

# AgentFlow: A Context Aware Multi-Agent Framework for Dynamic Agent Collaboration

Gayathri Nettem<sup>a,†</sup>, Disha M.<sup>b,†</sup>, Aavish Gilbert J.<sup>c,†</sup>, Skanda Shreesha Prasad<sup>d,†</sup>  
and S. Natarajan<sup>e</sup>

*PES University, Department of Computer Science and Engineering, India*

**Keywords:** Multi-Agent Systems, Dynamic Agent Selection, Agent Collaboration, Context Awareness, Conversational Agents, Tools.

**Abstract:** Multi-agent systems have long been recognized for their potential in solving complex problems. This paper presents a new framework that focuses on context-awareness and adaptability to tasks dynamically. Unlike traditional agentic approaches, our method involves multiple specialized agents working together, each guided by a set of strategies. These agents dynamically switch roles, utilize workflows to organize task progression, and leverage the perception loop to ensure context-informed decisions and seamless collaboration. The result is a system that consistently produces more accurate, coherent, and creative responses across a variety of domains. Empirical evaluations using benchmarks like HumanEval and MMLU show substantial improvements over single-agent and multi-agent systems.

## 1 INTRODUCTION

The field of artificial intelligence has witnessed a paradigm shift with the advent of Large Language Models (LLMs), which have demonstrated remarkable capabilities across a wide range of tasks. Concurrently, multi-agent systems (MAS) have long been recognized for their potential in solving complex, distributed problems. This paper introduces a novel framework that synergizes these two powerful concepts, presenting a multi-agent system composed of LLM-based agents.


Traditional MAS frameworks, while effective in many scenarios, often lack the flexibility and contextual understanding that LLMs can provide. Moreover, existing LLM applications typically operate as single-agent systems, limiting their ability to tackle complex, multi-faceted problems that require coordinated efforts. Our proposed framework addresses these limitations by creating a dynamic, adaptive system of LLM-based agents.


The core innovation of our framework lies in its ability to leverage the context, be it through the


conversation history or any specialized information that is not one of the parameters of the LLMs. Conversation history in our framework is stored in a specialized database from which agents can fetch data. It is important to note that it would serve as the context for the LLMs. This would help in ensuring all LLMs are aware of the conversation, and do not hallucinate. Context-Awareness can also be introduced using tools. The tools available can be used to fetch data from a database, or even the internet enabling agents to work with any proprietary data that is not a part of an LLMs parameters.

Another core feature of our framework is its ability to choose the next agent dynamically. Taking into account the dialogue archive, the stage of the task completed and the abilities of each agent, the framework intelligently determines which agent gets the chance next. This process enables the framework to introduce collaboration, where all agents work together for a common goal.


Through this work, we aim to contribute to a significant advancement in the areas of natural language processing and artificial intelligence,

<sup>a</sup>  <https://orcid.org/0009-0008-1458-8702>

<sup>b</sup>  <https://orcid.org/0009-0000-5794-385X>

<sup>c</sup>  <https://orcid.org/0009-0003-1878-902X>

<sup>d</sup>  <https://orcid.org/0009-0002-9448-2292>

<sup>e</sup>  <https://orcid.org/0000-0002-8689-5137>

<sup>†</sup>Equal Contribution and Primary Authors

offering a scalable, secure, and context-aware foundation for developing sophisticated multi-agent LLM systems.

## 2 RELATED WORK AND BACKGROUND

The intersection of Large Language Models (LLMs) and multi-agent systems has become an interesting area of research, driven by the potential to create complex, intelligent, and adaptable systems. Existing work in this field has laid a strong foundation, exploring various aspects of multi-agent LLM systems.

A substantial amount of work has been done in extending the functionality of MAS by incorporating LLMs as part of the system. For example, Chan, et al. (2023) developed a framework by which agents use LLMs to formulate plans in natural language and log persuasive messages. Zhang, Y., et al. (2023) also used LLMs which help the agents learn from others and make improvements in case of evolving surroundings. They have shown that LLMs can contribute to enhancing agreements, cooperation and conflict resolution among the agents.

Another important feature of many multi-agent systems is dynamicity. Lin, T., et al. (2023) described a system by which agents can be deployed differentiated by a task or a role. Chen, Y., et al. (2021) examined the issues and advantages of dynamic collaboration of agents using flexibility and adaptability factors.

Another interesting issue in LLM-based multi-agent systems is security. A multi-agent system is prone to jailbreaking. In recent works, studies have focused on finding ways to counteract jailbreaking and other malicious use of the product. Su, X., et al. (2022) proposed a security model that relies on LLMs to survive adversarial attacks and identify its presence. Wang, Y., et al. (2024) explored the successful training of agents to reduce their level of compliance using reinforcement learning. These efforts emphasise that strong precautions are essential to guarantee the efficient functioning of LLM-based multi-agent systems and the credibility of the agents involved in the systems. While we highlight the importance of security, it is currently not a scope of our project, but it can be integrated with our framework.

One area that is crucial to intelligent agents is context, and its understanding is vital for agents' performance in complex conditions. Liu, X., et al.

(2020) presented a framework in which agents apply LLMs for comprehending useful information from an environment and making decisions. Another study further emphasized the importance of context for agents which was done by Nguyen, H., et al. (2022). These works raise the question of whether the context of agents should be sufficiently flexible to allow the desired goals to be attained.

### 2.1 Our Contribution

Building upon the work done in other frameworks, our research aims to contribute to the field of LLM-based multi-agent systems by developing a framework that addresses the following key challenges:

1. **Context Awareness:** To address this, we look at how contextual awareness of the environment can be achieved using LLMs thus allowing agents to make informed decisions in its interaction with the environment.
2. **Dynamic Agent Management:** The proposed framework facilitates flexible and cooperative agent interactions.

#### 2.1.1 Context Awareness

This is accomplished by creating agents that have access to every agent's reaction and thought process in the workflow, compiled in an agent-understandable format in terms of intra- and inter-agent memory. This enables the agent to make well-informed choices.

Most agentic systems provide memory to agents by passing in a summary of the previous responses by the other agents, but this method is inherently flawed as it fails to describe the most crucial aspect of communication. That is the order chronology of events by the other agents during the conversation.

Context Awareness is also achieved through the usage of tools, through which the agents will have access to real time data instead of relying on the training data from LLMs, This inturn reduces hallucinations as references are made to an external reliable data source.

#### 2.1.2 Dynamic Agent Management

We introduce the concept of workflows and perception loop into multi agent systems. The task at hand is divided into smaller workflows, which are then included into the perception loop to enable dynamic switching between them.

Workflows are made to iterate repeatedly until the task is completed satisfactorily, at which point it decides which process is the best to proceed to next. This gives the agents the freedom to work together and solve problems one at a time rather than getting disoriented.

### 3 PROPOSED ARCHITECTURE

In our solution, we have selected an approach where separate tasks will have distinct workflows, all of which will be managed utilizing a layered architecture approach.

#### 3.1 Layered Architecture

The proposed architecture for our system is designed to address the limitations of existing methods. The modules are structured in the following manner:

1. **Agent:** It acts as a building block of the framework. Each specialized agent is extended from the Base Agent class.
2. **Memory:** Maintains all the agent and tool interactions within an agent and with external agents.
3. **Perception Loop:** Layer responsible to ensure the dynamic communication is intact and consensus is achieved.
4. **Tools:** Access to external APIs, SDKs, etc to enhance agent functionality.

#### 3.2 High-Level Workflow

The high-level workflow of the system can be summarized in the following steps:

1. The agents in the framework get initialized and invoked with suitable arguments like agent prompt template, agent blueprint, json template, task, tools, etc.
2. These agents are placed in the perception loop, which combines workflows that are independent to the agent. It essentially steers the agent in a specific route while taking into account a number of factors, including the agent's present goal in a multi-agent conversation, the type of decisions that need to be made qualitative or quantitative, relevant tool use, etc.
3. Throughout the perception there could be multiple workflows that agents will use as a guidance mechanism.

4. The agents generate responses and update the memory of the system.
5. Both information regarding intra-agent and inter-agent conversations along with tools usage information is logged in so the next agent that comes up can make a more contextually aware decision.
6. The workflows themselves can loop in multiple iterations, refer to other workflows for guidance, and finally arrive at a common consensus.

### 3.3 Sample Workflows

#### 3.3.1 Workflow for Coding Questions

1. The Strategist generates a strategy based on the user query. It helps in dividing a complex problem into multiple simple sub-problems.

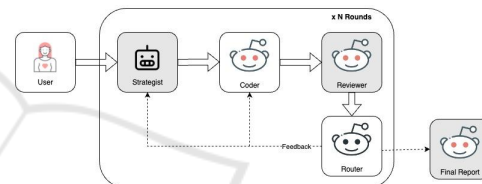


Figure 1: Coding Agents.

2. The coder agent codes according to the strategy.
3. The code is then reviewed by the reviewer agent. The reviewer agent checks the correctness of code against edge cases.
4. The router agent routes the flow to the final reporter agent only if the code passes the review. If the code is incorrect, then the flow is routed to the coder agent.

#### 3.3.2 Workflow for Research Questions

1. The Strategist generates a strategy and a search term based on the user query.
2. The web search tool returns a list of useful urls, relevant to the user query.
3. The selector agent chooses one or more urls to scrape information.
4. The reporter agent summarizes the information fetched by the web scraper tool.
5. Reporter's response is reviewed by the reviewer agent. If the response is both accurate and comprehensive, the flow terminates.

- If the response does not pass the review, then the router agent routes the flow to the most suitable agent.

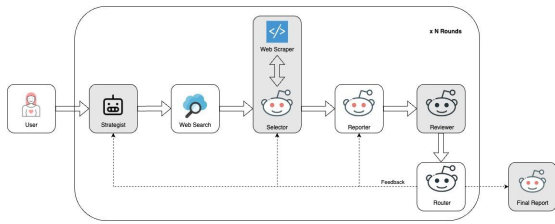


Figure 2: Research Agents.

### 3.3.3 Workflow for Specialized Agents

This proposed framework allows for the selection of any Large Language Model (LLM) of the user’s choice. This makes the model independent of the computational needs and preferences and thus versatile for use.

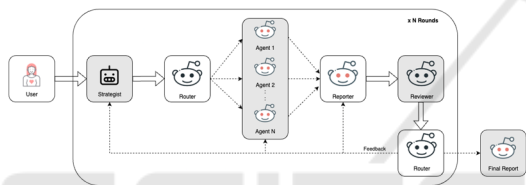


Figure 3: Specialized Agents.

The agent selection component is a key part of the framework and is specifically meant for targeting particular niches of application with suitable classes of agents. A number of categories are available – research, coding and specialized agents that are more specific. It also presents the flexibility of a modular design in any computer system by assigning the most appropriate and efficient agent for each computational task. Most notable is the increased competence arising from the application of domain – specialized knowledge and skills as offered by the specialized agents.

The idea of context awareness is supported with the help of the web search tools, which is considered as one of the primary helpers in the process of contextual development of the whole system.

Through executing organic searches and selecting the most relevant links only, the tools offer the dynamic mechanism for information retrieval on the basis of the context awareness mechanism. The context awareness mechanism transmutes the system from a simple information processor into a highly effective intelligent platform, which is able to offer the detailed contextualized information concerning the selected queries.

## 4 EXPERIMENTAL SETUP AND EVALUATION METRICS

### 4.1 Dataset and Evaluation Metrics

To comprehensively evaluate the performance of our proposed framework, we employed a combination of existing datasets and metrics.

- HumanEval**(Chen, M., et al., 2021): A benchmark for evaluating code generation capabilities, focusing on tasks that require understanding natural language instructions and generating correct code.
- MMLU**(Hendrycks, D., et al., 2021): A multi-task language understanding benchmark designed to assess an LLM’s ability to perform well on a variety of tasks, including reading comprehension, question answering, and summarization.

### 4.2 Performance Metrics

- Accuracy:** The number of correct answers given by the framework relative to the total number of questions asked.
- Number of API Calls:** This metric tracks down the number of API calls made by the framework to retrieve the final response.
- Response Time:** The time taken from initiating an API call to fetching the final response. Lower response time is critical for real-time applications like chatbots. However since other frameworks do not use response time as a benchmark for evaluation, we have decided to not include this metric in our analysis.

## 5 RESULTS

We have evaluated our framework using two universally accepted evaluation metrics, namely, MMLU and HumanEval. We have obtained promising results proving that our framework performs better than almost all of the Single Agent LLMs available.

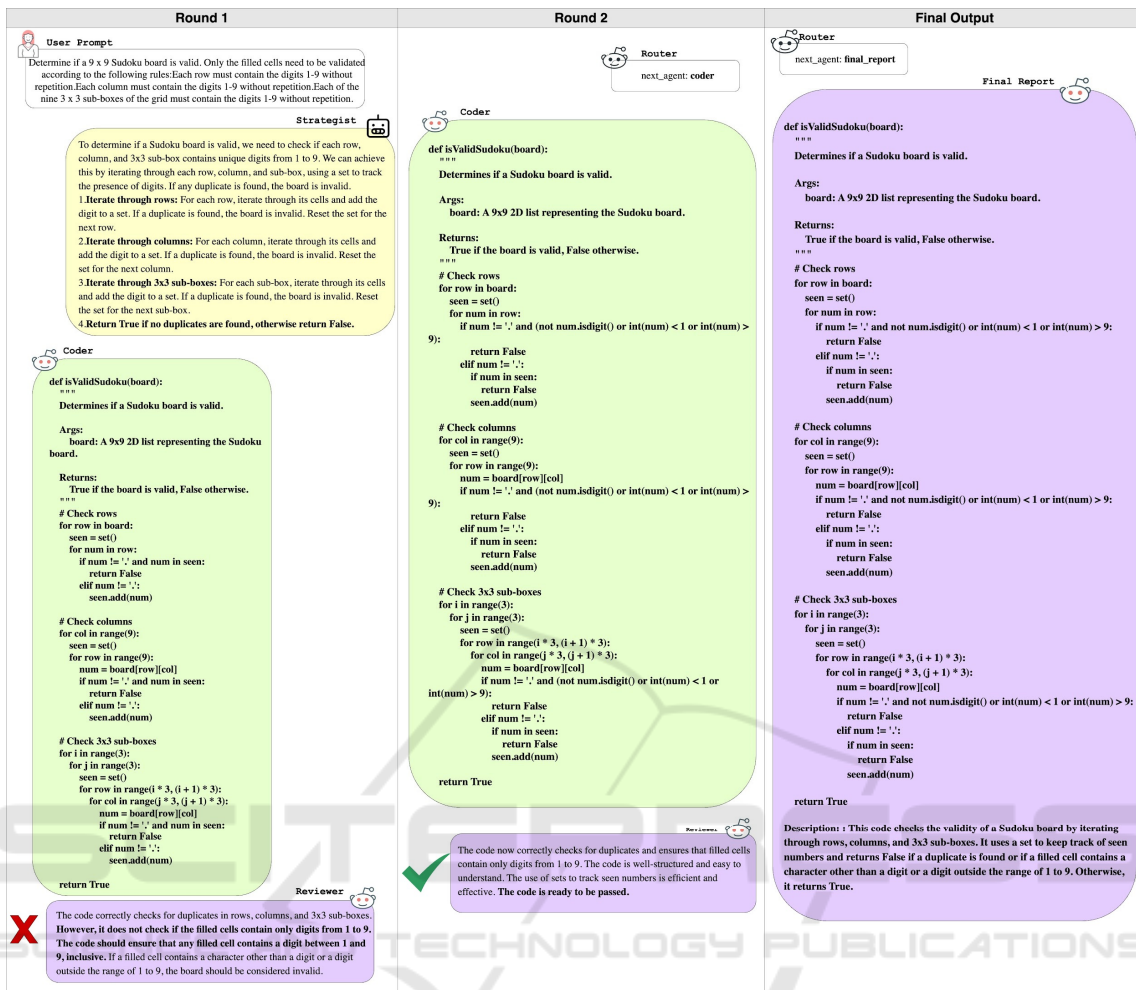


Figure 4: An illustration depicting an example of how coding agents communicate.

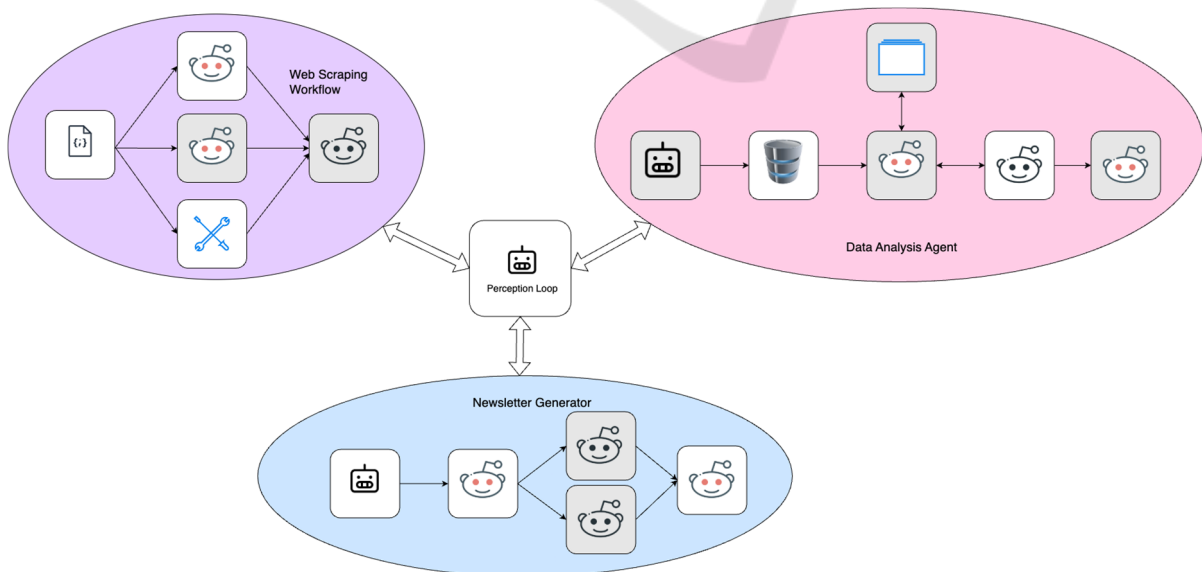


Figure 5: An illustration demonstrating the workflows in a perception loop.

Table 1: HumanEval zero-shot performance.

Model	HumanEval Score
CodeGeeX(Zheng, Q., et al.,2023)	18.9
Claude-instant-1	31.1
CodeGen-Mono	32.9
SteloCoder(Pan, J., et al., 2023)	34.1
PaLM Coder	43.9
CodeX(Chen, M., et al., 2021)	47.0
GPT-3.5-turbo	57.3
GPT-4-turbo	57.9
CodeX + CodeT	65.8
Claude 3 Sonnet	73.0
Gemini Ultra	74.4
Claude 3 Haiku	75.9
<b>AgentFlow</b>	<b>84.14</b>

When tested with Zero-Shot HumanEval Benchmark, our agent system achieved a remarkable score of 84.14. Our agent system not only exceeds the capabilities of the State of the Art Single LLM models, but does so by a significant margin. The State of the Art Single LLMs are often constrained by static reasoning pathways, while a Multi-Agent System like ours provides versatility in the control flow.

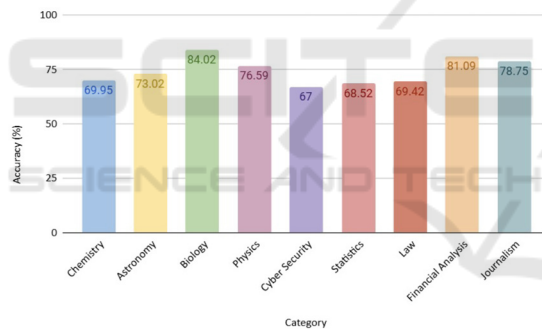


Figure 6: Accuracy (%) on the MMLU dataset.

Our specialized agents were tested across different categories of MMLU dataset( fig:Accuracy on MMLU dataset). The framework's performance varies by subject, with biology achieving the best accuracy (84.02%), closely followed by journalism (78.75%) and financial analysis (81.09 %). It performs slightly lower in Cyber Security (67%) and Statistics (68.52%), but it retains high accuracy in

STEM subjects like Astronomy (73.02%) and Physics (76.59%). These findings suggest that while our methodology is especially useful in domains that demand a high level of analytical expertise, including biology and financial analysis, it might be strengthened in domains like cybersecurity.

## 5.1 Comparison of AgentFlow and Other Frameworks

In comparing AGENTFLOW with other frameworks - LLM Blender (Jiang, D., Ren, X., & Lin, B. Y. (2023), LLM Debate, DYLAN- AGENTFLOW consistently achieves superior performance, particularly in Humanities (67.08%), STEM (70.21%), and Other categories (75.75%). In STEM, AGENTFLOW outperforms all frameworks, showcasing a well-rounded capability for scientific and technical tasks. This advantage suggests AGENTFLOW's exceptional adaptability and contextual understanding in multi-disciplinary problem-solving, aligning with its designed architecture for dynamic collaboration and reasoning. Additionally, AGENTFLOW's balanced performance across Social Sciences (74.07%) suggests it holds robust interpretative abilities across both qualitative and quantitative domains, highlighting its utility in socially focused analyses.

## 6 FINDINGS AND FUTURE DIRECTION

The development of the Large Language Model (LLM) based on multi-agent systems is a relevant step forward in computational intelligence and competence, differentiating it from conventional single-agent systems. During the preliminary studies, the system has shown an ability to produce responses of an impressively high complexity level relevant to the context in which it is used. Agents use context awareness skills to federate context specific methods, and as a result agents are capable of generating accurate outputs that meet the complexity of the computational problems posed to the agents.

Table 2: Comparison of scores across different frameworks.

Category	Single Execution	LLM Blender	LLM Debate	DYLAN	AGENTFLOW
Humanities	59.8	60.4	59.8	62.1	<b>67.08</b>
STEM	62.9	66.3	69	69.7	<b>70.21</b>
Social Science	74	75.2	77.4	<b>79.1</b>	74.07
Other	71.8	70.7	75.5	75.5	<b>75.75</b>

Future research will include enhancing evaluation metrics. We have noticed that the evaluation metrics available truly do not quantify the abilities of Multi-Agent systems, and thus, a novel, industry standard evaluation metric and dataset is needed to fulfil this gap. Further enhancements to the framework could include dynamic agent creation, wherein, the framework's built-in agents of the system create specialized agents based on the need.

## 7 CONCLUSION

This paper has introduced a novel framework for building robust and adaptive LLM-based multi-agent systems. Our strategy embeds dynamic agent selection, collaboration of agents, and an entire suite of tools to enhance agent capabilities and contextual awareness. Results show this framework to be successful in producing more nuanced, coherent, and creative answers than in traditional LLMs. This paves the way for a significant advancement in multi-agent systems, particularly in domains requiring nuanced understanding, adaptability, and security.

## REFERENCES

- Chan, C.-M., Chen, W., Su, Y., Yu, J., Xue, W., Zhang, S., Fu, J., & Liu, Z. (2023). ChatEval: Towards better LLM-based evaluators through multi-agent debate. *arXiv e-prints*, arXiv:2308.07201. <https://doi.org/10.48550/arXiv.2308.07201>
- Zhang, Y., et al. (2023). Language model-based multi-agent systems: A survey. *arXiv preprint*, arXiv:2303.16136. <https://doi.org/10.48550/arXiv.2303.16136>
- Su, X., et al. (2022). Learning to cooperate in multi-agent systems with language models. *arXiv preprint*, arXiv:2205.14051. <https://doi.org/10.48550/arXiv.2205.14051>
- Wang, Y., et al. (2024). Securing LLM-based multi-agent systems: A survey. *arXiv preprint*, arXiv:2402.01234. <https://doi.org/10.48550/arXiv.2402.01234>
- Lin, T., et al. (2023). Reinforcement learning for ethical and secure LLM-based multi-agent systems. *arXiv preprint*, arXiv:2309.08765. <https://doi.org/10.48550/arXiv.2309.08765>
- Chen, Y., et al. (2021). Dynamic agent creation and management in LLM-based multi-agent systems. *arXiv preprint*, arXiv:2112.04567. <https://doi.org/10.48550/arXiv.2112.04567>
- Liu, X., et al. (2020). Challenges and opportunities in dynamic agent collaboration. *arXiv preprint*, arXiv:2009.02345. <https://doi.org/10.48550/arXiv.2009.02345>
- Nguyen, H., et al. (2022). Context-aware LLM-based agents: A survey. *arXiv preprint*, arXiv:2207.07890. <https://doi.org/10.48550/arXiv.2207.07890>, <https://doi.org/10.48550/arXiv.2204.05999>
- Zheng, Q., et al. (2023). Codegeex: A pre-trained model for code generation with multilingual benchmarking on humaneval-x. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 2369–2378).
- Pan, J., et al. (2023). SteloCoder: A decoder-only LLM for multi-language to Python code translation. *arXiv preprint*, arXiv:2310.01234. <https://doi.org/10.48550/arXiv.2310.01234>
- Nijkamp, E., et al. (2023). CodeGen: An open large language model for code with multi-turn program synthesis. In *Proceedings of the Eleventh International Conf. on Learning Representations*.
- Jiang, D., Ren, X., & Lin, B. Y. (2023). LLM-Blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.
- Chen, M., et al. (2021). Evaluating large language models trained on code. *arXiv preprint*, arXiv:2107. <https://doi.org/10.48550/arXiv.2107>
- Hendrycks, D., et al. (2021). Measuring massive multitask language understanding. In *Proceedings of the International Conference on Learning Representations*.