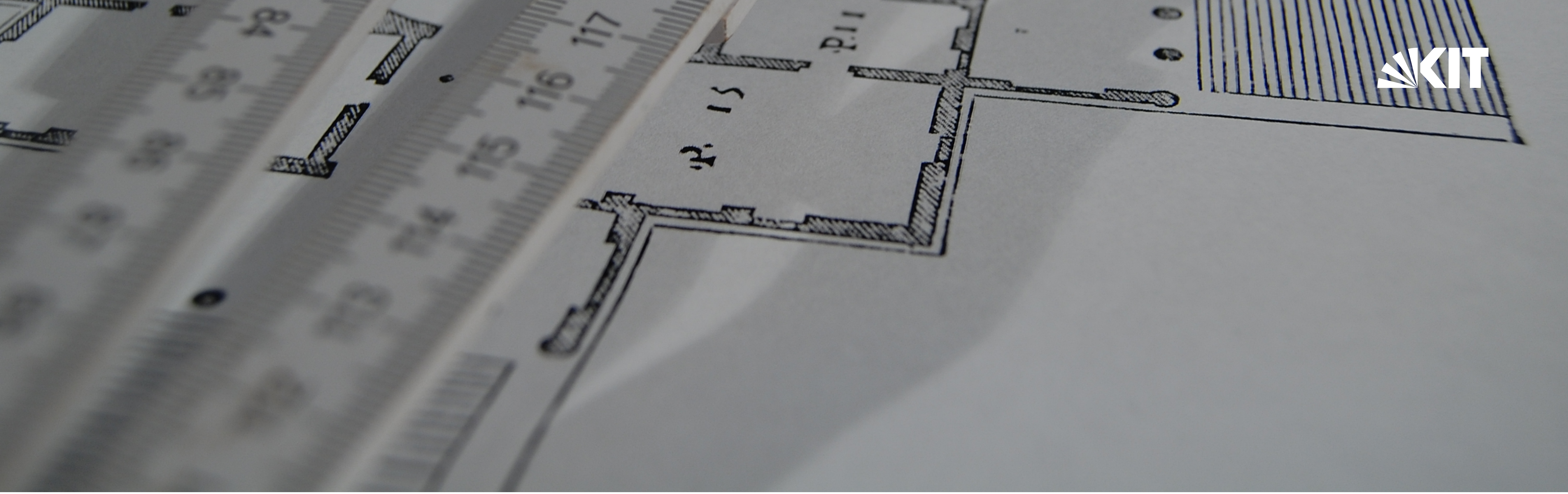


Automated Prompt Engineering for Traceability Link Recovery

Bachelor’s Thesis Proposal Presentation

Institute of Information Security and Dependability (KASTEL), Prof. Anne Koziolek
Modelling for Continuous Software Engineering (MCSE), Supervisor: M.Sc. Dominik Fuchß

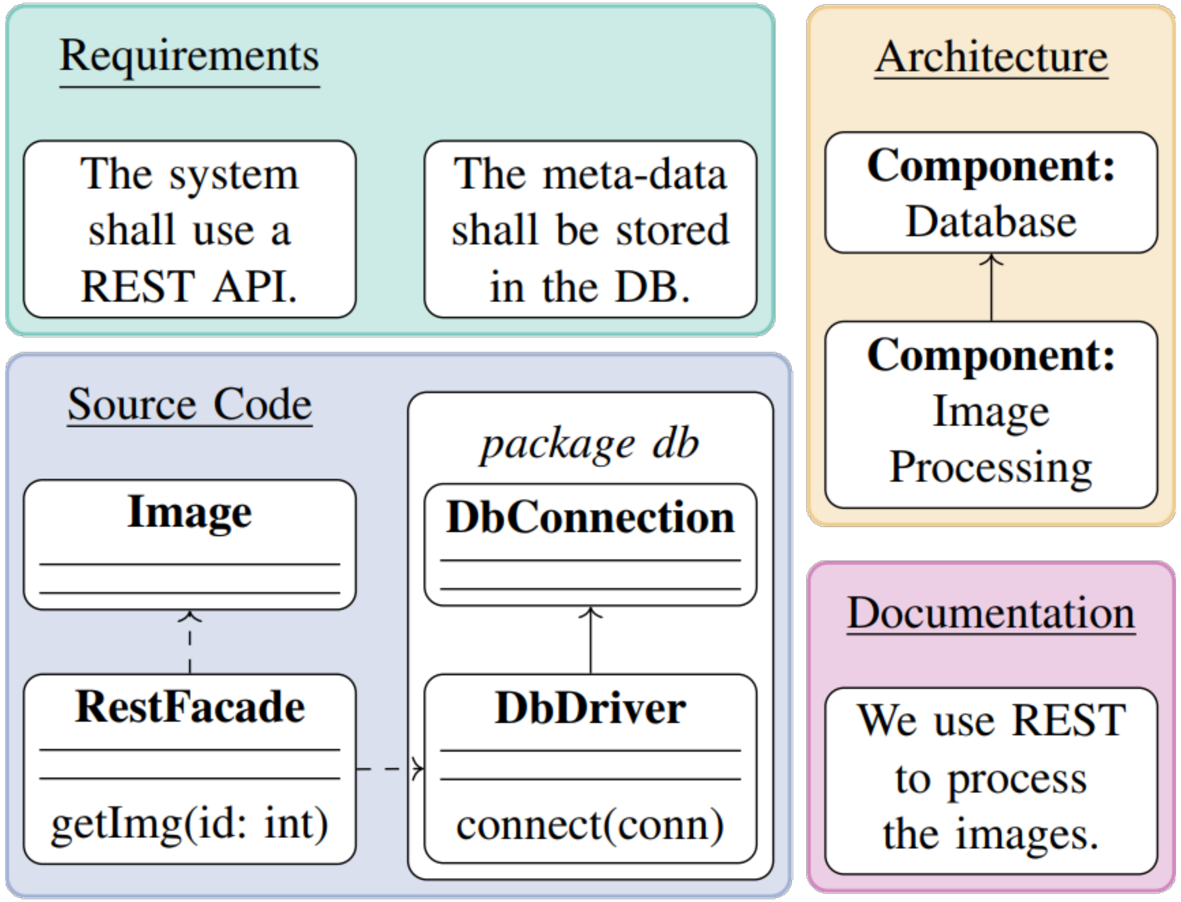
Daniel Schwab | 23. June 2025

PIBA

- **Problem**: Traceability Link Recovery with LiSSA does not perform well enough in larger RE2RE projects
- **Idea**: Improve recovery using automatically optimized prompts
- **Benefit**: Less manual labor required to find suitable prompts and increased recovery rates
- **Actions**: Implement automatic prompt engineering algorithm into the LiSSA framework and evaluate performance

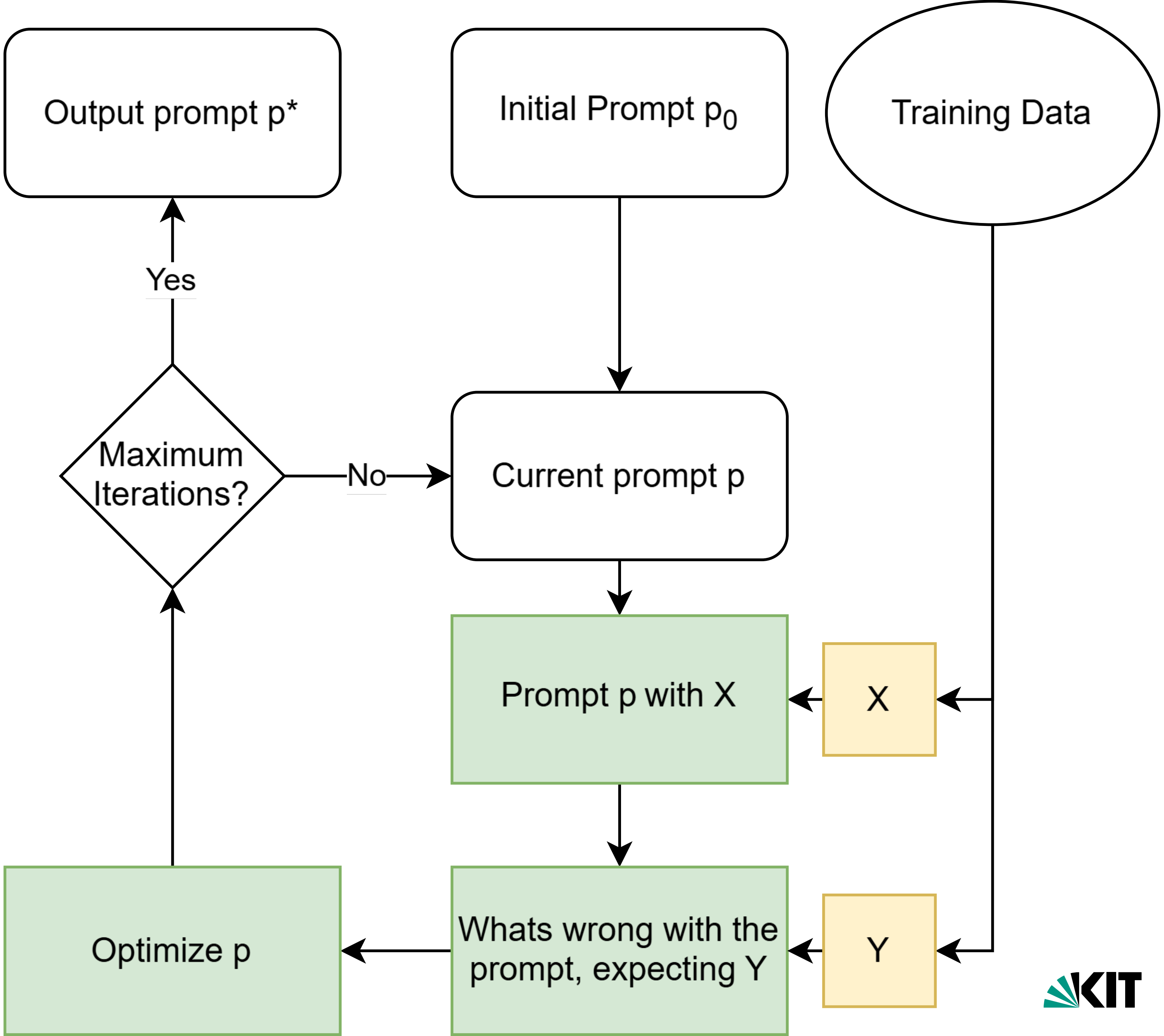
What is Traceability Link Recovery (TLR)

- Many artifacts are created during software development
- Often, inconsistencies will be present, such as naming
- Goal: Link elements across multiple domains or versions to ensure consistency and validation
- Image: Overview of possible artifacts for TLR by Fuchß et al. 2025

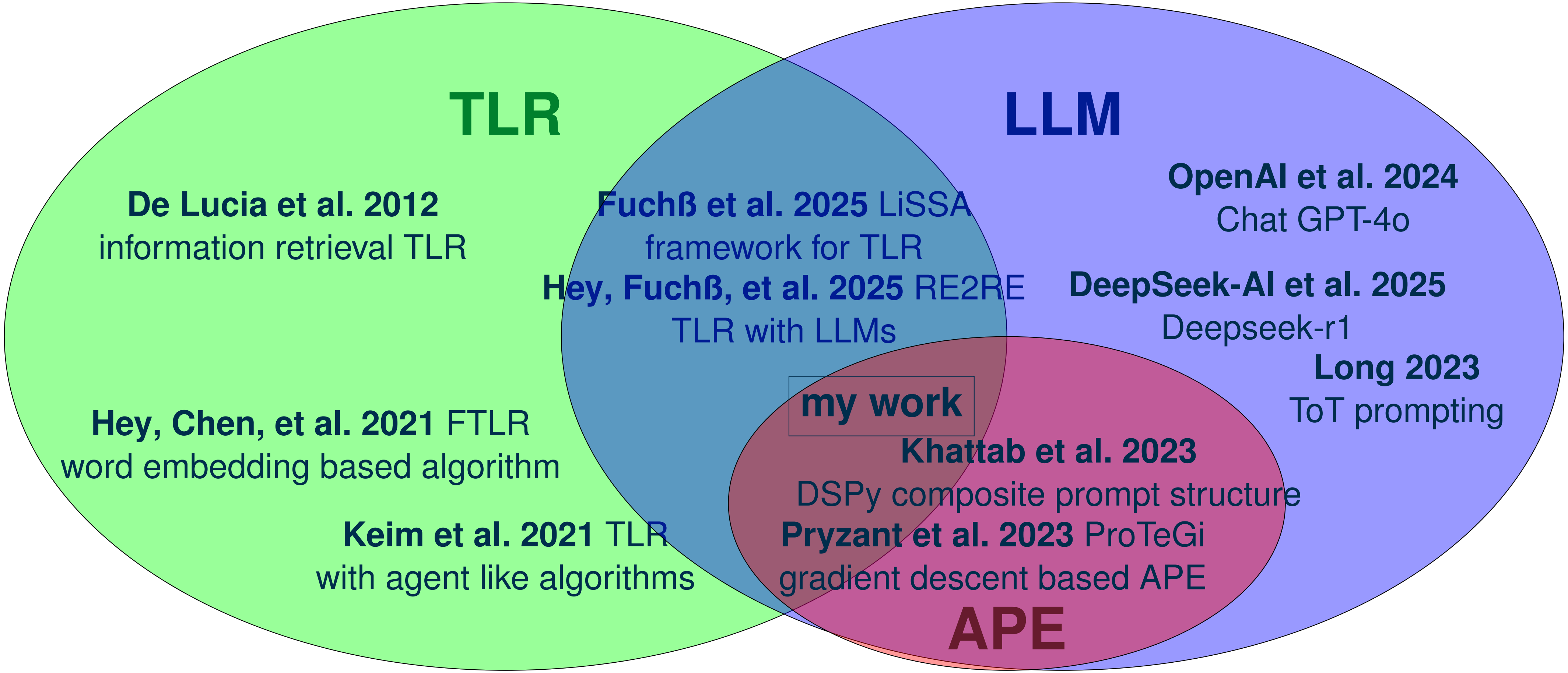


What is Automated Prompt Engineering

- Use the LLM to refine prompts instead of manually formulating them
- Improve initial prompt by training with a subset of the actual data
- Optimization prompt to fix previous shortcomings

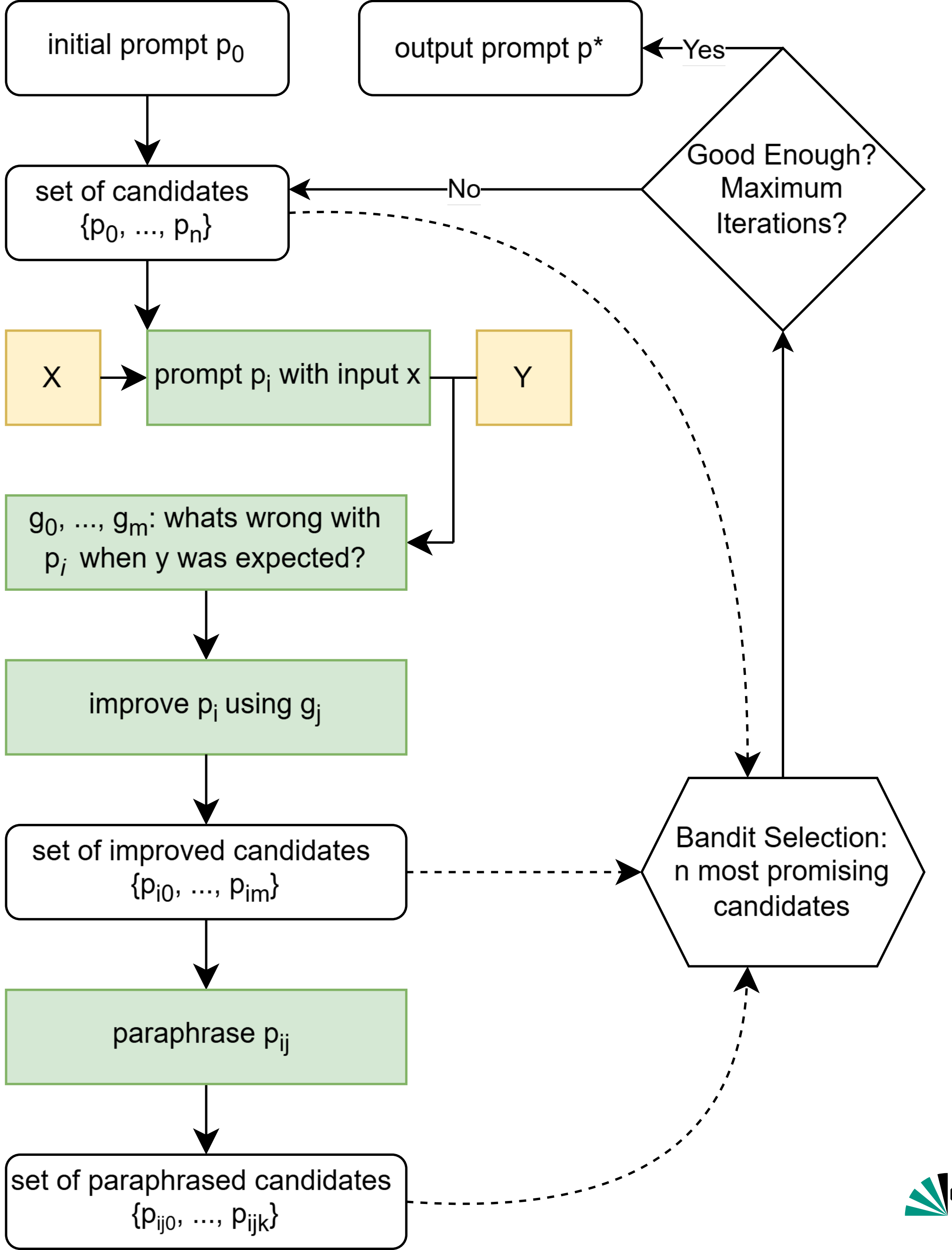


Related Work



Automated Prompt Engineering with Gradient Descent

- Improved APE algorithm by Pryzant et al. 2023
- Generate many new prompt candidates on each layer
- Steer them against the error direction using gradients
- Select the most promising candidates cheaply using a well-studied multiple-armed bandit algorithm



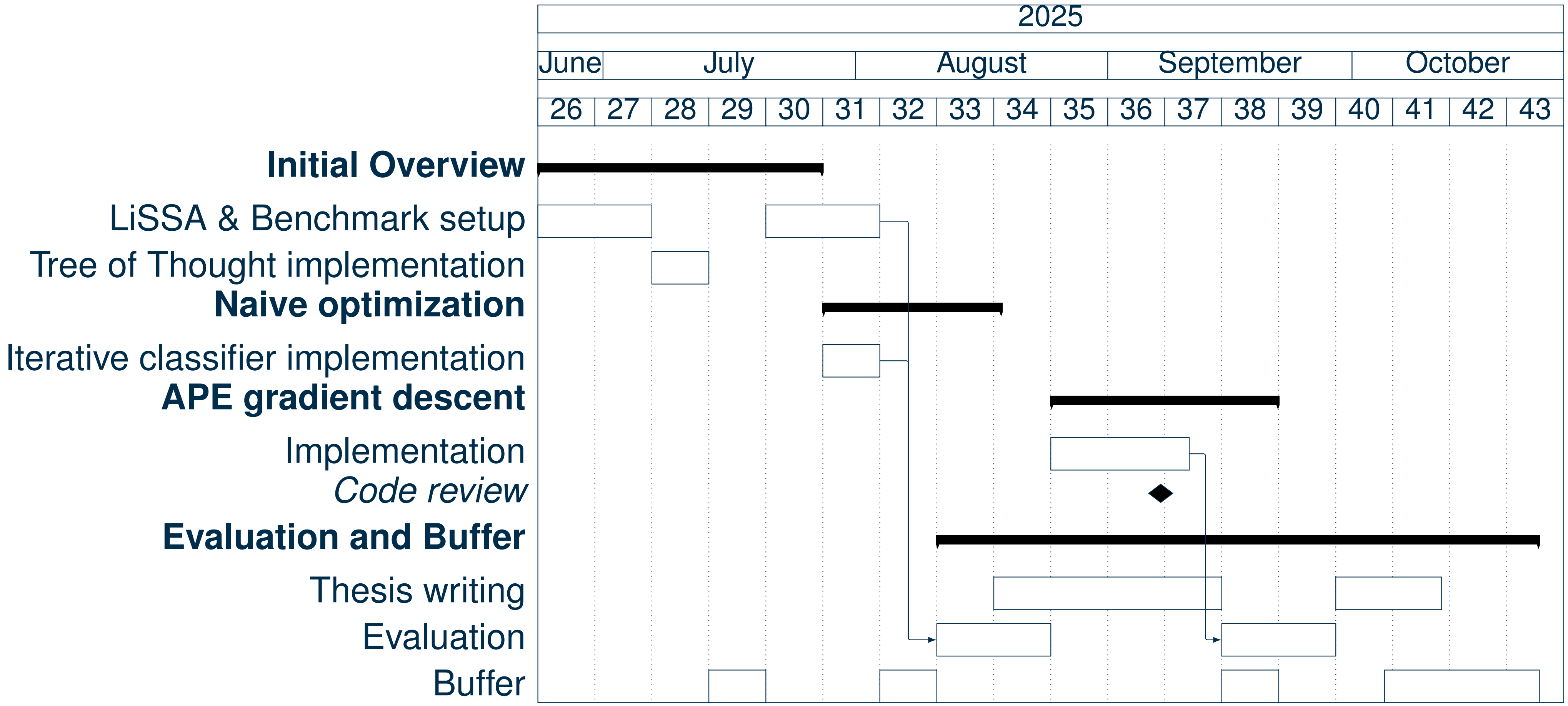
Evaluation

- Compare performance (precision, recall, F1-score, F2-score) against current manually designed zero-shot and chain-of-thought prompt
- Apply variations of different initial prompts and different LLMs to the optimization problem and compare outputs

Dataset	Metric	IR only	KISS GPT-4o	CoT GPT-4o
CCHIT	P.	.198	.234	.367
	R.	.157	.157	.138
	F1	.175	.188	.200
	F2	.164	.168	.158
Dronology	P.	.386	.394	.512
	R.	.695	.695	.655
	F1	.497	.503	.575
	F2	.600	.603	.620
Average including datasets that are omitted in this table	P.	.329	.340	.497
	R.	.500	.500	.458
	F1	.387	.401	.451
	F2	.445	.452	.452

Table: Reduced results by Hey, Fuchß, et al. 2025, Table 2

Thesis Schedule



Acronyms

- TLR: Traceability Link Recovery
- LLM: Large Language Model
- APE: Automatic Prompt Engineering
- RE2RE: Requirements-to-Requirements
- FTLR: Fine-grained Traceability Link Recovery
- LiSSA: Linking Software System Artifacts
- DSPy: Declarative Self-improving Python
- ProTeGi: Prompt Optimization with Textual Gradients
- ToT: Tree-of-Thought
- IR: information retrieval
- KISS: keep it short and simple
- CoT: Chain-of-Thought

Literatur I

[1] Andrea De Lucia et al. “Information Retrieval Methods for Automated Traceability Recovery”. In: *Software and Systems Traceability*. Ed. by Jane Cleland-Huang, Orlena Gotel, and Andrea Zisman. London: Springer, 2012, pp. 71–98. ISBN: 978-1-4471-2239-5. DOI: 10.1007/978-1-4471-2239-5_4. URL: https://doi.org/10.1007/978-1-4471-2239-5_4 (visited on 06/21/2025).

[2] DeepSeek-AI et al. *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. Jan. 22, 2025. DOI: 10.48550/arXiv.2501.12948. arXiv: 2501.12948 [cs]. URL: http://arxiv.org/abs/2501.12948 (visited on 06/21/2025). Pre-published.

[3] Dominik Fuchß et al. “LiSSA: Toward Generic Traceability Link Recovery through Retrieval-Augmented Generation”. In: *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. International Conference on Software Engineering (ICSE 2025), Ottawa, Kanada, 27.04.2025 – 03.05.2025. 2025, p. 723. DOI: 10.1109/ICSE55347.2025.00186. URL: https://publikationen.bibliothek.kit.edu/1000179816 (visited on 05/20/2025).

Literatur II

[4] Tobias Hey, Fei Chen, et al. "Improving Traceability Link Recovery Using Fine-grained Requirements-to-Code Relations". In: *2021 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. 2021 IEEE International Conference on Software Maintenance and Evolution (ICSME). Sept. 2021, pp. 12–22. DOI: 10.1109/ICSME52107.2021.00008. URL: https://ieeexplore.ieee.org/abstract/document/9609109 (visited on 06/14/2025).

[5] Tobias Hey, Dominik Fuchß, et al. "Requirements Traceability Link Recovery via Retrieval-Augmented Generation". In: *Requirements Engineering: Foundation for Software Quality*. Ed. by Anne Hess and Angelo Susi. Cham: Springer Nature Switzerland, 2025, pp. 381–397. ISBN: 978-3-031-88531-0. DOI: 10.1007/978-3-031-88531-0_27.

[6] Jan Keim et al. "Trace Link Recovery for Software Architecture Documentation". In: *Software Architecture*. Ed. by Stefan Biffl et al. Cham: Springer International Publishing, 2021, pp. 101–116. ISBN: 978-3-030-86044-8. DOI: 10.1007/978-3-030-86044-8_7.

Literatur III

[7] Omar Khattab et al. “DSPy: Compiling Declarative Language Model Calls into State-of-the-Art Pipelines”. In: The Twelfth International Conference on Learning Representations. Oct. 13, 2023. URL: https://openreview.net/forum?id=sY5N0zY5Od (visited on 06/15/2025).

[8] Jieyi Long. *Large Language Model Guided Tree-of-Thought*. May 15, 2023. DOI: 10.48550/arXiv.2305.08291. arXiv: 2305.08291 [cs]. URL: http://arxiv.org/abs/2305.08291 (visited on 06/02/2025). Pre-published.

[9] OpenAI et al. *GPT-4o System Card*. Oct. 25, 2024. DOI: 10.48550/arXiv.2410.21276. arXiv: 2410.21276 [cs]. URL: http://arxiv.org/abs/2410.21276 (visited on 06/21/2025). Pre-published.

[10] Reid Pryzant et al. “Automatic Prompt Optimization with “Gradient Descent” and Beam Search”. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. EMNLP 2023. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 7957–7968. DOI: 10.18653/v1/2023.emnlp-main.494. URL: https://aclanthology.org/2023.emnlp-main.494/ (visited on 06/02/2025).