engineering proceedings

MDPI

Proceeding Paper

Generative AI for Code Translation: A Systematic Mapping Study †

Aymane Rgaguena *, Imane Chlioui and Maryam Radgui

SI2M Laboratory, National Institute of Statistics and Applied Economics, Rabat 10112, Morocco; ichlioui@insea.ac.ma (I.C.); m.radgui@insea.ac.ma (M.R.)

* Correspondence: aymanergaguena01@gmail.com

† Presented at the 7th edition of the International Conference on Advanced Technologies for Humanity (ICATH 2025), Kenitra, Morocco, 9–11 July 2025.

Abstract

Generative artificial intelligence (AI) has greatly advanced the code translation process, particularly through large language models (LLMs), which translate source code from one programming language into another. This translation has historically been error-prone, labor-intensive, and highly dependent on manual intervention. Although traditional tools such as compilers and transpilers have restrictions in managing complex programming paradigms, recent generative AI models most importantly those based on transformer architectures, have shown promise. This systematic mapping study intends to evaluate and compile studies on Generative AI applications in code translation released between 2020 and 2025. Using five main criteria, the publication year and channel, research type, publication type, empirical study type, and AI models. A total of 53 relevant articles were chosen and examined. The results show that conferences and journals are the most often used publishing venues. Although historical-based evaluations and case studies were the empirical methodologies most often used, researchers have primarily focused on implementing transformer-based artificial intelligence models.

Keywords: Generative AI; code translation; Transformer Models; large language models; empirical evaluation; code migration; software development

check for updates

Academic Editors: Brahim El Bhiri and Ayman S. Mosallam

Published: 15 October 2025

Citation: Rgaguena, A.; Chlioui, I.; Radgui, M. Generative AI for Code Translation: A Systematic Mapping Study. *Eng. Proc.* **2025**, *112*, 33. https://doi.org/10.3390/engproc2025112033

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Interoperability among many programming languages is the foundation of the multilingual ecosystem expanding around software development. Cross-platform software development, legacy system migration, and current application adaptation to new environments all routinely require code translation, the process of translating source code from one language to another [1]. Still, hand code translating is tedious and prone to errors. One must have a complete awareness of the target and the source languages [2]. Conventional rule-based tools including compiler and transpiler often fail to capture the subtleties of many programming paradigms, generating translations that are not ideal and demand a lot of human intervention [3]. A fresh approach has surfaced recently to raise code translating accuracy and quality. Large-scale models trained on vast code corpora are developed to grasp and replicate code structure, syntax, and semantics. These models generate context-aware and logically coherent translations [4], thereby transcending traditional rule-based transformations. Strong outcomes in code conversion, refactoring, and adaptation across paradigms have come from advanced transformer-based systems including AlphaCode,

TransCoder, and Codex [5]. These tools enable better scalability, efficiency, and flexibility in code translating processes by depending on the analysis of big repositories. These developments still face difficulties. Among them are the need of managing cross-language variants, guarantee semantic preservation, and create strong measures of translating quality. Moreover, human validation is still indispensable to guarantee accuracy and fit with the original program logic .

The structure of this paper is as follows: Section 2 outlines the research methodology, including the process of study selection and the formulation of research questions. Section 3 presents the findings of the mapping study. Section 4 provides an interpretation of the results along with a discussion. Section 5 addresses the threats to validity, and concludes the paper and highlights directions for future research.

2. Research Methodology

By means of organization and classification of current studies, a systematic mapping study (SMS) seeks to offer a comprehensive view of research within a given field. SMS emphasizes on providing an overview of the research terrain rather than method and result analysis found in systematic literature reviews. This SMS attempts to provide a thorough summary of published Generative AI for code translation works starting in 2020. The aim is to fully grasp the body of current research on this subject and finally arrive at the general objective of responding to five research questions (Cf. Table 1).

Table 1. Research questions on Generative AI for code translation.

ID	Research Question	Rationale
RQ1	What are the years of publication for research on Generative AI for Code Translation?	Distinct periods of heightened interest and chronological patterns of research activity were revealed.
RQ2	What types of publications contribute the most to this research area?	Determines whether certain publication outlets have consistently hosted work in this area.
RQ3	What are the research types in studies on Generative AI for Code Translation?	Identifies the different types of studies focusing on Generative AI for Code Translation.
RQ4	What are the most commonly used AI models and architectures for Code Translation?	Indicates which models and architectures are most prominently utilised in Generative AI for Code Translation.
RQ5	Which datasets are frequently used for training and evaluating Generative AI models for Code Translation?	Identifies key data resources leveraged by researchers, offering insight into how model performance is benchmarked.
RQ6	What types of empirical studies have been conducted in the application of Generative AI for Code Translation?	Categorises the investigations as observational, experimental, or case-based.

Following the definition of the research questions, we then look for pertinent papers to fulfill our research goals. Using a methodical search on five digital databases—IEEE Xplore, SpringerLink, ScienceDirect, ACM Digital Library, and Google Scholar—we find candidate papers for our research project. The search is carried out with a search string developed from main terms and their synonyms applied in the research questions (RQs).

A search string was identified, and the Boolean OR operator was used to connect alternative words, while the Boolean AND operator was employed to join major concepts.

The complete search string was defined as follows: ("Generative AI" OR "AI Generation Models" OR "Neural Generative Models" OR "Artificial Intelligence Generation" OR "Autoregressive Models" OR "Self-supervised Learning Models" OR "Large Language Models" OR "Transformer Models") AND ("Code Translation" OR "Programming Language Conversion" OR "Source Code Migration" OR "Automated Code Transformation" OR "Cross-Language Code Translation" OR "Code Adaptation" OR "Software Language Translation").

The search for relevant candidate documents and tools that support the goal of the mapping project depends on this kind of study. To reach this, we thus closely go over the title, abstract, and keywords of every work. We have created inclusion and exclusion rules that guide the decision-making process to ensure the relevance and quality of the selected publications. These criteria are supposed to reduce the candidate paper pool and provide a complete picture to handle our research subjects.

Inclusion Criteria:

- **IC1:** Only fully published research papers from conferences, books or journals.
- **IC2:** Generative AI for Code Translation that covers issues including source code transformation, code migration, or AI-driven programming language conversion must take the front stage.
- **IC3:** Only English-language papers are considered.

Exclusion Criteria:

- **EX1**: Publication Type: Short papers, posters, and editorials are excluded.
- **EX2:** Studies that do not focus on Generative AI applied to Code Translation are excluded.
- **EX3:** Identical studies found in multiple databases are filtered out, retaining only the most complete version.

Using these criteria methodically guarantees a thorough selection process that improves the relevance and quality of the last set of papers used to address our research questions. Data extraction forms designed to meet each paper's specific analysis procedures in line with our defined research questions (RQs), as per Table 1.

RQ1: With every paper that was selected, we captured key contextual information such as document type (whether it is a journal article, conference paper or a book chapter), publication source (name of publisher or digital repository), and the date of publication [6].
RQ2: Each paper was classified in relation to the primary existing research types.

- Evaluation Research (ER): These are the papers that analyze how well existing Generative AI models for code translation perform after measuring them against certain benchmarks [7].
- Solution Proposal (SP): Papers that either develop new Generative AI techniques or significantly enhance existing ones with respect to code translation
- Experience Papers (EP):These are papers in which authors elaborate on their experiences and the difficulties faced while implementing AI models during translation of codes into different languages [2].
- Review: These are papers summarizing the most detailed and up to date information about the use of Generative AI in code translation [2].

RQ3: To identify the publication types (journal articles, conference papers, and book chapters) that emerged most frequently and significantly within this branch of research [8].
RQ4: Collected information on the datasets most answered questions related to training and evaluating Generative AI models performance for code translation purposes. This includes dataset titles, their providers, dataset quantities, and the reasons why they were used [5].
RQ5: The literature frequently cited AI models and architectures in the literature, which include: transformer based models, LSTM models, autoregressive models, and big lan-

guage models.
RQ6: The type of empirical research can be classified as (Survey, historical evaluation (HBE), Case study) [9].

3. Results and Discussion

The research questions (RQs) listed in Table 1 are analyzed and discussed in this section. Accordingly, we first outline the study selection results and then discuss the findings for RQs 1–5.

3.1. Overview of the Selected Studies

Using a well-crafted search string, six major digital libraries were first searched for pertinent research on Generative AI for code translation. A total of 115 papers were identified out of this process. Following the Inclusion Criteria (IC) for titles, keywords, and abstracts, we screened out studies failing the criteria, so greatly lowering the total count. After applying the Inclusion Criteria (IC) to titles, keywords, and abstracts, we filtered out studies that did not meet the standards, significantly reducing the number. The remaining studies were then subjected to the Exclusion Criteria (EC), which eliminated those that deviated from our study emphasis. In order to answer the research questions, we finally found and kept 53 pertinent papers, methodically examined and reviewed by two authors.

3.2. RQ1: What Are the Years of Publication for Research on Generative AI for Code Translation?

As shown in Table 2, a total of 53 selected studies on Generative AI for Code Translation were published across various years between 2020 and 2025.

Table 2. Publication year distribution of Generative AI research papers.

Publication Year	Number of Papers	Percentage (%)
2020	1	1.89
2021	3	5.66
2022	6	11.32
2023	10	18.87
2024	29	54.72
2025	4	7.55
Total	**53**	**100**

The distribution of publications shows that the most recent years have dominated most of the studies, reflecting the rapid development and increasing relevance in this field of research. The emergence of transformer-based models, along with the widespread acceptance of large language models (LLMs), offers some explanation for this surge in research in recent years [1].

These developments can also encourage research in the topic because of their incredible ability to complete complicated programming paradigms and to understand code across languages. As there were the least number of studies in 2020 (1.89%), the previous years (2020–2022) were also comparatively low for textbook publications. During this period, the idea of studying Generative AI through the lens of code translation was still new, which may be another reason for the low publication rates [10]. Research during this time was generally focused on creating LLMs for natural language processing tasks rather than for code translation. Furthermore, usage of transformer architectures explicitly intended for code has only recently become prevalent alongside advancements in model fine-tuning and training procedures.

3.3. RQ2: What Types of Publications Contribute the Most to This Research Area?

As shown in Figure 1, the graph illustrates the statistics of conference and journal publications, highlighting the notable increase in research activity in 2024. In 2023, a more obvious increase occurs where conference and journal publications both increase, but more are still through conference. In 2024, the greatest increase occurs, by far, for conference publications, although journal articles increase as well, just not to the same degree. Therefore, 2024 has the greatest number of articles for either type. In 2025, a decrease occurs, in comparison to 2024, for journal and conference publications.

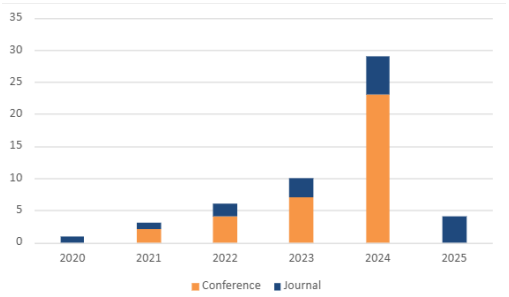

Figure 1. Number of papers published per year and publication channel.

Conferences play a key role in the swift dissemination of new ideas and experimental outcomes, enabling researchers to promptly share their latest developments with the community. This was particularly evident in 2024, when conference literature increased significantly owing to intense research activity [3,10–12].

The interactive format of professional conferences speaks of collaboration and feedback that helps to sharpen early-stage findings; this is particularly valuable in fast-moving fields that redesign new innovations to keep practitioners up to date. In contrast, journals offer an opportunity for more in-depth analysis and validation. While the overall number remains consistently fewer, journals also recorded a higher number of publications in 2024, indicating that some ideas presented at conferences developed into important storylines for full-fledged studies. When researchers present new ideas at conferences and develop them into a journal article, it is a balance between rapid dissemination of info and the scientific rigor to make the work or the thinking known to the world.

3.4. RQ3: What Are the Research Types in Studies on Generative AI for Code Translation?

Figure 2 shows that Evaluation Research (ER) is the dominant type, as it accounts for almost 42% of the overall research volume with a clear peak in 2024, meaning many sought out researched endeavors and published articles relative to this year in an attempt to evaluate generative means based on the works of others. Thus, as this technology gains prominence, researchers are evaluating the reliability and effectiveness of new approaches. Second, the solution proposal (SP) accounts for almost 28% of the total research volume. Its frequency increases similarly to that of ER across the years, peaking in 2024, and thus it is clear that researchers were looking to propose new solutions almost as much as they were testing and validating their results. Third, the Combined Evaluation Research/Solution Proposal (ER/SP) represents almost 25% of the total research output. Its frequency also peaks in 2024. Finally, review studies (Re) are the least frequent at almost 5% of the total research output, with a slight peak in 2023 [12,13].

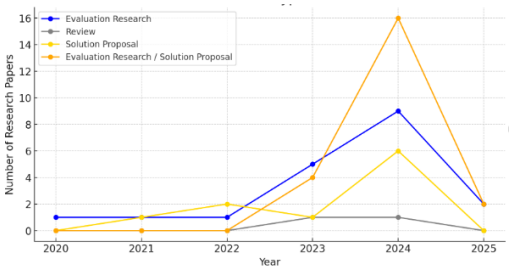

Figure 2. Evolution of research types identified over the years.

3.5. RQ4: What Are the Most Commonly Used AI Models and Architectures for Code Translation?

The heatmap presented in Figure 3 shows that in 2024, Large Language Models (LLMs) had 18 studies. The Transformer-based models had three studies in 2024 and two studies in 2023, Instruct-tuned LLMs and Tree-to-Tree LSTM networks were also incorporated in 2024 studies as a tool to experiment various variables. Individual studies since 2023 have centrally focussed on using an LLM or transformer model.

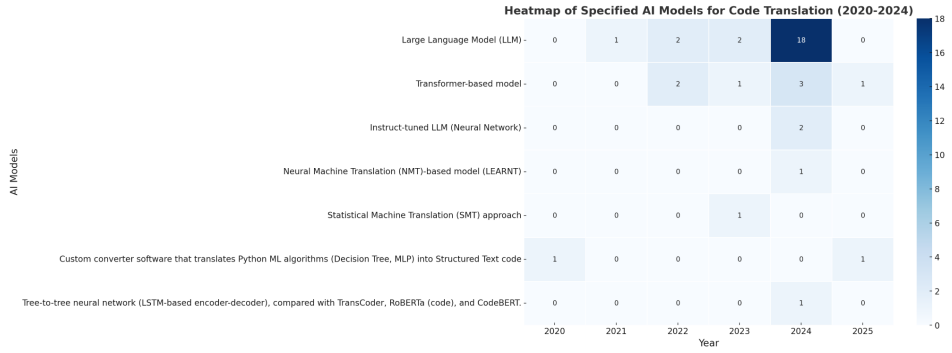

Figure 3. Evolution of research types identified over the years.

From 2020 to 2024, LLMs became the most used models for code translation, leading with 18 studies in 2024 due to their contextual and semantic capabilities [1,10]. Transformers remained consistent across years for their sequence modeling strengths [13], while instruct-tuned LLMs emerged for more precise translation. Legacy models like NMT and SMT declined in use due to limited complexity handling [4]. Isolated uses of custom tools and LSTM-based models appeared experimentally. Overall, the trend favors LLMs and Transformers for their robustness and adaptability.

3.6. RQ5: Which Datasets Are Frequently Used for Training and Evaluating Generative AI Models for Code Translation?

Table 3 shows that the second most popular dataset in the generative AI code translation field is CodeBERT, appearing in seven instances. Next, CodeSearchNet, CodeNet, GraphCodeBERT appear in three instances.

Table 3. Frequently used datasets.

Dataset	Number of Studies	References
CodeBERT	7	[14]
CodeSearchNet	3	[15]
CodeNet	3	[16]
GraphCodeBERT	3	[17]
TransCoder	1	[18]
CoNaLa	1	[19]

CodeBERT is lauded because it is pre-trained in various programming languages and tasks and possesses great semantics and syntax of code representation. Such reliance by researchers indicates a consensus that it effectively produces accurate translations of code [20]. Following CodeBERT, the other most cited datasets are CodeSearchNet, CodeNet, and GraphCodeBERT, which are each found in three studies. This indicates that in relation to the rest of these three studies, they are of similar value. These are large-scale code bases and coding examples that allow for generation and translation tasks across the board [13].

3.7. RQ6: What Types of Empirical Studies Have Been Conducted in the Application of Generative AI for Code Translation?

An examination of empirical types as shown in Figure 4 across the generative AI studies from 2020 to 2025 shows that the highest is "Not Mentioned" in 2024 among the highest occurrences. The second highest, "Experimental Study," rises significantly from 2023. Yet "Case Study" and "Historical Based Evaluation" remain consistently low over all years.

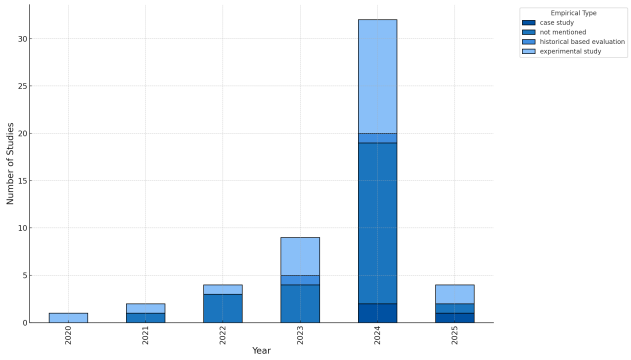

Figure 4. Distribution of empirical types (2020–2025).

The early years of 2020-21 suggest that only a few studies were conducted as the field was still feeling out Generative AI potentials with code translation. The year range of 2022–2023 is much higher as 2022 was the year of feasibility testing, suggesting that by 2023, things had leveled out enough for dedicated research and testing to occur, implying that the functionality and accuracy of Generative AI had stabilized enough for researchers to explore within. The fact that it occurs when the technology matures indicates that a great deal was spent in assessment and established avenues yet still a great deal needs to be done, either from a standstill of questioning the avenues or from a realization that gaps exist which need sustained efforts where avenues are solidified.

4. Threats to Validity

The main threats to the validity of this study can be looked at from three angles:

- Study Selection: To identify the relevant papers for this mapping study, we formulated a search string aligned with our research questions, which we used to conduct an automatic search across different digital databases related to this line of work. We then applied specific selection criteria to filter the results. However, some relevant studies may not have been captured through this search process. To address this limitation, we reviewed the reference lists of the selected papers to identify and include additional pertinent studies.
- Publication Bias: As many studies report only positive findings in the context of this study, we reduce mitigate accept dismiss this threat by including comparisons to models studied with unsatisfactory alternatives. A common tendency in the literature is for researchers to emphasize the superior performance of their proposed models,

which may contribute to an overestimation of their actual effectiveness. To mitigate this threat to validity, we introduced a third inclusion criterion aimed at selecting studies that conduct models comparisons studied with unsatisfactory alternatives.

- Data Extraction Bias: To minimize the risk of inaccuracies during this process, two authors independently assessed each selected paper and extracted the necessary information pertinent to addressing the research questions.

5. Conclusions

This systematic mapping study's results provide a holistic overview of the generative AI code translation research realm from 2020 to 2025. Ultimately, 53 relevant studies were identified, mapped, and categorized according to the year/presence in channels, type of research, type of publication, type of empirical study, and largely referenced AI models. The past five years contributed to the majority of publications in the past decade in channels with conference proceedings and journals with an academic inclination; ER and SP are the most common types of research, which means there is a good balance between empirical testing of code translation implementations and proposing new directions. The most referenced AI models are LLMs, meaning that all other new proposals are merely subsets under them.

The most effective way to achieve accurate and efficient code translation for uniquely created solutions is still through the generative transformer-based architecture. When it comes to code translation, LLMs not only gained popularity since 2020 but also, in 2025, slightly depreciate the relevance of NMT and SMT. The most frequently used datasets are CodeBERT, CodeSearchNet, and CodeNet, all useful for training and testing implementations.

Therefore, this review suggests that in the last decade, the current developments in code translation via generative AI have made significant progress. The future seems to steer towards transformer-based foundations and tested applications in the real world. This paper should be used as a foundation for future research within this expanding field.

Author Contributions: Conceptualization, A.R.; methodology, A.R.; software, A.R.; validation, A.R.; formal analysis, A.R.; resources, A.R., I.C. and M.R.; writing—original draft preparation, A.R.; writing—review and editing, A.R., I.C. and M.R.; visualization, A.R.; supervision, I.C. and M.R.; project administration, A.R., I.C. and M.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Macedo, M.; Tian, Y.; Cogo, F.R.; Adams, B. Exploring the Impact of the Output Format on the Evaluation of Large Language Models for Code Translation. In Proceedings of the International Conference on AI Foundation Models and Software Engineering (Forge), Lisbon, Portugal, 14 April 2024.
2. Luo, Y.; Yu, R.; Zhang, F.; Liang, L.; Xiong, Y. Bridging Gaps in LLM Code Translation: Reducing Errors with Call Graphs and Bridged Debuggers. In Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering (ASE), Sacramento, CA, USA, 27 October–1 November 2024.
3. Gandhi, S.; Patwardhan, M.; Khatri, J.; Vig, L.; Medicherla, R.K. Translation of Low-Resource COBOL to Logically Correct and Readable Java leveraging High-Resource Java Refinement. In Proceedings of the 1st International Workshop on Large Language Models for Code (LLM4Code), Lisbon, Portugal, 20 April 2024.

4. Huang, D.; Liu, X.; Wei, J.; Jiang, L. A Survey on Neural Code Translation. In Proceedings of the International Conference on Neural Information Processing (ICONIP), Auckland, New Zealand, 2–6 December 2024; Springer: Berlin/Heidelberg, Germany, 2024.
5. Li, T.R.; Zhang, H.; Chen, P.; Qian, X. A Comparative Study of LLMs in Code Translation. In *ACM Conference on Software Language Engineering (SLE)*; ACM: New York, NY, USA, 2024.
6. Petersen, K.; Feldt, R.; Mujtaba, S.; Mattsson, M. Systematic Mapping Studies in Software Engineering. In Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE), Bari, Italy, 26–27 June 2008.
7. Fang, D.; Song, P.; Zhang, Z. LLM-Powered Code Translation with Semantic Preservation. In Proceedings of the International Conference on Software Engineering (ICSE), Lisbon, Portugal, 14–20 April 2024.
8. Kitchenham, B.; Charters, S. *Guidelines for Performing Systematic Literature Reviews in Software Engineering*; EBSE: Durham, UK, 2007.
9. Fernández, D.M.; Passoth, J.-H. Empirical software engineering: From discipline to interdiscipline. *J. Syst. Softw.* **2019**, *148*, 170–179. [CrossRef]
10. Oliveira, F.; Costa, A.; Barreto, R.; Quintero, S. LLM-Coded: A Code Translation Framework for Improving Software Compatibility. In Proceedings of the International Conference on Software Maintenance and Evolution (ICSME), Flagstaff, AZ, USA, 6–11 October 2024.
11. He, J.; Yang, Z.; Sun, L.; Tang, Q. Cross-Language Code Translation via LLM-Based Transfer Learning. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Jeju, Korea, 3–9 August 2024.
12. Zhu, M.; Karim, M.; Lourentzou, I.; Yao, D.D. Semi-Supervised Code Translation Overcoming the Scarcity of Parallel Code Data. In Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering (ASE), Sacramento, CA, USA, 27 October–1 November 2024.
13. Lei, B.; Ding, C.; Chen, L.; Lin, P.-H.; Liao, C. Creating a Dataset for High-Performance Computing Code Translation using LLMs: A Bridge Between OpenMP Fortran and C++. In Proceedings of the High Performance Extreme Computing Conference (HPEC), Boston, MA, USA, 25–29 September 2023.
14. Feng, Z.; Guo, D.; Tang, D.; Duan, N.; Feng, X.; Gong, M.; Shou, L.; Qin, B.; Liu, T.; Jiang, D.; et al. CodeBERT: A Pre-Trained Model for Programming and Natural Languages. *arXiv* **2020**, arXiv:2002.08155.
15. Husain, H.; Wu, H.-H.; Gazit, T.; Allamanis, M.; Brockschmidt, M. CodeSearchNet Challenge: Evaluating the State of Semantic Code Search. *arXiv* **2019**, arXiv:1909.09436.
16. Puri, R.; Kung, D.S.; Janssen, G.; Zhang, W.; Domeniconi, G.; Zolotov, V.; Dolby, J.; Chen, J.; Choudhury, M.; Decker, L.; et al. CodeNet: A Large-Scale AI for Code Dataset for Learning a Diversity of Coding Tasks. *arXiv* **2021**, arXiv:2105.12655.
17. Guo, D.; Ren, S.; Lu, S.; Feng, Z.; Tang, D.; Liu, S.; Zhou, L.; Duan, N.; Svyatkovskiy, A.; Fu, S.; et al. GraphCodeBERT: Pre-training Code Representations with Data Flow. *arXiv* **2020**, arXiv:2009.08366.
18. Lachaux, M.-A.; Roziere, G.; Chanussot, L.; Lample, G. Unsupervised Translation of Programming Languages. *arXiv* **2020**, arXiv:2006.03511. [CrossRef]
19. Xu, F.F.; Jiang, Z.; Yin, P.; Vasilescu, B.; Neubig, G. Incorporating External Knowledge through Pre-training for Natural Language to Code Generation. In Proceedings of the the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 6045–6052.
20. Xiao, C.; Tang, M.; Zhao, W. Deep Transfer Learning for Code Translation Across Languages. In Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), Barcelona, Spain, 25–29 August 2024; ACM: New York, NY, USA, 2024.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.