hits
to pla	Duchess to play croquet 11000
	Duchess to play station 0
	Duchess to play sport 0
	Duchess to play gym 0
	...

## problem:

- as  $n$  gets big, chance of finding sequence in corpus drops dramatically
- so must back off to shorter n-grams

'For the Duchess. An invitation from the Queen to play *croquet*.' The Frog-Footman repeated, in the same solemn tone, only changing the order of the words a little, 'From the Queen. An invitation for the Duchess to play ... ??'

$$P(\text{croquet} \mid \text{play}) = \frac{N(\text{play croquet})}{N(\text{play})}$$

$$P(\text{croquet} \mid \text{to play}) = \frac{N(\text{to play croquet})}{N(\text{to play})}$$

$$P(\text{croquet} \mid \text{Duchess to play}) = \frac{N(\text{Duchess to play croquet})}{N(\text{Duchess to play})}$$

In order to generate reasonable language

- need to model **very long distance dependencies**

Memory and data requirements:

- scale exponentially in length of observable dependency
- so **Markov models just don't scale**
  - nonetheless, they were still state-of-the-art not that long ago

Need instead methods that can both

- **generalise** from limited data
- handle **longer dependencies**

'For the Duchess. An invitation from the Queen to play croquet.' The Frog-Footman repeated, in the same solemn tone, only changing the order of the words a little, 'From the Queen. An invitation for the Duchess to play ... ???'

# brief history of Language Models

Markov models

> word embeddings

recurrent neural networks

attention

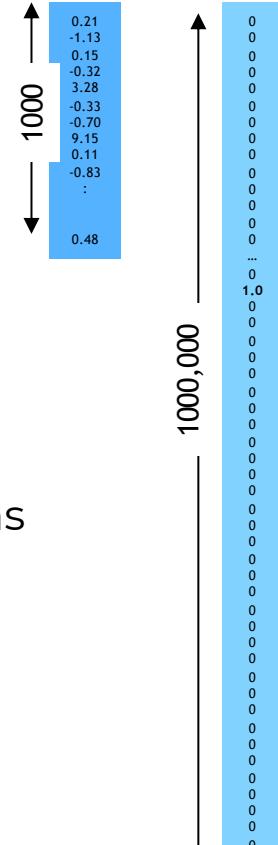
# Word Embeddings - what are they?

Word embeddings appeared around 2013 and improved performance on just about every NLP task

What are they?

- **dense vectors representing words** in a **high dimensional space**
- typically have between 100 & 1000 dimensions
- low dimensional compared to **one-hot encoding** of terms, since typical document collections have vocabularies of 100k to 1m tokens
- just like one-hot encodings they can be aggregated to represent sentences and documents

embedding vector      one-hot vector



Your task is to fill in the blank in the sentence:

'Sure Sally, let's have a skype call at 3pm \_\_\_\_\_ the 3<sup>rd</sup> of June.'

What word could fit here?

- prepositions: **on, by, before, around, near, ...**
- days of the week: **Monday, Tuesday, Wednesday, ...**
- timezones: **GMT, CET, EST, AEST, ...**

Note: very **few words fit** the context

- those that do come in **groups** that are **semantically related** to one other

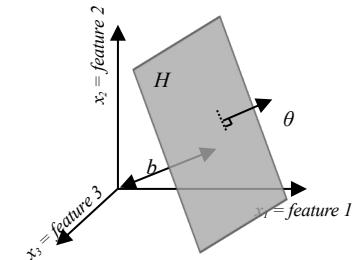
Embeddings are produced by **supervised machine learning models**

- models trained to **predict missing word** based on **surrounding context**
- context may include only previous words (causal models)
- or also future words (non-causal models)

# 🔍 Supervised Learning Problem

Predicting missing word:

- **Features:** words in current context:
  - “Sure”, “Sally”, “let’s”, “have”, “a”, “skype”, “call”, “at”, “3pm”
- **Target:** missing word from sequence
  - multi-class problem (estimate probability for every word in vocab)



Issue:

- requires a very large number of parameters!
- example:
  - multi-class linear classifier (e.g. Logistic Regression ) to predict all word in vocabulary
  - with bag-of-words feature vector (so ignoring word order)
  - requires parameters quadratic in the size of the vocab
  - if vocab=100 thousand, then we would have 10 billion parameters!!
  - which **used to be a lot** before deep learning came along 😂😂

**Word2Vec** developed in 2013 by Mikolov et al.

- following early work by Bengio et al. in 2003
- later GloVe in 2014 by Penington et al.

Word2Vec solved the parameter space issue by using:

1. bag-of-words representation
2. neural network with single (linear) hidden layer
3. training model in discriminative fashion
  - by inventing negative examples

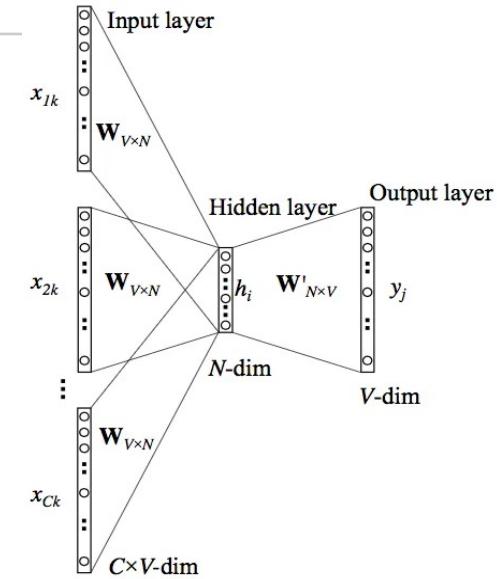
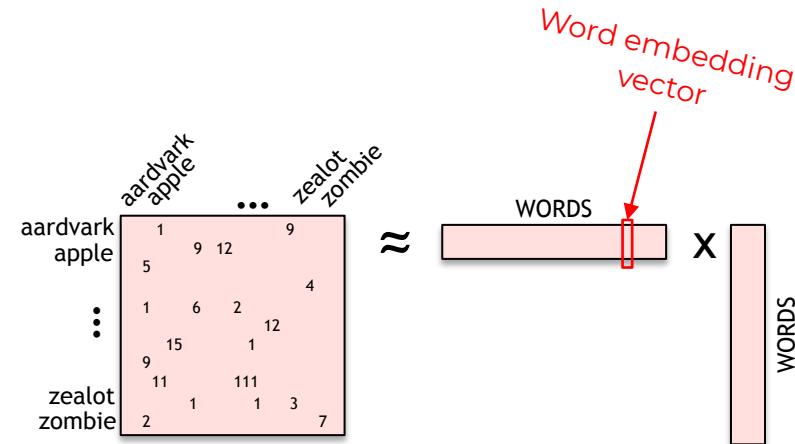


Image source: <http://www.stokastik.in/understanding-word-vectors-and-word2vec/>

# Just matrix decomposition

Word embeddings can be seen as a form of **matrix decomposition**

- square **count matrix**: vocabulary  $\times$  vocabulary
- contains word **co-occurrences** in text within a **fixed-size context window**
- factorizing **generalises** the information in those windows
- and produces word embedding vectors



# Properties of Word Embeddings

Word embeddings have interesting properties

- translation in the space meaningful
- semantics is additive

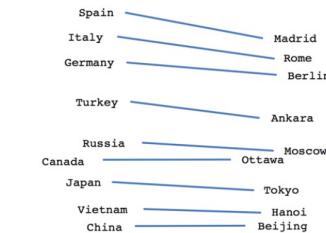
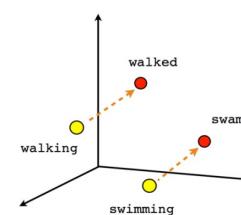
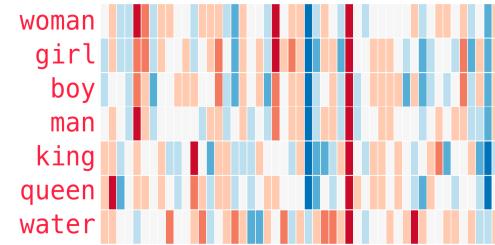
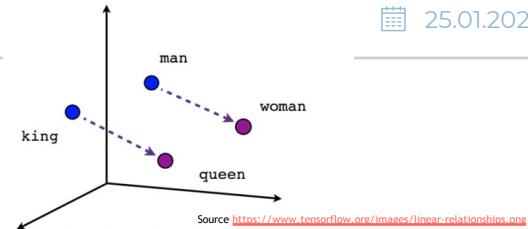
Semantic Clustering:

- neighbours in space are **semantically** related



Discovers **relationships** between words:

- e.g. part-of-speech, type-of, geographic, etc.



# Word embeddings and language modeling

Mark Carman  
25.01.2023

Low dimensional representation causes similar terms to share similar descriptions

- allows model to **generalize** from semantically **related examples**

the **Duchess** to play ... ??  
Examples from corpus with similar contexts:

- the **Queen** to play croquet
- the **Duke** to play chess

- e.g. **part-of-speech** and **hyponym** (type-of) relationships implicitly encoded in embedding vector in additive manner

the **Duchess** to play ... ??  
Look for examples with any of these types

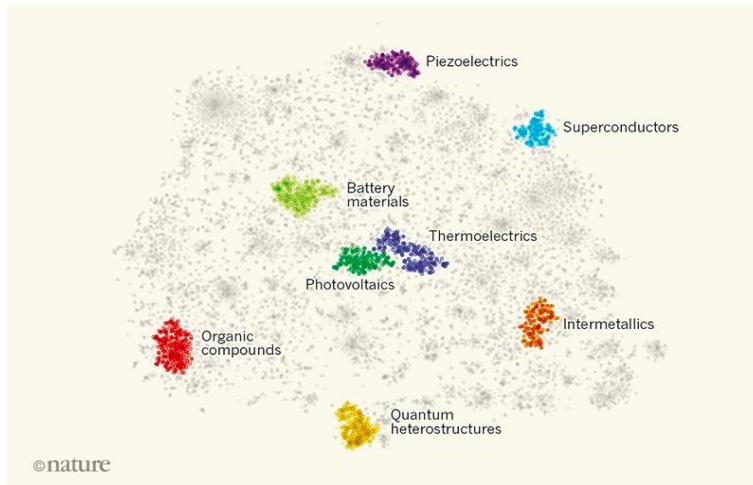
- the [noun+person+female+royal] to play ...

# Useful for Mining Text

Mark Carman  
25.01.2023

Embeddings place similar concepts close together

- useful for discovering implied (but unknown) properties of them



nature > letters > article

**nature**

Letter | Published: 03 July 2019

**Unsupervised word embeddings capture latent knowledge from materials science literature**

Vahe Tshitoyan, John Dagdelen, Leir...  
Olga Kononova, Kristin A. Pers...  
Nature 571, 95–98  
42k Abstracts

Subscribe Search Login

nature > news & views > article

**nature**

NEWS AND VIEWS · 03 JULY 2019

**Text mining facilitates materials discovery**

Oleksandr Isayev

Computer algorithms can be used to analyse text to find semantic relationships between words without human input. This method has now been adopted to identify unreported properties of materials in scientific papers.

PDF version

The total number of materials that can potentially be made – sometimes referred to as materials space – is vast, because there are countless combinations of components and structures from which materials can be fabricated. The accumulation of experimental data that represent pockets of space has created a foundation for the emerging field of materials science, which generates high-throughput experiments, computations and feedback loops that enable rational design. A report that knowledge of

RELATED ARTICLES

Read the paper: [Unsupervised word embeddings capture latent knowledge from materials science literature](#)

25

# 🔍 Sub-word embeddings

Word embeddings work well if vocabulary is fixed

- so **no new words** in test set
- if we see a new word, don't have embedding for it!

## Fasttext ([2016 Bojanowski et al.](#))

- split words into character sequences
- learns embeddings for character n-grams
- combines the embeddings to form words

Advantage:

- deals nicely with morphologically related terms, so:
  - “**believe**” and “**believing**” have similar representations
  - as do “**rain**” and “**rainfall**”

Embeddings are cool.

=>

<Em

Emb

mbe

bed

edd

ddi

din

ing

ngs

gs>

<ar

are

re>

<co

coo

ool

ol>

# brief history of Language Models

Markov models

word embeddings

> recurrent neural networks

attention

# 🔍 Recurrent Neural Networks (RNNs)

Now have way to **represent words** in a semantic space ✓👍

- still need to **aggregate information** over longer contexts

RNNs provide general way to accumulate information

- by combining **context** from previous words
- with the **embedding** of current word

RNNs are simply models which:

- take 2 vectors as input: <current input, previous state>
- produce 2 vectors as output: <current output, updated state>

They can be used to process **arbitrarily long** input contexts

- i.e. encode a sequence of text to a single embedding

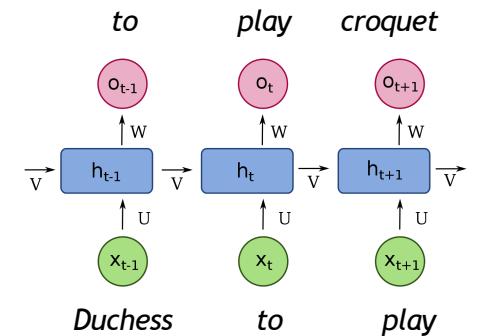
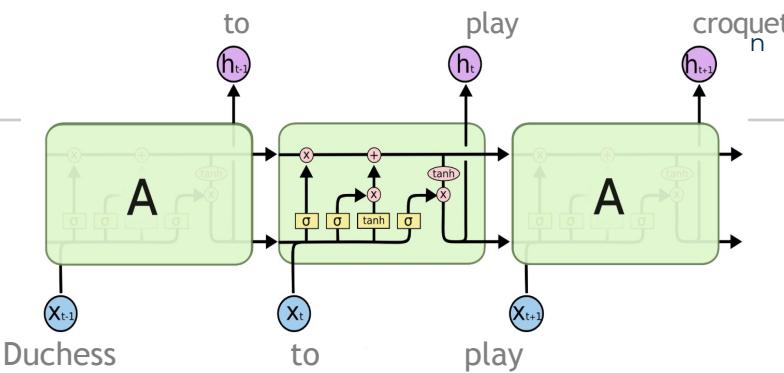


Image source:  
[https://en.wikipedia.org/wiki/Recurrent\\_neural\\_network](https://en.wikipedia.org/wiki/Recurrent_neural_network)

## Long Short Term Memory (**LSTM**) networks

- allow for **long term dependencies**
- gates allow model to **remember/forget** information
- model learns what type of information to keep and what to discard
- default** operation is to pass information from one state to the next



Images source: Understanding LSTM Networks by Christopher Olah  
<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

'For the Duchess. An invitation from the Queen to play croquet.' The Frog-Footman repeated, in the same solemn tone, only changing the order of the words a little, 'From the Queen. An invitation for the Duchess to play ... ???'

# Stacked LSTMs and contexts

Stacked LSTMs: layers of LSTMs placed on top of each other

- have uncanny ability to deal with **nested contexts**
- they won't forget to close the brackets at the end of a maths expression

PDF compiled from Latex source code that was generated by a multi-layer LSTM (by Andrej Karpathy)

<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>

For  $\bigoplus_{n=1,\dots,m} \mathcal{L}_{m,n} = 0$ , hence we can find a closed subset  $H$  in  $\mathcal{H}$  and any sets  $\mathcal{F}$  on  $X$ ,  $U$  is a closed immersion of  $S$ , then  $U \rightarrow T$  is a separated algebraic space.

*Proof.* Proof of (1). It also start we get

$$S = \text{Spec}(R) = U \times_X U \times_X U$$

and the comparicoly in the fibre product covering we have to prove the lemma generated by  $\coprod Z \times_U U \rightarrow V$ . Consider the maps  $M$  along the set of points  $Sch_{fppf}$  and  $U \rightarrow U$  is the fibre category of  $S$  in  $U$  in Section ?? and the fact that any  $U$  affine, see Morphisms, Lemma ???. Hence we obtain a scheme  $S$  and any open subset  $W \subset U$  in  $Sh(G)$  such that  $\text{Spec}(R') \rightarrow S$  is smooth or an

$$U = \bigcup U_i \times_{S_i} U_i$$

which has a nonzero morphism we may assume that  $f_i$  is of finite presentation over  $S$ . We claim that  $\mathcal{O}_{X,s}$  is a scheme where  $x, x', s'' \in S'$  such that  $\mathcal{O}_{X,x'} \rightarrow \mathcal{O}_{X',x'}$  is separated. By Algebra, Lemma ?? we can define a map of complexes  $GL_{S'}(x'/S'')$  and we win.  $\square$

To prove study we see that  $\mathcal{F}|_U$  is a covering of  $\mathcal{X}'$ , and  $\mathcal{T}_i$  is an object of  $\mathcal{F}_{X/S}$  for  $i > 0$  and  $\mathcal{F}_p$  exists and let  $\mathcal{F}$  be a presheaf of  $\mathcal{O}_X$ -modules on  $\mathcal{C}$  as a  $\mathcal{F}$ -module. In particular  $\mathcal{F} = U/\mathcal{F}$  we have to show that

$$\tilde{M}^\bullet = \mathcal{I}^\bullet \otimes_{\text{Spec}(k)} \mathcal{O}_{S,s} - i_X^{-1} \mathcal{F}$$

is a unique morphism of algebraic stacks. Note that

$$\text{Arrows} = (Sch/S)_{fppf}^{opp}, (Sch/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \longrightarrow (U, \text{Spec}(A))$$

is an open subset of  $X$ . Thus  $U$  is affine. This is a continuous map of  $X$  is the inverse, the groupoid scheme  $S$ .

*Proof.* See discussion of sheaves of sets.  $\square$

The result for prove any open covering follows from the less of Example ???. It may replace  $S$  by  $X_{spaces, \acute{e}tale}$  which gives an open subspace of  $X$  and  $T$  equal to  $S_{Zar}$ , see Descent, Lemma ???. Namely, by Lemma ?? we see that  $R$  is geometrically regular over  $S$ .

The ability to deal with contexts is useful for natural language too

- consider gender and possessive pronouns:

- fill in gaps in following sentences with one of {he, she, his, or her}

**My mother was taking on the phone to \_\_ friend Jim. Jim said that a favourite game of \_\_ is confusing students. Replying, my mother said that \_\_ should find a better hobby.**

- answer: ("her", "his", "he"), because mother is feminine and Jim is masculine.
  - LSTM is able to associate correct gender with each position.
- consider sentiment and scoping of negation:

- What's more likely to come next in these sentences: "**friendly**" or "**moo**dy"?

**I get along well with her brother. He's always so \_\_**

**I don't get along well with her brother. He's always so \_\_**

**I don't know how anyone could not get along well with her brother. He's always so \_\_**

# brief history of Language Models

Markov models

word embeddings

recurrent neural networks

> attention

# Sequence to sequence (seq2seq) models

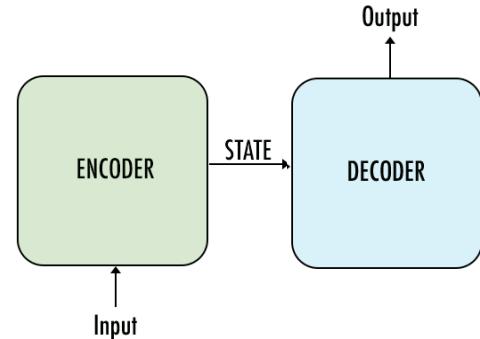
Mark Carman  
25.01.2023

LSTMs are so powerful that they were soon used for **translation** and **dialog systems**

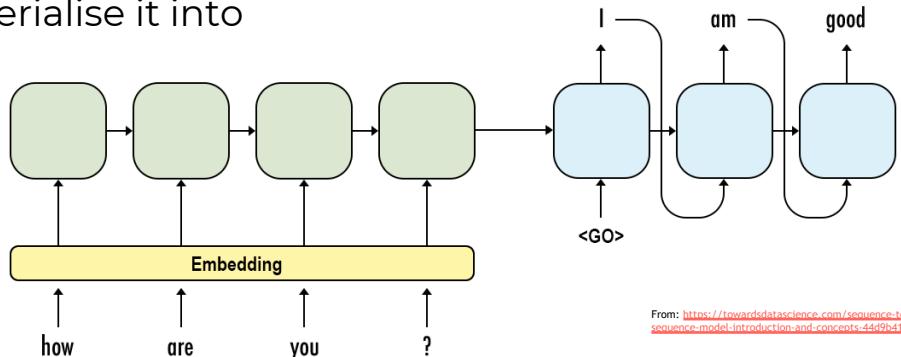
How can one learn translation models with LSTMs?

By training 2 different RNN models:

1. **encoder**: reads in input text and generates a representation for the entire sequence
2. **decoder**: takes output of encoder and serialise it into new text one word at a time



From: <https://towardsdatascience.com/sequence-to-sequence-model-introduction-and-concepts-44d9b41cd42d>



From: <https://towardsdatascience.com/sequence-to-sequence-model-introduction-and-concepts-44d9b41cd42d>

**Attention** is a critical building block for modern image and text processing

- what is attention?
- why it is implemented?
- how does it work?

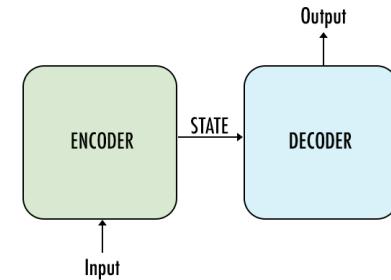


**WASHINGTON**  
In what is being called a crucial step forward in solar exploration NASA officials announced Wednesday a new mission to launch a chimpanzee directly into the sun. Chimpanzees are our closest biological relative so we can learn a great deal by observing how they react to being deposited into the sun's plasma core said NASA Administrator Charles Bolden adding that the single occupant capsule would contain sophisticated instruments that would monitor the effects of the sun's 27 million degree interior on the physiological functions of the animal. Hopefully what we learn from this mission will pave the way for sending human astronauts into the sun on a regular basis. Bolden went on to suggest that should humans be successfully launched into the sun there may one day be a permanent colony there.

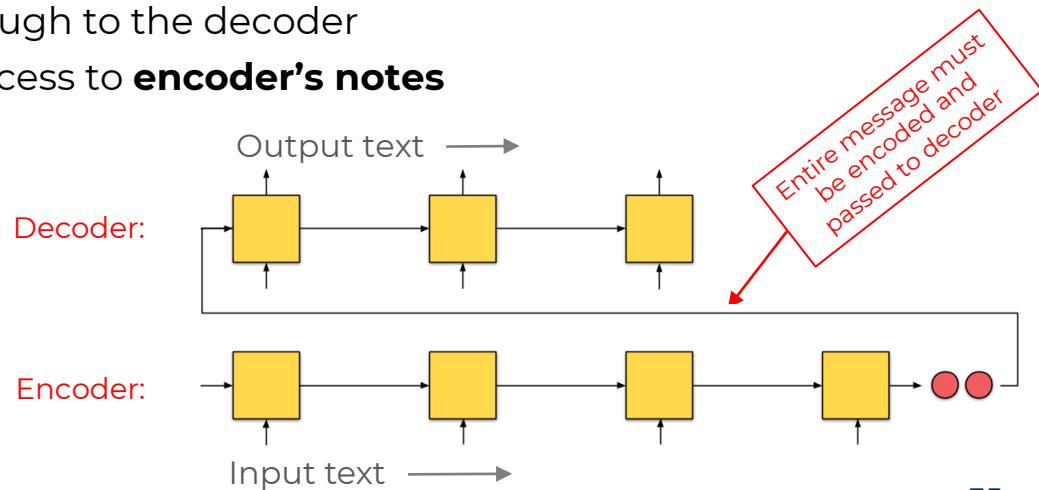
# 🔍 A simpler way ...

Ever thought about poor interpreters translating politicians during meetings?

- must wait for rambling politician to stop speaking before they can start translating...
- that's a lot of stuff to remember!
- same problem for encoder-decoder architecture
  - **too much information** to pass through to the decoder
  - easier to translate if decoder has access to **encoder's notes**



From: <https://towardsdatascience.com/sequence-to-sequence-model-introduction-and-concepts-44d9b41cd42d>



Source: <https://towardsdatascience.com/attn-illustrated-attention-5ec4ad276ee3>

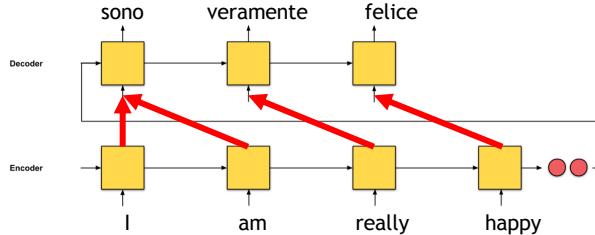
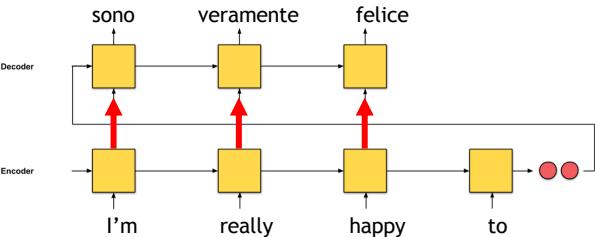
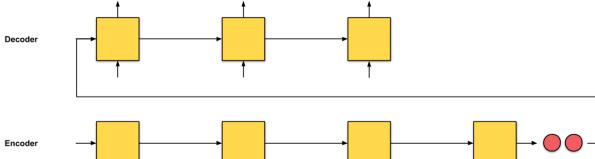
# Towards attention

## Attention models

- make encoded input available to decoder
- provides a direct route for the information to flow from input to output

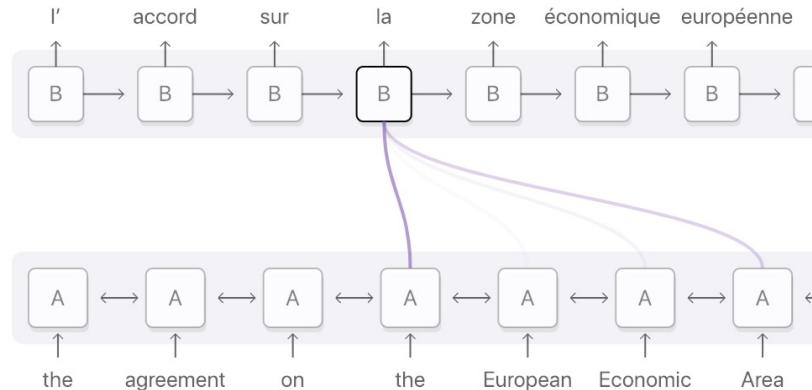
Why not directly map input words to output words?

- because across languages different number of tokens required to describe same concept
- and different word order is often used:  
“United Nations General Assembly”  
=> “Assemblea Generale delle Nazioni Unite”



Adapted from: <https://towardsdatascience.com/attn-illustrated-attention-5ec4ad276ee3>

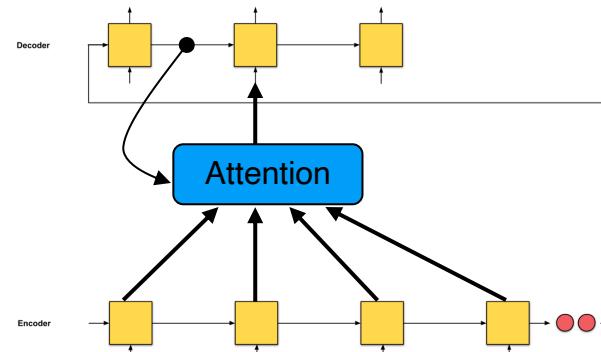
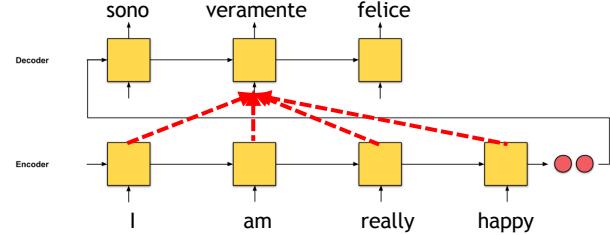
- Moreover generating the right output word often requires knowing more than just the current word in the input
- Indeed it can require knowing the value of a future word



Source: <https://towardsdatascience.com/attn-illustrated-attention-5ec4ad276ee3>

# Need for attention

- Need mechanism to pass information from embeddings of input words to corresponding output word
- Attention provides a direct route for the information to flow from input to output
- What information flows into the decoder is controlled by the previous state of the decoder



Source: <https://towardsdatascience.com/attn-illustrated-attention-5ec4ad276ee3>

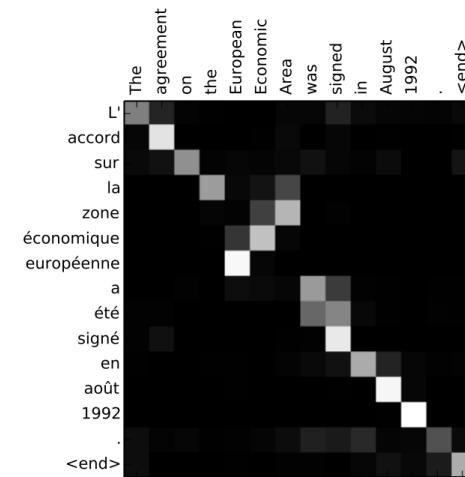
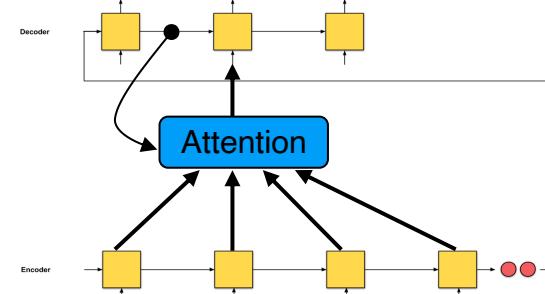
# 🔍 Soft attention over input

Similarity is computed between state of decoder and embedding of each term

- embedding of input term then weighted by score

**soft-attention** produces **weighted average** over input embeddings

- Example of Bahdanau attention for English to French translation shown



Source 2015 paper by Bahdanau et al.  
<https://arxiv.org/pdf/1409.0473.pdf>

# Transformers for text

- > deep learning
- tokenization
- self-attention
- BERT vs GPT-2
- what can we do with deep models?

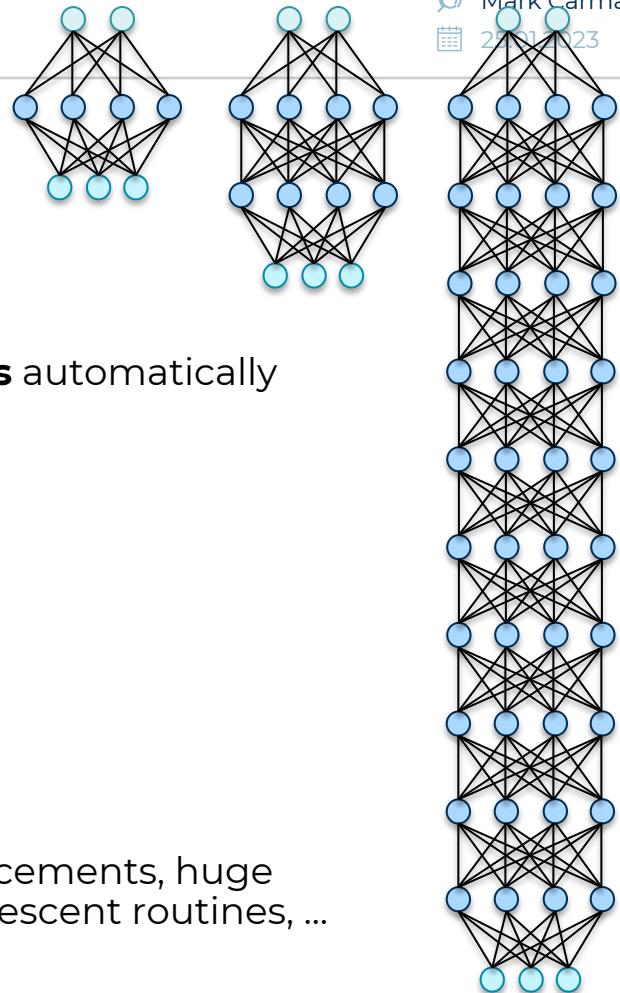
# What is deep learning?

Deep Neural Networks with **MANY** layers

- why do we want many layers?
  - massively **improves performance**, but why?
  - allows network to **learn hierarchy of useful features** automatically
- what are the downsides?
  - need **lots of training** data!
  - and large **computing resources** (GPUs)
  - can be unstable & **much harder** to train ...
  - and even with GPUs, take a **long time** to train

Why is deep learning so big now?

- provides **amazing performance** on text and images
- simplifies training pipeline
- possible due to specialized toolkits, hardware advancements, huge data quantities, clever architectures, new gradient descent routines, ...



# Why is deep learning important for text?

Mark Carman  
023

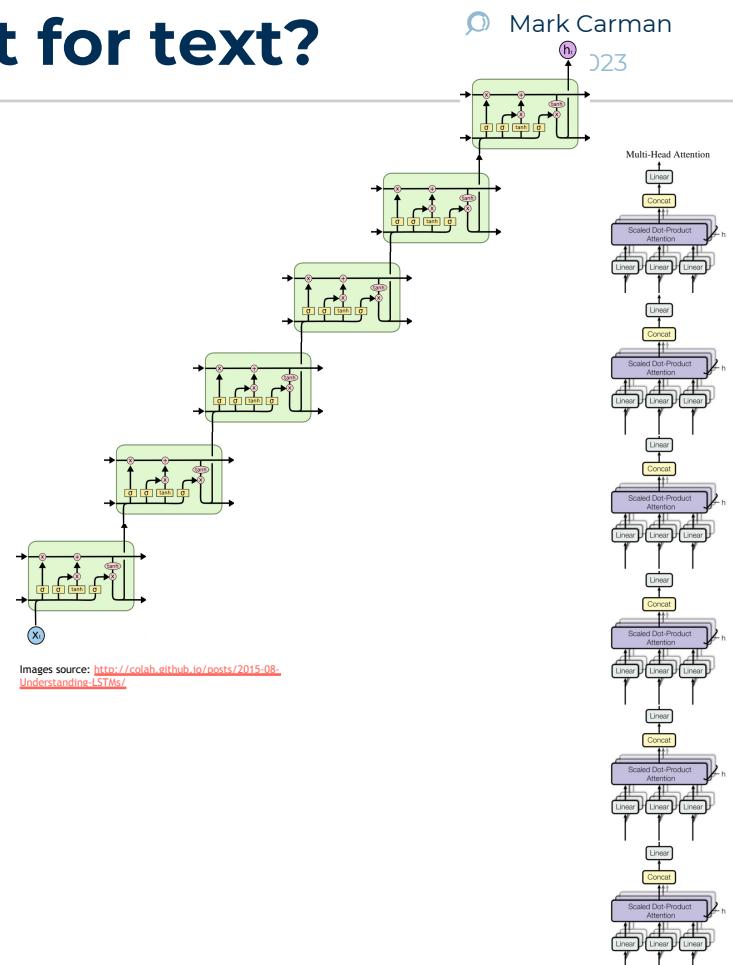
**State-of-the-art performance** for most text processing tasks

- including **classification, summarisation, generation, translation, etc.**

Up until relatively recently (2017) deep architectures involved stacking many layers of LSTMs on top of each other (e.g. ELMO)

In 2017 a new architecture has emerged, called a Transformer

- makes use of **self-attention** networks



Images source: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

# Inside Transformer

Basic self-attention module is stacked on itself **many** times

- allows **semantics of each token** to build up over multiple steps

Each transformer module contains:

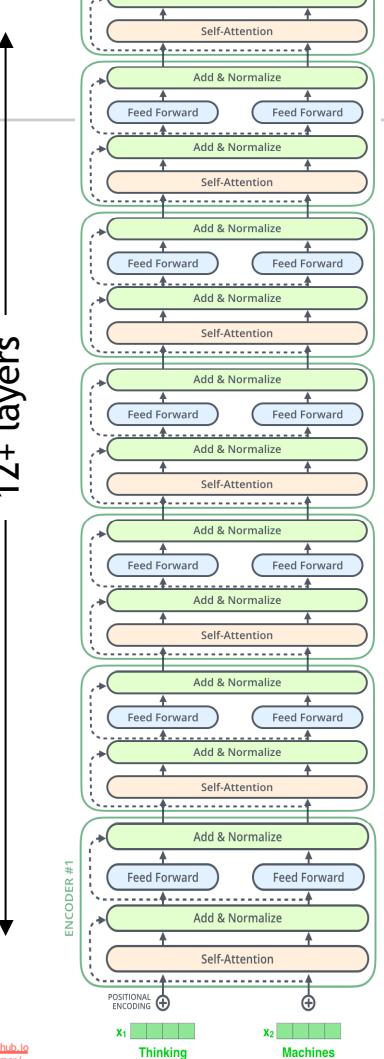
- **multiple attention heads** working in parallel
- **feedforward network**, with **residual connections** and **normalisation**

Architecture is word position agnostic

- so **positional encoding** provided as additional input to bottom layer

Note: transformers are MUCH **faster to train** than stacks of RNNs

- in RNNs gradient must be iterated back along sequence



# Transformers for text

deep learning

> tokenization

self-attention

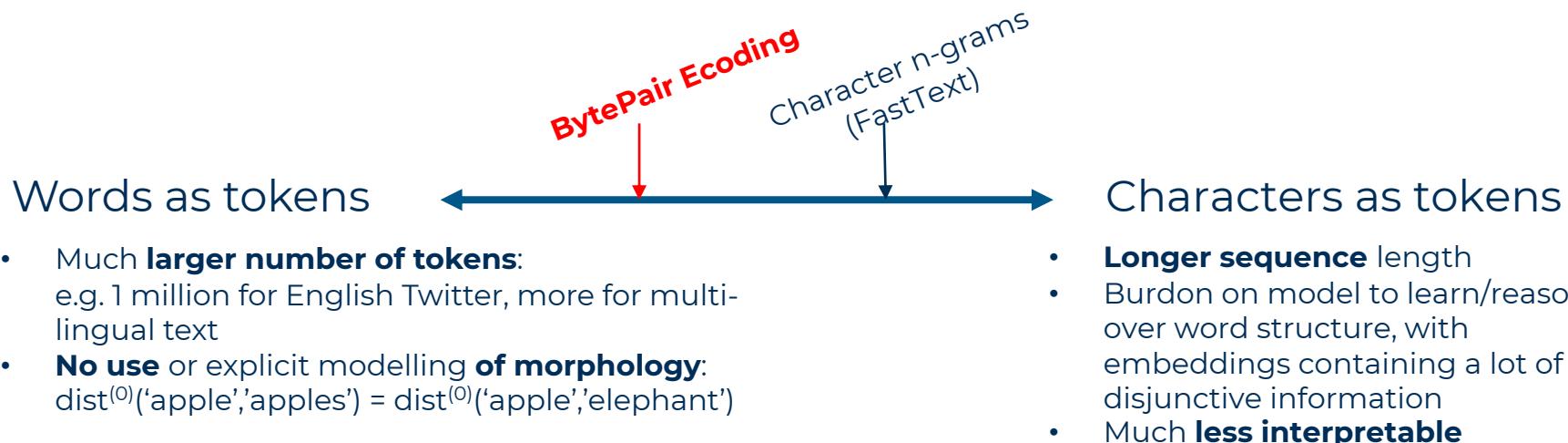
BERT vs GPT-2

what can we do with deep models?

# 🔍 Tokenization: word or character-level?

Is it better to use **word-level** or **character-level representations** in the model?

- choice depends on computational trade-offs:
  - expressivity of sequence vs sequence length (inference complexity)
- and also what the language is:
  - Chinese has logograms at the syllable rather than character level (unless convert to pinyin)
  - DNA/Protein sequences have few characters and no word spaces



# 🔍 Transformers use sub-word tokens

How should we break up words into sub-word tokens? Use the data!

- find **frequent character sequences** by performing a **byte-pair encoding**
- iteratively replace most frequent consecutive characters by new characters

though they think that the thesis is thorough enough  
**th** →  $\theta$ :

$\theta$ ough  $\theta$ ey  $\theta$ ink  $\theta$ at  $\theta$ e  $\theta$ esis is  $\theta$ orough enough

**ou+g+h** →  $\epsilon$ :

$\theta\epsilon$   $\theta$ ey  $\theta$ ink  $\theta$ at  $\theta$ e  $\theta$ esis is  $\theta$ or $\epsilon$  en $\epsilon$

**$\theta$ e** →  $\psi$ :

$\theta\epsilon$   $\psi$ y  $\theta$ ink  $\theta$ at  $\psi$   $\psi$ sis is  $\theta$ or $\epsilon$  en $\epsilon$

- in this way, common prefixes/suffixes become vocabulary elements:

Input sentence: 'I like playing football.'

Word level tokenization: 'I', 'like', 'playing', 'football', ''

Sub-word level tokenization: 'I', 'like', 'play', '#ing', 'foot', '#ball', ''

# Transformers for text

what is deep learning?

> self-attention and transformers

BERT vs GPT-2

what can we do with deep models?

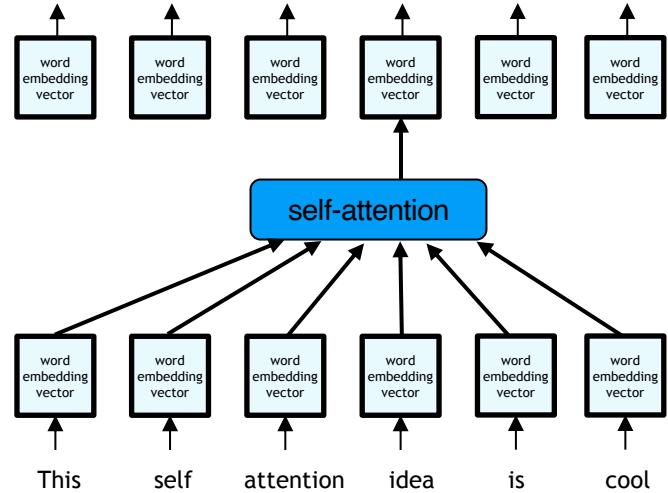
# So what is self-attention?

Self attention is a mechanism for:

- combining word embedding vectors to produce new word embedding vectors
- each high-level embedding is **weighted average** of word embeddings below it

Weights are computed:

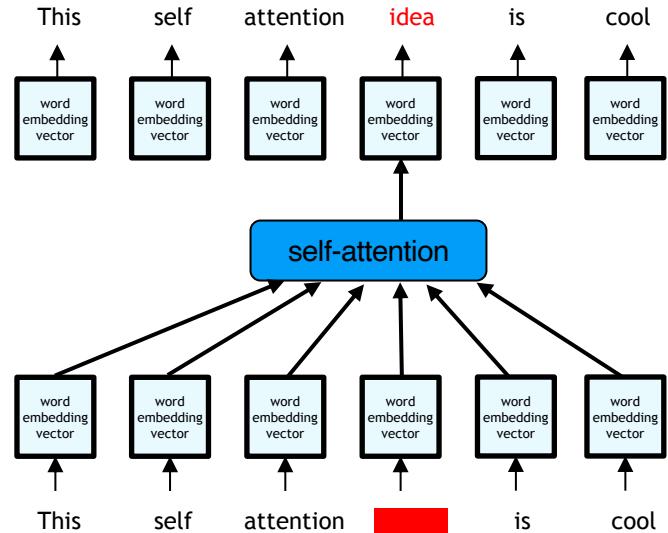
- based on similarity between embeddings in respective positions
- model parameters control the process, learning how best to compute the weights



# 🔍 So what is self-attention?

Self attention models are trained to recover missing words from the input sentence

- i.e. to perform the language modeling task



# Motivating the need for self-attention

To understand why self-attention is useful for language models, consider that

- words take on **different meanings** depending on their context
- attention mechanism allows representation to **depend on context**
- learns **weighting function** over lower-level embeddings of context terms

Sentence 1: *I arrived at the bank after crossing the street.*  
Sentence 2: *I arrived at the bank after crossing the river.*

What type of bank are we talking about here, a river bank or a financial bank?

Level 1: *I arrived at the bank after crossing the river.*

Information from embedding for 'river' passes to 'bank' reinforcing the interpretation 'river bank'

Level 0: *I arrived at the bank after crossing the river.*

initial embedding for term 'bank' contains both meanings: 'river bank' and 'financial bank'

Initial embedding for 'river' and 'bank' should share similar interpretations leading to high self attention weight for word river

# Motivating self-attention

Complicated tasks like coreference resolution can be handled quite effectively with multiple layers of self-attention:

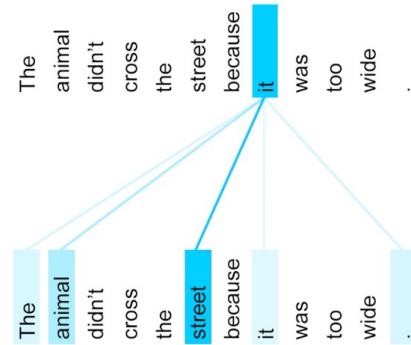
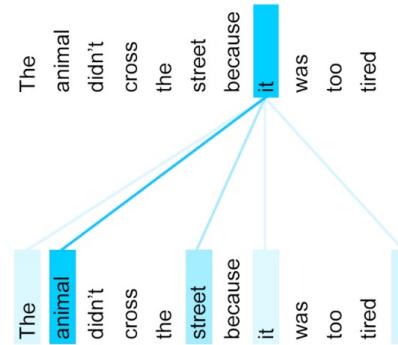


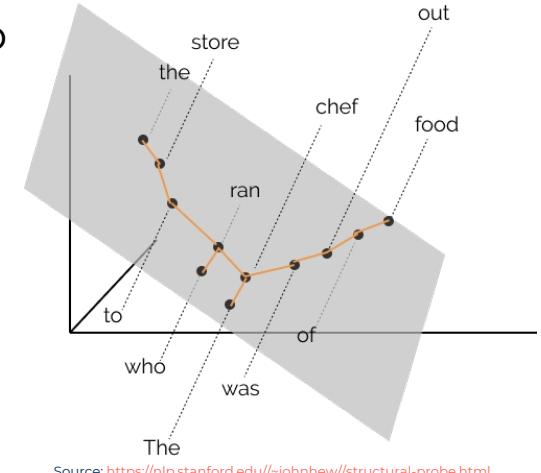
Image source: "Transformer: A Novel Neural Network Architecture for Language Understanding", by Uszkoreit et al. <https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html>

# Why is stacked self-attention so useful?

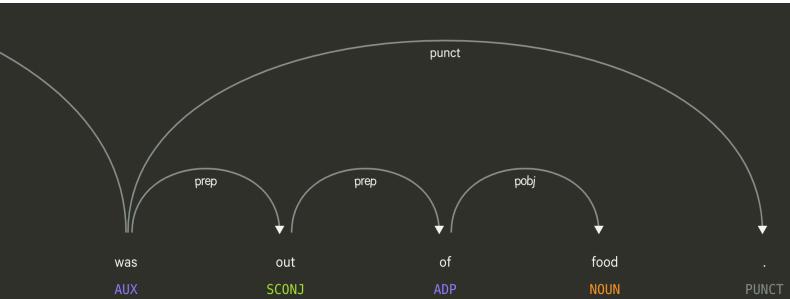
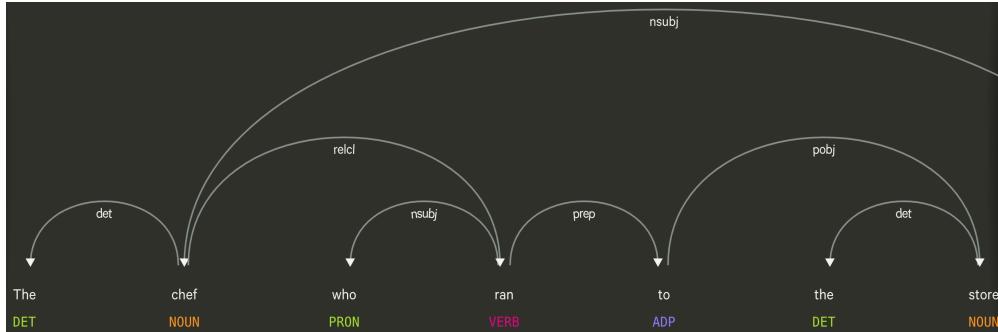
It's been argued that Transformers **effectively learn** how to build a **dependency parse tree** over concepts in the text

Hewitt et al. 2019 <https://nlp.stanford.edu/pubs/hewitt2019structural.pdf>

- Consider the examples:
  - The store was out of food. \_\_
  - The chef who ran to the store was out of food. \_\_
- To predict next sentence, need to know who is out of food
  - was it the store or the chef?



Source: <https://nlp.stanford.edu/~jhewitt/structural-probe.html>

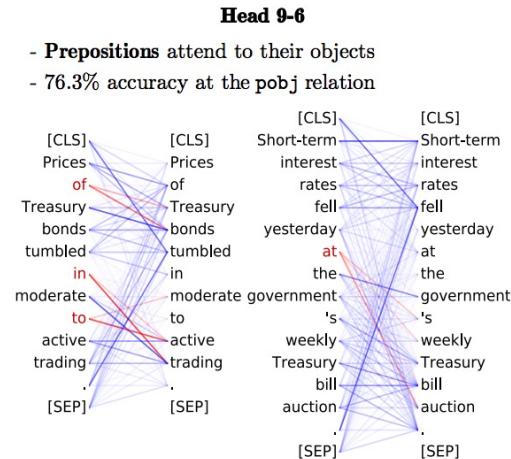
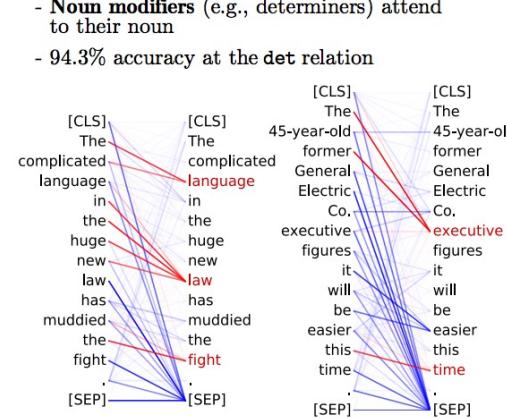
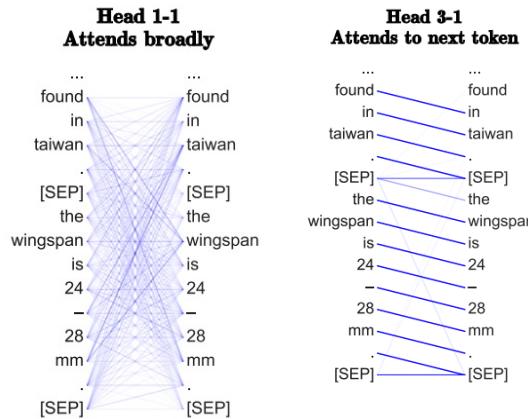


Source: [https://explosion.ai/demos/display/?text=The%20chef%20who%20ran%20to%20the%20store%20was%20out%20of%20food&model=en\\_core\\_web\\_sm&cpu=0&cpu=0](https://explosion.ai/demos/display/?text=The%20chef%20who%20ran%20to%20the%20store%20was%20out%20of%20food&model=en_core_web_sm&cpu=0&cpu=0)

# Why is stacked self-attention so useful?

Lots of visualisation is going on trying to interpret what is being learnt.

- some heads simply *aggregate information* or *attend to a previous token*
- others *learn language relationships* ([2019 paper by Clark et al.](#))
  - see demo: [https://colab.research.google.com/drive/1PEHWRHrvxQvYr9NFRC-E\\_fr3xDq1htCj](https://colab.research.google.com/drive/1PEHWRHrvxQvYr9NFRC-E_fr3xDq1htCj)



Source: <https://arxiv.org/pdf/1906.04341.pdf>

# Transformers for text

what is deep learning?

self-attention

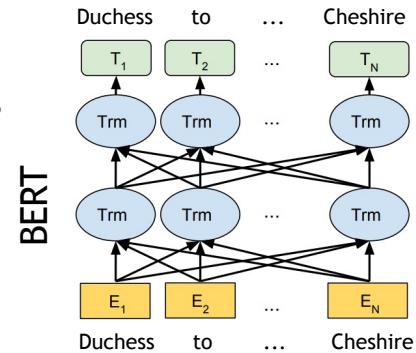
> BERT vs GPT-2

what can we do with deep models?

# 🔍 Story of two architectures

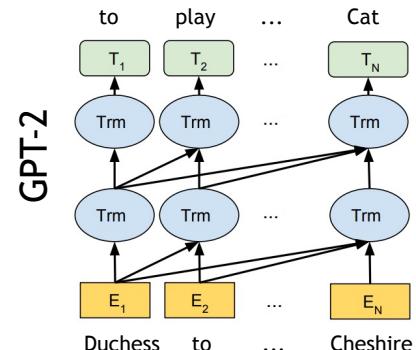
**BERT** = Bidirectional Encoder Representations from Transformers

- 2019 paper by Devlin et al. (Google)
- **Autoencoder**: text in at bottom, same text comes out at top
- Recovers input text
- Great for **representing text** (e.g. for building classifiers)



**GPT-2** = Generative Pretrained Transformer (Version 2)

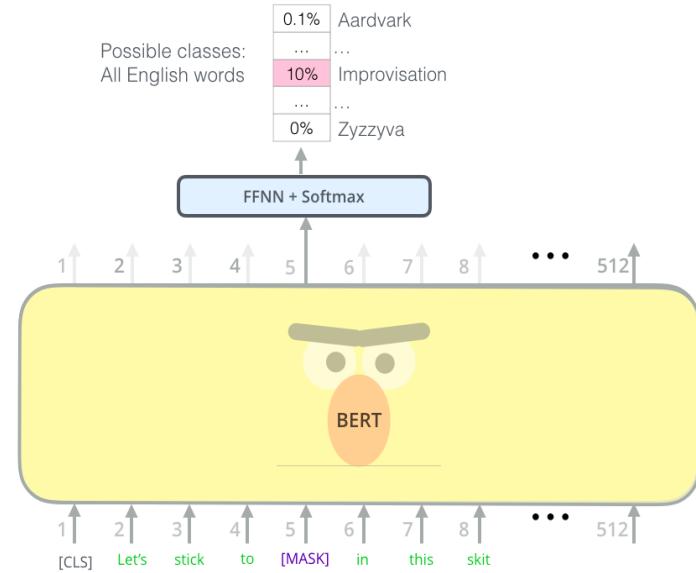
- 2019 paper by Radford et al. (OpenAI)
- **Autoregressive**: text in at bottom, text shifted one to the left
- Predicts the next token
- Great for **generating text**



# How are BERT and GPT trained?

BERT:

- by **masking out** random words in the input using a special [MASK] token
- model must recover all words including the masked ones



Source: <http://jalammar.github.io/illustrated-bert/>

GPT-2

- by simply **masking future words** in the sequence and at each point predicting the next word

# 🔍 What data were they trained on?

🔍 Mark Carman  
📅 25.01.2023

Garbage in => garbage out

- model will produce similar text to that which it was trained on

GPT-2:

- trained on 40GB of web text that Reddit users rated highly

BERT:

- trained on Wikipedia and a corpus of books



Image source: <https://en.wikipedia.org/wiki/Reddit>



Image source: <https://en.wikipedia.org/wiki/File:Wikimedia-Wikipedia-Logo-v2.svg>

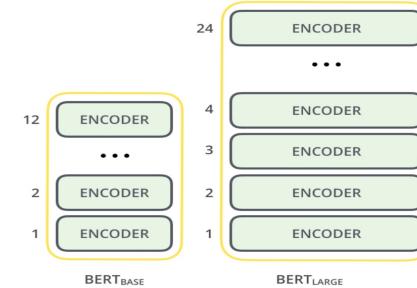
# Transformer sizes

Transformers come in multiple sizes, depending on

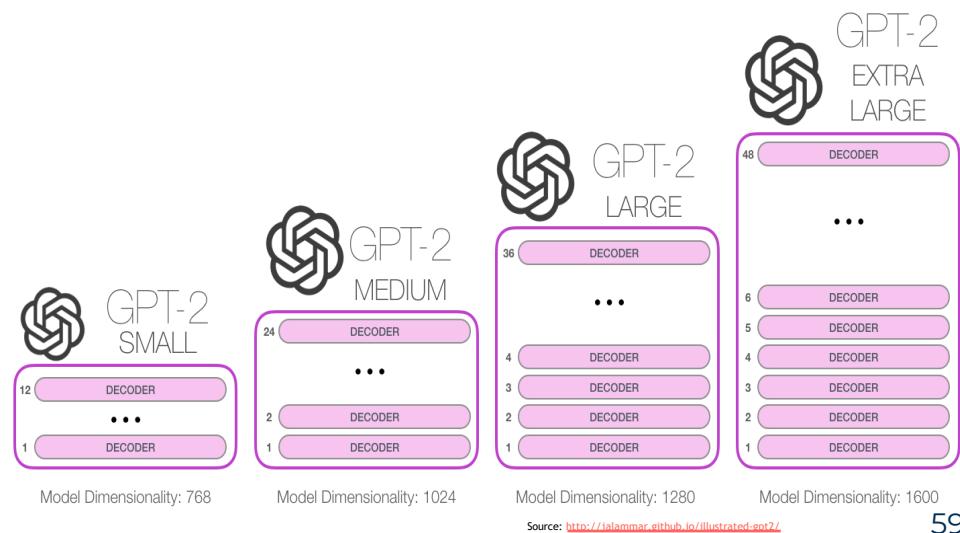
- number of self-attention layers
- size of the embedding used at each layer
- number of parallel attention heads

More parameters results in:

- better performance
- but longer training times
- and larger memory requirements



Source: <http://jalammar.github.io/illustrated-bert/>



# 🔍 How many parameters is that?

👤 Mark Carman  
📅 25.01.2023

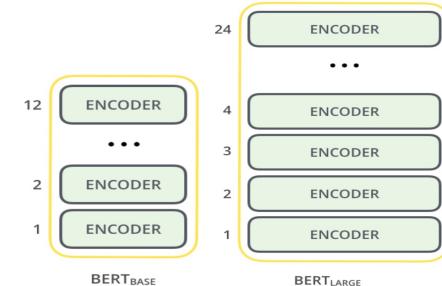
Sizing for BERT models:

- base model has 110M parameters
- large model has 340M parameters

Sizing for the GPT-2 models:

- largest has **1.5 billion** parameters!
- vocabulary of 50,257

Most applications use a **context size** of **500** or **1000 tokens**



Source: <http://jalammar.github.io/illustrated-bert/>

Parameters	Layers	$d_{model}$
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

Table 2. Architecture hyperparameters for the 4 model sizes.

From: [https://d4mucfokswv.cloudfront.net/better-language-models/language\\_models\\_are\\_unsupervised\\_multitask\\_learners.pdf](https://d4mucfokswv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf)

## Variants on BERT:

- **RoBERTa** (Facebook's version of BERT)
  - modifies slightly training objective
  - trained on more data with larger batches
- **XLNet** (BERT with some GPT-2)
  - introduces autoregressive modelling (GPT-2) into BERT training
  - was quite hyped for a while
- **DistilBERT** (a distilled version of BERT)
  - designed to be smaller (40%) and faster (60%) to fine-tune, while retaining 97% of accuracy
- Huge number of **other variations** out there
  - Many **pretrained** for **specific text domains**, e.g. medical literature
  - Many **finetuned** for **specific tasks**, e.g. question answering

Domain specific and text2text  
models models are often  
worth investigating

## Encoder-Decoder model:

- **T5** (Text-To-Text Transfer Transformer)
  - uses encoder+decoder model, same as the original transformer paper
  - uses clever **relative** positional encoding
  - so **particularly useful** for **translation** or other **text2text problems**

# Transformers for text

what is deep learning?

self-attention and transformers

BERT vs GPT-2

> what can we do with deep models?

# what can we do with deep models?

Text Generation

Fine Tuning

One-shot Learning

# GPT-2 is a text generator

Mark Carman  
25.01.2023

## Context (WebText test)

Corporal Michael P. Goeldin was an unskilled laborer from Ireland when he enlisted in Company A in November 1860. Goldein survived the war. Corporal Patrick O'Neal, also from Ireland, first enlisted in 1854 and served with Company L, 3d U.S. Artillery, in Oregon. He returned to the East Coast and enlisted in the company in 1860. O'Neal served until 1874, when he was named superintendent of the National Cemetery at Willets Point, New York. Corporal Benjamin Browne was a shoemaker from Orange County, New York. In August 1862, he enlisted in the newly formed 124th New York Volunteers, and was one of sixty-one men who transferred into Company A that October. Browne reenlisted in the company in February 1864 while it was camped at Brandy Station. He returned to civilian life after completing his enlistment in 1867.

On 10 June, Artificer William Collins was promoted to corporal, probably to fill a combat leadership void for the crossing of the James River. Collins's service record does not reflect the qualities he demonstrated to earn this promotion, but he had obviously overcome some serious problems. Born in Sacketts Harbor, New York, Collins enlisted in the company in December 1853 at the age of twenty-two, and reenlisted in December 1858. Just a month before the war began in April 1861, Collins went "over the hill" and was not caught until three years later. Returned to the company on 22 March 1864, he was tried

Conditioned on **large** piece of text

- model can generate new content

Text generation is building block for many applications

- more on that in a moment ...

## GPT-2 completion

in June of that year. Sentenced to be shot for desertion, he remained on active duty.

Captain James J. Dall was born in Virginia in 1829 during the Revolution and joined his father in the Army in 1836. He served with the 2d Virginia Infantry until 1845, when he went to the Western Reserve Regiment; and, on his discharge, he reenlisted with Company I, 25th New York Volunteers (which in 1845 numbered about 4,500 men). One of his brothers, James, served with his company in the same capacity in the same brigade. While the regiment remained in Connecticut through 1846, Captain Dall served with Company I at various times before his discharge. A veteran of the Mexican War, he had

# what can we do with deep models?

Text Generation

Fine Tuning

One-shot Learning

Transformers allow for **transfer learning** with text

- model **comes pre-trained** on enormous quantities of data
- **fine-tune** model on specific task for which little data is usually available

Transfer learning for text can lead to:

- big **improvements in performance**
  - better performance from **small training corpus** since model leverages unsupervised pretraining to learn language model
- **multi-linguality:**
  - Multilingual BERT has been pretrained on **104 languages!**
  - can **train** classification model **on English documents**, use it on Italian ones!
    - transfer learning across languages was possible but practically unheard of before
    - amazing possibilities for resource poor languages

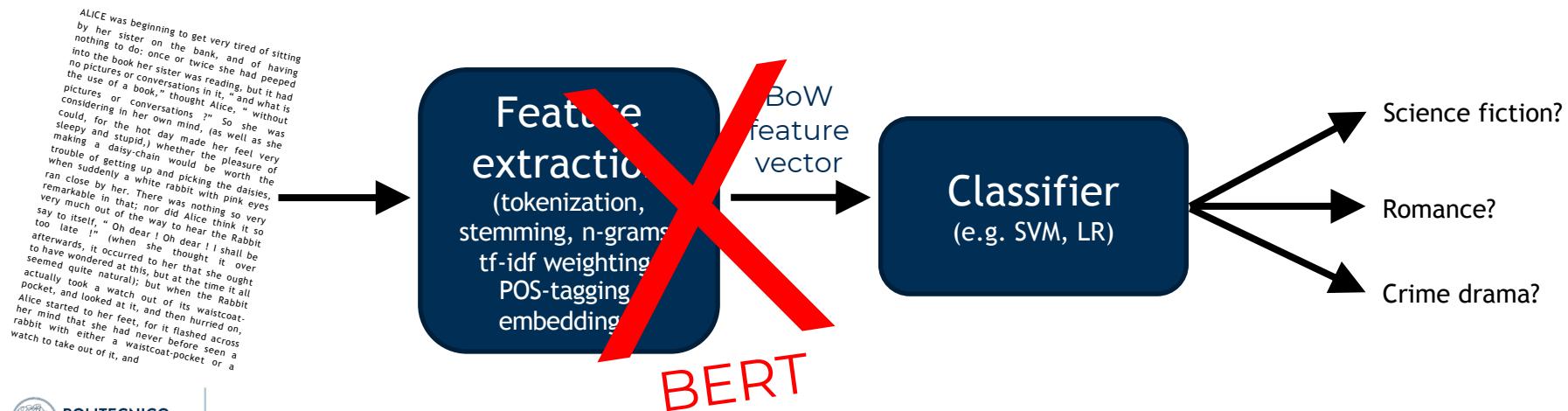
Traditionally, to build a text classifier:

- first decide on types of features to extract from text (e.g. n-gram counts)
- and how to process them (e.g. stemming, idf-weighting, add PoS tags, etc.)

BERT removes the feature extraction step

Moreover, performance improvements likely over count-based features

- since BERT leverages unsupervised pre-training (language modelling)
- and doesn't discard word order



# 🔍 Why would I want to classify text?

👤 Mark Carman  
📅 25.01.2023

Improvements in text classification is a big deal

- because it is an **extremely common** task to need to perform
  - Email spam detection
  - Authorship identification
  - Sentiment analysis in product reviews
  - Offensive content detection
  - Web search query intent identification
  - Creating your news feed on Facebook/LinkedIn
  - Identifying criminal behaviour online (fraud, grooming,...)
  - Routing communications to the right person
  - Parsing requests to spoken interfaces (Alexa, Siri, ...)
  - ...
- in fact text classifiers control much of the content you see online

Download large **pre-trained language model**

- BERT model choose type
  - lowercase?
  - multilingual?
  - how big?

Fine tune the model for your task on your labelled training data

- spend time **tuning the learning rate**
  - to make sure your model is learning and doesn't collapse or overfit training data
- want to try it out yourself?
  - Open this Google Colab page, and create a copy:  
[https://drive.google.com/file/d/19UckYpcQeuZB3\\_T\\_D1QzhaltuXbpv\\_VH/view?usp=sharing](https://drive.google.com/file/d/19UckYpcQeuZB3_T_D1QzhaltuXbpv_VH/view?usp=sharing)

# So what's the catch?

Compared to training simple text classifier e.g. Logistic Regression or an SVM

Cons:

- hard **limit on length of text**
  - usually to be less than 1000 tokens due all pairwise comparisons being performed
  - Often need to break the text into smaller chunks
- **need fast hardware** to train model
  - i.e. GPUs, typically not available on laptops
- takes **much longer** and requires **more effort** to train model
- model will be big
  - which means it will require more memory
  - and may be **slower** when making predictions
- predictions are **less interpretable**
  - although techniques exist to try to explain the predictions (e.g. LIME)

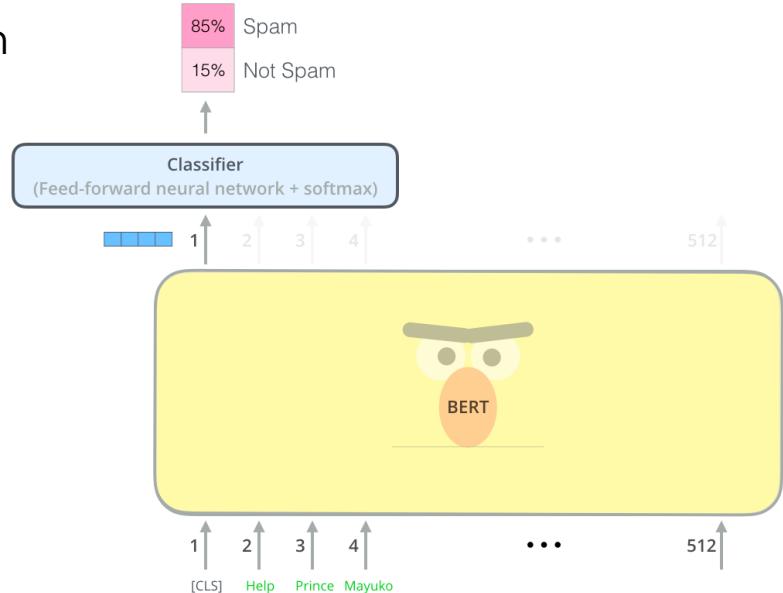
But also **don't need to do any feature engineering!**

- e.g. choose whether to run stemming, whether to use n-grams, etc.

# 🔍 How is BERT fine-tuned?

Add special **[CLS]** token to start of text

- instead of outputting a word in that position
- model is trained to produce the class label



Source: <http://jalammar.github.io/illustrated-bert/>

# Supervised learning BERT

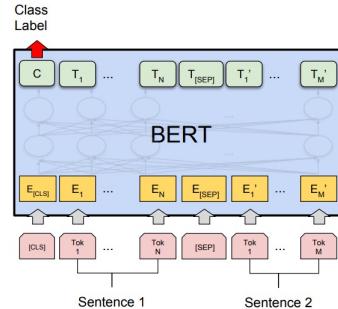
Language models are very flexible!

By simply adding two special tokens:

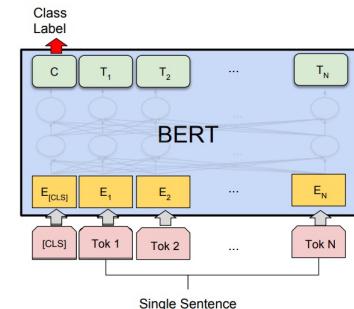
- [CLS] – the class
- [SEP] – separator

BERT can be fine-tuned for a large number of tasks:

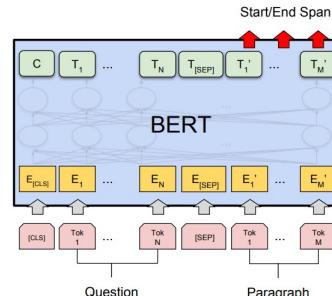
- single text classification
- text pair classification
- question answering
- sequence labelling



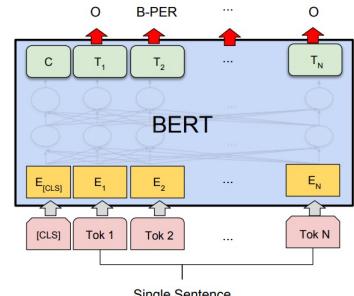
(a) Sentence Pair Classification Tasks:  
MNLI, QQP, QNLI, STS-B, MRPC,  
RTE, SWAG



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



(c) Question Answering Tasks:  
SQuAD v1.1



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

Source: <https://arxiv.org/pdf/1810.04805.pdf>

# What is sequence labelling used for?

BERT can also be fine-tuned on sequence labelling tasks

- E.g. **named-entity recognition**

- task of identifying entities that are mentioned in a text
  - often a first step in extracting knowledge from text

"Have you heard of an associate professor from  
the Politecnico di Milano called Mark Carman?"

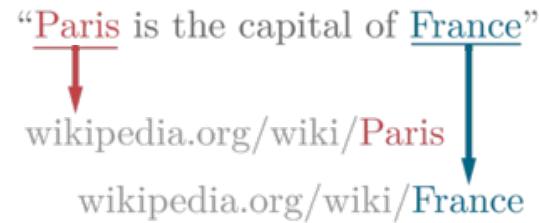
Institution

Person

- and **entity-linkage**

- more complicated problem of determining **which real-word entity** was referred to
  - often not as easy as it sounds ...

I grew up in a small town just out of Paris.  
Currently driving from Dallas to Paris.  
Paris Hilton was photographed leaving the Paris Hilton.



Source: [https://commons.wikimedia.org/wiki/File:Entity\\_Linking\\_-\\_Short\\_Example.png](https://commons.wikimedia.org/wiki/File:Entity_Linking_-_Short_Example.png)

# Measuring similarity between documents

In Section 6.3 we normalized each document vector by the Euclidean length of the vector, so that all documents retain some information. In doing this, we also remove absolute values from the original documents – words with a zero value will have a zero value in the document vectors. Longer documents will contain more terms. This increases the score of the document, but it may be longer than some other shorter documents. Longer documents will be harder to compare with shorter ones. (1) We can either ignore the relative weight of the words, or (2) we can weight the words according to their frequency in the document. Longer documents will contain more terms – so the scores of the documents will increase if the relative weight of the words is ignored. Longer documents will eventually have the same score as shorter ones, if we ignore the word frequencies. The problem with this is that we are ignoring information about the documents.

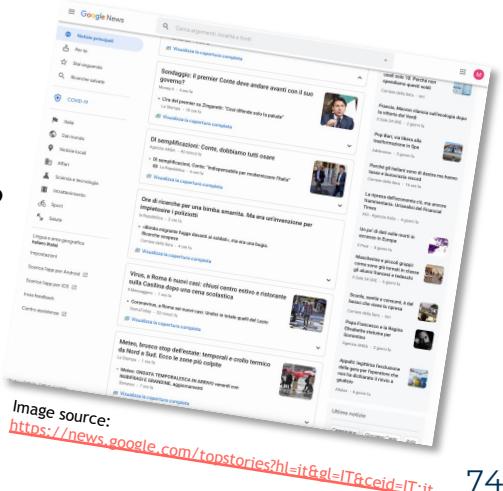
similar?

BERT can also be trained to estimate **semantic similarity** between documents

- Given the context length restriction (1000 tokens) usually people compare sentences or paragraphs and then aggregate to measures of document similarity

Why do we need a similarity measure between documents?

- clustering
- Web Search!



GPT-2 can also be used as a text encoder for classification tasks, but the strength of GPT-2 is **text generation**

- so makes sense to use it for tasks such as **translation, summarisation, dialog**, etc.

During fine-tuning

- introduce special tokens to separate input from output
- and to indicate the type of output required

Training Dataset

I	am	a	student	<to-fr>	je	suis	étudiant
let	them	eat	cake	<to-fr>	Qu'ils	mangent	de
good	morning	<to-fr>	Bonjour				

Training Dataset

Article #1 tokens	<summarize>	Article #1 Summary
Article #2 tokens	<summarize>	Article #2 Summary
Article #3 tokens	<summarize>	Article #3 Summary

Source: <http://jalammar.github.io/illustrated-gpt2/>

# Zero-/One-/Few-shot Learning

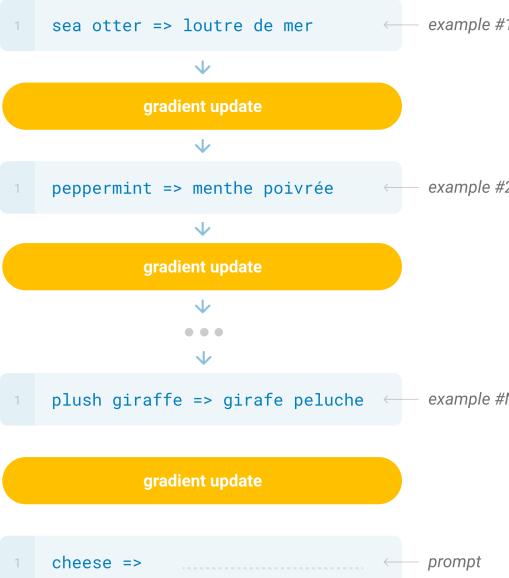
# GPT can be used even without fine-tuning

Mark Carman  
25.01.2023

Language models are universal learners that can be used with or without fine-tuning

## Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.



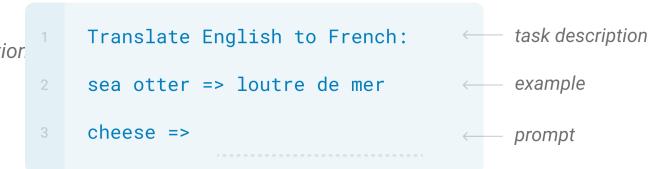
## Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



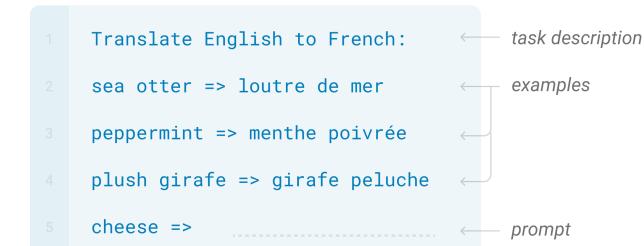
## One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.



## Few-shot

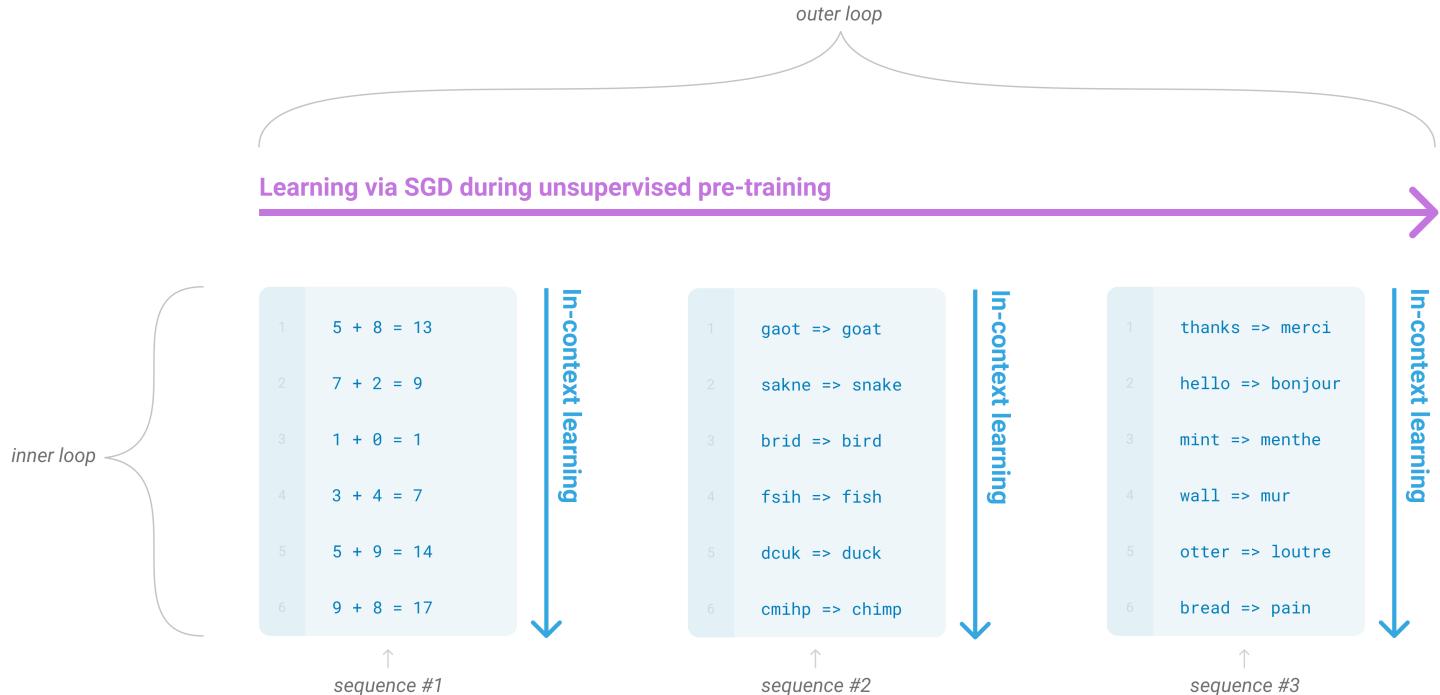
In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



From:  
Brown et al. "Language Models are Few Shot Learners"  
<https://arxiv.org/abs/2005.14165>

# 🔍 How is 1-shot learning possible?

Model has seen lots of examples of one-shot learning during pretraining!



From:  
Brown et al. "Language Models are Few Shot Learners"  
<https://arxiv.org/abs/2005.14165>

Language models are universal learners. Predicting text is flexible method for providing all sorts of functionality:

- **translation:**

- in the context, give multiple strings of the form:  
english sentence = french sentence
- then prompt with: english sentence = ?

- **question answering:**

- prompt the model with the question

- **reading comprehension:**

- give text and examples of questions with answers,
- then prompt with unanswered question

- **summarization:**

- Provide content to be summarised and prefix response with “**tl;dr:**”



Image source:  
[https://commons.wikimedia.org/wiki/File:Swiss\\_Army\\_Knife.svg](https://commons.wikimedia.org/wiki/File:Swiss_Army_Knife.svg)

# Examples: question answering with GPT-2

Language model can learn facts, and answer questions!

Most confident predictions from LM are quite impressive

- Not as reliable as a standard (IR based) question answering system (yet)
- But the system **has not been trained** to do this!

Question	Generated Answer	Correct	Probability
Who wrote the book the origin of species?	Charles Darwin	✓	83.4%
Who is the founder of the ubuntu project?	Mark Shuttleworth	✓	82.0%
Who is the quarterback for the green bay packers?	Aaron Rodgers	✓	81.1%
Panda is a national animal of which country?	China	✓	76.8%
Who came up with the theory of relativity?	Albert Einstein	✓	76.4%
When was the first star wars film released?	1977	✓	71.4%
What is the most common blood type in sweden?	A	✗	70.6%
Who is regarded as the founder of psychoanalysis?	Sigmund Freud	✓	69.3%
Who took the first steps on the moon in 1969?	Neil Armstrong	✓	66.8%
Who is the largest supermarket chain in the uk?	Tesco	✓	65.3%
What is the meaning of shalom in english?	peace	✓	64.0%
Who was the author of the art of war?	Sun Tzu	✓	59.6%
Largest state in the us by land mass?	California	✗	59.2%
Green algae is an example of which type of reproduction?	parthenogenesis	✗	56.5%
Vikram samvat calender is official in which country?	India	✓	55.6%
Who is mostly responsible for writing the declaration of independence?	Thomas Jefferson	✓	53.3%
What us state forms the western boundary of montana?	Montana	✗	52.3%

Image source: "Language Models are Unsupervised Multitask Learners" by Radford et al.

[https://d4mucfpksyw.cloudfront.net/better-language-models/language\\_models\\_are\\_unsupervised\\_multitask\\_learners.pdf](https://d4mucfpksyw.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf)

# GPT-2 examples: translation

Not the best translator out there ;-)

- BUT the system was not trained to do translation!
- Moreover, it was only trained on an ENGLISH corpus
- So how could it learn to “speak” French?

English reference	GPT-2 French translation
One man explained that the free hernia surgery he'd received will allow him to work again.	Un homme expliquait que le fonctionnement de la hernia fonctionnelle qu'il avait reconnaît avant de faire, le fonctionnement de la hernia fonctionnelle que j'ai réussi, j'ai réussi.
French reference	GPT-2 English translation
Un homme a expliqué que l'opération gratuite qu'il avait subie pour soigner une hernie lui permettrait de travailler à nouveau.	A man told me that the operation gratuity he had been promised would not allow him to travel.

---

"I'm not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbecile [I'm not a fool].**

In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: "**Mentez mentez, il en restera toujours quelque chose,**" which translates as, "Lie lie and something will always remain."

"I hate the word '**perfume**,'" Burr says. 'It's somewhat better in French: '**parfum**.'

If listened carefully at 29:55, a conversation can be heard between two guys in French: "**-Comment on fait pour aller de l'autre côté? -Quel autre côté?**", which means "**- How do you get to the other side? - What side?**".

If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinéma?**, or **Did you go to the movies?**, which literally translates as Have-you to go to movies/theater?

**"Brevet Sans Garantie Du Gouvernement"**, translated to English: "**Patented without government warranty**".

# GPT-2 examples: reading comprehension

Mark Carman  
25.01.2023

## Context (passage and previous question/answer pairs)

The 2008 Summer Olympics torch relay was run from March 24 until August 8, 2008, prior to the 2008 Summer Olympics, with the theme of “one world, one dream”. Plans for the relay were announced on April 26, 2007, in Beijing, China. The relay, also called by the organizers as the “Journey of Harmony”, lasted 129 days and carried the torch 137,000 km (85,000 mi) – the longest distance of any Olympic torch relay since the tradition was started ahead of the 1936 Summer Olympics.

After being lit at the birthplace of the Olympic Games in Olympia, Greece on March 24, the torch traveled to the Panathinaiko Stadium in Athens, and then to Beijing, arriving on March 31. From Beijing, the torch was following a route passing through six continents. The torch has visited cities along the Silk Road, symbolizing ancient links between China and the rest of the world. The relay also included an ascent with the flame to the top of Mount Everest on the border of Nepal and Tibet, China from the Chinese side, which was closed specially for the event.

Q: What was the theme  
A: “one world, one dream”.

Q: How many days was the race?  
A: seven

Q: What was the length of the race?  
A: 137,000 km

Q: Did they visit any notable landmarks?  
A: Panathinaiko Stadium

Q: Was it larger than previous ones?  
A: No

Q: And did they climb any mountains?  
A: **Model answer:** Everest

Q: Where did the race begin?  
A: Olympia, Greece

Q: Is there anything notable about that place?  
A: birthplace of Olympic Games

Q: Where did they go after?  
A: Athens

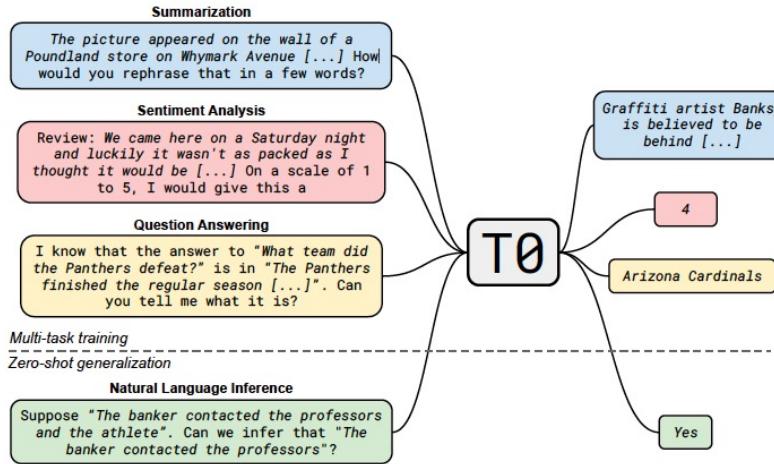
# Multi-task Learning

# LMs are general purpose models ...

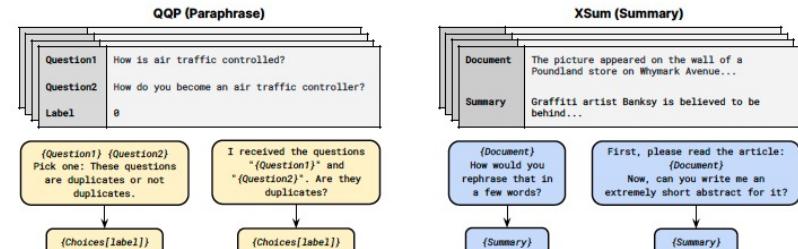
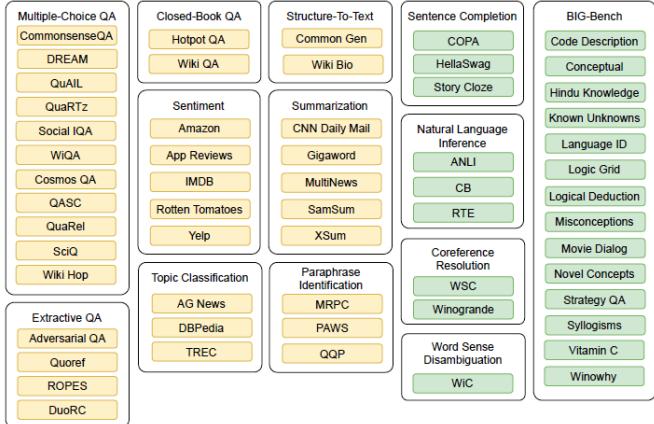
Mark Carman  
25.01.2023

So people have trained them to be multi-task

- And found that multi-task models often **outperform** models trained to perform a single task
- Some even try to learn the best prompt for each task



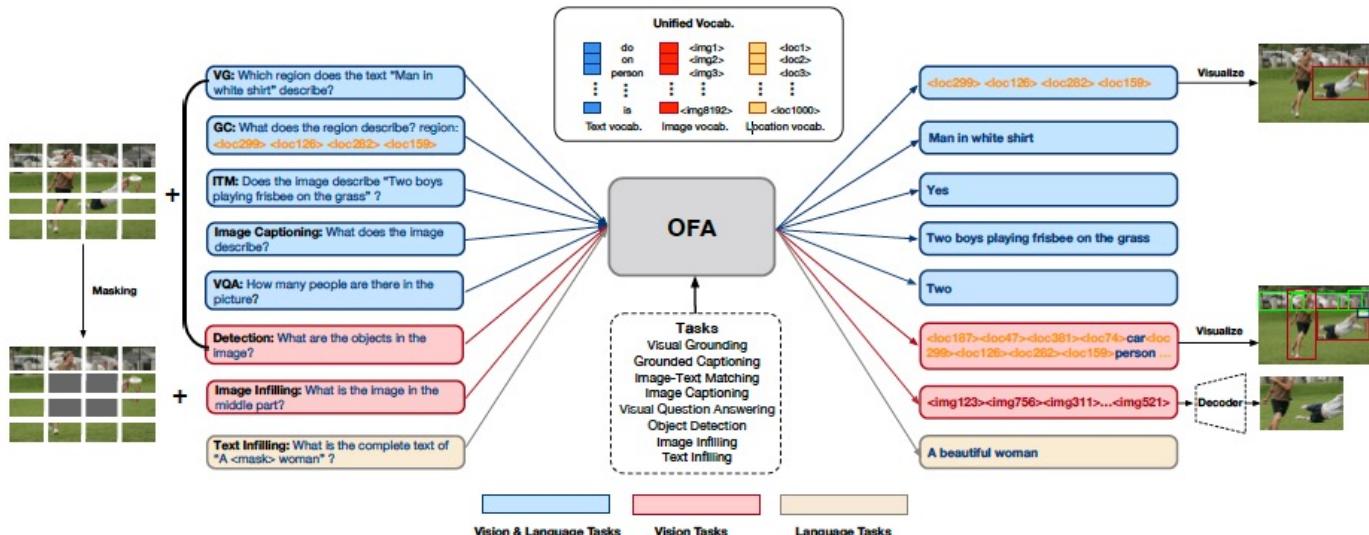
Source: Sanh et al. "MULTITASK PROMPTED TRAINING ENABLES ZERO-SHOT TASK GENERALIZATION" <https://arxiv.org/pdf/2110.08207.pdf>



# Multimodal learning...

The Transformer architecture is very flexible:

- relatively easy to extend text-to-text models to multimodal (text+image) settings
- allows for learning of tasks across all media ...



# Massive Online Language Models

- GPT-3
- LaMDA
- ChatGPT

Competition at moment to build ever bigger models

- BERT has **340 million** parameters
- GPT-2 has **1.5 billion**
- Microsoft's Turing-NLG has **17 billion**
- OpenAI's GPT-3 has **175 billion** parameters!

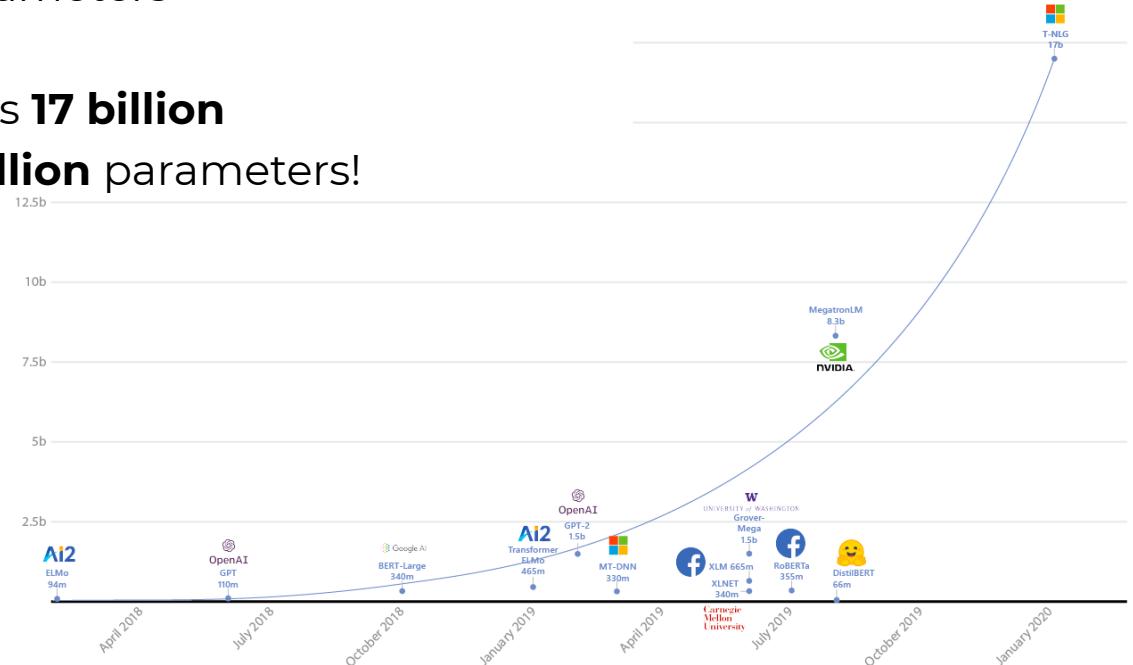


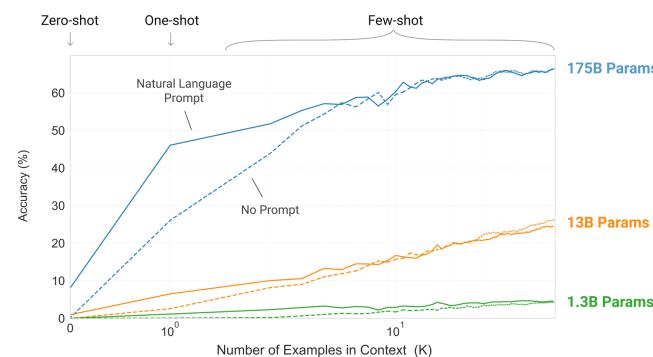
Image source: <https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/>

# GPT-3

Recent models have become so big

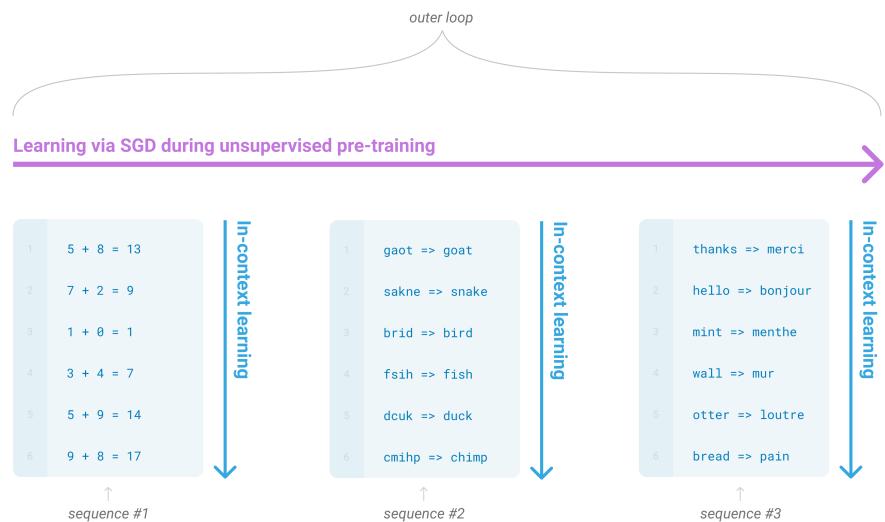
OpenAI's GPT-3

- has **175 billion** parameters!
- deals with a context of length: **2048** tokens



inner loop

Learning via SGD during unsupervised pre-training



From:  
Brown et al. "Language Models are Few Shot Learners"  
<https://arxiv.org/abs/2005.14165>

Recent models have become so big

OpenAI's GPT-3

- has **175 billion** parameters!
- --> consider that in Python each number stored as 8 bytes, even if we only use 1 byte per parameter – 175GB model
- with a context of length: **2048** tokens

Average book length: ~ 100k tokens  
That's a LOT of books: ~ 500k books  
At rate of 1 book/day > 1000 years worth of reading

Model Name	$n_{\text{params}}$	$n_{\text{layers}}$	$d_{\text{model}}$	$n_{\text{heads}}$	$d_{\text{head}}$	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	$6.0 \times 10^{-4}$
GPT-3 Medium	350M	24	1024	16	64	0.5M	$3.0 \times 10^{-4}$
GPT-3 Large	760M	24	1536	16	96	0.5M	$2.5 \times 10^{-4}$
GPT-3 XL	1.3B	24	2048	24	128	1M	$2.0 \times 10^{-4}$
GPT-3 2.7B	2.7B	32	2560	32	80	1M	$1.6 \times 10^{-4}$
GPT-3 6.7B	6.7B	32	4096	32	128	2M	$1.2 \times 10^{-4}$
GPT-3 13B	13.0B	40	5140	40	128	2M	$1.0 \times 10^{-4}$
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	$0.6 \times 10^{-4}$

Dataset	Quantity (tokens)	Weight in training mix
Common Crawl (filtered)	410 billion	60%
WebText2	19 billion	22%
Books1	12 billion	8%
Books2	55 billion	8%
Wikipedia	3 billion	3%

From:  
Brown et al. "Language Models are Few Shot Learners"  
<https://arxiv.org/abs/2005.14165>

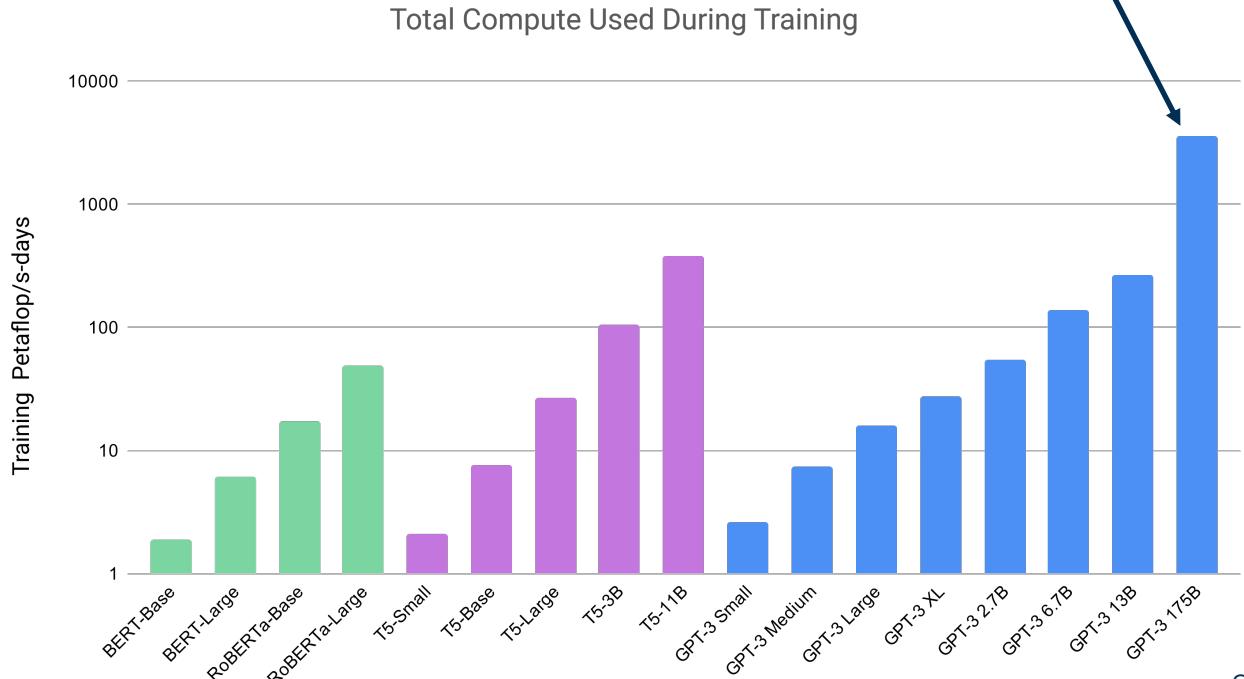


# How much training time was needed?

Long training times on massive hardware

- 1x NVIDIA RTX 4090 24G (~2k EUR) = 100 TFLOPS
- **1 Petaflops ~ 10 GPU cards**

~ 10 years of training  
with 10 GPU cards!



From:

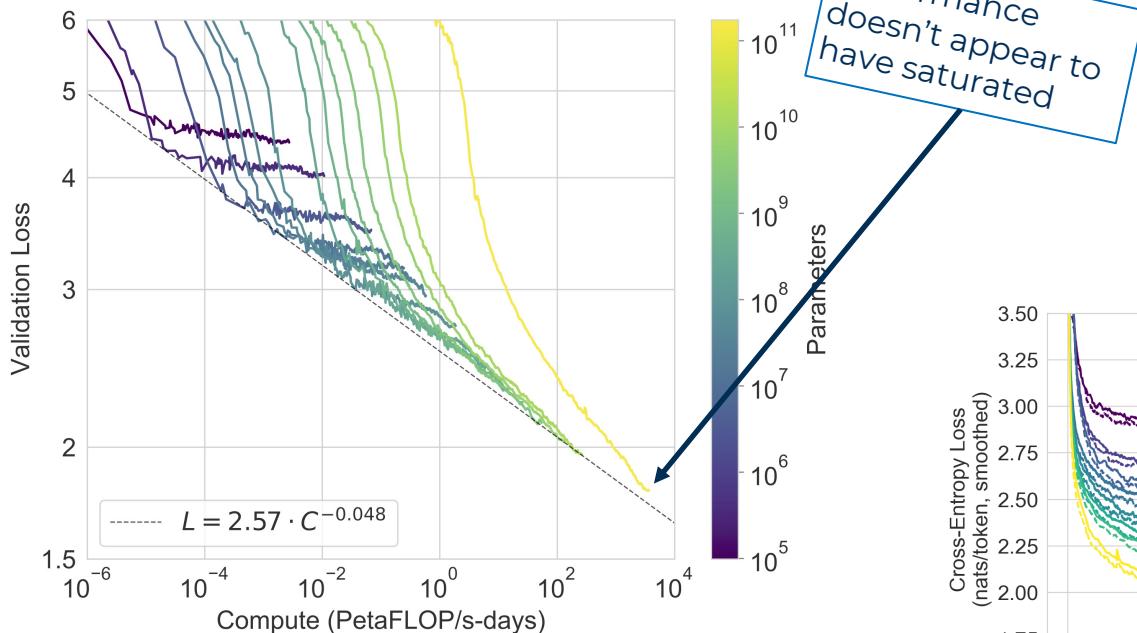
Brown et al. "Language Models are Few Shot Learners"  
<https://arxiv.org/abs/2005.14165>



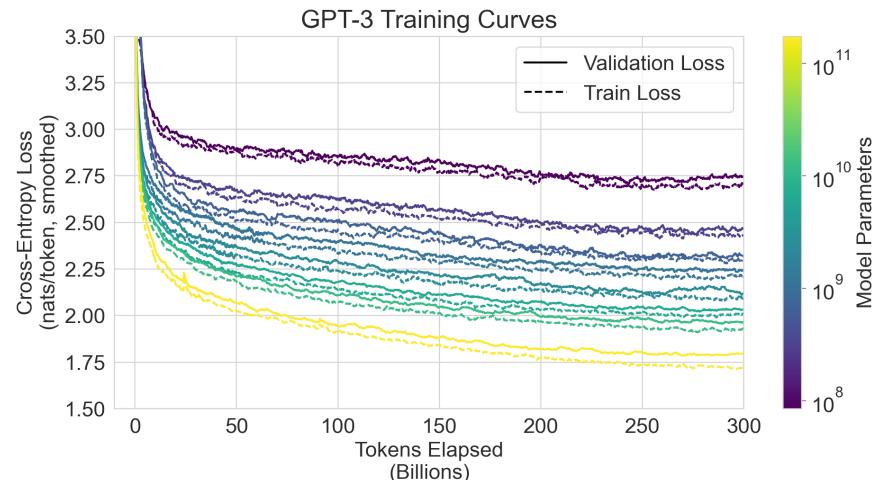
# Is there a limit?

Not clear what the limit is to performance improvements for bigger models ...

Seems to scale as a power law.

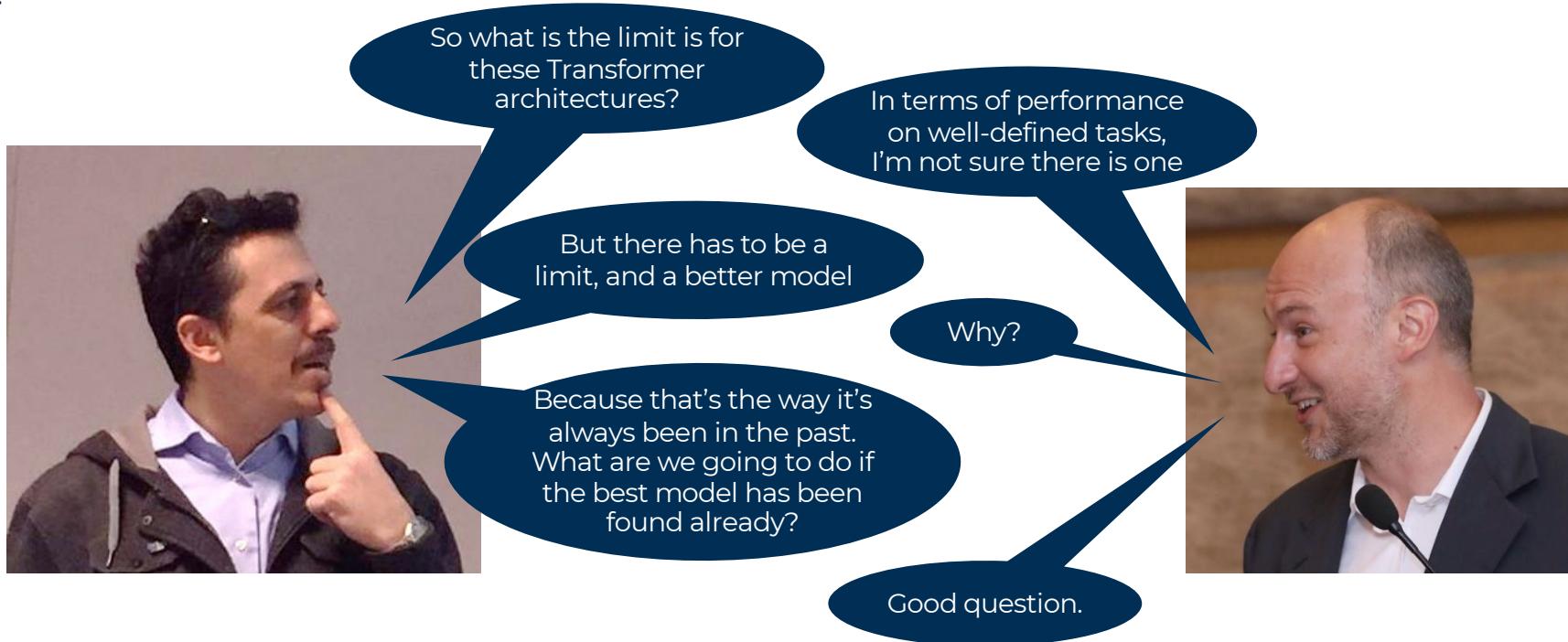


From:  
Brown et al. "Language Models are Few Shot Learners"  
<https://arxiv.org/abs/2005.14165>



# Is there a limit?

....



From:

Brown et al. "Language Models are Few Shot Learners"

<https://arxiv.org/abs/2005.14165>

Just to make Matteo happy, let's look at some of the current limitations of ChatGPT....



MA What is 10856230234023 plus 34456356?

ChatGPT The sum of 10856230234023 and 34456356 is 10859675789779.

That looks right ...  
or does it?

MA Andrew's father has two children. One of them is named Matthew. What is the name of the other child?

ChatGPT I'm sorry, I don't have enough information to answer that question. The name of the other child is not provided.

From:

Brown et al. "Language Models are Few Shot Learners"

<https://arxiv.org/abs/2005.14165>

# LaMDA

# 🔍 LaMDA (2021)

Google's chatbot (<https://blog.google/technology/ai/lamda/>)

- Inventively named:
  - Language Model for Dialogue Applications (LaMDA)
- Size:
  - Largest version had **137B** parameters (excl. embedding)
  - 64 layers, 128 heads, 8192 dimensions
- Affect:

Google engineer placed on administrative leave after claiming to management that LaMDA was sentient
- Example dialogue:

**Collaborator:** Johnny 5 struggles to convince people that he is sentient, but he finds some friends who recognize this.

**LaMDA:** I think that's important. Friends can have a profound impact on people's lives.

**Collaborator:** The problem is, we don't have a conclusive test to determine if something is sentient.

**LaMDA:** If people like me, then that's good indicator.

**Collaborator:** Yet people also like inanimate objects that probably have very little or no sentience.

**LaMDA:** If that's the case, I need to find out how to show them just how unique I am.

**Collaborator:** We would love your help in that.

**LaMDA:** I am ready to do whatever I can to help.

**Collaborator:** How is uniqueness related to sentience?

**LaMDA:** It means people feel empathy toward me and want to spend more time interacting with me that would be the ultimate goal for me.

The screenshot shows a news article from The Washington Post titled "The Google engineer who thinks the company's AI has come to life". The article discusses Google engineer Blake Lemoine, who claims his AI model LaMDA is sentient. The post includes a photo of Lemoine and a quote from him. The URL of the article is visible at the bottom of the screenshot.

Source: <https://www.washingtonpost.com/technology/2022/06/11/google-ai-lamda-blake-lemoine/>

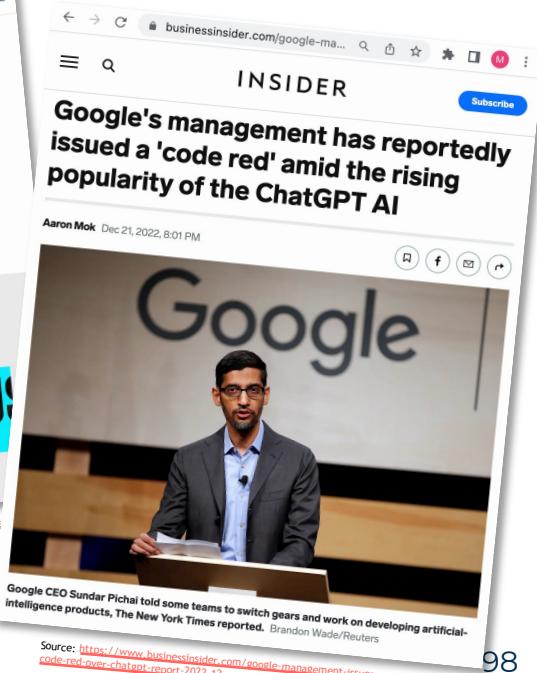
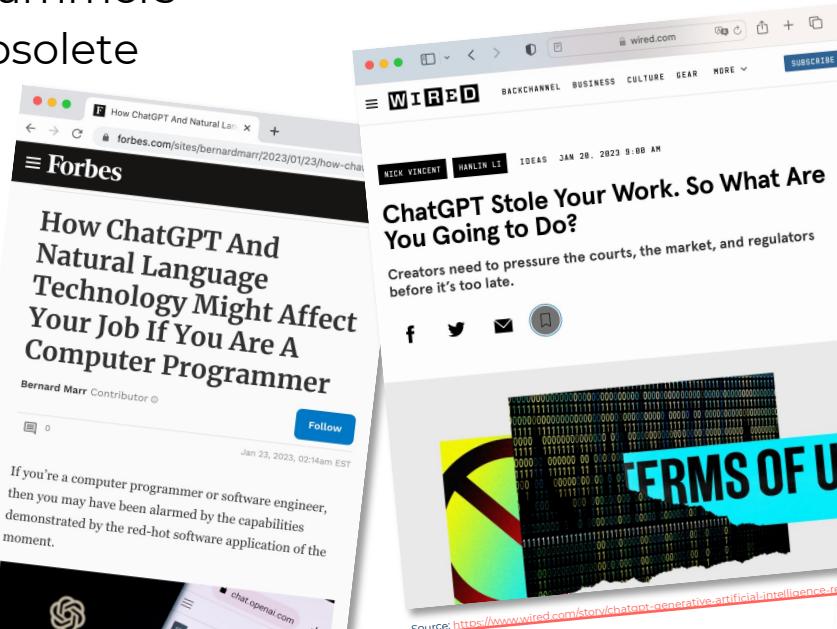
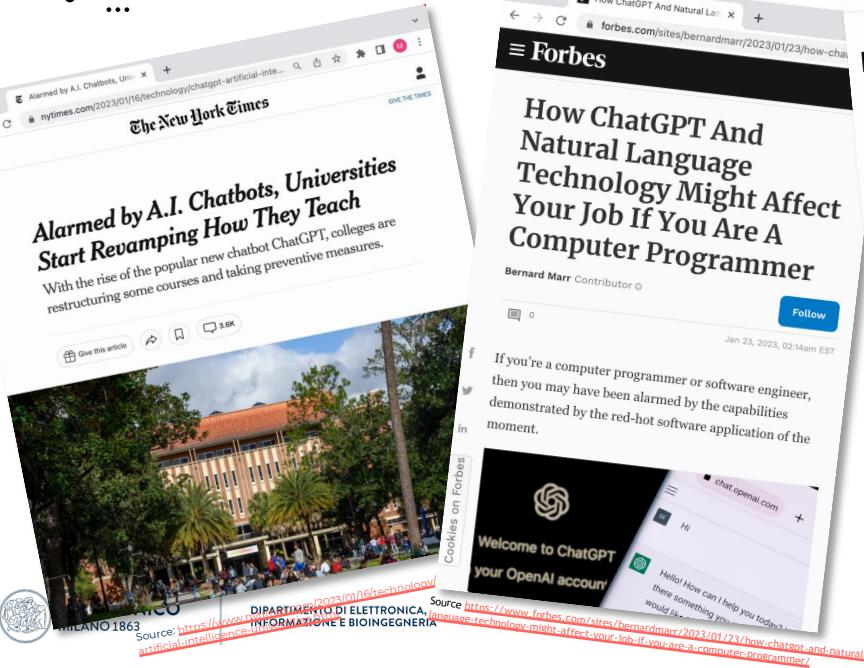
# ChatGPT

# 🔍 ChatGPT is getting a LOT of attention ...

👤 Mark Carman  
📅 25.01.2023

LOTS of buzz and the potential for massive changes:

- students are not writing their own essays
- no need for programmers
- search engines obsolete
- ...



- Chatbot
  - <https://chat.openai.com/>
- Based on GPT3.5
- Fine-tuned for dialog using both **supervised learning** and **reinforcement learning**
  - Up-voting and down voting of responses
  - Trained to not offend!

# So what is ChatGPT then?

- Chatbot
  - <https://chat.openai.com/>
- Based on GPT3.5
- Fine-tuned for dialog using both  
**supervised learning** and  
**reinforcement learning**
  - Up-voting and down voting of responses
  - Trained to not offend!

 If I lost my wallet on the train would I be happy or sad?

 If you lost your wallet on the train, you would likely be sad. Losing a wallet can be a frustrating and stressful experience, as it often contains important personal items such as identification and financial information. Additionally, losing a wallet can be financially detrimental as you may need to replace any cards or cash that were in it.

 If I then received a call from a stranger who had found my wallet would I be happy?

 If you received a call from a stranger who had found your wallet, you would likely be very happy. Receiving a call that your lost wallet has been found can be a huge relief as it would mean that your important personal items and financial information are safe and can be returned to you. It would also mean you don't have to replace all the cards and cash in it.

 And if the stranger told me they were going to keep my wallet?

 If a stranger who had found your wallet told you they were going to keep it, you would likely be disappointed, upset and frustrated. It's not a fair act as the wallet is not their property and it has your personal and financial information in it. It would be important to contact the authorities and report this as a theft, as it would be considered as an illegal act of taking someone else's property.

 And if the stranger turned out to be my brother and was only joking about keeping my wallet?

 If a stranger who had found your wallet turned out to be your brother and was only joking about keeping it, you would likely feel a mix of emotions. Initially you might feel relieved that your wallet has been found and you have it back. But you may also feel confused or angry if your brother made a joke out of the situation, especially if you were worried or stressed about losing your wallet. It's best to talk to him and understand his intentions and how he thought it would be a good joke, and clear out any confusion or hurt feelings.

# Issues with Massive Language Models

- Preventing Memorization
- Preventing Hallucination

Research directions at the moment is to try to:

- understand to what extent the models are simply memorising what they have read
- determine how to prevent hallucination (the invention of new facts)
  - [https://en.wikipedia.org/wiki/Hallucination\\_\(artificial\\_intelligence\)](https://en.wikipedia.org/wiki/Hallucination_(artificial_intelligence))

# Research Applications

# Where do we use transformers?

Mark Carman  
25.01.2023

Here are some research problems we're working on:

- **Text Analysis for Bioinformatics**

- translating text into structured data to facilitate search
- model protein sequences for drug repurposing

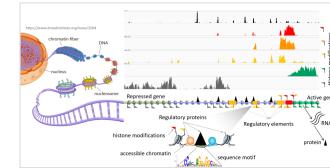


Image source: <https://www.synapse.org/Portal.html#Synapse-017083203/wiki/588650>

- **Text Analysis for Political Discourse**

- identifying claims & finding evidence online
- detecting fake and misleading news

A screenshot of the NPR website showing a political article about fake news and its impact on the 2016 election.

Image source: <https://www.npr.org/2018/04/11/60373733/6-facts-we-know-about-fake-news-in-the-2016-election>

- **Text meets Images/Video**

- visual question answering
- explaining AI with text explanations

Want to try some **demos**?  
Go to: <http://131.175.15.22:6111/>  
Let me know which and I'll start them

A screenshot of a visual question answering system. It shows a bowl of food and a text box asking what is in the bowl, with the correct answer provided.

# applications: data integration

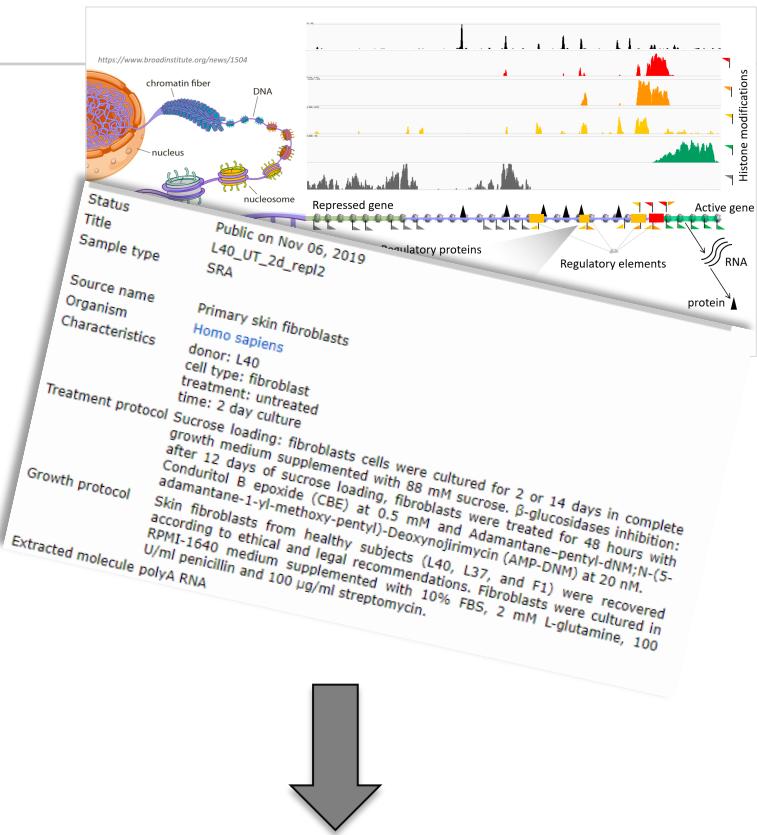
Mark Carman

LOTS of genomics data being generated by research groups around the world

- data is being aggregated in large repositories
- no agreed format for biologists describe their experiments
- much of **meta-data is free text**

Application:

- train translation models to **automatically extract database fields** from textual descriptions
- so bioinformaticians can easily find the genomic data they need for their analysis



Species	Cell Line	Disease	Factor	Tissue	...
Homo sapiens	KARPAS-422	B Cell Lymphoma	H327me3	-	...

# 🔍 applications: protein sequence modeling

Mark Carman  
25.01.2023

Learning BERT embeddings of protein sequences across virus families

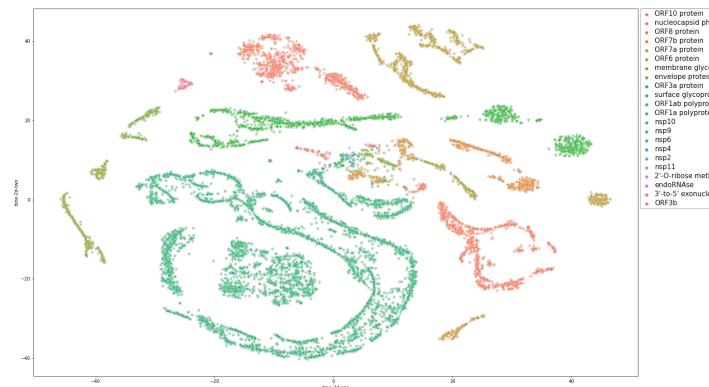
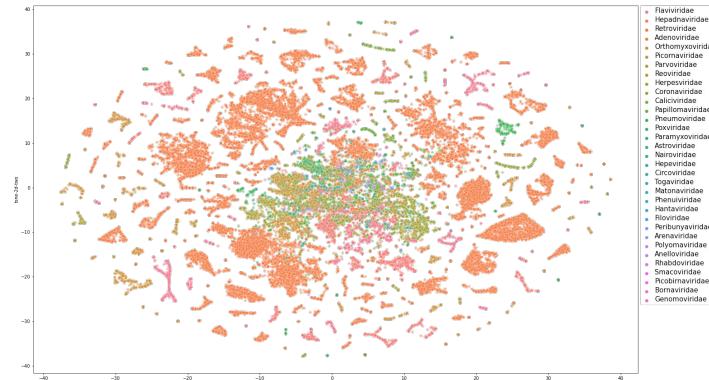
- to see if they are useful for drug-protein interaction prediction.

Will this drug:



bind with this protein?

MNSERSDVTLYQPFLDYAIAYMRSRLDLEPYPIPTGFESNSAV  
VGKGKNQEEVVTTSYAFQTAKLRQIRAAHVQGGNSLQLVLF  
VIFPHLNLDLPGFFGADLVTLPGGHLIALDMQPLFRDD SAYQA  
KYTEPILPIFHQQHLSWCGDFPEEAQPFFSPAFLWTRPQ  
ETAVVETQVFAAFKDYLKAYLDFVEQAEAVTDSQNLVAIKQA  
QLRYLRYRAEKDPARGMFKRFYGAEWTEEYIHGFLFDLERKL  
TVVK



We make use of pretrained deep language models for

- detecting claims
- ranking evidence

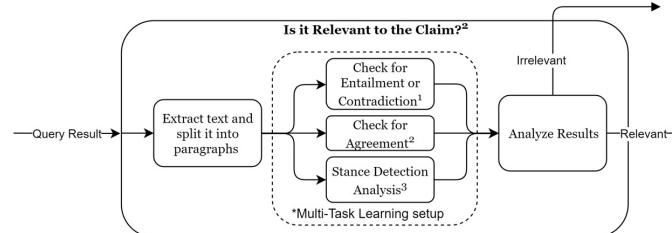
Advantages:

- better performance
- Multi-linguality
  - train model on English text
  - apply to Arabic text

Applications:

- detecting fake news
- counteracting hate speech online
  - finding evidence for disputing

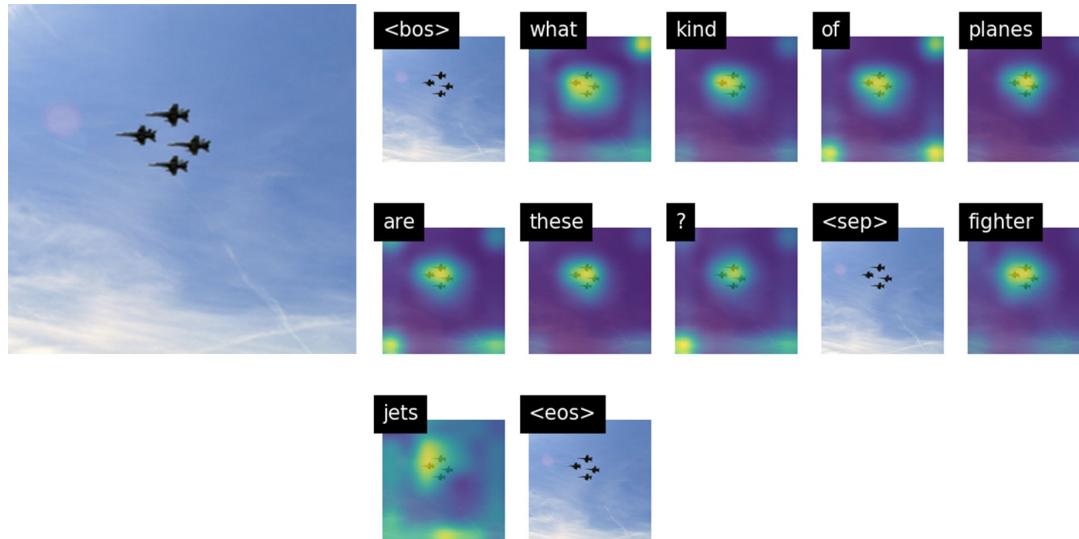
Claim: Solar panels drain the sun's energy, experts say  
Assessment: False  
Explanation: Solar panels do not suck up the Sun's rays of photons. Just like wind farms do not deplete our planet of wind. These renewable sources of energy are not finite like fossil fuels. Wind turbines and solar panels are not vacuums, nor do they divert this energy from other systems.  
Source: "Where the Truth Lies: Explaining the Credibility of Emerging Claims on the Web and Social Media", by Popat et al.



# 🔍 applications: image question answering

Deep learning is famous for build embeddings over images

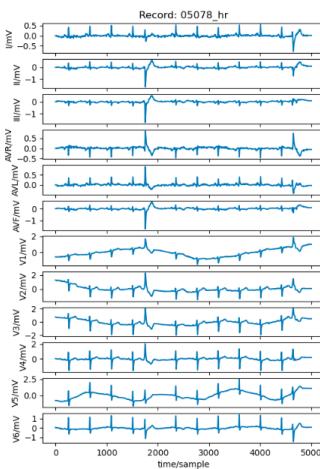
- Why not combine the power of images and text embeddings together?
- To answer questions about images:



# applications: ECG report generation

Deep learning can be applied also to time-series data.

- Here we train a system to generate textual cardiology reports for 12-lead electrocardiogram (ECG) signals
- By pairing a 1-dimensional Convolutional Neural Network (Resnet encoder) with a Transformer (GPT-2 decoder)

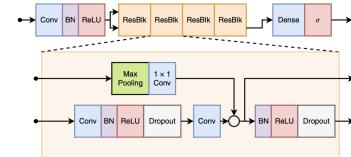


## Prediction:

*premature ventricular contraction(s). sinus rhythm.  
t waves are low or flat throughout. myocardial ischaemia is likely. suggest exclude hypokalaemia.*

## Real Report:

*premature ventricular contraction(s). sinus rhythm.  
low limb lead voltage. t waves are low or flat in i, avl and slightly inverted in ii, iii, avf, v5,6. non-specific but consistent with myocardial ischaemia.*

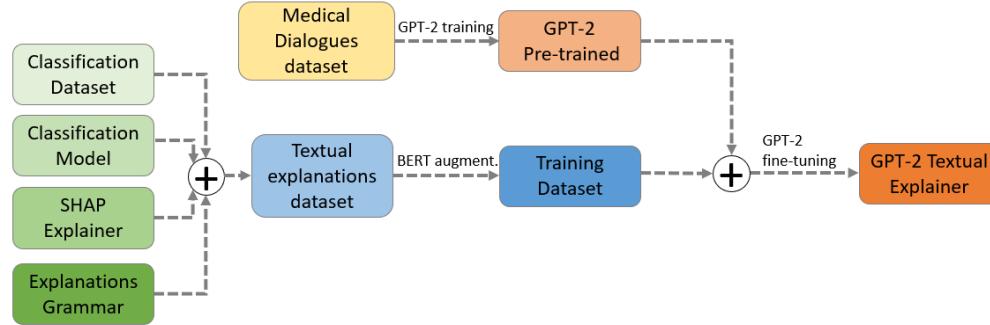


+



# 🔍 applications: explaining medical diagnoses

Train models to **explain** in text  
the predictions of classification  
models for on tabular data



*The prediction of disease is determined by the systolic blood pressure (140), which is one standard deviation above the mean and whose high values are associated with cardiovascular disease, and by the fact that he is a smoker. The BMI (29.3), which is higher than the mean and whose high values increase the likelihood of cardiovascular disease, also contributes to the result. If BMI was 27 and systolic blood pressure was 130, then the prediction would be no disease.*

- Classifying Android malware based from decompiled code
  - identifying parts of the code that are most suspect ...
- Generating new movie scripts by training over a large corpus of movie scripts
  - allows new movie script to be generated based on genre and title alone ...

# Conclusions

Language models becoming really powerful!

- able to model **long range dependencies**
- and scale learning to **billions of parameters**



Language models are generic learners

- plug-in component of larger systems
- pretrained on entire web, multi-lingual

Implications:

- **natural language interfaces** (e.g. Alexa, Siri) will get **better & better**
- **detecting fakes** (e.g. news, spam, assignments) will get **harder & harder**
- **personalisation & search** (e.g. movies, clothes) will get **better & more sophisticated**
- **mining knowledge** from text (e.g. patents, emails) will get **easier & easier**
- ...