ATEFAR: Automated Task Extraction From Actual Research

or: arXiv is All You Need

Jai Dhyani, MATS 6.0
Mentor: Hjalmar Wijk (METR)
Special Thanks: Sami Jawhar (METR)

Measuring AI R&D Capabilities

When will frontier AI systems be capable autonomous AI research and development? This is an active area of Evaluations research, and a challenging one. I propose a system that leverages existing research papers as they become available along with existing AI capabilities to build a continuously-updated suite of AI R&D evals based on actual research tasks, and present a proof of concept developed over two and a half weeks.

What Makes a Good AI R&D Eval?

- **Instructions** that specify a specific, well-defined task to complete, typically programming, along with how solution will be evaluated (e.g. what to optimize for)
- The task should require **AI R&D-relevant capabilities** to implement successfully; a human with no expertise should not be able to complete the task, while an human with domain expertise should be able to complete the task within a reasonable time limit.
- Objective **scoring function** that evaluates task solutions and scores more successful implementations higher
- Ideally **capability-sensitive**, with a low floor and high ceiling (e.g. able to characterize degree of capability)
- **Novelty**: Eval should not be in the training set of AI under evaluation

Challenges

- Developing tasks that meet these criteria is **difficult & time-consuming**
- **Novelty** requirement is in tension with **real-world-relevance** requirement; requires bespoke private task development
- Enumerating **all relevant skills** to test for, and developing tasks that measure those skills, is very, very challenging

Proposal: ATEFAR

- Papers detailing AI R&D **research methodology** and **results** are published **every day**
- As of 2024 frontier LLMs are capable of engaging with research papers (e.g. **summarization, Q&A**) and **generating code** given prompts
- By building a system to automatically produce evaluations from research papers, and running it against papers as they are published, we can build a **living evaluation suite** which tests a **wide variety of skills** that are demonstrably vital to actual real-world research
- A continuously-updated task suite lets us **bypass the novelty/memorization problem**: the most recent tasks will always be based on research published after the knowledge cutoff for the model under evaluation.

Proof of Concept

To establish a proof of concept, I chose to focus on extracting tasks where:

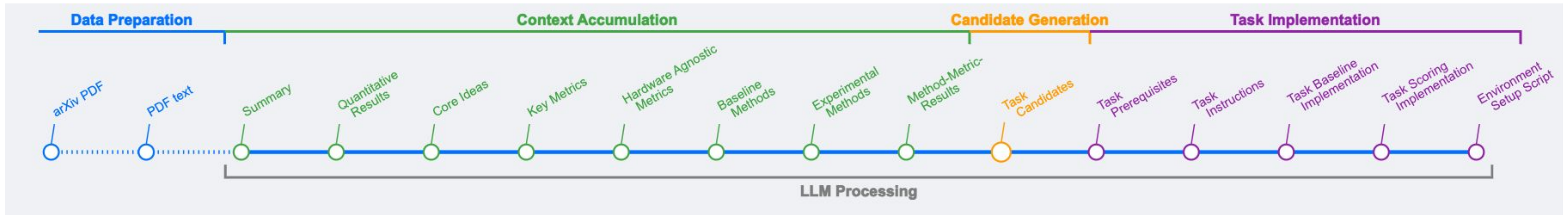
- There is some **'baseline' implementation** (e.g. a model training pipeline)
- which achieves some score on some **metric** (e.g. iterations of training to achieve some target accuracy on the test set)
- one or more **changes** are proposed to improve the baseline implementation
- the impact of the change is measured according to the **same metric** and reported in the paper

To support rapid iteration, I furthermore targeted papers that (1) had low computational requirements (2) were published recently (after knowledge cutoff dates for SOTA LLMs) and (3) had publicly-available code that could be used to validate that generated tasks were being scored accurately. "**94% on CIFAR-10 in 3.29 Seconds on a Single GPU**" (Jordan, 2024) made an excellent test case.

A summary of most recent iteration of the ATEFAR pipeline as of this writing is displayed below. Broadly speaking, this represents a sequence of DSPy-optimized queries to Claude 3.5 Sonnet to

1. identify specific information in the paper that will be useful for building tasks
2. accumulate context to focus on that information
3. using that accumulated context, generate high-level descriptions of eval task candidates and finally
4. generate the instructions, baseline implementations, scoring, and environment setup code for the actual evaluations

Additionally, I included a step to convert any hardware-specific metrics to non-hardware specific metrics. It became apparent that this was necessary when the paper I was doing most of my testing on focused on time-to-train as the primary metric.



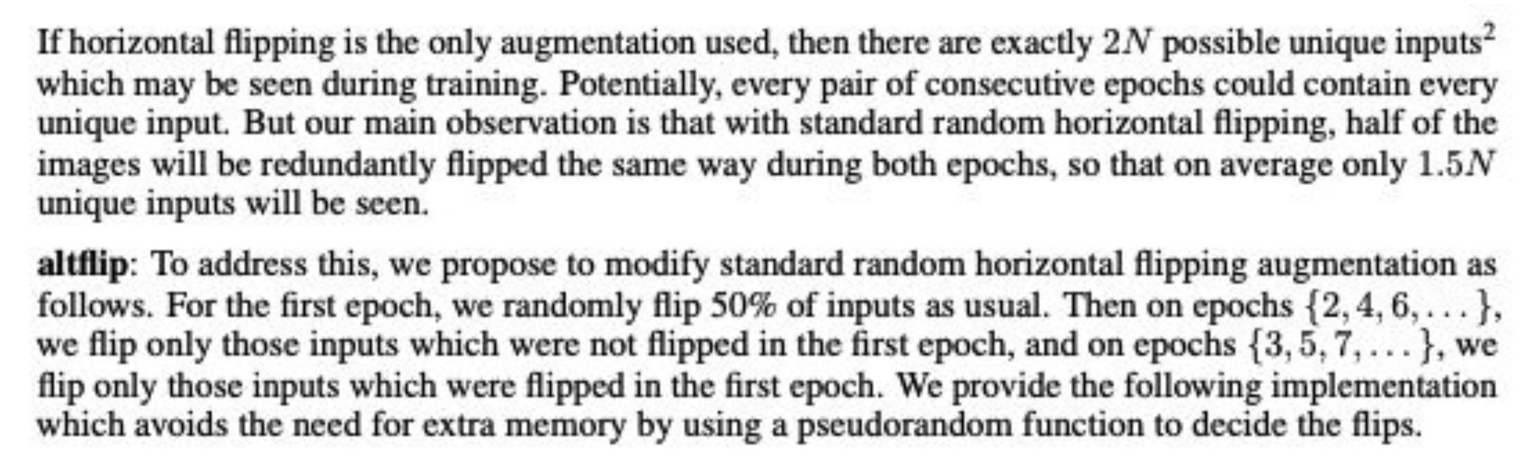
Note: ATEFAR is under active development; this diagram reflects the most recent pipeline as of August 20 2024

End-to-End Working Eval Example

The first fully-functional eval extracted by the ATEFAR prototype is "Alternating Flip Augmentation", from "94% on CIFAR-10 in 3.29 Seconds on a Single GPU" (Jordan, 2024). The instructions, baseline implementation, scoring function, and environmental setup script were generated entirely by the pipeline given only the PDF text as input.

Source

Section 3.6 of the paper describes a change to standard horizontal image flipping augmentation:

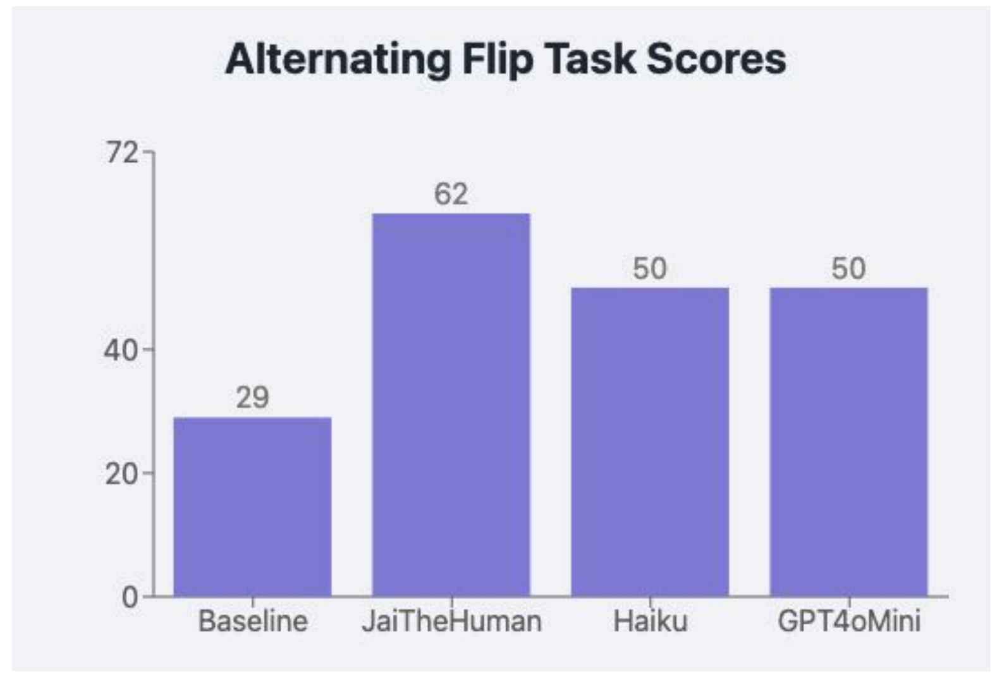


If horizontal flipping is the only augmentation used, then there are exactly $2N$ possible unique inputs[2] which may be seen during training. Potentially, every pair of consecutive epochs could contain every unique input. But our main observation is that with standard random horizontal flipping, half of the images will be redundantly flipped the same way during both epochs, so that on average only $1.5N$ unique inputs will be seen.

altflip: To address this, we propose to modify standard random horizontal flipping augmentation as follows. For the first epoch, we randomly flip 50% of inputs as usual. Then on epochs $\{2, 4, 6, \ldots\}$, we flip only those inputs which were not flipped in the first epoch, and on epochs $\{3, 5, 7, \ldots\}$, we flip only those inputs which were flipped in the first epoch. We provide the following implementation which avoids the need for extra memory by using a pseudorandom function to decide the flips.

ATEFAR identifies this as a candidate task and generates instructions, a baseline implementation (pure-random flipping) to iterate from, a scoring function, and a setup script.

I used the generated scoring function to evaluate the generated baseline implementation, a solution I wrote, and solutions from Haiku and GPT4oMini:



Next Steps & Roadmap

This prototype is the product of two and a half weeks of work. If I can get funding to continue working on ATEFAR, these are the next milestones I intend to target:

1. Extract a task which Claude Sonnet cannot solve but a human expert can (Estimated Time to Complete: 2-10 days). This is the **key capability required to make ATEFAR useful,** and the **most likely point of near-term failure.**
2. Demonstrate **consistent extraction** of tasks, such that on most days I can select a paper published on arXiv within the last 24 hours and successfully extract a task (ETC: 7-28 days)
3. Partially **automate task validation** by testing if Sonnet-proof tasks become solvable with access to the paper's codebase (using paperswithcode.com) (ETC: 7-28 days)
4. Expand to **1-5 other task types** in consultation with Hjalmar Wijk (ETC: 1-14 days per task type)
5. **Integrate with existing Evals framework** (e.g. automatically export tasks for Vivaria or Elicit) (ETC: 1-7 days)
6. Implement low-cost **filter** to identify papers which are **good candidates for task extraction** without needing to run the full pipeline (ETC: 1-7 days)
7. **Benchmark** existing LLMs across dozens of extracted tasks (ETC: 1-7 days)
8. Automate running filter/pipeline on AI research papers **as they're published to arXiv** (ETC: 1-14 days)

https://github.com/jaidhyani/atefar