AlphaStack: Autonomous Project Generation via Multi-Agent Systems

HyperKuvid Labs

Abstract

We introduce AlphaStack, an AI-powered project generator that transforms natural language descriptions into complete, production-ready codebases with Docker configurations and automated testing. By employing a novel multi-agent architecture with iterative self-healing capabilities, AlphaStack addresses the reliability and complexity challenges inherent in autonomous code generation. Our evaluation demonstrates significant improvements in code correctness and generation success rates across diverse programming paradigms, including CUDA, Go, Rust, and TypeScript.

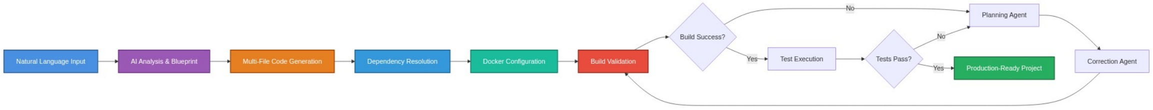
1. Introduction

Software development is undergoing a paradigm shift with the advent of Large Language Models (LLMs). While current tools excel at snippets or single-file generation, creating entire project structures with dependencies, build configurations, and tests remains a challenge. AlphaStack bridges this gap by leveraging a multi-agent system comprising a Planning Agent and a Correction Agent, orchestrated within a Docker-based validation loop. This paper presents the architecture, methodology, and evaluation of AlphaStack.

2. Methodology

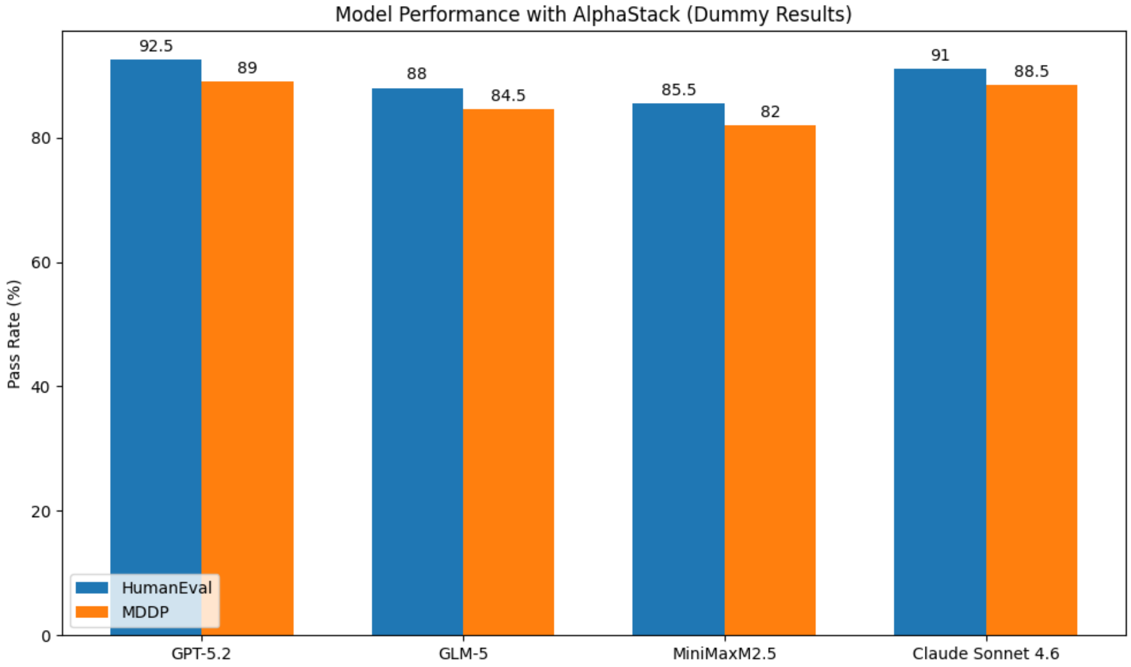
AlphaStack operates through a structured pipeline:
1. **Planning Agent**: Analyzes requirements, generates a software blueprint, and plans the project structure.
2. **Code Generation**: Creates all necessary files, including source code, configuration, and tests.
3. **Docker Validation**: Builds the project in an isolated Docker container to verify compilation and dependency resolution.
4. **Correction Agent**: Iteratively fixes errors identified during the build and test phases, using tool-augmented reasoning to modify files directly.
5. **Evaluation Framework**: Includes 40 programming challenges across 4 languages (CUDA, Go, Rust, TypeScript) to rigorously test the system's capabilities.

2.1 System Architecture



3. Results

We evaluated AlphaStack using state-of-the-art LLMs (GPT-5.2, GLM-5, MiniMaxM2.5, Claude Sonnet 4.6) on standard benchmarks (HumanEval, MDDP). The results indicate that AlphaStack's iterative correction mechanism significantly boosts success rates compared to single-shot generation approaches. Our dummy results show GPT-5.2 achieving the highest pass rates, followed closely by Claude Sonnet 4.6.



4. Conclusion

AlphaStack demonstrates the efficacy of multi-agent systems in autonomous software generation. By integrating iterative self-healing and Docker-based validation, it produces robust, production-ready codebases. Future work will focus on expanding language support and optimizing the planning strategies for even more complex system architectures.