Large Language Models

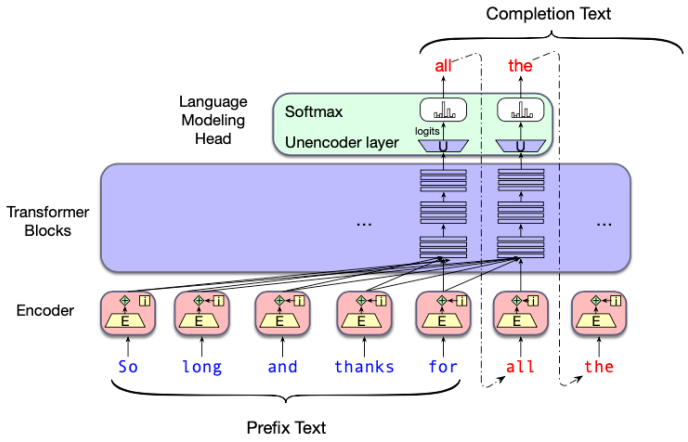
Summary of Chapter 10 from
Speech and Language Processing,
Jurafsky and Martin, Aug. 20, 2024 draft
Michael Wollowski

Some Terms in NN – Batch Size

- If a NN is trained on one item at a time, the weights, may fluctuate.
- Instead, train on a batch of items.
- Run the forward pass several items, the batch, and average the errors.
- Adjust the weights based on the averages of the batch.

Text Completion

- A language model is given a text prefix and is asked to generate a possible completion.
- As each token gets generated, it is added as a prefix for generating the next token.

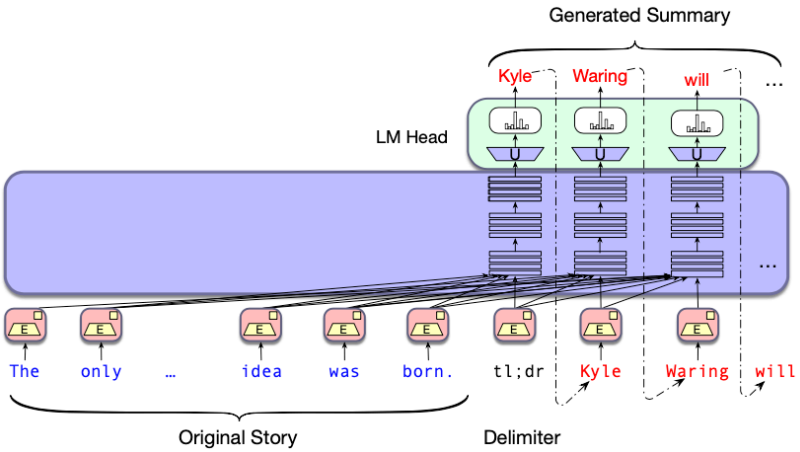


Text Summarization

- In text summarization, we take a long text and produce a summary of it.
- We can cast summarization as language modeling.
- Give an LLM a text, followed by a token like **tl;dr;**
- This token is short for ‘too long; did not read’
- We can perform conditional generation as follows:
  - Give the language model the text and token
  - Ask it to generate words, one by one
  - Take the entire response as a summary.

Text Summarization

- Take a long text and produce a summary of it.
- Give an LLM a text, followed by a token like **tl;dr;**



Sampling for LLM Generation

- Decoding: Which single word should one generate next based on the context and based on the probabilities that the model assigns to possible words.
- Autoregressive generation or causal LM generation is the process of decoding one word at a time.
- The most common method for decoding in large language models is sampling.

Random Sampling

- A simple way is to always generate the most likely word given the context.
- This is called *greedy decoding*.
- Problem with greedy decoding: the words it chooses are predictable.
- The resulting text is generic and often quite repetitive.

Top-*k* Sampling

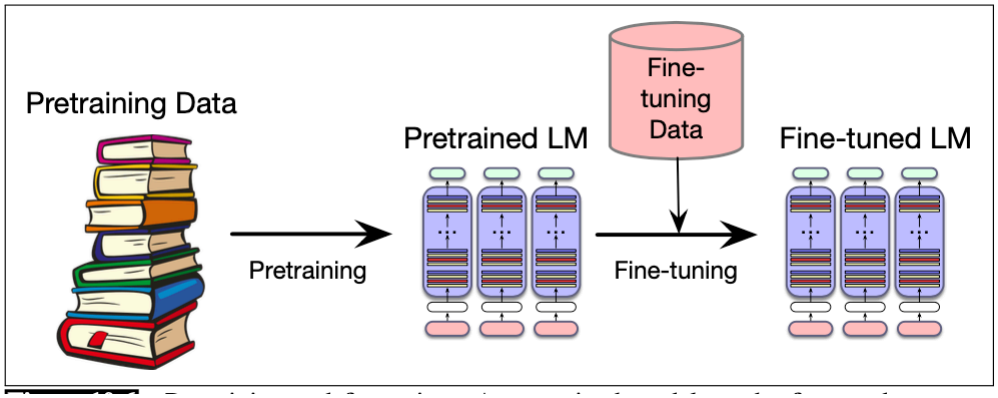
- A generalization of greedy decoding.
- Rather than choosing the single most probable word.
- Truncate the distribution to the top k most likely words.
- Renormalize to produce a legitimate probability distribution.
- Then randomly sample from within these k words.

Temperature Sampling

- Rather than truncate the distribution, reshape it.
- The intuition comes from simulated annealing:
  - at a high temperature, it is very flexible and can explore many possible states,
  - at a lower temperature, it is likely to explore a subset of lower energy states
- In low-temperature sampling, we smoothly increase the probability of the most probable words and decrease the probability of the rare words.

Pre-training and fine-tuning LLMs

- A pre-trained model can be fine-tuned to a particular domain, dataset, or task.



Self-supervised training algorithm

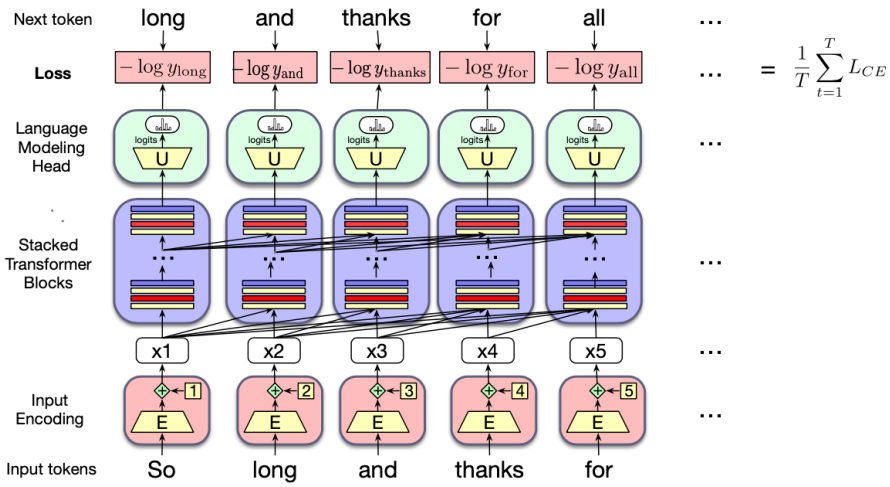
- Transformers are trained on a corpus of text.
- At each time step *t*, we ask the model to predict the next word.
- We call such a model *self-supervised,* because the natural sequence of words is its own supervision.
- We simply train the model to minimize the error in predicting the true next word in the training sequence.
- During training, the probability assigned to the correct word is used to calculate the loss for each item in the sequence.
- The weights in the network are adjusted to minimize the average loss over the training sequence via gradient descent.

Some Terms in NN – Teacher Forcing

- Rather than feeding the model its best case from the previous time step.
- Give the model the correct history sequence to predict the next word.

Self-supervised training algorithm for Transformers

- At each step, given all the preceding words, the final transformer layer produces an output distribution over the entire vocabulary.



Training corpora for LLMs

- Large language models are mainly trained on text scraped from the web, augmented by more carefully curated data.
- Since those training corpora are so large, they are likely to contain many natural examples that can be helpful for NLP tasks:
  - question and answer pairs (for example from FAQ lists),
  - translations of sentences between various languages,
  - documents together with their summaries, and so on.

Training corpora for LLMs

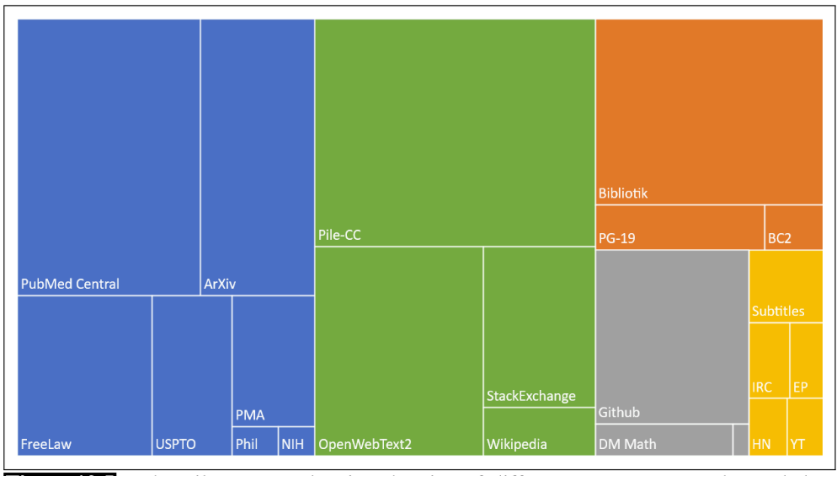
- Web text is usually taken from corpora of automatically-crawled web pages like the *common crawl*.
- It is a series of snapshots of the entire web produced by the non-profit Common Crawl that each have billions of webpages.
- Various cleanups of common crawl data exist.
- One is *Colossal Clean Crawled Corpus* (C4)
- It is a corpus of 156 billion tokens of English that is filtered in various ways.
- Filtering includes:
  - Removing duplicated data,
  - removing non-natural language like code,
  - sentences with offensive words from a blocklist.

The Pile

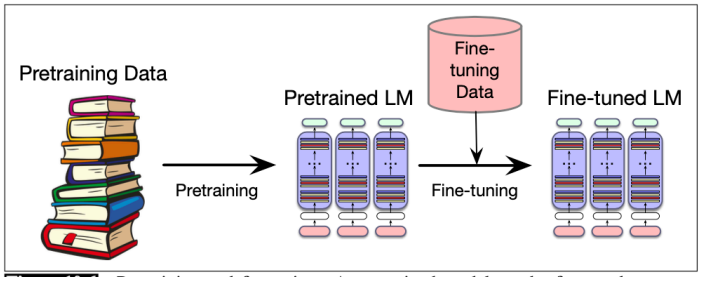
- This C4 corpus seems to consist in large part of patent text documents, Wikipedia, and news sites
- Wikipedia plays a role in lots of language model training, as do corpora of books and code.
- The Pile contains much more varied data.

The Pile

- Colors:
  - Academic
  - Internet
  - Prose
  - Dialoge
  - Misc



Finetuning



- After an LLM has been trained on the large corpus, it can be used outright, such as for ChatGPT.
- However, the general nature of the C4 or the Pile may not have sufficient data to apply an LLM in a specific domain or task.
- Example: a language model that's specialized to medical text
- In such a case, we can continue training the model on relevant data from the new domain or language.

Finetuning

- Although the enormous pretraining data for a large language model includes text
- from many domains, it's often the case that we want to apply it in a new domain or
- task that might not have appeared sufficiently in the pre-training data. For example,
- we might want a language model that's specialized to legal or medical text. Or we
- might have a multilingual language model that knows many languages but might
- benefit from some more data in our particular language of interest. Or we want a
- language model that is specialized to a particular task.
- In such cases, we can simply continue training the model on relevant data from
- the new domain or language (Gururangan et al., 2020). This process of taking a fully
- pretrained model and running additional training passes on some new data is called
- finetuning. Fig. 10.6 sketches the paradigm.