

## **ANALYZING STACK OVERFLOW DISCUSSIONS ON C, JAVA, AND PYTHON: A MIXED-METHOD STUDY ON QUESTION TYPES AND TOPICS**

**Yusuf Sulisty Nugroho<sup>\*1</sup>, Aldin Nasrun Minalloh<sup>2</sup>, Keke Rachma Devi<sup>3</sup>, Syful Islam<sup>4</sup>**

<sup>1,2,3</sup>Informatics Engineering, Faculty of Communication and Informatics, Universitas Muhammadiyah Surakarta, Indonesia

<sup>4</sup>Computer Science and Engineering, Engineering Faculty, Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Bangladesh

Email: <sup>1</sup>[yusuf.nugroho@ums.ac.id](mailto:yusuf.nugroho@ums.ac.id), <sup>2</sup>[L200214208@student.ums.ac.id](mailto:L200214208@student.ums.ac.id), <sup>3</sup>[L200190165@student.ums.ac.id](mailto:L200190165@student.ums.ac.id), <sup>4</sup>[syfulcse@bsmrstu.edu.bd](mailto:syfulcse@bsmrstu.edu.bd)

(Article received: December 08, 2024; Revision: January 01, 2024; published: January 04, 2024)

### **Abstract**

*The modern software development characteristic is significantly shaped by the evolution of programming languages. The increasing complexity of these languages demands effective tools and resources for learning and troubleshooting. As a result, forums such as Stack Overflow (SO) have become crucial for addressing technical issues that arise during program execution, especially for novice programmers. Although discussions on SO are common, there hasn't been a clear description of the question types and topics for the three main programming languages, i.e., C, Java, and Python. This gap is problematic as it limits the ability of educators, platform designers, and developers to effectively address the specific needs of users. Without such insights, novice programmers may struggle to find relevant guidance, potentially hindering their learning and slowing the adoption of best practices. To fill this gap, we conducted a qualitative and quantitative study on these three language-related discussions shared on SO. By utilizing a dataset of 4,499,718 questions extracted from SOTorrent, we applied a manual labeling method to classify questions into categories such as "How," "What," and "Why." Furthermore, we implemented Latent Dirichlet Allocation (LDA) for topic modeling to understand the prevalent discussion topics. The results show that "How" questions dominate across all languages, particularly in Python (60.94%), reflecting a high demand for practical implementation guidance. Analysis of discussion topics indicates that C is centered on system programming and low-level operations, while Java discusses more on application development and object-oriented programming. In contrast, Python focuses more on data handling and structures. These insights suggest that while practical support is necessary for learners, a deeper understanding of programming concepts and the need for customized instructional resources to support developers are important. The findings contribute to the community and relevant fields by offering actionable insights to improve the usability of SO as a learning and problem-solving platform.*

**Keywords:** *discussion topic, programming language, question type, Stack Overflow.*

## **1. INTRODUCTION**

The current development of software involves a wide range of technologies, tools [1], platforms, and programming languages [2]. Programming languages have emerged as a prominent tool, serving as a means of communication between humans and computers [3]. Despite their utility, novice programmers often face significant challenges in applying these languages [4], necessitating robust instructional resources [5]. This need has prompted experts to create various instructional materials [6], such as books, files, and online tutorials. These resources provide significant benefits to learners and can contribute to the growing popularity of certain programming languages [7].

As the availability of resources and the activity of developers increases, the popularity of programming languages also tends to grow [8], [9]. However, programmers often encounter issues when using programming languages, such as errors during program execution [10], [11]. To address the problems they face, many programmers turn to online question-and-answer platforms like Stack Overflow (SO) [12]. SO, has emerged as an invaluable platform for addressing programming challenges [13]. Programmers from various backgrounds [14] can post questions on SO covering a variety of topics [15]. In addition, SO accommodates a wide range of user needs, including discussions on specific topics such as network simulators [16], [17] and react library [18].

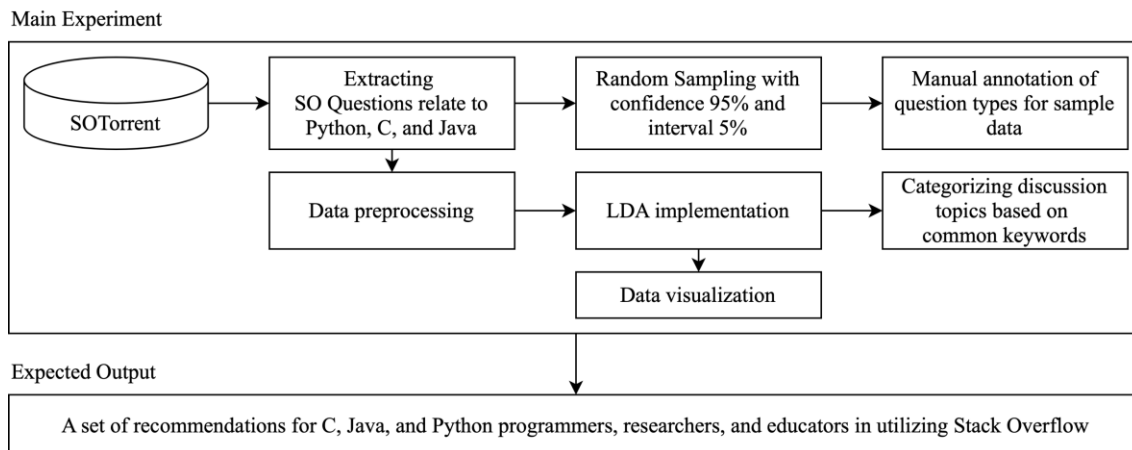


Figure 1. The procedure of the study, includes two main parts: (i) main experiment which begins with data collection from SOTorrent and move on to qualitative and quantitative analyses, and (ii) output of the study.

Numerous studies on SO have provided insights into topics of interest to developers [15]. One study exploring SO discussions identified eight topics related to network simulators, which were subsequently grouped into five main categories [16]. Another study suggested that machine learning techniques can be used to identify key terms of interest to developers [19]. SO also plays a crucial role in shaping the online discussion paradigm [20], [21] with more than 80 other Q&A websites adopting the same foundational platform [13]. Software developers utilize SO to discuss and exchange ideas on various topics, including programming languages [22]. Despite previous studies highlighting the significance of SO, no research has specifically analyzed the types of questions and topics most frequently discussed on SO related to the three programming languages: C, Java, and Python.

The challenges faced by users, specifically novice programmers, such as debugging errors or understanding abstract programming concepts [23], underscore the importance of this study. By analyzing SO discussions, specifically related to C, Java, and Python, this research provides insights into common problems and solutions that can directly benefit beginner programmers. For example, identifying the prevalence of “How” questions highlights the need for practical guidance, while understanding the topic of discussion helps in adjusting resources to address specific challenges [24].

Based on the background, this study analyzes the questions posted by programmers on SO related to the three programming languages: C, Java, and Python. These languages were selected due to their status as the most popular programming languages since 2019 [25]. By applying qualitative and quantitative methods, we provide insights into common programming challenges, facilitating educators and developers with targeted strategies for effective learning. The objective of this study is to understand the types of questions and discussion topics associated with these three languages on the SO platform. This research contributes to the field of

computer science, specifically software engineering area, by shedding light on the dynamics of SO discussions, highlighting its role as an important resource for both novice and experienced programmers, particularly regarding C, Java, and Python.

## 2. RESEARCH METHOD

In this study, we conducted our experiments by following the procedure as illustrated in Figure 1. The details of the procedure are explained as follows.

### 2.1. Data Collection

The analysis of question types and discussion topics posted by programmers on SO begins with the data collection process. This step involves gathering the necessary data for the research. As shown in Figure 1, the data is sourced from SOTorrent [26]. In the first phase of data collection, we applied three specific keywords to filter the questions based on their tags, i.e. “python,” “C,” and “java.” After removing duplications, this process yielded a dataset containing 4,499,718 questions related to three programming languages discussed on SO, as presented in Table 1.

Table 1. Number of SO questions related to each programming language per August 2024, where Python has been the most discussed language on SO, followed by Java and C.

Programming Language	Number of Questions
C	403,205
Java	1,912,899
Python	2,183,614
Total	4,499,718

### 2.2. Analysis

1) *Type of Questions*: This study begins by manually labeling the collected dataset to identify the most frequently discussed question types for each of the three programming languages on SO. The labels used to categorize question types follow those from previous research [16], as described below:

- **How:** Questions asking for instructions on how to perform a task. For example, “How to use the C socket API in C++ on z/OS.” This question seeks instructions on using the C socket API with C++.
- **What:** Questions requesting more abstract, conceptual information, seeking decision support, or inquiring about non-functional requirements. For example, “On what platforms is JavaFX supported?” This question asks which platforms support JavaFX.
- **Why:** Questions asking for reviews, reasons, or explanations for something. For example, “Why are Java and Python's collection methods different?” This question seeks an explanation for the differences between the collection methods in the two programming languages.
- **Others:** Questions that cannot be classified using keyword searches within the title and body of the post.
- **404 (Not Found):** Questions that are no longer accessible.

Before manual labeling begins, the required sample size for each programming language is determined. The sample size is calculated using a sample calculator with a confidence level of 95% and an interval of 5% [27]. Based on these calculations, a random sample of 384 questions is selected for each programming language. Subsequently, the first and third authors individually label the first 30 samples based on the defined question types. The Kappa agreement score is then calculated using a Kappa Calculator. In this step, the Kappa score for all programming languages is 80%, which is categorized as “substantial agreement.” Based on this motivating score, the manual labeling was then carried out by the first author for the remaining sample data.

2) *Discussion Topics:* In this analysis, we applied topic modeling to uncover latent data and automatically determine the topics shared on SO discussions by programmers. To perform this study, we utilized Latent Dirichlet Allocation (LDA) due to its popularity as a topic modeling technique in software engineering [28]. LDA identifies hidden topic structures by discovering word mixtures related to other topics [29]. Typically, LDA works by taking individual documents and several parameters as input and then producing an output model consisting of normalized weights based on probabilities. It involves converting documents into a dictionary format and then into document or corpus form, followed by applying the LDA algorithm to form topic models [30]. These probabilities refer to two types: (a) the probability that a specific document generates a particular topic, and (b) the probability that a specific topic generates particular words from a given vocabulary.

The topic modeling process using LDA in this study involves several steps:

- Preparing a dataset of questions from the three programming languages: C, Java, and Python.
- Cleaning the data by removing unnecessary elements such as punctuation and symbols from the dataset using the `remove_punctuation` function. Moreover, links, HTML tags, stopwords, and other irrelevant elements are removed.
- Preprocessing the data to structure the text through two stages: *tokenizing*, which splits the text into individual words or tokens, and *stemming*, which groups words with similar root meanings but different affixes.
- Converting each text into a vector using the `CountVectorizer` function from the `sklearn` library. Then, the LDA model is applied by specifying the number of topics to be modeled in the vectorized data.

Using the result from LDA, we visualized the common keywords that appear in the discussions of each programming language. To visualize them, we implemented a word cloud, also known as a text cloud, a visual representation method for displaying text data [31], where word frequency in a selected written material, such as lecture notes, book text, or website content, is visually represented [32].

Based on the common keywords mentioned by programmers in the SO discussions, we then categorized them into 6 general discussion topics, as follows:

- **Data Handling and Structures:** This topic covers discussions about how data is organized, stored, and manipulated in programming languages. Furthermore, the topic includes specific data handling structures such as columns and dataframes in Python. Keywords: *data, variable, array, value, list, struct, multiple, column, dataframe, pointer*.
- **Error Handling and Debugging:** This topic deals with the challenges that developers face when encountering errors or bugs in their code and how they resolve them. Keywords: *error*.
- **String Manipulation and File Handling:** This topic involves how strings are processed and manipulated within the language, and how files are handled, including reading, writing, and parsing. Keywords: *string, char, file*.
- **Application Development and Object-Oriented Programming:** This topic focuses on building applications, especially using object-oriented programming principles like classes and methods. Keywords: *application, object, method, class, function, program*.
- **Libraries and Frameworks:** This topic addresses the use of libraries and popular frameworks that simplify development, particularly in web and mobile applications. Keywords: *pandas, django, spring, android*.

- **System Programming and Low-Level Operations:** This topic includes lower-level programming tasks, particularly those related to operating systems, hardware interaction, and system-level programming. Keywords: *linux, code, memory, function*.

Finally, the results of this study are used as the foundations to construct a set of recommendations for programmers (especially for beginners), researchers, and teachers in utilizing SO platforms. Collectively, these actions aim to maximize the potential of SO as a learning and problem-solving resource, building a more inclusive and effective ecosystem for the global programming community.

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Type of Questions

Table 2. Frequency of questions by type in 3 programming languages. The “How” type dominates the questions across all languages, highlighting the significant need for programmers for hands-on guidance.

Types	Number of Questions					
	C		Java		Python	
How	199	(51.82%)	171	(44.53%)	234	(60.94%)
What	89	(23.18%)	139	(36.20%)	104	(27.08%)
Why	88	(22.92%)	72	(18.75%)	45	(11.72%)
Others	8	(2.08%)	1	(0.26%)	0	(0.00%)
404	0	(0.00%)	1	(0.26%)	1	(0.26%)
<b>Total</b>	<b>384</b>	<b>(100.00%)</b>	<b>384</b>	<b>(100.00%)</b>	<b>384</b>	<b>(100.00%)</b>

The results presented in Table 2 indicate that the “How” type of questions dominates across all three programming languages. In Python, 60.94% of the questions fall under this category, followed by C with 51.82% and Java with 44.53%. This suggests that programmers frequently seek guidance on implementing specific instructions or functions, particularly in Python, which might be attributed to its versatility and widespread use in various domains such as data science and web development. The high percentage of “How” questions reflects the community's need for practical implementation

support in these languages, emphasizing the importance of clear instructional resources.

In contrast, the “What” and “Why” question types occur less frequently. For Java, “What” questions constitute 36.20% of the sample, while Python and C account for 27.08% and 23.18%, respectively. This suggests that Java users may focus more on conceptual or abstract aspects of the language. “Why” questions, which explore reasoning or causes, are less common, with Python having the fewest (11.72%). These insights highlight that while practical, direct guidance (“How”) is crucial, as well as deeper conceptual understanding (“What” and “Why”), particularly in Java.

#### 3.2. Discussion Topics

Figure 2 describes the common keywords extracted from discussions about C, Java, and Python, and highlights the distinct focus areas and applications of each language. In C, as shown in Figure 2a, terms like “linux,” “memory,” “pointer,” and “struct” point to its use in low-level system programming and memory management, emphasizing its role in performance-critical applications. Java discussions, on the other hand, frequently mention “application,” “object,” “method,” and “spring,” suggesting a focus on object-oriented programming, enterprise application development, and mobile development (with “android”), as presented in Figure 2b. As illustrated in Figure 2c, Python, with keywords like “dataframe,” “pandas,” “django,” and “multiple,” is centered around data analysis, scientific computing, and web development, reflecting its versatility and popularity in data-driven and web application projects. All three languages share keywords like “data,” “file,” “error,” and “function,” indicating a common focus on data handling, file management, and debugging across languages.

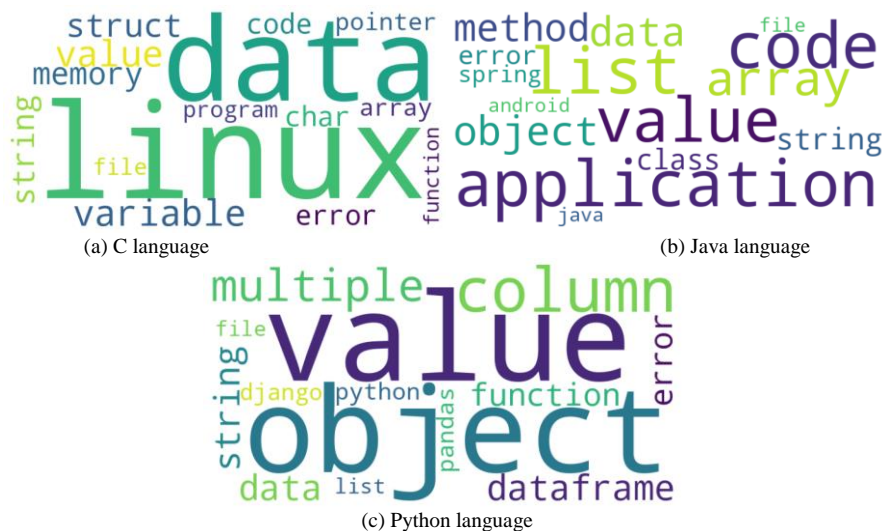


Figure 2. The most frequent keywords appear in the SO discussions of each programming language. Despite their uniqueness, all three languages share similar keywords, such as ‘data,’ ‘error,’ ‘string,’ and ‘file,’ indicating their common topics of discussion.

Table 3. Six discussion topics and common keywords across C, Java, and Python

Discussion Topic	Programming Language	Common Keywords
Data Handling and Structures	C	data, variable, array, struct, pointer
	Java	value, list, array, data
	Python	value, multiple, column, dataframe, data, list
Error Handling and Debugging	C	error
	Java	error
	Python	error
String Manipulation and File Handling	C	string, char, file
	Java	string, file
	Python	string, file
Application Development and OOP	C	function, program
	Java	application, object, method, class
	Python	object, function
Frameworks and Libraries	Java	spring, android
	Python	django, pandas
System Programming and Low-level Operations	C	linux, code, memory, function
	Java	code

Table 3 describes the discussion topics on SO based on keywords across C, Java, and Python. Although it shows that each programming language has its own unique topics, however, there are some similarities. For instance, Data Handling and Structures appears consistently across all three languages, involving keywords such as “data,” “array,” and “value.” C focuses on lower-level structures like “struct” and “pointer,” whereas Java and Python center more around higher-level constructs like “list,” “column,” and “dataframe.” This indicates that while C is often used for low-level data manipulation, Java and Python are utilized for more abstract data handling. Another common theme is Error Handling and Debugging, with “error” being a prevalent keyword in all three languages, suggesting that developers frequently discuss troubleshooting and debugging challenges in SO regardless of the language.

In addition, more specialized topics highlight the distinct use cases of each language. For example, String Manipulation and File Handling is common to all languages. However, Python and Java stand out with a more focus on application development frameworks. Frameworks and Libraries are more emphasized in Java (e.g., “spring,” “android”) and Python (e.g., “django,” “pandas”), reflecting their prevalent use in web development and data science, respectively. Meanwhile, C dominates discussions around System Programming and Low-Level Operations with keywords like “linux,” “code,” “memory,” and “function.” This highlights C language is relevant in performance-critical applications and operating system development, while Java and Python serve as popular choices for building frameworks, providing libraries, and services.

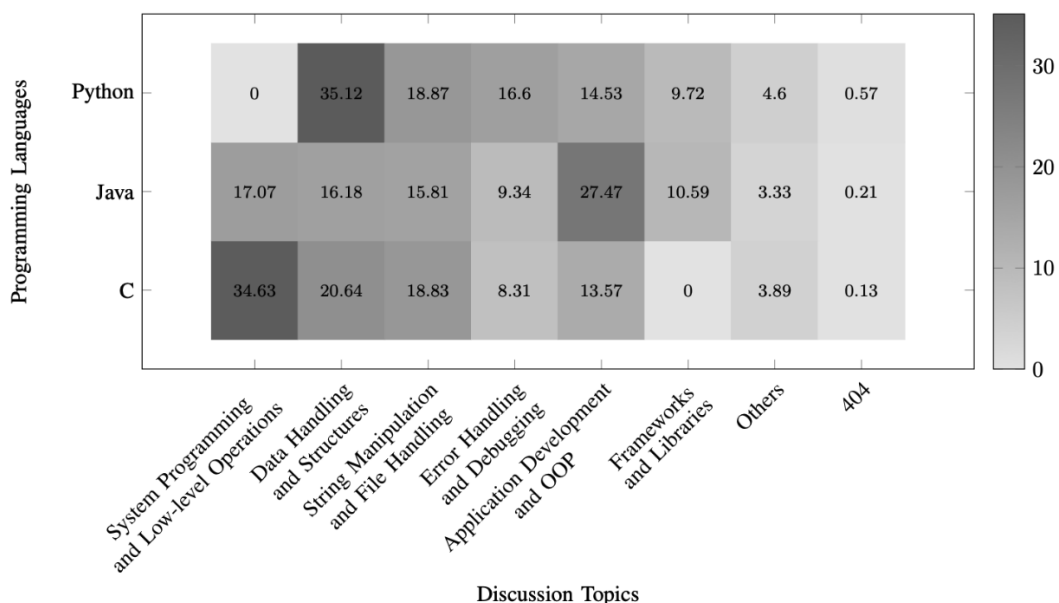


Figure 3. The frequency of topic classification discussed on SO for each programming language. System programming and low-level operations have been found majorly in C, while application development and OOP were the most common topics in Java-related discussions. On the other hand, data handling and structures dominates the Python-related questions shared on SO.

To better understand the distribution of various discussion topics across three programming

languages (C, Java, and Python) in SO, we visualize the result in a heatmap, as shown in Figure 3. It

describes the frequency distribution of discussion topics for three programming languages on Stack Overflow, namely C, Java, and Python. System programming and low-level operations show the highest concentration for C, with a frequency of 34.63%, indicating that C is predominantly discussed the low-level operations. On the other hand, Java has the most discussions around application development and object-oriented programming (OOP), with a notable frequency of 27.47%, reflecting its primary role in enterprise and application-level development.

For Python, the most discussed topic is data handling and structures, with a frequency of 35.12%, which highlights the widespread use of Python in data science and manipulation tasks. Other topics such as string manipulation and error handling are also frequently discussed across languages. Although their frequency varies, Python-related discussions seem relatively balanced distributions in these areas. C, however, shows little engagement with frameworks and libraries, scoring 0% in this category, highlighting its use in more fundamental system-level programming compared to higher-level languages like Python and Java.

This result provides several insights about the different uses and strengths of each language. C is highly specialized for low-level tasks, reflecting its system-oriented nature, whereas Java-related discussions are focused on application development and OOP, confirming its dominance in the software industry. On the other hand, due to its dominance in data-related discussions, Python is seen as a valuable tool in contemporary development environments and an important language for data science and analysis. This comparison offers a clear understanding of how developers utilize these languages based on their fundamental strengths.

## 4. IMPLICATIONS

The findings of this study produce implications for programmers, researchers, and educators, emphasizing the need to address language-specific challenges and optimize the use of SO as a resource:

### 4.1. Implications for Programmers

- SO provides a unique opportunity for developers to access practical guidance and involve in community-driven problem-solving. The high percentage of “How” questions related to Python underscores the need for hands-on implementation resources.
- Developers can benefit from deeper engagement in topic-specific discussions, such as system programming in C or application development in Java, to refine their expertise and broaden their technical skillset.
- Active participation in answering questions not only enriches the SO community but also

increases personal understanding of programming concepts.

### 4.2. Implications for Researchers

- The unique discussion patterns for C, Java, and Python highlight areas requiring further investigation, such as the complexity of unanswered questions and the factors influencing the responsiveness.
- Integrating advanced natural language processing (NLP) techniques may improve the classification of question types and topics, leading to more robust insights.
- Researchers can also use the findings of this study to develop tools or frameworks aimed at addressing common programming challenges across languages.

### 4.3. Implications for Educators

- The insights of this study into the challenges faced by programmers can guide the development of targeted instructional materials, such as practical tutorials for Python and conceptual resources for Java.
- Educators should consider aligning curriculum topics with the most frequently discussed issues on SO, ensuring relevance to students’ real-world programming challenges.
- Motivating students to actively participate in SO can help in developing their critical thinking, problem-solving skills, and a sense of community involvement.

## 5. CONCLUSION

The study of SO discussions on C, Java, and Python shows distinct patterns in the types of questions asked and the topics that discussed about these three programming languages. The “How” type of questions, which focus on the practical implementation of functions or tasks, dominate across all languages, particularly in Python (60.94%), highlighting the need for guidance in solving specific coding problems. On the other hand, conceptual questions, such as “What” and “Why,” appear more frequently in Java, indicating a higher interest in understanding the language's abstract principles, including object-oriented programming and framework usage, as seen with keywords like “object,” “method,” and “spring.”

Regarding discussion topics, every language has unique key topics that correspond with its main use cases. C discussions mostly focus on system programming and low-level operations, with frequent mentions of “linux,” “code,” and “memory.” Java discussions are more oriented toward application development, with keywords like “application,” “object,” and “method” pointing to its use in enterprise and mobile development. Python, on the

other hand, shows a clear focus on data handling and structures, reflected in keywords like “value,” “column,” and “multiple.” These differences highlight how each language is designed to meet the specific needs of developers and how SO is an invaluable tool for solving both conceptual and technical problems.

This study describes the dynamics of SO discussions on C, Java, and Python, providing important insights into the types and topics of questions asked. The findings highlight the practical and conceptual challenges faced by developers, indicating the importance of SO as a learning tool and discussion platform. Future work can explore more advanced analysis techniques, such as deep learning-based natural language processing models, to gain further insights into developer needs and challenges. In addition, a comparative study of how question complexity affects response time and user engagement on SO across different programming languages could provide valuable recommendations for improving developer support on such platforms.

## REFERENCES

- [1] C. Raibulet, F. A. Fontana, and I. Pigazzini, “Teaching software engineering tools to undergraduate students,” in *Proceedings of the 11th International Conference on Education Technology and Computers*, pp. 262-267, 2019.
- [2] M. Marron, “Toward programming languages for reasoning: Humans, symbolic systems, and AI agents,” in *Proceedings of the 2023 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, pp. 136-152, 2023.
- [3] J. Noble and R. Biddle, “programmingLanguage as Language,” *Proceedings of the 2023 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, 2023.
- [4] I. Karvelas, “Investigating Novice Programmers’ Interaction with Programming Environments,” in *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, pp. 336-337, 2019.
- [5] P. Denny, B. A. Becker, N. Bosch, J. Prather, B. Reeves, and J. Whalley, “Novice reflections during the transition to a new programming language,” in *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education-Volume 1*, pp. 948-954, 2022.
- [6] T. Kato, Y. Kambayashi, and K. Oda, “An implementation of educational programming environment using tangible materials,” *Human Systems Engineering and Design (IHSED 2023): Future Trends and Applications*, vol. 112, no. 112, 2023.
- [7] D. M. Arya, J. L. Guo, and M. P. Robillard, “Properties and Styles of Software Technology Tutorials,” *IEEE Transactions on Software Engineering*, vol. 50, p. 159-172, 2023.
- [8] P. Chakraborty, R. Shahriyar, and A. Iqbal, “Empirical analysis of the growth and challenges of new programming languages,” in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1. IEEE, pp. 191-196, 2019.
- [9] P. Chakraborty, R. Shahriyar, A. Iqbal, and G. Uddin, “How do developers discuss and support new programming languages in technical Q&A site? an empirical study of go, swift, and rust in stack overflow,” *Information and Software Technology*, vol. 137, p. 106603, 2021.
- [10] H. Yang, Y. Nong, S. Wang, and H. Cai, “Multi-Language Software Development: Issues, Challenges, and Solutions,” *IEEE Transactions on Software Engineering*, vol. 50, pp. 512-533, 2024.
- [11] N. Brown, P. Weill-Tessier, M. Sekula, A. Costache, and M. Kölling, “Novice Use of the Java Programming Language,” *ACM Transactions on Computing Education*, vol. 23, pp. 1-24, 2022.
- [12] J. Liu, S. Baltes, C. Treude, D. Lo, Y. Zhang, and X. Xia, “Characterizing search activities on stack overflow,” in *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 919-931, 2021.
- [13] I. G. Ndukwe, S. A. Licorish, and S. G. MacDonell, “Perceptions on the utility of community question and answer websites like stack overflow to software developers,” *IEEE Transactions on Software Engineering*, vol. 49, no. 4, pp. 2413-2425, 2022.
- [14] M. Nivala, A. Sereenko, T. Osborne, and T. Hillman, “Stack Overflow – Informal learning and the global expansion of professional development and opportunities in programming?” in *2020 IEEE Global Engineering Education Conference (EDUCON)*, 402-408, 2020.
- [15] A. Barua, S. W. Thomas, and A. E. Hassan, “What are developers talking about? An analysis of topics and trends in stack overflow,” *Empirical software engineering*, vol. 19, pp. 619-654, 2014.
- [16] S. Islam, Y. S. Nugroho, and M. J. Hossain,

- “What network simulator questions do users ask? a large-scale study of stack overflow posts,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 3, pp. 1622-1633, 2021.
- [17] Y. S. Nugroho, S. Islam, D. Gunawan, Y. I. Kurniawan, and M. J. Hossain, “Dataset of network simulator related-question posts in stack overflow,” *Data in Brief*, vol. 41, p. 107942, 2022.
- [18] G. T. Kurniaji, Y. S. Nugroho, and S. Islam, “A preliminary empirical study of react library related questions shared on stack overflow,” *Computer Science and Information Technologies*, vol. 4, no. 1, pp. 14-23, 2023.
- [19] A. A. Bangash, H. Sahar, S. Chowdhury, A. W. Wong, A. Hindle, and K. Ali, “What do developers know about machine learning: a study of ml discussions on Stack Overflow,” in *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*. IEEE, pp. 260-264, 2019.
- [20] H. Zhang, S. Wang, T. H. Chen, and A. E. Hassan, “Reading answers on stack overflow: Not enough!” *IEEE Transactions on Software Engineering*, 47(11), 2520-2533, 2019.
- [21] G. G. Giwangkoro and Y. S. Nugroho, “Unveiling Research Trends in Stack Overflow: A Comprehensive Analysis of General Discussion Theme,” In *Proceeding of the 2024 International Conference on Smart Computing, IoT and Machine Learning (SIML)*, 130-136, 2024.
- [22] R. Abdalkareem, E. Shihab, and J. Rilling, “What do developers use the crowd for? a study using stack overflow,” *IEEE Software*, vol. 34, no. 2, pp. 53-60, 2017.
- [23] H. Oka, A. Ohnishi, T. Nishida, T. Terada, and M. Tsukamoto, “A Choice-Based Programming Learning Method to Develop Problem-Solving Skills,” *IEEE Access*, vol. 12, pp. 119550-119562, 2024.
- [24] W. Yang, C. Zhang, and M. Pan, “Understanding the Topics and Challenges of GPU Programming by Classifying and Analyzing Stack Overflow Posts,” in *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2023.
- [25] D. Lu, J. Wu, Y. Sheng, P. Liu, and M. Yang, “Analysis of the popularity of programming languages in open source software communities,” in *2020 International Conference on Big Data and Social Sciences (ICBDSS)*. IEEE, pp. 111-114, 2020.
- [26] S. Baltes, C. Treude, and S. Diehl, “Sotorrent: Studying the origin, evolution, and usage of stack overflow code snippets,” in *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*. IEEE, pp. 191-194, 2019.
- [27] Y. S. Nugroho, D. Gunawan, D. A. Puspa Putri, S. Islam, and A. Alhefdhi, “A study of vulnerability identifiers in code comments: Source, purpose, and severity,” *Journal of Communications Software and Systems*, vol. 18, no. 2, pp. 165-174, 2022.
- [28] C. C. Silva, M. Galster, and F. Gilson, “Topic modeling in software engineering research,” *Empirical Software Engineering*, vol. 26, no. 6, p. 120, 2021.
- [29] A. Anandkumar, D. P. Foster, D. J. Hsu, S. M. Kakade, and Y.-K. Liu, “A spectral algorithm for Latent Dirichlet Allocation,” *Advances in neural information processing systems*, vol. 25, 2012.
- [30] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet Allocation,” *Journal of Machine Learning Research*, vol. 3, no. Jan, pp. 993-1022, 2003.
- [31] F. Heimerl, S. Lohmann, S. Lange, and T. Ertl, “Word cloud explorer: Text analytics based on word clouds,” in *2014 47th Hawaii International Conference on System Sciences*. IEEE, pp. 1833-1842, 2014.
- [32] W. Cui, Y. Wu, S. Liu, F. Wei, M. X. Zhou, and H. Qu, “Context preserving dynamic word cloud visualization,” in *2010 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, pp. 121-128, 2010.