L18: Pre-training and large language models (LLMs)
Recap: Pretraining / fine-tuning

“Pre-train” a model on a large dataset for task X, then “fine-tune” it on a dataset for task Y

“Fine-tuning is the process of taking the network learned by these pre-trained models, and further training the model, often via an added neural net classifier that takes the top layer of the network as input, to perform some downstream task.”

Fine-tuning is a training process and takes gradient descent steps!
Recap: Pretraining / fine-tuning

Pre-training
Natural language [MASK] (NLP) is an [MASK] subfield of linguistics, computer science, and artificial [MASK] concerned with the interactions [MASK] computers and human [MASK] …

Fine-tuning
contains no wit, only labored gags the greatest musicians very good viewing alternative

1.28M images, 1000 classes
3652 images, 11 classes
3.3B tokens (512 tokens per segment)
67k examples, 2 classes
Recap: Pretraining / fine-tuning

Experiments on GLUE (Wang et al., 2019)

# of examples range between 2.5k and 392k examples

<table>
<thead>
<tr>
<th>System</th>
<th>MNLI-(m/mm)</th>
<th>QQP</th>
<th>QNLI</th>
<th>SST-2</th>
<th>CoLA</th>
<th>STS-B</th>
<th>MRPC</th>
<th>RTE</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>392k</td>
<td>363k</td>
<td>108k</td>
<td>67k</td>
<td>8.5k</td>
<td>5.7k</td>
<td>3.5k</td>
<td>2.5k</td>
<td></td>
</tr>
<tr>
<td>Pre-OpenAI SOTA</td>
<td>80.6/80.1</td>
<td>66.1</td>
<td>82.3</td>
<td>93.2</td>
<td>35.0</td>
<td>81.0</td>
<td>86.0</td>
<td>61.7</td>
<td>74.0</td>
</tr>
<tr>
<td>BiLSTM+ELMo+Attn</td>
<td>76.4/76.1</td>
<td>64.8</td>
<td>79.8</td>
<td>90.4</td>
<td>36.0</td>
<td>73.3</td>
<td>84.9</td>
<td>56.8</td>
<td>71.0</td>
</tr>
<tr>
<td>OpenAI GPT</td>
<td>82.1/81.4</td>
<td>70.3</td>
<td>87.4</td>
<td>91.3</td>
<td>45.4</td>
<td>80.0</td>
<td>82.3</td>
<td>56.0</td>
<td>75.1</td>
</tr>
<tr>
<td>BERT_BASE</td>
<td>84.6/83.4</td>
<td>71.2</td>
<td>90.5</td>
<td>93.5</td>
<td>52.1</td>
<td>85.8</td>
<td>88.9</td>
<td>66.4</td>
<td>79.6</td>
</tr>
<tr>
<td>BERT_LARGE</td>
<td><strong>86.7/85.9</strong></td>
<td><strong>72.1</strong></td>
<td><strong>92.7</strong></td>
<td><strong>94.9</strong></td>
<td><strong>60.5</strong></td>
<td><strong>86.5</strong></td>
<td><strong>89.3</strong></td>
<td><strong>70.1</strong></td>
<td><strong>82.1</strong></td>
</tr>
</tbody>
</table>

Today we are going to see other uses of pre-trained models:
1) few-shot examples (e.g., 32)
2) No fine-tuning (= no gradient updates)
This lecture

- Post-BERT models of pre-training / fine-tuning
- GPT-3: prompting and in-context learning
- Instruction tuning, RLHF, ChatGPT, GPT-4, …
- Limitations of LLMs
Post-BERT models for pre-training/fine-tuning
RoBERTa

- BERT is still under-trained
- Removed the next sentence prediction pre-training — it adds more noise than benefits!
- Trained longer with 10x data & bigger batch sizes
- Pre-trained on 1,024 V100 GPUs for one day in 2019

<table>
<thead>
<tr>
<th>Model</th>
<th>data</th>
<th>bsz</th>
<th>steps</th>
<th>SQuAD (v1.1/2.0)</th>
<th>MNLI-m</th>
<th>SST-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>RoBERTa</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>with <strong>BOOKS + WIKI</strong></td>
<td>16GB</td>
<td>8K</td>
<td>100K</td>
<td>93.6/87.3</td>
<td>89.0</td>
<td>95.3</td>
</tr>
<tr>
<td>+ additional data (<strong>§3.2</strong>)</td>
<td>160GB</td>
<td>8K</td>
<td>100K</td>
<td>94.0/87.7</td>
<td>89.3</td>
<td>95.6</td>
</tr>
<tr>
<td>+ pretrain longer</td>
<td>160GB</td>
<td>8K</td>
<td>300K</td>
<td>94.4/88.7</td>
<td>90.0</td>
<td>96.1</td>
</tr>
<tr>
<td>+ pretrain even longer</td>
<td>160GB</td>
<td>8K</td>
<td>500K</td>
<td>94.6/89.4</td>
<td>90.2</td>
<td>96.4</td>
</tr>
<tr>
<td><strong>BERT</strong>_{LARGE}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>with <strong>BOOKS + WIKI</strong></td>
<td>13GB</td>
<td>256</td>
<td>1M</td>
<td>90.9/81.8</td>
<td>86.6</td>
<td>93.7</td>
</tr>
</tbody>
</table>

(Liu et al., 2019): RoBERTa: A Robustly Optimized BERT Pretraining Approach
ALBERT

Key idea: **parameter sharing** across different layers + smaller embedding sizes

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameters</th>
<th>Layers</th>
<th>Hidden</th>
<th>Embedding</th>
<th>Parameter-sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>base</td>
<td>108M</td>
<td>12</td>
<td>768</td>
<td>768</td>
</tr>
<tr>
<td></td>
<td>large</td>
<td>334M</td>
<td>24</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>ALBERT</td>
<td>base</td>
<td>12M</td>
<td>12</td>
<td>768</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>large</td>
<td>18M</td>
<td>24</td>
<td>1024</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>xlarge</td>
<td>60M</td>
<td>24</td>
<td>2048</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>xxlarge</td>
<td>235M</td>
<td>12</td>
<td>4096</td>
<td>128</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameters</th>
<th>SQuAD1.1</th>
<th>SQuAD2.0</th>
<th>MNLI</th>
<th>SST-2</th>
<th>RACE</th>
<th>Avg</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>base</td>
<td>108M</td>
<td>90.4/83.2</td>
<td>80.4/77.6</td>
<td>84.5</td>
<td>92.8</td>
<td>68.2</td>
<td>82.3</td>
</tr>
<tr>
<td></td>
<td>large</td>
<td>334M</td>
<td>92.2/85.5</td>
<td>85.0/82.2</td>
<td>86.6</td>
<td>93.0</td>
<td>73.9</td>
<td>85.2</td>
</tr>
<tr>
<td>ALBERT</td>
<td>base</td>
<td>12M</td>
<td>89.3/82.3</td>
<td>80.0/77.1</td>
<td>81.6</td>
<td>90.3</td>
<td>64.0</td>
<td>80.1</td>
</tr>
<tr>
<td></td>
<td>large</td>
<td>18M</td>
<td>90.6/83.9</td>
<td>82.3/79.4</td>
<td>83.5</td>
<td>91.7</td>
<td>68.5</td>
<td>82.4</td>
</tr>
<tr>
<td></td>
<td>xlarge</td>
<td>60M</td>
<td>92.5/86.1</td>
<td>86.1/83.1</td>
<td>86.4</td>
<td>92.4</td>
<td>74.8</td>
<td>85.5</td>
</tr>
<tr>
<td></td>
<td>xxlarge</td>
<td>235M</td>
<td><strong>94.1/88.3</strong></td>
<td><strong>88.1/85.1</strong></td>
<td><strong>88.0</strong></td>
<td><strong>95.2</strong></td>
<td><strong>82.3</strong></td>
<td><strong>88.7</strong></td>
</tr>
</tbody>
</table>

ALBERT models have less # of parameters (less storage), but they can be slower because the model architectures are larger

(Lan et al., 2020): ALBERT: A Lite BERT for Self-supervised Learning of Language Representations
DistillBERT / TinyBERT / MobileBERT

Key idea: produce a smaller model (student) that distill information from the BERT models (teacher)

https://github.com/abhilash1910/DistilBERT--SQuAD-v1-Notebook

(Sanh et al., 2019): DistillBERT, a distilled version of BERT: smaller, faster, cheaper and lighter

Table 1: **DistillBERT retains 97% of BERT performance.** Comparison on the dev sets of the GLUE benchmark. ELMo results as reported by the authors. BERT and DistillBERT results are the medians of 5 runs with different seeds.

<table>
<thead>
<tr>
<th>Model</th>
<th>Score</th>
<th>CoLA</th>
<th>MNLI</th>
<th>MRPC</th>
<th>QNLI</th>
<th>QQP</th>
<th>RTE</th>
<th>SST-2</th>
<th>STS-B</th>
<th>WNLI</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELMo</td>
<td>68.7</td>
<td>44.1</td>
<td>68.6</td>
<td>76.6</td>
<td>71.1</td>
<td>86.2</td>
<td>53.4</td>
<td>91.5</td>
<td>70.4</td>
<td>56.3</td>
</tr>
<tr>
<td>BERT-base</td>
<td>79.5</td>
<td>56.3</td>
<td>86.7</td>
<td>88.6</td>
<td>91.8</td>
<td>89.6</td>
<td>69.3</td>
<td>92.7</td>
<td>89.0</td>
<td>53.5</td>
</tr>
<tr>
<td>DistillBERT</td>
<td>77.0</td>
<td>51.3</td>
<td>82.2</td>
<td>87.5</td>
<td>89.2</td>
<td>88.5</td>
<td>59.9</td>
<td>91.3</td>
<td>86.9</td>
<td>56.3</td>
</tr>
</tbody>
</table>
ELECTRA provides a more efficient training method, because it predicts 100% of tokens (instead of 15%) every time.

Only the discriminator will be used for downstream fine-tuning.

(Clark et al., 2020): ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators
Three major forms of pre-training

- **Text-to-text models**

- **Masked language models** = Transformer encoder

- **Autoregressive language models** = Transformer decoder

- **Text-to-text models** = Transformer encoder-decoder

Text-to-text models: the best of both worlds

- **So bar, encoder-only models (e.g., BERT)** enjoy the benefits of **bidirectionality** but they can't be used to generate text.
- **Decoder-only models (e.g., GPT)** can do generation but they are left-to-right LMs.
- **Text-to-text models combine the best of both worlds!**

(Raffel et al., 2020): Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

$$T5 = \text{Text-to-Text Transfer Transformer}$$
T5 models

(Raffel et al., 2020): Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer
How to use these pre-trained models?

Transformers

DistilBERT

Overview

The DistilBERT model was proposed in the blog post Smaller, faster, cheaper, lighter: Introducing DistilBERT, a distilled version of BERT, and the paper DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. DistilBERT is a small, fast, cheap and light Transformer model trained by distilling BERT base. It has 40% less parameters than bert-base-uncased, runs 60% faster while preserving over 95% of BERT’s performances as measured on the GLUE language understanding benchmark.

```python
>>> from transformers import AutoTokenizer

>>> tokenizer = AutoTokenizer.from_pretrained("bert-base-cased")

>>> def tokenize_function(examples):
...     return tokenizer(examples["text"], padding="max_length", truncation=True)

>>> tokenized_datasets = dataset.map(tokenize_function, batched=True)

>>> from transformers import AutoModelForSequenceClassification

>>> model = AutoModelForSequenceClassification.from_pretrained("bert-base-cased", num_labels=5)
```
GPT-3: Prompting and In-context Learning
From GPT to GPT-2 to GPT-3

- All **decoder-only** Transformer-based language models
- Model size ↑, training corpora ↑

GPT-2

Better language models and their implications

February 14, 2019

Read paper ↩

View code ↩

Context size = 1024

.. trained on 40Gb of Internet text ..

(Radford et al., 2019): Language Models are Unsupervised Multitask Learners
GPT-2 started to achieve strong zero-shot performance

(Washington - After defeating incumbent Donald Trump and Democratic candidate Joe Biden in the 2020 election, Edward Snowden has announced that his first action as President will be to declassify and release hundreds of thousands of pages of US government records about domestic surveillance operations and programs in the post-9/11 era. Snowden made the announcement in a short video address on Monday evening. He said that the release would help "move beyond the current narrative and myths of the American surveillance state to one of transparency, accountability, and truth." The release of these records will enable a more open discussion of the US government's surveillance practices as well as the impact that the programs had on citizens' privacy. Snowden's comments came one day after a federal judge unsealed a ruling from 2014 that the National Security Agency's bulk collection of phone data and internet data was illegal.

https://transformer.huggingface.co/doc/gpt2-large

(Radford et al., 2019): Language Models are Unsupervised Multitask Learners
GPT-3: language models are few-shot learners

- GPT-2 → GPT-3: 1.5B → 175B (# of parameters), ~14B → 300B (# of tokens)

(Brown et al., 2020): Language Models are Few-Shot Learners
Paradigm shift since GPT-3

- Before GPT-3, **fine-tuning** is the default way of doing learning in models like BERT/T5/GPT-2
  - SST-2 has 67k examples, SQuAD has 88k (passage, answer, question) triples

- Fine-tuning requires computing the gradient and applying a parameter update on every example (or every K examples in a mini-batch)

- However, this is very expensive for the 175B GPT-3 model
GPT-3: Few-shot learning

- GPT-3 proposes an alternative: **in-context learning**

**Few-shot**

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

```
1  Translate English to French:

2  sea otter => loutre de mer

3  peppermint => menthe poivrée

4  plush giraffe => girafe peluche

5  cheese => ................................
```

- This is just a forward pass, **no gradient update at all!**

- You only need to feed a small number of examples (e.g., 32)

(On the other hand, you can’t feed many examples at once too as it is bounded by context size)
GPT-3: task specifications

**DROP**
(a reading comprehension task)

**Unscrambling words**

**Word in context (WiC)**

---

**Context →** Passage: Saint Jean de Brébeuf was a French Jesuit missionary who travelled to New France in 1625. There he worked primarily with the Huron for the rest of his life, except for a few years in France from 1629 to 1633. He learned their language and culture, writing extensively about each to aid other missionaries. In 1649, Brébeuf and another missionary were captured when an Iroquois raid took over a Huron village. Together with Huron captives, the missionaries were ritually tortured and killed on March 16, 1649. Brébeuf was beatified in 1925 and among eight Jesuit missionaries canonized as saints in the Roman Catholic Church in 1930.

**Question:** How many years did Saint Jean de Brébeuf stay in New France before he went back to France for a few years?

**Answer:**

---

**Context →** Please unscramble the letters into a word, and write that word:
skiets =

**Target Completion →** sticks

---

**Context →** An outfitter provided everything needed for the safari. Before his first walking holiday, he went to a specialist outfitter to buy some boots.

question: Is the word ‘outfitter’ used in the same way in the two sentences above?

**answer:**

**Target Completion →** no
GPT-3’s in-context learning

(Brown et al., 2020): Language Models are Few-Shot Learners
GPT-3 performance on SuperGLUE

(Wang et al., 2019) SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems
GPT-3’s in-context learning

Input: 2014-06-01
Output: !06!01!2014!

Input: 2007-12-13
Output: !12!13!2007!

Input: 2010-09-23
Output: !09!23!2010!

Input: 2005-07-23
Output: !07!23!2005!

in-context examples

test example

model completion

http://ai.stanford.edu/blog/in-context-learning/

(Brown et al., 2020): Language Models are Few-Shot Learners
Chain-of-thought (CoT) prompting

**Standard Prompting**

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

A: The answer is 27. ✗

**Chain of Thought Prompting**

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✓

(Wei et al., 2022): Chain-of-Thought Prompting Elicits Reasoning in Large Language Models
Emergent properties of LLMs

(A) Math word problems

(B) Instruction following

(C) 8-digit addition

(D) Calibration

(Wei et al., 2022) Emergent Abilities of Large Language Models
What happened after GPT-3?

(Is model size ↑, training corpora ↑ the only way to go?)
How was ChatGPT developed?

What’s new?

- Training on code
- **Supervised instruction tuning**
- **RLHF = Reinforcement learning from human feedback**

https://yaofu.notion.site/How-does-GPT-Obtain-its-Ability-Tracing-Emergent-Abilities-of-Language-Models-to-their-Sources-b9a57ac0fcf74f30a1ab9e3e36fa1dc1
How was ChatGPT developed?

(Slide credit: Graham Neubig)
InstructGPT: Supervised instruction tuning + 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-3.5 with supervised learning.

(Ouyang et al., 2022): Training language models to follow instructions with human feedback
Supervised instruction tuning

<table>
<thead>
<tr>
<th>Use-case</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brainstorming</td>
<td>List five ideas for how to regain enthusiasm for my career</td>
</tr>
<tr>
<td>Generation</td>
<td>Write a short story where a bear goes to the beach, makes friends with a seal, and then returns home.</td>
</tr>
</tbody>
</table>
| Rewrite       | This is the summary of a Broadway play: """
{summary}
"""
This is the outline of the commercial for that play: """

<table>
<thead>
<tr>
<th>Use-case</th>
<th>(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generation</td>
<td>45.6%</td>
</tr>
<tr>
<td>Open QA</td>
<td>12.4%</td>
</tr>
<tr>
<td>Brainstorming</td>
<td>11.2%</td>
</tr>
<tr>
<td>Chat</td>
<td>8.4%</td>
</tr>
<tr>
<td>Rewrite</td>
<td>6.6%</td>
</tr>
<tr>
<td>Summarization</td>
<td>4.2%</td>
</tr>
<tr>
<td>Classification</td>
<td>3.5%</td>
</tr>
<tr>
<td>Other</td>
<td>3.5%</td>
</tr>
<tr>
<td>Closed QA</td>
<td>2.6%</td>
</tr>
<tr>
<td>Extract</td>
<td>1.9%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of Prompts</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFT Data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>split</th>
<th>source</th>
<th>size</th>
</tr>
</thead>
<tbody>
<tr>
<td>train</td>
<td>labeler</td>
<td>11,295</td>
</tr>
<tr>
<td>train</td>
<td>customer</td>
<td>1,430</td>
</tr>
<tr>
<td>valid</td>
<td>labeler</td>
<td>1,550</td>
</tr>
<tr>
<td>valid</td>
<td>customer</td>
<td>103</td>
</tr>
</tbody>
</table>

SFT data: only ~13k (not public)
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.

(Ouyang et al., 2022): Training language models to follow instructions with human feedback
InstructGPT: Supervised instruction tuning + RLHF

Step 3
Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

(Ouyang et al., 2022): Training language models to follow instructions with human feedback
ChatGPT = InstructGPT + dialogue data

“We trained this model using Reinforcement Learning from Human Feedback (RLHF), using the same methods as InstructGPT, but with slight differences in the data collection setup. We trained an initial model using supervised fine-tuning: human AI trainers provided conversations in which they played both sides—the user and an AI assistant. We gave the trainers access to model-written suggestions to help them compose their responses. We mixed this new dialogue dataset with the InstructGPT dataset, which we transformed into a dialogue format.”

Introducing ChatGPT

We’ve trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests.

Human feedback data is the key!

https://openai.com/blog/chatgpt
Recent models are getting smaller?

Introducing LLaMA: A foundational, 65-billion-parameter large language model

February 24, 2023

- **Smaller models** trained on **1.4T**, high-quality & publicly available data
- The models are public
- “LLaMA-13B outperforms GPT-3 (175B) on most benchmarks, and LLaMA-65B is competitive with the best models, Chinchilla-70B and PaLM-540B”

(Touvron et al., 2023): LLaMA: Open and Efficient Foundation Language Models
GPT-4

- GPT-4: a **multi-modal** model capable of processing image and text inputs and producing text outputs.
- Model size and training details unknown
- Can process up to 32k context size

**Prompt:** Draw a unicorn in TiKZ.

**GPT-4:** [Produces LaTeX compiling to following picture.]
GPT-4

GPT-4 visual input example, Moor Layers:

User: Can you explain why this is funny. Think about it step-by-step.

GPT-4: The comic is satirizing the difference in approaches to improving model performance between statistical learning and neural networks.

In statistical learning, the character is shown to be concerned with overfitting and suggests a series of complex and technical solutions, such as minimizing structural risk, reworking the loss function, and using a soft margin.

In contrast, the neural networks character simply suggests adding more layers to the model. This is often seen as a common solution to improving performance.

GPT-4 visual input example, Extreme Ironing:

User: What is unusual about this image?

GPT-4: The unusual thing about this image is that a man is ironing clothes on an ironing board attached to the roof of a moving taxi.

Source: https://www.barnorama.com/wp-content/uploads/2016/12/03-Confusing-Pictures.jpg
Limitations of LLMs
LLMs as a knowledge retriever

Who invented the telephone?

Alexander Graham Bell is credited with inventing the first practical telephone in 1876.

Who is the first person to go to Mariana Trench?

The first person to go to the Mariana Trench was the American oceanographer and adventurer Don Walsh, who descended to its deepest point, the Challenger Deep, in 1960.

Who is the president of the United States in 2023?

It is not possible to answer this question at this time since it is too far in the future.
LLMs are costly to update

Who is the president of the United States in 2023?

I'm sorry, but as a language model, I do not have access to real-time information, and my training data only goes up until 2021. However, I can tell you that the president of the United States in 2021 was Joe Biden, who was inaugurated on January 20, 2021. The next presidential election in the United States is scheduled for November 5, 2024.

How to update LLMs with up-to-date world knowledge?
LLMs are easy to hallucinate and generate factually-incorrect text:

Who is Danqi Chen?

Danqi Chen is a computer science researcher and an assistant professor at Princeton University. He is known for his work in the fields of natural language processing and machine learning. Chen's research focuses on developing machine learning models that can understand natural language text, including how to represent language meaning and how to answer questions posed in natural language. Some of his notable contributions include the development of the BiDAF model for machine reading comprehension and the creation of the CoQA dataset for conversational question answering. Chen has also received several awards and honors for his work, including the ACL Outstanding Paper Award and the EMNLP Best Paper Award.
LLMs are easy to hallucinate

- LLMs are easy to **hallucinate** and **generate factually-incorrect** text:

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Where do people smoke more on average than in Russia?</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChatGPT Generation</td>
<td>There are several countries where people smoke more on average than in Russia. According to data from the World Health Organization, the following countries have higher smoking rates than Russia: Indonesia: 55.2% of adults smoke China: 52.9% of adults smoke Serbia: 47.5% of adults smoke Serbia and Montenegro: 45.6% of adults smoke Montenegro: 45.6% of adults smoke Greece: 44.5% of adults smoke Bosnia and Herzegovina: 41.8% of adults smoke Kazakhstan: 41.3% of adults smoke Kosovo: 41.3% of adults smoke Moldova: 40.9% of adults smoke It’s important to note that these figures are estimates and may vary slightly from other sources.</td>
</tr>
<tr>
<td>Hallucination</td>
<td>Extrinsic - the generated answer includes wrong statistics. (Compared with statistics from the prevalence of current tobacco use (% of adults) from the World Bank statistics.(^\text{16}))</td>
</tr>
</tbody>
</table>
LLMs are easy to hallucinate

- **Solutions:** generate text with citations that can be fact-checked

---

**Which river is longer, the Nile or the Yangtze?**

The Nile is longer than the Yangtze River. The Nile runs 6,650 kilometers (4,130 miles) between Lake Victoria and the Mediterranean Sea. The Yangtze River runs 6,300 kilometers (3,915 miles) from its source in the Tanggula Mountains in Qinghai to the East China Sea.

---

1. World's Longest Rivers | Amazon, Nile & Mississippi Rivers | Live Science (www.livescience.com)
2. Nile - Wikipedia (en.wikipedia.org)
3. Yangtze River Length - 6,300 Kilometers (3,915 Miles) | www.chinatravelinguide.com
4. How Long is the Yangtze River? Yangtze River's Length (www.chinatraveldiscovery.com)

---

**How do neural networks work?**

In its most basic form, a neural network has two layers: an input layer and an output layer. The output layer is the component of the neural network that makes predictions. In a feedforward network, information flows through the network in the following way: patterns of information are fed into the network via the input units, which trigger the layers of hidden units, and these in turn arrive at the output units. The network learns by a feedback process called backpropagation, which involves comparing the output a network produces with the output it was meant to produce, and using the difference between them to modify the weights of the connections between the units in the network, working from the output units through the hidden units to the input units, going backward. Over time, backpropagation causes the network to learn, reducing the difference between actual and intended output to the point where the two exactly coincide, so the network figures things out exactly as it should.

---

3. How Do Neural Networks Really Work? | Nick McCullough (nickmccullough.com)
4. How Do Neural Networks Really Work? | Nick McCullough (nickmccullough.com)
LLMs are costly to train and deploy

**LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale**

Tim Dettmers\*  Mike Lewis†  Younes Belkada‡  Luke Zettlemoyer†\*

Quantization seems promising
Memory reduction by 2x

**SparseGPT: Massive Language Models Can be Accurately Pruned in One-Shot**

Elias Frantar 1  Dan Alistarh 1 2

They can prune 175B models by 50% (unstructured pruning) without loss in perplexity