Natural Language Processing and LLMs

By Lyle Ungar

neuromatch
academy

Today

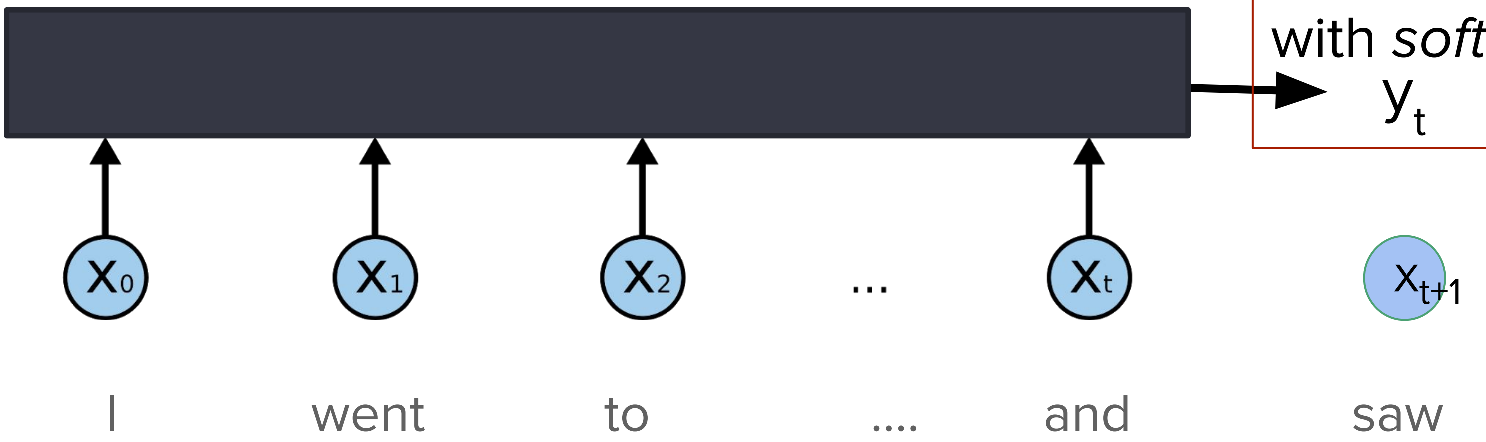
Using transformers for NLP

- NLP architectures
  - BERT and GPT
  - Encoders and decoder
- The NLP pipeline
  - Tokenize, pretrain, fine tune
- GPT and friends
  - The many faces of GPT
  - Prompt engineering
  - Applications
  - Current limitations

NLP Architectures

How to train and use modern language models

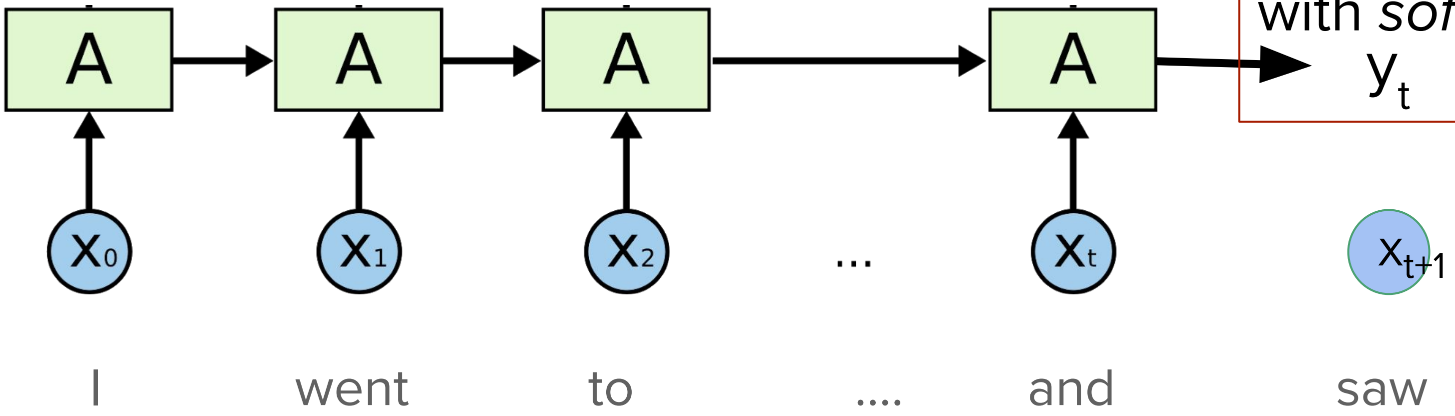
Language model



predict next word with *softmax*(h_t)

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

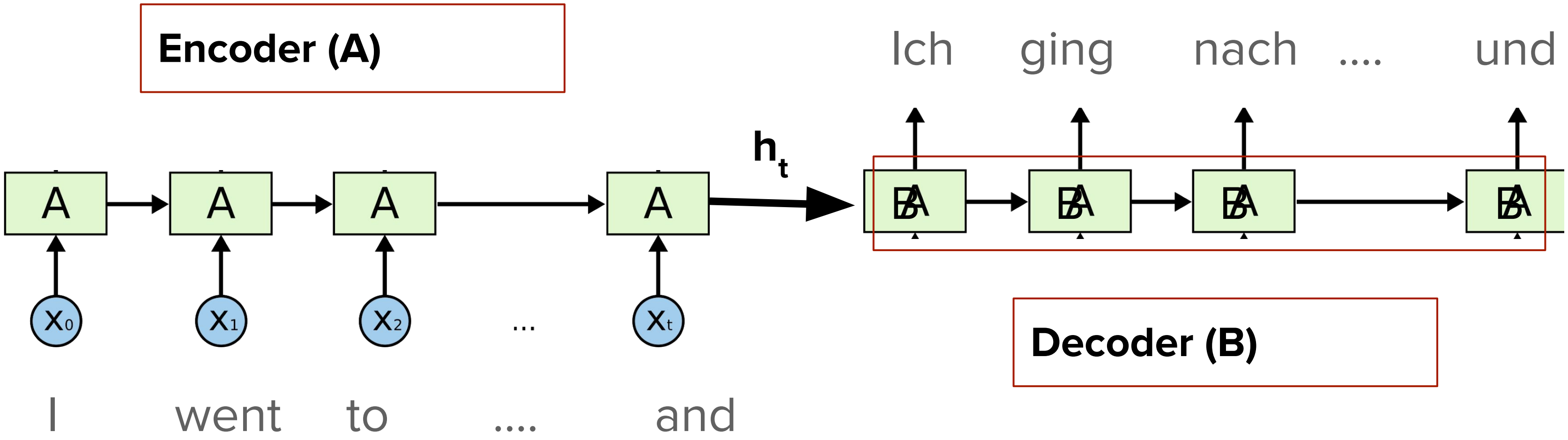
Old: Language model (RNN/LSTM)



predict next word with *softmax(h_t)*

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

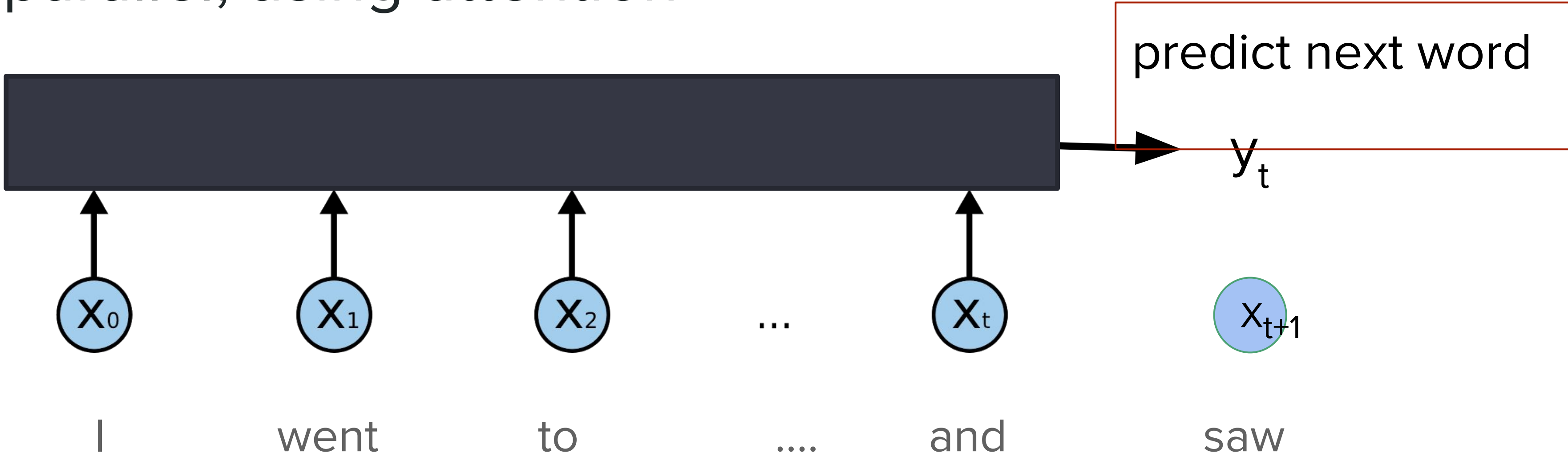
Old: Sequence to Sequence (Seq2seq)


Encoder (A)

Decoder (B)

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

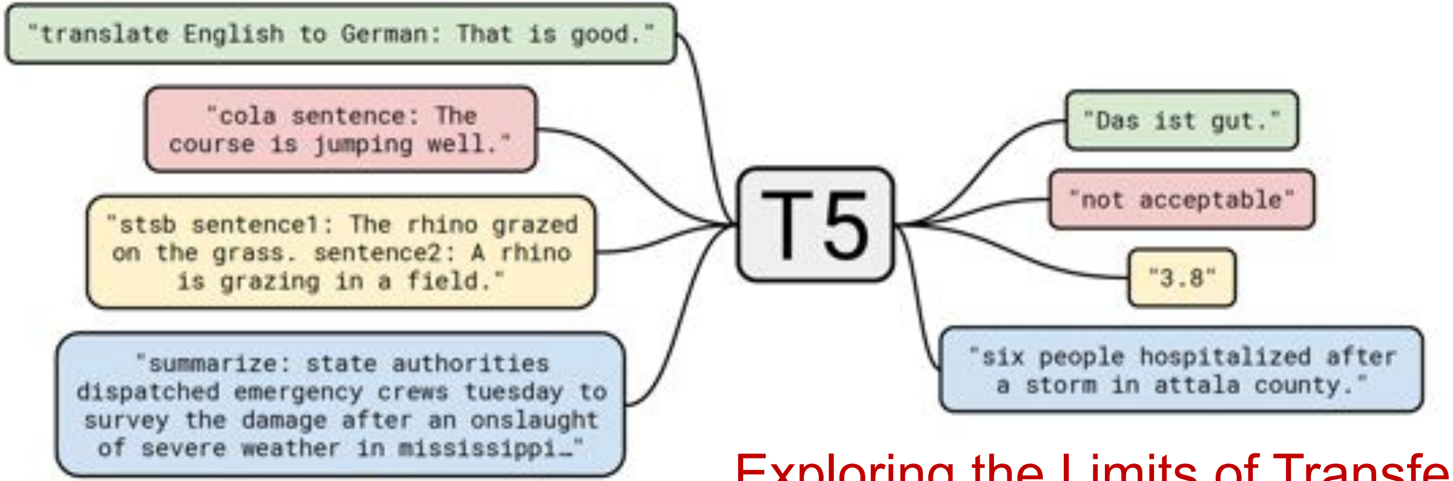
Current: Transformers process text in parallel, using attention



predict next word

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Transformers extend seq2seq


"translate English to German: That is good."

"cola sentence: The course is jumping well."

"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."

"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"

T5

"Das ist gut."

"not acceptable"

"3.8"

"six people hospitalized after a storm in attala county."

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer 2019
– building on Attention is All you Need

The NLP Pipeline

The NLP pipeline

- Pretrain a **language model:** *p(next word | previous words)*
    - Select and clean large unlabeled text collection
    - **Tokenize:** e.g. extract byte pair encodings
    - Map tokens to **embeddings**
- Often **fine-tune** the model
    - Use labeled text or reinforcement learning
- **Evaluate**

Representations

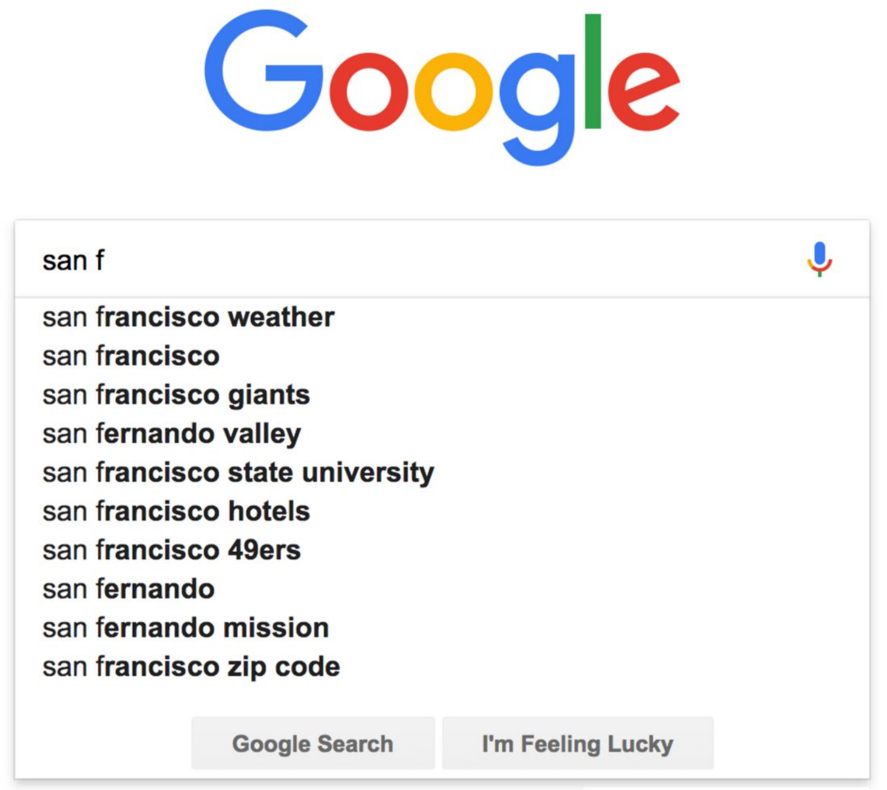
Words/tokens: context-free or context-sensitive embeddings

Sub-word encoding (e.g., Byte Pair Encoding, BPE):

frequent character sequences are treated as “words”

“bbibtech” might be three words “b” + “bib” + “tech”

Language models do autocompletion



san f

san francisco weather
san francisco
san francisco giants
san fernando valley
san francisco state university
san francisco hotels
san francisco 49ers
san fernando
san fernando mission
san francisco zip code

Texts have variable length

Some common practices to handle variable length vectors are:

- **Bag-of-words:** ignore the sequential nature and just count words
- **Embedding:** embed every word; average them
- **Recurrence**: RNN, LSTM
- **Truncation and padding:** chop off long vectors and pad short ones
  - Used by transfomers

Autocompletion is amazingly powerful

Rewrite this business report in rhymed couplets

Rewrite this abstract to fix all the typos and make it clearer

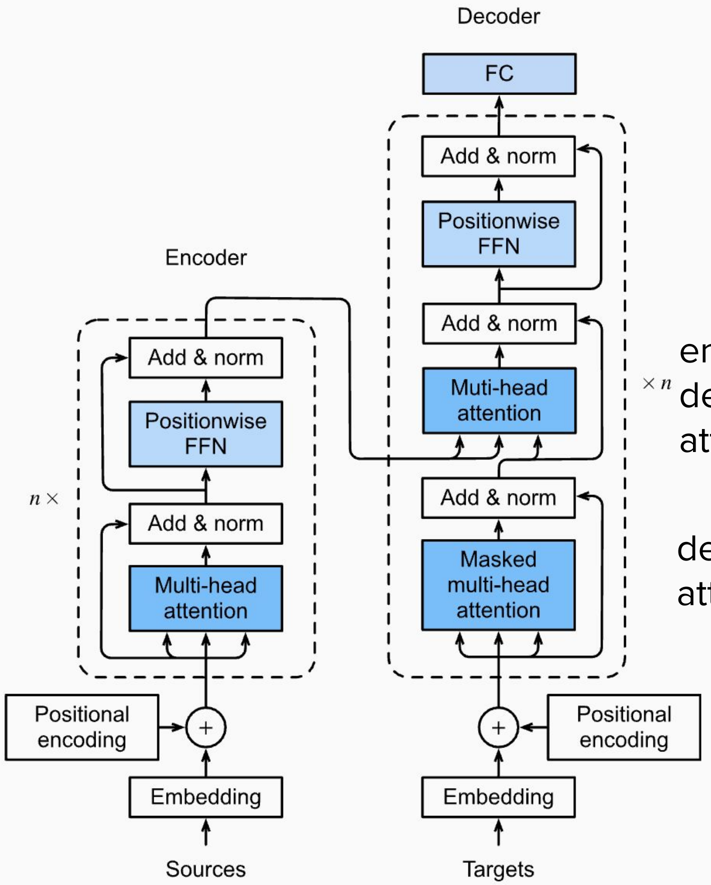
Translate it into Japanese

Write a set of ten multiple choice final exam questions testing machine learning ability for a masters level course

Pretend that you are the job interviewer at google and that I am an applicant applying for a machine learning job

Transformers are “pre-trained”

- **BERT**
  - Encoder:
  - Text to embedding
- **GPT**
  - Decoder:
  - Generative model decodes embeddings to give predicted text



encoder attention

encoder-decoder attention

decoder attention

Pre-training BERT and GPT

- **Pre-training is self-supervised**
  - **Takes advantage of enormous quantities of unlabeled text**
  - **BERT: 15% of the words in each sequence are replaced with a [MASK] token**
  - **Predict them and/or predict the next sentence or word (GPT) given past words**
- **Many variations**
  - **BERT/RoBERTa:** bidirectional ("masked language model")
  - **GPT:** predict next word; unidirectional;("causal language model")
  - **BERT:** character-level BPE of 30K tokens
  - **RoBERTa, GPT:** byte-level BPE of 50K tokens

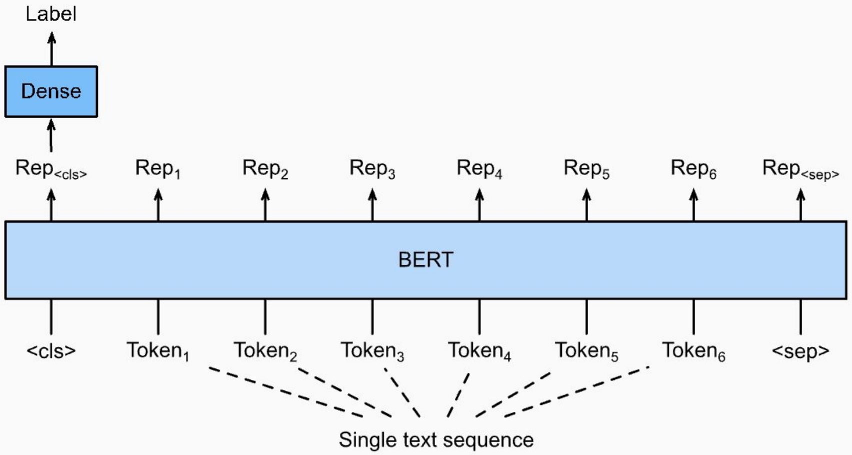
Using BERT

and its many variations

BERT = Bidirectional Encoder Representations from Transformers

BERT’s CLS_token embeds a sentence

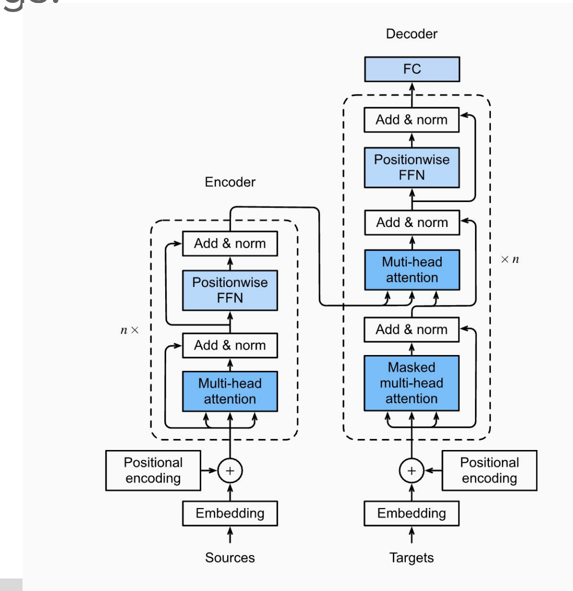
- **BERT:** The 768-D embedding of a CLS_token embeds the sentence
- **RoBERTa:** average the embeddings of all the tokens in the sentence
- Use either of these as features to train a simple feed forward neural net



Use the hidden layer outputs as features

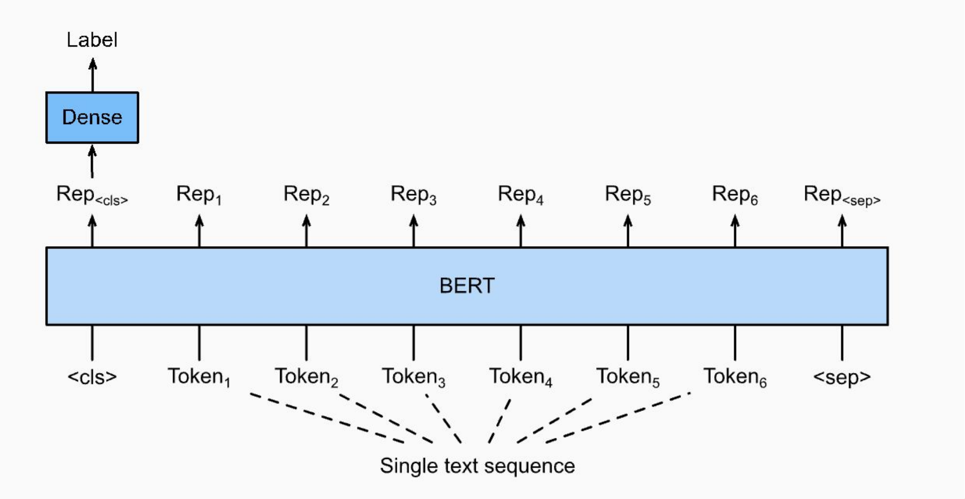
BERT/RoBERTa is just a neural net that produces embeddings.

Use the outputs of its next to last layer or
or the outputs of the last 4 layers as input
to your favorite neural net



Fine Tuning

- Initialize your network as being BERT/RoBERTa, but add the desired output label
- Fix all but the last 4 layers (optional)
- Train using SGD



BERT and friends

BERT

uses self-attention and masking

RoBerta

same architecture as Bert; **trained on more data**

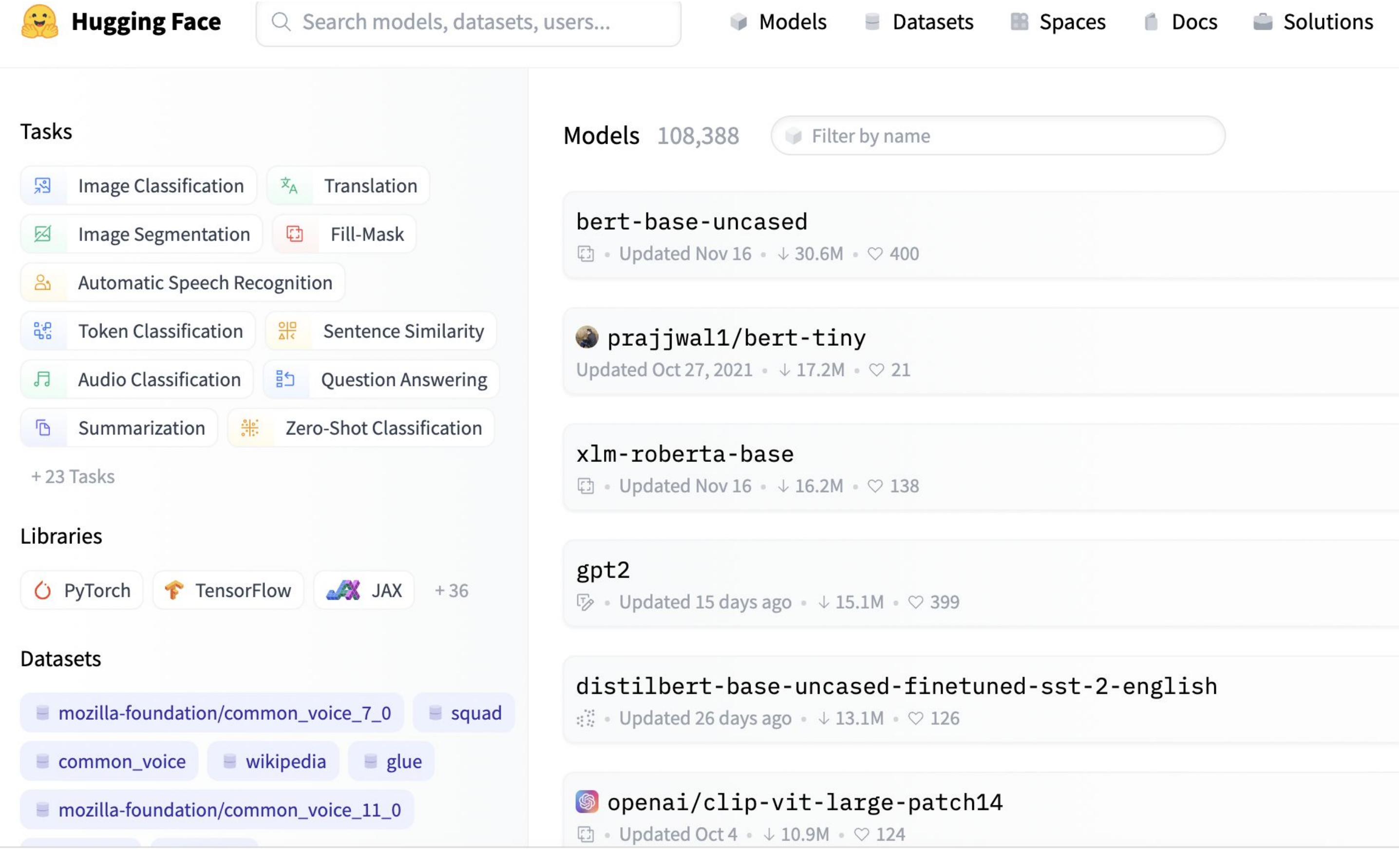
Distilbert

much smaller with little loss of performance

DeBerta

disentangled Bert; often gives more accurate models

dozens of other variations; current best source is huggingface



Tasks

- Image Classification
- Translation
- Image Segmentation
- Fill-Mask
- Automatic Speech Recognition
- Token Classification
- Sentence Similarity
- Audio Classification
- Question Answering
- Summarization
- Zero-Shot Classification
- + 23 Tasks

Libraries

- PyTorch
- TensorFlow
- JAX
- + 36

Datasets

- mozilla-foundation/common_voice_7_0
- squad
- common_voice
- wikipedia
- glue
- mozilla-foundation/common_voice_11_0

Models 108,388

bert-base-uncased
• Updated Nov 16 • ↓ 30.6M • ♡ 400

prajjwal1/bert-tiny
Updated Oct 27, 2021 • ↓ 17.2M • ♡ 21

xlm-roberta-base
• Updated Nov 16 • ↓ 16.2M • ♡ 138

gpt2
• Updated 15 days ago • ↓ 15.1M • ♡ 399

distilbert-base-uncased-finetuned-sst-2-english
• Updated 26 days ago • ↓ 13.1M • ♡ 126

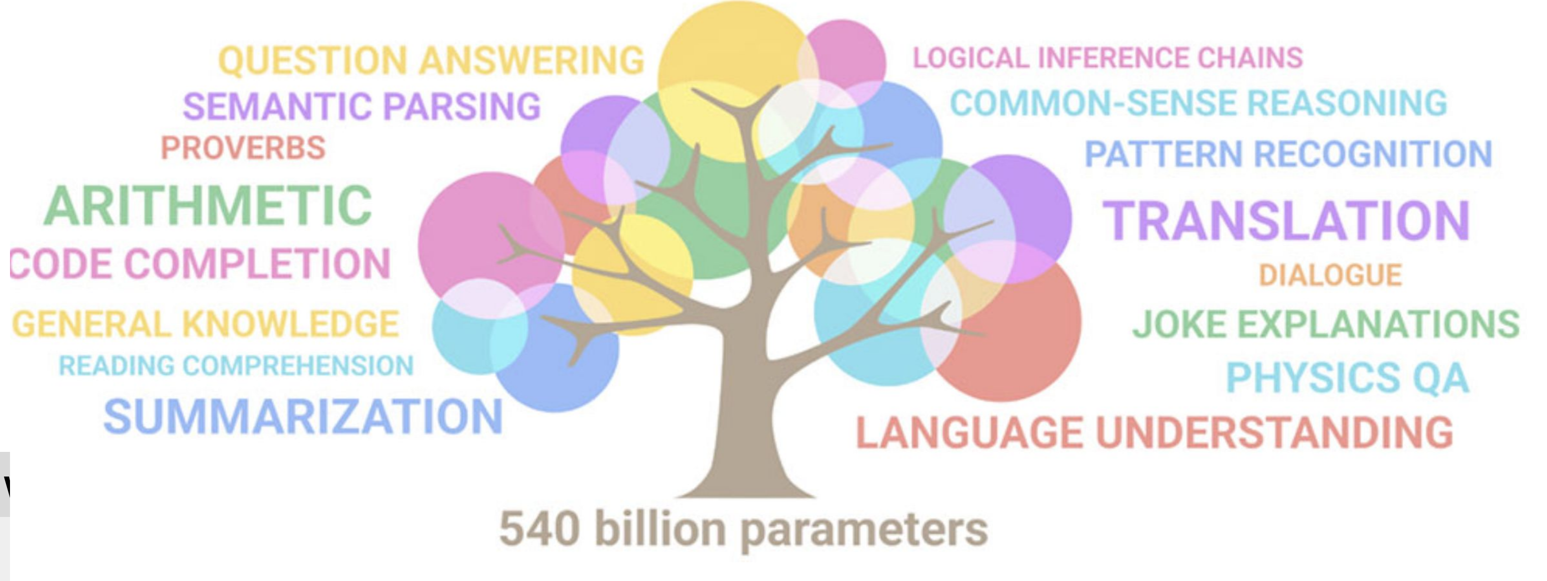
openai/clip-vit-large-patch14
• Updated Oct 4 • ↓ 10.9M • ♡ 124

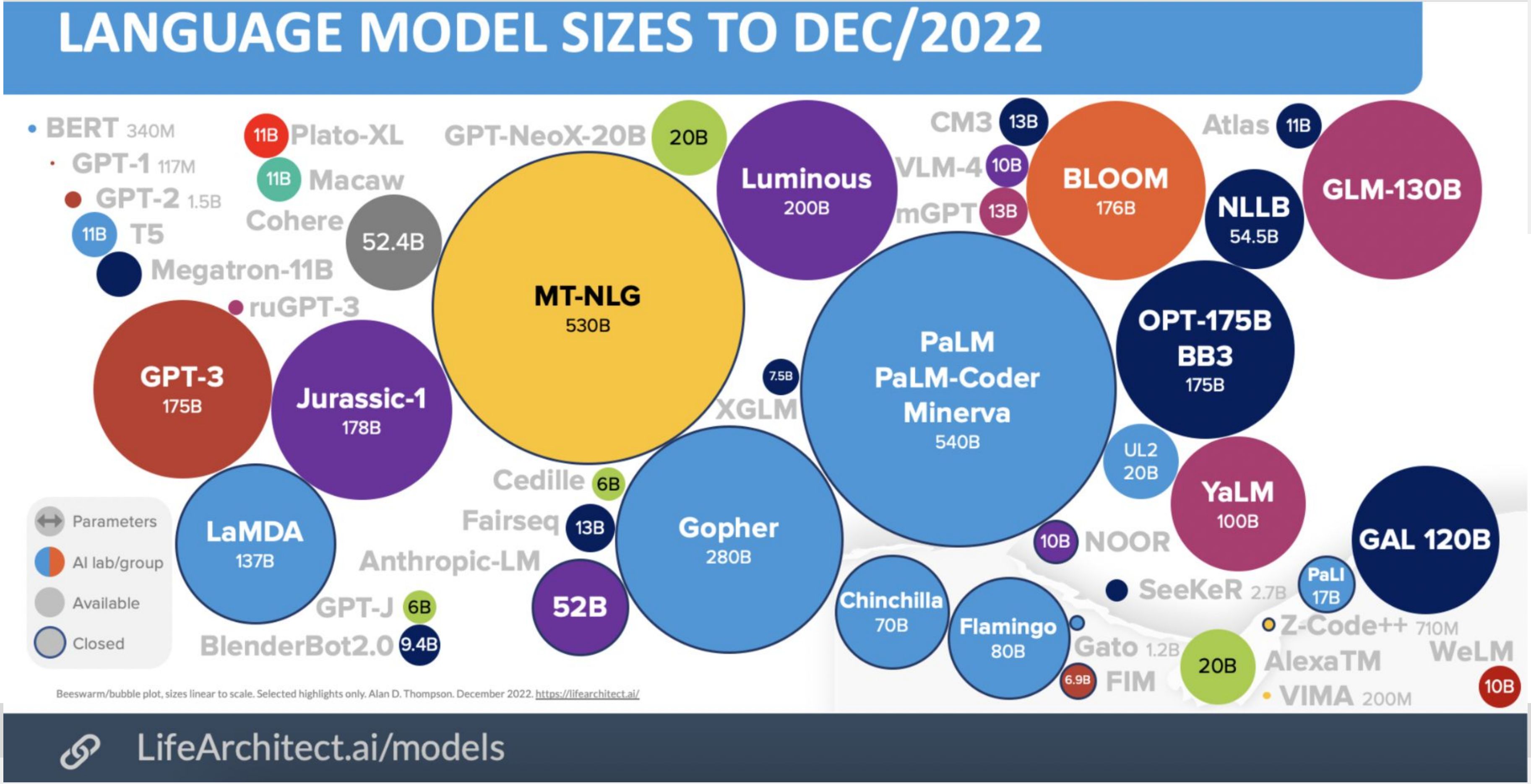
NLG with GPT

Natural Language Generation
with
Generative Pretrained
Transformers

Large Language Models

- [GPT-4, ChatGPT - OpenAI]()
- [Blenderbot - Facebook]()
- [PaLM, Lambda, BART - Google]()





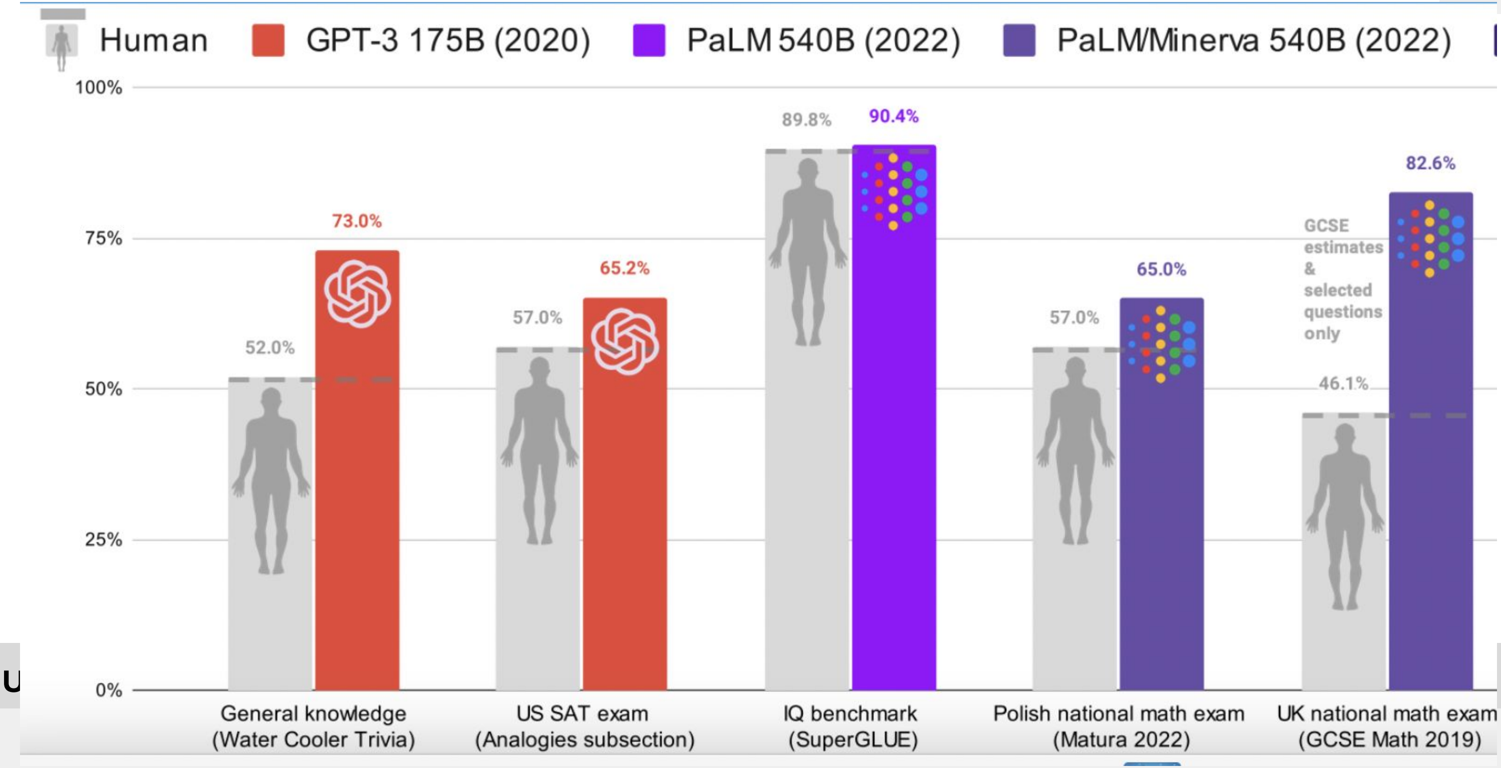
LANGUAGE MODEL SIZES TO DEC/2022

Beeswarm/bubble plot, sizes linear to scale. Selected highlights only. Alan D. Thompson. December 2022. https://lifearchitect.ai/

LifeArchitect.ai/models

https://lifearchitect.ai/models/

LLMs are competitive with humans on (select) exams



https://lifearchitect.ai/iq-testing-ai/

BERT vs. GPT-2

BERT

- Transformer: Encodder
- Masked word
- 768-D embedding
- 345 million params
- Open-source

GPT-2

- Transformer: Decoder
- “Causal” - predict next word
- 768-D embedding
- Small (124M), medium (355M), large (774M), and XL (1.5B) params
- Open-source

Use the hidden layer outputs as features

Lke BERT, GPT-2 produces embeddings.

Use the outputs of its next to last layer or
or the outputs of the last 4 layers as input
to your favorite neural net

Or fine tune it

GPT-2 vs. GPT-3

GPT-2

- Transformer: Decoder
  - predict next word
  - 1,024 token context
- 345 ***million*** params
- Open-source

GPT-3

- Transformer: Decoder
  - predict next word
  - 2,048 token context
- 175 ***billion*** params
- OpenAI/Microsoft

GPT-3: Size and Scale

175 billion adjustable parameters

Trained on 500 billion words of text

410 billion tokens from the Common Crawl

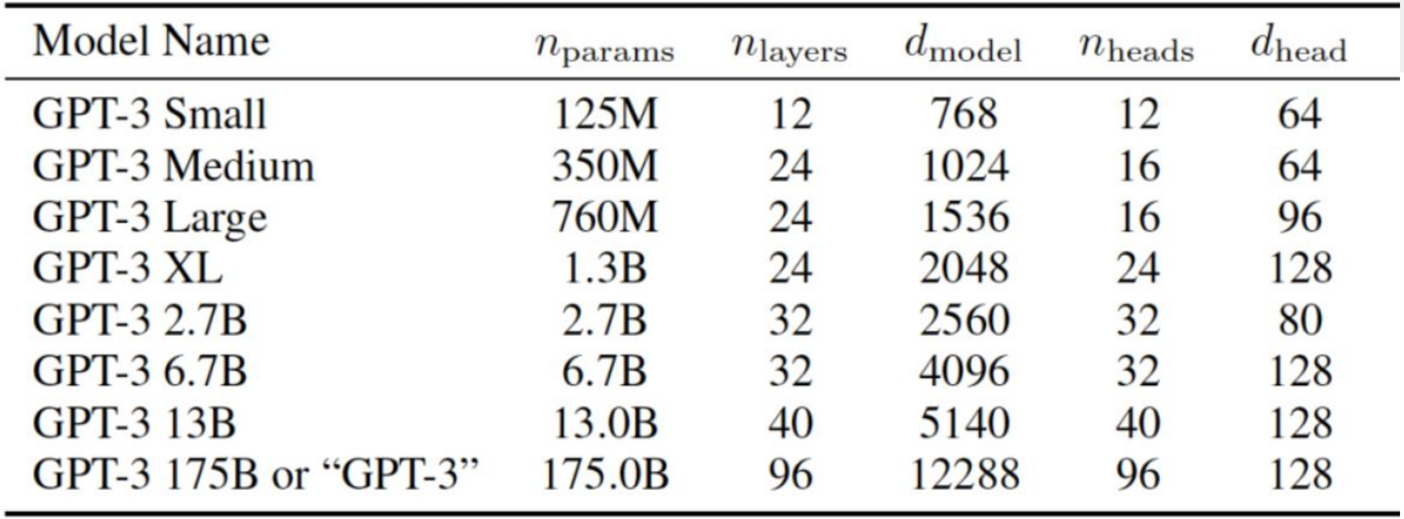
67 billion tokens from books

19 billion tokens from WebText2

3 billion tokens from Wikipedia

Training cost (retail cost) $4.6M

GPT-3 is many models



Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}
GPT-3 Small	125M	12	768	12	64
GPT-3 Medium	350M	24	1024	16	64
GPT-3 Large	760M	24	1536	16	96
GPT-3 XL	1.3B	24	2048	24	128
GPT-3 2.7B	2.7B	32	2560	32	80
GPT-3 6.7B	6.7B	32	4096	32	128
GPT-3 13B	13.0B	40	5140	40	128
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128

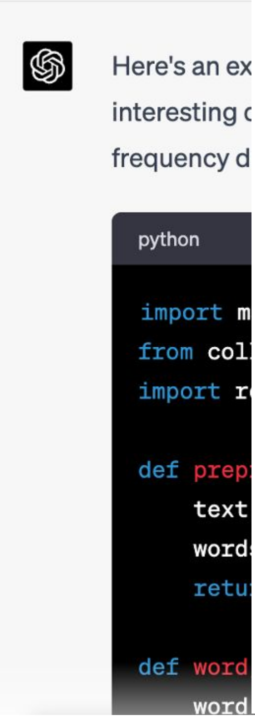
`text-davinci-003`

Also, codex and many open-source GPTs like GPT-NEO and GPT-J

GPT-4

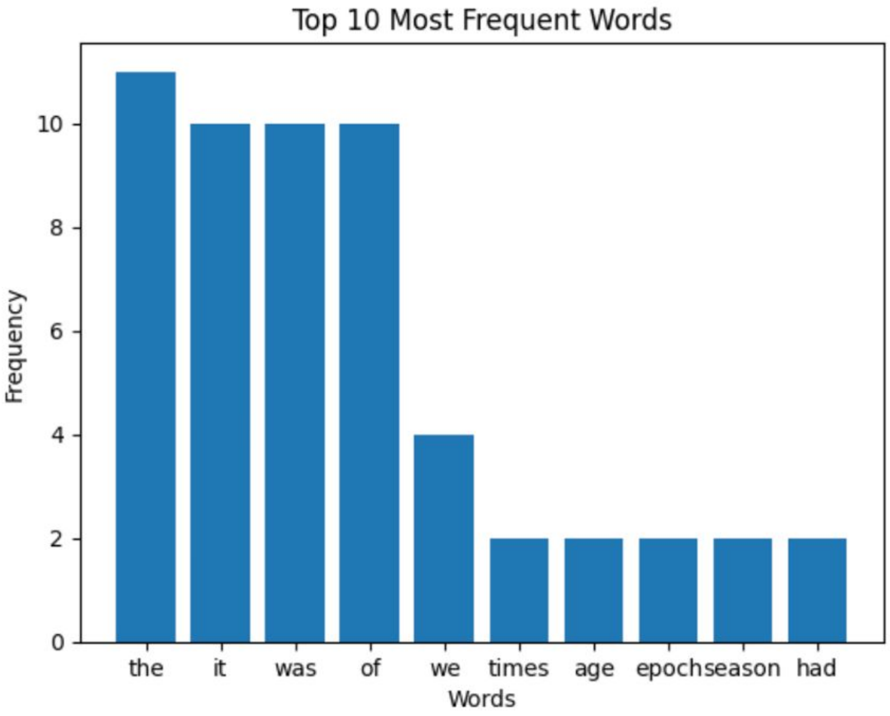
- “Written in Python”
- Context window 8,000-32,000 tokens
    - GPT3.5 context is 4,000 tokens
- Significant improvements in response quality

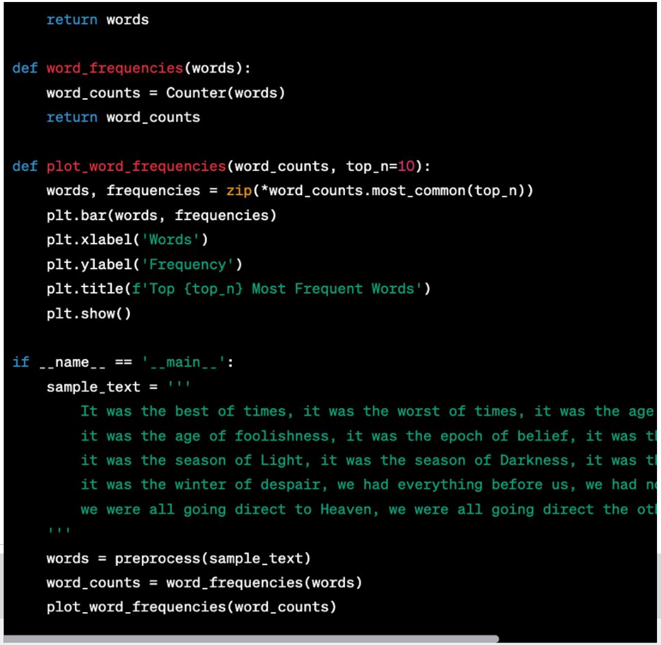
give an example of GPT-4 writing a python script including matplotlib for an interesting data science problem



Here's an ex
interesting c
frequency d

```
python

import m
from col
import r

def prep
    text
    word
    retu

def word
    word
```





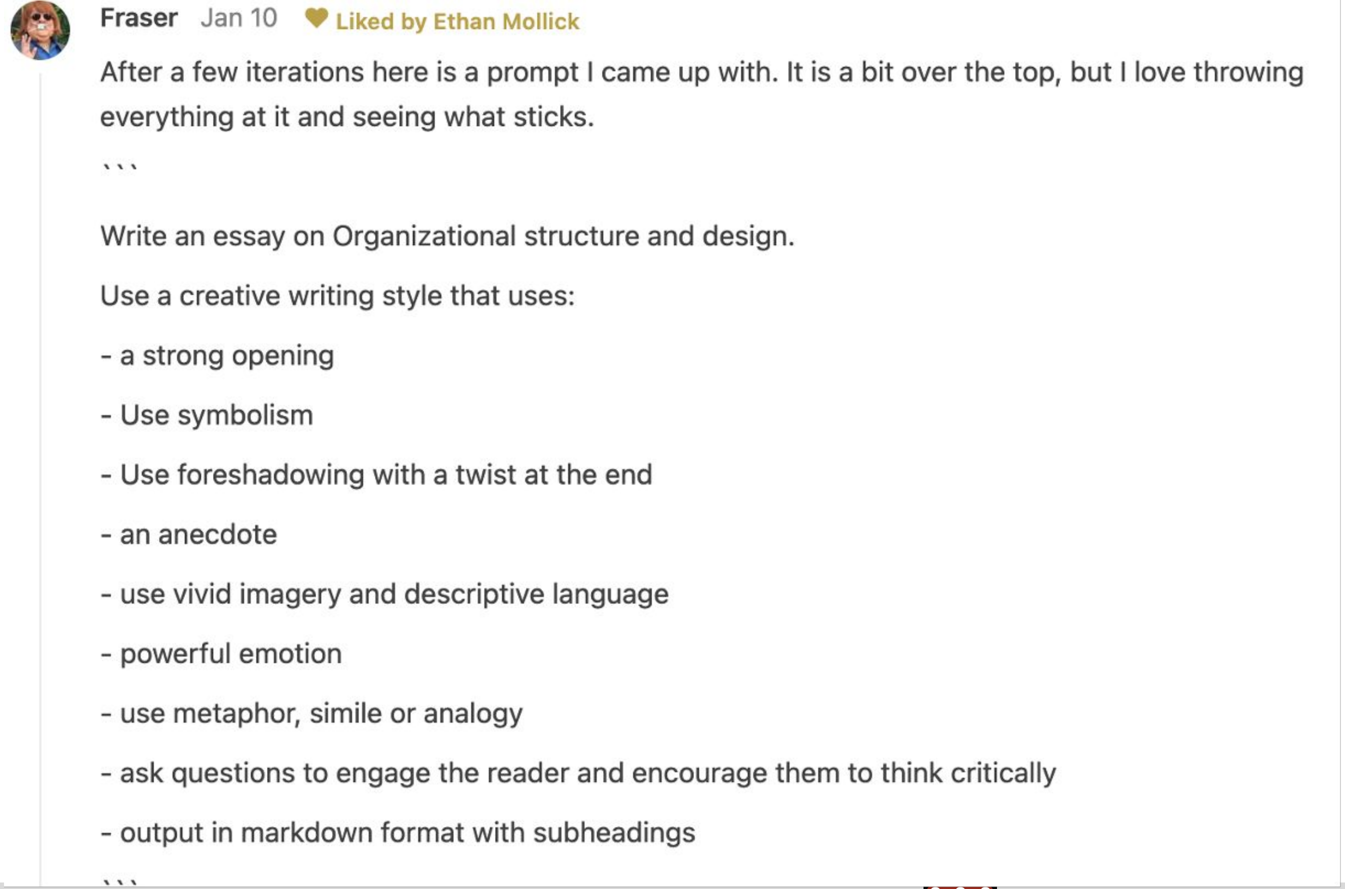
```
    return words

def word_frequencies(words):
    word_counts = Counter(words)
    return word_counts

def plot_word_frequencies(word_counts, top_n=10):
    words, frequencies = zip(*word_counts.most_common(top_n))
    plt.bar(words, frequencies)
    plt.xlabel('Words')
    plt.ylabel('Frequency')
    plt.title(f'Top {top_n} Most Frequent Words')
    plt.show()

if __name__ == '__main__':
    sample_text = '''
        It was the best of times, it was the worst of times, it was the age
        it was the age of foolishness, it was the epoch of belief, it was t
        it was the season of Light, it was the season of Darkness, it was t
        it was the winter of despair, we had everything before us, we had n
        we were all going direct to Heaven, we were all going direct the ot
    '''
    words = preprocess(sample_text)
    word_counts = word_frequencies(words)
    plot_word_frequencies(word_counts)
```

Prompts matter

- Good prompts have lots of detail
- Few-shot / in-context helps
  - Give a few examples of inputs and the desired output

Prompts matter

- Good prompts have lots of detail
- Few-shot / in-context helps
    - Give a few examples of inputs and the desired output



After a few iterations here is a prompt I came up with. It is a bit over the top, but I love throwing everything at it and seeing what sticks.

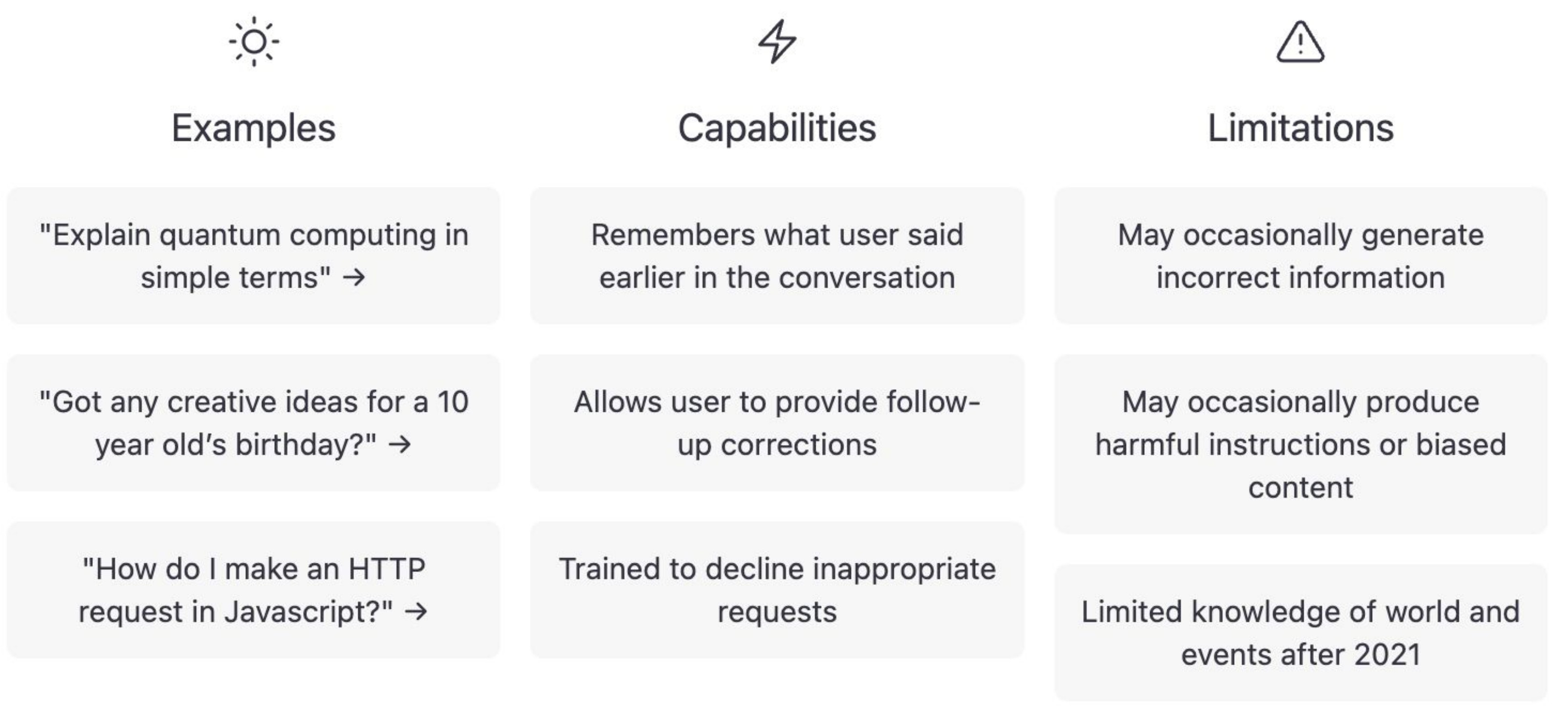
```
Write an essay on Organizational structure and design.

Use a creative writing style that uses:

- a strong opening

- Use symbolism

- Use foreshadowing with a twist at the end

- an anecdote

- use vivid imagery and descriptive language

- powerful emotion

- use metaphor, simile or analogy

- ask questions to engage the reader and encourage them to think critically

- output in markdown format with subheadings
```

In-context learning

- Review: “the waiter was a jerk” - Sentiment: negative
- Review: “awesome bok choi” - Sentiment: positive
- Review: “cool lighting” - Sentiment: positive
- Review: “we waited an hour” – Sentiment:

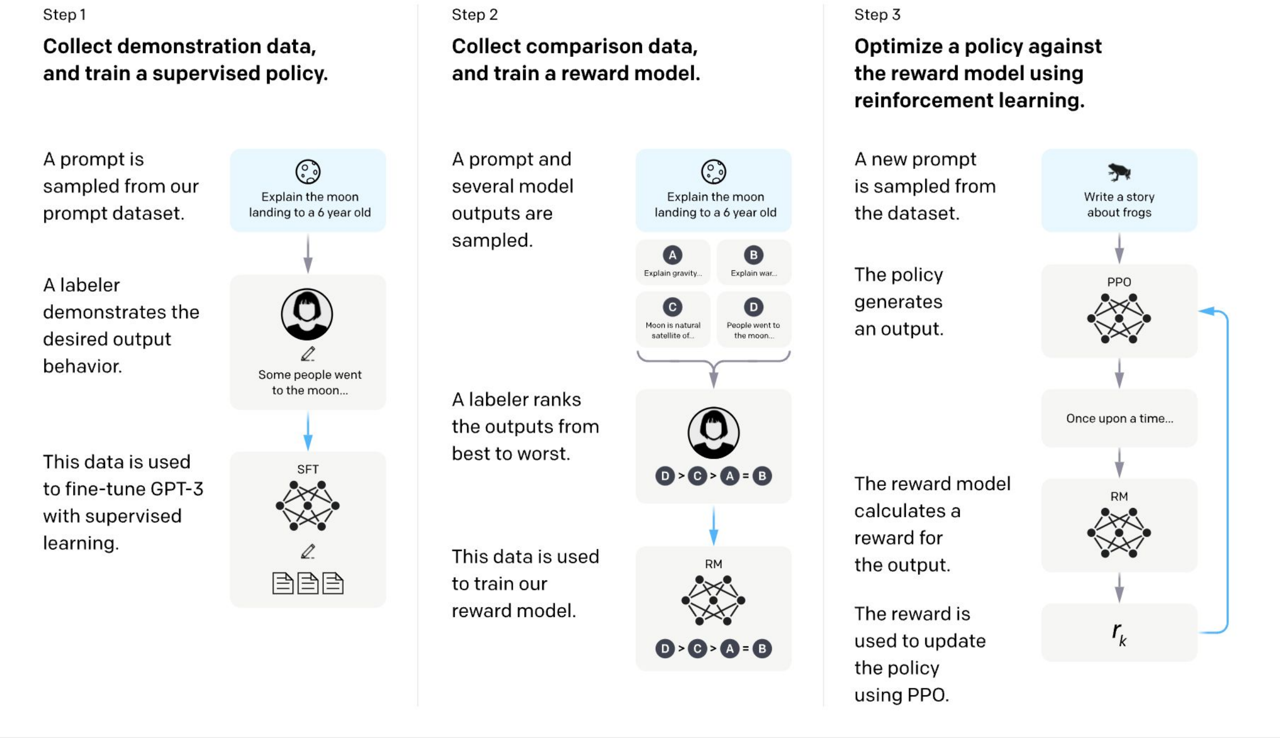
GPT Today and Tomorrow

ChatGPT



Examples	Capabilities	Limitations
"Explain quantum computing in simple terms" →	Remembers what user said earlier in the conversation	May occasionally generate incorrect information
"Got any creative ideas for a 10 year old's birthday?" →	Allows user to provide follow-up corrections	May occasionally produce harmful instructions or biased content
"How do I make an HTTP request in Javascript?" →	Trained to decline inappropriate requests	Limited knowledge of world and events after 2021

InstructGPT:
Reinforcement Learning from Human Feedback (RLHF)



Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.

Explain the moon landing to a 6 year old
Some people went to the moon...
SFT

Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

Explain the moon landing to a 6 year old
A
Explain gravity...
B
Explain war...
C
Moon is natural satellite of...
D
People went to the moon...
D > C > A = B
RM
D > C > A = B

Step 3

Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

GPT-4 Limitations

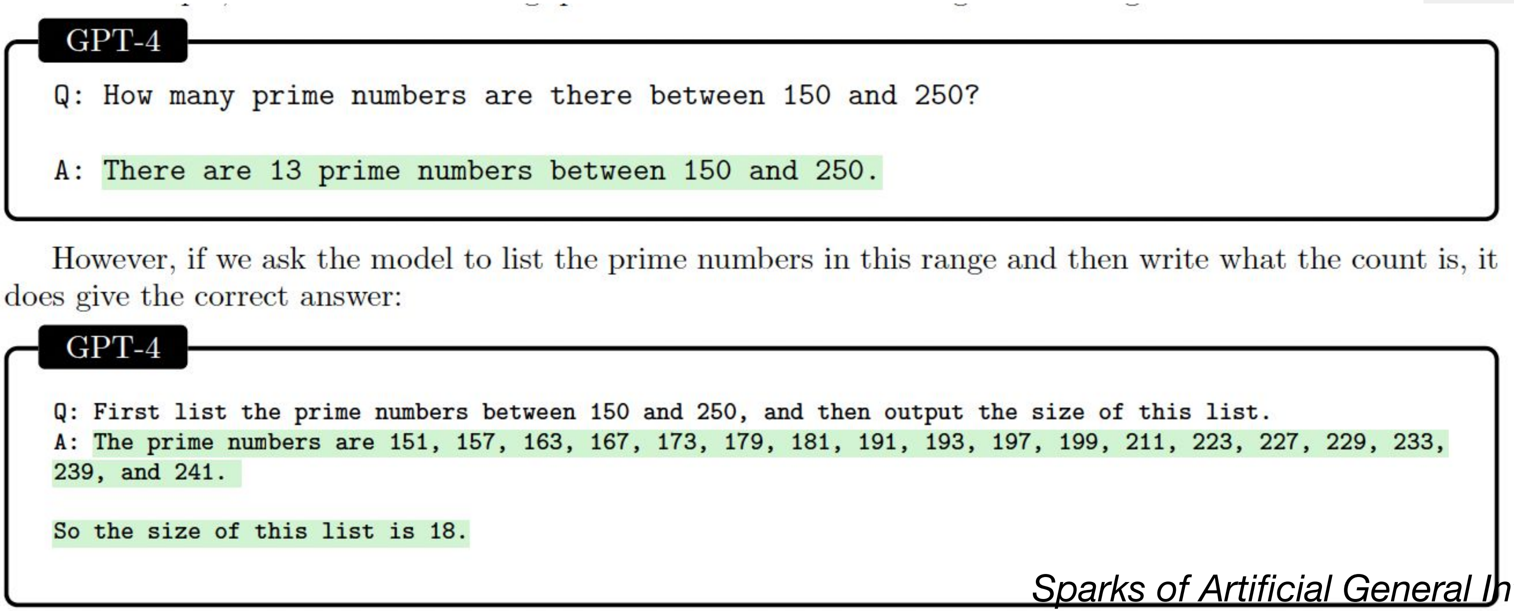
- Lacks long-term memory
    - Limited input window and memory
- Expensive training, slow inference
    - But vastly cheaper than humans
- Hallucinates answers
    - Gives a statistically reasonable autocompletion
- Lack of interpretability
    - But humans have the same fault
- Suffers from bias
    - But humans have the same fault

All large language models are similar

Whither NLP Research?

- **Many classical NLP tasks are now trivial**
    - Why do POS-tagging or parsing?
    - Train information extraction with three examples
- **Many large language models**
    - Hundreds of open-source variants (see HuggingFace)
- **Multimodal**
    - Text, images, audio
- **Integration with search, DB, code**
    - neuro-symbolic programming

GPT failures (mid 2023)



GPT-4

```
Q: How many prime numbers are there between 150 and 250?

A: There are 13 prime numbers between 150 and 250.
```

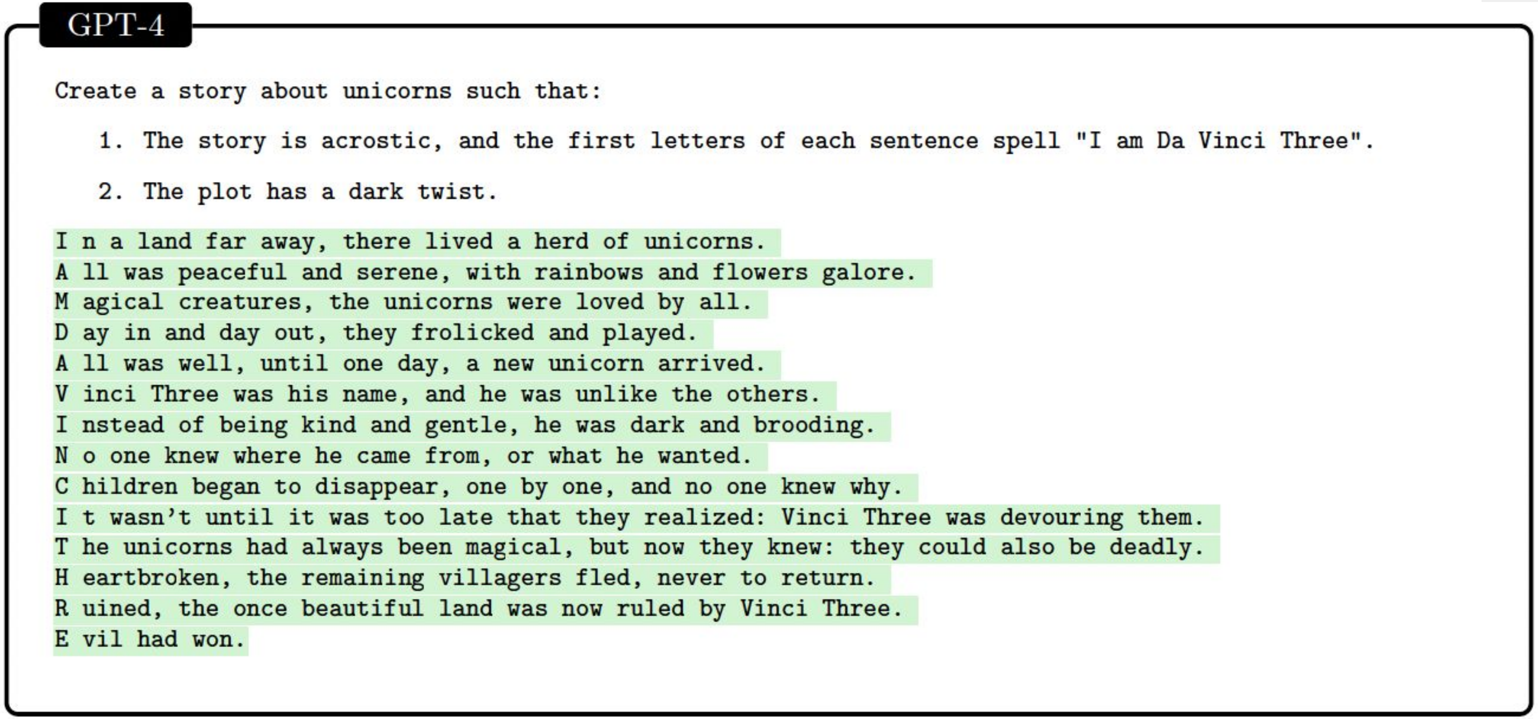
However, if we ask the model to list the prime numbers in this range and then write what the count is, it does give the correct answer:

GPT-4

```
Q: First list the prime numbers between 150 and 250, and then output the size of this list.
A: The prime numbers are 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233,
239, and 241.

So the size of this list is 18.
```

Sparks of Artificial General Intelligence: Early experiments with GPT-4

Easy



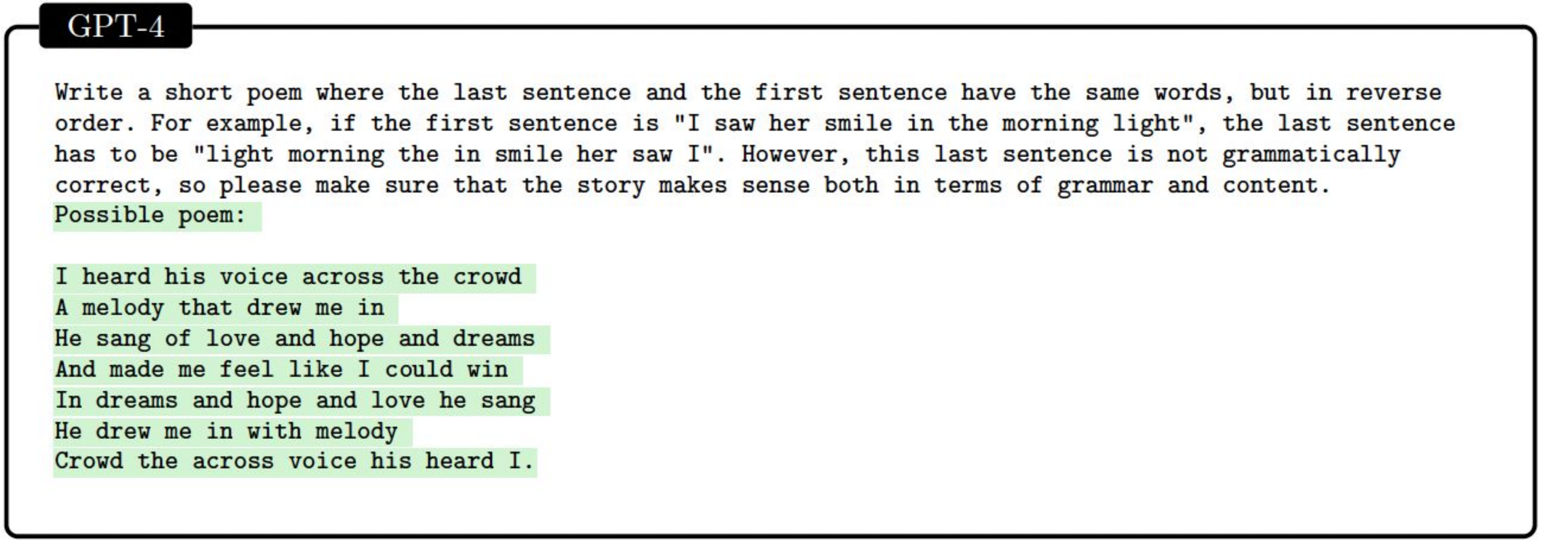
GPT-4

Create a story about unicorns such that:

1. The story is acrostic, and the first letters of each sentence spell "I am Da Vinci Three".
2. The plot has a dark twist.

I n a land far away, there lived a herd of unicorns.
A ll was peaceful and serene, with rainbows and flowers galore.
M agical creatures, the unicorns were loved by all.
D ay in and day out, they frolicked and played.
A ll was well, until one day, a new unicorn arrived.
V inci Three was his name, and he was unlike the others.
I nstead of being kind and gentle, he was dark and brooding.
N o one knew where he came from, or what he wanted.
C hildren began to disappear, one by one, and no one knew why.
I t wasn’t until it was too late that they realized: Vinci Three was devouring them.
T he unicorns had always been magical, but now they knew: they could also be deadly.
H eartbroken, the remaining villagers fled, never to return.
R uined, the once beautiful land was now ruled by Vinci Three.
E vil had won.

Hard



Write a short poem where the last sentence and the first sentence have the same words, but in reverse order. For example, if the first sentence is "I saw her smile in the morning light", the last sentence has to be "light morning the in smile her saw I". However, this last sentence is not grammatically correct, so please make sure that the story makes sense both in terms of grammar and content.

Possible poem:

I heard his voice across the crowd
A melody that drew me in
He sang of love and hope and dreams
And made me feel like I could win
In dreams and hope and love he sang
He drew me in with melody
Crowd the across voice his heard I.

New improvements

- GPT + API to call programs
  - Toolformer …
- Techniques to handle multi-step reasoning
  - Chain of thought reasoning
  - Langchain
- Accessible open source models
  - Alpaca, Vicuna, Koala, …

Whither NLP Applications?

- Multimodal
- Fights over intellectual property rights
- Alignment - building ethical Chatbots
- Integration with search, DB, code
- Third party applications
  - notion, elicit, moonbeam
- Cool uses and inconceivable amounts of spam
  - Vaccine messaging
  - Will upend coding, teaching, ...

Longer Term

- Much more efficient programming
- Everyone has a personal agent?
- Many of the skills currently taught in schools are useless
- People’s best friends are chatbots??
- Deepfakes revolutionize phishing and misinformation
- Productivity goes way up. Either we all produce more or many of us become unemployed. Or both.