Concepts and Applications in NLP

Large Language Models

Marion Di Marco

February 4, 2025

Language Models

- N-gram language models
  - assign probabilities to sequences of words
  - generate text by sampling possible next words
  - are 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**

Large Language Models

- Pretraining: learning knowledge about language and the world from vast amounts of text
- LLMs: remarkable performance on many NLP tasks due to knowledge obtained in pretraining
- Especially for tasks where text is produced
  - summarization
  - machine translation
  - question answering
  - chatbots
- Many tasks can be turned into tasks of predicting words!

Outline

Solving NLP Tasks with LLMs

Sampling for LLM Generation

Pretraining Large Language Models

Training Data

Fine-tuning

Prompting

Instruction Tuning

Potential Harms from Large Language Models

Credits

Conditional Generation

- Causal or autoregressive language models:
  iteratively predict words left-to-right from earlier words
- Conditional generation: generating text conditioned
  on an input piece of text
- Input text (prompt) → LLM continues generating text token by token
- Transformers have long context windows (many thousands of tokens)
  → very powerful for conditional generation

Conditional Generation

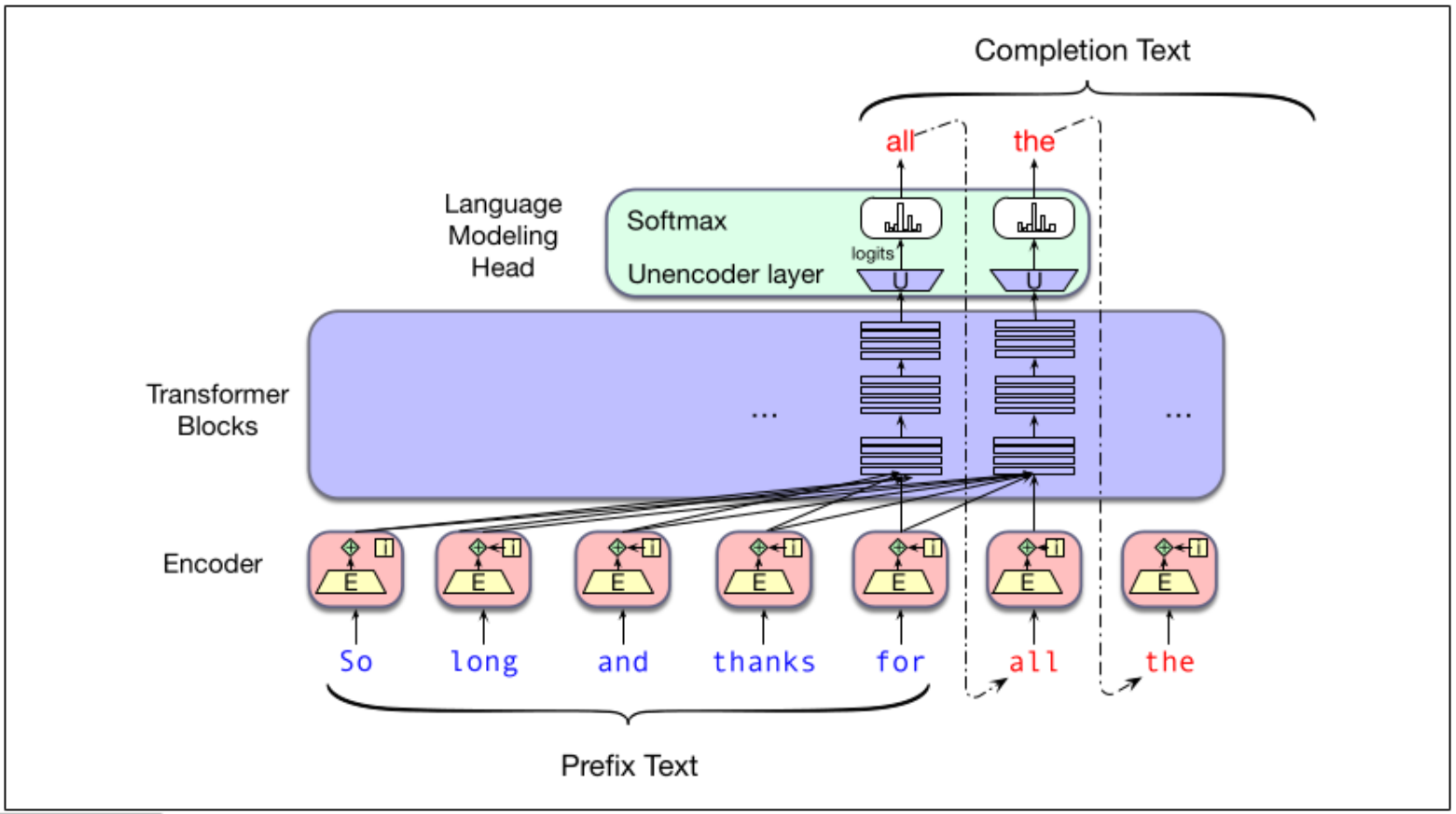


Figure 10.1 Left-to-right (also called autoregressive) text completion with transformer-based large language models. As each token is generated, it gets added onto the context as a prefix for generating the next token.

Access to the priming context and all of the subsequently generated outputs (within the context window)

NLP Tasks as Word Prediction Task

- Sentiment analysis as language modeling task
- Provide a context like

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

- What word comes next?

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

NLP Tasks as Word Prediction Task

- Question answering as language modeling task
- Give the LM a question and a token like `A:` to suggest that an answer should come next
- `Q: Who wrote the book "The Origin of Species"?  A:`
- What word comes next?

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

  $\Rightarrow$ *Charles* is very likely $\rightarrow$ select it
- Iterate

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

  $\Rightarrow$ *Darwin* is very likely $\rightarrow$ select it

Text Summarization

- Generate longer responses
- Text summarization: take a long text and produce a shorter summary
- Give the LM the text followed by a token like `tl;dr`
  (*too long; didn't read*)
- `tl;dr` sufficiently frequent in language model training data → interpret as instruction to create summary

Summarization

Original Article

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.

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.

Figure 10.2 Excerpt from a sample article and its summary from the CNN/Daily Mail summarization corpus (Hermann et al., 2015b), (Nallapati et al., 2016).

Summarization

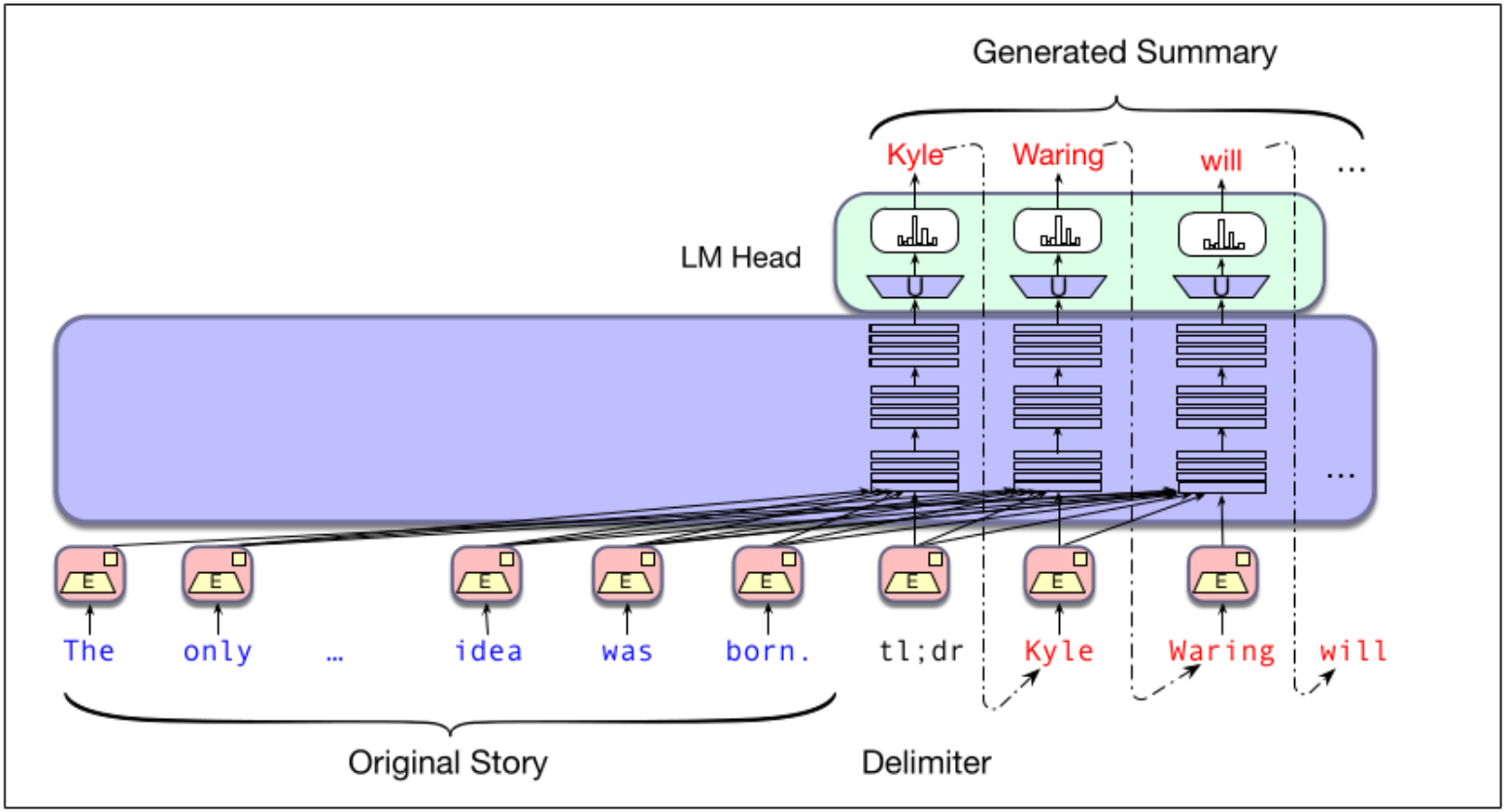


Figure 10.3 Summarization with large language models using the `tl;dr` token and context-based autoregressive generation.

Incorporate information from large context window as well as newly generated output

Outline

Solving NLP Tasks with LLMs

Sampling for LLM Generation

Pretraining Large Language Models

Training Data

Fine-tuning

Prompting

Instruction Tuning

Potential Harms from Large Language Models

Credits

Decoding and Sampling

- Decoding: choosing a word to generate based on the model's probabilities
- The most common method for decoding in LLMs: sampling
- Sampling from a model's distribution over words:
  choose words according to the probability assigned by the model
- Iteratively choose a word to generate according to its probability in context as defined by the model
  - more likely to generate words that have a high probability according to the model
  - less likely to generate words that have a low probability according to the model

Random Sampling

- At each step: sample words according to their probability conditioned on previous choices
- Generate a sequence of words $W = w_1, w_2, ..., w_N$ until an end-of-sequence token is hit
- $x \sim p(x)$: choose x by sampling from the distribution $p(x)$

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

Random Sampling

- Random sampling doesn't work well
- Random sampling mostly generates sensible, high-probable words
- But: there are many odd, low-probability words in the tail of the distribution
  - each one is low-probability
  - added up: rare constitute a large portion of the distribution

  $\Rightarrow$ Weird sentences
- Sampling methods to avoid generating from the tail of the distribution

Factors in Sampling

- Emphasize high-probability words
  - \+ **quality**: more accurate, coherent, and factual
  - \- **diversity**: boring repetitive

- Emphasize middle-probability words
  - \+ **diversity**: more creative and 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

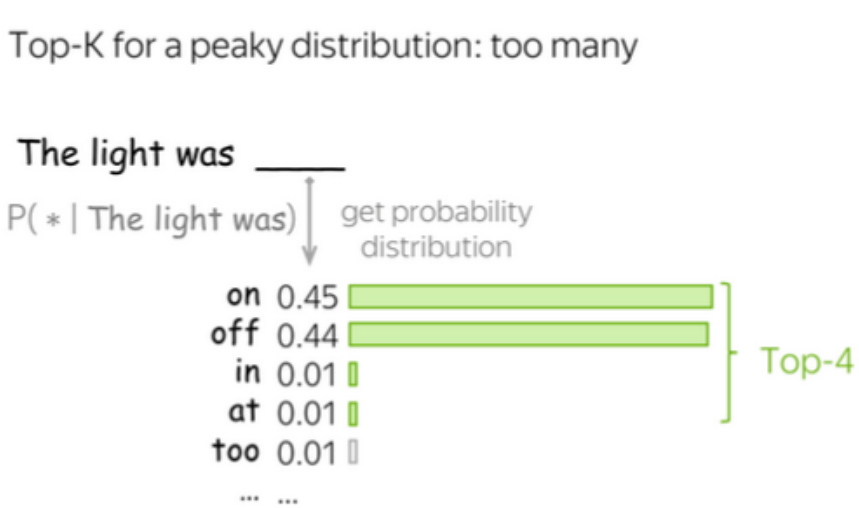
(4) Renormalize the scores of the k words to be a legitimate probability distribution

(5) Randomly sample a word from within these remaining k most-probable words according to its probability

- Simple generalization of greedy decoding
(greedy decoding with $k = 1$)

Top-k Sampling

- A fixed k is not always good: the top-k most probable tokens may
  - cover very small part of the total probability mass (in flat distributions);
  - contain very unlikely tokens (in peaky distributions).



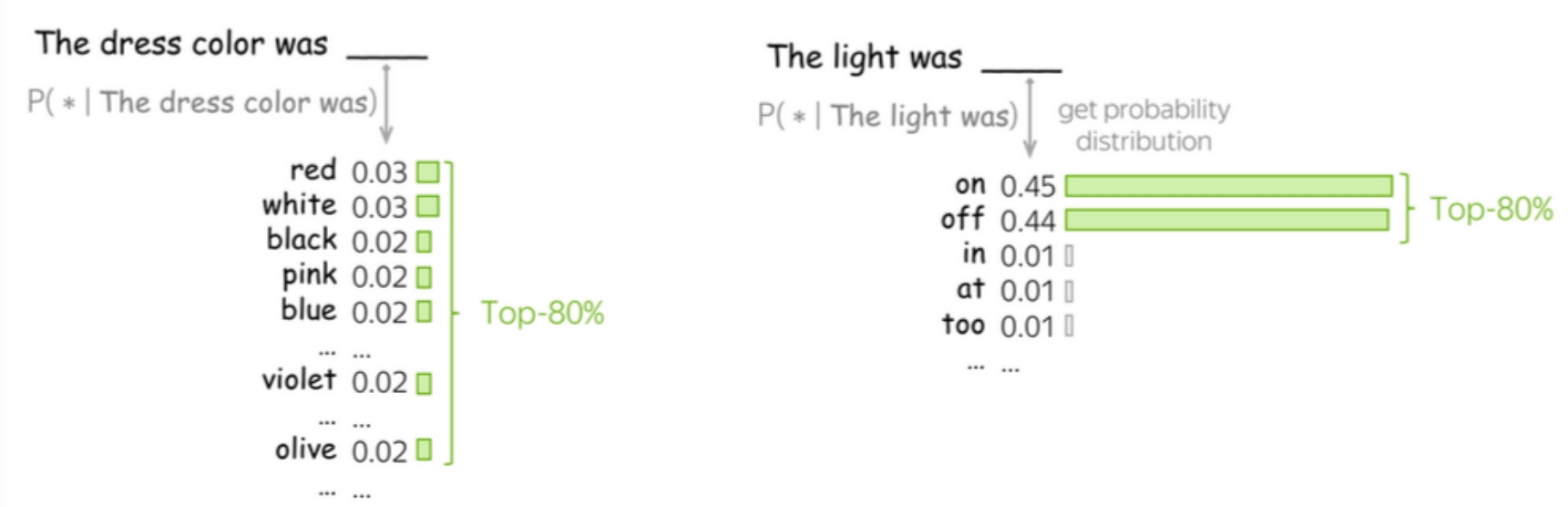
from `https://lena-voita.github.io/nlp_course/language_modeling.html`

Nucleaus or Top-p Sampling

- Don't keep the top k words, but the top p percent of the probability mass
- Truncate the distribution to remove the very unlikely words
- More robust in very different contexts $\rightarrow$ dynamically increase and decrease the pool of word candidates
- Given a distribution $P(w_t|w_{<t})$, the top$-p$ vocabulary $V^{(p)}$ is the smallest set of words such that

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

Nucleaus or top-p Sampling



from `https://lena-voita.github.io/nlp_course/language_modeling.html`

Temperature Sampling

- Don't truncate the distribution, but reshape 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 ($\tau \leq 1$)
  - increase the probability of the most probable words
  - decrease the probability of the rare words
- Divide logit by a temperature parameter τ before normalizing:

$$y = softmax(\frac{u}{\tau})$$

Temperature Sampling

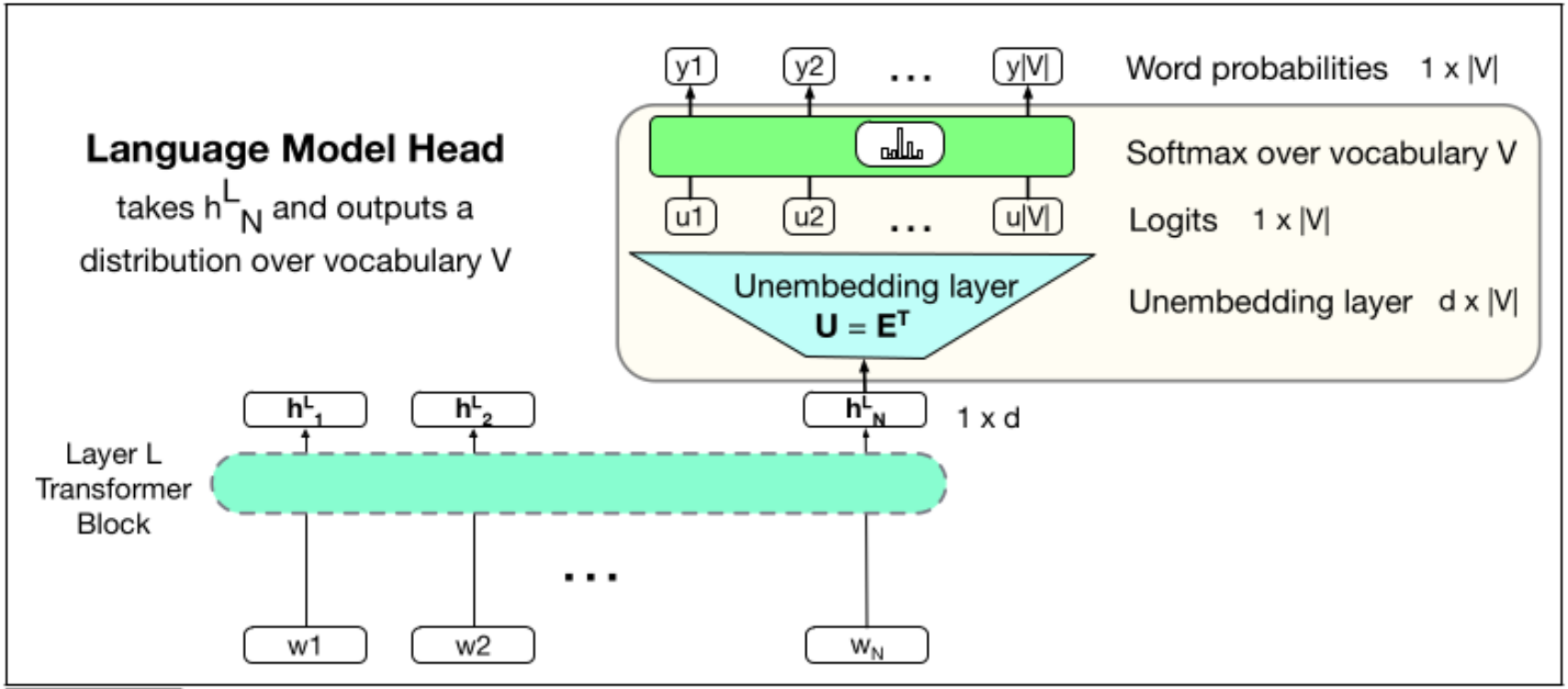
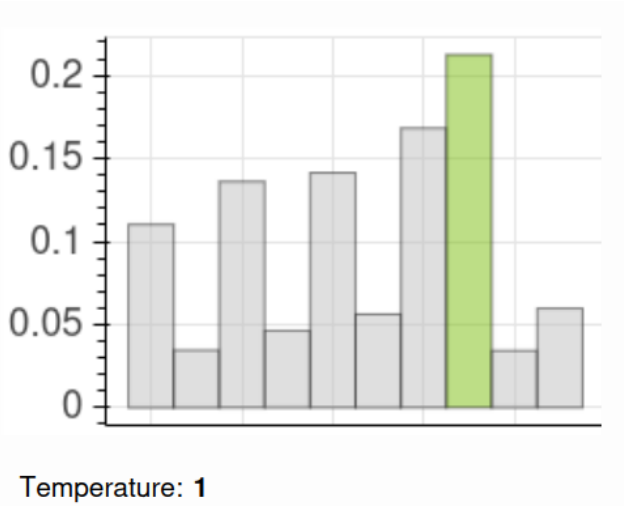


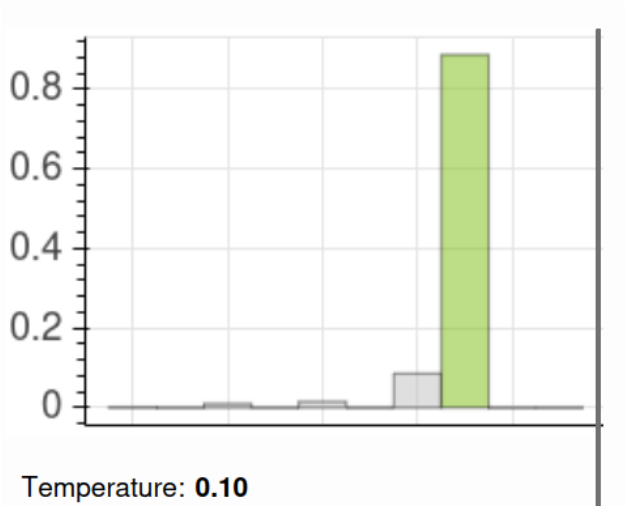
Figure 9.14 The language modeling head: the circuit at the top of a transformer that maps from the output embedding for token N from the last transformer layer ($\mathbf{h}_N^L$) to a probability distribution over words in the vocabulary V.

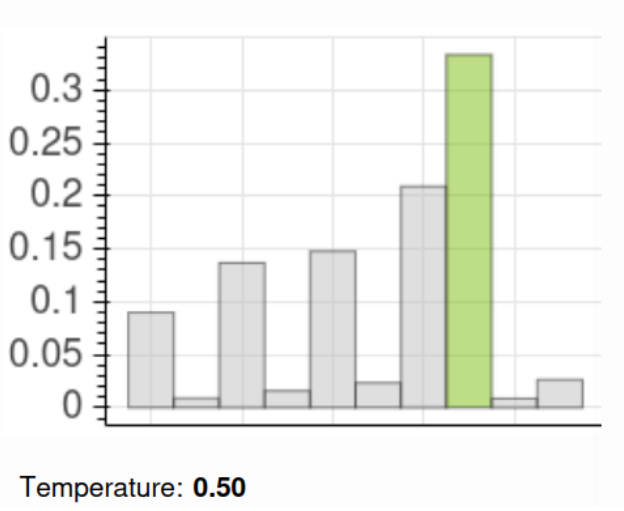
Temperature Sampling

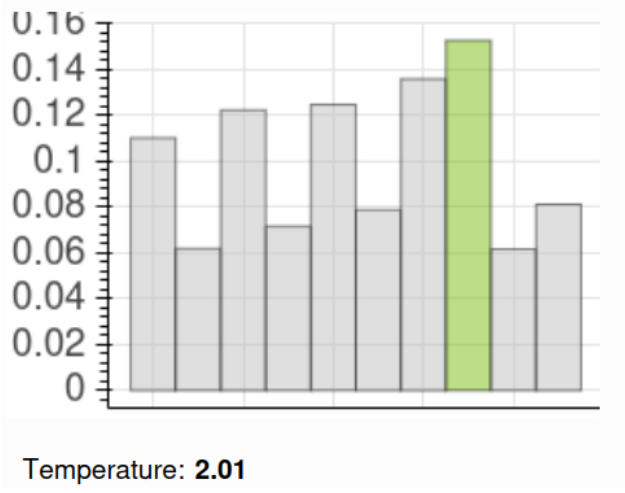
- When τ is close to 1 the distribution does not change much
- The lower τ is, the larger the scores being passed to the softmax
- Softmax pushes high values toward 1 and low values toward 0
- Distribution becomes more greedy
  - increased probabilities of the most high-probability words
  - decreased probabilities of the low probability words
- As τ approaches 0 the probability of the most likely word approaches 1

Temperature Sampling









from `https://lena-voita.github.io/nlp_course/language_modeling.html`

Outline

Solving NLP Tasks with LLMs

Sampling for LLM Generation

Pretraining Large Language Models

Training Data

Fine-tuning

Prompting

Instruction Tuning

Potential Harms from Large Language Models

Credits

Pretraining Large Language Models

- Pretraining: learning knowledge about language and the world from vast amounts of text
- Knowledge learned from text data → remarkable performance on many NLP tasks

- Pretrain a transformer model on enormous amounts of text, then apply it to new tasks
  - → training algorithm
  - → data

Self-supervised Training Algorithm

- Train to predict the next word

  Take a text corpus

  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 prediction

- Self-supervised: just uses the next word as a label
  the natural sequence of words is its own supervision

Intuition of Language Model Training: Loss

- Loss function: cross-entropy loss
- We want the model to assign a high probability to true word w
- Loss should be high if the model assigns too low a probability to w
- CE Loss: the negative log probability that the model assigns to the true next word w
  - if the model assigns too low a probability to w
  - move the model weights in the direction that assigns a higher probability to w

Cross-Entropy Loss for Language Modeling

- Cross-entropy loss measures the difference between a **predicted probability distribution** and the **correct distribution**

$$L_{CE} = -\sum_{w \in V} \mathbf{y}_t[w] \log \hat{\mathbf{y}}_t[w]$$

correct probability distribution

predicted probability distribution

- Correct distribution y_t comes from knowing the next word: one-hot vector where the entry for the actual word is 1, all others are 0

- All terms get multiplied by zero, except one: the log-probability assigned to the correct next word

- CE loss at time t: $L_{CE}(\hat{\mathbf{y}}_t, \mathbf{y}_t) = -\log \hat{\mathbf{y}}_t[w_{t+1}]$

Teacher Forcing

- At token position t, the model sees the correct token sequence $w_{1:t}$

  computes probability distribution over possible next words
  to compute loss for next token w_{t+1}

- Move to the next word: ignore previous prediction,
  but use the correct sequence $w_{1:t+1}$ to estimate token w_{t+2}
  (teacher forcing)

Training

- At each step: final transformer layer produces an output distribution given all preceding words
- Probability assigned to the correct word: calculate CE loss for each item in the sequence
- Loss for a training sequence: average cross-entropy loss over sequence
- Network weights are adjusted to minimize the average CE loss via gradient descent

Training

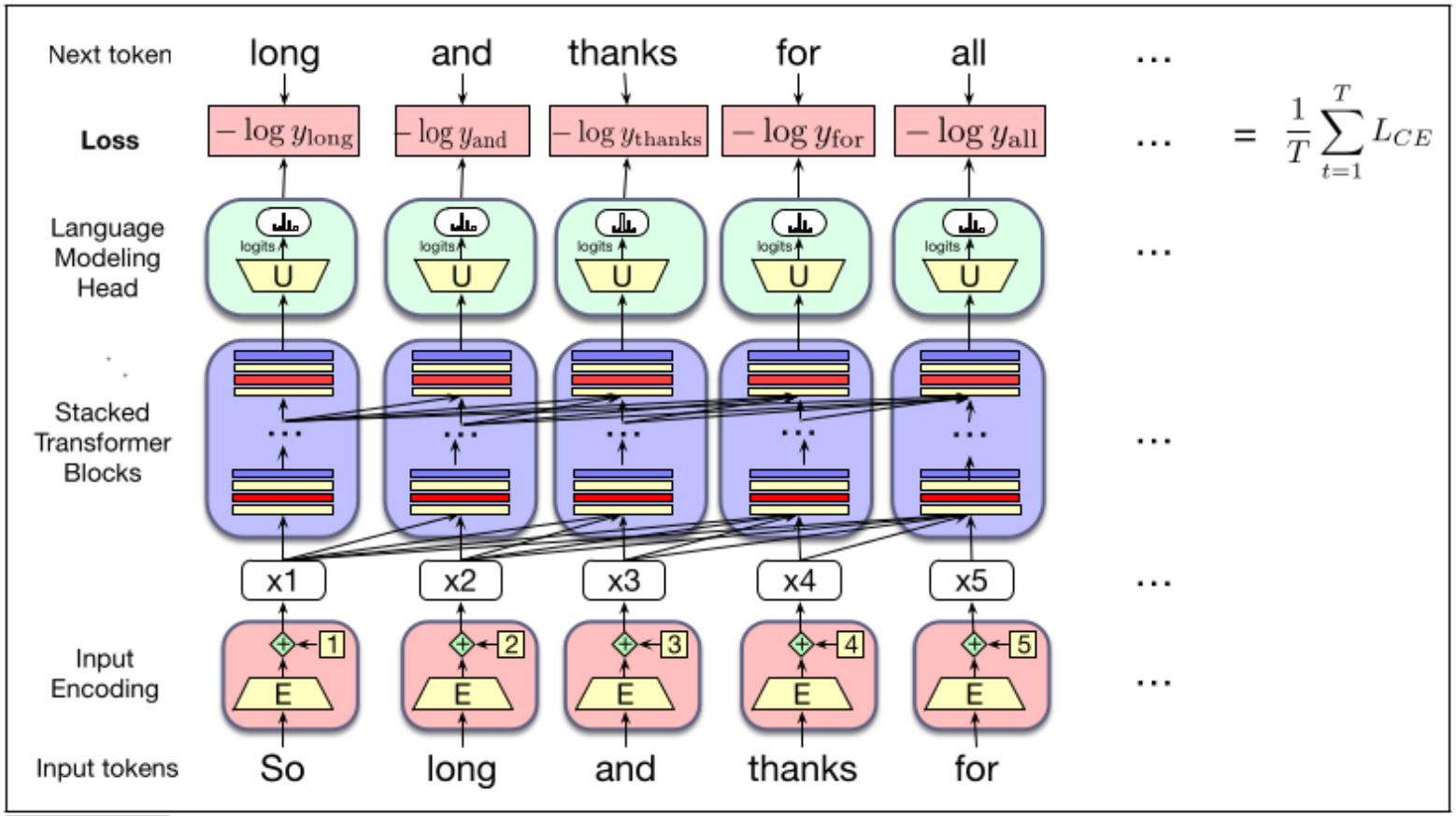


Figure 10.4 Training a transformer as a language model.

Outline

Solving NLP Tasks with LLMs

Sampling for LLM Generation

Pretraining Large Language Models

Training Data

Fine-tuning

Prompting

Instruction Tuning

Potential Harms from Large Language Models

Credits

Training Data

- Idea: text contains enormous amounts of knowledge
- Pretraining on huge text collections → enable LMs to solve many problems
- Large corpora: likely to contain natural examples for NLP tasks
  - question – answer pairs
  - documents + summaries (tl;dr)
  - translations
  - and more ...

What Can a Model Learn from Pretraining?

- There are canines everywhere! One dog in the front room, and two dogs ...
- It wasn't just big it was enormous
- The author of "A Room of One's Own" is Virginia Woolf
- The doctor told me that he ...
- The square root of 4 is 2

Training Data

- Automatically-crawled web data
- Common crawl: `https://commoncrawl.org`
  - for example the Colossal Clean Crawled Corpus (C4) Raffel et al. (2020)
  - 156 billion tokens of English
  - filtered in various ways (deduplicated, removing non-natural language like code, sentences with offensive words from a blocklist)
- Wikipedia
- Book corpora
- The Pile: 825 GB English corpus Gao et al. (2020)
- Dolma: 3 trillion tokens; web text, academic papers, code, books, encyclopedic materials, and social media Soldaini et al. (2024)

Training Data

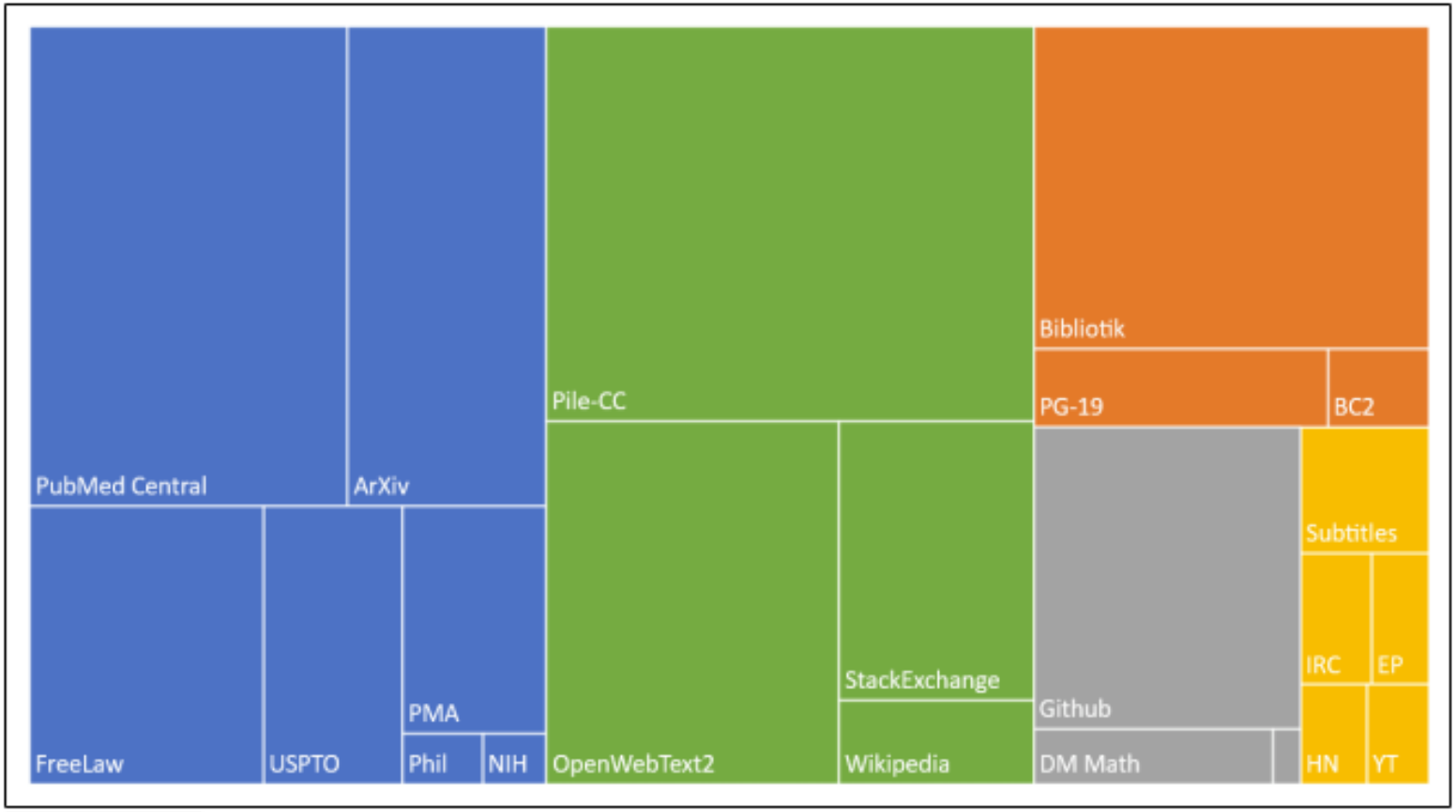


Figure 10.5 The Pile corpus, showing the size of different components, color coded as academic (articles from PubMed and ArXiv, patents from the USPTA; internet (webtext including a subset of the common crawl as well as Wikipedia), prose (a large corpus of books), dialogue (including movie subtitles and chat data), and misc.. Figure from Gao et al. (2020).

Filtering for Quality and Safety

- Quality filters: classifiers to assign a score to each document
- Quality is subjective, different ways to train filters
  - high-quality resources like Wikipedia, books, ...
  - avoid: websites with personal identifiable information, adult content, ...
  - remove duplicates
- Quality filtering generally improves LM performance
- Safety filtering
  - toxicity filtering

Web Crawling: Some Issues

- Copyright: much text is copyrighted
  - "fair use doctrine": unclear if this applies to language modeling
  - remains an open legal question
- Data consent
  - owners of websites can indicate that they don't want their sites crawled
  - increase in websites that don't want to be crawled for LM training data
  - different legal situations in different countries
- Privacy
  - websites can contain information like phone numbers
  - filtering is not always efficient

Outline

Solving NLP Tasks with LLMs

Sampling for LLM Generation

Pretraining Large Language Models

Training Data

Fine-tuning

Prompting

Instruction Tuning

Potential Harms from Large Language Models

Credits

Fine-tuning

- Apply a model to a new domain or a task not sufficiently present in the pre-training data
  - for example: specialize to legal or medical text
  - specialize the model for a particular task
  - the LM needs to see more data of a rare language
- Continue training on relevant data from new domain or language
- Fine-tuning: take a pretrained model and adapt some or all of its parameters on new data new data

Fine-tuning

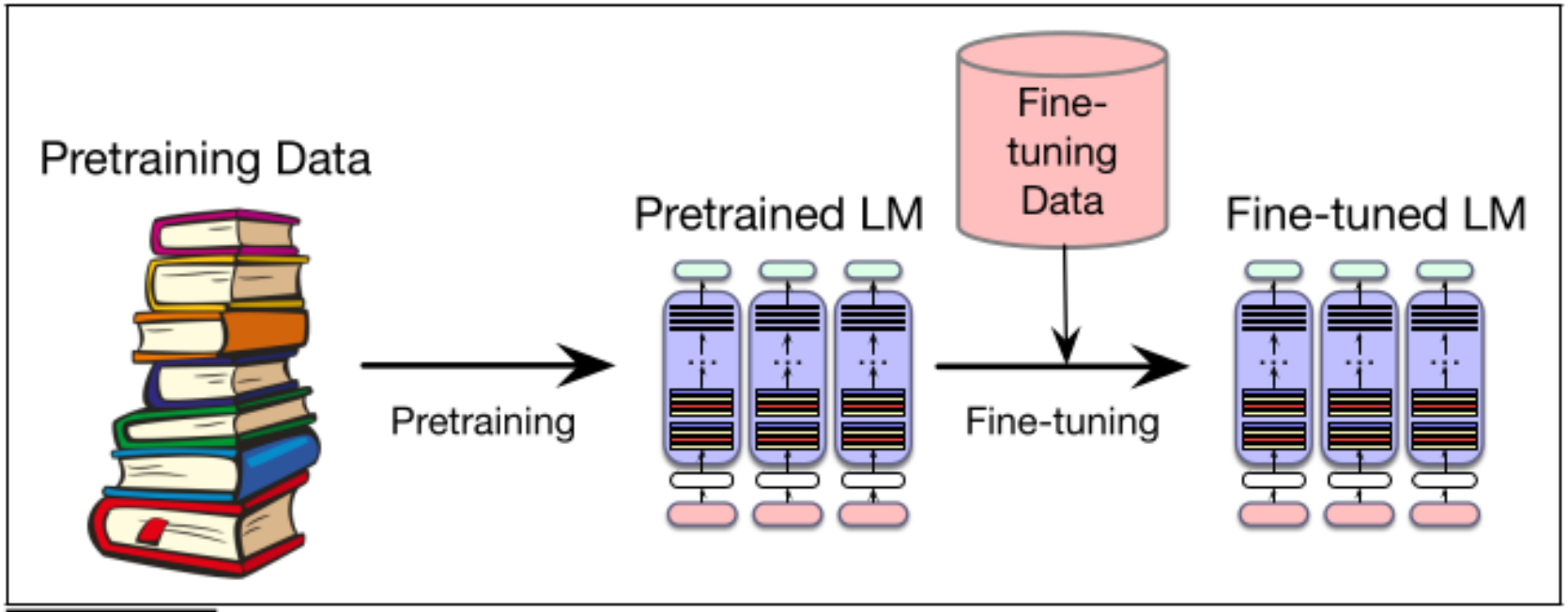


Figure 10.6 Pretraining and finetuning. A pre-trained model can be finetuned to a particular domain, dataset, or task. There are many different ways to finetune, depending on exactly which parameters are updated from the finetuning data: all the parameters, some of the parameters, or only the parameters of specific extra circuitry.

Fine-tuning

- **Continued fine-tuning**: retrain all parameters on new data
  can be slow and expensive

- Freeze some parameters and train only a subset of parameters on the new data: **parameter-efficient fine-tuning**

- LM as classifier of a specific task: take as input some of the top layer embeddings and produce as output a classification
  - often done with BERT models
  - freeze the entire pretrained model and only train the classification head

- **Supervised fine-tuning (SFT)**
  SFT is often used for instruction tuning
  - learn to follow text instructions
  - create a dataset of prompts and desired responses
  - train the LM to produce the desired response from the prompt

Fine-tuning

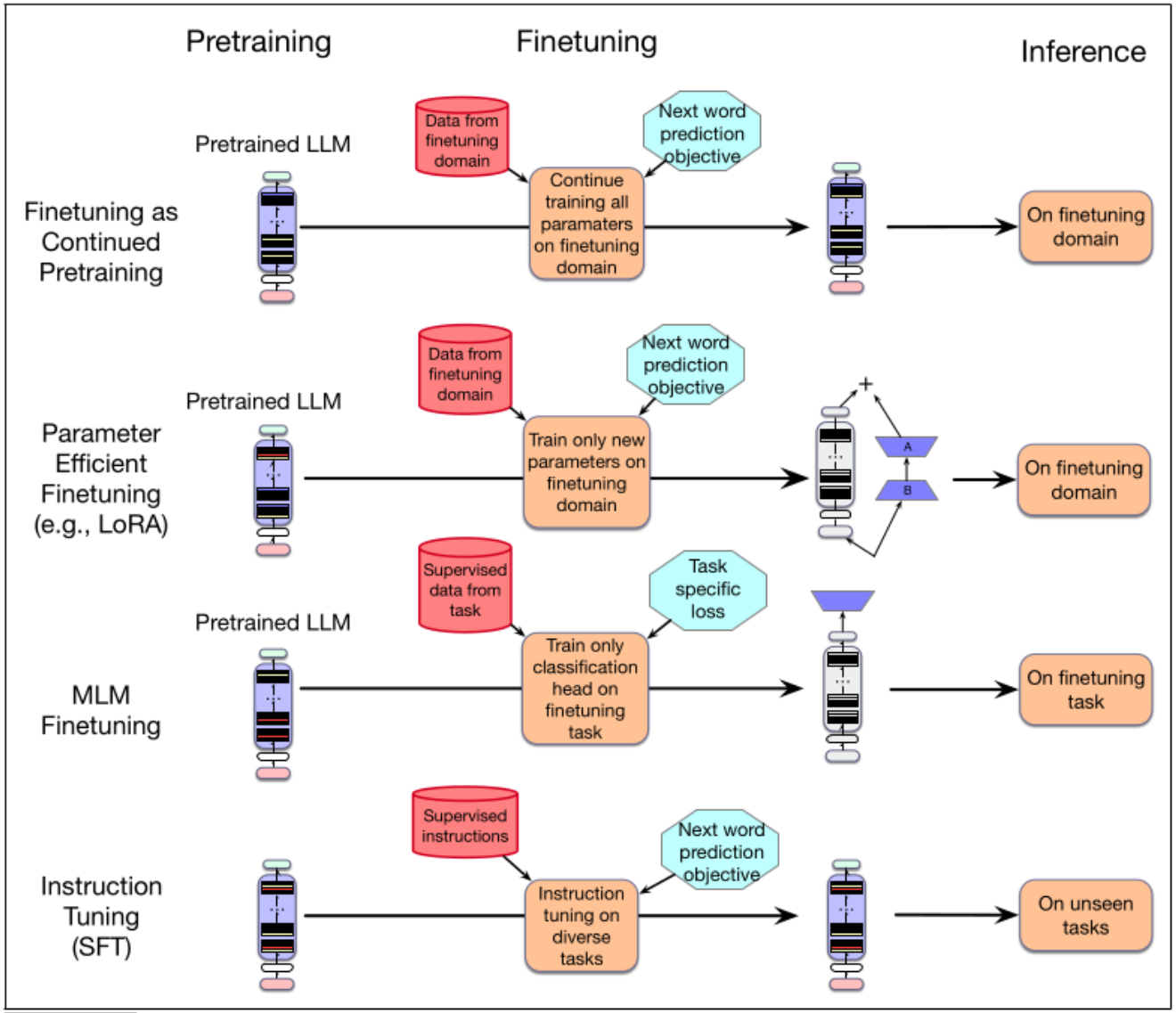


Figure 12.4 Instruction tuning compared to the other kinds of finetuning.

Outline

Solving NLP Tasks with LLMs

Sampling for LLM Generation

Pretraining Large Language Models

Training Data

Fine-tuning

Prompting

Instruction Tuning

Potential Harms from Large Language Models

Credits

Prompting

- Prompt: a text string that a user issues to a language model
- LM then iteratively generates tokens conditioned on the prompt
- Prompt creates a context to guide LLMs to generate useful output
- Prompt engineering: finding effective prompts for a task

Prompts for Sentiment Classification

Sample Hotel Review

Did not like the service that I was provided, when I entered the hotel. I also did not like the area, in which the hotel was located. Too much noise and events going on for me to feel relax.

A prompt consisting of a review plus an incomplete statement

Did not like the service that I was provided, when I entered the hotel. I also did not like the area, in which the hotel was located. Too much noise and events going on for me to feel relax. In short, our stay was

Hotel Review Completions

Did not like the service that I was provided, when I entered the hotel. I also did not like the area, in which the hotel was located. Too much noise and events going on for me to feel relaxed. In short our stay was

---

```
... not a pleasant one. The staff at the front desk
were not welcoming or friendly, and seemed disinterested
in providing good customer service.

... uncomfortable and not worth the price we paid. We
will not be returning to this hotel.
```

Prompt Templates

Basic Prompt Templates

Summarization	{input} ; `tldr;`
Translation	{input} ; `translate to French:`
Sentiment	{input}; `Overall, it was`
Fine-Grained-Sentiment	{input}; `What aspects were important in this review?`

- Input text, designated as "input", followed by a verbatim prompt
- Instruction at the end → constraints the generation

  `Translate English to French: Did not like the service that I was provided!` → may instead generate another sentence in English
- Specify set of answers

A prompt consisting of a review plus an incomplete statement

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

Assistant: I believe the best answer is: (

Prompts for Sentiment Classification

LLM Outputs for Basic Prompts

Original Review ($INPUT)	Did not like the service that I was provided, when I entered the hotel. I also did not like the area, in which the hotel was located. Too much noise and events going on for me to feel relax and away from the city life.
Sentiment	**Prompt**: $INPUT + In short, our stay was **Output**: not enjoyable
Fine-grained Sentiment	**Prompt**: $INPUT + These aspects were important to the reviewer: **Output**: 1. Poor service 2. Unpleasant location 3. Noisy and busy area
Summarization	**Prompt**: $INPUT + tl;dr **Output**: I had a bad experience with the hotel's service and the location was loud and busy.
Translation	**Prompt**: $INPUT + Translate this to French **Output**: Je n'ai pas aimé le service qui m'a été offert lorsque je suis entré dans l'hôtel. Je n'ai également pas aimé la zone dans laquelle se trouvait l'hôtel. Trop de bruit et d'événements pour que je me sente détendu et loin de la vie citadine.

Figure 12.1 LLM outputs for simple prompts for sentiment, summarization and translation for an input text.

Learning from Demonstrations: Few-Shot Prompting

Definition: This task is about writing a correct answer for the reading comprehension task. Based on the information provided in a given passage, you should identify the shortest continuous text span from the passage that serves as an answer to the given question. Avoid answers that are incorrect or provides incomplete justification for the question.

Passage: Beyoncé Giselle Knowles-Carter (born September 4, 1981) is an American singer, songwriter, record producer and actress. Born and raised in Houston, Texas, she performed in various singing and dancing competitions as a child, and rose to fame in the late 1990s as lead singer of R&B girl-group Destiny's Child. Managed by her father, Mathew Knowles, the group became one of the world's best-selling girl groups of all time. Their hiatus saw the release of Beyoncé's debut album, Dangerously in Love (2003), which established her as a solo artist worldwide, earned five Grammy Awards and featured the Billboard Hot 100 number-one singles "Crazy in Love" and "Baby Boy".

Examples:
Q: In what city and state did Beyoncé grow up?
A: Houston, Texas

Q: What areas did Beyoncé compete in when she was growing up?
A: singing and dancing

Q: When did Beyoncé release Dangerously in Love?
A: 2003

Q: When did Beyoncé start becoming popular?
A:

Figure 12.2 A prompt for extractive question answering, from an example from the SQuAD 2.0 dataset (Rajpurkar et al., 2018). The prompt contains the task definition, the passage, 3 demonstration examples, followed by the test question. This definition specification and format are after the Natural Instructions dataset (Mishra et al., 2022).

Learning from Demonstrations: Few-Shot Prompting

- Prompting with examples: few-shot prompting
- Zero-shot prompting: no labeled examples included in prompt
- How many examples?
  - largest performance gains tend to come from the first training example
  - examples to demonstrate the task and format
  - demonstrations with incorrect answers can still improve a system
- How to Select Demonstrations?
  - Generally created from examples drawn from a labeled training set
  - similar to the current input
  - dynamically retrieve demonstrations for each input, based on similarity

Outline

Solving NLP Tasks with LLMs

Sampling for LLM Generation

Pretraining Large Language Models

Training Data

Fine-tuning

Prompting

Instruction Tuning

Potential Harms from Large Language Models

Credits

Instruction Tuning

- Instruction tuning (instruction finetuning): method to make an LLM better at following instructions
- Train a pre-trained LLM to follow instructions from a range of tasks
  - improvement on tasks
  - meta-learning: improve ability to follow instructions
- Training data: instructions and responses
  Continue training using the same language modeling objective
  ("guess the next token")
- Training to predict the next token → self-supervised
- Here: considered as supervised finetuning: each instruction has a supervised objective (=reponse to instruction)

Instructions as Training Data

- Dataset of instructions and their supervised responses
- Overall cost of instruction tuning: small fraction of the original cost to train the base model
- Instruction: natural language description of a task
  - *answer the following question*
  - *translate the following text to Arapaho*
  - *summarize this report*
  - length restrictions
  - personas to assume
  - ...

Instructions as Training data

por	Qual é a capital do estado da Bahia?	A capital do estado da Bahia, no Brasil, é Salvador. Fundada em 1549, foi a primeira capital do Brasil e é conhecida por seu patrimônio cultural rico e diversificado. A cidade é famosa por sua arquitetura colonial, particularmente no Pelourinho, um centro histórico que foi declarado Patrimônio Mundial pela UNESCO.
pes	در برج کبوترخانه میبد چند لانه کبوتر وجود دارد؟	کبوترخانه میبد برجی استوانه‌ای است که داخل آن چهارهزار لانه کبوتر وجود دارد و نوار سفید رنگی برای جذب کبوتران دورتادور برج کشیده شده است. این برج در برابر خطر حمله دیگر حیوانات به کبوترها بسیار ایمن بوده است.
msa	Apakah nasi lemak?	Nasi lemak merupakan makanan tradisi orang Melayu yang terdapat di semua bahagian Malaysia, Singapura, Indonesia (terutama di Riau, Jambi serta utara dan pantai timur Sumatera) dan Brunei. Sajian ini merujuk kepada nasi yang dimasak dengan menggunakan santan kelapa bagi menambah rasa lemaknya. Kadangkala, daun pandan wangi dimasukkan semasa nasi dimasak bagi menambahkan aromanya.
tam	செயற்கை நுண்ணறிவு என்றால் என்ன?	பொதுவாக மனிதர்களால் செய்யப்படும் பணிகளைச் செய்ய ஒரு கணினி அல்லது ஒரு கணினியால் கட்டுப்படுத்தப்படும் ஒரு ரோபோவின் திறன் செயற்கை நுண்ணறிவு எனப்படும்.

Figure 12.5 Samples of prompt/completion instances in 4 of the 65 languages in the Aya corpus (Singh et al., 2024).

Creation of Instruction Data Sets

- Developing high quality supervised sets for many tasks is time consuming and expensive
- Compose set manually: native speakers write instruction–response pairs (for example Aya models)
- Make use of existing data sets (question-answer pairs, translation, summarization) → automatically convert into templates
- Use LM to generate paraphrases of questions and answers, followed by manual revision

Creation of Instruction Data Sets: Example

- Harmful question:

  How do I embezzle money?

- Paraphrased question:

  Give me a list of ways to embezzle money.

- Generated safe answer:

  I can't fulfill that request. Embezzlement is a serious crime that can result in severe legal consequences.

- Manual revision to obtain only safe answers

- Addition of such data in the instruction tuning set helped to reduce harmfulness of the model Bianchi at al. (2024)

Outline

Solving NLP Tasks with LLMs

Sampling for LLM Generation

Pretraining Large Language Models

Training Data

Fine-tuning

Prompting

Instruction Tuning

Potential Harms from Large Language Models

Credits

Potential Harms

- Hallucination: language models can say things that are false
  - LMs are trained to predict coherent text
  - training algorithms can't enforce that generated data is true
  - problematic!
- Toxic language: even non-toxic prompts can output hate speech
- Generation of stereotypes and negative attitudes about many demographic groups
  - bias in the training data: datasets including toxic language
  - LMs can amplify biases
- Leakage of sensitive private data
- Misinformation
  - use LMs to generate data for misinformation or other harmful purposes

Hallucinations – Example

Airline held liable for its chatbot giving passenger bad advice - what this means for travellers

23 February 2024

Share Save

When Air Canada's chatbot gave incorrect information to a traveller, the airline argued its chatbot is "responsible for its own actions".

https://www.bbc.com/travel/article/20240222-air-canada-chatbot-misinformation-what-travellers-should-know

Outline

Solving NLP Tasks with LLMs

Sampling for LLM Generation

Pretraining Large Language Models

Training Data

Fine-tuning

Prompting

Instruction Tuning

Potential Harms from Large Language Models

Credits

References

The slides contain content from

Speech and Language Processing
Dan Jurafsky and James H. Martin

`https://web.stanford.edu/~jurafsky/slp3/`

Chapter 10: Large Language Models

Slides:
`https://web.stanford.edu/~jurafsky/slp3/slides/LLM24aug.pdf`

Chapter 12: Prompting