

AI Link Collection Report - 2025-07-08

Generated on July 08, 2025 at 11:16 AM

Total Items: 2

Report Statistics	
Total Links Processed	2
Generation Date	2025-07-08
Generation Time	11:16:10

■ Link Collection Details

1. The QuArch Experiment: Crowdsourcing a Machine Learning Dataset for Computer Architecture

■ <https://www.sigarch.org/the-quarch-experiment-crowdsourcing-a-machine-learning...>

■ Full Article Content:

The QuArch Experiment: Crowdsourcing a Machine Learning Dataset for Computer Architecture The QuArch Experiment: Crowdsourcing a Machine Learning Dataset for Computer Architecture by Shvetank Prakash and Vijay Janapa Reddi on Jan 7, 2025 | Tags: AI Agents, Benchmarks, Datasets, Machine Learning Shvetank Prakash and Vijay Janapa Reddi on Jan 7, 2025 | Tags: AI Agents Benchmarks Datasets Machine Learning.et_post_meta_wrapper Introduction Introduction The rise of large language models (LLMs) and generative artificial intelligence (GenAI) presents new opportunities to build innovative tools and is already enabling revolutionary AI-based tools in various domains. However, a significant gap remains in the language model (LM) knowledge required to deliver high-quality, AI-based tools for engineers. Figure 1 illustrates the performance of LMs across various subjects in the MMLU-Pro question-answering benchmark, with engineering consistently ranking among the lowest-performing topics. The rise of large language models (LLMs) and generative artificial intelligence (GenAI) presents new opportunities to build innovative tools and is already enabling revolutionary AI-based tools in various domains. However, a significant gap remains in the language model (LM) knowledge required to deliver high-quality, AI-based tools for engineers. Figure 1 illustrates the performance of LMs across various subjects in the MMLU-Pro question-answering benchmark, with engineering consistently ranking among the lowest-performing topics. Figure 1: Average performance of language models across various subjects in the MMLU-Pro benchmark. Figure 1: Average performance of language models across various subjects in the MMLU-Pro benchmark. Benchmarks play an important role in driving progress within our field (for better or for worse), influencing both advancements and potential setbacks. Similarly, high-quality datasets are essential for developing and enhancing machine learning (ML) methods. Yet computer architecture is notably underserved in this regard. Benchmarks play an important role in driving progress within our field (for better or for worse), influencing both advancements and potential setbacks. Similarly, high-quality datasets are essential for developing and enhancing machine learning (ML) methods. Yet computer architecture is notably underserved in this regard. We lack domain-specific datasets that are necessary to benchmark progress and foster AI-driven innovation. This striking gap calls for attention given architecture's historical relationship with AI. We lack domain-specific datasets that are necessary to benchmark progress and foster AI-driven innovation. This striking gap calls for attention given architecture's historical relationship with AI. This blog post explores what today's language models understand about computer architecture. Given their relatively modest performance, we introduce QuArch v1.0 as the first step in a broader initiative to create an open-source, community-driven dataset to help usher in an AI-driven computer system design era. At the end of this blog, you'll find a call to participate, encouraging you to contribute your expertise and shape the future of this initiative. This blog post explores what today's language models understand about computer architecture. Given their relatively modest performance, we introduce QuArch v1.0 as the first step in a broader initiative to create an open-source, community-driven dataset to help usher in an AI-driven computer system design era. At the end of this blog, you'll find a call to participate, encouraging you to contribute your expertise and shape the future of this initiative. The State of AI in Computer Architecture. The State of AI in Computer Architecture AI has long been a powerful tool in computer architecture, particularly for optimizing various hardware and software components such as branch

predictors, memory controllers, resource allocation, coherence, cache allocation, replacement policies, scheduling, accelerator design, cloud resource sharing, power consumption, and security. ML has enhanced virtually every aspect of computing systems. AI has long been a powerful tool in computer architecture, particularly for optimizing various hardware and software components such as branch predictors memory controllers resource allocation coherence cache allocation replacement policies scheduling accelerator design, cloud resource sharing power consumption security. ML has enhanced virtually every aspect of computing systems. The most recent breakthroughs in LLMs and GenAI, built upon decades of architectural innovation, now stand to revolutionize this relationship further. Consequently, we are starting to see new initiatives like the NSF GenAI4HW & HW4GenAI Workshop at MICRO 2024, the NSF ImageNets4EDA Workshop, and the Architecture 2.0 workshop reflect the growing interest in leveraging these technologies to create a virtuous cycle of innovation. The most recent breakthroughs in LLMs and GenAI, built upon decades of architectural innovation, now stand to revolutionize this relationship further. Consequently, we are starting to see new initiatives like the NSF GenAI4HW & HW4GenAI Workshop at MICRO 2024 NSF ImageNets4EDA Workshop, and the Architecture 2.0 workshop reflect the growing interest in leveraging these technologies to create a virtuous cycle of innovation. This momentum is prominent in electronic design automation (EDA), where researchers develop specialized datasets and algorithms for chip design tasks (Liu et al., Chin et al.). Recent work on Register-Transfer Level (RTL) generation (Liu et al., Blocklove et al., Zhang et al., Tsai et al.) and hardware verification & security (Cosler et al., Ahmad et al., Afsharmazayejani et al.) demonstrate how domain-specific data can enable ML to tackle complex hardware design challenges. This momentum is prominent in electronic design automation (EDA), where researchers develop specialized datasets and algorithms for chip design tasks (Liu et al. Chin et al.). Recent work on Register-Transfer Level (RTL) generation (Liu et al. Blocklove et al. Zhang et al. Tsai et al.) and hardware verification & security (Cosler et al. Ahmad et al. Afsharmazayejani et al.) demonstrate how domain-specific data can enable ML to tackle complex hardware design challenges. While EDA focuses on implementation-level tasks, we lack similar resources for system-level reasoning, which is essential for many tasks in computer architecture. EDA machine learning datasets enable automation of RTL generation and verification. We need something that focuses on broader architectural reasoning to inform fundamental design decisions. This creates a complementary pipeline: architects need enhanced AI tools to explore and assess high-level designs that flow into EDA tools for implementation. Ultimately, we need high-quality, large-scale data that captures architectural expertise to enable this vision. While EDA focuses on implementation-level tasks, we lack similar resources for system-level reasoning, which is essential for many tasks in computer architecture. EDA machine learning datasets enable automation of RTL generation and verification. We need something that focuses on broader architectural reasoning to inform fundamental design decisions. This creates a complementary pipeline: architects need enhanced AI tools to explore and assess high-level designs that flow into EDA tools for implementation. Ultimately, we need high-quality, large-scale data that captures architectural expertise to enable this vision. Introducing QuArch. Introducing QuArch To address the dataset gap, we discuss QuArch, a dataset aimed at higher-level architectural decisions—the kind of reasoning that precedes and informs detailed implementation. QuArch (pronounced ‘quark’) is a specialized question-answering (QA) dataset designed to evaluate and enhance the computer architecture knowledge embedded in LMs. Its goal is to assess (and help bridge) the knowledge gap and drive AI-driven innovation in architecture. The QAs follow a multiple-choice format similar to other ML datasets to ensure educational value and feasible assessment. Below, we provide a simple example of QA: To address the dataset gap, we discuss QuArch, a dataset aimed at higher-level architectural decisions—the kind of reasoning that precedes and informs detailed implementation. QuArch (pronounced ‘quark’) is a specialized question-answering (QA) dataset designed to evaluate and enhance the computer architecture knowledge embedded in LMs. Its goal is to assess (and help bridge) the knowledge gap and drive AI-driven innovation in architecture. The QAs follow a multiple-choice format similar to other ML datasets to ensure educational value and feasible assessment. Below, we provide a simple example of QA: Question 1 Q: Which of the following is a key feature of the Reduced Instruction Set Computer (RISC) architecture? • Complex instructions that can perform multiple operations in a single cycle Complex instructions that can perform multiple operations in a

single cycle • A large number of general-purpose registers to reduce memory access

A large number of general-purpose registers to reduce memory access • A small and fixed instruction format to simplify hardware design

A small and fixed instruction format to simplify hardware design • A memory architecture that prioritizes cache size over latency

A memory architecture that prioritizes cache size over latency

When evaluating LLM responses to the above question, subtle yet meaningful differences emerge. Models like GPT-4 and Claude provided diverging answers: some chose option B, others selected option C, and some attempted to rationalize between the two. Others gave outright incorrect answers, highlighting diversity in interpretive approaches. These differences arise from variations in how models interpret the question's focus, prioritize architectural features, and articulate reasoning. This example shows that even advanced LLMs can struggle with nuanced, domain-specific questions, which emphasizes the need for datasets like QuArch to enhance their understanding of the complex reasoning required in computer architecture.

When evaluating LLM responses to the above question, subtle yet meaningful differences emerge. Models like GPT-4 and Claude provided diverging answers: some chose option B, others selected option C, and some attempted to rationalize between the two. Others gave outright incorrect answers, highlighting diversity in interpretive approaches. These differences arise from variations in how models interpret the question's focus, prioritize architectural features, and articulate reasoning. This example shows that even advanced LLMs can struggle with nuanced, domain-specific questions, which emphasizes the need for datasets like QuArch to enhance their understanding of the complex reasoning required in computer architecture. The 1,500 QAs in QuArch v0.1 have been expertly labeled by industry experts and Ph. D. students in computer architecture. They cover a wide range of areas, as shown in Figure 2. The topics included represent the diverse scope of modern computer architecture. Processor architecture QAs comprise 32% of the dataset, with memory systems at 22% and interconnection networks at 10%. The 1,500 QAs in QuArch v0.1 have been expertly labeled by industry experts and Ph. D. students in computer architecture. They cover a wide range of areas, as shown in Figure 2. The topics included represent the diverse scope of modern computer architecture. Processor architecture QAs comprise 32% of the dataset, with memory systems at 22% and interconnection networks at 10%. Figure 2: Distribution of computer architecture topics in QuArch v0.1. Figure 2: Distribution of computer architecture topics in QuArch v0.1.

Why Question-Answering for Architecture? Computer architecture encompasses many complex challenges spanning the entire technology stack. Addressing these cross-stack challenges requires synthesizing high-level insights with low-level implementations, demanding a deep understanding of workload behavior, architectural constraints, and system-level interactions. Computer architecture encompasses many complex challenges spanning the entire technology stack. Addressing these cross-stack challenges requires synthesizing high-level insights with low-level implementations, demanding a deep understanding of workload behavior, architectural constraints, and system-level interactions. Question-answering systems are particularly well-suited to support these multifaceted needs. QA tasks inherently require knowledge retrieval, reasoning, complex situational modeling, and input interpretation & manipulation—capabilities essential for solving architectural problems. By training LMs on QA datasets like QuArch, we can enhance their ability to assist in tasks ranging from optimizing cache hierarchies to designing efficient interconnects. Question-answering systems are particularly well-suited to support these multifaceted needs. QA tasks inherently require knowledge retrieval, reasoning, complex situational modeling, and input interpretation & manipulation — capabilities essential for solving architectural problems. By training LMs on QA datasets like QuArch, we can enhance their ability to assist in tasks ranging from optimizing cache hierarchies to designing efficient interconnects. Historically, QA datasets have been instrumental in advancing domain-specific, LLM-based tools across fields like medicine, math, law, finance, and software engineering. These tools enable applications such as diagnostic assistance, equation solving, legal research, financial modeling, and code debugging. Similar opportunities exist to build exciting agentic workflows for computer architecture. Still, the availability of data—or the lack thereof—remains a critical factor in revolutionizing traditional tools and unlocking new paradigms. Historically, QA datasets have been instrumental in advancing domain-specific, LLM-based tools across fields like medicine, finance, and software engineering. These tools enable applications such as diagnostic assistance, equation solving, legal research, financial modeling, and code

debugging. Similar opportunities exist to build exciting agentic workflows for computer architecture. Still, the availability of data—or the lack thereof—remains a critical factor in revolutionizing traditional tools and unlocking new paradigms.

Evaluating Architectural Knowledge of Language Models

Using QuArch, we systematically evaluated language models' computer architecture knowledge across multiple topics. Our evaluation revealed both promising results and concerning gaps. Using QuArch, we systematically evaluated language models' computer architecture knowledge across multiple topics. Our evaluation revealed both promising results and concerning gaps. The top-performing model, shown in Figure 3, achieved an 84% accuracy rate on basic architectural concepts, demonstrating imperfect but strong foundational knowledge. However, this performance consistently dropped across some topics. Three topics, in particular, caused the biggest drops in accuracy across models: Interconnection Networks, Memory Systems, and Benchmarking & Measurement. See details here. The top-performing model, shown in Figure 3, achieved an 84% accuracy rate on basic architectural concepts, demonstrating imperfect but strong foundational knowledge. However, this performance consistently dropped across some topics. Three topics, in particular, caused the biggest drops in accuracy across models: Interconnection Networks, Memory Systems, and Benchmarking & Measurement. See details Figure 3.

Comparison of model accuracies on QuArch v0.1 QAs. See details here. Figure 3.

Comparison of model accuracies on QuArch v0.1 QAs. See details More notably, performance dropped significantly by 12% when using smaller, open-source models (under 10B parameters). Furthermore, as discussed in the next section, we empirically observed that this gap becomes particularly pronounced when transitioning from factual questions to complex reasoning tasks. More notably, performance dropped significantly by 12% when using smaller, open-source models (under 10B parameters). Furthermore, as discussed in the next section, we empirically observed that this gap becomes particularly pronounced when transitioning from factual questions to complex reasoning tasks. These results show that while current models show promise in understanding basic architectural concepts, they still have weaknesses that will limit their ability to handle the complex reasoning and deep technical understanding that computer architects apply daily. Moreover, the disparity between small and large model performance suggests that effective, AI-based architecture tools will likely demand substantial resources or require greater specialization. These results show that while current models show promise in understanding basic architectural concepts, they still have weaknesses that will limit their ability to handle the complex reasoning and deep technical understanding that computer architects apply daily. Moreover, the disparity between small and large model performance suggests that effective, AI-based architecture tools will likely demand substantial resources or require greater specialization.

Beyond Factual QA: Towards Complex Architectural Reasoning

Beyond QuArch's 1,500 QAs, which primarily evaluate the domain knowledge embedded in LMs, we also conducted a case study analyzing a publicly available Ph. D. qualifying exam question that demands advanced reasoning skills. Specifically, we examined optimizations like vectorization. Our findings revealed that even state-of-the-art models, such as GPT-4o, needed help interpreting loop dependencies and conditions essential for proper compiler vectorization. For instance, models misinterpreted simple dependency differences between $i+1$ and $i-1$, highlighting significant gaps in their reasoning capabilities. Beyond QuArch's 1,500 QAs, which primarily evaluate the domain knowledge embedded in LMs, we also conducted a case study analyzing a publicly available Ph. D. qualifying exam question that demands advanced reasoning skills. Specifically, we examined optimizations like vectorization. Our findings revealed that even state-of-the-art models, such as GPT-4o, needed help interpreting loop dependencies and conditions essential for proper compiler vectorization. For instance, models misinterpreted simple dependency differences between $i+1$ and $i-1$, highlighting significant gaps in their reasoning capabilities. The above case study demonstrates the need for more sophisticated datasets and targeted training to enhance model performance in complex architectural tasks. To assess and improve reasoning capabilities, we must move beyond factual QAs to develop complex, multi-step problems. These real-world problems must evaluate architectural reasoning across dimensions like performance analysis, design exploration, and hardware-software co-design. Building this comprehensive evaluation framework requires diverse expertise and architectural challenges that only our community can provide, which brings us to the next stage. The above case study demonstrates the

need for more sophisticated datasets and targeted training to enhance model performance in complex architectural tasks. To assess and improve reasoning capabilities, we must move beyond factual QAs to develop complex, multi-step problems. These real-world problems must evaluate architectural reasoning across dimensions like performance analysis, design exploration, and hardware-software co-design. Building this comprehensive evaluation framework requires diverse expertise and architectural challenges that only our community can provide, which brings us to the next stage. Join the QuArch Community: Contribute to QuArch v1.0. Join the QuArch Community: Contribute to QuArch v1.0 Our initial QuArch development efforts engaged 50 undergraduate students primarily from electrical engineering and computer science backgrounds, offering valuable insights but revealing significant QA validation challenges due to varying levels of expertise. This experience highlighted the need for a broader, community-driven approach to develop a robust, high-quality dataset—a strategy that has proven successful across numerous domains. Crowdsourced datasets have consistently driven ML innovation, from ImageNet’s transformative impact on computer vision to Mozilla Common Voice’s advancements in speech recognition. Our initial QuArch development efforts engaged 50 undergraduate students primarily from electrical engineering and computer science backgrounds, offering valuable insights but revealing significant QA validation challenges due to varying levels of expertise. This experience highlighted the need for a broader, community-driven approach to develop a robust, high-quality dataset—a strategy that has proven successful across numerous domains. Crowdsourced datasets have consistently driven ML innovation, from ImageNet’s transformative impact on computer vision Mozilla Common Voice’s advancements in speech recognition. How You Can Help How You Can Help We invite computer architects, educators, industry professionals, and enthusiasts to contribute through our structured three-stage roadmap: We invite computer architects, educators, industry professionals, and enthusiasts to contribute through our structured three-stage roadmap: • Stage 1: Expanding Fundamental Domain Knowledge: Annotate QAs on concepts like processor execution, memory hierarchy, and parallelism, building the critical foundation for advanced reasoning and design. Stage 1: Expanding Fundamental Domain Knowledge: Annotate QAs on concepts like processor execution, memory hierarchy, and parallelism, building the critical foundation for advanced reasoning and design. • Stage 2: Input Interpretation & Manipulation: Curate QAs based on tasks like analyzing instruction traces and optimizing code for engineering applications. We aim to create diverse practical questions from academia, industry, and public platforms by crowdsourcing questions. Stage 2: Input Interpretation & Manipulation: Curate QAs based on tasks like analyzing instruction traces and optimizing code for engineering applications. We aim to create diverse practical questions from academia, industry, and public platforms by crowdsourcing questions. • Stage 3: Reasoning & Complex Situational Modeling: Craft reasoning-driven QAs that mirror real-world architectural challenges and require retrieval, reasoning, planning, and modeling skills. This stage enables models to reason about trade-offs and system-level design scenarios to push AI’s capabilities. Stage 3: Reasoning & Complex Situational Modeling: Craft reasoning-driven QAs that mirror real-world architectural challenges and require retrieval, reasoning, planning, and modeling skills. This stage enables models to reason about trade-offs and system-level design scenarios to push AI’s capabilities. All contributors will receive appropriate credit and be featured on our community leaderboard. All contributors will receive appropriate credit and be featured on our community leaderboard. Why Get Involved Why Get Involved The computer architecture community has a strong legacy of collaboration with many of our most impactful tools built and sustained through community efforts, exemplified by projects like the gem5 simulator, RISC-V ISA, McPAT, and SimpleScalar. These projects demonstrate how collective contributions can create resources that advance research and industry practice. The computer architecture community has a strong legacy of collaboration with many of our most impactful tools built and sustained through community efforts, exemplified by projects like the gem5 simulator RISC-V ISA SimpleScalar. These projects demonstrate how collective contributions can create resources that advance research and industry practice. As we enter the era of AI-assisted architecture, QuArch offers a unique opportunity to shape the future of computer architecture through community efforts. All contributors will receive proper attribution, including co-authorship on QuArch publication(s), acknowledgment in documentation, and recognition on our community leaderboard. As we enter the era of AI-assisted architecture, QuArch offers a unique opportunity to shape the

future of computer architecture through community efforts. All contributors will receive proper attribution, including co-authorship on QuArch publication(s), acknowledgment in documentation, and recognition on our community leaderboard. **Getting Involved** If you're ready to help shape QuArch, sign up here to join our project using Label Studio for dataset annotation. Once your account is approved, it will take a brief minute for access to the QuArch v1.0 project and labeling interface (Figure 4) to appear. Please review our FAQs for labeling guidelines and best practices. If you're ready to help shape QuArch, sign up join our project using Label Studio for dataset annotation. Once your account is approved, it will take a brief minute for access to the QuArch v1.0 project and labeling interface (Figure 4) to appear. Please review our for labeling guidelines and best practices. **Figure 4: Interface for QuArch dataset labeling.** You can also submit your own QAs through this form to contribute additional questions or other input. You can also submit your own QAs through this form to contribute additional questions or other input. If you are also interested in joining the broader Architecture 2.0 community, please fill out this form to be contacted about updates and meet-ups for QuArch and other related activities. If you are also interested in joining the broader Architecture 2.0 community, please fill out this form to be contacted about updates and meet-ups for QuArch and other related activities. **Conclusion.** QuArch is an effort to push the boundaries of computer architecture through crowdsourcing. The goal is to leverage diverse questions from academia and industry while avoiding traditional IP constraints. By contributing to this dataset, we are shaping AI-driven tools and technologies that will impact the future. We hope you join the effort! QuArch is an effort to push the boundaries of computer architecture through crowdsourcing. The goal is to leverage diverse questions from academia and industry while avoiding traditional IP constraints. By contributing to this dataset, we are shaping AI-driven tools and technologies that will impact the future. We hope you join the effort! **Acknowledgments.** The development of QuArch was an extraordinary collective effort made possible by the dedication and contributions of a diverse and talented group of individuals across academia and industry. We want to especially acknowledge Amir Yazdanbakhsh (Google DeepMind), who, together with us, developed the vision for Architecture 2.0 that gave birth to QuArch. We are deeply grateful to our collaborators: Fin Amin (NCSU), Arnav Balyan, Yash Choudhary (IIT Bombay), Andy Cheng (Harvard), Sofia Giannuzzi (Harvard), Shreyas Grampurohit (IIT Bombay), Radhika Ghosal (Harvard), Jeffrey Ma (Harvard), Ankita Nayak (Qualcomm AI Research), Aadya Pipersenia (IIT Bombay), Jessica Quaye (Harvard), Arya Tschand (Harvard), Ike Uchendu (Harvard), and Jason Yik (Harvard) for their significant contributions to QuArch v0.1 and the arXiv report, which laid the foundation for many of the ideas presented in this blog. This year-long endeavor was further enriched by the efforts of over 40 students from around the world, whose hard work and creativity were important in shaping the early development of QuArch. The development of QuArch was an extraordinary collective effort made possible by the dedication and contributions of a diverse and talented group of individuals across academia and industry. We want to especially acknowledge Amir Yazdanbakhsh (Google DeepMind), who, together with us, developed the vision for Architecture 2.0 that gave birth to QuArch. We are deeply grateful to our collaborators: Fin Amin (NCSU), Arnav Balyan, Yash Choudhary (IIT Bombay), Andy Cheng (Harvard), Sofia Giannuzzi (Harvard), Shreyas Grampurohit (IIT Bombay), Radhika Ghosal (Harvard), Jeffrey Ma (Harvard), Ankita Nayak (Qualcomm AI Research), Aadya Pipersenia (IIT Bombay), Jessica Quaye (Harvard), Arya Tschand (Harvard), Ike Uchendu (Harvard), and Jason Yik (Harvard) for their significant contributions to QuArch v0.1 and the arXiv report, which laid the foundation for many of the ideas presented in this blog. This year-long endeavor was further enriched by the efforts of over 40 students from around the world, whose hard work and creativity were important in shaping the early development of QuArch. **About the Authors.** Shvetank Prakash is a fourth-year Ph. D. candidate at Harvard University focused on system design automation using machine learning techniques and open-source and emerging technologies for ultra-low power ML hardware architectures. Shvetank Prakash is a fourth-year Ph. D. candidate at Harvard University focused on system design automation using machine learning techniques and open-source and emerging technologies for ultra-low power ML hardware architectures. Vijay Janapa Reddi is an Associate Professor at Harvard University, where his research focuses on computer architecture and machine learning systems for autonomous agents. He serves as Vice President and co-founder of MLCommons, driving community-driven

initiatives in machine learning innovation. He is passionate about expanding access to applied machine learning through open-source education, as evidenced by his open-source book "Machine Learning Systems" (mlsysbook.ai) and the widely-adopted TinyML course series on edX. Vijay Janapa Reddi is an Associate Professor at Harvard University, where his research focuses on computer architecture and machine learning systems for autonomous agents. He serves as Vice President and co-founder of MLCommons, driving community-driven initiatives in machine learning innovation. He is passionate about expanding access to applied machine learning through open-source education, as evidenced by his open-source book "Machine Learning Systems" (mlsysbook.ai) and the widely-adopted TinyML course series on edX. Disclaimer: These posts are written by individual contributors to share their thoughts on the Computer Architecture Today blog for the benefit of the community. Any views or opinions represented in this blog are personal, belong solely to the blog author and do not represent those of ACM SIGARCH or its parent organization, ACM. Disclaimer: These posts are written by individual contributors to share their thoughts on the Computer Architecture Today blog for the benefit of the community. Any views or opinions represented in this blog are personal, belong solely to the blog author and do not represent those of ACM SIGARCH or its parent organization, ACM. Share this: • Click to share on Facebook (Opens in new window) Facebook Click to share on Facebook (Opens in new window) Facebook • Click to share on X (Opens in new window) ## X Click to share on X (Opens in new window) • Click to share on LinkedIn (Opens in new window) LinkedIn Click to share on LinkedIn (Opens in new window) LinkedIn.entry-content.et_post_meta_wrapper

- **Shared by:** Sasha Newman-Oktan
- **Shared on:** Jan 07, 2025 at 05:17 PM
- **Domain:** www.sigarch.org
- **Length:** 4,404 words
- **Processed:** Jul 08, 2025 at 11:15 AM

2. GitHub - Shubhamsaboo/awesome-llm-apps: Collection of awesome LLM apps with AI Agents and RAG using OpenAI, Anthropic, Gemini and opensource models.

■ <https://github.com/Shubhamsaboo/awesome-llm-apps>

■ Website Information:

****Website Description:**** The GitHub repository 'awesome-llm-apps' by Shubhamsaboo serves as a comprehensive collection of innovative applications utilizing Large Language Models (LLMs) integrated with AI agents and Retrieval-Augmented Generation (RAG) techniques. This repository showcases a variety of LLM applications powered by leading models from OpenAI, Anthropic, Gemini, and various open-source alternatives, catering to developers and researchers interested in leveraging advanced AI technologies. Targeted towards AI enthusiasts, developers, and researchers, the repository provides valuable resources, including tutorials and examples for implementing AI agents and multi-agent systems. Key features include a curated list of applications, detailed documentation, and the ability to run models locally, making it an essential resource for anyone looking to explore the capabilities of LLMs in practical scenarios.

- **Shared by:** Cristin Connerney
- **Shared on:** Jan 07, 2025 at 10:29 AM
- **Domain:** github.com
- **Length:** 1,781 words
- **Processed:** Jul 08, 2025 at 11:16 AM