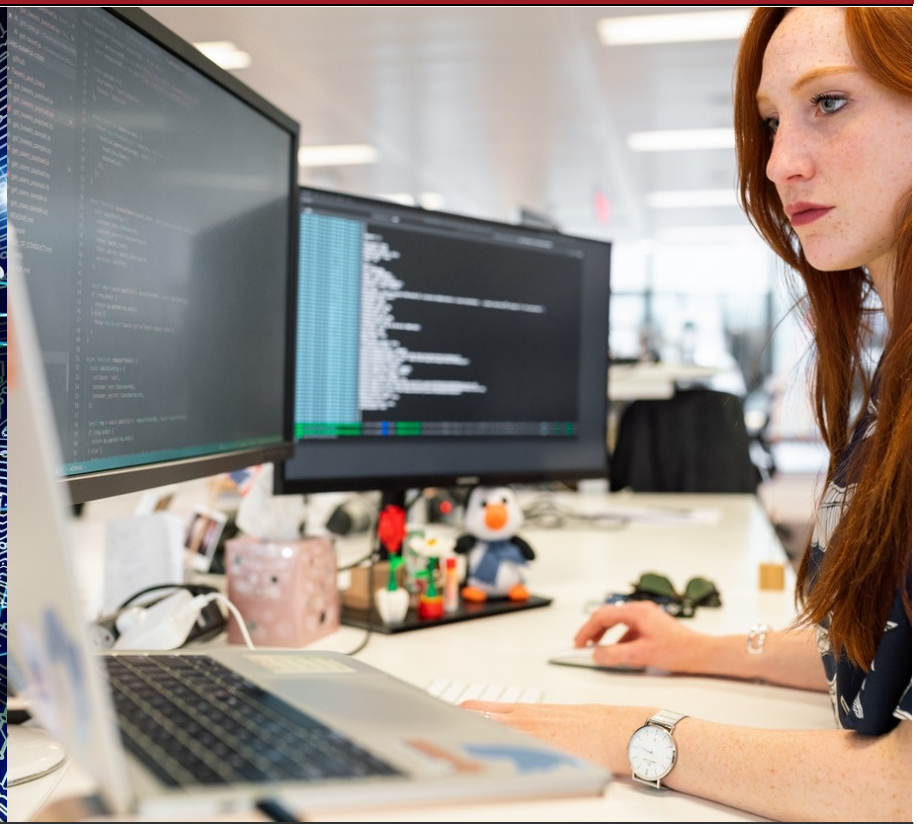
Software Engineering for Artificial Intelligence

Introduction

TECHNISCHE UNIVERSITÄT DARMSTADT



Attendance via Zoom

TECHNISCHE UNIVERSITÄT DARMSTADT

Let‘s try to make this a great experience for all of us:

Please check your setup before the meeting. We start all calls 10 minutes early, where you can do so.

Please connect before the meeting starts.

Please join using your full name. If you use a nickname or pseudonym, tell the advisors (needed for grading).

We encourage you to use a microphone and a camera: It improves the overall experience in interactive parts.

Please mute your microphone when not speaking

Agenda

TECHNISCHE UNIVERSITÄT DARMSTADT

- Motivation
- This Seminar
- Seminar Structure & Grading
- Schedule
- Registration

Advisors & Contact

TECHNISCHE UNIVERSITÄT DARMSTADT

Any questions, suggestions, interested in research or collaborations?

Talk to us or drop a mail!

Prof. Mira Mezini

mezini@st.informatik.tu-darmstadt.de

Dr. Krishna Narasimhan

kri.nara@st.informatik.tu-darmstadt.de

Daniel Sokolowski, MSc

daniel.sokolowski@unisg.ch

Most prob faster reply…

Non-SE4AI things (Research, thesis, jobs, general advice)

SE4AI: Organizational questions, enrollment and submissions

Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT

Imagine: We build together *AcaWhooo!* a „Google Translate“ for scientific text.

What are challenges apart from accuracy?

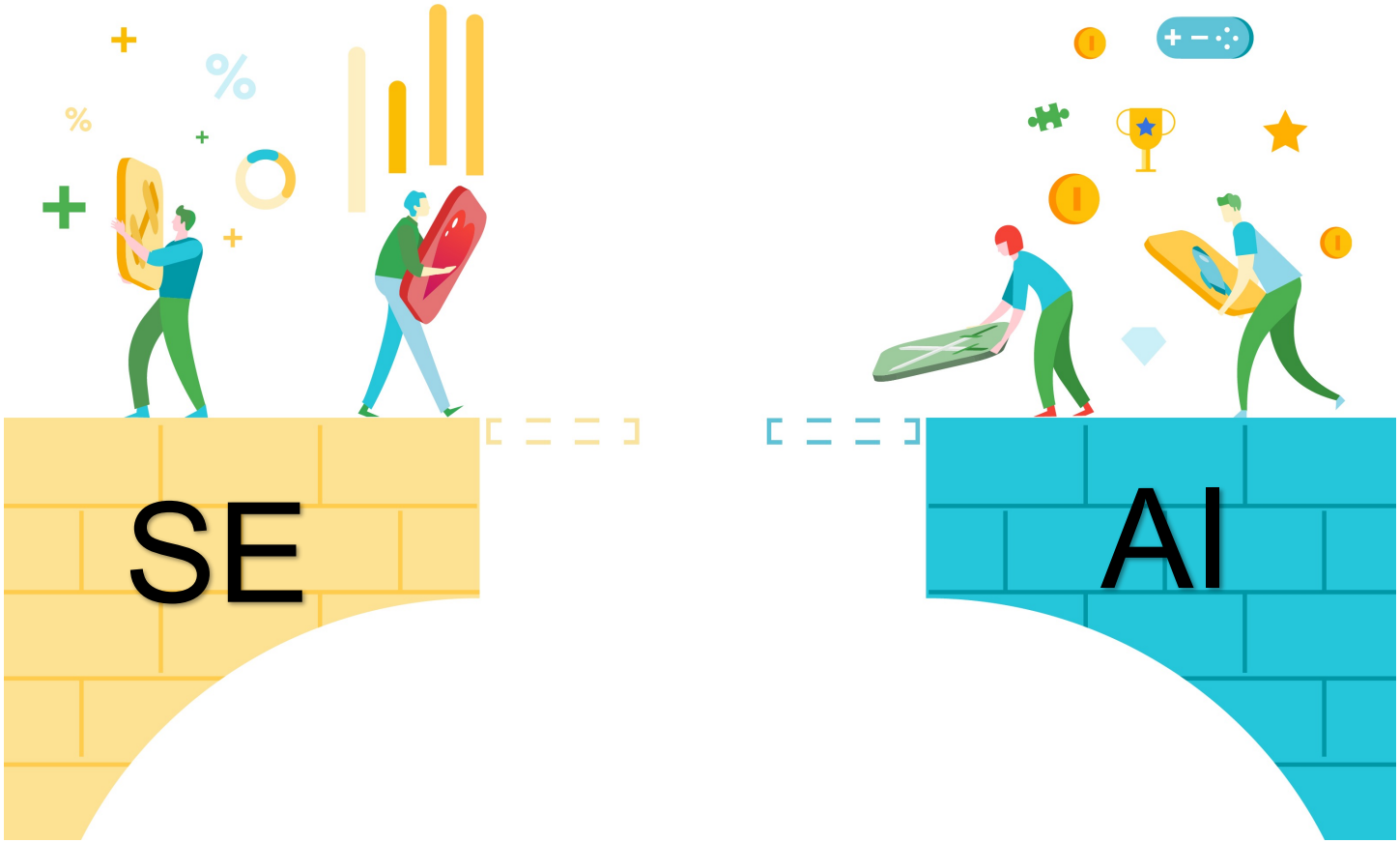
From Data Science to Production

TECHNISCHE UNIVERSITÄT DARMSTADT

- A data scientist can build our program, but…
  - They are used to fixed datasets and focus heavily on accuracy.
  - They prototype, often using Jupyter notebooks, etc.
  - They are experts in modeling and feature egineering, but stability, size, updateability and other aspects, which are important in production, do mostly not matter.
- A software engineer is focused on production grade software
  - Concerned about **many different kinds of product quality**: performance, security, safety, stability, release time, cost, customer satisfaction, maintainability, reliability, scalability, fault tolerance, ...
- **Both worlds need to be brought together!**

From Data Science to Production

TECHNISCHE UNIVERSITÄT DARMSTADT



This Seminar Changed!

TECHNISCHE UNIVERSITÄT DARMSTADT

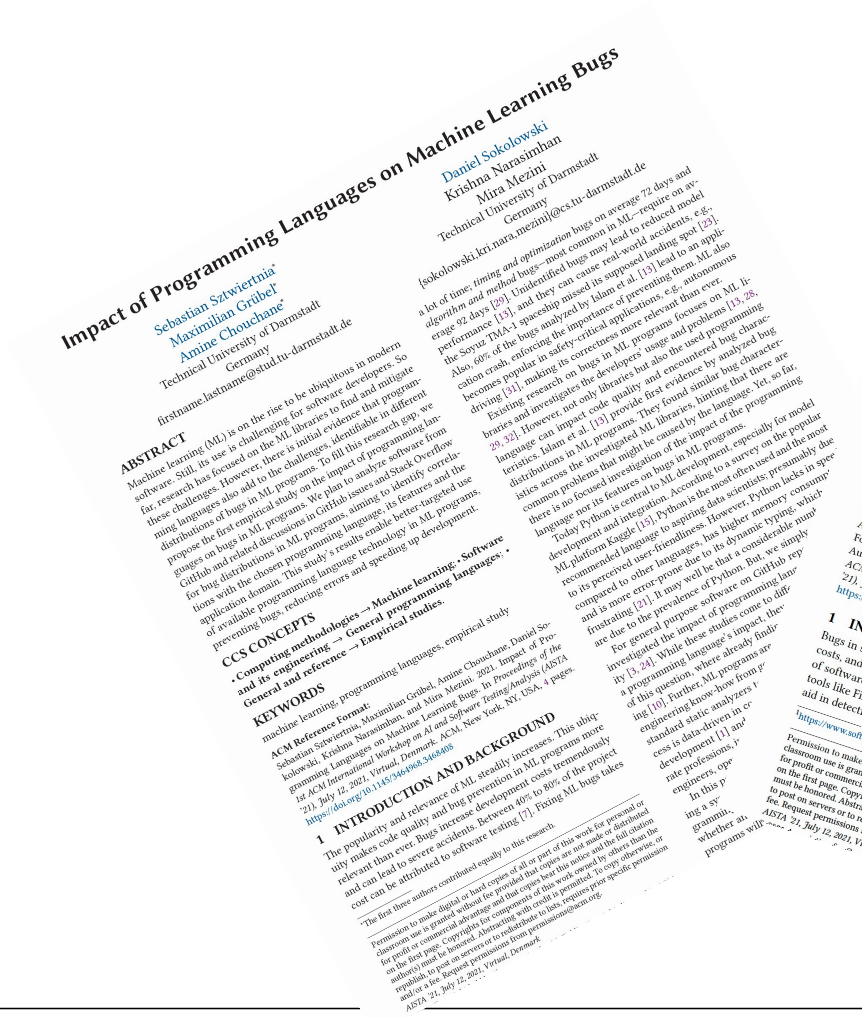
- SOSE 2020: Overview Seminar
  - Presented various topics in improving SE for AI Systems
  - Gave an overview and some insights, but not very deep
- Since WISE 2020: **Think Tank Seminar** 
  - Interactively critique and analyse one problem in-depth
  - Develop ideas collectively to improve the state-of-the-art

In this Seminar

TECHNISCHE UNIVERSITÄT DARMSTADT

- We learn about one specific problem domain within SE4AI.
- We learn how to write scientific text collaboratively.
- We will critically analyse state-of-the-art papers in this domain.
- We will write a vision paper (4-5 pages) proposing new ideas based on the SOTA.
- Potentially become published in International workshops

WISE 2020 peer-reviewed papers in Intl. Workshop



Impact of Programming Languages on Machine Learning Bugs

Sebastian Sztwiertnia*
Maximilian Grübel*
Amine Chouchane*
Technical University of Darmstadt
Germany
firstname.lastname@stud.tu-darmstadt.de

Daniel Sokolowski
Krishna Narasimhan
Mira Mezini
Technical University of Darmstadt
Germany
{sokolowski,kri.nara,mezini}@cs.tu-darmstadt.de

ABSTRACT

Machine learning (ML) is on the rise to be ubiquitous in modern software. Still, its use is challenging for software developers. So far, research has focused on the ML libraries to find and mitigate these challenges. However, there is initial evidence that programming languages also add to the challenges, identifiable in different distributions of bugs in ML programs. To fill this research gap, we propose the first empirical study on the impact of programming languages on bugs in ML programs. We plan to analyze software from GitHub and related discussions in GitHub issues and Stack Overflow for bug distributions in ML programs, aiming to identify correlations with the chosen programming language, its features and the application domain. This study's results enable better-targeted use of available programming language technology in ML programs, preventing bugs, reducing errors and speeding up development.

CCS CONCEPTS

• Computing methodologies → Machine learning; • Software and its engineering → General programming languages; • General and reference → Empirical studies.

KEYWORDS

machine learning, programming languages, empirical study

NerdBug: Automated Bug Detection in Neural Networks

Foad Jafarinejad
Department of Computer Science,
Technical University of Darmstadt
Darmstadt, Hessen, Germany
foad.jafarinejad@stud.tu-darmstadt.de

Krishna Narasimhan
Department of Computer Science,
Technical University of Darmstadt
Darmstadt, Hessen, Germany
kri.nara@st.informatik.tu-darmstadt.de

Mira Mezini
Department of Computer Science,
Technical University of Darmstadt
Darmstadt, Hessen, Germany
mezini@st.informatik.tu-darmstadt.de

ABSTRACT

Despite the exponential growth of deep learning software during the last decade, there is a lack of tools to test and debug issues in deep learning programs. Current static analysis tools do not address challenges specific to deep learning as observed by past research on bugs specific to this area. Existing deep learning bug detection tools focus on specific issues like shape mismatches. In this paper, we present a vision for an abstraction-based approach to detect deep learning bugs and the plan to evaluate our approach. The motivation behind the abstraction-based approach is to be able to build an intermediate version of the neural network that can be analyzed in development time to provide live feedback programmers are used to with other kind of bugs.

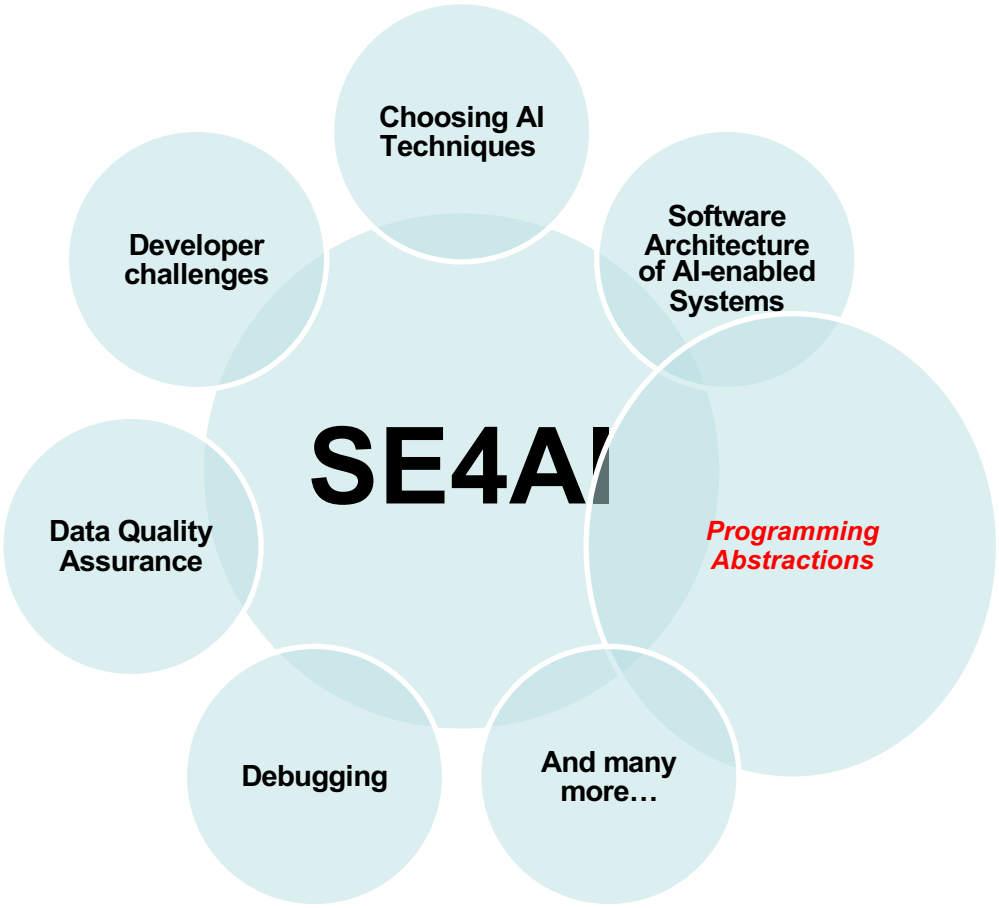
CCS CONCEPTS

• Software and its engineering → Software defect analysis; Maintaining software; Model-driven software engineering; Correctness; Context specific languages; System modeling languages; Software testing and debugging; Scripting languages; Software development techniques.

KEYWORDS

Bug Detection, Debugging, Machine Learning, Deep Learning

Topics of SE4AI

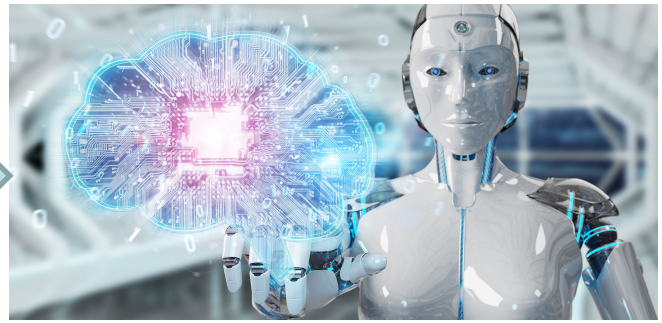


Our Focus

PROGRAMMING ABSTRACTIONS TO HELP AI DEVELOPERS

Behind every Intelligent system is a developer struggling with AI modelling

TECHNISCHE UNIVERSITÄT DARMSTADT

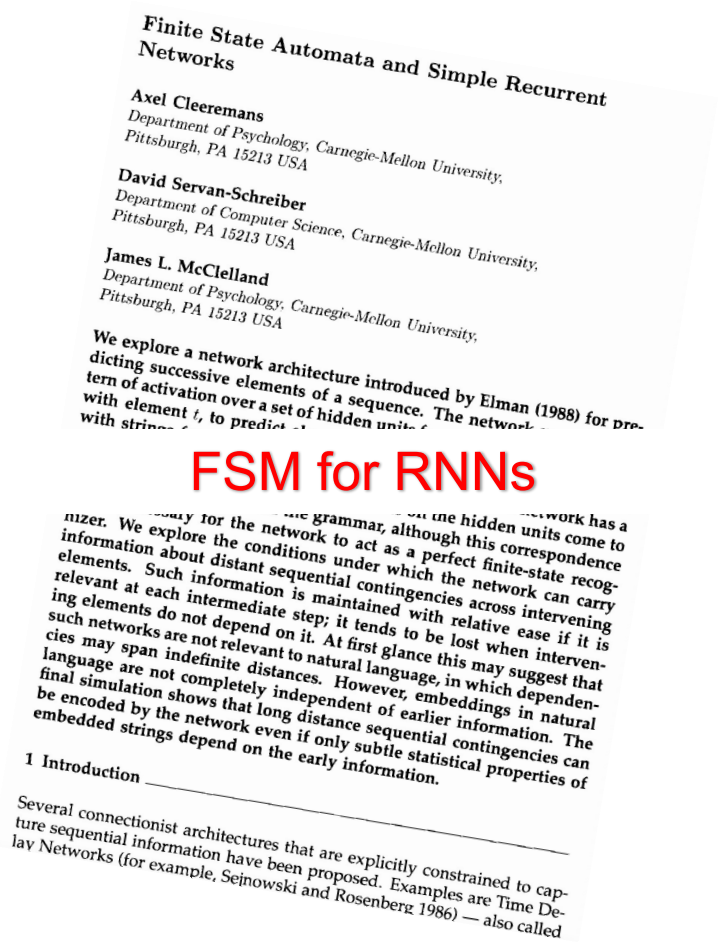


Right abstractions help

TECHNISCHE UNIVERSITÄT DARMSTADT

RASP cheat sheet

RASP for Transformers



Finite State Automata and Simple Recurrent Networks

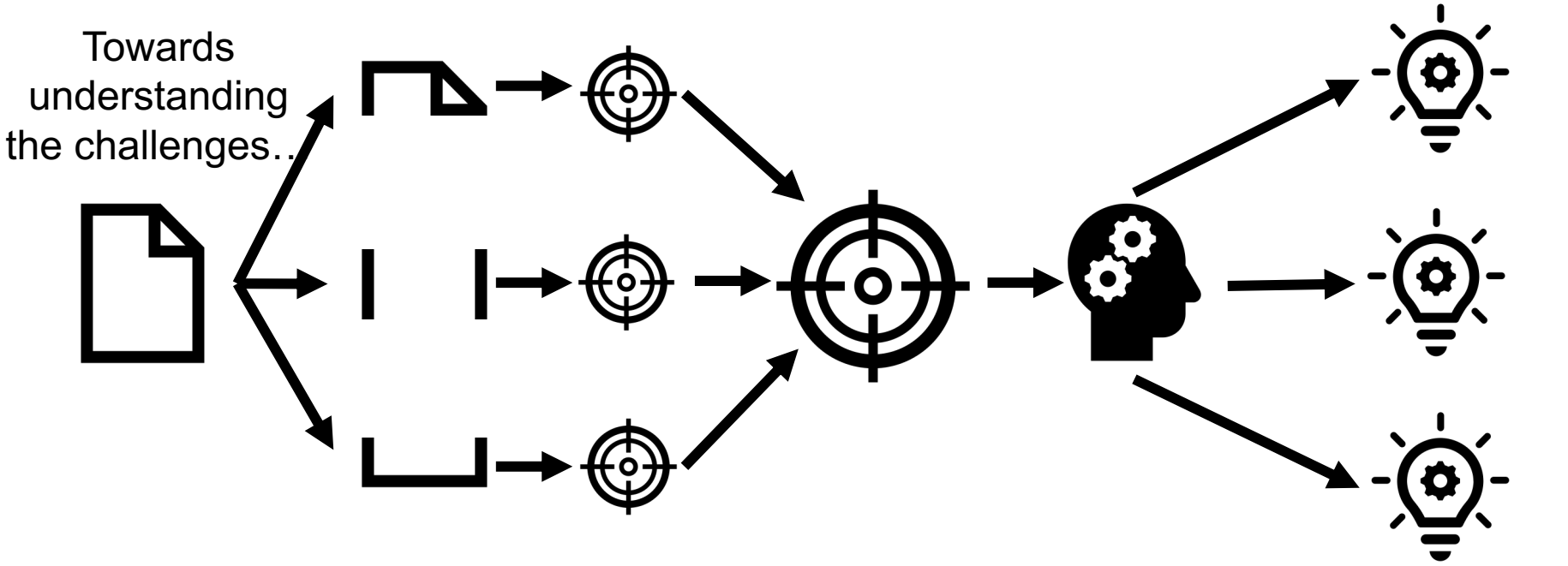
FSM for RNNs

Recommended papers for this Course

TECHNISCHE UNIVERSITÄT DARMSTADT

- Solo: Enforcing Differential Privacy Without Fancy Types. 2021 · ArXiv · https://arxiv.org/abs/2105.01632
- PolyDL: Polyhedral Optimizations for Creation of High Performance DL primitives. 2021 · TACO · Tavarageri, Heinecke, Avancha, Goyal, Upadrasta, Kaul. https://arxiv.org/abs/2006.02230
- Neural architecture search as program transformation exploration. 2021 · ASPLOS · Turner, Crowley, O'Boyle. https://arxiv.org/abs/2102.06599
- Thinking Like Transformers. 2021 · ICML · Weiss, Goldberg, Yahav. https://arxiv.org/pdf/2106.06981.pdf
- A Study of Continuous Vector Representations for Theorem Proving. 2021 · ArXiv · Purgal, Parsert, Kaliszyk. https://arxiv.org/abs/2101.09142
- Paradoxes of probabilistic programming. 2021 · POPL · Jacobs. https://arxiv.org/abs/2101.03391
- Modularizing Deep Learning via Pairwise Learning With Kernels. 2021 · IEEE transactions on neural networks and learning systems · Duan, Yu, Príncipe. https://arxiv.org/abs/2005.05541
- λ_s: computable semantics for differentiable programming with higher-order functions and datatypes. 2021 · POPL · Sherman, Michel, Carbin. https://arxiv.org/abs/2007.08017
- Reverse AD at Higher Types: Pure, Principled and Denotationally Correct. 2021 · ESOP · V'ak'ar. https://arxiv.org/abs/2007.05283
- DeepProbLog: Neural Probabilistic Logic Programming 2018 · NeurIPS· Manhaeve et al. https://arxiv.org/abs/1907.08194
- Deepstochlog: Neural stochastic logic programming 2021 · ArXiv· Winters et al. https://arxiv.org/abs/2106.12574
- SLASH: Embracing Probabilistic Circuits into Neural Answer Set Programming 2021 · ArXiv· Skryagin et al. https://arxiv.org/abs/2110.03395
- Neural Logic Machines 2019· ICLR· Dong et al. https://arxiv.org/pdf/1904.11694.pdf
- Reactive probabilistic programming 2020· PLDI· Baudart et al. https://dl.acm.org/doi/pdf/10.1145/3385412.3386009

Seminar Overview (Focus – Programming Abstractions for AI)



Schedule

TECHNISCHE UNIVERSITÄT DARMSTADT

Today	• Kick-off meeting
Oct 28	• Seminar registration
Oct 29	• Registration confirmation
Nov 2	• Paper Assignment
Nov 9	• Paper summary submission
Nov 9	• Paper summary presentation/ discussion
Dec 5	• SOTA summary submission
Dec 7	• SOTA summary presentation and brainstorm ideas
Dec 14	• Vision paper finalizing
Jan 11, Jan 25	• Workshop
Feb 6	• Vision paper submission
Feb 8	• Last meeting

Grading

TECHNISCHE UNIVERSITÄT DARMSTADT

- Individual Summary
  - Presentation (5 mins – ~5 slides) – 10%
  - Submission (1 page + references) – 20%
- Collaborative Summary (2 pages + references) – 20%
- Participation – 10%
- Vision Paper (4-5 pages + references) – 40%
  - 2 pages summary
  - 1-2 pages for the idea
  - 1 page for the impact analysis

Registration

TECHNISCHE UNIVERSITÄT DARMSTADT

- Send a mail by October 28th to kri.nara@st.informatik.tu-darmstadt.de and daniel.sokolowski@unisg.ch
  - Include why you are interested to participate in this seminar (max 4-5 sentences)
  - If you have related experiences, mention them (courses, projects, …)
- If more than 8 registrations, we will select based on the mail
- We confirm the registration by October 29th via mail
- (Do not forget to register in TUCaN as well)

Question & Answers

TECHNISCHE UNIVERSITÄT DARMSTADT

Acknowledgements & License

- Material Design Icons, by Google under Apache-2.0
- Other images are either by the authors of these slides, attributed where they are used, or licensed under Pixabay or Pexels