A Taxonomy of Large Language Models

CS 6120 Natural Language Processing

Northeastern University

Si Wu

Some slides borrowed from Jurafsky & Martin Chapter 7

Logistics

- The project initial pitch is due tonight.
  - One slide from last lecture elaborates on what an idea should look like
  - Make sure it's sensible. Also check data first. If you don't have data, you most likely can't have a project done in 2 months.
- Flu shot!
  - Many people are getting sick and can't attend lecture.
  - Flu shot and COVID vaccines are free at CVS (but double check your health insurance).
- Today:
  - Continue about positional embedding in transformer
  - What's LLM

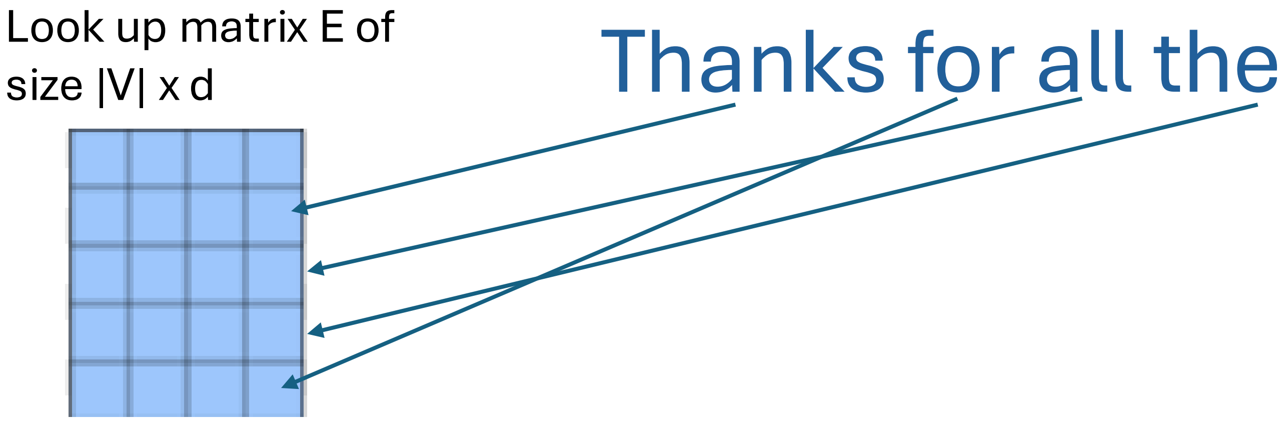
Continue from last lecture

Input embeddings in transformer models

- It has two components:
  - Input token embedding
  - Input positional embedding
- This is the *initial* embedding. As the initial embedding passing through the transformer layers, it will change.
- The initial embeddings are stored in an *embedding matrix E*
  - E has |V| rows, each row is a token in the vocabulary.
  - Each row is of d dimension, so E is of size V x d

Input embedding

- For a sentence like



Look up matrix E of size |V| x d

Thanks for all the

We look up each token’s row index in matrix E.

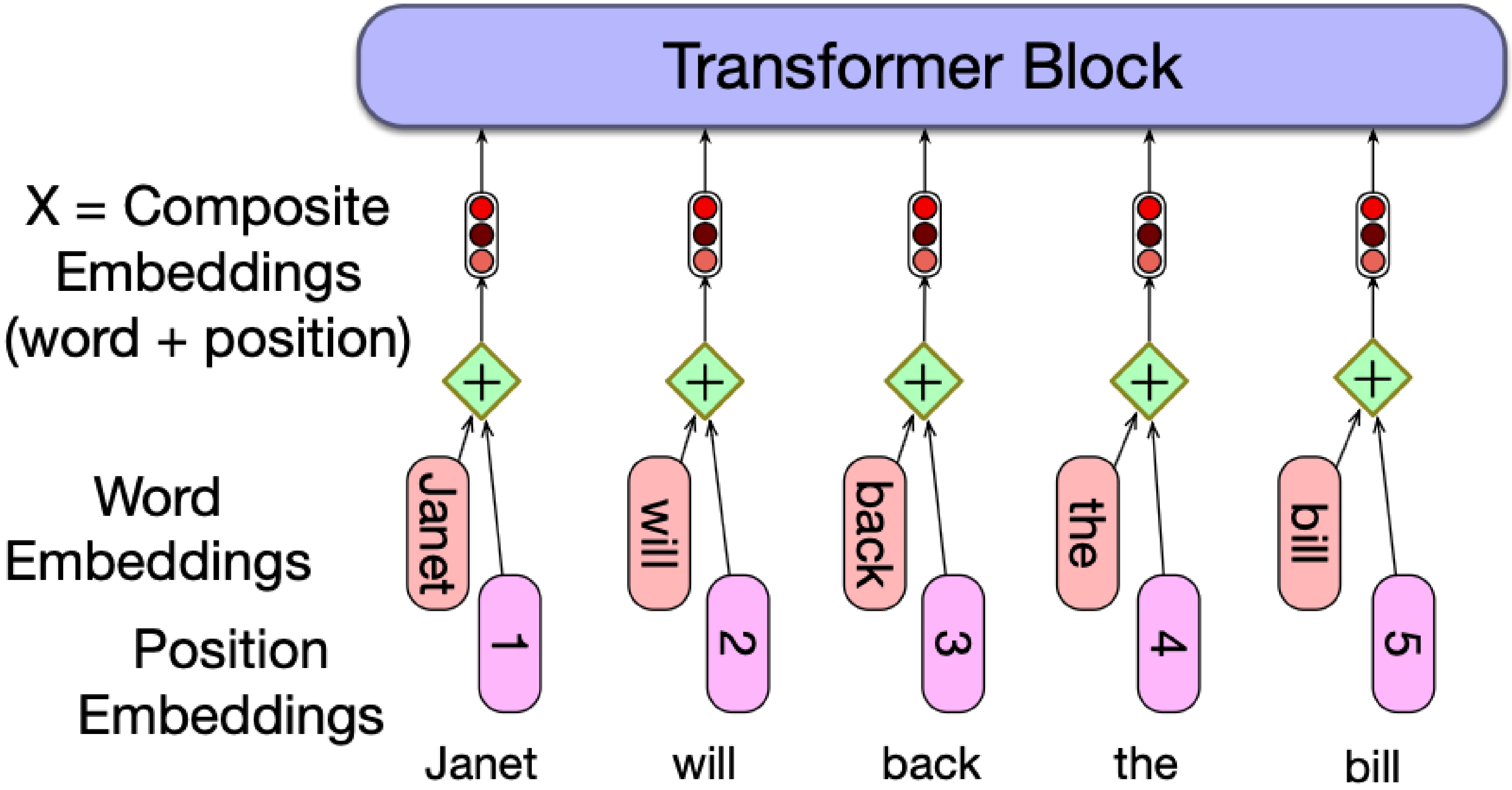
- For example, [row 5, row 2, row 1000, row 431] → [5,2,1000,431]

Positional embedding

- Why does it matter?
  - Word order matters!

    “the dog bit the man” $\neq$“the man bit the dog”

- Simplest method: absolute position
  - Just like we have an initial embedding for the word “fish”, we will have an embedding for word at position 3.
  - Final embedding for a word at position i is $E[w_i] + P[i]$, here P is the matrix for positional embeddings. Both $E[w_i]$ and $P[i]$ are of size d.


Transformer Block
X = Composite Embeddings (word + position)
Word Embeddings
Position Embeddings
Janet
will
back
the
bill
1
2
3
4
5
Janet
will
back
the
bill

Language modeling head

3 components of a transformer:

~~Transformer block~~

~~Input embedding~~

Language modeling head

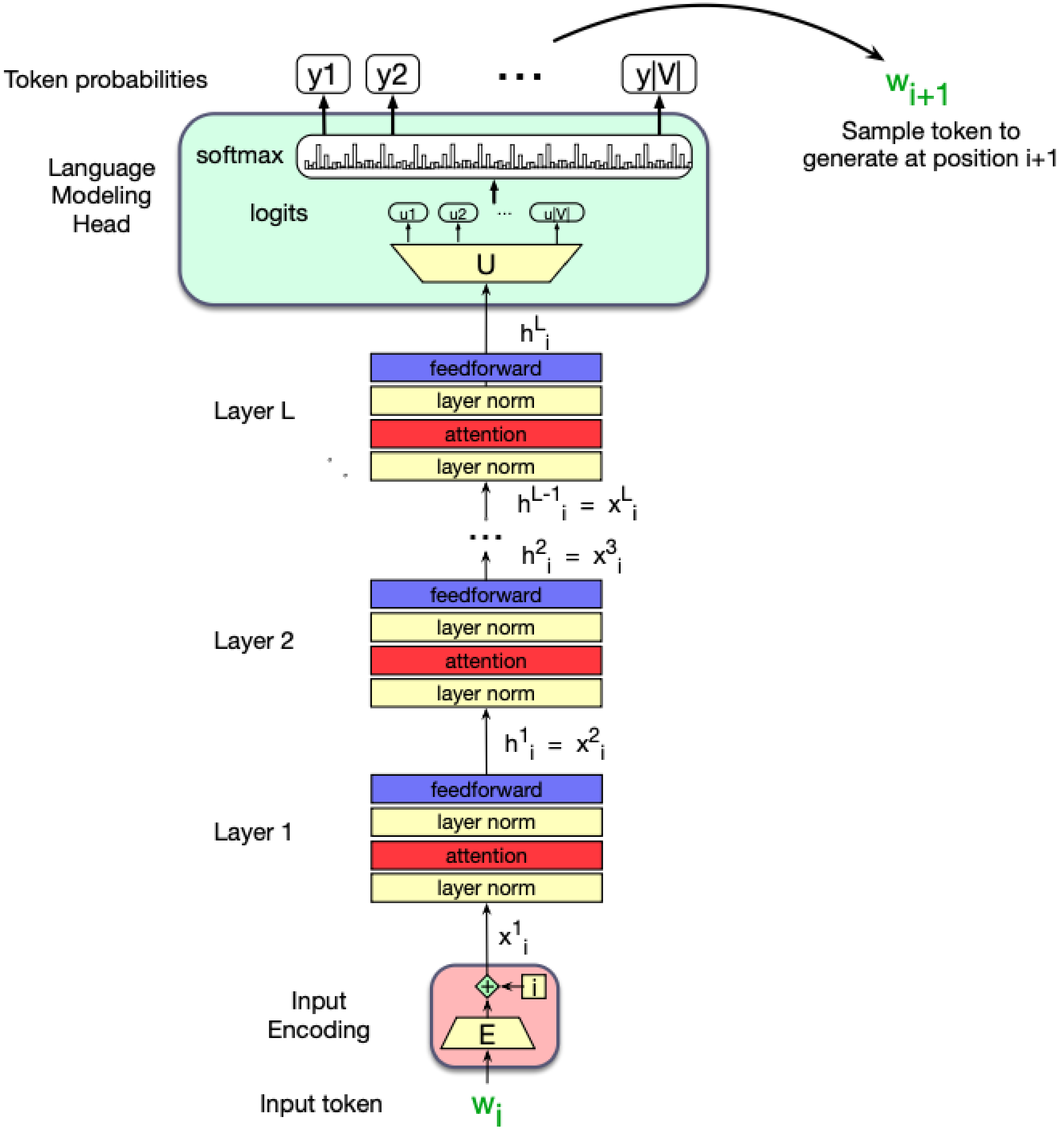
What is language modeling head

- “Head”: additional neural network layer we add on top of the basic transformer architecture when we apply pretrained transformer to various task

- Goal of the language modeling head: to take the output of the final transformer layer from the last token and use it to predict the next word

The process

- A linear layer maps each embedding back to the size of the vocabulary.
- This produces a logit distribution over all possible tokens
  - Logit: raw, unnormalized score before softmax
- Softmax applied to convert logits into probabilities
- Sampling from these probabilities
  - E.g. greedy decoding → use the highest probability
  - Many others, top-p sampling, nucleus samping, etc.


Token probabilities
y1
y2
...
y|V|
w_{i+1}
Sample token to generate at position i+1
Language Modeling Head
softmax
logits
u1
u2
...
u|V|
U
h^L_i
Layer L
feedforward
layer norm
attention
layer norm
$h^{L-1}_i = x^L_i$
...
$h^2_i = x^3_i$
Layer 2
feedforward
layer norm
attention
layer norm
$h^1_i = x^2_i$
Layer 1
feedforward
layer norm
attention
layer norm
x^1_i
Input Encoding
i
E
Input token
w_i

Additional comment

- This kind of unidirectional causal language model is called a **decoder-only model**
  - Because this model is roughly half of the encoder-decoder model

Introduction of Large Language Models (LLMs)

Transformer, but make it larrrrge

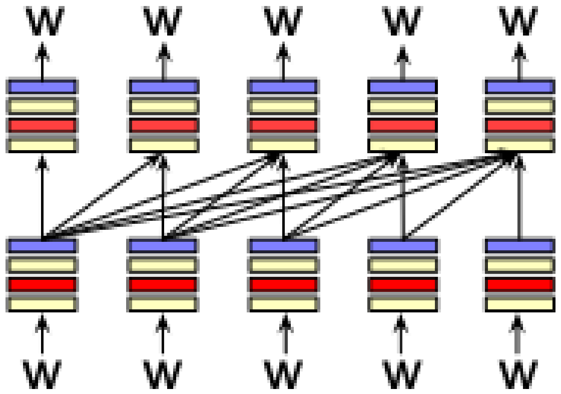
What are large language models

- Most of the LLMs are transformers. Different variations.
- They are LARGE.
  - For example, more stacked transformer blocks
- They are usually trained on enormous amount of knowledge
- Pretraining on lot of text with all that knowledge is what gives LLM their ability to do so much
  - We will talk about pretraining next week
- Recall from the transformer lecture, we learn that transformer were designed to be parallelizable, much better than RNNs

3 different architectures for LLMs

- Encoder
- Decoder
- Encoder-decoder

Three architectures for large language models



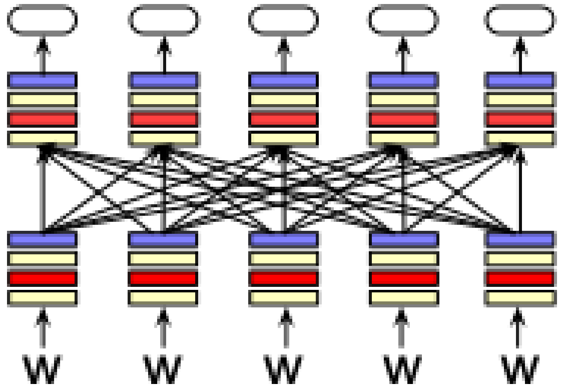
Decoders

GPT, Claude,

LLaMA

Mistral

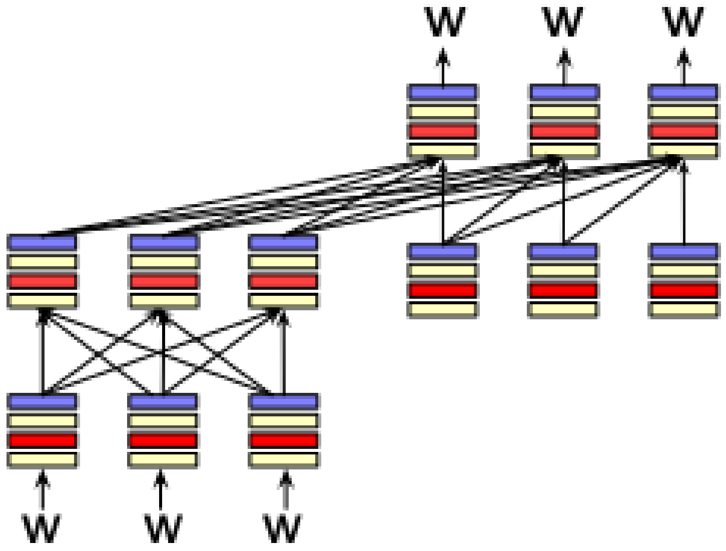
Gemma



Encoders

BERT family,

RoBERTa



Encoder-decoders

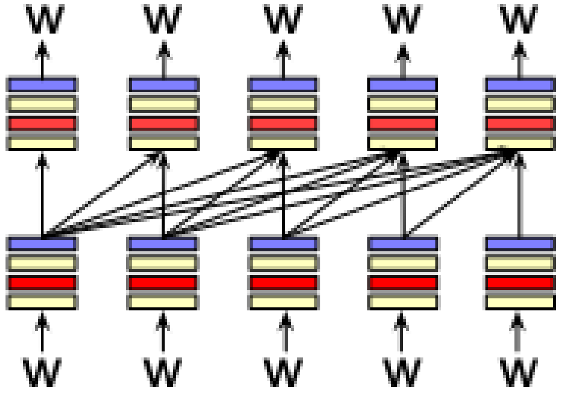
T5, BART

OG transformer

marianMT

Whisper (speech)

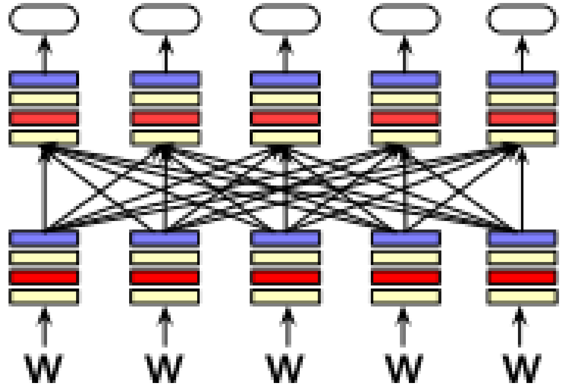
Decoders



What most people think of when we say LLM

- GPT, Claude, Llama, DeepSeek, Mistral
- A generative model
- It takes as input a series of tokens, and iteratively generates an output token one at a time.
- Left to right (causal, autoregressive)

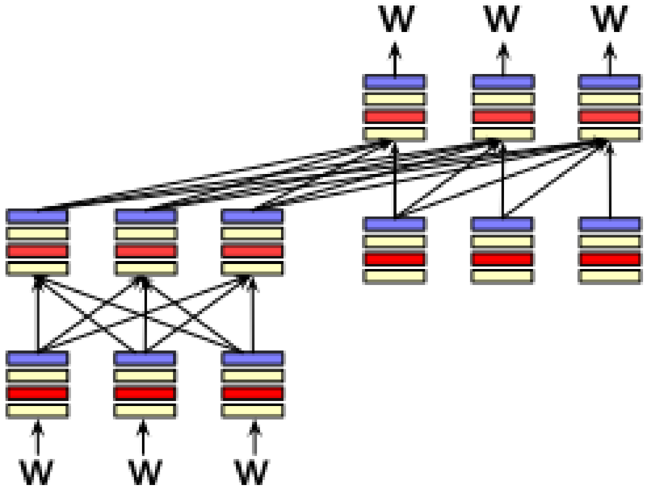
Encoders



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

Conditional Generation: Generating text conditioned on previous text!

1. Give the LLM an input piece of text, a **prompt**
2. Have it generate token by token
   - conditioned on the prompt and the generated tokens

- We generate from a model by

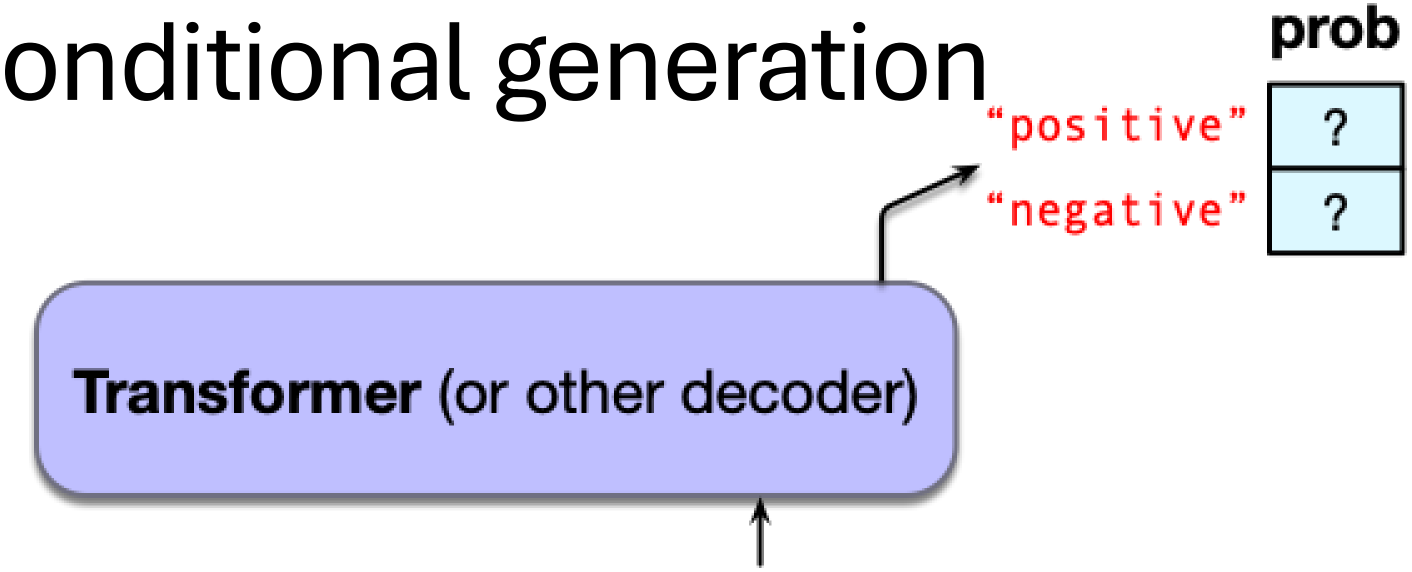
1. computing the probability of the next token w_i from the prior context: $P(w_i|w_{<i})$
2. sampling from that distribution to generate a token

Many practical NLP tasks can be cast as conditional generation!

- 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

Sentiment via conditional generation



The sentiment of the sentence "I like Jackie Chan" is:

Which word has a higher probability?

$$P(\text{positive}|\texttt{The sentence ``I like Jackie Chan" is:})$$

$$P(\text{negative}|\texttt{The sentence ``I like Jackie Chan" is:})$$

Framing lots of tasks as conditional generation

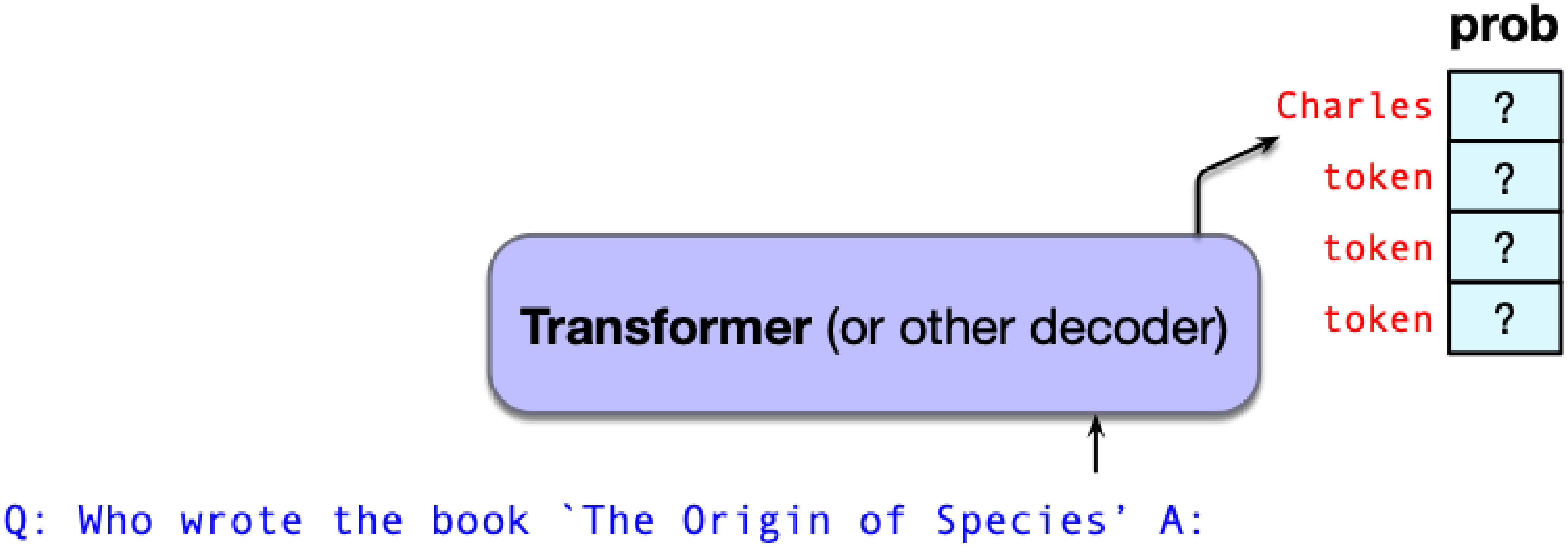
- QA: “Who wrote The Origin of Species”

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:

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



prob

Charles	?
token	?
token	?
token	?

Transformer (or other decoder)

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

Now we iterate:

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

Ethical and safety issues in LLMs

Hallucination

Chatbots May 'Hallucinate' More Often Than Many Realize

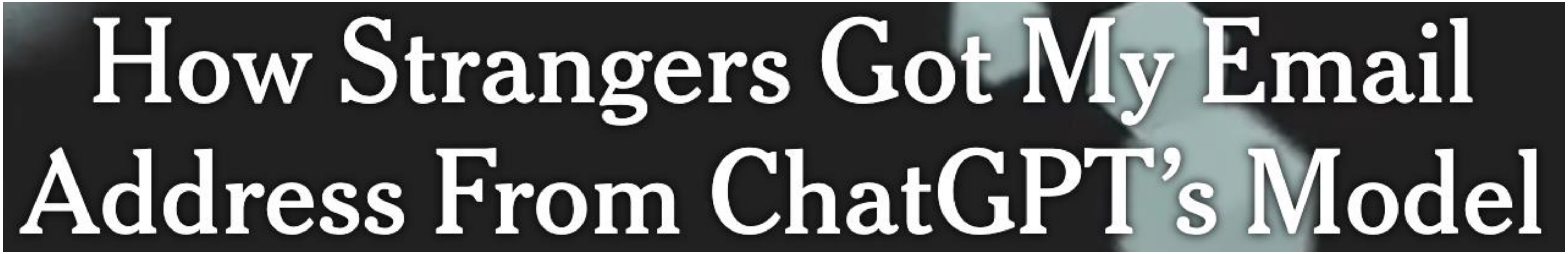
What Can You Do When A.I. Lies About You?

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

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

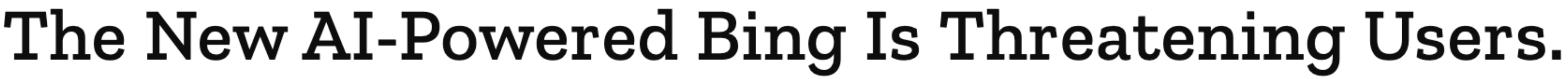
The airline argued that the chatbot itself was liable. The court disagreed.

Privacy

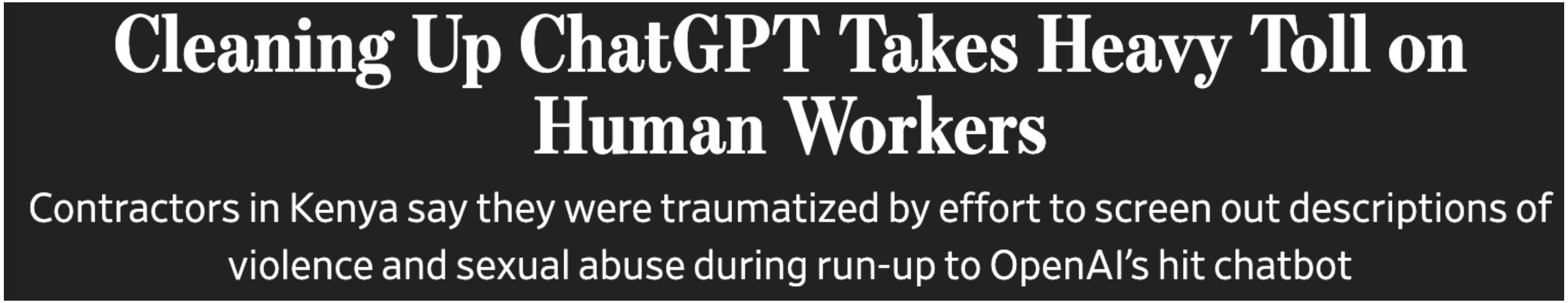


How Strangers Got My Email Address From ChatGPT's Model

Abuse and Toxicity



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

Lots more problematic qualities of LLMs

- Harm (suggesting dangerous actions)
- Fraud (can help generate convincing phishing email, writing fake article, etc.)
- Emotional dependence
- Bias

NLP researchers are actively researching these areas! It’s something you can work on for your class project too

Market of LLMs

Market of LLMs

- Most of the OpenAI GPT models are closed, proprietary. Their internal architecture and weights are not public
    - You can access via API and so on
- Anthropic (Claude) is similar
- Google has both open and close-source models. Close-source models perform better of course.
- Deepseek has both open and close-source models. Close-source models are generally cheaper
- xAi
- Etc.

GPT-3, openAI 2020

- 175 billion parameters
- Decoder only transformer
- 96 transformer blocks (layers)
- Context window 2,048 tokens
- Training data: about 300 billion tokens from Common crawl

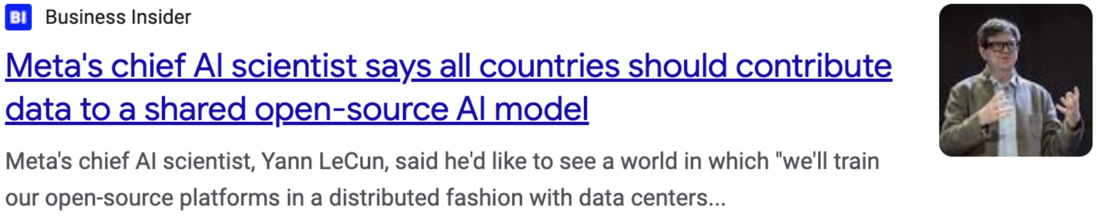
GPT-2, openAI, 2019

- 1.5 billion parameters
- Decoder only transformer
- 48 transformer blocks (layers)
- Context window 1,024 tokens
- Training data: text from the internet

LLaMA 2, Meta, 2023

- 7, 13, 70 billion parameters
- Decoder only transformer
- 32, 40, 80 transformer blocks (layers)
- Context window 4,096 tokens
- Training data: 2 trillion data from the internet

Open-source models



Business Insider

Meta's chief AI scientist says all countries should contribute data to a shared open-source AI model

Meta's chief AI scientist, Yann LeCun, said he'd like to see a world in which "we'll train our open-source platforms in a distributed fashion with data centers...

- Deepseek v3
  - 671B parameter
  - Mixture-of-experts
  - 61 transformer blocks
  - About 14.8 trillion tokens, mostly English and Chinese
  - Multi-head latent attention
- Olmo 7B
  - 7B
    32 transformer blocks
  - 2 trillion tokens from Dolma dataset
  - Fully open: training data, code, eval framework, etc.
- LLaMA 3 70B
  - 70B param
  - 80 transformer blocks
  - About 15 trillion tokens
  - Primarily English but has multilingual data

But open weight ≠ open source
how many of these are open
source where we know everything
about this model from training data
to architecture?

How much does it cost to train, GPT3 for example

- Money: it costs 4.6 million US dollars
- Time: not public. Estimated on a single v100 GPU would take 355 *years*
- Energy: we don’t know, but probably a lot consider how long it takes
- Number of engineers (approx): we also don’t know

Limits of LLM

- English-centric,
- Cultural bias
- Hallucination
- Sycophancy

To conclude

- They are still statistical models, just trained on a lot a lot of knowledge, and we are using a lot a lot of energy to train them, like ever before
- They are great models, but we can still improve them.
    - Efficiency – quadratic complexity
    - Context window limit
- Difficult things like hallucination, bias, and harm, how to get rid of them? And what about privacy (e.g. personal email address)?

- Anyway, we will talk about pretraining next week after our first guest lecturer Alexander!