Minireview Article

Byte by Byte: A 5 Year Bibliometric Review of Programming Language Research Dynamics in Southeast Asia (2018-2023)

ABSTRACT

Aims: To conduct a systematic examination and bibliometric analysis of Scopus-indexed literature focusing on emerging trends in programming languages research within Southeast Asia.
Study design: This study employs a mixed method approach, incorporating both qualitative and bibliometric analysis.
Place and Duration of Study: Publication data for review was obtained from the Scopus database, covering the period from 2018 to 2023, with a specific focus on the progress in programming language and semantics research within ASEAN countries.
Methodology: We used the Preferred Reporting Items for Systematic Reviews and Meta-analysis (PRISMA) protocol to collect publication data. Bibliometric data was visualized through Biblioshiny and VOSviewer.
Results: From 2018 to 2023, the research production involving programming languages and semantics across ASEAN countries has been strong, yielding a total of 233 documents from 160 unique sources. However, the annual growth rate was at -10.87%. There were a total of 882 authors with only 10 sole authors in the field. 46.78% of the documents had international co-authorship with an average of 4.03 authors per document. The literature spanned across 764 unique author keywords and 7424 citations with an average of 11.12 citations per document.
Conclusion: Southeast Asia has a rich and collaborative research space in the field of Programming Languages but it faces several barriers such as the absence of a unified research agenda, the lack of adequate funding, and the relatively weak industrial base.

Keywords: Programming Language, Semantics, Scopus, ASEAN.

1. INTRODUCTION

The domain of Programming Language research is a field that broadly focuses on the design, implementation, analysis, and evaluation of programming languages and their compilers. Programming Languages are the fundamental tools for computer science and software development and these tools have to evolve over time to meet the changing needs and challenges of various domains and applications. Programming Language research in the context of the ASEAN is faced with challenges that are specific to the characteristics of the region such as linguistic diversity, socio-economic conditions, and a diverse level of ICT development index. [1]

According to Ko (2016), programming language research can be viewed from several perspectives, such as technical, socio-technical, and cultural. While programming languages viewed from a technical point of view is simply just a formal means for specifying computer behavior, Ko posits that it is important to value the socio-technical aspect – as a human-computer communication tool, and cultural aspect – as artifacts that reflect the values, norms, and practices of the communities that use them. [2]

With the rise of NLP research, it has never been more essential to ensure that non-western nations do not get left behind. Some works [3] have discussed the feasibility of Large Language models, such as ChatGPT, as compilers for 4th generation programming languages. It is important, therefore, to review and assess how the ASEAN region keeps pace with the global world.

In this paper, we present a systematic literature review of the emerging trends in programming languages research in Southeast Asia between 2017 and 2023. The period that we seek to investigate is marked by rapid technological advancements in AI, NLP, and Programming Languages, and it is crucial to understand how these trends are reflected in the ASEAN region. It seeks to understand the major contributors, the evolution of programming languages research, the key themes and topics, and potential future directions of this research in the ASEAN region. By recognizing the key players, the driving forces behind the progress in this field in the region can be better understood. The paper will also be looking at the practical applications that exist in the field of programming languages in various sectors which requires an examination of the socio-economic and cultural factors that may have influenced these trends. The findings of this review will not only contribute to the existing literature of knowledge but also help mold the future trajectory of programming language research in Southeast Asia.

1. 1 Research Questions

1. What are the emerging trends in programming languages research in Southeast Asia between 2017 and 2023?
2. Who are the major contributors (individuals and institutions) to the advancements in programming languages research in Southeast Asia during this period, and what impact has their work had on the field?
3. How has the use and application of programming languages evolved over time in Southeast Asia, and what factors have influenced these changes?
4. What are the key themes and topics in programming languages research in Southeast Asia?
5. Based on the current research landscape, what potential future directions can be anticipated for programming languages research in Southeast Asia?

1. 2 Conceptual Framework

We describe the topic using the materials of Hallinger and Kovačević (2019) as the conceptual framework of the study. [4]

1. Size: Initially, 184,807 Scopus-indexed articles were filtered through PRISMA with a final document size of 233. The selection and filtering ensures that all materials are relevant and contribute to the research space being investigated.
2. Time: We filtered papers down to the last 5 years from 2018-2023. This search space ensured that all material collected is recent and relevant. Furthermore, a trend in AI and NLP research began around 2018 which is worth investigating.
3. Space: The analysis aims to study the ASEAN research programming language research space only.

4. Composition: Clustering and Network visualization was done through VOSviewer and other relevant tools to identify the composition that makes up the research space.

2. MATERIALS AND METHODS

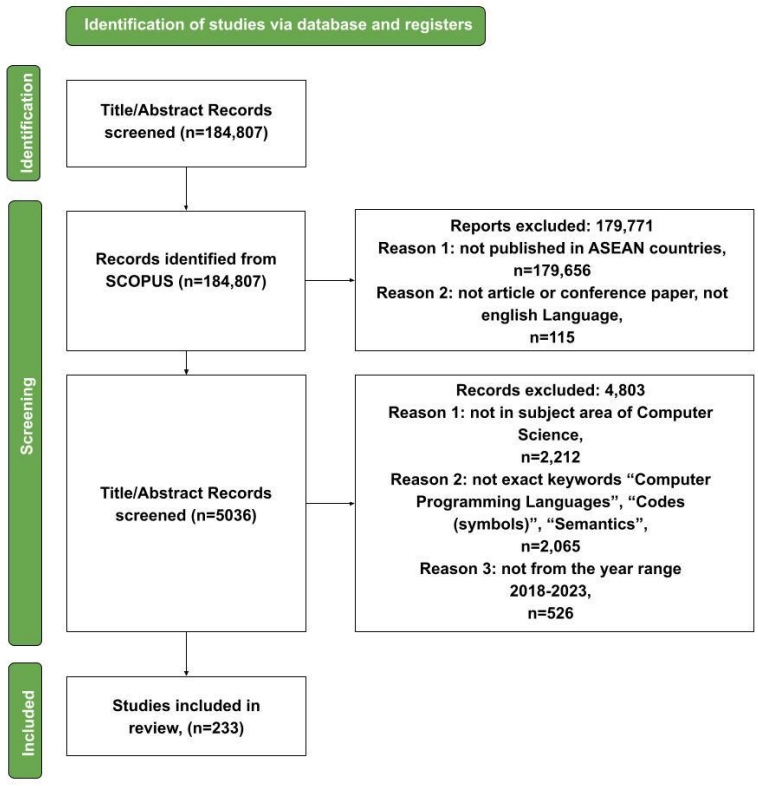
This study employs a mixed method approach, incorporating both qualitative and bibliometric analyses. The study utilized publication data obtained from the Scopus database, covering the period from 2018 to 2023, with a specific focus on the progress in programming language and semantics research within ASEAN countries. Visualizations will incorporate those presented by Linnenluecke, Marrone, Singh (2020), Moral-Muñoz (2020), Hallinger, & Kovačević (2019), and López-Herrera, Herrera-Viedma, & Herrera (2011). [5, 6, 4, 7]

2.1 Collection of Publication Data

The study utilized the Preferred Reporting Items for Systematic Reviews and Meta-analysis (PRISMA) protocol to collect publication data [8]. The process, encompassing identification, screening, and inclusion stages, adhered closely to the PRISMA guidelines. It is important to highlight that this study exhibits a bias towards publications indexed in Scopus.

The acquisition of publication data, in accordance with the PRISMA procedure [8], and relevant 'n' values are depicted in Figure 1.

Figure 1. Data collection procedure using PRISMA methodology



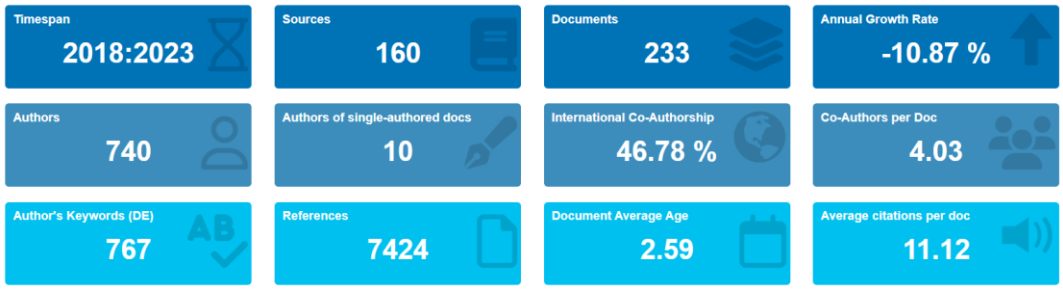
3. RESULTS AND DISCUSSION

In this section, the findings from a comprehensive investigation into computer programming languages and semantics advancements across ASEAN countries are presented. This overview utilized Biblioshiny, VosViewer, and scopus functions.

3.1 Main Information

Figure 2 provides an overall summary of the compiled information. Spanning from 2018 to 2023, the research production involving programming languages and semantics across ASEAN countries has been strong, yielding a total of 233 documents from 160 unique sources. However, the field demonstrates a notable negative annual growth rate of -10.87%, which indicates a possible loss of interest of the ASEAN countries in this domain of research. Collaboration is also evident with the involvement of 882 authors, with only 10 acting as sole authors. International co-authorship also plays an important role, constituting 46.78% of collaborations, while the average co-authorship per document is at 4.03 authors. The depth of the literature is highlighted by the 767 unique author keywords and a comprehensive reference list featuring 7424 citations. Lastly, it is also important to note that the average number of citations per document is 11.12, confirming the influence and importance of publications in this field of study.

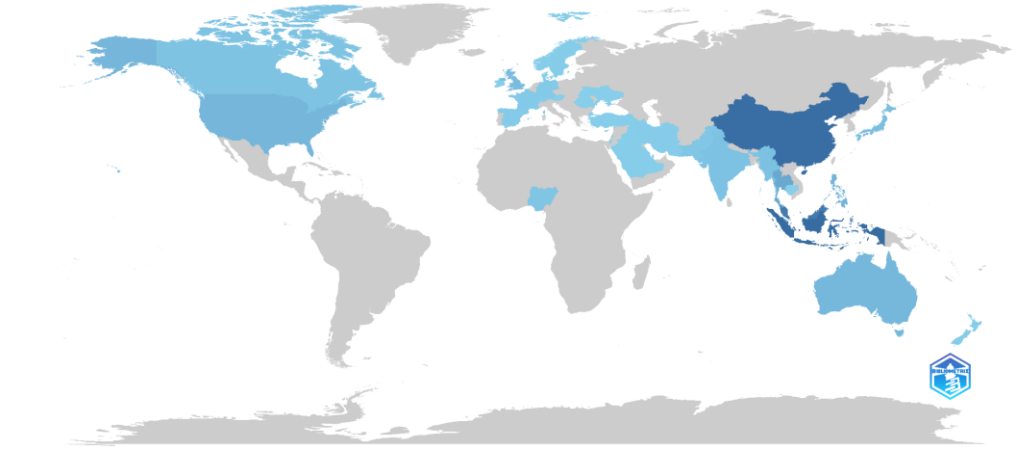
Figure 2. Main Information



3.2 Publication productivity heat map

Figure 3 shows the publication productivity heat map. Those portions shaded blue means that there were scientific productions related to programming languages in those regions. The darker blue shades indicate a higher number of production.. Based on the given graph, main contributors to the subject at hand are Singapore, Indonesia, China. And Malaysia. Following these countries in the number of contributions are Thailand, Philippines, USA, Australia, UK, and Japan. It is noteworthy that while our topic of interest is emerging trends in programming languages research in ASEAN, the heat map reveals contributions from nations beyond this region. This observation underscores a global reach in research collaboration, with researchers in ASEAN countries engaging in partnerships beyond their regional and continental boundaries. This highlights a broader international participation in the subject matter, fostering a collaborative and inclusive research environment.

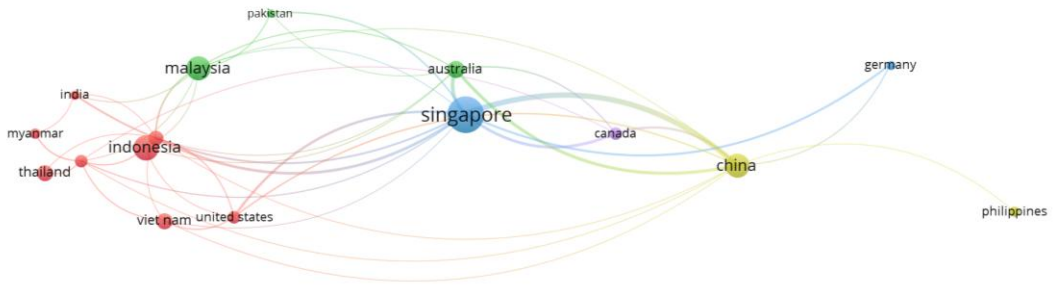
Figure 3. Publication productivity heat map



3.3 Co-authorship among countries

Figure 4 illustrates the co-authorship among countries in the field of programming languages research, utilizing a threshold of a minimum of five documents per country to generate the graph. The graph's inclusion of countries beyond the ASEAN region highlights the global importance and interest in programming languages research. Singapore emerges as a central hub, actively collaborating with a wide array of countries worldwide, including China, Canada, Germany, Australia, Pakistan, Malaysia, the UK, the US, Vietnam, Japan, and India. Singapore's pivotal role in nurturing these extensive international partnerships underscores their dedication to accumulating knowledge and advancing research in the field, demonstrating the global significance of the subject. Notably, the graph exclusively showcases ASEAN countries, specifically Singapore, Malaysia, Vietnam, Indonesia, Myanmar, Thailand, and the Philippines. This selection, despite the primary focus on emerging trends in programming language research within ASEAN, emphasizes broader impact and collaboration in the broader global context.

Figure 4. Co-authorship among countries



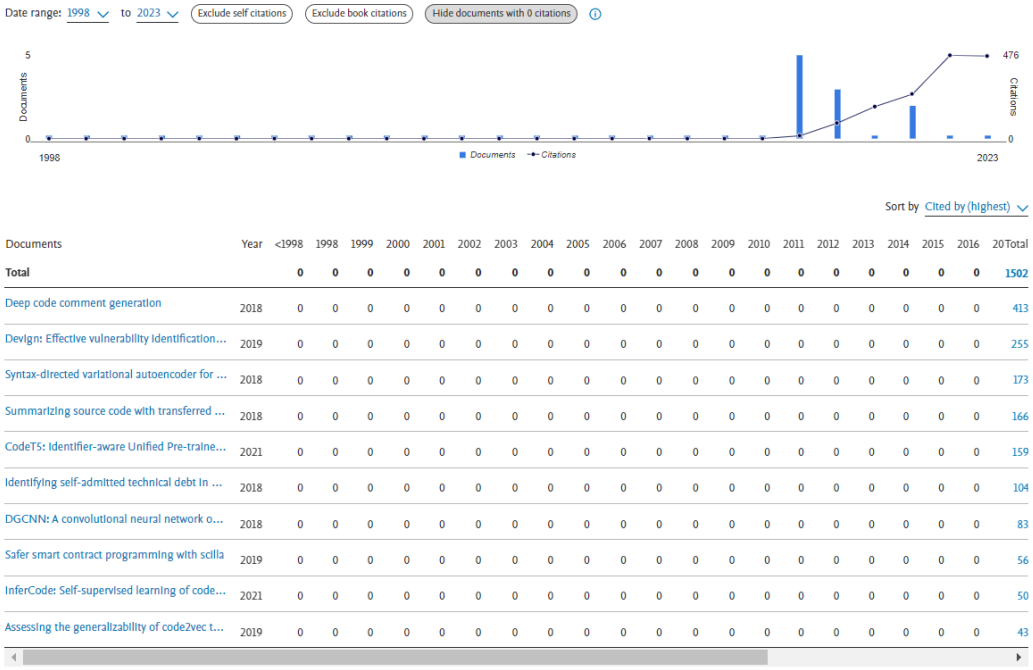
3.4 Citation overview for the top 10 cited documents

In the realm of programming languages, the top 10 cited documents provide a comprehensive overview of significant advancements and trends. Topping the list is the work on "Deep Code Comment Generation" [9] with an impressive 413 citations, reflecting the community's keen interest in the automatic generation of meaningful comments for code. Following closely is "Devign: Effective Vulnerability Identification by Learning Comprehensive Program Semantics via Graph Neural Networks," [10] an innovative approach employing graph neural networks to effectively identify vulnerabilities by learning comprehensive program semantics, garnering 255 citations. The significance of structured data is underscored by the "Syntax-directed Variational Autoencoder for structured data," [11] a paper with 173 citations, showcasing the importance of capturing and generating structured information. Additionally, the role of API knowledge in code summarization is highlighted by "Summarizing Source Code with Transferred API Knowledge" (166 citations) [12]. "CodeT5: Identifier-aware Unified Pre-trained Encoder-Decoder Models for Code Understanding and Generation" (159 citations) [13], emphasizes the prominence of pre-trained models in understanding and generating code. Notably, the exploration of self-admitted technical debt in open source projects using text mining is addressed by "Identifying Self-admitted Technical Debt with Context-Based Ladder Network" (104 citations) [14], shedding light on the practical implications of text analysis in software engineering. The intersection of neural networks and graph structures is evident in "DGCNN: A Convolutional Neural Network over Large-scale Labeled Graphs" (83 citations) [15], showcasing the increasing relevance of graph-based approaches.

Furthermore, the importance of secure smart contract programming is emphasized by "Safer Smart Contract Programming with Scilla" (56 citations) [16], illustrating the growing concern for robustness in blockchain applications. Self-supervised learning takes center stage in "InferCode: Self-Supervised Learning of Code Representations by Predicting Subtrees" (50 citations) [17], where code representations are learned by predicting subtrees, marking a shift towards unsupervised techniques in code analysis.

Lastly, the assessment of generalizability in token embeddings is explored by "Assessing the Generalizability of Code2Vec Token Embeddings" (43 citations) [18], shedding light on the ongoing efforts to enhance the adaptability of code representations. In summary, these top-cited documents collectively highlight the diverse facets of code analysis, ranging from deep learning techniques and vulnerability identification to the exploration of program semantics and the integration of API knowledge, underscoring the multidimensional nature of research in this domain.

Fig 5. Top 10 cited documents



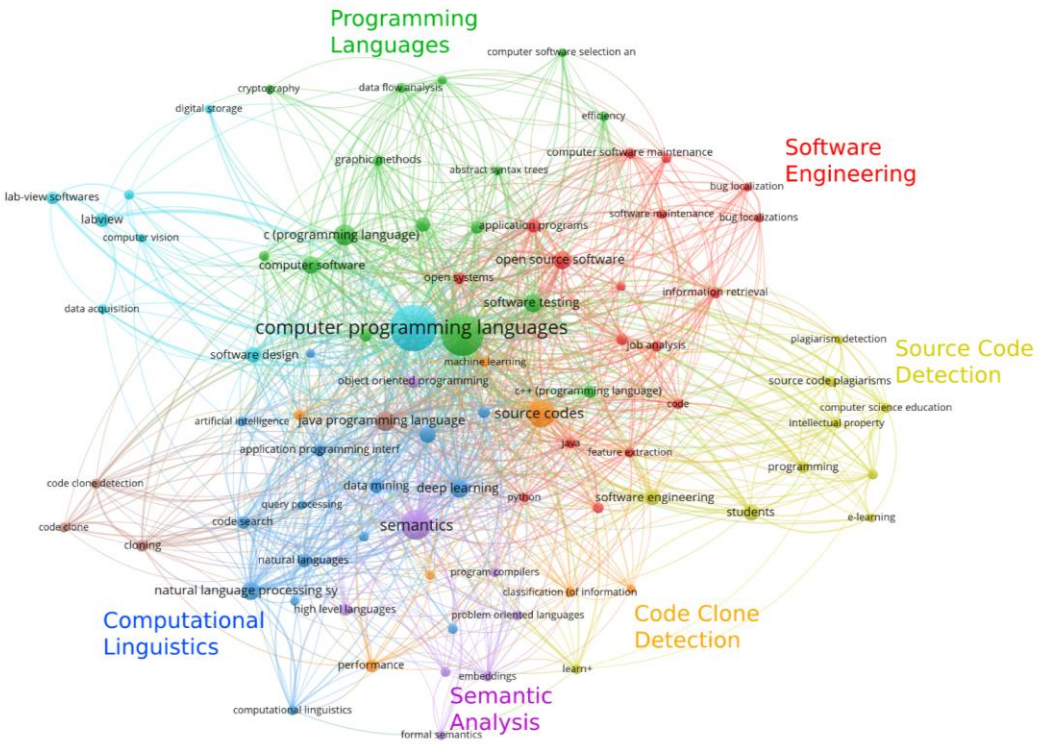
Sort by Cited by (highest)

Documents	Year	<1998	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	20Total
Total		**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1502**
Deep code comment generation	2018	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	413
Devign: Effective vulnerability identification...	2019	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255
Syntax-directed variational autoencoder for ...	2018	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	173
Summarizing source code with transferred ...	2018	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	166
CodeT5: Identifier-aware Unified Pre-traine...	2021	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	159
Identifying self-admitted technical debt in ...	2018	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	104
DGCNN: A convolutional neural network o...	2018	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	83
Safer smart contract programming with scilla	2019	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	56
InferCode: Self-supervised learning of code...	2021	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50
Assessing the generalizability of code2vec t...	2019	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	43

3.5 Co-occurrence of keywords

In Figure 6, key co-occurring clusters are shown. The most prominent node, "computer programming languages," acts as a nexus for various thematic clusters. The blue cluster revolves around the application and evolution of programming languages, delving into topics such as application programming interfaces, deep learning, and natural languages. Meanwhile, the green cluster focuses on the efficiency and cryptographic aspects of computer software, incorporating data flow analysis and graphic methods. The red cluster explores open source software, software maintenance, bug localization, and information retrieval, shedding light on critical aspects of software development and upkeep. The yellow cluster, encompassing plagiarism detection, intellectual property, and e-learning, unveils the intersection of programming languages with academic integrity and educational methodologies. The orange cluster zeroes in on source codes, performance, and the classification of information, elucidating the integral role programming languages play in system optimization. The purple cluster delves into semantics, embeddings, and problem-oriented languages, shedding light on the nuances of language design and application. Lastly, the brown cluster encapsulates discussions on cloning, code clone detection, and the ethical dimensions of these practices. Our findings highlight the multidimensional landscape of programming languages in the ASEAN region, providing a nuanced understanding of the prevailing trends and paving the way for further exploration and innovation in this dynamic field.

Figure 6. Co-occurence of keywords



3.6 Discussion

The research on computer programming languages and semantics has been growing steadily in recent years. This is due in part to the increasing complexity of software systems, which has made it more difficult to develop and maintain them. As a result, there is a growing need for research that can help to improve the quality and reliability of software.

Based on the clusters found in figure 6 illustrating the co-occurence of keywords among the documents found (n=233), there were 6 distinct topics that were discussed. These topics revolved around Software Engineering and Application (n=100); Code Optimization and Clone Detection (n=13); Computational Linguistics, NLP and Comment Generation (n=43); General Programming Language Theory and Analysis (n=28); Source Code Detection and Plagiarism Checkers (n=13); and Semantic Analysis and Syntax Tree generation (n=36).

The cluster for software engineering and application revolved around addressing practical problems in the development lifecycle and contributing to the improvement of software quality, efficiency, and maintainability [19, 20]. These works were motivated by challenges and issues faced by industry professionals and organizations in various fields, most of these works also utilized some form of machine learning [21, 22].

For the cluster of code optimization and clone detection the papers generally discussed security aspects in code [10], program analysis and profiling [23], and code clone detection removal [24]. The papers delved into different techniques in identifying duplicated or similar

codes, understanding the behavior of software, and addressing security vulnerabilities in code using machine learning.

The cluster of computational linguistics and natural language processing generally discussed natural language to code translation [25] and sentiment analysis [26]. These papers revolved around the development of computational models and algorithms that can understand, generate, and manipulate human language, and some papers also delved into code comment generation [9].

The cluster for general programming language theory and analysis discussed formal language syntax and semantics, programming language design principles, and some also delved into the improvement of introductory learning of programming languages using artificial intelligence [27].

For source code detection and plagiarism checkers the published works focused on the development, improvement, and evaluation of techniques and tools designed to identify instances of code similarity, reuse, and plagiarism, studies here also utilized the power of machine learning [28, 29].

The cluster for semantic analysis and syntax tree generation narrowed in on the formal methods and algorithms used to understand how programming languages represent meaning and its practical applications in compiler construction, program analysis, and language design [30, 31].

From the clusters above, one of the most significant trends in the field of computer programming languages and semantics is the growing use of machine learning and artificial intelligence techniques [32, 33]. These techniques can be used to automate a variety of tasks that are currently performed manually by software engineers, such as code generation, testing, and debugging. This can help to reduce the cost and time of software development, and it can also improve the quality of the resulting software.

Another emerging trend in the field of computer programming languages and semantics is the use of structured data to represent programs [17, 34]. This approach has the potential to improve the efficiency and accuracy of a variety of programming language tasks, such as code analysis and refactoring.

The research on computer programming languages and semantics is a rapidly evolving field. As new technologies emerge, researchers are finding new and innovative ways to use them to improve the development and maintenance of software [35, 30, 32]. This research is essential to the continued growth and evolution of the software industry.

Another trend that is gaining traction in the Southeast Asian programming language research community is the development of new programming languages [36, 37]. These new languages are designed to address the specific needs of developers in the region. For example, some of these languages are designed to be more efficient than existing languages, while others are designed to be more secure.

The research on computer programming languages and semantics is a critical part of the software development process. By understanding the underlying principles of programming languages and semantics, researchers can develop new tools and techniques that can help to improve the quality and reliability of software.

The Southeast Asian community of programming language research is growing rapidly. This is due in part to the increasing number of universities and research institutions in the region that are offering programs in computer science and software engineering. Additionally, there is a growing demand for software developers in Southeast Asia, which is driving the need for research in programming languages.

This growing programming language research community is diverse, with researchers from a variety of countries and backgrounds. This diversity is a strength of the community, as it brings together different perspectives and approaches to research. The community is also collaborative, with researchers working together to solve common problems and share their findings.

The Southeast Asian programming language research community is making significant contributions to the field of computer science. The research being conducted by this community is helping to improve the quality and reliability of software, and it is also helping to develop new programming languages that are better suited to the needs of developers in the region.

There are several significant challenges that impact the research community's growth and development. One major obstacle is the absence of a unified research agenda, which can hinder collaboration and the establishment of common goals within the community. Additionally, the lack of adequate funding for research poses a considerable barrier, limiting the community's ability to pursue innovative projects and address pressing issues. The region also grapples with the challenge of a relatively weak industrial base, which may impede the practical application of research findings and collaboration with industry stakeholders. Furthermore, the scarcity of a critical mass of researchers compounds these challenges, as it may lead to a limited pool of expertise and hinder the community's overall impact. Addressing these challenges is crucial for fostering a robust and thriving programming language research community in Southeast Asia.

Overcoming challenges in the Southeast Asian programming language research community involves developing a unified research agenda, increasing funding, building industry partnerships, and attracting more researchers. By fostering collaboration, securing financial support, integrating with industry, and expanding the talent pool, the community can enhance its cohesion, impact, and sustainability.

The Southeast Asian programming language research community has the potential to become a world leader in the field. By addressing the challenges facing the community, and by working together to achieve a common goal, the community can make significant contributions to the field of computer science.

4. CONCLUSION AND RECOMMENDATION

4.1 Conclusion

In conclusion, the dynamic field of computer programming languages and semantics is experiencing rapid evolution, particularly in Southeast Asia where a vibrant and collaborative research community is actively engaged.

Several compelling trends are shaping the landscape, driven by key factors. Firstly, the escalating complexity of software systems underscores the need for programming languages and semantics that can express system behavior with clarity and conciseness, fueling research into more expressive and user-friendly options.

Secondly, the increasing importance of securing and ensuring the reliability of software, particularly as it plays a critical role in our infrastructure, is propelling investigations into programming languages and semantics that can detect and prevent errors while resisting security threats.
Lastly, the advent of cutting-edge technologies like machine learning and artificial intelligence is further propelling research in programming languages and semantics to accommodate the intricate relationships between data and models.

These trends collectively contribute to the dynamic and impactful landscape of computer programming languages and semantics research in Southeast Asia. The trends outlined above are just a few of the many exciting developments that are taking place in this field. As the field continues to evolve, it is likely to play an increasingly important role in the development of secure, reliable, and efficient software systems.

The Southeast Asian programming language research community faces several barriers, such as the absence of a unified research agenda, the lack of adequate funding, and the relatively weak industrial base.

4.2 Recommendation

We recommend that the Southeast Asian programming language research community take the following steps to address the challenges it faces:

4.2.1 Develop a unified research agenda.

The software engineering community in Southeast Asia is a diverse one, with researchers from a variety of countries and backgrounds. This diversity can be a strength, but it can also make it difficult to focus the community's efforts and facilitate collaboration. One way to address this challenge is to develop a unified research agenda. This agenda would identify key areas of research that are important to the software engineering community in Southeast Asia as a whole. It would also provide a framework for organizing and prioritizing research efforts.

A unified research agenda would have a number of benefits. First, it would help to focus the community's efforts and make it easier to identify and pursue the most important research questions. Second, it would facilitate collaboration by providing a common framework for researchers to work together. Third, it would help to ensure that research findings are relevant to the needs of practitioners in Southeast Asia.

There are a number of ways to develop a unified research agenda. One approach would be to convene a group of experts from the software engineering community in Southeast Asia to identify key areas of research. Another approach would be to survey the software engineering community in Southeast Asia to identify the most pressing research needs. Once a list of key research areas has been identified, the next step would be to develop a framework for organizing and prioritizing research efforts. This framework could be based on a number of factors, such as the importance of the research area, the availability of resources, and the potential for impact.

Developing a unified research agenda is a complex and challenging task, but it is an important one. A unified research agenda would help to strengthen the software engineering community in Southeast Asia and make it more effective in addressing the challenges of software development in the region.

4.2.2 Increase funding for research.

Software engineering research is essential to the advancement of the field. However, funding for software engineering research in Southeast Asia is often limited. This is a problem because it prevents the software engineering community in Southeast Asia from pursuing innovative projects and addressing pressing issues.

There are a number of ways to increase funding for software engineering research in Southeast Asia. One approach is to lobby governments to increase funding for software engineering research. Another approach is to partner with industry to fund research projects. A third approach is to create endowments to support software engineering research.

Increasing funding for software engineering research in Southeast Asia is critical to the future of the field. By investing in software engineering research, we can help to improve the quality of software, reduce the cost of software development, and accelerate the software development process in the region.

4.2.3 Strengthen ties with industry stakeholders.

Industry stakeholders have a shared interest in the creation of quality software. However, the research and industrial groups often have different perspectives on how to achieve this goal. The software engineering community can strengthen its ties with industry stakeholders by:

- Participating in industry conferences and workshops.
- Collaborating with industry on research projects.
- Providing consulting services to industry.
- Hiring industry professionals as faculty members.

By strengthening its ties with industry stakeholders, the software engineering community in Southeast Asia can ensure that its research findings are relevant to the needs of practitioners and that they have a real impact on the software development process in the region.

We believe that these steps will help to foster a robust and thriving programming language research community in Southeast Asia.

REFERENCES

1. Measuring the Information Society Report. ITU. 2017. Accessed 13 September 2023. Available: http://handle.itu.int/11.1002/pub/80f52533-en.

2. Ko J. What is a programming language, really?. 2016. Accessed 14 September 2023. doi: https://doi.org/10.1145/3001878.3001880.

3. Marcondes F, Almeida J, Novais P. Large Language Models: Compilers for the 4th Generation of Programming Languages? (Short Paper). Schloss Dagstuhl - Leibniz-Zentrum für Informatik. 2023. doi: https://doi.org/10.4230/OASIcs.SLATE.2023.10

4. Hallinger P, Kovačević J. A Bibliometric Review of Research on Educational Administration: Science Mapping the Literature, 1960 to 2018. Review of Educational Research. 2019. doi: https://doi.org/10.3102/0034654319830380.

5. Linnenluecke MK, Marrone M, Singh AK. Conducting Systematic Literature Reviews and Bibliometric Analyses. Australian Journal of Management. 2019. doi: https://doi.org/10.1177/0312896219877678.

6. Moral-Muñoz A, Herrera-Viedma E, Santisteban-Espejo A, Cobo M. J. Software tools for conducting bibliometric analysis in science: An up-to-date review. El Profesional de la Información. 2020. doi: https://doi.org/10.3145/epi.2020.ene.03.

7. Cobo MJ, López-Herrera AG, Herrera-Viedma E, Herrera F. Science mapping software tools: Review, analysis, and cooperative study among tools. Journal of the American Society for Information Science and Technology. 2011. doi: https://doi.org/10.1002/asi.21525.

8. Page MJ., McKenzie JE, Bossuyt PM, Boutron I, Hoffmann TC, Mulrow CD, et al. The PRISMA 2020 Statement: An Updated Guideline for Reporting Systematic Reviews. Systematic Reviews 2021. doi: https://doi.org/10.1186/s13643-021-01626-4.

9. Hu X, Li G, Xia X, Lo D, Jin Z. Deep Code Comment Generation. Proceedings of the 26th Conference on Program Comprehension. 2018. doi: https://doi.org/10.1145/3196321.3196334.

10. Zhou Y, Liu S, Siow J, Du X, Liu Y. Devign: Effective Vulnerability Identification by Learning Comprehensive Program Semantics via Graph Neural Networks. 2019. doi: https://doi.org/10.48550/arXiv.1909.03496.

11. Dai H, Tian Y, Dai B, Skiena S, Song L. Syntax-Directed Variational Autoencoder for Structured Data; International Conference on Learning Representations, ICLR. 2018. doi: https://doi.org/10.48550/arXiv.1802.08786.

12. Hu X, Ge L, Xia X, Lo D, Lu S, Jin Z. Summarizing Source Code with Transferred API Knowledge. Singapore Management University Institutional Knowledge (InK) (Singapore Management University) 2018. doi: https://doi.org/10.24963/ijcai.2018/314.

13. Wang Y, Wang W, Shafiq J, Steven. CodeT5: Identifier-Aware Unified Pre-Trained Encoder-Decoder Models for Code Understanding and Generation. 2021. doi: https://doi.org/10.18653/v1/2021.emnlp-main.685.

14. Gong A, Fukumoto F, Panitan Muangkammuen, Li, J, Yu, D. Identifying Self-Admitted Technical Debt with Context-Based Ladder Network. Communications in computer and information science 2023; 84–97. doi: https://doi.org/10.1007/978-981-99-8184-7_7.

15. Phan AV, Nguyen ML, Nguyen YL, Bui, LT. DGCNN: A Convolutional Neural Network over Large-Scale Labeled Graphs. Neural Networks 2018;108:533–543. doi: https://doi.org/10.1016/j.neunet.2018.09.001.

16. Sergey I, Nagaraj V, Johannsen J, Kumar A, Trunov A, Hao KCG. Safer Smart Contract Programming with Scilla. Proceedings of the ACM on Programming Languages. 2019;3(OOPSLA):1–30. doi: https://doi.org/10.1145/3360611.

17. Bui Q, Yu Y, Jiang L. InferCode: Self-Supervised Learning of Code Representations by Predicting Subtrees. arXiv (Cornell University). 2020. doi: https://doi.org/10.48550/arxiv.2012.07023.

18. Kang HJ, Bissyandé, TF, Lo D. Assessing the Generalizability of Code2vec Token Embeddings. IEEE Xplore. doi: https://doi.org/10.1109/ASE.2019.00011.

19. Le Ba Cuong, Van Son Nguyen, Đức Anh Nguyễn, Phạm Ngọc Hùng, Dinh Hieu Vo. JCIA: A Tool for Change Impact Analysis of Java EE Applications. Advances in intelligent systems and computing. 2018;105–114. doi: https://doi.org/10.1007/978-981-10-7512-4_11.

20. Rachmawati M, Budiman A, Magdalena Batubara, M. Enhancing File Security by using Vigenere Cipher and Even Rodeh Code Algorithm. 2018 International Conference on Computer, Control, Informatics and Its Applications (IC3INA); 2018.

21. Clark RA, Pua Y.-H. SeeSway – A free web-based system for analysing and exploring standing balance data. 2018;159:31-36. doi: https://doi.org/10.1016/j.cmpb.2018.02.019.

22. Anacan R, Cabautan A, Cayabyab JM, Miguel SX, Modrigo V, Rosites CJ, et al. Development of Oil Quality Estimator Using Machine Vision System. 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM). 2018; doi: https://doi.org/10.1109/HNICEM.2018.8666427

23. Mokhtar NN, Mubarak-Ali A.-F, Hamza MAM. Enhanced Pre-processing and Parameterization Process of Generic Code Clone Detection Model for Clones in Java Applications. International Journal of Advanced Computer Science and Applications. 2020; doi: https://doi.org/10.14569/ijacsa.2020.0110669.

24. Arshad S, Abid S, Shamail S. CodeBERT for Code Clone Detection: A Replication Study. 2022 IEEE 16th International Workshop on Software Clones (IWSC). 2022;39-45.

25. Chen F, Fard FH, Lo D, Bryskin T. On the transferability of pre-trained language models for low-resource programming languages. Singapore Management University Institutional Knowledge (InK) (Singapore Management University). 2022; doi: https://doi.org/10.1145/3524610.3527917.

26. Aminuddin R, Zulkefli AZ, Moketar NA, Khyrina Airin Fariza AS. Sentiment analysis of online product reviews using Lexical Semantic Corpus-Based technique. 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE). 2021; doi: https://doi.org/10.1109/ISCAIE51753.2021.9431818

27. Wee C, Yap KM, Lim WN. iProgVR: Design of a Virtual Reality Environment to Improve Introductory Programming Learning. IEEE Access. 2022;10:100054–100078. doi: https://doi.org/10.1109/access.2022.3204392.

28. Suttichaya V, Eakvorachai N, Lurkraisit T. Source Code Plagiarism Detection Based on Abstract Syntax Tree Fingerprintings.2022 17th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP). 2022; doi: https://doi.org/10.1109/iSAI-NLP56921.2022.9960266.

29. Rahiman NFA, Gapar JMd, Rabab Alayham Abbas H. CopyPoppy – A Source Code Plagiarism Detector. 2022 IEEE 10th Conference on Systems, Process & Control (ICSPC). 2022; doi: https://doi.org/10.1109/ICSPC55597.2022.10001740

30. Zhang L, Liu C, Xu Z, Chen S, Fan L, Chen B, Liu Y. Has My Release Disobeyed Semantic Versioning? Static Detection Based on Semantic Differencing. arXiv (Cornell University); 2022.
doi: https://doi.org/10.48550/arXiv.2209.00393

31. Chen X, Lin Z, Trinh M-T, Grigore R. Towards a Trustworthy Semantics-Based Language Framework via Proof Generation. In: Silva, A., Leino, K.R.M. (eds) Computer Aided Verification. CAV 2021. Lecture Notes in Computer Science(), vol 12760. Springer, Cham; 2021.

doi: https://doi.org/10.1007/978-3-030-81688-9_23

32. Binh NT, Minh NNH, Hanh LTM. Ml-Codesmell: A code smell prediction dataset for machine learning approaches. In Proceedings of the 11th International Symposium on Information and Communication Technology (SoICT '22). Association for Computing Machinery. 2022;368–374.
doi: https://doi.org/10.1145/3568562.3568643

33. Schumi R, Sun J, ExAIS: Executable AI Semantics. In Proceedings of the 44th International Conference on Software Engineering (ICSE '22). Association for Computing Machinery. 2022; 859-870.
doi: doi.org/10.1145/3510003.3510112

34. Islam R, Bui Q, Wang K, Yu Y, Jiang L, Alipour MA. On the generalizability of Neural Program Models with respect to semantic-preserving program transformations. Information and Software Technology, IST Journal 2021. 2021.
doi: https://doi.org/10.1016/j.infsof.2021.106552.

35. Schumi R, Sun J. Semantic-Based Neural Network Repair. In Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2023). Association for Computing Machinery. 2023; 150-162.
doi: https://doi.org/10.1145/3597926.3598045

36. Yin X, et al. SafeOSL: Ensuring memory safety of C via ownership- based intermediate language.. Software: Practice and Experience, vol. 52, no. 5. 2021; 1114-1142.
doi: https://doi.org/10.1002/spe.3057.

37. Lin Y, Cheng X, Gao D. Control-Flow Carrying Code. Singapore Management University Institutional Knowledge (InK) (Singapore Management University). 2019.
doi: https://doi.org/10.1145/3321705.3329815.