

CS 1671/2071

Human Language Technologies

Session 17: Transformers part 2, introduction to LLMs

Michael Miller Yoder

March 19, 2025

Course logistics

- Quiz on Canvas **due tomorrow, Thu Mar 20**
- Project progress report is **due next Thu Mar 27**. See the [project website](#) for instructions
 - **Part 1:** Data statistics and exploratory data analysis (EDA)
 - **Part 2:** A result from baseline/initial approach
 - **Part 3:** Proposal on how to use LLMs for your task
 - **Part 4:** Open questions and challenges
- I am in the process of setting up OpenAI API account to use (\$150 for class). In the meantime look into using Gemini free credits or other LLMs

NLP talk by Anjalie Field 4:30pm today in SENSQ 5317

Anjalie Field

Johns Hopkins University

Time: 03/19/2025, 4:30 - 5:30 PM (EST)

Place: In-person (SQ 5317)

Bio: Anjalie Field is an Assistant Professor in the Computer Science Department at Johns Hopkins University. She is also affiliated with the Center for Language and Speech Processing (CLSP) and the new Data Science and AI Institute. Her research focuses on the ethics and social science aspects of natural language processing, which includes developing models to address societal issues like discrimination and propaganda, as well as critically assessing and improving ethics in AI pipelines. Her work has been published in NLP and interdisciplinary venues, like ACL and PNAS, and in 2024 she was named an AI2050 Early Career Fellow by Schmidt Futures. Prior to joining JHU, she was a postdoctoral researcher at Stanford, and she completed her PhD at the Language Technologies Institute at Carnegie Mellon University.

Title: Fairness and Privacy in High-Stakes NLP

Abstract: Practitioners are increasingly using algorithmic tools in high-stakes settings, like healthcare, social services, policing, and education with particular recent interest in natural language processing (NLP). These domains raise a number of challenges, including preserving data privacy, ensuring model reliability, and developing approaches that can mitigate, rather than exacerbate historical bias. In this talk, I will discuss our recent work investigating risks of racial bias in NLP child protective services and ways we aim to better preserve privacy for these types of audits in the future. Time permitting, I will also discuss, our development of speech processing tools for policy body camera footage, which aims to improve police accountability. Both domains involve challenges in working with messy minimally processed data containing sensitive information and domain-specific language. This work emphasizes how NLP has potential to advance social justice goals, like police accountability, but also risks causing direct harm by perpetuating bias and increasing power imbalances.



Structure of this course

MODULE 1

Prerequisite skills for NLP

text normalization, linear alg., prob., machine learning

Approaches

How text is represented

NLP tasks

MODULE 2

statistical machine learning

n-grams

language modeling
text classification

MODULE 3

neural networks

static word vectors

language modeling
text classification

MODULE 4

transformers and LLMs

contextual word vectors

language modeling
text classification
sequence labeling

MODULE 5

NLP applications and ethics

machine translation, chatbots, information retrieval, bias

Review: Describe self-attention in transformers

Lecture overview: Transformers part 2, intro to LLMs

- Transformer input and output details
 - Position embeddings
 - Language modeling head
- Intro to LLMs
 - Pretraining LLMs
 - Sampling for LLM generation
 - Harms from LLMs
- Coding activity: fine-tune GPT-2

- Transformer input and output

Token and Position Embeddings

- The matrix X (of shape $[N \times d]$) has an embedding for each word in the context.
- This embedding is created by adding two distinct embeddings for each input: **token** and **position** embeddings
- Since self-attention doesn't build in order information, we need to encode the order of the sentence in our keys, queries, and values

Token Embeddings

Embedding matrix E has shape $|V| \times d$

- One row for each of the $|V|$ tokens in the vocabulary.
- Each word is a row vector of d dimensions

Given: string *"Thanks for all the"*

1. Tokenize with BPE and convert into vocab indices

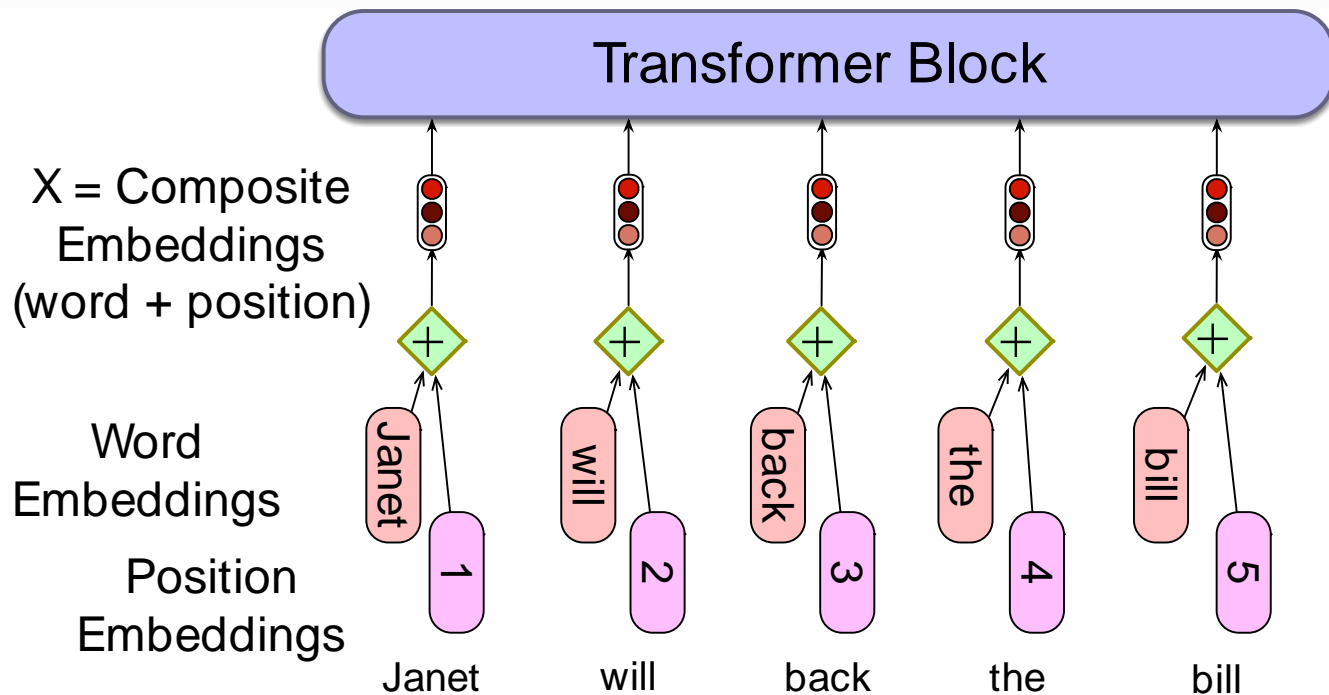
$$w = [5, 4000, 10532, 2224]$$

2. Select the corresponding rows from E , each row an embedding (row 5, row 4000, row 10532, row 2224).

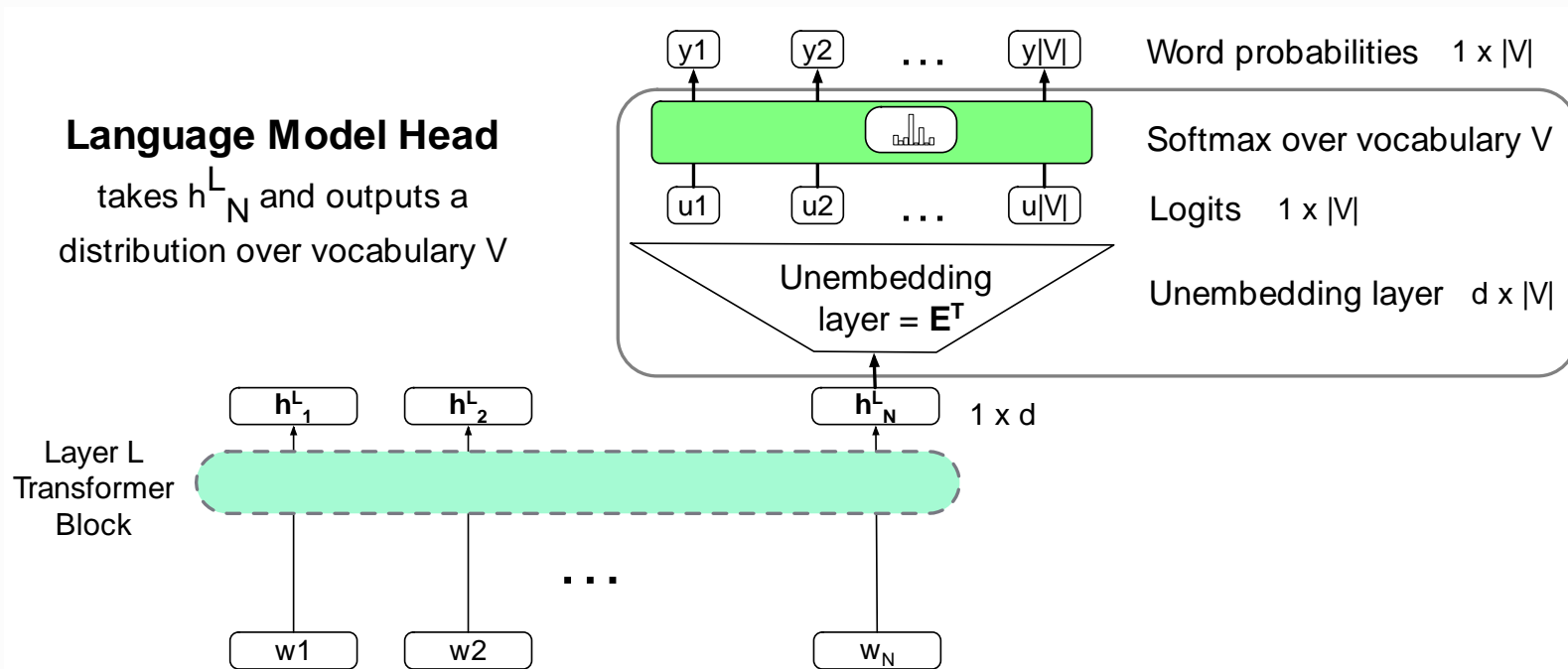
Position Embeddings

- There are many methods, but we'll just describe the simplest: absolute position.
- Goal: learn a position embedding matrix E_{pos} of shape $1 \times N$
- Start with randomly initialized embeddings
 - one for each integer up to some maximum length.
 - i.e., just as we have an embedding for the word *fish*, we'll have an embedding for position 3 and position 17.
- As with word embeddings, these position embeddings are learned along with other parameters during training.

Each x is just the sum of word and position embeddings

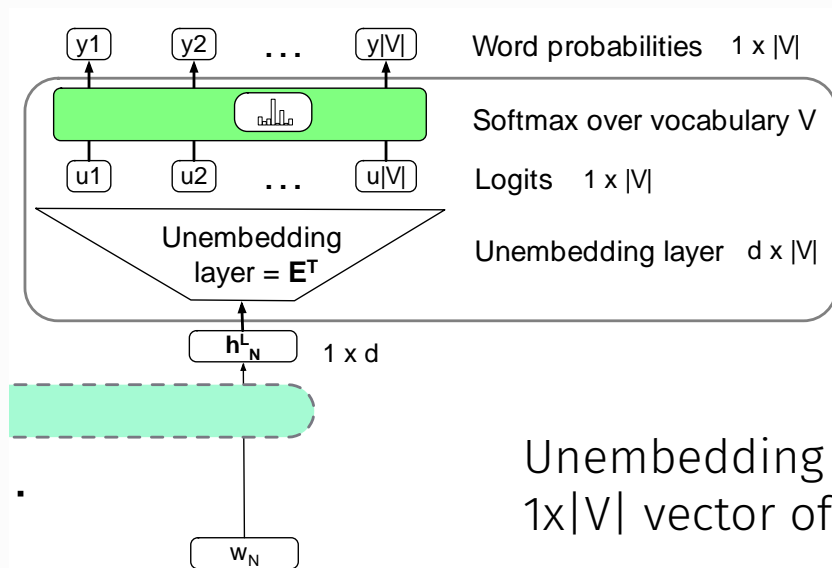


Language modeling head



Language modeling head

Unembedding layer: FFN layer projects from h_N^L (shape $1 \times d$) to probability distribution vector over the vocabulary

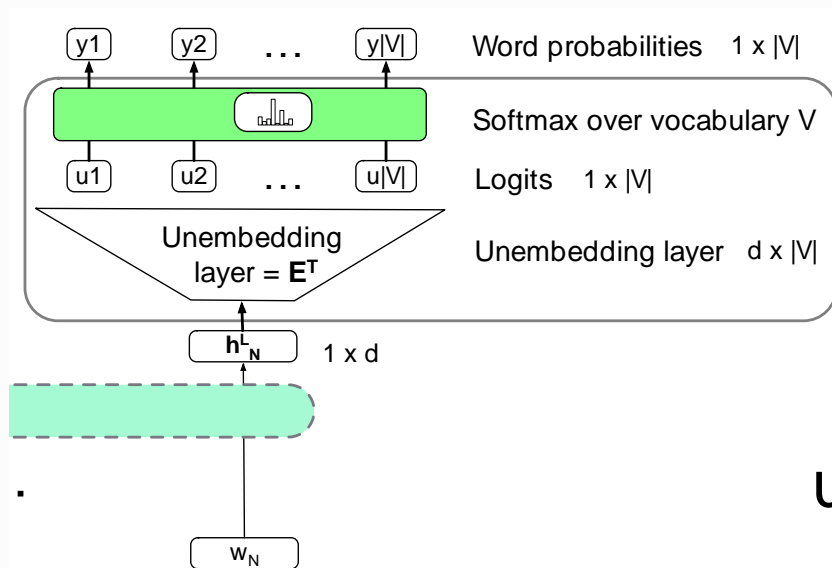


Why "unembedding"? **Tied** to E^T

Weight tying, we use the same weights for two different matrices

Unembedding layer maps from an embedding to a $1 \times |V|$ vector of logits

Language modeling head



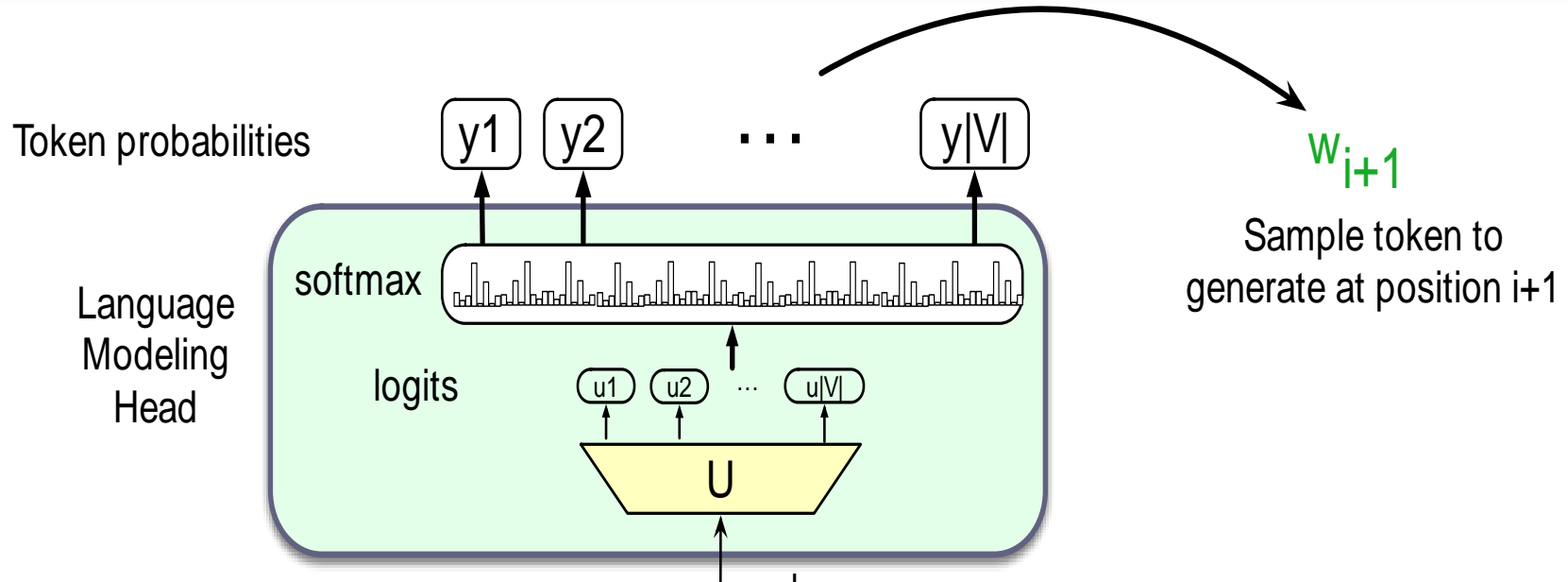
Logits, the score vector u

One score for each of the $|V|$ possible words in the vocabulary V . Shape $1 \times |V|$.

Softmax turns the logits into probabilities over vocabulary. Shape $1 \times |V|$.

$$u = h_N^L E^T$$
$$y = \text{softmax}(u)$$

The final transformer language model



Intro to large language models (LLMs): pretraining and finetuning

Language models

- Remember the simple n-gram language model
 - Assigns probabilities to sequences of words
 - Generate text by sampling possible next words
 - Is trained on counts computed from lots of text
- Large language models are similar and different:
 - Assigns probabilities to sequences of words
 - Generate text by sampling possible next words
 - **Are trained by learning to guess the next word**

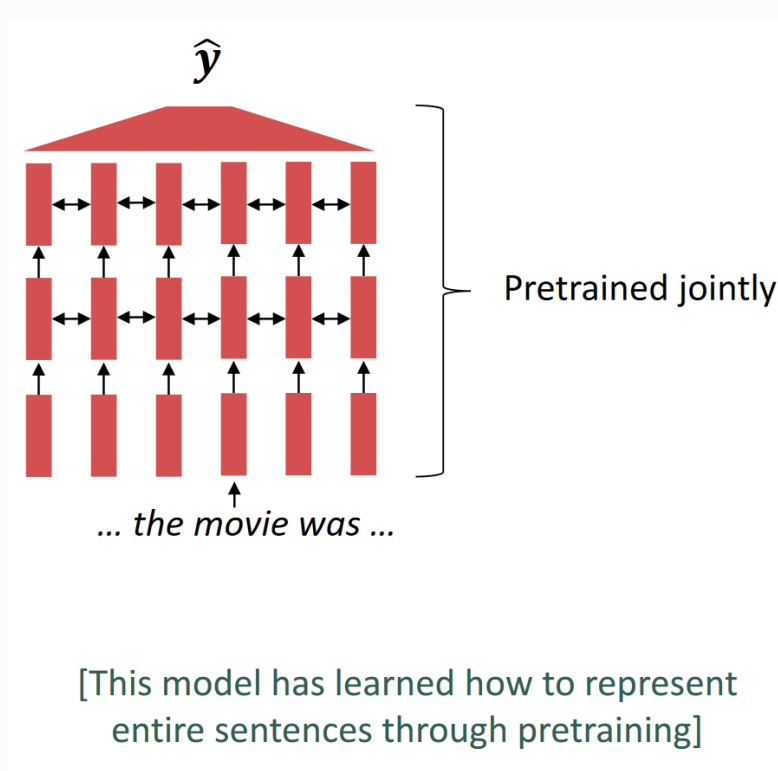
Large language models

- Even though pretrained only to predict words
- Learn a lot of useful language knowledge
- Since training on a **lot** of text

Pretraining whole models

In contemporary NLP:

- All (or almost all) parameters in NLP networks are initialized via **pretraining**.
- Pretraining methods **hide parts of the input** from the model, and train the model to reconstruct those parts.
- This has been exceptionally effective at building strong:
 - representations of language
 - parameter initializations for strong NLP models
 - probability distributions over language that we can sample from



What can we learn from reconstructing the input?

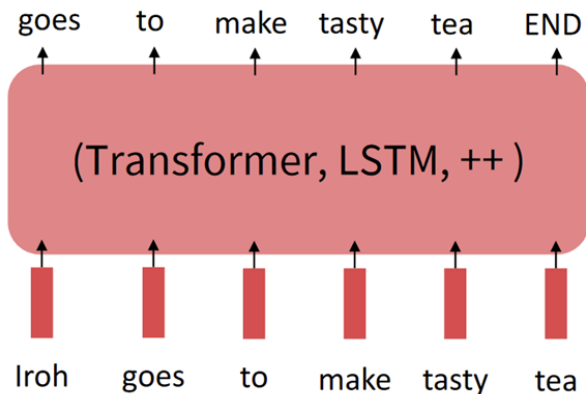
- MIT is located in _____, Massachusetts.
- I put ___ fork down on the table.
- The woman walked across the street, checking for traffic over ___ shoulder.
- I went to the ocean to see the fish, turtles, seals, and _____.
- Overall, the value I got from the two hours watching it was the sum total of the popcorn and the drink. The movie was _____.
- Iroh went into the kitchen to make some tea. Standing next to Iroh, Zuko pondered his destiny. Zuko left the _____.
- I was thinking about the sequence that goes 1, 1, 2, 3, 5, 8, 13, 21, _____

The pretraining + finetuning paradigm

Pretraining can improve NLP applications by serving as parameter initialization.

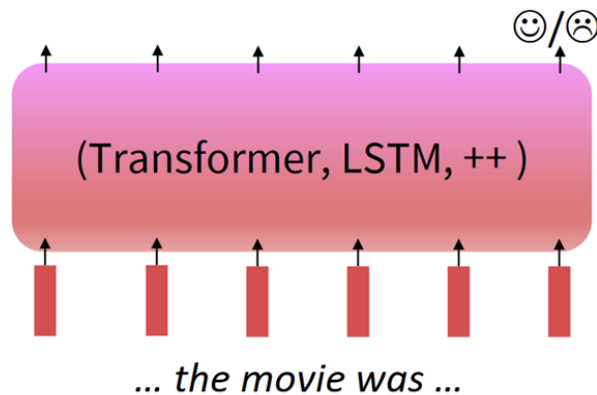
Step 1: Pretrain (on language modeling)

Lots of text; learn general things!



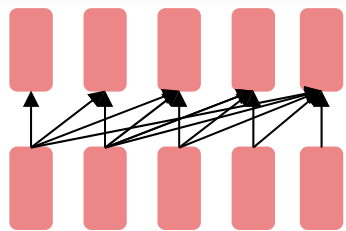
Step 2: Finetune (on your task)

Not many labels; adapt to the task!



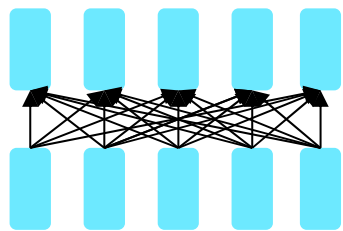
3 types of LLMs:
encoders, encoder-decoders, decoders

Three architectures for large language models



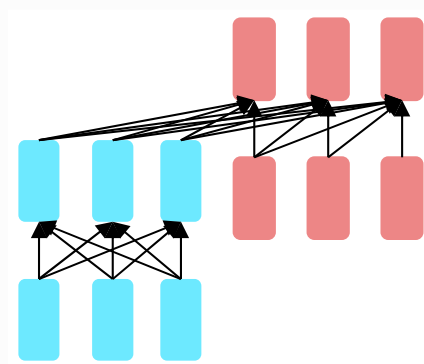
Decoders

GPT, Claude,
Llama, Mixtral



Encoders

BERT family,
HuBERT



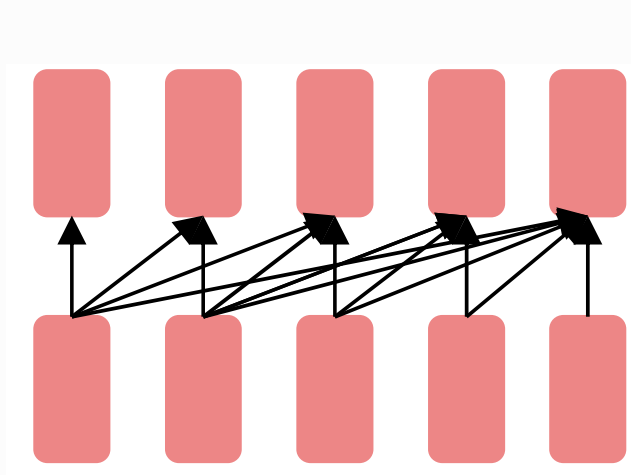
Encoder-decoders

Flan-T5, Whisper

Decoder-only models

Also called:

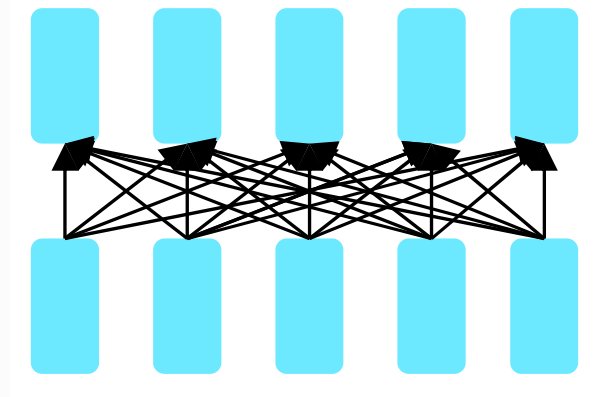
- Causal LLMs
 - Autoregressive LLMs
 - Left-to-right LLMs
-
- Predict words left to right



Encoders

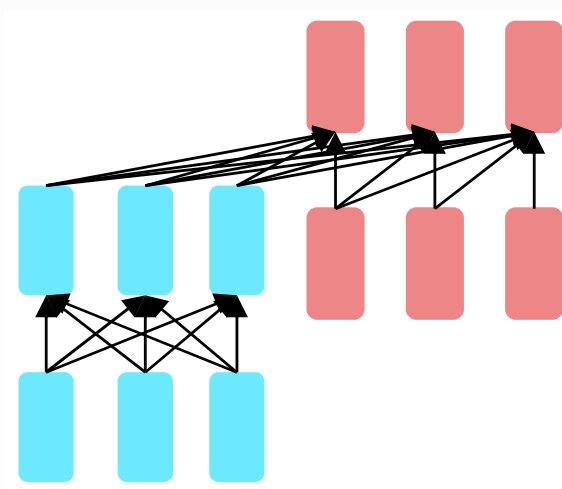
Many varieties!

- Popular: Masked Language Models (MLMs)
- BERT family
- Trained by predicting words from surrounding words on both sides
- Are usually **finetuned** (trained on supervised data) for classification tasks.



Encoder-Decoders

- Trained to map from one sequence to another (sequence to sequence)
- Very popular for:
 - machine translation: map from one language to another
 - speech recognition: map from acoustics to words



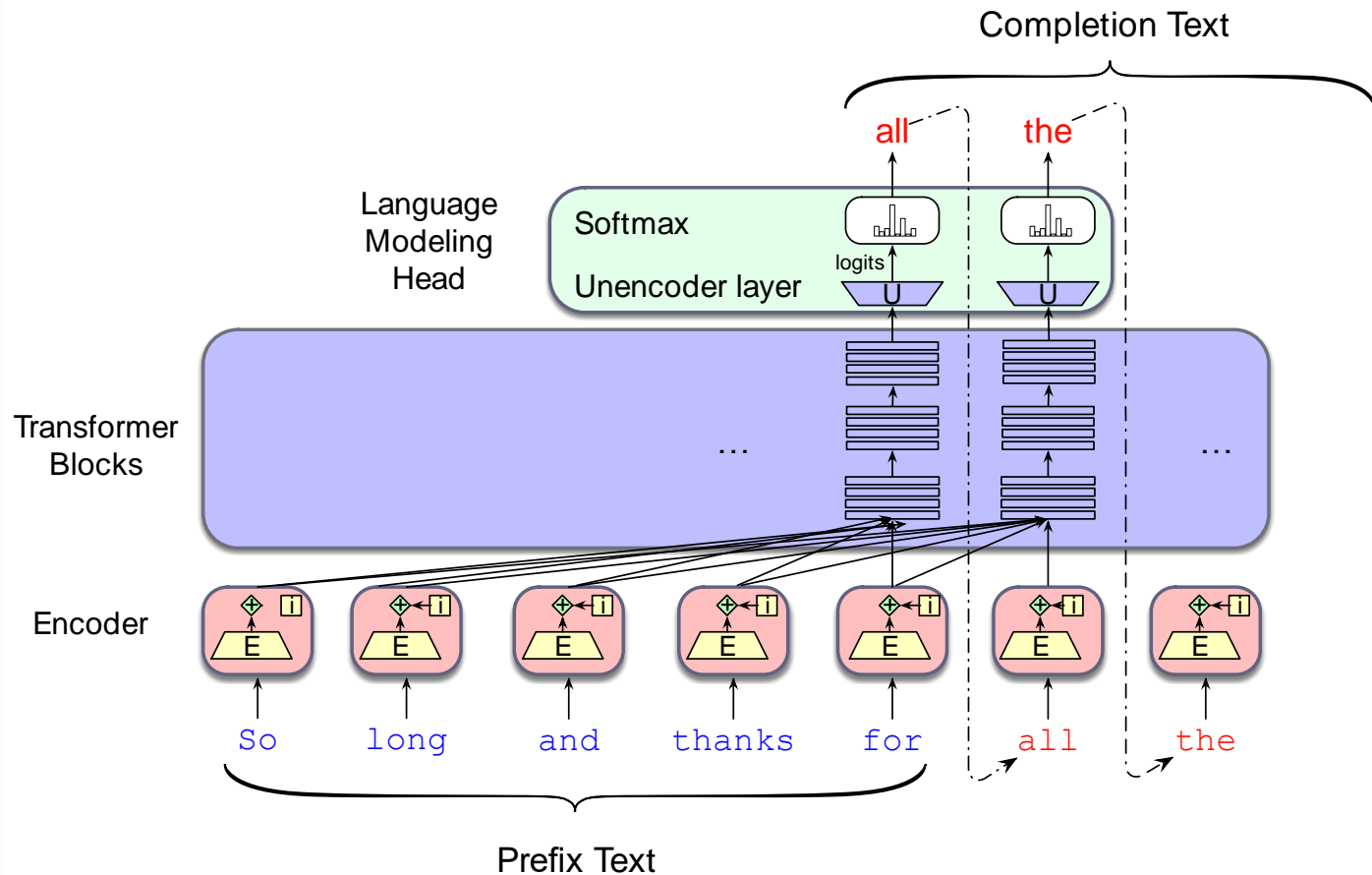
Decoder LLMs

Decoder-only models can handle many tasks

- Many tasks can be turned into tasks of predicting words!

Conditional generation

Generating text conditioned on previous text!



Many practical NLP tasks can be cast as word prediction!

Sentiment analysis: “I like Jackie Chan”

1. We give the language model this string:
The sentiment of the sentence "I like Jackie Chan" is:
2. And see what word it thinks comes next:
 $P(\text{positive} | \text{The sentiment of the sentence ``I like Jackie Chan" is:})$
 $P(\text{negative} | \text{The sentiment of the sentence ``I like Jackie Chan" is:})$

Framing lots of tasks as conditional generation

QA: “Who wrote The Origin of Species”

1. We give the language model this string:

Q: Who wrote the book ``The Origin of Species"? A:

2. And see what word it thinks comes next:

$P(w|Q)$: Who wrote the book ``The Origin of Species"? A:)

3. And iterate:

$P(w|Q)$: Who wrote the book ``The Origin of Species"? A: Charles)

Summarization

The only thing crazier than a guy in snowbound Massachusetts boxing up the powdery white stuff and offering it for sale online? People are actually buying it. For \$89, self-styled entrepreneur Kyle Waring will ship you 6 pounds of Boston-area snow in an insulated Styrofoam box – enough for 10 to 15 snowballs, he says.

Original

But not if you live in New England or surrounding states. “We will not ship snow to any states in the northeast!” says Waring’s website, ShipSnowYo.com. “We’re in the business of expunging snow!”

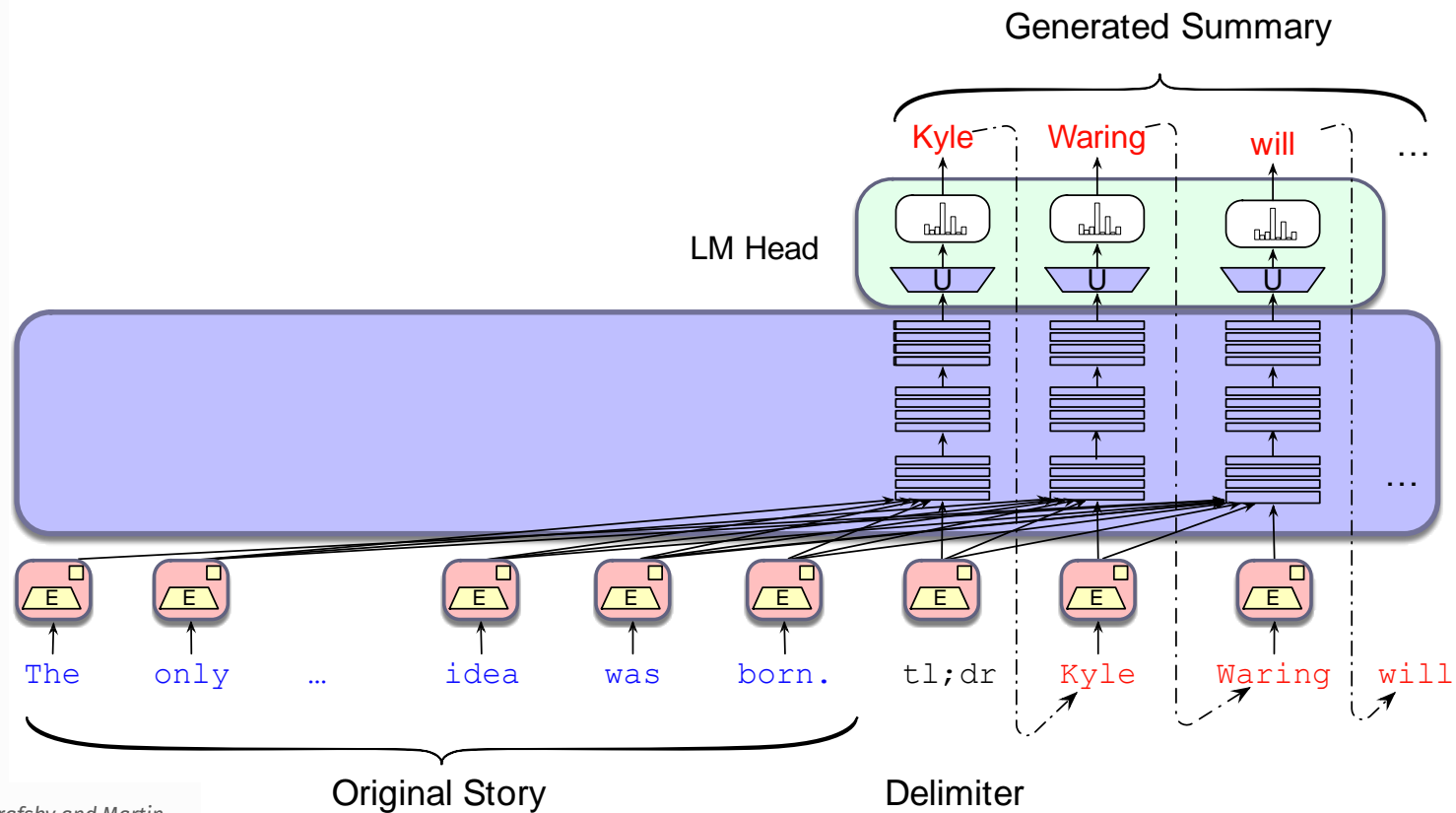
His website and social media accounts claim to have filled more than 133 orders for snow – more than 30 on Tuesday alone, his busiest day yet. With more than 45 total inches, Boston has set a record this winter for the snowiest month in its history. Most residents see the huge piles of snow choking their yards and sidewalks as a nuisance, but Waring saw an opportunity.

According to Boston.com, it all started a few weeks ago, when Waring and his wife were shoveling deep snow from their yard in Manchester-by-the-Sea, a coastal suburb north of Boston. He joked about shipping the stuff to friends and family in warmer states, and an idea was born. [...]

Summary

Kyle Waring will ship you 6 pounds of Boston-area snow in an insulated Styrofoam box – enough for 10 to 15 snowballs, he says. But not if you live in New England or surrounding states.

LLMs for summarization (using tldr)

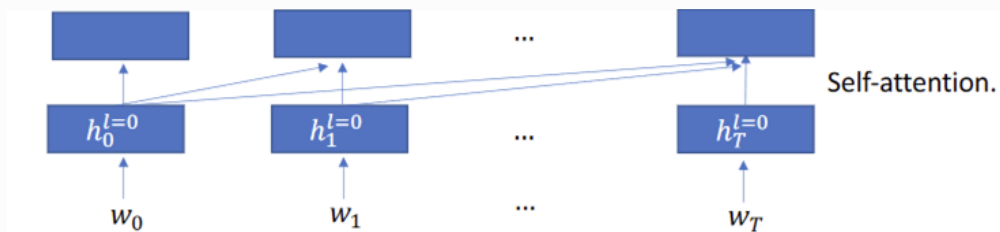


Pretraining decoder LLMs

- Take a corpus and ask the model to predict the next word!
- Train the model using gradient descent to minimize the error
- Same loss function as other neural models: cross-entropy loss
- Move the weights in the direction that assigns a higher probability to the true next word

Decoding: apply a “causal mask” for self-attention

- To do auto-regressive LM, we need to apply a “causal” mask to self-attention, forbidding it from getting future context.
- At timestep t , we set $a_i = 0$ for $i > t$



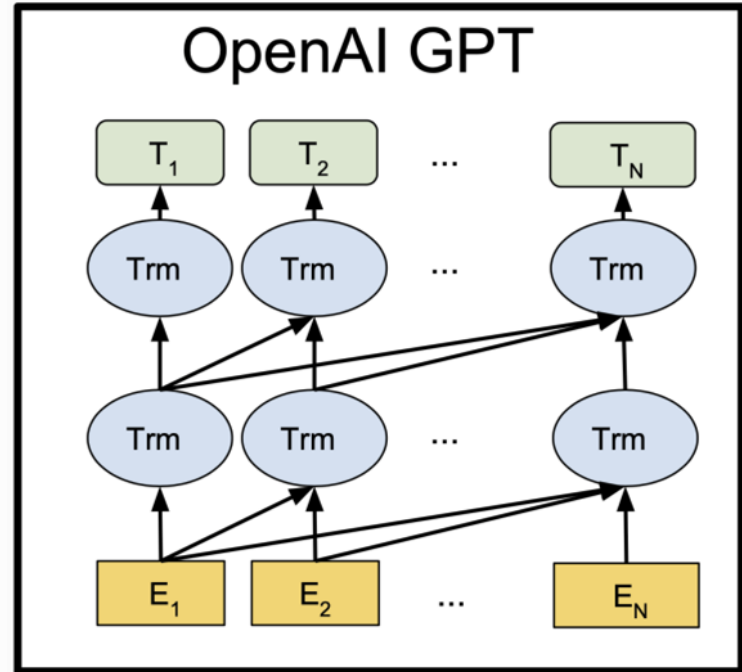
For encoding these words

We can look at these (not greyed out) words

	[START]	The	chef	who
[START]		$-\infty$	$-\infty$	$-\infty$
The			$-\infty$	$-\infty$
chef				$-\infty$
who				

Generative Pretrained Transformer (GPT; Radford et al. 2018)

- 2018's GPT was a big success in pretraining a decoder!
- Transformer decoder with 12 layers, 117M parameters.
- 768-dimensional hidden states, 3072-dimensional feed-forward hidden layers.
- Trained on BooksCorpus: over 7000 unique books.
 - Contains long spans of contiguous text, for learning long-distance dependencies.



GPT-2, GPT-3, GPT-4 from OpenAI

- They are basically larger and larger autoregressive transformer LMs trained on larger and larger amounts of data
- They have shown amazing language generation capability when you give it a prompt (aka. prefix, the beginning of a paragraph)



Generation example from the GPT-2 model

SYSTEM PROMPT
(HUMAN-WRITTEN)

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

MODEL COMPLETION
(MACHINE-WRITTEN,
10 TRIES)

The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

A sample from GPT2 (with top-k sampling)

Sampling for LLM generation

Decoding and Sampling

- This task of choosing a word to generate based on the model's probabilities is called **decoding**.
- The most common method for decoding in LLMs: **sampling**.
- Sampling from a model's distribution over words:
 - choose random words according to their probability assigned by the model.
- After each token we'll sample words to generate according to their probability *conditioned on our previous choices*,
 - A transformer language model will give the probability

Random sampling

```
i ← 1  
wi ∼ p(w)  
while wi ≠ EOS  
    i ← i + 1  
    wi ∼ p(wi | w<i)
```

Random sampling doesn't work very well

- Even though random sampling mostly generate sensible, high-probable words,
- There are many odd, low- probability words in the tail of the distribution
- Each one is low- probability but added up they constitute a large portion of the distribution
- So they get picked enough to generate weird sentences

Factors in word sampling: **quality** and **diversity**

Emphasize **high-probability** words

- + **quality**: more accurate, coherent, and factual,
- **diversity**: boring, repetitive.

Emphasize **middle-probability** words

- + **diversity**: more creative, diverse,
- **quality**: less factual, incoherent

Top-k sampling:

1. Choose # of words k
2. For each word in the vocabulary V , use the language model to compute the likelihood of this word given the context $p(w_t | w_{<t})$
3. Sort the words by likelihood, keep only the top k most probable words.
4. Renormalize the scores of the k words to be a legitimate probability distribution.
5. Randomly sample a word from within these remaining k most-probable words according to its probability.

Temperature sampling

Reshape the distribution instead of truncating it

Intuition from thermodynamics,

- a system at high temperature is flexible and can explore many possible states,
- a system at lower temperature is likely to explore a subset of lower energy (better) states.

In **low-temperature sampling**, ($\tau \leq 1$) we smoothly

- increase the probability of the most probable words
- decrease the probability of the rare words.

Temperature sampling

Divide the output by a temperature parameter τ before passing it through the softmax.

Instead of

$$\mathbf{y} = \text{softmax}(u)$$

We do

$$\mathbf{y} = \text{softmax}(u/\tau)$$

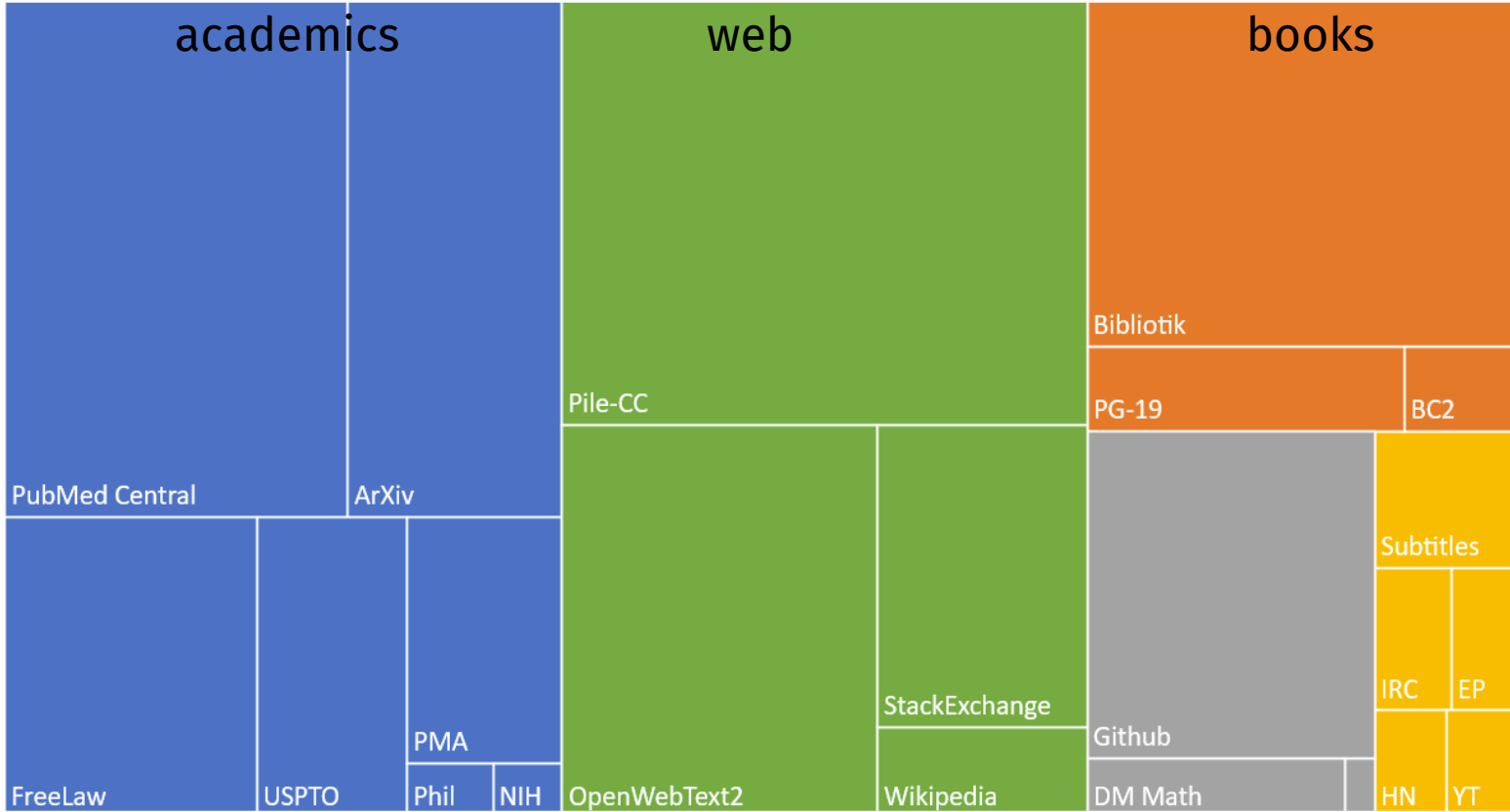
A lower τ pushes high-probability words higher and low probability word lower due to the way softmax works

● Pretraining data and harms of LLMs

LLMs are mainly trained on the web

- Common crawl, snapshots of the entire web produced by the non-profit Common Crawl with billions of pages
- Colossal Clean Crawled Corpus (C4; [Raffel et al. 2020](#)), 156 billion tokens of English, filtered
- What's in it? Mostly patent text documents, Wikipedia, and news sites

The Pile: a pretraining corpus



dialog

Slide adapted from Jurafsky and Martin

Big idea

- Text contains enormous amounts of knowledge
- Pretraining on lots of text with all that knowledge is what gives language models their ability to do so much

But there are problems with scraping from the web

- **Copyright:** much of the text in these datasets is copyrighted
 - Not clear if fair use doctrine in US allows for this use
 - This remains an open legal question
- **Data consent**
 - Website owners can indicate they don't want their site crawled
- **Privacy:**
 - Websites can contain private IP addresses and phone numbers

Harms from LLMs

What Can You Do When A.I. Lies About You?

People have little protection or recourse when the technology creates and spreads falsehoods about them.

Hallucination

Air Canada loses court case after its chatbot hallucinated fake policies to a customer

The airline argued that the chatbot itself was liable. The court disagreed.

Copyright

Authors Sue OpenAI Claiming Mass Copyright Infringement of Hundreds of Thousands of Novels

Privacy

How Strangers Got My Email Address From ChatGPT's Model

Harms from LLMs

Toxicity and abuse

The New AI-Powered Bing Is Threatening Users.

Cleaning Up ChatGPT Takes Heavy Toll on Human Workers

Contractors in Kenya say they were traumatized by effort to screen out descriptions of violence and sexual abuse during run-up to OpenAI's hit chatbot

Misinformation

Chatbots are generating false and misleading information about U.S. elections

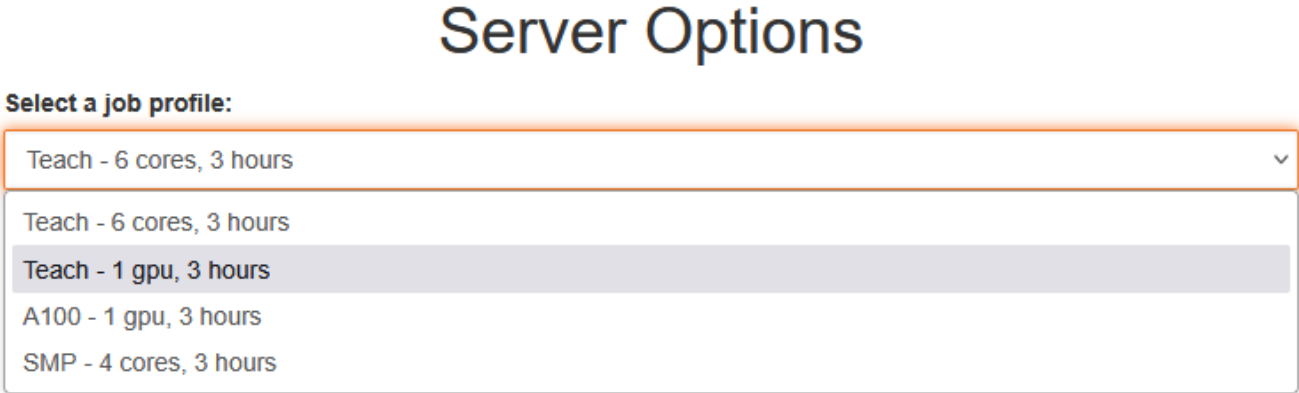
Conclusion

- Transformer-based language models pretrained on lots of text are called **large language models (LLMs)**
- LLMs can have decoder-only, encoder-only, or encoder-decoder architectures
- Decoder-only LLMs can cast many different NLP tasks as word prediction
- There are many different sampling approaches that balance diversity and quality in text generation from LLMs
- Harms from LLMs include hallucinating false information, leaking private information from training data, generating abuse and misinformation

Coding activity

Notebook: finetune GPT-2 on Shakespeare

- [Click on this nbgitpuller link](#) or find the link on the course website
- **Important difference from normal:** Open a 'Teach – 1 gpu, 3 hours' server



The image shows a screenshot of a web interface titled "Server Options". Below the title, there is a label "Select a job profile:" followed by a dropdown menu. The dropdown menu is open, showing a list of server profiles. The profile "Teach - 1 gpu, 3 hours" is highlighted in grey, indicating it is the selected option. Other visible options include "Teach - 6 cores, 3 hours", "A100 - 1 gpu, 3 hours", and "SMP - 4 cores, 3 hours".

Job Profile
Teach - 6 cores, 3 hours
Teach - 1 gpu, 3 hours
A100 - 1 gpu, 3 hours
SMP - 4 cores, 3 hours

- Open `session17_gpt2_shakespeare.ipynb`