Lecture 29: Wrap-up!

Julia Hockenmaier
juliahmr@illinois.edu
What have you learned in this class?

What is challenging about natural language?

— Natural language is characterized by Zipfian (long-tailed) distributions: most words, constructions, etc. are very rare

— Ambiguity is pervasive

— A lot of information that is conveyed (and understood) by human speakers is not stated explicitly, but requires additional knowledge and inference
What have you learned in this class?

**Linguistic phenomena:**
- The structure of words
- The structure of sentences
- The structure of discourse
- The structure of dialogue
What have you learned in this class?

Representations:

— **Embeddings** that are induced from data
— **Symbolic representations** that are designed to capture linguistic **structure** (e.g. POS tags, syntactic dependencies) or linguistic **meaning** (e.g. NER labels, semantic roles, rhetorical relations), and the importance of

Models:

— Some statistical NLP and ML models: LMs, HMMs, PCFGs, Naive Bayes
— Various neural architectures
  (Feed-forward, RNN, LSTM, GRU, CNN, transformers)
The billion-parameter gorilla(s) in the room
What else we need to discuss

Why do LLMs like GPT-4 etc. work so well? 
(Do they actually work well? How do we know that?)

How much attention should you pay to discussions about the “sentience” of LLMs or “artificial general intelligence”? 

How much attention should you pay to discussions about the impact of LLMs on education, work, society? 

Is NLP “solved”? If not, what remains to be done?

What ethical considerations do we need to keep in mind when developing or using NLP tools and datasets?
Why do LLMs work so well?
What can LLMs do?

LLMs can generate very fluent and cohesive text

ChatGPT, Bard etc. are LLM-powered chatbots
- Multi-turn conversations with human speakers
- Models get updated based on user input (at scale)

LLMs can be used on various NLP tasks
- Fine-tuning
- Zero-shot, few-shot, etc. prompting
Why do LLMs work so well?

**Size** matters:
Training data, compute, parameters

Caveat for evaluations on tasks with published datasets:
We don’t always know what models have been trained on

**Human-in-the-loop** training methods are very helpful (especially at scale)

But: **LLMs are not perfect**
There is no mechanism in an LLM to guarantee that answers are factually correct
LLMs are prone to **confabulations** (‘hallucinations’)

## LLM sizes

<table>
<thead>
<tr>
<th>Model</th>
<th>#params</th>
<th>training corpus (#words)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT (2018)</td>
<td>~0.34B</td>
<td>~3B</td>
</tr>
<tr>
<td>GPT 2 (2019)</td>
<td>~1.50B</td>
<td>~10B</td>
</tr>
<tr>
<td>GPT 3 (2020)</td>
<td>~175.00B</td>
<td>~499B</td>
</tr>
<tr>
<td>LaMDA (2022)</td>
<td>~137.00B</td>
<td>~1,560B</td>
</tr>
<tr>
<td>GPT 4 (2023)</td>
<td>(???) ~1,000.00B</td>
<td>???</td>
</tr>
</tbody>
</table>

https://en.wikipedia.org/wiki/Large_language_model
Reinforcement Learning with Human Feedback

Reinforcement Learning:
Different from unsupervised and supervised learning
Assume the model to be trained receives a reward when executing an action (or action sequence) during training
Train the model to maximize (expected) reward

Reinforcement Learning with Human Feedback:
Sample different outputs for the same input from a model (LLM)
Ask humans to rank these outputs (for the same input)
Use these human ratings to train a ranking model that captures the raters’ preferences (supervised learning)
Use the ranking model to compute the reward for the model that is being trained to predict outputs (e.g. the LLM)
InstructGPT (Ouyang et al. 2022)

1.3B parameter GPT-3 fine-tuned with RLHF (outperforms 175B parameter GPT-3)

**Prompt types** used to generate samples for RLHF

<table>
<thead>
<tr>
<th>Use-case</th>
<th>(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generation</td>
<td>45.6%</td>
</tr>
<tr>
<td>Open QA</td>
<td>12.4%</td>
</tr>
<tr>
<td>Brainstorming</td>
<td>11.2%</td>
</tr>
<tr>
<td>Chat</td>
<td>8.4%</td>
</tr>
<tr>
<td>Rewrite</td>
<td>6.6%</td>
</tr>
<tr>
<td>Summarization</td>
<td>4.2%</td>
</tr>
<tr>
<td>Classification</td>
<td>3.5%</td>
</tr>
<tr>
<td>Other</td>
<td>3.5%</td>
</tr>
<tr>
<td>Closed QA</td>
<td>2.6%</td>
</tr>
<tr>
<td>Extract</td>
<td>1.9%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use-case</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brainstorming</td>
<td>List five ideas for how to regain enthusiasm for my career</td>
</tr>
<tr>
<td>Generation</td>
<td>Write a short story where a bear goes to the beach, makes friends with a seal, and then returns home.</td>
</tr>
<tr>
<td>Rewrite</td>
<td>This is the summary of a Broadway play:</td>
</tr>
<tr>
<td></td>
<td>&quot;&quot;&quot;&quot;</td>
</tr>
<tr>
<td></td>
<td>{summary}</td>
</tr>
<tr>
<td></td>
<td>&quot;&quot;&quot;&quot;</td>
</tr>
<tr>
<td></td>
<td>This is the outline of the commercial for that play:</td>
</tr>
<tr>
<td></td>
<td>&quot;&quot;&quot;&quot;</td>
</tr>
</tbody>
</table>
How do you get LLMs to perform a particular NLP task?
Zero/One/Few-shot Prompting

Zero-Shot prompting:
Give the LLM a prompt for a task it has never seen before.

One-Shot prompting:
Give the LLM a prompt for a task it has never seen before, and one example of inputs and desired outputs for this task.

Few-Shot prompting:
Give the LLM a prompt for a task it has never seen before, and a few examples of inputs and desired outputs for this task.
Zero/One/Few-shot Learning

Zero-Shot learning:
Give the LLM a prompt for a task it has never seen before.

One-Shot and Few-Shot learning:
Give the LLM a prompt for a task it has never seen before, and allow it to update its parameters based on one (or a small number of) example(s) of an input and a desired output for this task.
Providing prompts that contain reasoning steps that justify the answer can enable LLMs to return output that also describes reasoning steps.
(Relatively large) LLMs can be **fine-tuned** on datasets that contain instructions for a variety of NLP tasks to perform well on unseen NLP tasks.
Instruction Tuning (Wei et al. NeurIPS 2022)

(A) Pretrain–finetune (BERT, T5)

Pretrained LM → Finetune on task A → Inference on task A

- Typically requires many task-specific examples
- One specialized model for each task

(B) Prompting (GPT-3)

Pretrained LM → Inference on task A

Improve performance via few-shot prompting or prompt engineering

(C) Instruction tuning (FLAN)

Pretrained LM → Instruction-tune on many tasks: B, C, D, ...
Model learns to perform many tasks via natural language instructions

→ Inference on unseen task

→ Inference on task A
“Emergent abilities” of LLMs

The performance of increasingly larger LLMs (size = #parameters, #compute, #training data) can often not be predicted by simple scaling laws

Performance often “jumps” as model size increases e.g. Wei et al, 2022 [https://arxiv.org/abs/2206.07682]
Open questions

How well can LLMs reason?

How can we address/identify/prevent LLM hallucinations/confabulations?

How can we prevent LLMs from generating output that is harmful/toxic?
What about "sentience" or "AGI"?
Sentience and AGI

Nothing in an LLM is “sentient” (capable of experiencing feelings/sensations) or “conscious” (aware of its own existence.

There is a lot of hype in public discourse, e.g.

Are LLMs a version of Artificial General Intelligence?
Will LLMs replace humans?

Be aware: Some of this is rooted in transhumanism/longtermism (a school of philosophy with roots in eugenics etc.) — see e.g. the writings of AI ethicist Timnit Gebru
Ethical considerations for NLP

Ethics and NLP

NLP touches on ethics in numerous ways:

— The **data** we work with, natural language, is *produced by (and may talk about) real people.*
  Do we respect the **rights of the individuals** that produced or are mentioned in the data? (privacy, anonymity, etc.)
  Do we understand how the **population of individuals** that produced our data differs from the general population?
  Do we understand what **biases** are inherent in this data?

— The **applications** we develop have more and more **real-world uses** and (unintended) **consequences**.
  We need to be aware of the potential for **benefit** and **abuse**
ACM Code of Ethics

Computing professionals should...

... contribute to society and to human well-being,
    acknowledging that all people are stakeholders in computing
    This includes obligations to promote fundamental human rights,
    and to minimize negative consequences of computing; and to strive for
    environmental sustainability.

... avoid harm
    Harm includes unjustified disclosure of information

... take action not to discriminate
    Technologies should be inclusive and accessible; the creation of
    technologies that disenfranchise or oppress people should be avoided

... respect privacy
    Collection and use of private data comes with responsibilities;

... maintain high standards of professional competence,
    conduct and ethical practice
    This includes awareness of the social context in which work will be used

https://www.acm.org/code-of-ethics
Privacy and copyright concerns

Copyright concerns and plagiarism

Do companies violate copyright/licensing terms by training their models on web data? Should this be seen as violations?

Does an LLM violate copyright law if it generates copyrighted output it has memorized?

Is it plagiarism to use LLMs without attribution?

Do LLMs leak private or confidential information in the training data?

Do LLMs that are updated based on user input leak private or confidential information provided to them?

Can tech companies claim ownership of the data users enter?
Normative vs. descriptive ethics

**Normative** ethics: what we **want** the world to be like

**Descriptive** ethics: what the world **is** like.

Example: Gender bias in NLP:

A coreference system that cannot attach female pronouns to the word “doctor” is both *normatively* and *descriptively* wrong. Racially or gender-based word embeddings are *normatively* wrong (if we don’t want them to be biased), but might be *descriptively* correct (in the sense that they reflect how societies talk about race/gender)

Bias and fairness in NLP
Social Bias in NLP

“Bias” has a number of technical senses in machine learning/stats:
(biased coins, inductive bias, or the bias—variance tradeoff)
This needs to be distinguished from social bias (e.g. gender/racial/class bias, …) that a system’s behavior may exhibit.

Social bias results in… [Barocas et al, Crawford 2017]
… Allocational harms: a system allocates resources/opportunities (credit scores, job ads, goods to buy) differently to different social groups
… Representational harms: a system represents different social groups in a less positive light than others

Identifying social bias is inherently normative

Bias in NLP is a “hot” topic, but a lot of NLP work on bias does not engage deeply enough with the relevant social science literature, or with the communities affected by this bias.

S.L. Blodgett et al. Language (Technology) is Power: A critical survey of “Bias” in NLP
Descriptive bias

Garg et al, PNAS April 17 2018 *Word embeddings quantify 100 years of gender and ethnic stereotypes*
https://www.pnas.org/content/115/16/E3635

Measure the strength of association between words representing social groups (women/men, Asians/Caucasians/…) and words representing professions, attributes, etc.

Embeddings reflect real differences (few carpenters are female, many nurses are), but also track gender and ethnic stereotypes (women are “charming”/“maternal”/…), and their changes over time
Normative bias: NLP performance on AAE

Many NLP tools have poor accuracy on “non-standard” varieties of English that differ from the varieties in common corpora.

For example, toxicity detectors are less accurate on tweets written in African-American English (AAE).

If AAE tweets are deemed more offensive…
… AAE speakers might be more likely to be blocked
… AAE speakers might feel the need to communicate differently than how they normally would
(or not use social media)
… this stigmatization may exacerbate existing discrimination
NLP and Endangered languages

Steven Bird: Decolonising Speech and Language Technologies

Speech/NLP has been used to automate language documentation for endangered (indigenous) languages.

But...

... there is little evidence that documentation saves dying languages
... documentation and the NLP technology are developed by outsiders who don’t engage with the language communities (‘colonizers’), and who don’t understand how language is used in the community, or what tools would be of use to the community.
Why NLP?
Socially useful NLP applications

**Assistive technology** (text-to-speech, voice search, image description for the blind) helps people with disabilities

**Machine translation, summarization, better search engines** all provide unprecedented access to information to the general public

Identifying fake news, trolls, toxic comments can prevent harmful information to spread.

Social media monitoring can also be used to assist in disasters, or to identify health issues

But this can also be abused for surveillance.
What's next in NLP?