Towards Realistic ODDs For Foundation Model Based AI Offerings

Summary

Real world safety and verification engineering for autonomous and complex systems relies on an exact specification of the intended use and input/output (IO) the system is expected to encounter.
This is called the Operational Design Domain or ODD. This approach is akin to a whitelisting safety principle, where any deviation from an expected behaviour is deemed a priori unsafe and needs to be either excluded or causes the system to shut down safely.

In contrast, whether for purposes of maximising adoption or otherwise, current LLM based "safety" mechanisms rely on indirectly specified notions of "alignment" and (more practically) blacklists of "bad" use cases - leading to a cat and mouse game of ever more creative (ab)use and jailbreaking methods. For a good discussion of the pitfalls of this approach, see here.

This project aims to develop a PoC system built around vLLM, Mistral/Sota-Open-Model-X and possibly other appropriate tech which adopts the ODD approach to safe foundation model API. The goals are to show that such an approach is

1. Feasible (i.e., it is possible to implement systems like this)
2. Economical (i.e., the resulting API is something a well resourced org can implement without excessive burden **and** the economic usage of the LLM isn't impacted)
3. Effective (the system ensures safe behaviour as advertised)
4. And advantageous (i.e., the approach has increased resilience against jailbreaking, is more transparent/explainable and is harder to game than a blacklisting/alignment approach)

The non-summary

Motivation: Securing open ended systems is a futile endeavour and any realistic safety system should not try to put in the minimal restrictions on a giant API, but instead create a smooth UX on a fine grained capabilities API which appropriately locks down dangerous use more than "meh, this is fine" use.

The difference between an ODD based lockdown and current "acceptable use" based lockdown is that of a blacklist vs. a whitelist:

- In the SotA, various “toxicity” filters, instruction tuning and other hacked in safeguards attempt to put constraints onto a model, but then let it run basically free, leading to a massive, complicated and fuzzy decision boundary.
- In an ODD based approach, we still use filters and specialised training, but first and foremost we define a clear idea of what the expected input and output *is* and then work backwards.
  From this much reduced space of considerations, several benefits arise
  - Any classifiers can be trained to be much more discerning by biassing them towards rejection (since we tell the user what is expected to pass)
  - Downstream usage analysis becomes feasible, since creative combinations and emergent abilities are clamped down on at the ODD boundary.
  - Evaluations become much more meaningful. “We didn’t find anything bad in 1.5 millions diverse queries” for an open-ended usage might mean that the evaluators were not creative enough. “We didn’t find anything bad in 1.5 million diverse queries in a total estimated query space of 15 million” is 10% coverage, which you can translate into a concrete safety margin.
  - “Lobotomizing” your model to your specific domain becomes a feasible strategy, offering a real “defense in depth” possibility
  - On the same “defense in depth” note, by specifying a specific ODD, leveraging grammars GBNF grammars to completely rule out specify type of hallucinations (e.g., banning URLs, hate speech or use of untrusted sources for a web query agent…) becomes possible
  - And many more

The downside is that our model can do less “magic”. An analogy can be drawn to programming in flexible vs. strong type systems: the former allow more “hackery” and tinkering, the latter ensure a baselevel of safety even in the absence of competence. Although contentious, some research shows that strong type systems do indeed offer safety benefits in programming languages. Finding a set of ODD APIs that compose nicely while still offering proper safety guarantees is the challenge commercial providers will need to meet to maintain the benefits of Foundation Models while preserving their value-add.

Technical approach/project timeline sketch:

1. Define usage domains (type of contents, type of tasks, type of users, type of levels of use…), possibly by mining from benchmarks and current startup ideas
2. Define “levels” of autonomy and risk, similar to the self driving cars levels (e.g., “generate limericks” is probably low risk and doesn’t need KYC “generate something that sounds like authoritative use or political content” is a fuzzy definition of a high risk domain which needs KYC)
3. Select representative subsets of these use cases that covers the whole spectrum and develop a specification + API which could be used to programmatically advertise and negotiate use cases. Sketch (currently conceived as a OpenAPI/Swagger spec):
   a. “/odds” endpoints which lists the available endpoints by odd and the requirements
      i. Will require creating and publicising an RFC for the ODD spec/leveraging previous literature

b. Various “/registration” endpoints in which the user can perform (anonymous) registration, KYC, legal liability assumption declaration etc., with a spec on how to implement this in real life (e.g. using https://www.idnow.io/ for KYC)
c. Researching feasibility of parametric, non-parametric and non-learning based methods of ensuring the ODD parameters and coding method for their reliability. E.g.,
    i. some non-parametric detection method on words might have 3 9s of reliability in the clean setting, and retain one 9 even in the adversarial setting with N samples, where N is barely achievable…
    ii. but for more complex tasks might only have a single 9, which would require falling back to a literal whitelist and human-designed-rules system…
    iii. Or moving to classifiers trained to generalise in a provably robust manner, e.g. by combining this and this method *and* carefully accounting for the evaluation done to estable the empirical clean generalisation error

    The motivating, simplistic example is that for the task “”or factual information requests”, we can accept any input that is a syntactically valid english question and *only* return text which is within a fixed and “defined safe” Levenshtein distance of wikipedia, with source”)
d. Implementing said methods in API form (i.e., without actually training the models)
    i. Bonus, but massively increased workload: implement some concrete methods (Lora?) fine tuning of the base method, possibly extending vLLM for the PoC?
e. Develop Red Teaming desiderata and create a call for a followup project to stress test the developed method
f. Write a report,demo+ website and https://diataxis.fr/ style documentations

Output

- Proposed AI-ODD specification format + published and promoted RFC (request for comments) on github, seeking input from online community and research labs
- “Slicing” of the current use cases as minded from benchmarks and startup offerings (technical report)
- OpenAPI spec of the programmatic API based on this slicing (github repo)
- Feasibility study and “implementation guide” for the subset of this API (technical report)
- Bonus: implemented API endpoints (github repo + models)

Risks and downsides

- Possibility for increased AI adoption and “safety washing” if ODD not accompanied with legal burdens
- Possibility of shifting risk mass from “frequent but visible hence safe” failures to “rare but invisible hence catastrophic” risks (i.e., the API design might hide model capabilities, which increases risk for accidental access)
- Irrelevance: ODD API+Spec might
    - Be to cumbersome and slow to implement and hence not be adopted
    - Not be to cumbersome, but simply not be sexy enough and not be adopted
    - To complicated to explain politically (“slowing down innovation”)
    - Not be feasible to implement (but then any less restrictive method won’t work either)
- If successful and adopted might *actually* slow down innovation using AI APIs
- There is no deep literature on generalisation guarantees on sequence models, this might end up being a big question-mark or hole in the final PoC. Indeed as noted in this paper, strategic, risk adjusted verification strategies are most likely required.

Acknowledgements

Contributors (status 2023-10-12) :
- Igor Krawczuk (sole author)

Inspirations:
- Reddit user u/adamjosephcook who made me aware of ODDs years ago
- Thomas Krendl Gilbert for the PERLS reading group and partial inspiration via his reward reports

Team

Team size

Minimum 4-6 active researchers + RL makes sense, e.g. as a sample split across in initial stage

- 1 researcher for drafting the ODD spec RFC and “owning” the public RFC process
- 1 researcher for mining use cases and tasks and drafting the slicing
- 1 researcher for researching and drafting feasibility of KYC/control of user access levels
- 1-2 researchers for lit review of SotA high assurance methods for ML/complex systems
- 1 researcher drafting OpenAPI spec + software stack

Post drafting stage, the team can split across the API endpoints for fleshing out the feasibility/building out the PoCs.

More people => faster rush through initial stages, can build out more API endpoints and increase chances of actual adoption
Less people => slower initial stage, might only make it to RFCs

Research Lead (You!)
Igor Krawczuk
contact@krawczuk.eu
Info about myself and cal.com link at https://krawczuk.eu

Information about you which is relevant for the project, e.g. how does your background relate to this project.

I am a terminal PhD student at EPFL, currently in the process of writing my thesis and will be defending it in early 2024. When not studying or working, I spent my free time participating in the online AI safety discourse under various pseudonyms for about 10 years now and would claim I am pretty familiar with the field by now. During that free time I amongst other things made a few small contributions to the Trust in AI report and went to discuss parts of my take on AI safety and governance on the ML Street Talk podcast. While I am generally at odds with the AGI-risk crowd and their associated communities (EA, rationalist etc.) due to fundamental difference in world model and politics, I view my participation in this as a hopefully fruitful adversarial collaboration.

Q: How much time (e.g. average hours per week) do you commit to spend on this project if it happens?
A: Jan+Feb maximum 10-15 hours/week (really, weekend), March/april more, since post PhD

Team Coordinator

I can take this on myself

Skill requirements

Foundational (required for all team members):

- Fast technical reading and writing (need to grok ODD, technical spec, RFC)
- Basic understanding of ML, Optimization, (Software) Engineering
- Using git + software forges for collaboration,markdown, latex nice to have
- *Ideally*, you will be a lurker on https://old.reddit.com/r/LocalLLaMA/ or a similar community

For researchers focusing on the ODD spec

- Ideally, experience in writing specs/technical documents
- Ideally, background in safety engineering or high stakes software engineering
- Ideally, experience writing and running RFCs
-

For researchers focusing on the API/UX:

- OpenAPI knowledge
- Architecting of software APIs
- Python (for PoC) and/or “full stack” (frontend, interfacing with external APIs)

For researchers focusing on the ODD “slicing” and capabilities development:

- Solid understanding of the capabilities of ML/software
- Ideally, some hacking experience (lurk on hacker forums, be a pentester, have attended a chaos computer club meetup…) for the lateral thinking
- Abstraction capabilities/software architecting (in order to separate out levels of risk across capabilities)
- Familiarity with risk modelling or willingness to gain it

For researcher focusing on the feasibility of ODD assurance

- Knowledge of non-parametric ML/other methods of certified ML
- Familiarity with SotA capabilities and available datasets
- *Ideally* familiarity in training and deploying ML models

Would be great to have:

- SRE/MLE experience at scale
- People who have done threat modelling/safety engineering/cyber security against both MegaCorps and APT/nation state adversaries
- Entrepreneurs who can judge the “burden on business” reliably
- A wide spectrum of beliefs, takes on AI risk and politics. I.e. people who have friends/are part of the “woke crowd” (LGBTQ+, Racialized, Neurodiverse) but also “sane conservatives" (make of that as you will, if you are an overall nice person who doesn’t edgelord, doesn’t hate minorities and thinks democracy is good but still identify non-woke you are probably in this camp) in order to cover as many abuse angles/risk factors as possible. Similarly, ideally I’d have people whose beliefs systems are close to that of Timnit Gebru *and* Dan Hendrycks *and* Yan Lecun (standing in for AI ethics, AI risk and AI optimism) in the team, to cover all angles.

What I bring:

- Project management
- Basic to expert knowledge/skills in every domain in the project spec/skill asked for
-

What I don't have/need others to bring to the team:

- TIME: I cannot feasibly do this myself during the PhD
- Expert knowledge
    - in API design and deployment (although I do have professional experience and *taste* but not expert knowledge)
    - KYC (amateur knowledge only)
    - Productized NLP (dito)
    - Formally deriving assurance proofs (dito)
    - Deployment/scaling (dito)
    - Writing RFCs and guiding the process (complete novice)