NeurIPS Paper Exploration - LLMs:

Last week, I attended the latter half of NeurIPS, including the generative AI in education (GAIED) workshop on December 15. While there, I compiled a list of main themes in the LLM space, both in the main conference papers and at the workshop. I've identified four main themes / lines of work that dominate the LLM discussion: evaluation, prompt engineering / functional extensions of LLMs, theoretical / conceptual analysis of LLM capabilities, and LLM security.

LLM Evaluation / Benchmarks:

- *PlanBench:* Evaluation on Planning
- *BenchCLAMP:* Syntactic and Semantic Parsing
- *Graph Reasoning Tasks / NLGraph*
- *A second paper on planning evaluation (using International Planning Competition as benchmark)*
- *Deductive Reasoning*
- *Program Synthesis:* HumanEval+
- *Factuality*: FELM
- (Less interesting) *M3Exam:* Evaluating on standard exam questions across languages / with vision.
- (Less interesting) Red Herrings

At a high level, there seems to be a lot of effort being put into evaluating LLMs, with particular emphasis going towards migrating benchmarks for common computer science and reasoning tasks into the LLM setting (e.g., planning problems, program synthesis). This is also reflective of the situation in education, where LLM evaluation is still in its infancy.

Naturally, the simplest paradigm, which dominates this theme, is to establish (input, output) data pairs for test cases, e.g., a problem and the optimal plan, and then compare LLM solutions to this output. Yet, there is a notable lack of new evaluation paradigms for tasks with sparser rewards / less well-defined ideal behaviors, such as education. In my view, areas like reinforcement learning and game-playing agents are precisely the examples we should be building towards in education, where our signals are also sparse and environment-driven. Yet, this line of work (building evaluation environment with meaningful simulated rewards) is not addressed. Until such ideas emerge, the best currently possible in education is to simply apply LLMs as evaluators of open-text interactions.

Prompt Engineering / LLM as subroutine:

- *DDCoT:* Multimodal CoT (reasoning and recognition)
- *LLM as a weak learner in boosting*
- *LLM as heuristic within Monte-Carlo Tree Search*
- *Big LLM "teaching" smaller LLMs*
- *Stimulus prompting:* Fine-tuning a smaller LLM on existing data to provide context/chain of thoughts to feed into larger LLM

In addition to LLM evaluation, a key direction emerging in the space is to combine LLMs with existing computer science approaches, such as search algorithms, to overcome their inherent limitations (hallucinations, lack of search functionality). In particular, some examples I found interesting are using LLMs within boosting frameworks (as a weak learner complementing other learners), applying LLMs as a guide to a tree search, and using larger LLMs as training for a smaller one, to obtain higher returns from smaller models.

In my work, I've seen and helped build approaches incorporating basic logical flows into prompting chains and sequences, e.g., if a student is happy, do this, else do that. This theme looks more into what I refer to as "algorithmic prompt strategies", such as tree-of-thoughts (depth-first-search with LLM evaluation) and graph-of-thoughts. I find this very interesting as it offers a principled way to deal with some LLM limitations while also building on the rich literature of computer science research.

LLM Analysis:

- *Biases and performance of LLMs when simulating personas*
- *Biases in CoT*
- *Emergent capabilities not real, but due to metric non-smoothness* (Best Paper Award)
- LLM memorization (undesirable as sensitive info can be recovered)

Given the huge size of mainstream LLMs and the plethora of claims being made on their capabilities, a key theme arising both at NeurIPS and more generally is the detailed analysis and verification of such claims. Among these papers, one I found particularly interesting, and which ultimately won Best Paper at the conference, takes a deeper look at the claimed "emergence" of capabilities with LLM scale, and ultimately demonstrates with a very simple metric manipulation (replacing hard metrics like accuracy with smoother ones like distances) that performance improvements are gradual and mostly follow the empirically observed power laws. Hence, the paper very much disproves the sharp emergence of new capabilities, instead highlighting that metric choices underpin this phenomenon.

In general, I find this direction extremely relevant as it really is the best alternative to having "nutrition facts" for LLMs. Given the extreme levels of hype surrounding these models and the extreme speed at which development occurs, it is imperative to conduct similar analyses and take LLMs for what they are: sophisticated algorithms with inherent quirks.

LLM Security:

- LLM Trojans (adversarial attack)
- LLM Backdoor Prevention

One key observation I've made early on with instruction-following LLMs is that their main draw is also a fatal flaw: Easier behavior manipulation with prompts also leads the way to easier derailment. Much like good prompts can make LLMs behave more reliably and achieve pre-set goals, a malicious set of prompts can lead to very poor consequences (information leaks, abuse, etc.). As a result, more and more work is being put into preventing malevolent LLM use, but also into identifying new, less known attacks that "jailbreak" existing LLMs. Taken together, both lines of research will be key to making LLMs safer and offer insights that complement the analysis section above.

Overall, both aforementioned sections (analysis and security) further underline how important it is to correctly set the framing for LLM use and restrict direct access to these models. Given the obvious security risks and inherent limitations of LLMs, it is important to set up pipelines that enable indirect and restricted access to LLMs, so as to minimize the chances of these models being led astray.

Beyond these main four themes, I've also explored some other papers that relate to LLM capabilities, but which are more empirical, or are geared to more distant objectives than can be ported to education.

LLM Memory Augmentation:

- LongMem

LLMs on 3D input:

- (Relevant to Symbolic Push) Knowledge graphs to define adapters for vision-language models
- LLM trained on 3D point clouds

On GAIED

On the 15th of December, I attended the GAIED workshop. My main take-aways there were:

- **LLM evaluation is a major pain point:** More than half the papers (to my estimate) had to curate their own datasets (most of which are very tiny) to test out GPT-4/3.5's performance on their tasks. This shows the need for a common push for community-level evaluation resources and a unified definition of key tasks to help prioritize progress in the field.

- **There is a plethora of settings in which LLMs can be useful, but we need to remember to validate this rigorously:** From LLMs for code repair to career counseling and tutoring, LLMs are being considered and applied in lots of settings, and have huge potential to democratize access to education. However, what is less explored, and is key to look into, is how effective they will ultimately be, and whether their scalability will ultimately lead to the learning gains we all hope for.

  In my view, I feel that careful prompt strategy and design building on existing learning systems and interventions at scale can lead to improvements in the engagement levels of students and in the authenticity of conversations when LLMs are used as agents. However, it remains key to validate this empirically by comparing learning gains with standard rule-based interventions. On the other hand, there is a very compelling case for using LLMs towards expediting content creation and curation, so that more compelling content can be produced more quickly. In itself, I think this latter point will prove very important in improving AI+ED systems.

- **There is need for a major community thought exercise on what we need from LLMs:** The panel (and Chris Piech's talk) make it very clear that there are lots of challenges to using LLMs in varying contexts (as tutors, as students, as content creators), and that we need a new approach to assessing these use cases in practice. One idea I find compelling, and which we need to flesh out, is to develop realistic environment *simulators* powered by LLMs, to better emulate learning environments, so as to (i) be able to evaluate LLMs at scale and cost-effectively with less human involvement, and (ii) to drive further improvements more rapidly. In a sense, this would be a "flight simulator" for teaching! To get there though, I think we really need to collectively decide on key milestones and scenarios that best represent the challenges of education, and to coordinate on shared resources and infrastructure towards this goal.

Other thoughts

- LLM Efficiency: Had a dedicated competition for it, and seems to be an important direction towards making LLMs more accessible. I met with an author of a recent paper on LLM low-bit quantization, which is used to run a 7 billion parameter model on a Colab notebook! There were lots of lessons to learn from the competition (setting a better training constraint function to level the playing field, establishing inference-time constraints, using a more relative efficiency measure than an absolute time threshold, etc.), which I can happily elaborate on in relevant discussions.