Atharva Naik

arnaik@andrew.cmu.edu   atharva-naik   Atharva Naik
https://atharva-naik.github.io/   ` Google Scholar

Education

2024 – · · · · **Ph.D. Language Technologies, Carnegie Mellon University**.
Advisors: Carolyn Rose, Daniel Fried
GPA: 4.14/4

2022 – 2024 **M.S. Language Technologies, Carnegie Mellon University**
GPA: 4.11/4

2018 – 2022 **B.Tech. Computer Science, Indian Institute of Technology, Kharagpur**
GPA: 9.66/10

Research

Statement: My research focuses on developing synthetic data generation and post-training methods to build Large Language Models of Code that can adapt to contextual knowledge, handle shifting data distributions, and reason dynamically at test time. I aim to apply these capabilities to (1) code quality and review, (2) reasoning, and (3) code security and AI safety.

Experience

2025 **Research Intern, Oracle Labs** MetaLint: Generalizable Idiomatic Code Quality Analysis with Instruction Following and Easy-to-Hard Generalization

2022 – 2024 **Research Assistant, Carnegie Mellon University** Reinforcement Learning for Code Quality & Review, Reasoning for AI Safety, Programming by Examples for Reasoning

2021 **Research Intern, Technische Universität Darmstadt** Neural Network Architecture for Faithful Interpretability in NLP.

Research Intern, Adobe RL agent for Creative Human-Human Collaboration.

Research Intern, University of Alberta Neuro-Symbolic Fuzzy Logic-based Reasoning for Explainable Natural Language Inference.

2019-2020 **Student Researcher, Autonomous Ground Vehicle (AGV) Group** Path Planning and Localization for Autonomous Driving.

Publications

Conference Publications

1. **A. Naik**, D. Agrawal, H. Sng, *et al.*, "Programming by example meets historical linguistics: A large language model based approach to sound law induction," in ***ACL***, 2025, pp. 29 628–29 647. URL: `https://aclanthology.org/2025.acl-long.1432/`.

2. **A. Naik**, M. Alenius, D. Fried, and C. Rose, "CRScore: Grounding automated evaluation of code review comments in code claims and smells," in *NAACL*, 2025, pp. 9049–9076. URL: `https://aclanthology.org/2025.naacl-long.457/`.

3. S. Gandhi, **A. Naik**, Y. Xie, and C. Rose, "An empirical study on strong-weak model collaboration for repo-level code generation," in ***EMNLP*** *(to appear)*, 2025.

4. **A. Naik**, J. R. Yin, A. Kamath, *et al.*, "Generating situated reflection triggers about alternative solution paths: A case study of generative ai for computer-supported collaborative learning," in ***AIED***, 2024.

5. **A. Naik**, J. R. Yin, A. Kamath, *et al.*, "Providing tailored reflection instructions in collaborative learning using large language models," in ***BJET***, vol. 56, 2024, pp. 531–550.
6. A. Rao, S. Vashistha, **A. Naik**, S. Aditya, and M. Choudhury, "Tricking LLMs into Disobedience: Understanding, Analyzing, and Preventing Jailbreaks," in ***LREC-COLING***, 2024.
7. **A. Naik**, S. Das, J. Vedurada, and S. Aditya, "Sync: A structurally guided hard negative curriculum for generalizable neural code search," in ***AACL***, 2023.
8. Z. Wu, Z. X. Zhang, **A. Naik**, Z. Mei, M. Firdaus, and L. Mou, "Weakly Supervised Explainable Phrasal Reasoning with Neural Fuzzy Logic," in ***ICLR***, 2023.
9. Y. Xie, **A. Naik**, D. Fried, and C. Rose, "CMTrans: Improving Code Translation with Comparable Corpora and Multiple References," in ***EMNLP Findings***, 2023.
10. S. Bv, J. A. Patel, **A. Naik**, Y. Butala, S. Sharma, and N. Chhaya, "Towards Enabling Synchronous Digital Creative Collaboration: Codifying Conflicts in Co-Coloring," in ***CHI Extended Abstracts***, 2022.
11. B. Santra, S. Roychowdhury, A. Mandal, *et al.*, "Representation Learning for Conversational Data using Discourse Mutual Information Maximization," in ***NAACL***, 2022.
12. Y. Wang, S. Mishra, P. Alipoormolabashi, *et al.*, "Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks," in ***EMNLP***, 2022.
13. R. Mukherjee, **A. Naik**, S. Poddar, S. Dasgupta, and N. Ganguly, "Understanding the Role of Affect Dimensions in Detecting Emotions from Tweets: A Multi-task Approach," in ***SIGIR***, 2021.

Preprints

1. **A. Naik**, A. Xie, A. Rao, *et al.*, "Secure and useful models are reasonable: Aligning code models via utility-preserving reasoning,"
2. **A. Naik**, D. Agrawal, M. Kapadnis, *et al.*, "Pbebench: A multi-step programming by examples reasoning benchmark inspired by historical linguistics," *arXiv preprint arXiv:2505.23126*, 2025.
3. **A. Naik**, L. Baghel, D. Govindarajan, D. Agrawal, D. Fried, and C. Rose, "Metalint: Generalizable idiomatic code quality analysis through instruction-following and easy-to-hard generalization," *arXiv preprint arXiv:2507.11687*, 2025.
4. M. N. Kapadnis, **A. Naik**, and C. Rose, "Crscore++: Reinforcement learning with verifiable tool and ai feedback for code review," *arXiv preprint arXiv:2506.00296*, 2025.
5. **A. Naik**, "On the limitations of embedding based methods for measuring functional correctness for code generation," *arXiv preprint arXiv:2405.01580*, 2024.
6. **A. Naik**, K. Zhang, N. Robinson, *et al.*, "Can large language models code like a linguist?: A case study in low resource sound law induction," 2024. arXiv: `2406.12725 [cs.CL]`.

Projects

MetaLint:

- Developed MetaLint, an instruction-following framework that detects and corrects non-idiomatic Python and Java code by aligning with evolving PEP and JEP standards. Utilized instruction fine-tuning and Direct Preference Optimization (DPO) to enable test-time adaptation to novel idioms.
- Achieved state-of-the-art recall of 70.43% for idiom detection and 26.73% for line-level localization on a challenging PEP-based benchmark, performing similarly to models like o3-mini despite using only a fine-tuned 4B parameter model.

Projects (continued)

- Demonstrated robust generalization across programming languages, model families, linters, and reasoning settings, highlighting MetaLint's adaptability to diverse code quality analysis tasks.

PBEBench:

- Developed PBEBench, a scalable, contamination-free benchmark generator for evaluating LLMs' inductive reasoning and planning in a knowledge free setting via multi-step string rewrite tasks, controlling difficulty with required rewrite steps, ordering constraints, and number of string examples.
- Analyzed reasoning bottlenecks and test-time scaling strategies, measuring performance saturation across top open and closed-source models like gpt-oss-120B and GPT-5.
- Found that while state-of-the-art reasoning models outperform non-reasoning LLMs, none yet achieve expert human-level performance on realistic historical linguistics tasks.

Programming by Examples Meets Historical Linguistics:

- Framed sound law induction (SLI) in historical linguistics as a programming-by-examples task and leveraged code LLMs to generate and evaluate Python sound law program, using synthetic data to improve their accuracy in explaining sound change examples.
- Showed that logically structured synthetic programs with realistic inputs outperform training on real sound law programs, highlighting key factors about the training data distribution for successful SLI.
- Accepted as an ACL long track oral presentation (top 10% of papers).

CRScore:

- Developed a reference-free metric for code review comment generation by combining LLMs and static analysis to produce pseudo-reference rubrics and interpretably scoring coverage across key quality dimensions using semantic embedding based similarity.
- Achieved second-highest correlation with human judgment after GPT-4o using a 7B Magicoder model, outperforming it as a direct judge and reducing evaluation complexity from $O(m \times d)$ to $O(d)$ (m - number of models evaluated, d - number of datapoints).
- Accepted as a NAACL long track oral presentation.

Skills

Coding	Python (expert), C/C++, Bash (familiar), Javascript (novice)
Frameworks	vLLM, DeepSpeed, PyTorch, HuggingFace, Fairseq, NLTK, spaCy, Tensorflow, FastAPI, Flask, Django, PyQt5, Jupyterlab, OpenCV, Git

Achievements

2025	**Amazon Trusted AI Challenge Finalist**, Led the Carnegie Mellon University team to the finals as one of the top four defender teams for developing secure code LLMs.
	ACL 2025 Oral Presentation, One of 243 papers selected from over 3,000 accepted submissions.
2024	**Amazon Trusted AI Challenge Top 10 Team**, Selected among 90 applicant teams and awarded a $250K research grant as team lead for CMU to advance the development of secure code LLMs.
	Best Paper & Best Student Paper Nominations, Artificial Intelligence in Education (AIED 2024).
2022	**2nd Place**, Deep Learning Labs OpenAI GPT-3 Hackathon.

Achievements (continued)

2021	🔖	**DAAD WISE Scholarship**.
	🔖	**MITACS Globalink Scholarship**.
	🔖	**Bronze**, Inter IIT Technology Meet (IIT Kharagpur contingent).
2019	🔖	**2nd Place**, Intelligent Ground Vehicle Competition (IGVC).
2018	🔖	All India Rank **1248** in JEE Advanced and **1618** in JEE Mains among 1M candidates.
	🔖	**Kishore Vaigyanik Protsahan Yojana (KVPY) Scholarship**.