ISSN 2186-7437

NII Shonan Meeting Report

No. 208

Trustworthy Machine Learning System Engineering Techniques for Practical Applications

Paolo Arcaini
Zhi Jin
Lei Ma

October 14–17, 2024

National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-Ku, Tokyo, Japan

Trustworthy Machine Learning System Engineering Techniques for Practical Applications

Organizers:
Paolo Arcaini (National Institute of Informatics, Japan)
Zhi Jin (Peking University, China)
Lei Ma (The University of Tokyo, Japan / University of Alberta, Canada)

October 14–17, 2024

Machine learning has experienced a fast boom over the past decade, which enables us to implement complex tasks that cannot be easily solved with traditional software systems, including those in safety-critical domains such as autonomous driving and healthcare. The recent report of JST forecasts a 60 trillion JPY market from AI applications in industry, in particular in domains such as transportation, healthcare, finance, etc., among which many sectors and applications require quality and reliability of the machine learning-based systems to be medium to high level. Such widespread adoption of AI needs proper system engineering support and toolchains. Both the research community and practitioners need to consider what would be a suitable next step and how to achieve trustworthy AI systems with development and use of systematic engineering tools. The increasing importance of engineering is also witnessed by the 2022 survey of McKinsey that reports that software engineers emerged as the most hired AI role over the past year.

The development paradigm of ML-based systems is fundamentally different from the one of traditional software systems. While traditional software systems are often implemented by means of rules (such as program code) that define system behaviour, machine learning systems adopt a data-driven development paradigm, where the decision logic of the software is automatically or semi-automatically learned from continuous aggregation of relevant data. Therefore, traditional software engineering techniques cannot be directly applied as they are, and they need to be adapted to handle the unique characteristics of ML-based systems from a wide range of engineering perspectives. Requirements engineering, for example, needs to be adapted to consider that requirements are satisfied in a quantitative way (e.g., a given precision level) and that some level of misbehaviour is inevitable in an ML-based system; therefore, we need new methods to specify data requirements, including those to flexibly capture uncertainty' requirements considering the need to address the uncertain nature of ML systems. Functionalities of the ML models depend on the training tasks and the training data; therefore the needed training tasks and the distribution of training data, including requirements for continuous sample, will need to be addressed. Fault localization must provide new definitions of faults for ML-based systems, and testing must define new criteria for exposing and correcting these faults. This kind of debugging and repair, typically based on explainability of ML systems, must consider the unstable nature of ML systems in which minor changes have a large effect on the system behaviour. In addition, as the key asset and artefact of ML-based systems, the data must also be carefully managed with more rigorous quality assurance and engineering techniques, with the records of its relation to the code and ML models.

Moreover, since total correctness is not possible, accuracy of requirements of ML-based systems should be prioritised differently, by considering the hazard that could occur if a specific requirement is not satisfied. These priorities will affect the whole development process, from training, to testing and debugging, across multiple development stages of ML-based systems.

Introduction

The goal of the meeting was to bring together software engineering and AI experts from academia and industry, featuring and taking a special focus on the engineering side of ML-based system, to discuss how to design new engineering practices, especially for ML-based systems that would allow to effectively engineer an ML based system in a more trustworthy way. The long term goal is to make the activities across the whole development lifecycle of ML-based software engineerable (Engineerable AI) as it is nowadays for traditional software. Machine learning system engineering community fastly grew in the past few years with some early stage results by researchers around the world.

The meeting was a great chance to gather the world-leading researchers and industry practitioners who achieved results over the past few years. We could exchange state-of-the-art ideas, techniques, and promote such an important research direction and its potential industrial applications, together conquering the currently urgent demand of trustworthiness of ML-based systems.

Overview of the meeting

The meeting started with an introduction from the organizers, describing the topic of the meeting, the overall plan, and the expected outcome of the meeting.

Then, during the first two days, all the participants gave a talk on their research topics, by highlighting open problems and possible future directions. These talks served as a basis for the discussion in the focus groups during the last two days.

Four main topics were identified for discussions:

1) Requirements of ML-based systems;

2) Neuro-symbolic foundations;

3) Quality assessment via testing; and

4) Quality improvement via automated repair.

In the following, we first report the meeting schedule, and then provide an overview of the talks. Finally, we report a summary of the discussion of each focus group.

Meeting Schedule

Check-in Day: October 13, 2024 (Sun)

- Welcome Banquet

Day1: October 14, 2024 (Mon)

- 9:00-10:00. Opening
- 10:00-11:00. 2 Presentations & Discussion
- 11:00-11:30. Break
- 11:30-12:00. 1 Presentation & Discussion
- 12:00-13:30. Lunch
- 13:30-15:20. 3 Presentations & Discussion
- 15:20-15:50. Break
- 15:50-17:45. 3 Presentations & Discussion
- 17:45-18:00. Wrap-up of first day
- 18:00-. Dinner

October 15, 2024 (Tue)

- 8:55-9:00. Opening
- 9:00-10:30. 2 Presentations & Discussion
- 10:30-11:00. Break and photo shooting
- 11:10-12:00. 1 Presentation & Discussion
- 12:00-13:30. Lunch
- 13:30-15:40. 4 Presentations & Discussion
- 15:40-16:15. Break
- 16:15-18:00. 3 Presentations & Discussion
- 18:00-. Dinner

Day3: October 16, 2024 (Wed)

- 9:00-10:30. 4 Presentations & Discussion
- 10:30-11:00. Break
- 11:00-12:00. Definition of an initial set of topics for the discussion in the focus groups
- 12:00-13:30. Lunch
- 13:30-. Excursion and Main Banquet

Day4: October 17, 2024 (Thu)

- 9:00-10:30. Discussion of the selected topics in the focus groups
- 10:30-11:00. Break
- 11:00-11:55. Presentations of the work of the focus groups
- 11:55-12:00. Wrap up
- 12:00-13:30. Lunch

Overview of Talks

AI-driven data diversification

Cyrille Artho, KTH Royal Institute of Technology, Sweden
Joint work with Mojtaba Eshghie, KTH Royal Institute of Technology, Sweden

Many AI-related projects suffer from limitations in their data sets. A few representative examples exist, but it is difficult to generalize them.

If examples are generated without guidance, the "ground truth" of generated examples is often based on existing tools, which leads to imprecise verdicts. If examples are perturbed by rules, we can preserve their specific characteristics, but the type of modified data we can generate maybe limited (due to limitations in the rules that modify the original examples).

We propose a new approach that uses large language models to modify examples and apply this idea to vulnerable smart contracts. A formal specification ensures that key characteristics are not changed. Experiments show that this approach is successful in growing the size of small data sets tenfold.

This presentation is a view on the general problem that underlies our contribution to ASE 2024 NIER titled "Oracle-Guided Vulnerability Diversity and Exploit Synthesis of Smart Contracts Using LLMs".

Safe AI

Lionel Briand, Lero Centre, University of Limerick, Ireland

Beyond Ego Vehicle Testing

Alessio Gambi, AIT – Austrian Institute of Technology, Austria

Current approaches for automatically testing Autonomous Driving Systems (ADS) in simulations generate diverse and challenging driving scenarios. However, they are fundamentally limited because (i) they challenge a single ego vehicle at the time using pre-programmed Non-playable Characters (NPC), (ii) use (existing) flat maps, and (iii) do not include humans either during test execution or test generation. Testing the ego-vehicle against only pre-programmed NPCs risks generating many irrelevant and non-bug-revealing test cases. Using only (existing) flat maps neglects to test the AVs under diverse environmental conditions. Finally, generating scenarios using meta-heuristics might not match developers' expectations.

In this talk, I presented recent results pushing state-of-the-art ADS Testing beyond ego vehicle testing, including a novel approach to test ADS interactions (joint work with Paolo Arcaini, NII, Japan), the Flexcrash Platform [Gambi et al. ISSTA'24, Tool-Demo] to study live interaction between humans and ADS (joint work with students at the IMC University of Applied Sciences Krems, Austria), and an approach leveraging Large Language Models (LLM) to generate three-dimensional virtual roads from descriptions in natural language. This research was partially funded by Flexcrash (Horizon Europe programme, grant agreement No. 101069674) and A-IQ READY (Chips Joint Undertaking, grant agreement No. 101096658).

Reducing the risk of untrustworthy AI applications

Randy Goebel, University of Alberta, Canada

A significant volume of research on natural language understanding and processing ("NLP") has considered how to accurately transform natural language to formal logics, in which case the problem of text entailment becomes that of logical entailment.

While there have been a variety of approaches to the transformation of language to logic, even the most sophisticated work (e.g., Montague's higher order intensional logics or Steedman's combinatory categorial grammar) leaves foundational challenges of context and dialogue at least unresolved. And though the transformations are tightly coupled with formal mechanisms for inference, those methods themselves are often difficult to implement.

So given progress in textual entailment (e.g., JURISIN Competition on Legal Information Extraction/Entailment (COLIEE) 2014), there naturally arises the trade off between depth of transformation versus performance on entailment models. Most of the current work on text entailment focuses on somehow building or learning possible entailment models, which can exploit component NLP models of verb cases, concept extraction, and open information extraction. But how far one can go is not clear, especially as there seems to be a wide variety of entailment tasks.

We consider some parameters on entailment models, and while short on answers, consider some measure alternatives for considering the tradeoffs, and whether they are fundamental empirical, or can exploit some foundational principles of NLP representation theory.

Explanation in ML and Its Reliability

Satoshi Hara, The University of Electro-Communications, Japan

Explainable AI technologies are studied extensively in the past few years. In this talk, I will present a brief overview of what is "explanation" for machine learning, and some representative researches in recent years. In particular, I introduce some reliability concerns regarding "explanation" in machine learning.

The reliability concern can be divided into two categories, technical reliability and social reliability. The technical reliability is on the technical validity of explanation, for example, whether some saliency map algorithms are really explaining the behavior of the models. The social reliability is on the possible negative social impacts of the explanation technologies, for example, whether the explanation algorithms can be used for malicious purposes.

Alignment of ML Performance for Safety in "Engineerable AI" Project

Fuyuki Ishikawa, National Institute of Informatics, Japan

Traditional safety analysis has considered case-by-case argument for avoidance of severe hazards caused by different types of triggers and causes. There is a gap between this principle of safety and the characteristic of machine learning to build a holistic model with the whole dataset containing a variety of cases. Specifically, deep neural networks (DNNs) consisting with millions of parameters are difficult to tune for fine-grained requirements for each specific case.

In the automotive domain, perception functions often depend on DNNs for automated driving and advanced assistance. We should apply the case-by-case analysis for specific types of risk, e.g., hitting a person in front of the car by false-negative detection. This kind of analysis leads to many fine-grained safety metrics over each of specific situations and specific error types. However, it is difficult to tune DNNs for the multiple metrics. It often happens that re-training after finding unsatisfactory performance has little controllablity as they "shuffle" enormous number of parameters in the neural network.

In our eAI ("Engineerable AI") project, we have been investigating a technical approach to performance alignment by analyzing, localizing, and fixing causes of critical errors, such as responsible neuron weight parameters. In this talk, we will introduce different methods following this approach, including DistrRep (ICST 2023) and NeuRecover (SANER 2022), and report experimental results with our industry partners.

A Software Architect's Perspective

James Ivers, Carnegie Mellon University, USA

AI-enabled systems differ from many traditional software systems in their reliance on software elements (AI models) whose results are characterized in part by uncertainty – they will be wrong some portion of the time. While advances will continue, this characteristic will remain. To trust such systems, we need to apply mechanisms that allow us to reason about trust in a system as a whole despite the presence of AI elements. Fortunately, this is not a new problem in system or software engineering, and many approaches and architecture patterns have been developed over decades to help. In safety-critical domains, various forms of fault and hazard analysis are used as a common enabling approach. Use of different architecture patterns for monitoring behavior and providing alternate control strategies are likewise common. In software engineering, we continue to put our trust in systems developed by imperfect developers. We have decades of experience using effective practices and tools to help bound the impact of human mistakes.

We are building on good foundations, but questions remain. Given that AI can be used in many different ways in applications, which types of application are adequately served by existing mechanisms and which require something new or slightly modified? What are the underlying application characteristics that help us make system-level decisions? What characteristics of AI-based software components are essential to communicate to system and software architects to enable their responsible integration into systems that we need to trust? These are important directions to pursue if we are to improve our understanding of and ability to engineer trust in AI-enabled systems.

Trustworthy Explanations of Machine Learning Models through Increased Alignment

Foutse Khomh, Polytechnique Montreal, Canada

As Explainable AI (XAI) tools proliferate, the lack of ground truth for black-box models creates challenges in interpreting conflicting explanations from different methods. In this talk, I advocate for aligning rather than comparing explanation techniques to yield trustworthy insights. We demonstrate that when explanation methods converge, they align with the interpretation of an additive model, offering a practical path to reliable explanations. By partitioning the input space through Functional Decomposition Trees (FD-Trees) and aggregating explanations within the "Rashomon Set" (models with similar performance but diverse explanations), we can systematically reduce disagreements and strengthen interpretability. This alignment-focused approach paves the way for clearer, consensus-driven insights into model behavior.

Risk assessment, safety alignment, and guardrails for generative models

Bo Li, University of Chicago / University of Illinois Urbana-Champaign, USA

The Challenges of Dynamic Optimization

Hiroshi Maruyama, Kao Corp. / The University of Tokyo / Preferred Networks, Japan

The goal of AI research has been to create human-like intelligence on machines. Large language models (LLMs) have nearly achieved this goal, and we now need to set our sights on developing super-human intelligence. This talk argues that one of the main objectives of super-human intelligence should be to discover new knowledge, rather than merely mimicking human knowledge from the past.

To achieve this new goal, we propose a novel form of optimization problem called "Dynamic Optimization." We consider a series of optimization problems $\tilde{P} = P_1,\ P_2,\ \ldots, P_t, P_{t+1}, \ldots$, where each $P_t = \langle D_t, u_t \rangle$ is a well-defined optimization problem with domain D_t and utility function u_t. This framework allows new solutions and contexts to be incorporated into the evolution of $\tilde{P}$.

We demonstrate that many existing optimization techniques, such as dynamic programming, are special cases of Dynamic Optimization. Furthermore, we argue that external intelligent agents, such as humans and/or LLMs, can provide insights into how $\tilde{P}$ should evolve, potentially leading to new discoveries.

Requirements Catchup

Bashar Nuseibeh, The Open University, UK

In Validity We Trust: Generating Reliable Test Cases

Vincenzo Riccio, Università di Udine, Italy

Test Input Generators (TIGs) are crucial in evaluating the correct behavior of several software systems, including those powered by Deep Learning. In particular, TIGs can exercise the software under test with critical inputs that should be similar to real-world scenarios. Unfortunately, automatically generated test inputs may be invalid, i.e., not recognisable as part of the input domain, and may not provide a trustworthy assessment of the software system quality. In this talk, I provide an overview on how TIGs can generate valid inputs, according to both automated and human validators. Moreover, I introduce recent advancements in distribution-aware Generative AI which made TIGs a powerful tool for creating and manipulating synthetic data, while also bringing new complexities and increasing the demands for training and resources. This research is supported by the Project SecCo-OC CUP N. D33C22001300002 PNRR M4 C2 I1.3 "SEcurity and RIghts in the CyberSpace (SERICS)" PE0000014 PE7 funded by Next-Generation EU.

Generative AI for Autonomous Driving Systems Testing

Andrea Stocco, Technical University of Munich, Germany

Recent advancements in Generative AI have made it a powerful tool for creating synthetic images. These can be useful in several domains, such as mitigating the reality gap between simulated and real-world testbeds of perception systems of autonomous vehicles. In this talk, we will investigate how to use image-to-image techniques to mitigate this gap and discuss the effectiveness and computational overhead of diffusion models for producing augmented simulator-generated images of driving scenarios representing new operational design domains. The talk concludes with practical reflections about the pros, cons, and lessons learned that motivate further research in the field.

More Data or More Domain Knowledge? — for LLMs for Code

Lin Tan, Purdue University, USA

In this talk, I will discuss the opportunities and solutions of software-AI synergy: (1) using software approaches to improve the dependability of deep-learning systems, and (2) leveraging deep-learning techniques such as large language models (LLMs) to improve software security and productivity.

On one hand, we generate specifications that describe the behaviors that need to happen given certain input, using differential testing and documentation analysis. We use software techniques such as fuzzing and differential testing, to improve the dependability of deep learning (DL) software.

On the other hand, recent approaches use DL techniques including LLMs to improve coding tasks including automated program repair. An important question is, whether adding more data to train DL models or adding domain knowledge to the models is a more promising or effective direction to improve

LLMs for code. I will discuss existing studies and techniques that answer this question positively or negatively.

Towards Trustworthy LLM Systems

Shuai Wang, Hong Kong University of Science and Technology, Hong Kong

Large Language Models (LLMs) have become a cornerstone of modern artificial intelligence, offering unprecedented capabilities in natural language processing and generation. Their dependability, however, is a multifaceted concept that encompasses accuracy, reliability, consistency, and privacy. Various recent studies have illustrated rather low reliability or unsafety in LLM systems, illustrating concerns that jeopardize real-world usage of LLMs. In this talk, Shuai will introduce their several recent works on assessing and enhancing LLM dependability from the model, software, and system perspectives. He will also discuss some promising future directions that facilitate building reliable LLM-integrated systems.

ML Software Engineering Patterns and Framework

Hironori Washizaki, Waseda University, Japan

Hironori has led the evolution project of the IEEE Computer Society's Guide to the Software Engineering Body of Knowledge (SWEBOK Guide V4). In this talk, he first provides an overview of the SWEBOK Guide and its latest updates, including a new topic, AI and Software Engineering.

Then, as a part of software engineering for AI, the talk introduces concepts of pattern languages in general and major ML software engineering patterns, including security and responsible ML patterns, as well as ML design patterns (IEEE Computer'22 Best Paper). In ML software engineering, there is often a gap between high-level abstract concepts and principles and low-level concrete tools and cases. Patterns encapsulating recurrent problems and corresponding solutions under particular contexts and pattern languages as organized and coherent patterns can fill such gaps, resulting in a common "language" for various stakeholders involved in often interdisciplinary ML software systems development.

The talk also presents a metamodel-based multi-view modeling framework for ML systems with ML software engineering patterns and ML pipeline integration to address the probabilistic nature of ML and its experimentative development approach (IEEE ICEBE'23 Best Paper, SQJ'24, FGCS'24). The framework provides an integrated platform between the modeling environment with ML patterns and ML training, performance monitoring, repair pipelines, and security risk management.

Towards Trustworthy AI: Lessons from Model Analysis and Testing

Xiaofei Xie, Singapore Management University, Singapore

Deep learning testing has widely studied in software engineering community, especially for ensuring trustworthy AI. Much like traditional software testing, numerous methods, such as coverage criteria and fuzz testing, have been developed and proven effective in identifying thousands of mispredictions. However, the sheer volume of these mispredictions prompts a deeper question: How meaningful are they, and how can they help developers improve models?

In this talk, I discussed the limitations of deep learning testing and draw comparisons between testing deep neural networks and traditional software systems. This talk highlighted two major challenges that necessitate rethinking existing deep learning testing practices: model and system specification in testing, and the need for root cause analysis of mispredictions. Based on these, I presented our recent works in this area. DistXplore rethinks testing objectives by focusing on generating diverse, hard-to-detect failures from a distributional perspective. Neural Path Coverage introduces a method for explaining the root cause of mispredictions, where abstract paths could represent the decision logic of deep neural networks. Lastly, I shared insights from our latest work on the regulation of generative AI, which can inform the extraction of meaningful specifications for testing.

LLM changes All or Nothing?

Nobukazu Yoshioka, Waseda University, Japan

LLM-based applications are becoming popular. It, however, is hard to assess the quality of the application, so it is possible for released one to harm users or society by chance. In this talk, we show the research challenges in developing LLM-based applications. Firstly, we describe the differences in the development of traditional ML-based and LLM-based applications from stakeholders, social aspects, and separation of duty to assess quality viewpoints. In detail, we need quality assessment on internal data that are used in prompt engineering of LLM, even if we never trained a machine-learning model. Furthermore, we need to specify the requirements for the usage of LLM produced by a third party to assess LLM-based applications, which is one of the research challenges. From the risk assessment viewpoint, we need to analyze the effects on the environment, including society, and the boundary is unclear. In other words, making boundaries clear is another challenge.

Simple and Effective Blackbox Attacks for Deep Neural Networks

Fuyuan Zhang, The University of Tokyo, Japan

Deep neural networks (DNNs) have achieved significant progress over the past decade and have been widely adopted in various industrial domains. However, a fundamental problem related to DNN robustness remains inadequately addressed, which can potentially lead to various quality issues after deployment, such as safety, security, and reliability. Adversarial attacks are among the most commonly studied techniques to penetrate DNNs. They work by misleading the DNN's decision through well-designed perturbations in the original inputs. In this talk, Fuyuan Zhang will introduce his recent work on developing fuzzing-based blackbox attacks for DNNs used for image classification. His approaches are more effective and query-efficient in generating adversarial examples than state-of-the-art blackbox attacks. Moreover, his approaches can find more subtle adversarial examples compared to other approaches.

Towards the Formal Methods for DNN-Controlled Systems with Safety Guarantee

Min Zhang, East China Normal University, China

Abstraction plays a crucial role in formal methods, simplifying target systems by filtering out extraneous details in models, thereby facilitating scalable verification. In this presentation, we outline our recent advancements in merging formal methods with machine learning, facilitated by abstraction, to develop certified safe DNN-controlled intelligent systems. We demonstrate the critical role of abstraction in this integration by showcasing: (i) the use of abstraction in devising a CEGAR approach to train DNNs on refined state spaces until verification against predefined properties is achieved; (ii) the enhanced efficiency and precision in computing reachable states for complex DNN-controlled systems; and (iii) training verification-friendly piece-wise linear controllers which can be faithfully modeled as hybrid automata and verified by off-the-shelf tools. More details can be found in our CAV 2022, 2024 and NeurIPS 2023 papers.

Reducing LLM Hallucination in Program Analysis Tasks

Xiangyu Zhang, Purdue University, USA

In this talk, I present our recent efforts in reducing LLM hallucination in program analysis tasks such as decompilation, data-flow analysis, and bug finding. Although many have started to use LLMs and Code-Language models in program analysis and program transformation tasks, the results haven't met our expectations. The reason is that these large models hallucinate a lot in complex tasks. There are various reasons behind this. For example, these models treat programs no different from natural language texts during pretraining, although the former have a fundamentally different nature (e.g., due to loops, recursions, and modular design). In addition, the models usually have limited input sizes, which are insufficient for complex tasks. I present a few methods we have developed to reduce hallucination in program analysis, including a novel pre-taining method that challenges the model to understand program semantics by understanding data-flow, a novel context propagation method that addresses model input limits, and a new end-to-end LLM based bug detection pipeline that does not directly prompt the LLM to find bugs, but rather requests the LLM to synthesize code to perform deterministic detection and result sanitization.

Testing of Advanced Systems involving Multiple Intelligent Agents

Xiao-Yi Zhang, University of Science and Technology Beijing, China

In this talk, we review our works on testing for advanced complex intelligent systems like Autonomous Driving Systems (ADS), simulators for multi-agent path-finding systems, and games. These works have been published in

TOSEM'24, ASE'21, FSE'24. Based on the properties of these systems under test, we proposed various approaches aiming to find valuable test cases effectively and efficiently. By summarizing previous works, we present our opinions that smart systems usually preserve complicated and implicit requirements which involve the interaction between the ego system under test and other intelligent artifacts or human beings. Therefore, to address various implicit requirements and generate practical, informative and useful test cases, it is better to consider or assume the environment contains multiple intelligent or human-style artifacts. Specifically, we not only model and simulate the system under test but also model the behaviors of other agents interacting with the ego agent, such that their behaviors are reasonable and realistic. Then, we can propose metrics to identify the failures during these interactions or caused by the interactions among multiple agents. In the talk, we show some possible instances and ways to conduct testing involving these interactions and present the advantages and potential achievements we can get during testing.

Testing of Autonomous Driving Systems Beyond Criticality

Zhenya Zhang, Kyushu University, Japan

This talk reviews our recent works about testing of autonomous driving systems. Starting with a survey of this topic [Tang et al. TOSEM'23], we find that most existing research about testing of autonomous driving systems targets criticality of driving scenarios, but pays less attention to other aspects, such as diversity, realism, which are also very important to the quality of test suite.

Our goal is to achieve better diversity of scenarios via a high level control of scenario generation. In [Tang et al. ISSRE'23], we integrate road structure into the definition of scenarios and generate critical scenarios that arise because of the specific road structures. In [Tang et al. ASE'24], we propose a framework that generate diverse scenarios by manipulating scenarios at function levels, with the assistance of large language models (LLMs). Moreover, we also demonstrate our UAV testing technique that participated in the tool competition in SBFT'24, in which we employ Monte Carlo tree search to balance between exploration and exploitation of the search space to achieve both criticality and diversity.

Focus group on "Requirements of ML-based systems"

The group identified three main research questions to investigate during the discussion related to requirements of ML-based systems:

Q1: How to handle after-the-fact requirements?

Q2: How to control expectations?

Q3: How to reconcile requirements by different stakeholders?

The summary of the discussion is shown in Fig. 1.

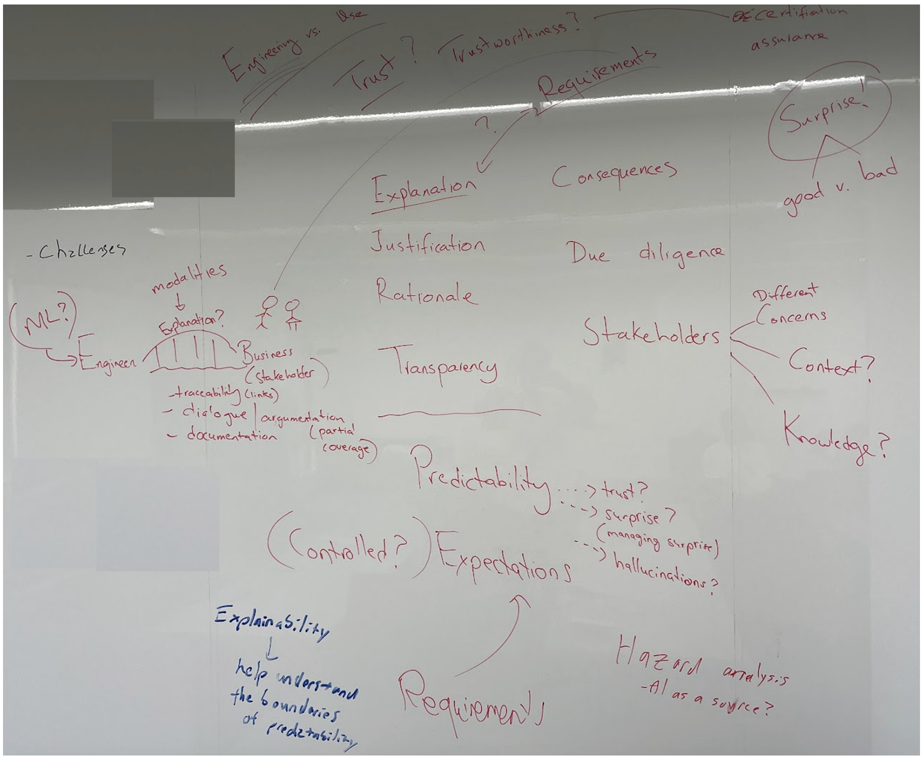

Figure 1: Summary of the discussion of the focus group on "Requirements of ML-based systems"

Focus group on "Neuro-symbolic foundations"

A framework for making choices of foundation models along the neuro-symbolic spectrum

This working group has the assumption that AI system design should consider how to couple at least two foundation models from the neuro-symbolic spectrum (illustrated in Fig. 2), the combination of which will support the development of superior AI systems.

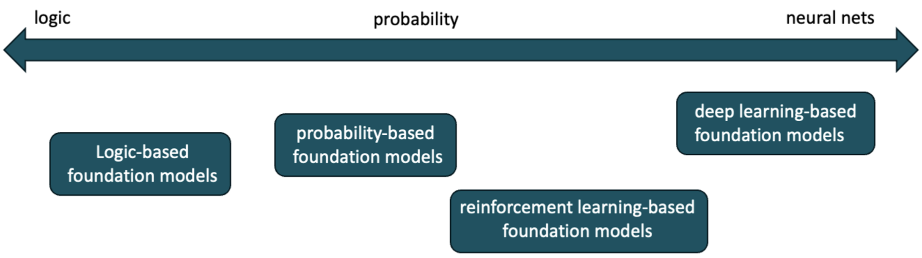

Figure 2: Making choices of foundation model approaches

Knowledge is represented at different levels of abstraction. It can be as concrete as labelled data, which is an instant of facts. It can be also as abstract as rules, mathematical formulas and physical laws, which represent the generality of instants. For a task, it is expected to train AI models on different abstraction levels of knowledge.

An example of connecting points of different logic representation (from Min Zhang) is to augment a "traditional" DNN foundation model with an abstracted model, as depicted in the diagram of Fig. 3.

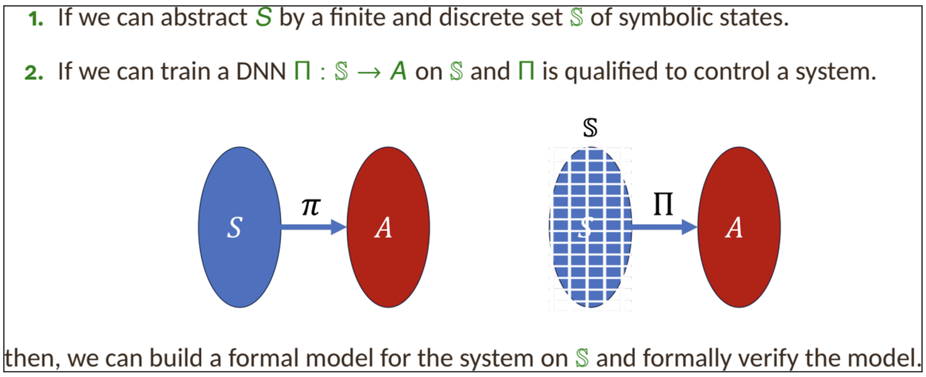

Figure 3: DNN coupled with abstraction of DNN model (from Min Zhang)

Challenges

The general idea of exploiting foundation models from the neuro-symbolic spectrum is to couple or conjoin current best practice in machine learning, largely associated with the creation of deep neural network representations (DNNs), with some kind of more abstract foundation choice, e.g., a more symbolic representation that somehow conforms with the DNN choice.

The challenges include how to pick 2 (or more) representation choices, and how to connect them in a way that provides measurable improvements in an overall AI system based on those choices.

Research directions

The directions of research for neuro-symbolic foundations in this framework needs to do two things:

(1) motivate the choice of two (or more) foundation models along the spectrum (e.g, as above in Fig. 3),

(2) develop precise measures of the overall added value (e.g., improved trustworthiness) for any AI system engineered with the mechanisms selected from (1), and

(3) formulate existing domain-specific systems (e.g., model + guards) as neural-symbolic methods, which allows people to realise the importance and prevalence of neural-symbolic solutions.

Summary from Anthropic Claude, Oct 16, 2024

This output from Claude Sonnet 3.5 is consistent, more or less, with the perspective of the neurosymbolic group.

Our group's Fig. 2 has DNN on the far right, and logic on the far left. Provide a brief explanation of the neurosymbolic spectrum of representation.

`The neurosymbolic spectrum of representation refers to the range of approaches that combine neural networks (connectionist models) with symbolic AI methods. This spectrum encompasses various ways of integrating these two paradigms to leverage their respective strengths. Here's a brief explanation:`

1. `Purely Neural (far left of the spectrum):`
    - `Relies entirely on neural networks`
    - `Learns representations from data`
    - `Examples: Deep learning models like CNNs, RNNs, Transformers`
2. `Neural-Guided Symbolic (left-center):`
    - `Uses neural networks to guide symbolic reasoning`
    - `Neural networks inform or constrain symbolic systems`
    - `Example: Neural theorem provers`
3. `Hybrid Systems (center):`
    - `Combines neural and symbolic components more equally`

- Often uses neural networks for perception and symbolic systems for reasoning
- Example: Neural-symbolic cognitive architectures

4. Symbol-Guided Neural (right-center):
    - Uses symbolic knowledge to inform or constrain neural learning
    - Incorporates prior knowledge into neural network architectures or training
    - Example: Knowledge-infused learning

5. Purely Symbolic (far right of the spectrum):
    - Traditional symbolic AI approaches
    - Relies on explicit rules and knowledge representation
    - Examples: Expert systems, logic programming

Key aspects across the spectrum:

- Representation: From distributed (neural) to localist (symbolic)
- Learning: From data-driven to knowledge-driven
- Reasoning: From implicit to explicit
- Interpretability: Generally increases from left to right
- Flexibility: Generally decreases from left to right

The goal of neurosymbolic AI is to combine the strengths of neural networks (learning from data, handling uncertainty) with those of symbolic AI (explicit reasoning, interpretability, incorporation of prior knowledge).

This spectrum is not rigid, and many approaches fall at different points along it or combine elements from multiple points. The field is actively evolving, with researchers exploring various ways to integrate neural and symbolic methods effectively.

Focus group on "Quality assessment via testing"

The group first identified the main goals of doing quality assessment. Then, it defined the scope, and finally it outlined the possible techniques and challenges.

Main goals

(1) Expose problems → reduce trustworthiness

- Misbehavior, misprediction

(2) Provide useful input to developers → increase trustworthiness

- Testing feeding (re)training, fine-tuning (add info), delete info?
- Testing feeding debugging

(3) Provide inputs to explanations → increase trustworthiness

- Oracle-driven explanation?
- Example: un-avoidable collisions

Focus/Scope/Running examples

Testing Autonomous Systems (e.g., Driving systems, Factories, robotics, etc.)

- Types of testing:
  - Simulation in the loop: online testing
  - Seed images: offline testing
  - Human-in-the-loop: validation, generation?
- Testing levels:
  - Component/ML Model
  - Integration
  - System
  - System-of-system
- Possible Testing Environments:
  - Virtual/Simulation (*)
  - Real-world

Techniques and challenges

- Impact analysis becomes extremely hard.
- Does regression make sense in simulation-based testing of systems that learn?
  - Can we adopt test selection?
  - Do we need to regenerate all the tests?

  - Is it enough to use the same test generator technique?

- Test transferability (as transfer learning) between simulators:
  - Simulator-dependent failures. If the same test transfers across simulators
  - Version-dependent failures. If the same test produces the same results in different versions

- How can we assess/measure the (testing) process's trustworthiness or (its improvement)?
  - Is this (related) adequacy?

- Reality gap: visual, physics

- Test Robustness: ground truth vs noise

- The Oracle problem:
  - Extract ground truth from documents (e.g., car specs, road specs, general laws of physics)
    * Example: perfect segmentation vs real-life input segmentation systems
    * Observe variations and explain them
  - Differential/Metamorphic testing and redundancy:
    * Invariant oracle (metrics sparse/dense, time/space) + confidence interval
    * Equivalence (metamorphic testing)
  - How to gather requirements and extracting oracles:
    * Exploratory testing to identify/elicit/expose implicit or missing requirements
    * traffic regulations
    * oracle/requirement selection, mapping tests to requirements
  - Scenario prioritization: use simulated scenarios to select real-life scenarios
    * Flaky-tests vs controllability of simulators

Focus group on "Quality improvement via automated repair"

During the discussion, the group first identified some open issues and then some possible future research directions.

Open issues

- Oracle problem for autonomous systems and ML-based systems
  - How can write specifications?
  - Can we recover formal specifications from informal text with LLMs? LLMs know common formalisms but still make mistakes.
  - Human-readable/understandable requirements. Use names/labels that a human would use.
  - There are some non-specifiable requirements
    - e.g., human-like driving or consistent with human expectation (behavior does not make no surprise to pedestrians
    - We may use ML for data mining to extract (approximate) formal specifications from data but it is costly if we need human annotations
    - Now "LLM as a judge" may be used to reflect a judgment with "commonsense". Is the observed driving behavior acceptable in the society? Is this test configuration realistic, and is the crash occurrence critical in the society?
  - Sometimes what is feasible is not known. We could ask perfect accuracy, but this could be impossible. There is some tradeoff between some requirements. And this tradeoff is not known in advance.
  - We can utilise the idea of Metamorphic Testing to solve the oracle problem? In traditional SE, metamorphic testing is used in a setting where oracle is not available.
    - Even a metamorphic relation that applies to only part of the oracle can be useful.
    - Metamorphic testing has been used to test DNNs for object detection and LLMs. LLMs are supposed to provide semantic equivalent answers to the same question (semantic equivalent questions).
- How to repair? What to repair? What is the goal of repair?
  - When we talk about DNN repair, do we have to follow the program repair paradigm in traditional SE?
    - No.
    - Traditional SE program repair considers all existing tests as a binding oracle and thus achieves only Pareto-optimal improvements. This is very effective at fixing small SW bugs but too limiting for improving on complex goals/tasks.

- What techniques are good in which situations, then?
  - Adding a lot of proper training data. It is effective, but very costly.
  - Change the model architecture. The space of design exploration is very large.
  - Parameter tuning:
    - Training with custom configurations (weighted loss function, over/down-sampling, etc.)
    - Repair by fault localization and metaheuristic optimization. It is effective for conservative updates
    - Formal repair (integer programming, etc.).
    - Hybrid: their combinations.

- What is a fault in neural networks?
  - It is like a difference between a binary concept of fault (in traditional code program) and quantitative concept of fault (some weights are deviated from their "oracles" farther, some are closer)
    - Maybe ML can be used to learn about the importance of different features in different situations? Some features are critical, others are "nice to have".
  - If we clearly know the specification of a certain part of the DNN, then we can compare the behaviour of this part against its specification to know whether there is a fault at this part. Otherwise, we cannot say there is a fault somewhere in the DNNs.
  - Maybe we can synthesise a specification for a specific component of the DNN after it is well-trained to learn what that component aims to do after training. Then, we use the synthesised specification for fault localization. Compared to traditional SE, we have specifications BEFORE writing the program. However, in ML setting, we try to find a specification for certain part of the component AFTER the programs are constructed, i.e., the DNNs finished training.

- Perception task vs. control task
  - we may be aware the difference when we discuss challenges
  - For control, we can often define clear formal specifications that must be satisfied (and can be)
    - e.g., the minimum constraint like: need to select "braking" if the distance to the preceding car is too small
  - For perception, the specification is very unclear with trade-offs
    - Sometimes we may try to make clear specifications for simple objects such as road signals (large red circle, white rectangle inside the circle with the <30% area, ...)
    - We are not so sure the acceptable and feasible accuracy
    - Some target objects such as pedestrian and bike rider shares common features and sometimes the problem is deciding the balance of thresholds

Future research directions

- We should have a feedback loop between repair and requirements elicitation.
- Formal/combinatorial approaches could help us cover the input space in a better way
- Diffusion models can be used to generate new data for the purpose of fault localization and DNN repair
- Fine-tuning or retraining could be used to improve an overall property of the network, which tries to solve the problem at system-level. DNN repair, instead, can be used to improve a local component of the DNN, which tries to solve the problem at a component-level.

List of Participants

- Paolo Arcaini, National Institute of Informatics, Japan
- Cyrille Artho, KTH Royal Institute of Technology, Sweden
- Lionel Briand, Lero Centre, University of Limerick, Ireland
- Alessio Gambi, Austrian Institute of Technology, Austria
- Randy Goebel, University of Alberta, Canada
- Satoshi Hara, The University of Electro-Communications, Japan
- Fuyuki Ishikawa, National Institute of Informatics, Japan
- James Ivers, Carnegie Mellon University, USA
- Foutse Khomh, Polytechnique Montreal, Canada
- Bo Li, University of Chicago / University of Illinois Urbana-Champaign, USA
- Lei Ma, The University of Tokyo, Japan / University of Alberta, Canada
- Hiroshi Maruyama, Kao Corp. / The University of Tokyo / Preferred Networks, Japan
- Bashar Nuseibeh, The Open University, UK
- Vincenzo Riccio, Università di Udine, Italy
- Andrea Stocco, Technical University of Munich, Germany
- Lin Tan, Purdue University, USA
- Shuai Wang, Hong Kong University of Science and Technology, Hong Kong
- Hironori Washizaki, Waseda University, Japan
- Xiaofei Xie, Singapore Management University, Singapore
- Nobukazu Yoshioka, Waseda University, Japan
- Fuyuan Zhang, The University of Tokyo, Japan
- Min Zhang, East China Normal University, China
- Xiangyu Zhang, Purdue University, USA
- Xiao-Yi Zhang, University of Science and Technology Beijing, China
- Zhenya Zhang, Kyushu University, Japan

Group photos

Figure 4: Group photos on October 14 and 15, 2024

Yosegaki

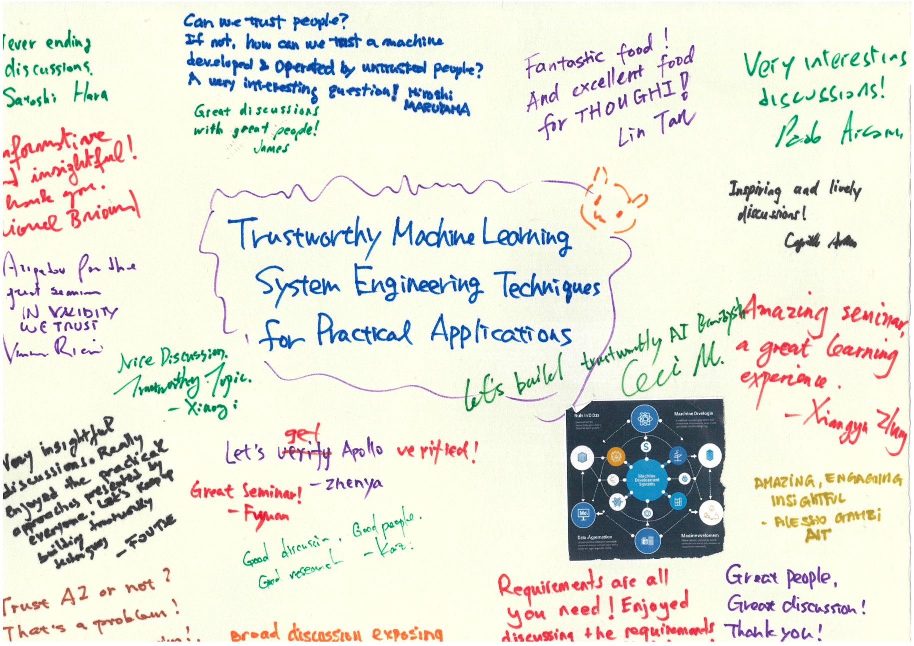

Figure 5: Yosegaki