Reading: “ChatGPT is a blurry JPEG of the Web.” Ted Chiang, New Yorker.
How do we use language models?

(This lecture is a biased and incomplete summary of what’s happening today in NLP.)
Three Variants of LM-Based NLP

1. LM as **encoder** of text, usually for analysis tasks (not generation); can be used to learn representations where information flows from all input tokens to all others. Typically finetuned with supervised learning. Widely deployed already. Example: BERT.

2. LM as **encoder and decoder** of text. Suitable for sequence-to-sequence tasks. Examples: modern machine translation models, T5.

3. LM as **decoder** of text, suitable for generation tasks (but also analysis); getting a lot of attention now. Example: GPT-3.

For all of these, there are many different training procedures, inference procedures, and evaluation frameworks!
The general trend over the past few years has been larger models (in terms of layers, number of parameters, etc.) pretrained on larger datasets; hence the term “large language model.” While there are new models coming out all the time, there aren’t enough systematic experiments, or consistent evaluations, to fully understand all the factors in LM success.

Open question: Is it possible to achieve results on par with the best seen, but with less data, less parameters, less computationally-intensive training?

I think the data matters most, and quality may be more important than quantity, for some definition of quality. See Chinchilla (Hoffmann et al., 2022) and Llama (Touvron et al., 2023).
The Cost of Scale

Costs (thanks to Luke Zettlemoyer), not including data preparation, comparison runs at smaller scale, architecture, experiments, etc.:

<table>
<thead>
<tr>
<th>Model</th>
<th>Params</th>
<th>Hardware</th>
<th>Days</th>
<th>Est. AWS Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPT-3 (OpenAI)</td>
<td>175B</td>
<td>1500 GPUs</td>
<td>60</td>
<td>$3M</td>
</tr>
<tr>
<td>PaLM (Google)</td>
<td>540B</td>
<td>6144 TPUs</td>
<td>57</td>
<td>$25M</td>
</tr>
</tbody>
</table>

Latest: Llama (Meta), which is somewhat more open.
Finetuning

Recall that, since ELMo, the best performance for LM-based models has usually been obtained by applying supervised learning, initialized by the pretrained LM, on task-specific data.

That’s still mostly true today, but fully finetuning the largest models is beyond most researchers’ budgets. (Research topic: efficient finetuning, e.g., with adaptors; Houlsby et al., 2019.)

The extremely long tail of possible NLP applications has become impossible to ignore, as we observe people interacting with ChatGPT (more about this later). Perhaps we’re moving away from supervised learning on large samples to ... something else.
Prompting

The simplest kind of generation from an LM is to continue from a prompt:

$$\text{argmax}_{x \in \mathcal{V}^\dagger} p_{\text{LM}}(x \mid x_{\text{prompt}})$$

(Various decoding algorithms, from greedy to beam search.)

Over the past five years, the LMs have become increasingly fluent, even under this simple use case.

As a result, considerable research on using LMs essentially on their own to do NLP. Early use case: testing LMs for factual “knowledge” by prompting them (Petroni et al., 2019). How far can we go with prompting?
Prompting

Consider a task in which we want to map inputs to outputs. Some evaluation settings:

- “Zero shot”: encode the input as a sequence, feed to the LM as a prompt, treat the continuation as output. E.g., “Translate from English to French: I’m hungry.”
- “Few shot”: like zero shot, but include a few input-output pairs to the prompt.

This trend was set off by GPT-3 (Brown et al., 2020). People sometimes call this setup “in-context learning,” but this is rather different from conventional machine learning (where parameters get updated, for example).

It’s very appealing because there’s no specialized training required! Once you build your LM, it can do anything.*

*Anything you can encode in a prompt. And not usually as well as a finetuned model where training data is available.
Instruction Tuning

Key idea: finetune a LM to be better at interpreting prompts that include explicit instructions about what you want.

To do this, we need data that includes instructions (in natural language) alongside input-output pairs. (Supervision returns!)

- Example models: Flan-T5 (Chung et al., 2022)
- Example dataset: Supernatural-Instructions (Wang et al., 2022)

Interesting result: can get strong performance with much smaller models (Schick and Schütze, 2021)!
A Parallel Development: Pretraining on Code

Applying language models to code has led to exciting developments in tools for programmers.

Example model: Codex (Chen et al., 2021)

It’s now standard to include some code in LMs trained primarily on natural language text. Some attribute the models’ apparent reasoning capabilities to this (Madaan et al., 2022).
Data, Again

In general, more data leads to better models.

There’s not a lot of discussion about how data is selected, except:

▶ Including source code, as discussed.
▶ Multilingual datasets seem to lead to some crosslingual transfer capabilities.
▶ “Quality filters” are sometimes applied, but these can have negative consequences, too (Gururangan et al., 2022).

In general, the datasets are too big for anyone to fully understand what’s in them.
As best we can tell, ChatGPT starts with GPT-3 (or a similarly strong base), pretrained LM (on text and code). It’s then finetuned to follow instructions (proprietary data, not instructions for classic NLP tasks), then finetuned on human feedback (e.g., learn a reward function from annotator judgments, then apply to the LM through reinforcement learning; Ouyang et al., 2022).
What Kinds of Things Does ChatGPT Do?

- Generates fresh text (reports, letters, stories, poems, etc.) that you request, on any (?) topic.
- Summarizes long text in shorter form.
- Revises text to have a different style.
- Generates code to your specifications.
- Maintains some coherence across iterations (“turns” of a “conversation”).
- If you point out problems in what it gives you, it might apologize and try again. Or antagonize you.
Concerns

Public misunderstanding

“Hallucinations” (bullshit)

Toxic language

Privacy

Data rights
The LM as a Component, Again

Despite the many surprising things models like ChatGPT can say, it’s important to remember that they are fluency machines; they do not model truth, correctness, or respect.

Big trends right now:

▶ Let LM interact with external tools (e.g., a search engine, databases, solvers, ...), including attribution/citation to sources of information.

▶ Relatedly: “nonparametric” aspects to LM design (the model uses a stored corpus it can access at inference time; Khandelwal et al., 2020)

▶ Multilingual language models trained on text from many languages; these show some “transfer” capabilities.

▶ LMs to synthesize data for NLP tasks, with some human input as well (e.g., Liu et al., 2022)

▶ Efficient finetuning (e.g., adaptors; Houlsby et al., 2019)

▶ Ensembles of smaller and/or cheaper LMs (Tim’s lecture)

▶ Multimodality, e.g., text-to-image, image-to-text, ...
References I


