Communications of the ACM

Opinion

Artificial Intelligence and Machine Learning

The Case for Compact AI

A response to recent largesse of large language modeling material.

By Tim Menzies

Posted Jul 10 2025

Reading the *Communications* March 2025 issue, it struck me how many articles assume large language models (LLMs) are the inevitable and best future path for artificial intelligence (AI). Here, I encourage readers to question that assumption.

To be clear: I use LLMs, a lot, for solo and tactical tasks such as condensing my arguments into this editorial response. But for strategic tasks that might be critiqued externally, I need other tools that are faster, simpler, and whose reasoning can be explained and audited. So while I do not want to replace LLMs, I want to ensure we are also supporting and exploring alternatives.

In software engineering (SE), very few researchers explore alternatives to LLMs. A recent systematic review found only 5% of hundreds of SE LLM papers considered alternatives.[1] This a major methodological mistake that ignores simpler and faster methods. For instance, UCL researchers found SVM+TF-IDF methods vastly outperformed standard "Big AI" for effort estimation (100 times faster, with greater accuracy).[7]

In SE, one reason for asking "if not LLM, then what?" is that software often exhibits "funneling": that is, despite internal complexity, software behavior converges to few outcomes, enabling simpler reasoning.[2,5] Funneling explains how my "BareLogic"[3] active learner can build models using very little data for (for example) 63 SE multi-objective optimization tasks from the MOOT repository.[4] These tasks are quite diverse and include software process decisions, optimizing configuration parameters, and tuning learners for better analytics. Successful MOOT modeling results in better advice for project managers, better control of software options, and enhanced analytics from learners that are better tuned to the local data.

MOOT includes 100,000s of examples with up to 1,000 settings. Each example is labeled with up to five effects. In practice, obtaining labels is slow, expensive, and error-prone. Hence, the task of active learners such as BareLogic is to find the best example(s), after requesting the number of labels.[6] To do this, BareLogic labels $N = 4$ random examples, then:

Feedback

1. Scores and sorts labeled examples by "distance to heaven" (where "heaven" is the ideal target for optimization, for example, weight=0, mpg=max).

2. Splits the sort into $\sqrt{N}$ *best* and $N - \sqrt{N}$ *rest* examples.

3. Trains a two-class Bayes classifier on the *best* and *rest* sets.

4. Finds the unlabeled example X that is most likely *best* via arg $\max_X$ (log(like(*best* | X)) − log(like(*rest* | X)))

5. Labels X, then increments N.

6. If N < Stop, go to step 1. Else return the top-ranked labeled example and a regression tree built from the N-labeled examples.

BareLogic was written for teaching purposes as a simple demonstrator of active learning. But in a result consistent with "funneling," this quick-and-dirty tool achieves near optimal results using a handful of labels, as shown by the histogram on the right side of the figure here, across 63 tasks. Eight labels yielded 62% of the optimal result; 16 labels reached nearly 80%, 32 labels approached 90% optimality, 64 labels barely improves on 32 labels, and so forth.

The lesson here is that achieving state-of-the-art results can be achieved with smarter questioning, not planetary-scale computation. Active learning addresses many common LLM concerns such as slow training times, excessive energy needs, esoteric hardware requirements, testability, reproducibility, and explainability. The accompanying figure was created without billions of parameters. Active learners need no vast pre-existing knowledge or massive datasets, avoiding the colossal energy and specialized hardware demands of large-scale AI. Further, unlike LLMs where testing is slow and often irreproducible, BareLogic's Bayesian active learning is fast (for example, for 63 tasks and 20 repeated trials, the figure here was generated in three minutes on a standard laptop). Most importantly, active learning fosters human-AI partnership.
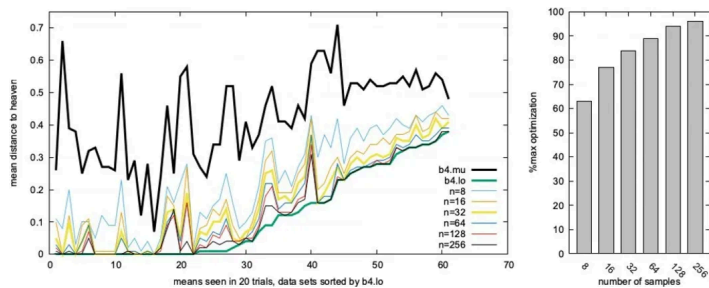

Figure. Twenty runs of BareLogic on 63 multi-objective tasks. Histogram shows mean (1 − (most − b4.min)/(b4.mu − b4.min)). 'most' is the best example returned by BareLogic; 'b4' are the untreated examples; 'min' is the optimal example closest to heaven.

Unlike opaque LLMs, BareLogic's results are explainable via small labelled sets (for example, N = 32). Whenever a label is required, humans can understand and guide the reasoning. The resulting tiny regression tree models offer concise, effective, and generalizable insights.

Active learning provides a compelling alternative to sheer scale in AI. Its ability to deliver rapid, efficient, and transparent results fundamentally questions the "bigger is better" assumption dominating current thinking about AI. It tells us that intelligence requires more than just size.

I am not the only one proposing weight loss for AI. The success of LLM distillation (shrinking huge models for specific purposes[8]) shows that giant models are not always necessary. Active learning pushes this idea even further, showing that leaner, smarter modeling can achieve great results. So why not, before we build the behemoth, try something smaller and faster?

References

References

1. Hou, X. et al. Large language models for SE: A systematic literature review. *TOSEM 33*, 8 (Sept. 2024).

2. Lustosa, A. et al. *More Signal: DRR for Better Optimizations of SE Tasks*. (2025); arXiv:2503.21086 [cs.SE] https://bit.ly/4ljC8oc.

3. Menzies, T. *BareLogic Python Source Code* (2024); https://bit.ly/3FPgoBz

4. Menzies, T. MOOT= Many multi-objective optimization tests (2024); https://bit.ly/4ewuRPP

5. Menzies, T., Owen, D., and Richardson, J. The strangest thing about software. *Computer 40*, 1 (2007).

6. Settles, D. Active learning literature survey. *Technical Report 1648*. U. Wisconsin-Madison Dept of CS (2009).

7. Tawosi, V., Moussa, R., and Sarro, F. Agile effort estimation: Have we solved the problem yet? *IEEE Trans SE 49*, 4 (2023), 2677–2697.

8. Zeming, L. et al. Survey on knowledge distillation for large language models. *ACM Trans. Int. Systems and Technology* (2024).

About the Authors

Tim Menzies (timm@ieee.org) is a professor of computer science at North Carolina State University in Raleigh, NC, USA.

Submit an Article to CACM

CACM welcomes unsolicited [submissions](#) on topics of relevance and value to the computing community.

Join the Discussion (0)