

Intermediate Representations for Improved Code Translation with LLMs

Code translation is the task of converting code from one programming language (e.g., Java) to another (e.g., Python) [1]. Early statistical efforts [2] have recently given way to methods based on large language models (LLMs). However, studies show that LLMs still produce buggy translations using a zero-shot prompt [1]. One promising avenue to improve translation accuracy is through intermediate representations, which provide structured guidance for the translation process. We investigate whether LLM-based code translation can benefit from intermediate representations (IRs), specifically in the forms of natural language (NL) summaries and abstract syntax trees (ASTs). We explore two main approaches to incorporate IRs in code translation: (1) a two-step approach (2S), where the LLM first translates the original code to IR and then translates this IR to the target language [3]; and (2) a chain-of-thought (CoT) prompting approach, where the LLM is instructed to use IR to explain its reasoning during translation.

For our experiments, we use two code translation benchmarks: sampled CodeNet [4] (languages: C, C++, Go, Java, Python) and AVATAR [5] (languages: Java, Python). In Phase 1, we experiment with different permutations of IRs (NL, AST, or both), and compare with the simple zero-shot (0SP) and one-shot prompt (1SP) baselines using the open-source GPT-4 LLM as the backbone (Open GPT4 8X7B [6]) on the sampled CodeNet. Based on Phase 1 results, we evaluated the two highest performing prompts with Open GPT4 8X7B, StarCoder [7] and CodeGen LLMs in Phase 2. Following [1], we use the percentage of successful translations (successful translated code would compile, pass runtime checks, and pass existing tests) as the performance metric.

From Table 1, CoT with NL performed the best, with a 13.8% and 6.7% improvement on the CodeNet and AVATAR dataset, respectively, compared to the initial zero-shot prompt with Open GPT4 8X7B. Our experiments highlight the potential for generalizing our findings and underscore the benefits of using IRs in code translation. Future work includes exploring additional languages and datasets.

Table 1: Results of Open GPT4 8X7B (GPT), StarCoder (Star), and CodeGen (Code); Success% = successful translation rate.

Prompt	Phase 1	Phase 2					
	CodeNet GPT	CodeNet		AVATAR			
		Star	Code	GPT	Star	Code	
0SP	28.6%	36.3%	18.4%	17.6%	20.4%	6.8%	
1SP	33.5%						
2S-NL	10.7%						
2S-AST	2.6%						
CoT-NL	42.4%	38.0%	4.9%	24.3%	20.6%	1.3%	
CoT-AST	32.2%						
2S-NL-AST	9.0%						
2S-AST-NL	11.2%						
CoT-NL-AST	39.6%	31.2%	2.9%	22.2%	16.9%	0.2%	
CoT-AST-NL	37.9%						

- [1] Rangeet Pan et al. Lost in translation: A study of bugs introduced by large language models while translating code. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, pages 1–13, New York, NY, USA, 2024. Association for Computing Machinery.
- [2] Yusuke Oda et al. Learning to generate pseudo-code from source code using statistical machine translation. In *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 574–584, Los Alamitos, CA, USA, 2015. IEEE, IEEE Computer Society.
- [3] Wasi Uddin Ahmad, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. Summarize and generate to back-translate: Unsupervised translation of programming languages. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1528–1542, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.
- [4] Ruchir Puri et al. Codenet: A large-scale ai for code dataset for learning a diversity of coding tasks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, San Diego CA, 2021. NeurIPS.
- [5] Wasi Uddin Ahmad, Md Golam Rahman Tushar, Saikat Chakraborty, and Kai-Wei Chang. Avatar: A parallel corpus for java-python program translation, 2023.
- [6] TheBloke. Thebloke/open_gpt4.8x7b-gguf, 2024.
- [7] Raymond Li et al. Starcoder: may the source be with you!, 2023.