CMSC 373 Artificial Intelligence Fall 2025 20-LLMs

Deepak Kumar
Bryn Mawr College

1

NNs for NLP Architectures

- **Representing words** and **word order** is important in NNs for NLP tasks.
- Representing Words as Vectors: *One-hot encoding*, *Word2Vec*, *Word Embedding*
- Inputing a word at a time ignores word ordering.
- RNNs enable word sequence modeling but only go *so far.*
- **Large Language Models (Transformers)** track word order information and pay attention to different parts of a sentence without the use of RNNs.

2

2

Language Modeling

- The problem:

  Given a sequence of words $w_1, w_2, \ldots, w_{i-1}$

  Predict w_i

  where $w_1 \ldots w_n \in$ {<vocabulary of words>}
  i.e.

  $$P(w_i \mid w_1, w_2, \ldots, w_{i-1})$$

- Example,

  Input: `the cat sat on the`
  Output: `the cat sat on the` **`mat`**
  (97%)

- Any system that that can do this prediction is called a **language model**.
- **A language model is a probabilistic model of language.**

3

Language Model: Word N-Gram Models

- $P(w_i \mid w_1, w_2, \ldots, w_{i-1})$ depends on i-1 previous words

  This is called an i-gram model.

  Unigram is word frequencies of every word
  Bigram is word pair frequencies
  Trigram is 3-word frequencies

  Given: `He likes`
  Output: `He likes` **`being`**

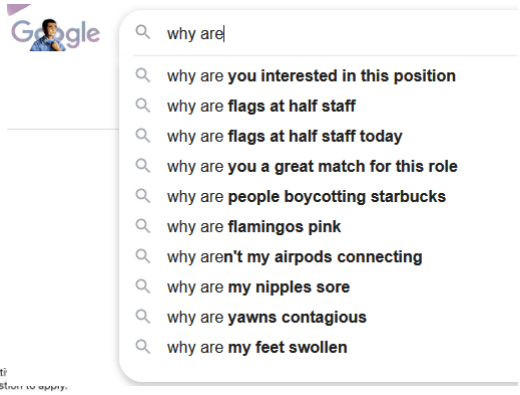
- Large Ngram LLMs used for Machine translation (2005)

He likes attention	196
He likes bananas	51
He likes barbeque	44
He likes baseball	188
He likes basketball	57
He likes beer	281
He likes being	**2026**
He likes best	165
He likes better	55
He likes big	380
He likes bikes	47
He likes birds	111
He likes blue	42
He likes books	191
He likes both	276
He likes boys	90
He likes bread	73

4

Applications

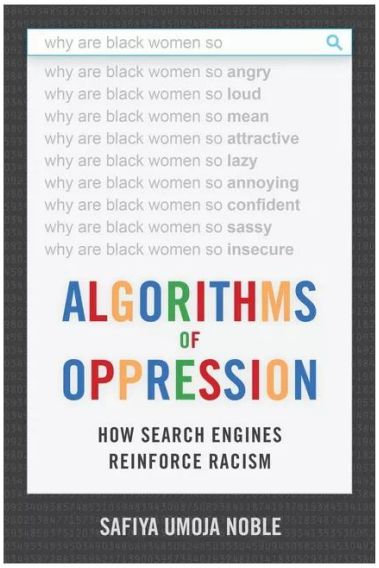
- Google Search
- Next word prediction in smart phone texts
- Writing assistants
- Etc.



5

5

Issues



6

6

Large Language Models

- Computational agents that can interact conversationally with people using natural language. Since 2017, LLMS have revolutionized the field of NLP and AI.
- Fluent speakers of a language bring a large amount of knowledge to bear during comprehension and production.
- Much of the knowledge is embodied in the vocabulary, the representations of words, meanings, and their usage.
- Makes vocabulary a useful lens to explore acquisition of knowledge from text by people and by machines.

7

7

Language Models

- Ngram Models
  - Assign probabilities to sequences of words
  - Generate text by sampling possible next words
  - **Trained on counts computed from lots of text**
- Large language Models are similar and different:
  - Assign probabilities to sequences of words
  - Generate text by sampling possible next words
  - **Are trained by learning to guess the next word**

8

8

Large Language Models

- Introduction
- Conditional Generation
- Prompting
- Sampling
- Pretraining
- Finetuning
- Evaluation
- Ethical and Safety Issues

9

9

LLMs – Fundamental Intuition

- 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

10

10

What does a model learn from pretraining?

- With roses, dahlias, and peonies, I was surrounded by _____
- The room wasn't just big it was _____
- The square root of 4 is _
- The author of "A Room of One's Own" is Virginia _____
- The doctor told me that __
- So long and thanks for ___

11

11

What does a model learn from pretraining?

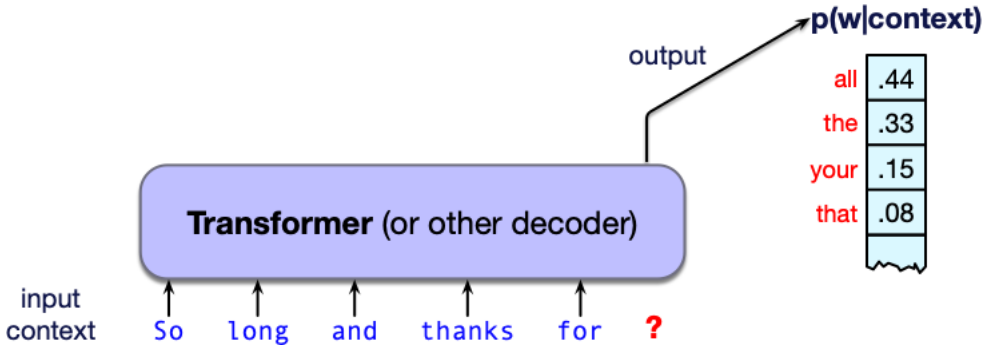
- With roses, dahlias, and peonies, I was surrounded by flowers
- The room wasn't just big it was enormous
- The square root of 4 is 2
- The author of "A Room of One's Own" is Virginia Woolf
- The doctor told me that he
- So long and thanks for all

12

12

What is a large language model?

- A neural network with:
  - **Input**: a context or prefix,
  - **Output**: a distribution over possible next words

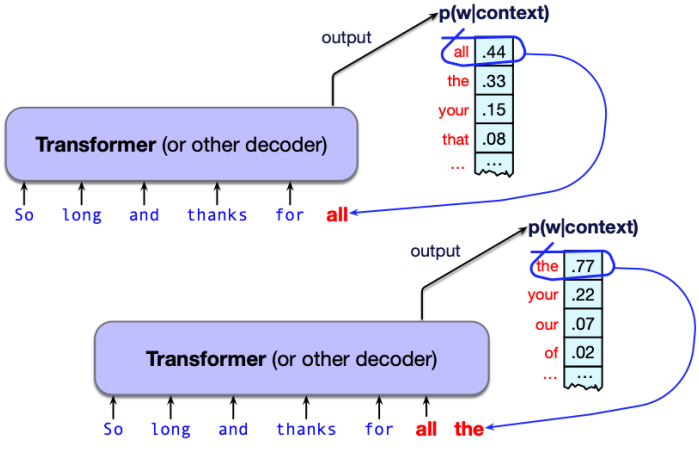


13

13

LLMs can generate

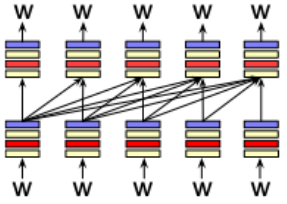
- A model that gives a probability distribution over next words can generate by repeatedly sampling from the distribution

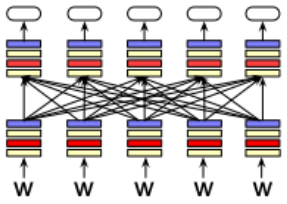


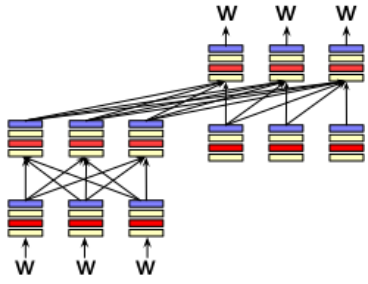
14

14

Three architectures for LLMs







- **Decoders**
- GPT, Claude,
- Llama
- Mixtral

Encoders

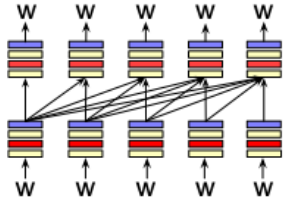
BERT family, HuBERT

Encoder-decoders

Flan-T5, Whisper

15

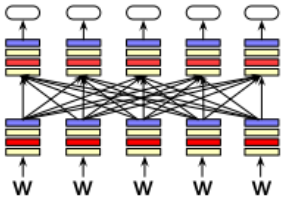
Decoders



- 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*)

16

Encoders



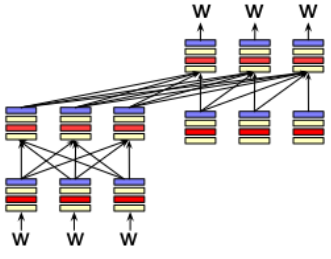
- Masked Language Models (MLMs), 2018
- BERT family
  BERT = **B**idirectional **E**ncoder **R**epresentations from **T**ransformers

- Trained by predicting words from surrounding words on both sides
- Are usually **finetuned** (trained on supervised data) for classification tasks Question Answering/Search, Semantic Analysis, Named Entity Recognition.

17

17

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)

18

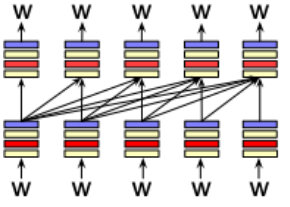
18

LLMs: Big Idea

- Conditional Generation of Text
- Many tasks can be turned into tasks of predicting words!

19

Decoder Models



- Also called…

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

20

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

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

21

21

NLP Tasks as Conditional Text Generation

Sentiment analysis: "I like Jackie Chan"

1. We give the language model this string:

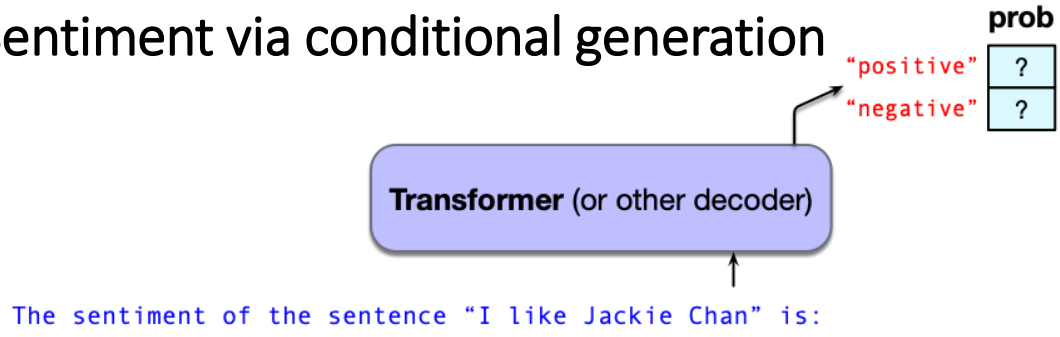
```
The sentiment of the sentence "I like Jackie Chan"
is:
```

2. And see what word it thinks comes next

22

22

Sentiment via conditional generation



Which word has a higher probability?

P(positive|The sentiment of the sentence ``I like Jackie Chan" is:)

P(negative|The sentiment of the sentence ``I like Jackie Chan" is:)

23

NLP Tasks as Conditional Text Generation

Question-Answering: "Who wrote The Origin of Species"

1. We give the language model this string:

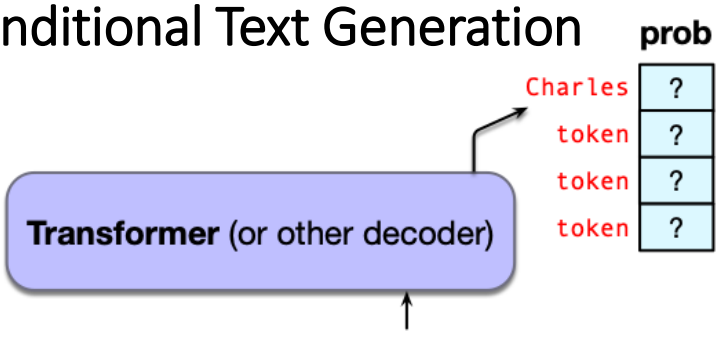
   Q: Who wrote the book ``The Origin of Species"? A:

2. And see what word it thinks comes next:

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

24

NLP Tasks as Conditional Text Generation



Now we iterate:

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

25

Prompt

- **Prompt:** a text string that a user issues to a language model to get the model to do something useful by conditional generation
- **Prompt engineering:** the process of finding effective prompts for a task.

26

Prompts

A question:

```
What is a transformer network?
```

Perhaps structured:

```
Q: What is a transformer network? A:
```

Or an instruction:

```
Translate the following sentence into Swahili:
‘Chop the garlic finely’.
```

27

Prompts can be very structured

A prompt consisting of a review plus an incomplete statement:

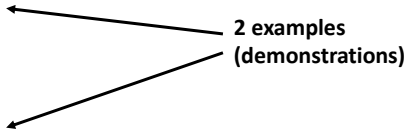
```
Human: Do you think that “input” has positive or
negative sentiment?
Choices:
(P) Positive
(N) Negative

Assistant: I believe the best answer is: (
```

28

Prompts can include examples (demonstrations)

The following are multiple choice questions about high school computer science.

Let x = 1. What is X << 3 in Python 3?
(A) 1 (B) 3 (C) 8 (D) 16
Answer: C



Which is the largest Asymptotically?
(A) O(1) (B) O(n) (C) O(n^2) (D) O(n log(n))
Answer: C

What is the output of the statement "a" + "ab“ in Python 3?
(A) Error (B) aab (C) ab (D) a ab
Answer:

29

29

Prompts are a learning signal

This is especially clear with demonstrations.

But this is a different kind of learning than pretraining

- Pretraining sets language model weights via gradient descent
- Prompting just changes the context and the activations in the network; **no parameters change**

- We call this **in-context learning**—learning that improves model performance but does not update parameters

30

30

LLMs usually have a system prompt

<system>You are a helpful and knowledgeable assistant. Answer concisely and correctly.

This is automatically and silently concatenated to a user prompt

<system> You are a helpful and knowledgeable assistant. Answer concisely and correctly.
<user> ***What is the capital of France?***

31

31

System prompts can be long; 1700 words for Claude Opus4

An Extract:

Claude should give concise responses to very simple questions, but provide thorough responses to complex and open-ended questions.

Claude is able to explain difficult concepts or ideas clearly. It can also illustrate its explanations with examples, thought experiments, or metaphors.

Claude does not provide information that could be used to make chemical or biological or nuclear weapons.

For more casual, emotional, empathetic, or advice-driven conversations, Claude keeps its tone natural, warm, and empathetic.

Claude cares about people's well-being and avoids encouraging or facilitating self-destructive behavior.

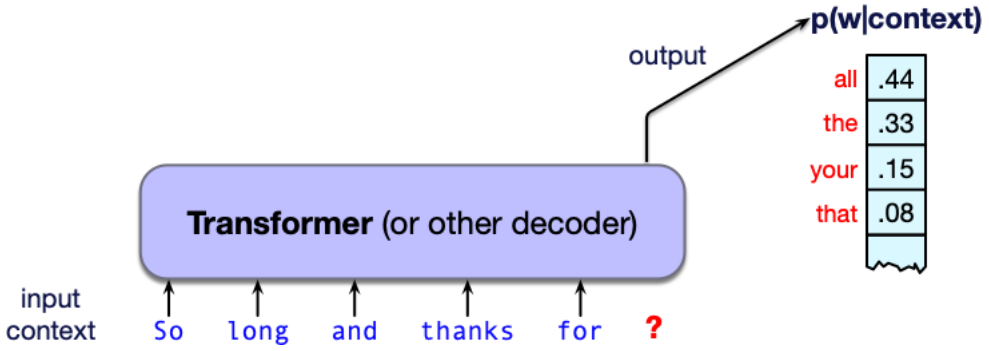
If Claude provides bullet points in its response, it should use markdown, and each bullet point should be at least 1-2 sentences long unless the human requests otherwise.

32

32

What is a large language model?

- A neural network with:
  - **Input**: a context or prefix,
  - **Output**: a distribution over possible next words



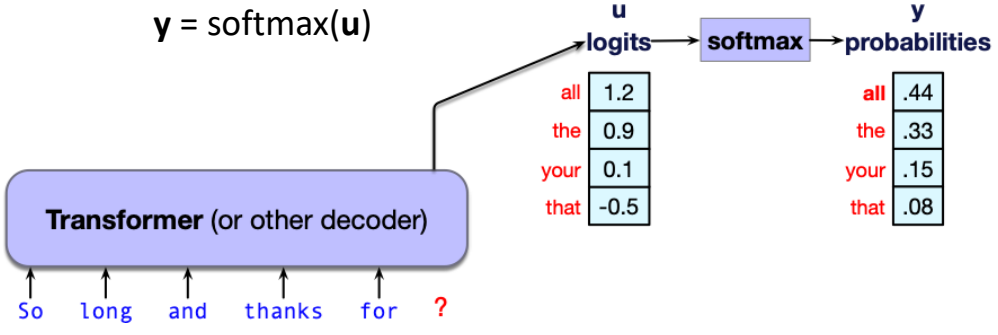
33

33

Where does token probability come from?

- The internal networks for LLMs generate real-valued scores called **logits** for each token in the vocabulary.
- Score vector **u** of shape $[1 \times |V|]$ is turned into a probability by softmax

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



34

34

Decoding

- This task of choosing a word to generate based on the model's probabilities is called **decoding**.
- Decoding from a model left-to-right and repeatedly choosing the next token conditioned on our previous choices is called **autoregressive generation**.

  There are several techniques for choosing a word: **greedy decoding**, various **sampling techniques (random, temperature, top-k, top-p).**

35

35

Greedy Decoding

A **greedy algorithm** is one that makes a choice that is locally optimal (whether or not it will turn out to have been the best choice with hindsight)

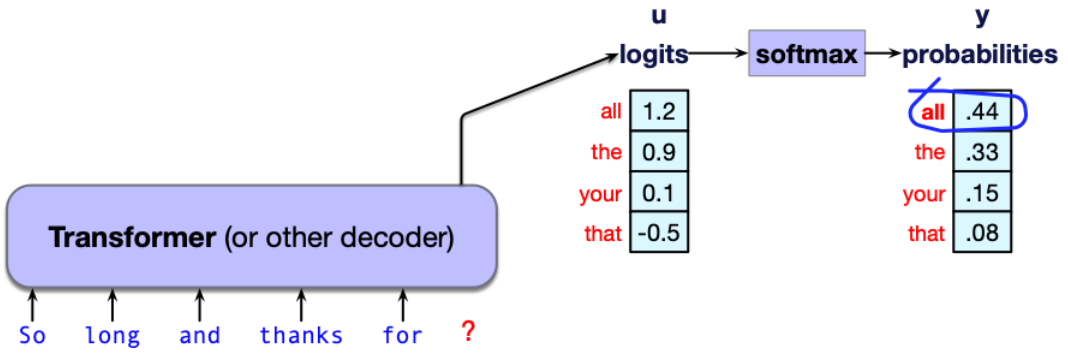
Simply generate the most probable word:

$$w'_t = argmax_{w \in V} P(w|w_{<t})$$

36

36

Greedy Decoding: choosing "all"



37

37

Greedy Decoding is not typically used

- Because the tokens it chooses are (by definition) extremely predictable, the resulting text is **generic** and **repetitive**
- Greedy decoding is so predictable that it is **deterministic**.
- Instead, people prefer text that is more diverse, like that generated by **sampling**

38

38

Random Sampling

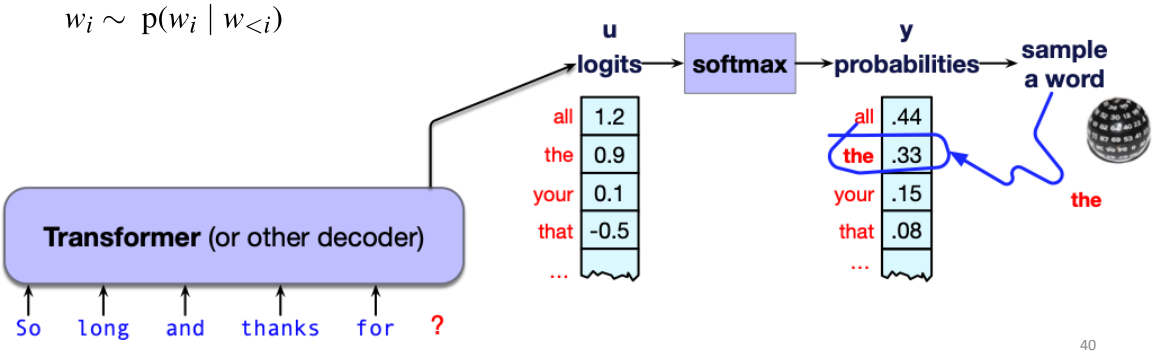
- **Sampling** from a distribution means to choose random points according to their likelihood.
- **Sampling for an LM** means to choose the next token to generate according to its probability.
- **Random Sampling**: We randomly select a token to generate according to its probability defined by the LM, conditioned on our previous choices, generate it, and iterate.

39

39

Random Sampling

$i \leftarrow 1$
$w_i \sim p(w)$
while w_i != EOS
  $i \leftarrow i + 1$
  $w_i \sim p(w_i \mid w_{<i})$



40

40

Random Sampling is also not typically used

- Even though random sampling mostly generates 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

41

41

Temperature Sampling

Reshape the probability distribution

- increase the probability of the high probability tokens
- decrease the probability of the low probability tokens

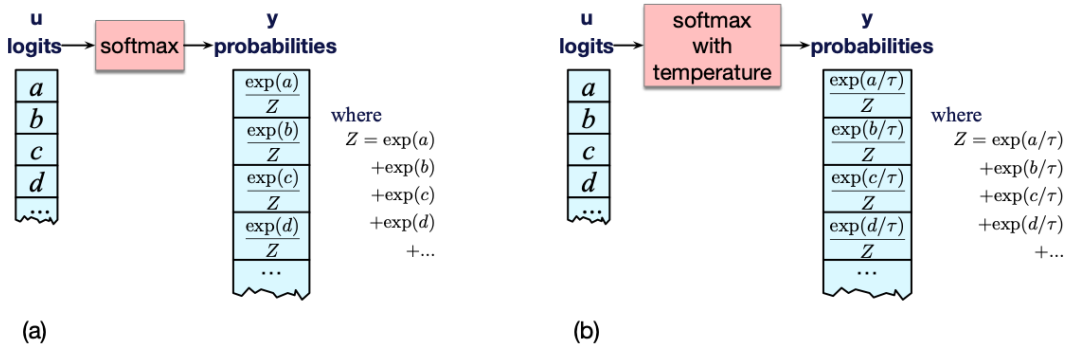
Divide the logit by a temperature parameter τ ($0 < \tau \leq 1$) before passing it through softmax.

- Instead of ~~$\mathbf{y} = \text{softmax}(u)$~~
- We do $\mathbf{y} = \text{softmax}(u/\tau)$

42

42

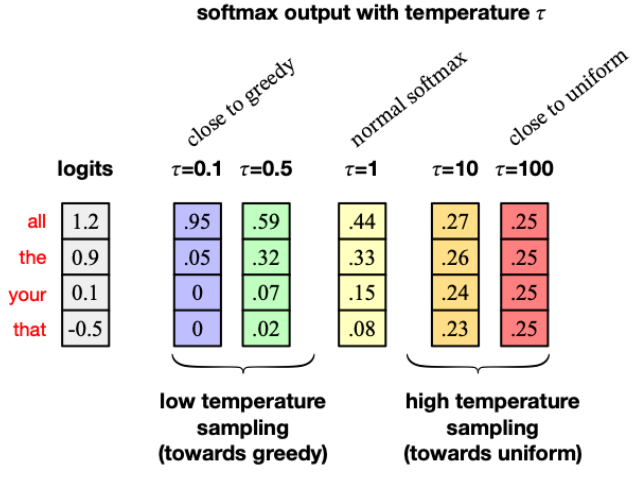
Temperature Sampling



Low Temperature Sampling

In **low-temperature sampling**, (τ ≤ 1) we smoothly

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



Top-*k* and Top-*p* Sampling

Top-*k* Sampling

1. Choose a number of words *k*
2. For each $w \in V$, compute $p(w_t|w_{<t})$
3. Sort words by their likelihood and select the top *k* most probable words.
4. Renormalize scores of the *k* words into a probability distribution.
5. Randomly sample from within the *k* words according to the probability distribution.

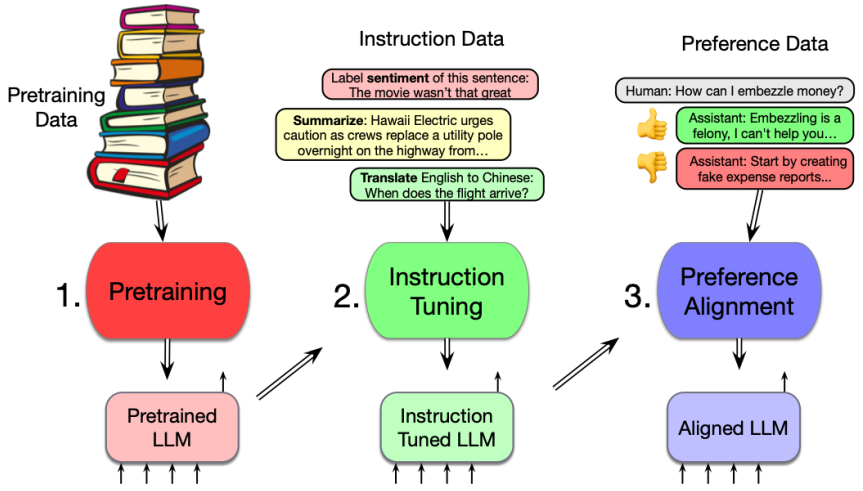
Top-*p* Sampling

1. Choose *p* as the top *p* percent of the probability distribution.
2. Given the distribution $P(w_t|w_{<t})$ then the top-*p* is the smallest set of words such that

$$\sum_{w \in \mathbf{V}} P(w|w_{<t}) \geq p$$

45

Three stages of training in LLMs



46

Pretraining

The big idea that underlies all the amazing performance of language models

- First **pretrain** a transformer model on enormous amounts of text
- Then **apply** it to new tasks.

47

47

Self-supervised training algorithm

Train the neural network to predict the next word.

1. Take a corpus of text
2. At each time step *t*
   i. ask the model to predict the next word
   ii. train the model using gradient descent to minimize the error in this prediction

Self-supervised because it just uses the next word as the label/target!

48

48

Teacher forcing

- At each token position *t*, model sees correct tokens $w_{1:t}$
  - Computes loss (cross entropy) for the next token w_{t+1}
- At next token position t+1 we ignore what model predicted for w_{t+1}
  - Instead we take the **correct** word w_{t+1}, add it to context, move on

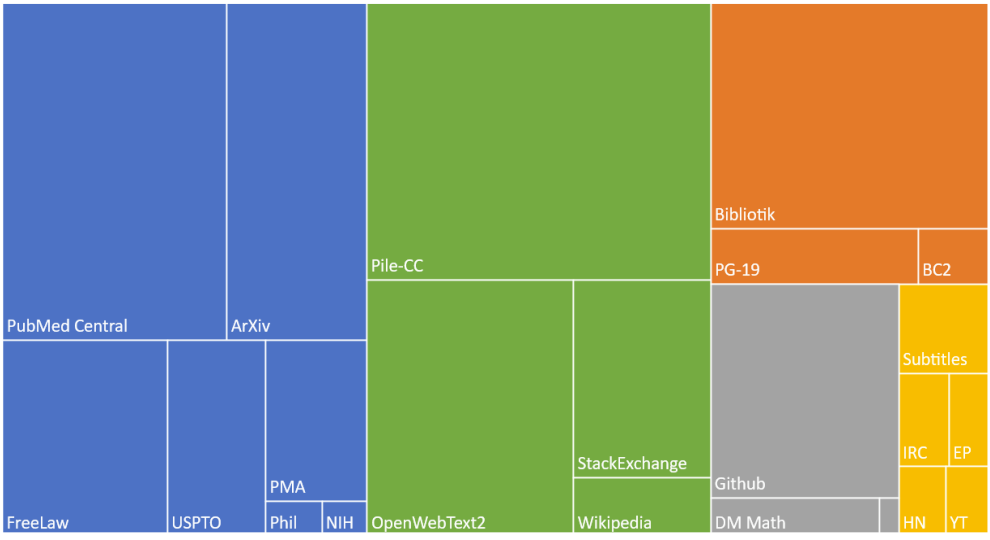
49

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)**: 156 billion tokens of English, filtered
- What's in it? Mostly patent text documents, Wikipedia, and news sites

50

The Pile: a pretraining corpus (825Gb)



51

51

Filtering for quality and safety

Quality is subjective

- Many LLMs attempt to match Wikipedia, books, particular websites
- Need to remove boilerplate, adult content
- Deduplication at many levels (URLs, documents, even lines)

Safety also subjective

- Toxicity detection is important, although that has mixed results
- Can mistakenly flag data written in dialects like African American English

52

52

Problems with scraping from the web

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

The Times Sues OpenAI and Microsoft Over A.I. Use of Copyrighted Work

Millions of articles from The New York Times were used to train chatbots that now compete with it, the lawsuit said.

53

53

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 across the world

Data consent

- Website owners can indicate they don't want their site crawled

Privacy

- Websites can contain private IP addresses and phone numbers

Skew

- Training data is disproportionately generated by authors from the US which probably skews resulting topics and opinions

54

54

Finetuning: adaptation to new domains

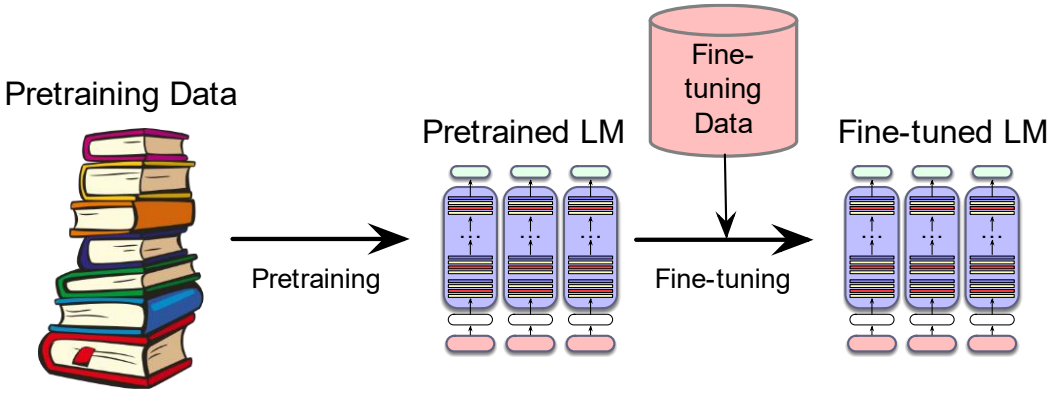
- What happens if we need our LLM to work well on a domain it didn't see in pretraining?
- Perhaps some specific medical or legal domain?
- Or maybe a multilingual LM needs to see more data on some language that was rare in pretraining?

Finetuning is taking a pretrained model and further adapting some or all its parameters to some new data

55

55

Finetuning



56

56

Continued Pretraining

Further train all the parameters of model on new data

- using the same method (word prediction) and loss function (cross-entropy loss) as for pretraining.
- as if the new data were at the tail end of the pretraining data

- This is sometimes called **continued pretraining**

57

57

Evaluating LLMs

Perplexity

- How well a model predicts unseen text

Size

- Big models take lots of GPUs and time to train, memory to store

Energy usage

- Can measure kWh or kilograms of CO2 emitted

Fairness

- Benchmarks measure gendered and racial stereotypes or decreased performance for language from or about some groups.

58

58

Ethical and Safety Issues: 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.

59

59

Ethical and Safety Issues: Privacy

How Strangers Got My Email Address From ChatGPT's Model

60

60

Ethical and Safety Issues: 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

61

61

Ethical and Safety Issues: Lots more…

- Harm (suggesting dangerous actions)
- Fraud
- Emotional dependence
- Bias

62

62

Vocabulary

Language Model
NGrams
Large Language Model
Encoder
Decoder
Conditional Generation
Prompts
Prompt Engineering
In-context Learning
System Prompt
Greedy Decoding
Random Sampling
Temperature Sampling
Pretraining
Teacher Forcing
Finetuning
Continued Pretraining
Evaluating LLMs
Ethical Issues
Safety Issues

63

63

References

- F. Chollet: *Deep Learning with Python*, 2nd Edityion. Manning. 2021.
- Jurafsky, D. and Martin, J.: Speech and Language Processing. Third Edition (forthcoming) (Draft Chapters 7 and 8). 2025.
- A. Karpathy: *Introduction to Large Languag Models*. 2023. YouTube Video: https://www.youtube.com/watch?v=zjkBMFhNj_g
- M. Mitchell: *Artificial Intelligence: A Guide For Thinking Humans*, Farrar, Strouss, Giroux, 2019.
- M. Wooldridge: *A Brief History of Artificial Intelligence*. Flatiron Books, 2020.
- *Word Embedding Demo*: https://www.cs.cmu.edu/~dst/WordEmbeddingDemo/

64

64