CS 2731
Introduction to Natural Language Processing

Session 16: LLM discussion and lab day

Michael Miller Yoder

October 23, 2024

University of Pittsburgh | School of Computing and Information

Course logistics

- Project proposal feedback is coming soon (by the end of the week)
- You can store project data at /ix/cs2731_2024f on the CRC (5 TB total space)
- Homework 3 is **due this Mon Oct 28**
  - Make sure you are doing **character-level** language modeling
  - Character-level GPT-2 output might not be great—that's okay
- Homework 4 will be released tomorrow

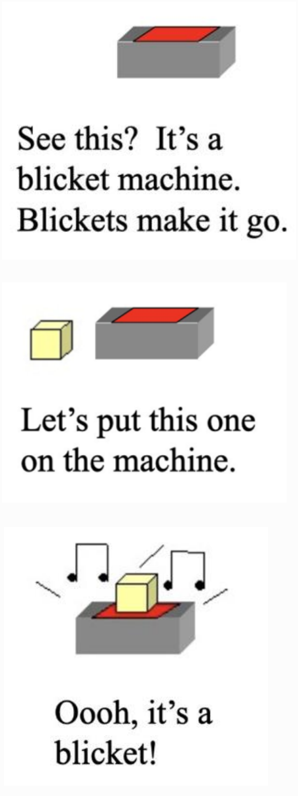
Overview: LLM discussion and lab day

- LLMs as cultural technologies discussion
- Briefly: post-training and in-context learning
- LLM activity: politeness classification with BERT

LLMs as “cultural technologies”

LLMs as “cultural technologies” [Yiu et al. 2023]

- People often debate whether LLMs are intelligent agents
- LLMs can be framed instead as “cultural technologies”: tech that enables transmission of cultural knowledge among people
  - Like earlier technologies of writing, print, libraries, internet search
  - “How you learn what grandma knows”
- Imitation vs innovation
  - Imitation: transmitting knowledge/skills from one agent to another (no notion of “truth”)
  - Innovation: “truth-seeking epistemic processes” that children do
    - Similar to “un-common sense” from Lorraine’s lecture
- Experiments
  - Design new tools (use a hanger to cut a cake)
  - “Blicket detector” to detect novel causal structure


See this? It’s a blicket machine. Blickets make it go.
Let’s put this one on the machine.
Oooh, it’s a blicket!

LLMs as cultural technologies [Yiu et al. 2023]

- What is intelligence? What is creativity?
  - Is finding patterns, like in image classifiers, “AI”? (Hugh)
  - Is creativity more than just rearranging patterns with trial and error? (Joel)
- Innovation and imitation
  - Innovation requires sampling from new distributions and LLMS just estimate existing distributions (Xianglong)
  - Can’t be innovators since don’t understand physical properties of objects that can be used in novel ways. Just know word associations (Maanya)
  - LLMs are getting better at these tough creativity tasks (Joel)
  - LLMs could be trained to be innovators (Kiran)
  - Imitation is often what’s needed for education (Jiyang, Jerry)
  - “Innovation comes from the imitation” (Geonyeong).
  - Really good imitation can be misrecognized as innovation or reasoning (Alex)
- Limitations of LLMs
  - LLMs lack human, physical experience of the world (Rojin)
  - No idea of truth can be very problematic in healthcare, legal advice (Anveshika)
- Relation to commonsense reasoning
  - Humans still succeed over models in edge-case scenarios (Kiran)
  - Hard to learn skills like Einstein’s house problems (Xianglong)

Post-training of LLMs

GPT-3, in-context learning, and very large models

So far, we've interacted with pretrained models in two ways:

- Sample from the distributions they define (maybe providing a prompt)
- Fine-tune them on a task we care about, and take their predictions.

Very large language models seem to perform some kind of learning **without gradient steps** simply from examples you provide within their contexts.

GPT-3 is the canonical example of this. The largest T5 model had 11 billion parameters.

GPT-3 has **175 billion parameters**.

Slide adapted from John Hewitt

Prompting LLMs

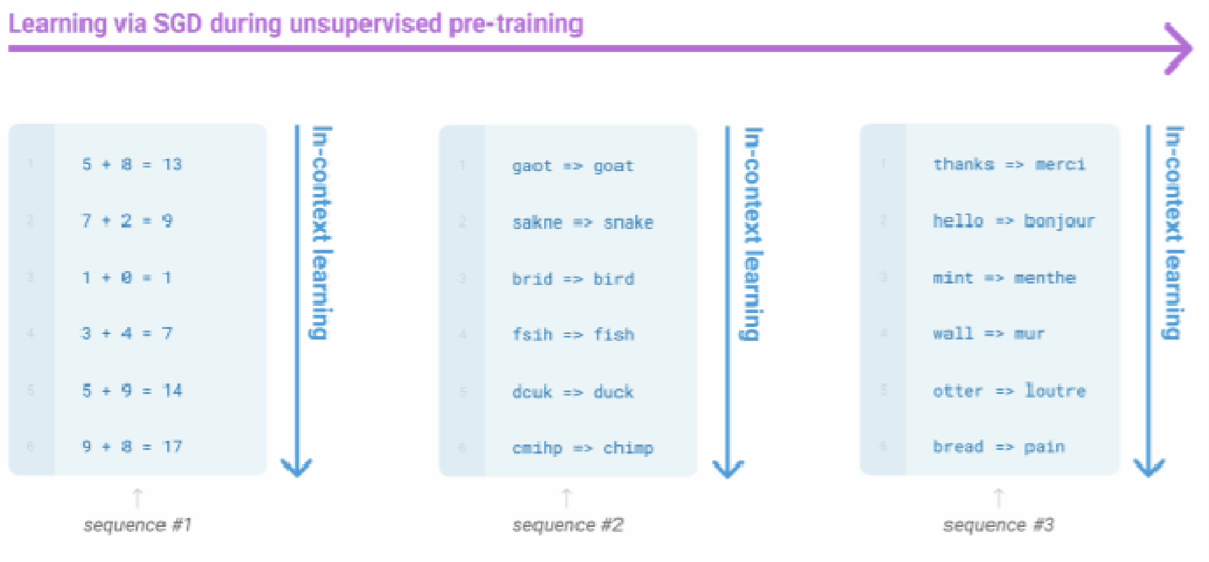
- **Prompt:** text string that a user issues to a language model to get some desired output
- **Prompt engineering:** finding effective prompts for a task
    - Design a template that has a free parameter for the <INPUT> to be filled in over a dataset
        - E.g. <INPUT> + tl;dr for summarization
- **Prompt optimization:** Structured search for prompts with improved performance
    - Usually breadth-first search through a set of paraphrases of a prompt. Then test how well it did against a set with gold labels for the task

In-context learning and very large models

Zero-shot prompting: no examples/demonstrations given

Few-shot prompting: a few examples/demonstrations of the kind of desired output is given in the prompt

Very large language models seem to perform some kind of learning **without gradient steps** simply from examples you provide within their contexts. The in-context steps seem to specify the task to be performed.



Learning via SGD during unsupervised pre-training

```
5 + 8 = 13
7 + 2 = 9
1 + 0 = 1
3 + 4 = 7
5 + 9 = 14
9 + 8 = 17
```

In-context learning

sequence #1

```
gaot => goat
sakne => snake
brid => bird
fsih => fish
dcuk => duck
cmihp => chimp
```

In-context learning

sequence #2

```
thanks => merci
hello => bonjour
mint => menthe
wall => mur
otter => loutre
bread => pain
```

In-context learning

sequence #3

Slide adapted from John Hewitt

Lab activity

BERT for classification

- Skeleton Colab notebook: https://colab.research.google.com/drive/1BkWCnGnuPpKrIKYhqpT6n2mD3CwVRBsb?usp=sharing

1. Train model
   - You get to choose the hyperparameters, which model, etc
2. Evaluate accuracy on the test set
   - Tell Michael your accuracy and he will write it on the board
3. Try other models and training hyperparameters