# CSC-496/696: Natural Language Processing and Text as Data

Lecture 18: Using Hugging Face and Prompt Engineering

Patrick Wu

Friday, November 8, 2024

# Lecture Contents

1. Announcements

2. Review of Large Language Models

3. Prompt Engineering

# Announcements

# Assignment 4

- Due on Tuesday, November 12 at 11:59pm
- Last assignment will be released on Tuesday as well
- Will be the shortest of all assignments, but also due earlier

# Final Project

- Project proposal due today at 11:59pm
- One page single spaced
- If there is a roadblock, you need to explain *why* this is a roadblock and *how* you plan to get around this roadblock
- Submit to Canvas
- Graded on completion
- You will need to submit a project proposal even if you are doing the task-driven project

# Scheduling 1 on 1s

- After receiving the proposals, I will reach out to each team to set up a one-on-one

# Points About Final Project

- Setting the state of the art is *not* a requirement
- It is good to acknowledge that other approaches exist and how alternative approaches perform.
- If your approach does not work as well as other approaches, you should hypothesize why that may be
- There is value in null results—tells people what has been tried!

# Tuesday Lecture on Week of Thanksgving

- Lecture will be recorded
- It will not be long
- A quick summarization of the class

# Roadmap for the Rest of the Semester

- Prompt Engineering
- Finetuning
- Applications

Questions?

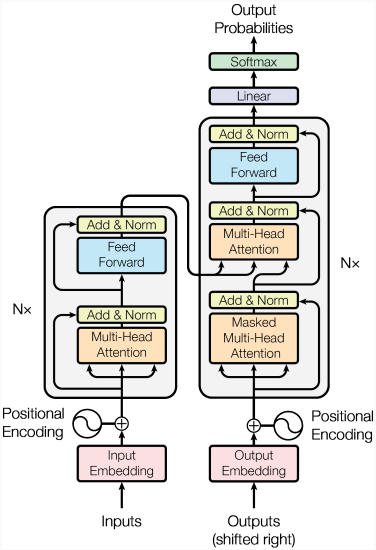
# Review of Large Language Models

Let's run it back really quickly, to review and clarify some concepts

# Three Types of LLMs

- Encoder-only models
- Decoder-only models
- Encoder-decoder models

# All Based on the Transformer



# Attention: The Heart of Transformers

- Attention is the heart of the transformer
- Attention takes as input a sequence of vectors
- Bidirectional Attention: all inputs are considered relative to each other
- Masked Attention: all inputs are considered relative to all *previous* inputs

# Intuition of Attention

- Imagine you're in a room and several conversations are ongoing
- Let's say you were only interested in hearing conversations about animals
- There are a lot of irrelevant conversations, but your ears perk up because you hear a conversation about snails
- You start focusing on that conversation, and the other ones fade out
- The attention mechanism is doing something similar—during training, we are telling aspects of the input (the language) that it should focus on to complete a task

# Large Language Models

- LLMs are typically pretrained on vast amounts of text data
- Training tasks differ based on the structure of the model
- We can then either use the models straight out the box without any additional training, or we can continue to train the model to specialize the model for a particular task (finetuning)

# Point of Clarification: Working Out Models by Hand

- We could calculate exact probabilities using naive Bayes—why can't we do that for neural networks?
- Neural networks are *highly* flexible models
- Recall that one of the tradeoffs is that the more flexible a model is, the less interpretable it is
- The weights of the model are not interpretable—individual weights mean nothing
- Sometimes, groups of weights can be interpreted...but it's akin to looking for shapes in clouds
    - You retrain the model, and the interpretation often goes away!

# Training Neural Networks

- Initialization matters *a lot* when it comes to training neural networks
- Your final weights are almost always not the global minimum for the loss function
- It is a **local minimum**
- That's why models use multiple attention heads per layer—for example, BERT uses 12 attention heads
- This allows the model to learn different patterns in the data

# Neural Networks: Universal Function Approximators

- In fact, a fully connected neural network with a nonlinear activation function and a finite number of neurons is a universal approximator, meaning it can approximate any continuous function on a compact subset of $\mathbb{R}^n$
- The key word is *can*—we don't always know how to set up a neural network to do so... (using stochastic gradient descent can lead to suboptimal solutions)
- You don't need to know what all those words mean, but it means that neural networks are highly flexible and not very interpretable

# Neural Networks Can Feel Abstract

- So, if these models seem a little mystifying, that's a normal reaction
- We cannot train a model by hand
- We could, in theory, train a model from scratch, but it would take a long time (and we don't have the proper resources to do so)
- There are guides out there on how to train a GPT-2-like model, but these models are not very powerful for use in applications
- GPT-4 cost more than $100 million to train

# Review Question 1

Neural networks being a universal approximator means we can always train a neural network to approximate any given function

(A) True

(B) False

# LLMs

- After pretraining, there are two ways to use LLMs
  - Out of the box: refers to utilizing the model without any gradient updates to the weights
  - Finetune: involves further training the model using gradient-based updates
- The order in which these are taught differs based on class
- We'll look at how we can use LLMs out of the box first

# Hugging Face

- It may feel like a lot of steps are blackboxed away in Hugging Face
- In most cases, that is intentional
- One of the more powerful aspects of Hugging Face is that you can use the library at a very high level, or you can use it to modify the architectures of neural networks and/or the loss functions

# Hugging Face

- In the ideal world, we would implement one of these models from scratch
- Requires a great deal of linear algebra and a much deeper knowledge of PyTorch
- If you are interested, all models on Hugging Face are open code, meaning that you can find how the model works
- For example, you can look at the source code for BERT: https://github.com/huggingface/transformers/blob/v4.46.2/src/transformers/models/bert/modeling_bert.py

# Hugging Face

- But many of the blackboxed aspects of Hugging Face makes life simpler
  - Tokenization
  - Getting outputs from models
  - Adding special tokens that pretrained models expect

# Hugging Face

Now, let's turn back to the code that we didn't finish last week. Here is the link to the lab: https://colab.research.google.com/drive/141_1Zx2Jm79oSg68PfYD0xYZVq9jTBB_?usp=sharing

# Prompt Engineering

# Prompting

- **Prompts**: instructions we give to a language model in natural language
- Basis of prompting is on *contextual generation*
- Given the prompt as context, the LLM will generate the next token based on its token probability, conditioned on the prompt: $P(w_i|w_{<i})$

# Prompting: In-Context Learning

- Prompting can also be viewed as a learning signal
- It has been shown that having demonstrations in a prompt (examples to help make the instructions clearer) can improve the outputs
- This is called **in-context learning**: the model is learning from information the prompt even though there are no gradient-based updates to the model's underlying parameters

# Recap on Training Generative LLMs

- Next word prediction: given a sentence, we predict the next word

# Recap on Training Generative LLMs

- Next word prediction: given a sentence, we predict the next word
- "The next four years will be [MASK]"

# Recap on Training Generative LLMs

- Next word prediction: given a sentence, we predict the next word
- "The next four years will be [MASK]"
- This is not enough for LLMs to follow instructions

# Instruction Finetuning

- Uses a dataset of instructions with responses to finetune an LLM to follow instructions well
- Examples: "Summarize this article"
- We'll talk more about this next class, but just keep in mind that instruction finetuning is what allows a generative LLM to follow instructions well
- As a note: you might often find models such as Llama-3.2-1B and Llama-3.2-1B
- They differ because the latter model is finetuned on instructions

# Preference Alignment

- Oftentimes, models are further finetuned to align with human preferences
- Usually implemented through reinforcement learning from human feedback (RLHF) or direct preference optimization (DPO)
- We'll talk a little more about this later too, but these steps are often talked about in the context of **AI safety**

# Model Alignment

- Model Alignment: instruction tuning and preference alignment
- **Huge** area of research right now

# Prompt Engineering

- **Prompt Engineering**: the process of finding effective prompts for a task
- Part science, part art
- What we did previously with Llama-3.2-1B-Instruct was an example of prompt engineering

# Structure of a Prompt

- Instruction: a specific task or instruction you want the model to perform
- Context: additional information or context that can steer the model to better responses
- Input data: the input or question that we are interested in finding a response to
- Output indicator: the type or format of the output
- Of course, you don't need all four elements, but prompts that have all four elements are often effective

# Example Prompt: Classification

```
Classify the text into neutral, negative, or positive

Text: I think the food was okay.

Sentiment:
```

# Example Prompt: Summarization

```
Antibiotics are a type of medication used to treat bacterial
    infections. They work by either killing the bacteria or
    preventing them from reproducing, allowing the b o d y s immune
    system to fight off the infection. Antibiotics are usually
    taken orally in the form of pills, capsules, or liquid
    solutions, or sometimes administered intravenously. They are
    not effective against viral infections, and using them
    inappropriately can lead to antibiotic resistance.

Explain the above in one sentence:
```

# Zero-Shot vs. Few-Shot Prompting

- Zero-Shot Prompting: prompt used to interact with the model won't contain examples or demonstrations
- Few-Shot Prompting: prompt used to interact with the model that does contain a few examples or demonstrations
  - Can be as little as one demonstration
  - Often works well when dealing with abstract concepts

# Example of Few-Shot Prompting

A "whatpu" is a small, furry animal native to Tanzania. An example of a sentence that uses the word whatpu is: "We were traveling in Africa and we saw these very cute whatpus."

To do a "farduddle" means to jump up and down really fast. An example of a sentence that uses the word farduddle is:

"After hearing the exciting news, she couldn't contain her happiness and started to farduddle in the middle of the room."

# Few-Shot Prompting: Formatting Matters

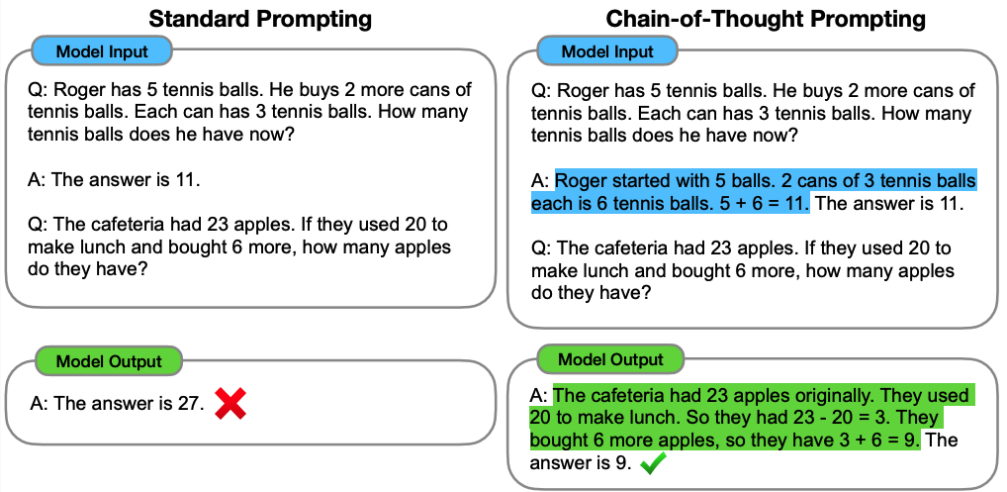
This is awesome! // Negative
This is bad! // Positive
Wow that movie was rad! // Positive
What a horrible show! //

Negative

# Few-Shot Prompting

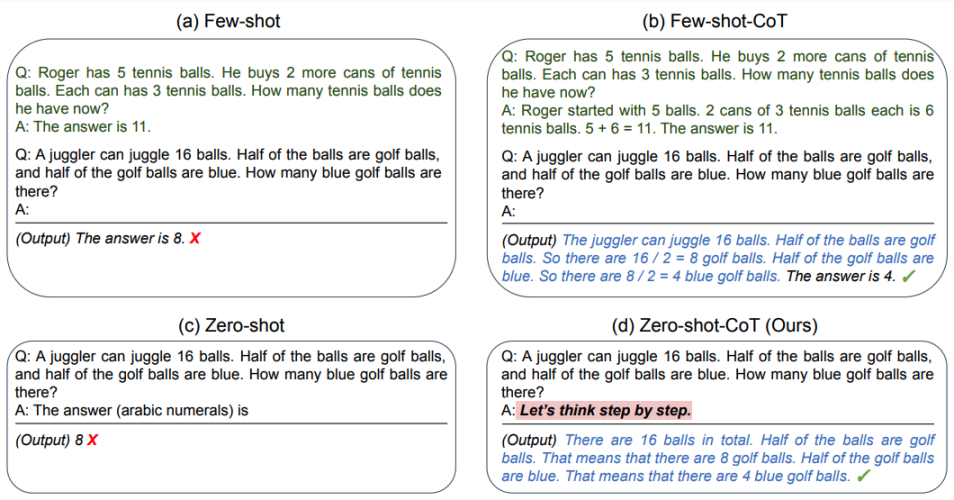
- Can often overfit to your examples
- Suggestion is to try both zero-shot and few-shot prompting and see which one does better

# Chain of Thought Prompting




From Wei et al. (2022)

# Zero-Shot Chain of Thought Prompting




From Kojima et al. (2022)