Generative AI and Variability
A Research Vision

Sandra Greiner **Klaus Schmid[2] Thorsten Berger[3] Sebastian Krieter[4] Kristof Meixner[5]**

Software Engineering Group
University of Bern, Switzerland

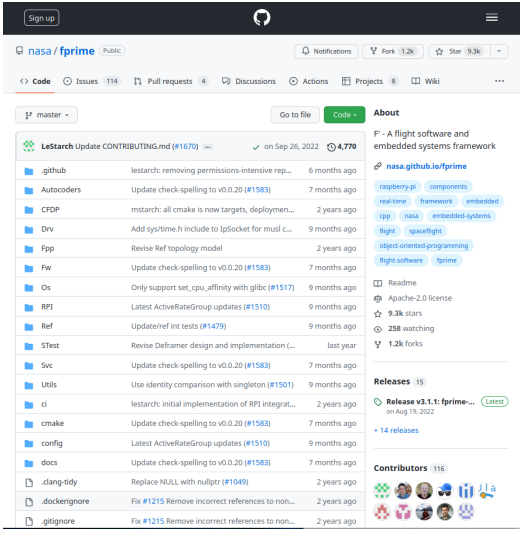
Februray, 9th 2024 **VaMoS '24 in Bern**

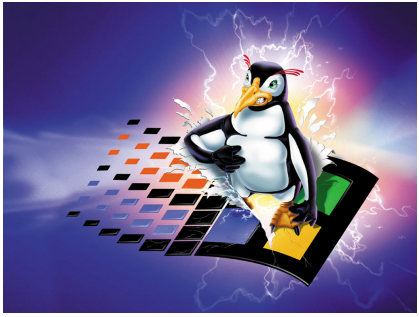
[2]University of Hildesheim [3]Ruhr University Bochum [4]University of Ulm [5]TU Vienna

Current State of

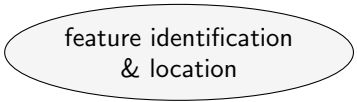
Managing Variable-Intensive Software

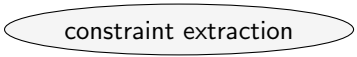
SOTA: Large (Evolving) Variability-Intensive Software





Problem





Problem

variability modeling

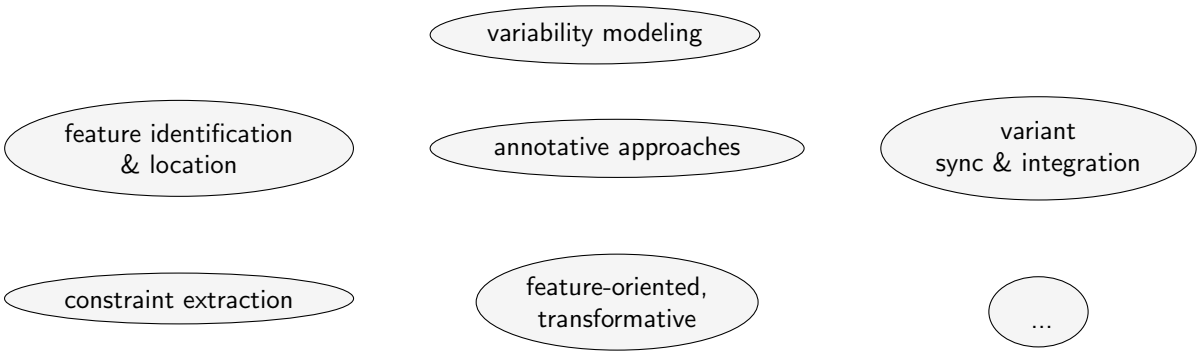
feature identification & location

annotative approaches

constraint extraction

feature-oriented, transformative

Problem


variability modeling
feature identification & location
annotative approaches
variant sync & integration
constraint extraction
feature-oriented, transformative
...

Problem

variability modeling

feature identification & location

annotative approaches

variant sync & integration

constraint extraction

feature-oriented, transformative

...

language-specific, hard to analyze large systems, and to maintain consistently ...

Problem

variability modeling

feature identification & location

annotative approaches

variant sync & integration

constraint extraction

feature-oriented, transformative

...

language-specific, hard to analyze large systems, and to maintain consistently ...

still insufficient genericity and automation

Availability and Generalization Power of Generative AI

Business / Tech

A year after ChatGPT's release, the AI revolution is just beginning

By Catherine Thorbecke, CNN
8 minute read · Updated 10:32 AM EST, Thu November 30, 2023

Availability and Generalization Power of Generative AI

Business / Tech

A year after ChatGPT's release, the AI revolution is just beginning

By Catherine Thorbecke, CNN
8 minute read · Updated 10:32 AM EST, Thu November 30, 2023

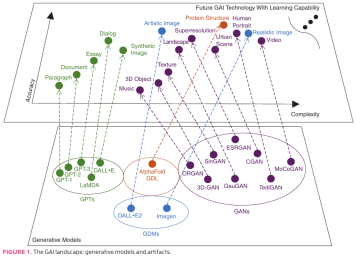


FIGURE 1. The GAI landscape: generative models and artifacts.

Availability and Generalization Power of Generative AI


Business / Tech

A year after ChatGPT's release, the AI revolution is just beginning

By Catherine Thorbecke, CNN
8 minute read · Updated 10:32 AM EST, Thu November 30, 2023

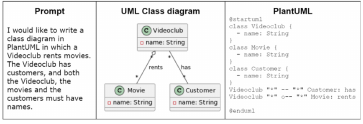


Fig. 2 Prompt used to ask ChatGPT to generate a UML class diagram of a video club system, and the resulting model

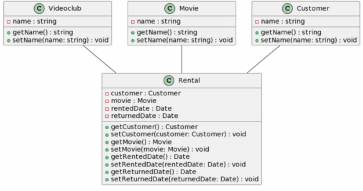


Fig. 3 Another model generated by ChatGPT in response to exactly the same prompt, but in a different session

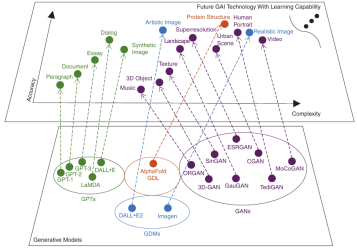


FIGURE 1. The GAI landscape: generative models and artifacts.

On Programming Variability with Large Language Model-based Assistant

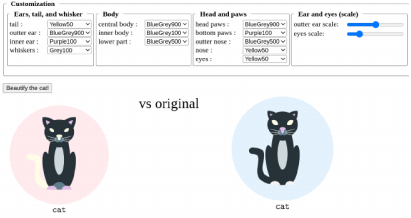
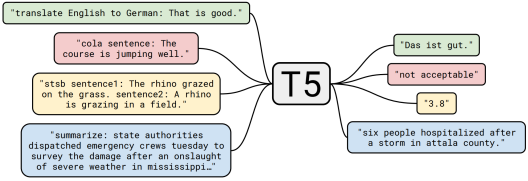


Figure 1: End-user customization of Cat TikZ code

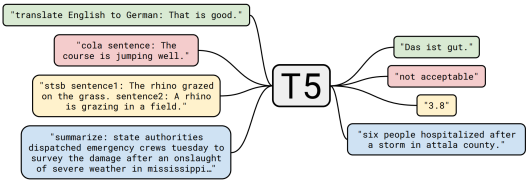
What does it need from research to support maintaining variability-intensive software systems with Generative AI?

Background

Generative AI – Large Language Models

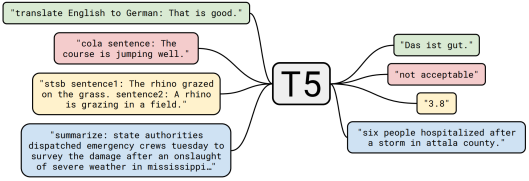


Generative AI – Large Language Models



The basic idea is to use a large language corpus to train a neural network to learn the language, by hiding part of the text and asking the network to guess the missing parts.

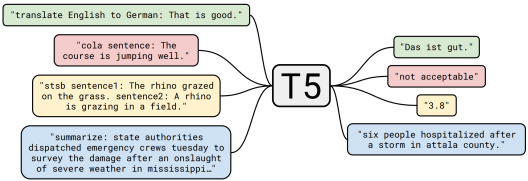
Generative AI – Large Language Models



The basic idea is to use a large language corpus to train a neural network to learn the language, by hiding part of the text and asking the network to guess the missing parts.

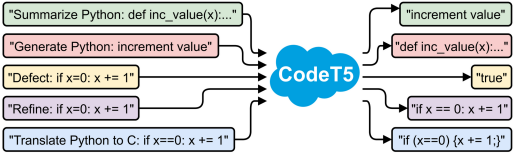
large pre-trained transformer models: convert *input* text into *output* text (encoder-decoder)

Generative AI – Large Language Models



The basic idea is to use a large language corpus to train a neural network to learn the language, by hiding part of the text and asking the network to guess the missing parts.

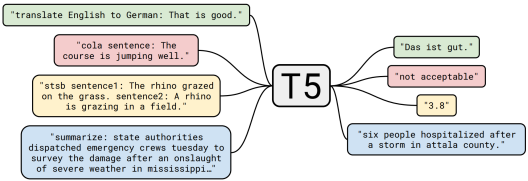
large pre-trained transformer models: convert *input* text into *output* text (encoder-decoder)



Generative AI for SE: pre-trained on several programming languages

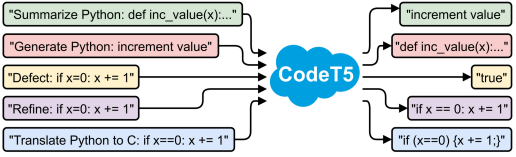
- defect & clone detection
- code summarization
- code translation,
- ...

Generative AI – Large Language Models



The basic idea is to use a large language corpus to train a neural network to learn the language, by hiding part of the text and asking the network to guess the missing parts.

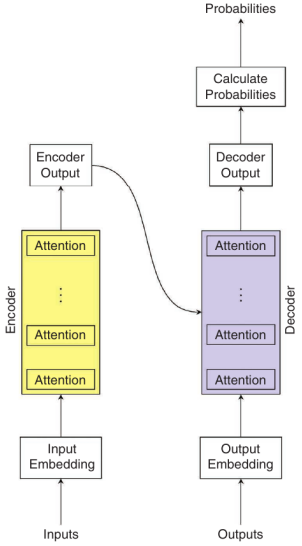
large pre-trained transformer models: convert *input* text into *output* text (encoder-decoder)



Generative AI for SE: pre-trained on several programming languages

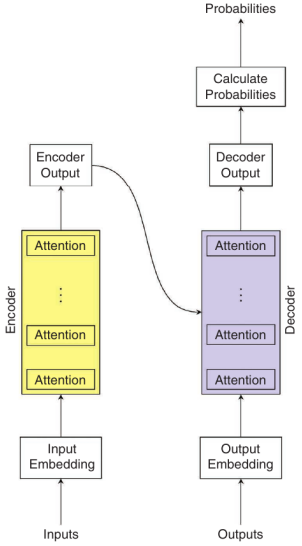
- defect & clone detection
- code summarization
- code translation,
- ...

Transformer Architecture



context window defines size of input and output (in tokens)

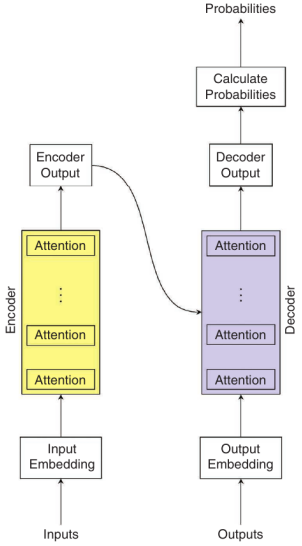
Transformer Architecture



context window defines size of input and output (in tokens)

typical sizes: 8k-32k

Transformer Architecture



context window defines size of input and output (in tokens)

typical sizes: 8k-32k

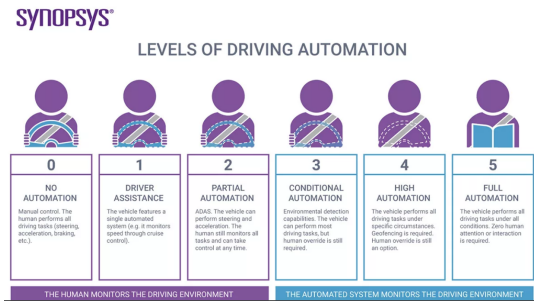
Further tuning through hyperparameters (temperature, learning rate, ...)

What does it need from research to support maintaining variability-intensive software systems with Generative AI?

Idea

How can we assess success of Gen-AI-enabled development of variability-intensive software?

How can we assess success of Gen-AI-enabled development of variability-intensive software?



Levels and Characteristics

How can we **classify** and assess progress
in using Generative AI for variability-aware SE ?

Levels and Characteristics

How can we **classify** and assess progress
in using Generative AI for variability-aware SE ?

Scale information amount processed in one step

Levels and Characteristics

How can we **classify** and assess progress
in using Generative AI for variability-aware SE ?

Scale information amount processed in one step

Task Type analytical vs. adaptive, to a productive end*

Levels and Characteristics

How can we **classify** and assess progress
in using Generative AI for variability-aware SE ?

Scale	information amount processed in one step
Task Type	analytical vs. adaptive, to a productive end*
Quality	level of reliability: random, junior developer, senior developer, even better

Levels and Characteristics

How can we **classify** and assess progress
in using Generative AI for variability-aware SE ?

Scale	information amount processed in one step
Task Type	analytical vs. adaptive, to a productive end*
Quality	level of reliability: random, junior developer, senior developer, even better
Automation	level of manual intervention

*not a specific variability-intensive task

Automation Levels

Automation Levels

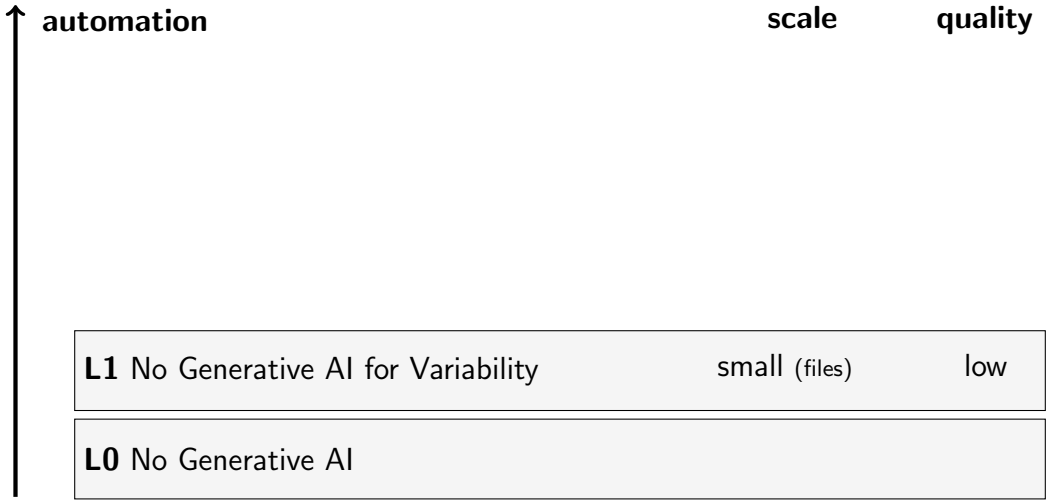
automation

scale

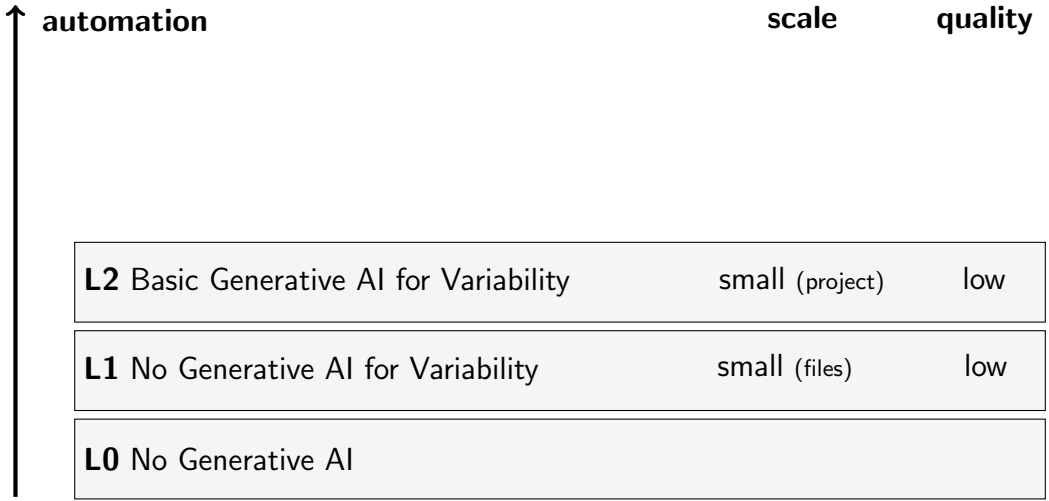
quality

L0 No Generative AI

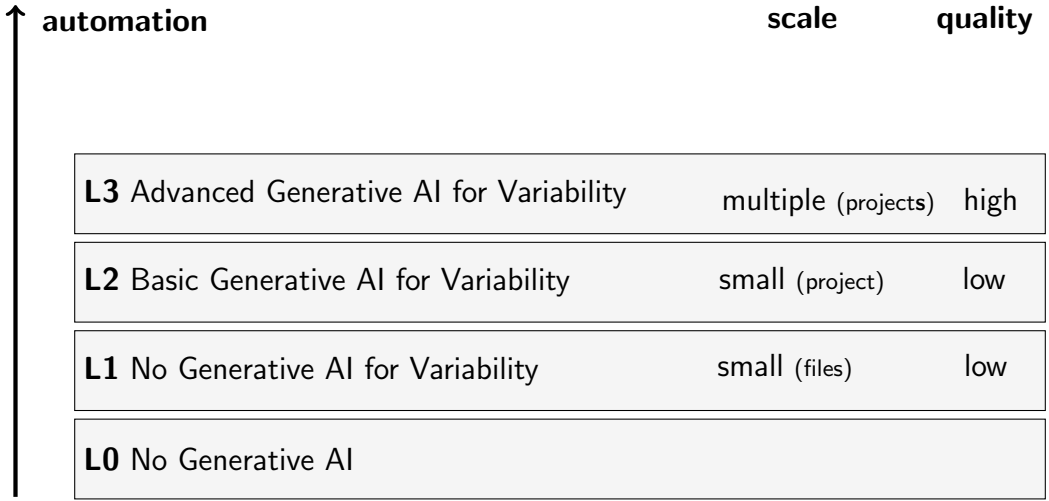
Automation Levels



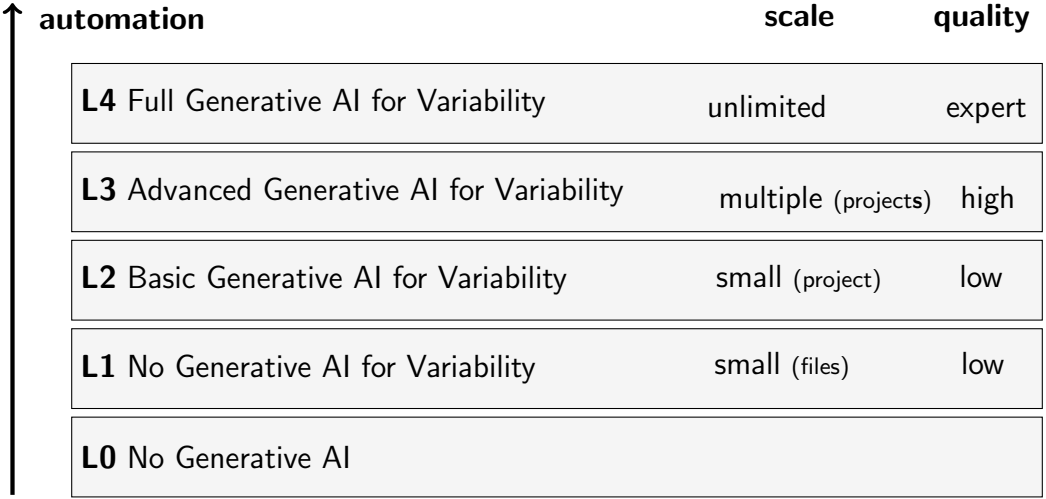
Automation Levels



Automation Levels



Automation Levels



How can we achieve increase?

How can we achieve increase?

Increase Scale

How can we achieve increase?

Increase Scale

- generic solutions
- SE solutions
- SPLE solutions (partitioning of large software projects)

How can we achieve increase?

Increase Scale

- generic solutions
- SE solutions
- SPLE solutions (partitioning of large software projects)

Increase Task

towards reaching productive usage

How can we achieve increase?

Increase Scale

- generic solutions
- SE solutions
- SPLE solutions (partitioning of large software projects)

Increase Task

towards reaching productive usage

- when and how integrate existing tools
- ensure traceability
- self-optimizations, multiple agents

How can we achieve increase?

Increase Scale

- generic solutions
- SE solutions
- SPLE solutions (partitioning of large software projects)

Increase Task

towards reaching productive usage

- when and how integrate existing tools
- ensure traceability
- self-optimizations, multiple agents

Increase Quality (and reliability)

How can we achieve increase?

Increase Scale

- generic solutions
- SE solutions
- SPLE solutions (partitioning of large software projects)

Increase Task

towards reaching productive usage

- when and how integrate existing tools
- ensure traceability
- self-optimizations, multiple agents

Increase Quality (and reliability)

decrease error probability
$\rightarrow$ pre-training, and fine-tuning

How can we achieve increase?

Increase Scale

- generic solutions
- SE solutions
- SPLE solutions (partitioning of large software projects)

Increase Task

towards reaching productive usage

- when and how integrate existing tools
- ensure traceability
- self-optimizations, multiple agents

Increase Quality (and reliability)

- decrease error probability
  → pre-training, and fine-tuning
- support existing (hidden) knowledge
  → prompting, model-driven techniques

How can we achieve increase?

Increase Scale

- generic solutions
- SE solutions
- SPLE solutions (partitioning of large software projects)

Increase Task

towards reaching productive usage

- when and how integrate existing tools
- ensure traceability
- self-optimizations, multiple agents

Increase Quality (and reliability)

decrease error probability
→ pre-training, and fine-tuning

support existing (hidden) knowledge
→ prompting, model-driven techniques

assessment strategies
→ novel datasets and metrics (real-world projects are too large, accuracy not sufficient)

How can we achieve increase?

Increase Scale

- generic solutions
- SE solutions
- SPLE solutions (partitioning of large software projects)

Increase Task

towards reaching productive usage

- when and how integrate existing tools
- ensure traceability
- self-optimizations, multiple agents

Increase Quality (and reliability)

decrease error probability
→ pre-training, and fine-tuning

support existing (hidden) knowledge
→ prompting, model-driven techniques

assessment strategies
→ novel datasets and metrics (real-world projects are too large, accuracy not sufficient)

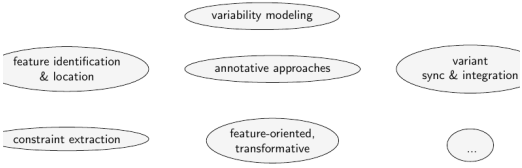
increase reliability
→ reasoning for results, certainty values

What's next?

exercise the levels for one SPL technique

explore combination of classical SPL techniques with Gen-AI support

Summary



language-specific, hard to analyze large systems, hard ot maintain consistently ...

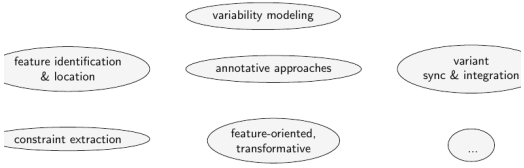
still insufficient genericity and automation

Business / Tech

A year after ChatGPT's release, the AI revolution is just beginning

By Catherine Thorbecke, CNN
8 minute read · Updated 10:32 AM EST, Thu November 30, 2023

Summary



language-specific, hard to analyze large systems, hard ot maintain consistently ...

still insufficient genericity and automation

Business / Tech

A year after ChatGPT's release, the AI revolution is just beginning

By Catherine Thorbecke, CNN
8 minute read · Updated 10:32 AM EST, Thu November 30, 2023

automation	scale	quality
L4 Full Generative AI for Variability	unlimited	expert
L3 Advanced Generative AI for Variability	multiple (projects)	high
L2 Basic Generative AI for Variability	small (project)	low
L1 No Generative AI for Variability	small (files)	low
L0 No Generative AI		

Increase Scale
- generic solutions
- SE solutions
- SPLE solutions (partitioning of large software projects)

Increase Task
towards reaching productive usage
- when and how integrate existing tools
- ensure traceability
- self-optimizations, multiple agents

Increase Quality (and reliability)
- decrease error probability
  → pre-training, and fine-tuning
- support existing (hidden) knowledge
  → prompting, model-driven techniques
- assessment strategies
  → novel datasets and metrics (real-world projects are too large, accuracy not sufficient)
- increase reliability
  → reasoning for results, certainty values

Thanks!

Feedback, Questions, ...?