The ChatGPT Effect: An Analysis of Topic Modeling and User Interaction on StackOverflow

THEODOR-FABIAN NICULAE, University of Twente, The Netherlands
t.f.niculae@student.utwente.nl

This thesis investigates the evolving dynamics of user interactions on Stack Overflow (SO) with the help of advanced topic modeling techniques. By employing methods such as Latent Dirichlet Allocation (LDA), BERTopic, BERTopic fine-tuned with KeyBERT and POS, and BERTopic quantized with LLaMA-3-8B, this thesis analyzed shifts in discussion topics across two distinct periods: April 2021-2022 and April 2023-2024. This research highlights the superiority of BERTopic quantized with LLaMA-3-8B, which greatly improves the coherence and diversity of topics compared to traditional models like LDA. The findings reveal that topics shifted from data manipulation and web development in 2021-2022 to cloud services, deployment strategies and modern JavaScript frameworks in 2023-2024. Additionally, the thesis investigates the impact of generative AI, specifically ChatGPT, on user interactions and content quality on SO. The analysis reveals a notable decrease in overall activity on SO, with fewer questions being posted and answered, slower response time and less average view counts in the later period. Despite the decline in activity, there was an increase in the complexity and detail of the posts. The study also found a shift in the popularity of certain technologies, with newer tools and frameworks gaining traction over traditional ones, such as tags related to AI, large language models and ChatGPT that saw an increase, reflecting the impact of these technologies on the types of questions asked. Through comprehensive empirical analysis, the study addresses research questions related to the evolving landscape of SO discussions. Beyond the empirical analysis between the two periods of time and comparing the different models for extracting the topics, this thesis serves also as a replicable pipeline that includes data gathering, preprocessing, and the application of novel large language model to improve BERTopic for automatic topic extraction. This pipeline offers a practical solution for enhancing SO's tagging system, which currently relies on simple tags like programming languages or high-level tasks. By improving content discoverability, this approach could help SO regain user engagement on the platform.

Additional Key Words and Phrases: Stack Overflow, AI, ChatGPT, Topic Modeling, User Interaction, LDA , BERTopic, LLaMA-3, Technology Trends, Lexical Diversity, Grammatical Structure

1 INTRODUCTION

In recent years, voluntary knowledge contribution on online platforms has become increasingly significant for users, platforms, and firms [45]. This trend is driven by the collaborative nature of the internet, where users share information, solve problems collectively, and contribute to a growing repository of knowledge. These platforms, such as SO [24], Quora [4], and Reddit [5], thrive on the active participation of their communities, where users ask questions, provide answers, and engage in discussions. The collaborative exchange of knowledge on these platforms offers several benefits. For users, it provides a valuable resource for learning and problem-solving. By tapping into the collective expertise of the community, individuals can quickly find solutions to their problems, gain new insights, and improve their skills. For platforms, user-generated content is a key asset, driving traffic, engagement, and ultimately, their value proposition. Firms benefit by gaining access to a wealth of information and insights that can inform product development, customer support, and market strategies. However, the effectiveness of these platforms depends on the active and voluntary participation of their users. The 90-9-1 rule, which suggests that only 1% of users actively create content, 9% respond to it, and 90% simply consume it [45] [44], highlights a critical challenge: motivating users to contribute actively. Strategies to enhance user engagement and contributions include gamification, recognition systems, and providing intrinsic and extrinsic incentives [36]. Despite these efforts, maintaining high-quality contributions remains a challenge. The vast and open nature of these platforms can lead to issues such as misinformation, duplicate content, and varying levels of content quality. Moderation systems, community guidelines, and the implementation of advanced technologies like AI for content curation and quality control are essential to address these challenges [21].

One of the most transformative advancements in this realm is the development and deployment of generative artificial intelligence (AI) models, notably ChatGPT [2], developed by OpenAI. Unlike traditional AI systems that rely on predefined rules, ChatGPT utilizes deep learning techniques to understand and generate text based on the vast datasets it has been trained on. The model's ability to simulate human conversation is underpinned by its architecture, which employs transformer networks [13]. The capabilities of ChatGPT are extensive. It can draft emails, write essays, create poetry, generate code, and answer questions on a variety of topics. This versatility is attributed to its training on diverse datasets that cover a wide array of subjects. The AI's performance is further enhanced by techniques like reinforcement learning from human feedback (RLHF), where human reviewers rate the quality of the AI's responses, which helps fine-tune its outputs to be more aligned with human expectations and preferences [41] [13]. This sophisticated AI, which uses large language models (LLMs) such as GPT-3 and the more advanced GPT-4, has revolutionized the way users interact with digital platforms by providing immediate human-like responses to a wide range of queries.

The introduction of ChatGPT and similar AI tools has had a profound impact on various online platforms, including question and answer (Q&A) sites such as SO, introducing new dynamics to this environment. SO has been a cornerstone for programmers seeking help with coding issues, offering a community-driven space where users can ask questions and provide answers. Several studies have explored the influence of ChatGPT on Q&A platforms, highlighting both the potential benefits and challenges. One significant

Master Thesis, Enschede, The Netherlands,

advantage of ChatGPT is its ability to provide quick and accurate responses, which can be particularly useful for handling repetitive and well-defined queries. This capability allows human users to focus on more complex and creative problem-solving tasks, potentially improving the overall quality of contributions on platforms like SO [30] [31]. However, the integration of ChatGPT also raises concerns. For instance, the presence of AI-generated content may reduce users' motivation to contribute, as they might rely on the AI for answers rather than engaging in knowledge sharing themselves. Additionally, while ChatGPT is known for its politeness and positive sentiment, it can also produce incorrect or misleading information, which poses risks for users who depend on its accuracy for critical tasks [8] [67].

The impact of ChatGPT on user behavior and content quality on SO has been mixed, as evidenced by various empirical studies. For example,[14] discusses the potential displacement of traditional Q&A contributions by AI-generated content. The study highlights how users might shift from asking questions on SO to relying more on AI tools like ChatGPT for immediate answers. Another study, [47] explores the comparative reliability of AI models in providing accurate and helpful responses. Moreover, [33] delves into the evolving trends and research topics within the SO community. This shift suggests a possible decline in human-generated content as users might prefer the convenience and speed of AI responses over traditional community interactions.

While these studies provide valuable insights into various aspects of SO's dynamics, the research aims to differentiate itself by focusing specifically on the temporal comparison of two distinct periods: April 1, 2021 - April 1, 2022, and April 1, 2023 - April 1, 2024. By analyzing changes in SO predefined tags, user's engagement for posting and responding to questions, view counts of questions, different timing analysis for answers and topic trends across these periods, we seek to offer a comprehensive view of how the introduction of ChatGPT has influenced the SO ecosystem.

Notably, most existing studies have analyzed data shortly after the introduction of ChatGPT, but not as recently as April 1, 2024. This extended timeframe allows us to capture more recent trends and provide a more up-to-date analysis of the impact of ChatGPT. For instance, papers [49] [14] note a temporal limitation by stopping at early June 2023, missing out the trends and topics introduced in the subsequent months. A larger group of people may started to use other platforms rather than SO and ask questions about other technologies rather than what were the main topics discussed in 2021-2022, as you're going to see in the later chapters of this paper. Additionally, most of the studies found on this thesis subject discussed in Chapter 2 such as [48],[49], [51], [53], [69], [29] and [12] used a simple LDA model for topic modeling. Advanced techniques such as BERTopic or BERTopic quantized with LLMs , with its ability to comprehend the semantic context of words and phrases, could provide richer and more meaningful topic representations than traditional methods like LDA can, as discussed in the limitations chapter of [49] and based on the evaluation metrics for different topic modeling techniques of [63]. This research will make a comparison between a LDA, BERTopic, KeyBERT, POS, which both are methods that fine-tune the topics generated by BERTopic and also employ BERTopic quantized with the LLaMA-3-8B LLM to provide deeper insights into the underlying themes and enhance the understanding of the text data. By evaluating these models and identifying the best-performing one based on a set of metrics such as Coherence Score and Topic Diversity, which contribute to the overall Topic Quality, we aim to conduct an in-depth analysis of the current activity on SO. This analysis focuses on uncovering the specific topics discussed, as the predefined tags on SO primarily represent programming languages or broad topics that do not provide a detailed understanding of the platform's activity [12]. This limitation in tagging can hinder content discoverability and user engagement. Several studies [62] [19] [7], particularly in the realm of social media optimization, have shown that improving content organization, in this case by refining tag generation through advanced topic modeling techniques, can indeed boost user interaction by making content more easily findable and relevant. When users can find discussions faster and accurate, closer to their intended search, engagement and participation tend to increase, leading to a more active and dynamic community. Through this method, we seek to reveal the actual discussions happening on SO, providing a more accurate and comprehensive view of user interactions and content trends. This approach will help in understanding the evolving nature of technical discussions, identifying emerging trends, and gaining insights into the real issues and areas of interest among the SO community. As shown in the following sections, the number of questions posted on the platform experienced a dramatic decline of over 50% during the second time period analyzed. The methods for extracting topics and the opportunity for utilizing them as question tags could potentially help increase user visits to the website. Additionally, we explored the grammatical structure and lexical diversity of the messages to gain further insights into the nature of the discussions to better understand how the complexity and focus of the posts have evolved.

The paper is organized as follows: it begins with a background section 1.1 that introduces the main components of the study. Next, the research questions 1.2 are outlined, followed by a related work section 2 that discusses the databases used for literature extraction and the criteria for selecting and assessing the papers. A summary of each paper is provided, highlighting key information such as datasets, topic modeling methods, and document analysis techniques. The focus then shifts to the data collection methodology, detailing how the data was obtained using the Stack Overflow API, the challenges encountered, and the high-level overview of the algorithms used. Based on the reviewed papers, the data preprocessing steps are described. The subsequent sections cover various topic modeling methods in detail. The results section answers the research questions, and the paper concludes with a discussion summarizing the findings, a future work section, and the overall conclusions of the study.

1.1 Background

1.1.1 Stack Overflow. SO is an online platform where developers pose questions and exchange knowledge [24]. It aims to provide precise answers to specific programming issues, thereby fostering a collaborative learning environment among developers. Over time, SO has evolved into a vital resource for developers to discuss various topics, contributing to the collective knowledge of the tech community.

1.1.2 Large Language Models LLMs. A LLM is an advanced machine learning model trained on vast amounts of text data to produce text that resembles human writing [35]. These models, which can contain millions to billions of parameters, are employed in numerous natural language processing tasks. For example, LLMs can be applied in topic modeling to identify and categorize underlying themes within a corpus of text, enabling more nuanced insights into data trends and patterns [63].

1.1.3 ChatGPT. ChatGPT, developed by OpenAI, is a language model designed to generate text that mimics human conversation based on given prompts. Built on the GPT architecture, ChatGPT can be utilized for a variety of purposes, including code generation, summarization, translation, testing, and documentation [42].

1.1.4 LLaMA. LLaMA (Large Language Model Meta AI), developed by Meta AI, is a series of LLMs available in different sizes, such as 7,8, 13, and 70 billion parameters [58]. Like ChatGPT, LLaMA models are built on transformer architecture and are capable of generating coherent and contextually relevant text from input prompts. This study utilizes the latest LLaMA version, specifically LLaMA-3 with 8 billion parameters, due to its balance of accessibility and performance, making it suitable for local deployment without extensive computational resources.

1.2 Research Questions

To thoroughly examine these changes, this thesis addresses the main research question:

- How has Stack Overflow's ecosystem evolved in response to the introduction of ChatGPT, and which topic modeling method is most effective for extracting and analyzing these changes?

In order to further explore these dynamics, we also want to answer the following sub-questions:

- How has the overall activity on SO changed from April 1, 2021, to April 1, 2022, compared to April 1, 2023, to April 1, 2024?
- Has the introduction of ChatGPT influenced the types of questions asked on SO?
- How do the different topic modeling methods compare in terms of topic quality?
- What are the main topics of discussion on SO in the specified periods, and how have these topics shifted over time?(based on the topics from the modeling methods)

In order to investigate how SO has evolved after the introduction of ChatGPT, we have chosen these subquestions to focus on. First, we aim to examine the overall activity on the platform by comparing trends between two different time periods. This includes analyzing how many questions were posted each month to determine whether user participation has increased or decreased. We will also evaluate the SO's predefined tag usage to understand which topics or technologies have gained or lost popularity over time. By tracking the number of answers per question and how long it takes to receive an accepted answer, we can assess how responsive the community has been. Additionally, we will examine whether the average number of views per question has changed and how many questions receive accepted answers, to check whether user engagement has shifted over time.

Next, we are exploring whether the types of questions asked on SO have changed since the introduction of ChatGPT. We will look at the frequency of AI-related tags, such as those for AI tools like ChatGPT and large language models, to see if there has been an increase in questions related to these topics. We are also interested in whether AI-related questions receive more answers compared to other types of questions, which could indicate a shift in focus toward AI. To further understand these changes, we will analyze how the complexity of questions has evolved, focusing on the grammatical structure and uniqueness of vocabulary used, which may reflect changes in how users pose questions as AI tools become more widespread.

For answering the second part of the main research question, we will compare different topic modeling methods to determine which one is the most effective for identifying meaningful trends in the data. We will evaluate how coherent and diverse the topics generated by each method are, enabling us to choose the most appropriate model for capturing key changes on the platform. Finally, we will identify the main topics being discussed on SO and track how these topics have shifted over time. By comparing the top topics from the two periods, identified in this case with the models presented in the thesis, the paper aims to understand which areas of interest have gained or lost popularity. These sub-questions were chosen to provide a comprehensive view of how SO has changed, and the detailed analysis will offer insights into how the platform has evolved in the era of AI and ChatGPT.

2 RELATED WORK

In the related work section of this thesis, we concentrate on two main areas to identify relevant literature: studies discussing SO with a focus on topic modeling, trend analysis, or data mining, and studies that specifically address topic modeling in Q&A platforms or for textual data. We delve into the field of topic modeling methodologies, particularly in the context of extracting topics from questions and answers on SO. This exploration addresses a notable gap in existing literature, which has often overlooked the application of natural language processing (NLP) techniques to analyze interactions on SO. Questions on such platforms are dynamic, with rapidly shifting and overlapping topics. Traditional topic modeling techniques may not adequately capture these nuances, but more sophisticated approaches can be designed to understand the flow and evolution of these discussions. This investigation seeks to identify NLP methodologies that can be effectively adapted and implemented in analyzing SO data, even though direct precedents might be limited. The aim is to enhance the understanding of the interactions and trends within this community. Additionally, we aim to explore empirical studies that have analyzed data from SO to understand how trends have changed over time, including the number of questions, answers, and other related metrics in order to have an overview over what conclusions can be drawned and what methods can be used to analyze the questions and their answers.

2.1 Databases and search strategy

In our comprehensive search for academic literature, we utilized a variety of databases including Scopus, ACM, IEEE Explorer, Google Scholar, arXiv, and ResearchGate. Initially, we focused on literature related to "SO," "text mining," "topic modeling," and "trend analysis." Additionally, we expanded our search to include broader terms such as "Q&A platforms", "textual data","text documents" to capture a diverse range of methodologies applicable to our research. To refine the search results, we made some queries more specific than others. For example, incorporating keywords like ('Large Language Model' OR 'LLM') into the initial query significantly reduced the number of documents retrieved, from 888 to 16.

Table 1. Database Search Queries

Search Query
('SO') AND ('text mining' OR 'topic modeling' OR 'trend analysis') AND ('Large Language Model' OR 'LLM')
('Q&A platform') AND ('text mining' OR 'topic modeling' OR 'trend analysis') AND ('Large Language Model' OR 'LLM')
('Large Language Model' OR 'LLM') AND ('text mining' OR 'topic modeling' OR 'trend analysis') AND ('textual data' OR 'text documents')

This strategy of using both broad and targeted queries allowed us to explore the full range of available literature while focusing on specific topics. To further streamline our search, we focused exclusively on journal articles and conference papers, which helped reduce the number of documents. Additional filters, such as language, were applied to concentrate on the most relevant materials. The final step involved a detailed review of titles and abstracts to identify relevant papers and manually exclude those that were irrelevant or redundant. Queries used in searching for papers can be seen in Table 1.

2.2 Inclusion and Exclusion Criteria

To ensure completeness and relevance in our study, specific inclusion and exclusion criteria were established. The inclusion criteria comprised of the following elements:

1. Published peer-reviewed papers and journals examining the use of topic modeling and analysis of SO data or Q&A platform or textual documents data.
2. Conference papers and professional journals sourced from reputable databases, including Scopus, ACM, IEEE Explorer, Google Scholar, arXiv, and ResearchGate.
3. Studies involving empirical analysis of SO data or textual documents, providing insights into various methodologies and their applications for.
4. Articles published in English to ensure the comprehensibility and standardization of the research data.

Our exclusion criteria included articles that did not focus explicitly on topic modeling and empirical analysis in textual documents, studies that only peripherally related to the key research themes such as papers with less insights and descriptions of the methods they've used and evaluations, Non-English publications and secondary sources, to maintain clarity and consistency in the analysis and research lacking in peer review or not published through recognized academic channels.

2.3 Data Extraction and Analysis

Data extraction involved collating essential details such as authors, publication year, and specific information regarding the algorithms used in each study. The analysis focused on the types of algorithms deployed for topic exctraction, the comparative metrics used to assess algorithm performance, and the methodologies applied in evaluating these models. Moreover statistical methods to analyze and draw conclusions from the data were taken into account. This approach allowed for a detailed examination of how different studies tackle topic modeling challenges within SO data, emphasizing the effectiveness and efficiency of various algorithmic strategies.

2.4 Quality Assessment

In our study, we evaluated the selected research papers using a comprehensive quality assessment framework, inspired by the standards set forth by York University's Centre for Reviews and Dissemination (CRD) Database of Abstracts of Reviews of Effects (DARE) criteria [38]. This evaluation aimed to ensure the rigor and reliability of the methods and evaluations reported in the papers related to topic modeling and empirical analysis of SO data.

QA1. Clarity and Appropriateness of Methodological Framework: Are the methods used for topic modeling and the evaluation framework in the reviewed studies clearly described and appropriate for analyzing SO data?

QA2. Exhaustiveness of the Literature Search: Does the methodology section indicate a comprehensive search and selection process for techniques and evaluations relevant to topic modeling and trend analysis?

QA3. Quality and Validity of the Methodological Approach: Have the authors provided a thorough assessment of the methodological quality and validity of the topic modeling techniques used in their studies?

QA4. Detailed Presentation of Methodological Execution and Evaluation: Are the methodological approaches and evaluation metrics used in the studies adequately described and justified?

Scoring Guidelines:

QA1: Y (Yes) if the methods are explicitly described and justified; P (Partly) if the description or justification is implicit; N (No) if the methods are not clearly defined. QA2: Y for a comprehensive review of various topic modeling methods, including searches in multiple databases and additional strategies; P for moderate search efforts; N for limited or narrow search strategies. QA3: Y for explicit and thorough quality and validity assessment of the methods; P for partial assessment; N for lack of explicit methodological quality assessment. QA4: Y for detailed descriptions of methodological execution and evaluation metrics; P for summary-level descriptions; N for insufficient detail on methodologies and evaluations. The scoring was applied as Y = 1, P = 0.5, N = 0.

2.5 Papers key information

In this subsection, first, we focus on the key aspects of the related papers that are most relevant to the pipeline we are creating and we also break down the contributions of each paper in depth for the readers interested in more detailed summaries of each individual paper. Specifically, we concentrate on the datasets used, preprocessing techniques, empirical analysis methods, and the topic modeling approaches employed. These steps guided us in developing a well-structured approach, drawing on the papers most closely aligned with our research topic.

2.5.1 Datasets. In terms of datasets, most papers relied on large-scale datasets from SO or other Q&A platforms, often spanning multiple years. Papers like [54], [12], and [48] used datasets from 2008 to 2010 or later, focusing on discussions specific to programming languages and developer trends. In contrast, more recent studies like [49] and [45] gathered data from SO and other platforms post-ChatGPT launch to assess the effects of AI-generated content. Papers such as [52] and [14] used datasets spanning longer timeframes, covering over a decade of user-generated content on SO and other platforms, while studies like [68] and [26] focused on specific topics or time periods related to software development and machine learning. Overall, the datasets ranged from several hundred thousand posts to millions, with papers like [14] analyzing over 58 million posts from multiple platforms. Some of these studies, however, could benefit from more nuanced datasets, as the broad focus on general discussions might overlook specific, short-term events that influence user behavior, such as major technology shifts.

2.5.2 Preprocessing. For preprocessing, many papers followed similar steps to clean and prepare the data. Papers such as [54], [12], [52], [49], and [48] all removed unnecessary content such as code snippets, HTML tags, and stop words to ensure cleaner input for topic modeling. Most of these papers also applied the Porter stemming algorithm to reduce words to their base forms, improving the consistency of the results. Similarly, [45], [66], and [14] filtered out low-quality posts and removed content that did not align with their focus, such as lengthy or irrelevant entries. [47] applied tokenization and removing posts that exceeded a certain length or token limit. Other studies like [68] and [34] followed typical steps of tokenization and lemmatization, ensuring that only meaningful text remained for analysis. A few papers, such as [27] and [28], applied stratified sampling to select representative data from SO and other forums, ensuring the dataset was balanced before proceeding to topic extraction.

2.5.3 Empirical analysis. For empirical analysis, a range of methods was used across the studies. Papers like [45] and [66] utilized difference-in-differences (DID) analysis to measure the impact of ChatGPT on SO's activity, particularly focusing on question frequency, length, and quality. Similarly, [14] tracked changes in human-generated content and analyzed patterns in posting activity by different programming languages. Several papers, such as [54], [12], and [48], focused on temporal trends, using statistical methods to track the rise and fall of certain topics over time. In contrast, [52] introduced unique metrics such as Accumulated Post Score (AMS) to gauge the attractiveness and difficulty of different topics. Papers like [27] and [47] compared AI-generated answers with human-generated ones, focusing on readability, comprehensiveness, and linguistic quality, using sentiment and similarity analysis to understand user preferences. Additionally, [68] applied post-classification methods to label challenge and solution topics in machine learning asset management, while [49] used regression and visualization techniques to analyze trends in AI-related discussions. While these methods provide valuable insights, some studies rely heavily on surface-level metrics like word count or voting scores, which might not fully capture deeper engagement or the true quality of contributions. More qualitative analyses could help contextualize these results.

2.5.4 Topic Modeling. When it comes to topic modeling, LDA was the most common method used across many studies, such as [54], [12], [52], [28], [39], and [48]. These papers used LDA to extract topics and analyze trends, often optimizing the number of topics based on coherence scores or manually validating the results. Papers like [49] and [26] compared LDA with newer methods such as BERTopic, concluding that BERTopic provided superior coherence and topic diversity, particularly in the context of LLMs. Papers such as [68] and [34] also relied on BERTopic for extracting nuanced topics in software development discussions, with additional methods like clustering and visualization techniques (e.g., t-SNE) to better understand topic relationships. More recent papers, such as [63], introduced advanced language models like ChatGPT and LLaMA for topic modeling, presenting an approach (PromptTopic) that integrated LLMs to generate topics and evaluate their diversity and coherence. This approach outperformed traditional LDA models in terms of capturing more complex semantic structures in the data.However, LDA, while widely used, has inherent limitations in capturing the deeper semantic meaning of texts, particularly in technical discussions where terminology and context are crucial. This issue is compounded by the need for manual intervention to merge similar topics, which introduces subjectivity and reduces consistency. More recent methods like BERTopic, though more effective, still depend on the quality of embeddings and clustering algorithms, which can sometimes overfit to specific contexts or miss subtle variations in topic relationships. Furthermore, few studies have thoroughly compared LDA with newer models in terms of long-term topic stability or cross-topic coherence.

2.5.5 Summaries of each individual paper. Study [54] examines the main topics and trends related to the Python programming language on SO by mining 2,461,876 posts from August 2008 to January 2019. The methodology involves filtering Python-related discussions using SO tags similar to [12], pruning low-quality posts to improve the quality of topic modeling by removing questions with negative score or questions that don't have accepted answer [11], and cleaning the textual content by removing code snippets e.g. $< code >$ [56], HTML tags e.g $< ahref = "..." >, < b >$, and so forth. Stop words were also removed as they don't help creating any meaningful topics. All their tokens were stemmed using the

Porter stemming algorithm [40]. LDA was used to extract 100 topics initially, which were then manually merged into 12 clusters by two Python experts. Temporal trends of these clusters were analyzed using a non-parametric statistical method (MK test) which assesses the existence of a monotonic increasing or decreasing trend (either linear or nonlinear) for a variable over the time [32]. They've also used the concept of Impact presented by [12] to obtain the portion of a topic in a time interval to track its prevalence. The results indicate that while standard Python features, web programming, and scientific programming are the most frequently discussed topics, web programming and Python standard features are declining in relative popularity, whereas scientific programming is increasing.

Second chosen study [45] investigates the impact of generative AI, specifically ChatGPT, on users' voluntary knowledge contributions on SO. The study employs a natural experiment and utilizes a difference-in-differences (DID) estimation approach to measure the effects on both the quantity and quality of users' contributions. The research model hypothesizes two competing directions for the impact of generative AI on users' answer generation: it could either reduce the number of answers due to increased cognitive load or enhance it by enabling faster, high-quality responses. The study also examines the heterogeneous effects from user and question perspectives, proposing that users with longer tenure or questions with higher upvote ratios might experience different impacts. Using data from SO from September 1, 2022, to December 4, 2022, the study identifies treated users as those generating answers similar to ChatGPT using the GPT-2 Output Detector. This method has been demonstrated to be effective in achieving high accuracy, up to 99.3%. To address potential misidentifications in the treatment identification process, the study utilizes the ChatGPT API to generate answers for the same posts that users had responded to in the dataset. The similarity between these generated answers and the existing answers is assessed using the Jaccard similarity approach [64]. Answers with a similarity score exceeding 0.9 are classified as being generated using the ChatGPT tool.The dataset includes 3,238,381 questions and 1,254,841 answers from 223,696 users. The empirical results show that generative AI tools lead to a 16.77% increase in the number of answers per day, a decrease in answer length by 22.64%, and more readable answers, while the quality, as measured by scores, remains unaffected.

Third study [66] focuses on the impact of LLMs on SO. Similar to [45] it utilizes a DID analysis and identifies a 2.64% reduction in question-asking post-ChatGPT launch, indicating a substitution effect due to the lowered search cost enabled by ChatGPT. The research employs a dataset from SO, covering two months before and after the ChatGPT launch, and a control dataset from the same period a year prior. The study employs several question-level characteristics to measure objective quality, including Length (word count) and Tags (number of associated tags). Two NLP metrics, SMOG (readability) and Cognition (cognitive effort), are used to assess text complexity and cognitive demand. An external metric, Score (upvotes minus downvotes), reflects the community's subjective assessment of question quality. Despite the identified changes, the overall quality of questions, as measured by user scores, does not significantly improve, suggesting that while ChatGPT influences the type and complexity of questions, it does not necessarily enhance the quality as perceived by the community. The study also finds that while questions became 2.7% longer and hence more sophisticated, they also became less readable by 2.55% and involved less cognitive effort by 0.4%. The research underscores the need for platforms to adapt to the evolving landscape of AI-assisted content generation to maintain engagement and quality in user-generated knowledge-sharing communities.

Next, [12] utilizes a SO dataset spanning 27 months, from July 2008 to September 2010, to analyze developer discussions and trends. The data preprocessing involves four steps similar to the paper [54], such as discarding code snippets enclosed in $< code >$ HTML tags to avoid noise from programming language syntax, removing all HTML tags, eliminating common English stop words such as "a", "the", and "is", and applying the Porter stemming algorithm to map words to their base forms. The clean data is then used using the LDA model, which identifies 40 topics of medium granularity. This amount of topics captured broad trends while maintaining topic distinctiveness. LDA operates on both uni-grams and bi-grams, as bi-grams have been shown to improve text analysis quality [55]. A threshold *gamma* of 0.10 is defined to filter out noisy topic memberships, ensuring only the dominant topics are considered in each document. To quantify and analyze the data, the study introduces several key metrics such as Topic Share, Topic Relationships, Topic Trends Over Time and Technology Trends Over Time. This first metric measures the proportion of posts that contain a particular topic. By calculating the share, the study can understand the relative popularity of each topic across the entire dataset. Topic Relationships determines the relationship between topics found in questions and their corresponding answers. By analyzing how topics in questions lead to topics in answers, the study can identify closely coupled topics and cross-cutting areas of concern for developers. Topic Trends Over Time analyzes the temporal trends of topics to measure their impact over time similar, which was used by the authors of [54], but initially created by this paper. Lastly, Technology Trends Over Time is used to compare and contrast the trends of related technologies, such as Android vs. iPhone or C# vs. Java. The topic modeling methodology creates 40 topics, which are too broad for detailed analysis of specific technologies. Therefore, the study proposes an analysis technique that combines topic modeling with user-created tags. A technology is defined as a cluster of tags related to a specific technical concept that falls under a given topic. For instance, the iPhone technology cluster includes tags like “iphone sdk-3.2” and “iphone-3gs”. This approach removes noise caused by inappropriately tagged posts and provides a more accurate account of trends by considering the proportion of a post related to a topic rather than the entire post. By identifying all tags related to a given topic and selecting those that are most popular and relevant, the study can measure the monthly impact of each technology. The results of these metrics reveal key insights, such as the increasing popularity of web development (especially jQuery), mobile applications (especially Android), Git, and MySQL, while discussions on platforms like .NET show a declining trend.

The fifth paper selected [52], provides an analysis of questions and answers on the Stack Exchange website. The data for analysis was obtained using the Stack Exchange Data Explorer, focusing on questions with a score of 1 or more and an accepted answer. Preprocessing the data focused on the same steps as papers [54], [12].Mallet LDA was used for topic modeling, setting the number of topics to 50 based on the coherence score. The most frequently asked questions are related to database systems, quality assurance, and agile software development. Compared to other papers, the attractiveness of topics was measured using the AMS, which accounts for upvotes, downvotes, comments, answers, and favorites. The most attractive topics were jobs and career, teamwork problems, and code readability. In contrast, network programming, software modeling, and access control were the least attractive topics. The study also analyzed the historical development of topic popularity from 2010 to 2020. The most rising trends were domain-driven design, asynchronous programming, and inheritance. Conversely, jobs and career, education and research, and software licensing saw significant declines in relative frequency. The number of unanswered questions remained constant, leading to a significant increase in their relative proportion from 32% to 56% between 2011 and 2020.The relationship between the sentiment of answers (measured in terms of subjectivity and polarity) and the reputation of their authors was also analyzed. Sentiment analysis was performed using the TextBlob library. The results showed no significant correlation between user reputation and the average subjectivity or polarity of their answers.

Next paper selected, [27] investigates the potential impact of ChatGPT on the traditional programmer help-seeking behavior exemplified by SO. The study addresses the correctness, consistency, comprehensiveness, and conciseness of ChatGPT answers compared to those provided by human experts on SO. The authors conducted a comprehensive analysis involving 517 SO questions and compared the ChatGPT-generated answers with the accepted human answers. The findings reveal that 52% of ChatGPT answers contained misinformation, 77% were verbose, and 78% exhibited inconsistencies with human answers. Despite these issues, user study participants preferred ChatGPT answers 35% of the time due to their comprehensiveness and well-articulated language style. However, users overlooked the misinformation in ChatGPT answers 39% of the time, underscoring the need for awareness and strategies to counteract misinformation in AI-generated responses. A mixed-methods research design was employed, including manual analysis, large-scale linguistic analysis, sentiment analysis, and user studies. The data collection involved stratified sampling of SO questions based on their popularity, recency, and type. ChatGPT answers were generated using the free version of the model and were stored for analysis. Manual analysis involved open coding to assess the correctness and quality of ChatGPT answers. Linguistic analysis utilized the Linguistic Inquiry and Word Count (LIWC) tool to evaluate the linguistic features, while sentiment analysis employed a RoBERTa-based model.

Paper [28] examines developers' discussions on Q&A forums including SO to understand software development approaches, their trends, and the challenges practitioners face. For answering their research questions, the authors used a mixed-method approach, including topic modeling and qualitative analysis. Their first research question explores the effectiveness of responses to questions about software development approaches. The authors computed the average number of answers for questions related to software development approaches and compared it to the respective value from previous research. They classified questions into three categories: successful (received an accepted answer), ordinary (received answers but no accepted answer), unsuccessful (received no answers) and analyzed the distribution of these categories and the growth trends. They found that 52.50% of questions were successfully answered, 41.24% were ordinary, and 6.26% were unsuccessful. The number of questions in this domain has shown sustainable growth, but the success rate has declined since 2014, indicating a need for more expert input on Q&A platforms. For identifying key discussion topics the authors used LDA for topic modeling. They treated each question's title, body, and corresponding answers as a single input document. They set the number of topics to 15, based on coherence scores and manual validation, and labeled each topic by inspecting the top keywords and related posts. Another interesting finding was in identifying popular and difficult topics. Popularity was gauged using views, scores, favorite count, and comments. Difficulty was assessed using the percentage of accepted answers, median duration to receive an accepted answer, and average percentage of answers to views. The geometric mean of these metrics provided a normalized value for comparison. Their last research question used AMS, same as paper [52] to rank developer posts and selected the top 200 posts for qualitative analysis. They used thematic analysis to identify challenges, categorizing them into sub-themes and higher-order themes. The study identified 49 challenges categorized into four high-level themes: project management, team management, optimization, and concepts and definitions. Project management challenges were the most significant.

Next, [49] analyze the discourse and trends in LLM research within the developer community on SO. It investigates the significance of specific tags, keywords, and themes to understand how developers discuss and perceive LLM technologies. The data were filtered to include only entries from 2017 onwards, considering the introduction of the Transformer architecture as a pivotal development in that year. Pre-processing involved similar steps with [54], [12], [52]. Linear regression and word cloud analysis identified the most frequently used tags and their growth trajectories. Tags like "huggingface-transformers," "openai-api," and "python" showed significant increases in usage. TF-IDF analysis identified the significance of individual terms within the dataset. Terms like "use," "model," "transformer," "bert," "python," and "data" were among the top 20 terms, indicating their centrality in LLM discussions. Heatmap analysis further explored the semantic interactions between these terms. The cosine similarity measure was used to determine the semantic connections, highlighting relationships like those between "huggingface," "transformer," and "bert." The optimal number of topics was determined to be 5 based on coherence scores and manual inspection. Network analysis examined the interrelationships of the keywords derived from the LDA model. Keywords were represented as nodes, and their co-occurrences were represented as edges. Limitations written by the authors state that the study's cut-off date in early June 2023, meaning that it excludes the most recent three months of discussions, a period marked by significant developments in open-source LLMs. The study also suggests that future research could benefit from using advanced tools like the transformer-based BERTopic, which can understand the semantic context of words and phrases better than traditional methods like TF-IDF and LDA. Additionally, analyzing extracted topics using models like GPT or LLaMA could provide deeper insights into the discourse surrounding LLMs.

Paper [26] presents an in-depth exploration of topic modeling methods applied to data from Web of Science and LexisNexis, covering from June 1, 2020, to December 31, 2023. Data was gathered using specific queries such as "Large language model," "LLM," and "ChatGPT." The collection included 10,563 news articles from LexisNexis and 11,070 academic papers from Web of Science. Preprocessing involved eliminating duplicates, removing data exceeding certain length thresholds, and using the spaCy library for lemmatization and stop words removal. This resulted in 3,917 texts from LexisNexis and 3,438 from Web of Science being used for the experiments. The authors evaluated four topic modeling methods: LDA, Nonnegative Matrix Factorization (NMF), Combined Topic Model (CTM), and BERTopic. Two metrics were used for evaluation Topic Diversity and Topic Coherence, coherence being used also in the other papers [52], [28], [49]. BERTopic demonstrated superior performance and was used for detailed topic analysis.The hyperparameters were fined tuned having UMAP, the number of nearest neighbors and components to 5, with a minimum distance of 0.0, and adopting cosine similarity as the score. For HDBSCAN, the minimum cluster size is set to 5, with all other parameters left at their default values. To ensure reproducibility they seeded all of their experiments with random seed of 42. The analysis included extracting the top 8 words for each topic and labeling them based on the most important keywords and the original data. The study employed various visualization techniques to present the findings such as heatmaps, t-SNE plots, used to reduce dimensionality and visualize the distribution of topics in a two-dimensional space and network analysis as [49].

Next, [47], investigates the capabilities of LLMs like ChatGPT 3.5 and LLaMA-2 in generating high-quality answers compared to human-generated answers on SO. The dataset spans from before and after the release of ChatGPT, comprising 205,777 questions before and 145,528 questions after. To comply with the 2048 token limitation for LLMs, questions exceeding this count were excluded. After filtering, 384 questions were randomly selected from each set for a 95% confidence level with a 5% margin of error. Preprocessing included calculating token requirements based on the question title, description, and associated tags. To ensure that the LLMs could generate contextually relevant and high-quality answers, the authors followed a structured and standardized prompting method. An example of a promt looks like this: "You are an expert in [list of tags]. Here is a question that needs your expertise: [Question Title]. Can you provide a detailed explanation on how to fix the problem described below?[Question Description]". Their study evaluated textual and semantic similarities between LLM-generated answers and human answers using cosine similarity and manual analysis. Cosine similarity was calculated using a pre-trained Sentence Transformers model (all-MiniLM-L6-v2). Both sets of answers were embedded into PyTorch tensors, and the similarity was computed. For semantic similarity the LLMs were prompted to compare the original SO answer and the LLM-generated answer, with the expected output on a scale from VERY LOW to VERY HIGH. The cosine similarity metric revealed that many LLM-generated answers had moderate to high similarity with human answers, but there were notable instances of low similarity due to structural and stylistic differences. LLMs struggled with maintaining semantic coherence in some answers, highlighting the challenge of ensuring both textual and semantic quality in generated content. Authors stated that future research could incorporate data from multiple platforms and explore advanced topic modeling techniques like BERTopic for richer insights similar to [49].

Paper [14] examines the impact of LLMs, specifically ChatGPT, on human-generated open data on several Q&A platforms, including SO, its Russian-language version, Mathematics Stack Exchange, Math Overflow, and the Chinese-language platform Segmentfault. The data spans from January 2019 to June 2023, including over 58 million posts from SO and additional posts from the other platforms. Preprocessing involved extracting posts and their metadata, such as votes, user information, and tags. Their findings suggest that activity on SO decreased by about 16% following the release of ChatGPT. There was no significant change in the voting patterns, indicating that ChatGPT is displacing a variety of posts, not just low-quality or duplicate content. Posting activity decreased more for widely used languages like Python and JavaScript compared to niche languages. The decline in activity was more pronounced for languages with a larger user base on GitHub. The discussion section of the paper highlights several key impacts of ChatGPT on digital public goods. The decrease in human-generated content on SO may limit the availability of open data for training future LLMs, potentially hindering the development of new models and reducing the overall quality of digital public goods. ChatGPT's ability to crowd out open data creation while capturing valuable user interactions gives OpenAI a competitive advantage, potentially leading to a more closed AI ecosystem. This shift from public knowledge sharing to private LLM interactions may affect the democratization of knowledge, increasing inequalities in access to information and technological tools. Additionally, the efficiency of LLMs like ChatGPT may narrow users' exposure to diverse sources of information, reinforcing mainstream perspectives and reducing the incentive to explore new or niche topics.

Next study [68] examines 15,065 Q&A posts from various developer forums to identify operational challenges and solutions, using a mixed-method approach and BERTopic for topic extraction. The study identifies 133 distinct challenge topics, grouped into 16 macro-topics, and 79 solution topics, classified under 18 macro-topics. The study identifies several key findings. In the realm of challenges in ML asset management, the most discussed macro-topic is software environment and dependency, which accounts for 18.89% of the posts, highlighting issues with managing software environments and dependencies. Another prevalent topic is model deployment and service, representing 10.59% of the discussions, focusing on the challenges of deploying and serving models. Additionally, there is significant interest and concern shown in model creation and training, covering 9% of the posts.For solutions in ML asset management, most proposed solutions (23.31%) address software environment and dependency issues. Feature and component development are commonly proposed solutions (15.35%) for source code management challenges. File and directory management solutions address various issues, accounting for 9.64% of the proposed solutions. In terms of discussion forum analysis, SO is the primary forum for asset management inquiries, accounting for 48.82% of the posts, followed by tool-specific forums at 34.19%, and repository-specific forums at 17.16%. Software environment and dependency issues are the most prevalent in all forums. BERTopic was fine-tuned to optimize performance. The hyperparameters used include a minimum cluster size of 30 for challenge modeling and 20 for solution modeling, a range of 1-100 for min samples in challenge modeling and 1-40 for solution modeling, 3-10 n components for challenge modeling and 3-8 for solution modeling, an ngram range of 1-3 for both, the embedding model "all-mpnet-base-v2", cosine similarity for metrics, and a random state of 42 for reproducibility. Key observations include the labeling of posts, which involves using 2-5 words starting with a verb and ending with a noun, with adjectives added selectively for clarity. This approach ensures consistency and informative context, avoiding preprocessing that could result in the loss of important data.

Study [34] study aims to address the complexities of interpreting extensive and often cumbersome API documentation by generating concise and meaningful summaries from informal sources like SO. Using the Stack Exchange API, the study retrieved all questions tagged with Android on SO from January 2009 to April 2022, along with their corresponding answers, resulting in a dataset of 3,698,168 unique posts. They've used BERTopic to identify discussed topics. A pre-trained model from Hugging Face, trained on over a million Wikipedia pages, was utilized. Computations were performed on Google Colab Pro with a T4 GPU, using cuML for GPU-accelerated machine learning. The algorithm identified 1,813 distinct topics, with 75% of the data concentrated in the top 80 categories. The research focus was narrowed to these top 80 topics, and the most prevalent topics were presented in a table, detailing the count of posts, topic names, and representative words. The most prevalent topics included project errors related to build gradle, fragment viewpager view issues, and notification activity service problems. A two-dimensional distance map depicted the relational layout and intertopic distances of these topics.

Next, [39] investigates the challenges developers face with open source software (OSS) licensing by analyzing questions from four Stack Exchange sites. After filtering and preprocessing the data like in previous studies, the final data set consisted of 6,697 questions and 11,596 answers. The study found that the licenses mentioned most frequently were GPL, MIT, and Apache. Analysis revealed an increasing trend in the number of different licenses mentioned over time, with a noticeable shift towards more permissive licenses like MIT. Using LDA for topic modeling, 16 main topics related to OSS licensing were identified. These topics were grouped into seven broader categories: Specific licenses (such as MIT, GPL, and Creative Commons), License conditions, Commercial vs. OSS, Modifications and warranties, Linking (static and dynamic), Repositories, and General OSS. Topic modeling determined that the optimal number of topics based on coherence scores and manual verification is of 16 topics. Popularity and difficulty were assessed using metrics such as views, favorites, scores, and the time taken to receive an accepted answer, same as paper [28]. Questions about commercial software and selling software garnered the most views, indicating high interest. Various graphs and visualizations were used to present the findings. One graph visualized the distribution of top licenses mentioned across different Stack Exchange sites, while another showed the percentage of licenses mentioned over time, highlighting trends such as the increasing prominence of permissive licenses like MIT.

The next to last one, paper [48] discovers the themes within the questions and answers, aiming to prevent the overflow of insignificant questions and unnecessary tags using LDA. The dataset comprises posts from SO between July 31, 2008, and March 27, 2009. The data extraction process resulted in 513,136 documents, including 111,871 questions and 401,265 answers. Each post contains metadata such as the title, body, creation date, post type, view counts, answer counts, and comment counts. The implementation used the Stanford Topic Modeling Toolbox (TMT), with the number of topics (K) set to 10 for medium granularity. Similar to paper [12], their study uses Topic Share, Topic Relationships and Topic Trends Over Time. The study categorized questions into three quality levels: Good Quality, Medium Quality, and Low Quality. Good Quality questions had accepted answers and scores greater than 7, Medium Quality questions had accepted answers and scores between 1 and 6, and Low Quality questions had no accepted answers and scores less than 0. Graphs illustrated the share and impact of each topic, showing the relative popularity and trends over time. For instance, the study observed that interest in certain topics like web development and database queries declined over time, while topics related to security and session management gained prominence. The quality analysis revealed that questions with higher scores and accepted answers tended to be more detailed and specific, while low-quality questions often lacked clarity or relevance. The study also developed a method to suggest tags for new questions based on the discovered topics, aiming to reduce the creation of unnecessary tags and improve the site's organization.

Lastly, paper [63] introduces PromptTopic, a novel approach to topic modeling that leverages the advanced language understanding capabilities of LLMs. ChatGPT and LLaMA are used which integrate word and sentence semantics for a more holistic topic modeling experience. The methodology of PromptTopic consists of three stages: Topic Generation, Topic Collapse, and Topic Representation Generation. The prompt setup for topic generation uses ChatGPT with N demonstration examples, each comprising prompt inputs and associated annotated answers. For LLaMA, instructional statements are omitted from the prompt due to its non-instruction-trained nature. The optimal value for N was found to be 4, producing the best topic generation performance for LLaMA, while ChatGPT was less sensitive to changes in N. Two approaches are introduced to collapse overlapping topics: Prompt-Based Matching (PBM) and Word Similarity Matching (WSM). PBM involves sorting unique topics based on frequency counts and merging them iteratively until only K topics remain. WSM computes topic similarity using Class-based Term Frequency-Inverse Document Frequency (c-TF-IDF) and merges highly similar topics until the desired number of topics, K, is achieved. To evaluate the performance of PromptTopic, well-established topic model metrics are used. c-TF-IDF scores are employed to compute the most representative words for each topic, initially obtaining the top 100 words and refining them down to the top 10 using LLMs. The quantitative evaluation used two well-established metrics: topic coherence and topic diversity. Topic coherence, measured using Normalized Pointwise Mutual Information (NPMI), assesses the relatedness of words within a topic. Topic diversity evaluates the proportion of unique words across all topics. The number of topics (K) was empirically selected for each dataset: 40 for 20 NewsGroup, and 20 each for Yelp Reviews and Twitter Tweet. PromptTopic-WSM consistently outperformed most baseline models in both metrics. LLaMA-13b, as a standalone offline model, showed comparable quality to ChatGPT while generating more diverse topics.

3 METHODOLOGY

3.1 Data Collection

To collect questions and answers from SO, a Python script was developed to fetch data incrementally using the Stack Exchange API [9]. This approach ensured compliance with Stack Overlfow API rate limits and efficient handling of the large datasets gathered. The retrieved data was saved in JSON format per day. At a later stage, the answers were merged with their specific questions based on the `Question ID`, ending up with two JSON files, one per year, having all of the questions merged with their answers.

3.2 API Throttling Information

According to the Stack Exchange API [9] documentation, several throttles are implemented to prevent abuse. Every application is subject to an IP-based concurrent request throttle. If a single IP makes more than 30 requests per second, new requests will be dropped. The ban period typically ranges from 30 seconds to a few minutes. The exact response when subject to this ban is undefined, as making more than 30 requests per second per IP is considered highly abusive. Using the access token for my application, the default size was of 10,000 requests per day.

Additionally, the API has a dynamic throttle that can temporarily limit requests to prevent overuse. If an application gets a response with a `backoff` field, it must wait the given number of seconds before making another request to the same method. This applies to similar routes, such as `/me` and `/users/{ids}`. The `backoff` field might not always appear for the same request, and any method, no matter how simple, can trigger this response.

3.3 Problems and solutions during data collection

During the data collection process, we encountered challenges when attempting to gather questions and their corresponding answers simultaneously from the Stack Exchange API[9]. The primary issue was the immediate termination of requests. This problem was likely due to the increased number of API requests generated by this approach, which could easily surpass the API's rate limits. Fetching a question and then immediately requesting its answers was doubling or even tripling the number of requests, because the page in a request to the Stack Exchange API could have contained up to 100 answers but not so many answers were per question so most of the request was wasted, also quickly hitting the concurrent request throttle of 30 requests per second. Additionally, this method increased the complexity of request handling and error management, making it more difficult to effectively manage API rate limits and backoff requirements. Due to these problems questions and answers were fetched separately and then merged together.

3.4 Retrieving questions/answers algorithm

The proposed script 1 respects the 30 requests per second limit by controlling the request rate. This is achieved by using a combination of retries and backoff handling. If the response indicates a throttle violation, the script pauses for the specified backoff period before retrying. The script also respects the daily request quota of 10,000 requests per user/application pair by tracking the number of requests made each day. If the daily limit is reached, the script pauses for 24 hours before continuing.

The data retrieval process involved several key steps. First, essential libraries were imported to handle HTTP requests, time conversions, data manipulation, JSON operations, and file system interactions. A function was implemented to convert date strings in the format `YYYY-MM-DD` to Unix timestamps. This conversion was necessary for specifying the time range in API requests.

The core function, designed to retrieve questions from the Stack Exchange API within a specified time range, handled pagination, retries in case of errors, and backoff periods for throttle violations. The function made API requests with the specified parameters, handled HTTP errors, SSL errors, and throttle violations by retrying the requests when necessary, and extracted the items from the API response, continuing fetching pages until there were no more results.

To manage the incremental fetching of data, another function divided the overall time range into daily batches. Each batch was processed and saved as a separate JSON file. After fetching the data, a final function read all JSON files from the output directory and concatenated them into a single pandas DataFrame for easy data manipulation for later on analysis.

Finally, the script set the query parameters (start date, end date, site, API key, output directory, and requests per day) and executed the data fetching process. This methodology ensured efficient and rate-limit-compliant data collection from the Stack Exchange API. The incremental fetching approach allowed the handling of large datasets.

3.5 Merging questions and answers together

For merging the SO questions with their corresponding answers the following script was made 2. Initially, it reads questions from specified directories in batches using the `read_json_files_in_batches` function, storing them in a dictionary with question IDs as keys. Each question entry in the dictionary also contains a list for storing answers. The script then reads answers from the answers' directory, matching each answer to its corresponding question using the `question ID`. The matched questions and answers are written to an output file in batches using the `write_to_output_file` function. The main function `process_data` coordinates these steps, ensuring efficient handling of large datasets by processing and storing data incrementally. The main execution part initializes the required directories and output paths, and calls `process_data` for the specified time periods.

Table 2. Topic modeling models used in selected papers

ID	Author	Description	LDA	BERT	Empirical Analysis
P1	Tahmooresi et al. [54]	Python trends on SO	✓		✓
P2	Shan et al. [45]	Impact of ChatGPT on SO			✓
P3	Xue et al. [66]	Impact of ChatGPT on SO			✓
P4	Barua et al. [12]	Analyzing SO topics with LDA	✓		✓
P5	Sulír et al. [52]	Analyzing Software Engineering Stack Exchange trends	✓		✓
P6	Kabir et al. [27]	Evaluating ChatGPT answers on SO			✓
P7	Arif et al. [28]	Analyzing software development approaches on Q&A forums	✓		✓
P8	Son et al. [49]	Analyzing LLM trends and developer interactions on SO	✓		✓
P9	Jung et al. [26]	Analyzing topic modeling methods for academic papers and news	✓	✓	
P10	Leuson et al. [47]	Analyzing ChatGPT's impact on SO		✓	✓
P11	Maria et al. [14]	ChatGPT reduces public data on SO			✓
P12	Zhao et al. [68]	Analyzing ML asset management challenges and solutions in SO		✓	✓
P13	Naghshzan et al. [34]	Improving API documentation with BERTopic and NLP in SO		✓	
P14	Papoutsoglou et al. [39]	Analyzing open source software licensing issues on Stack Exchange	✓		✓
P15	Singh et al. [48]	Analyzing SO using LDA for topic trends	✓		✓
P16	Han et al. [63]	PromptTopic: Improved topic modeling with LLMs		✓	

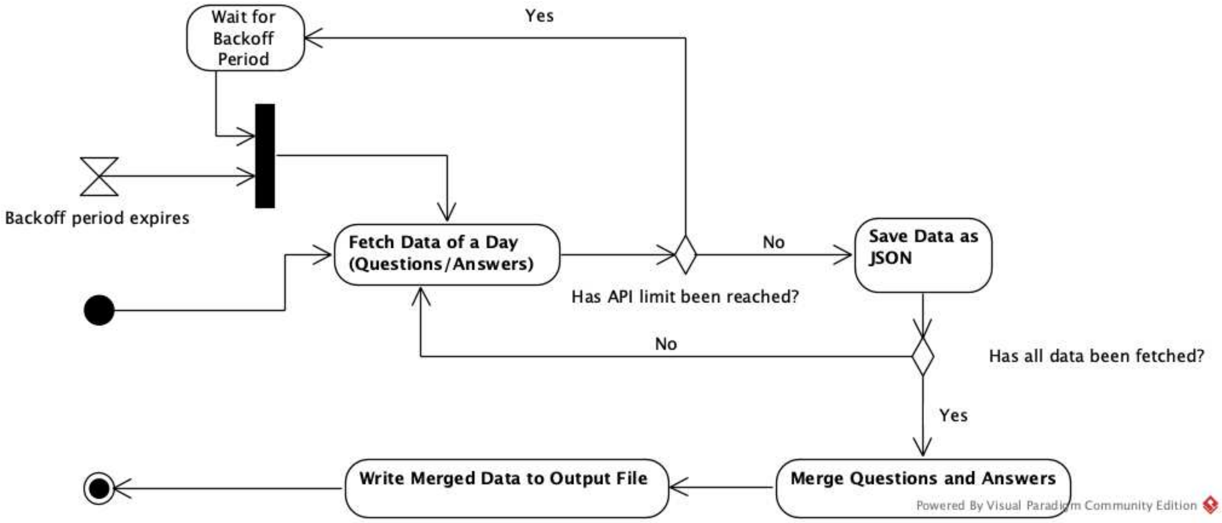

Wait for Backoff Period
Yes
Backoff period expires
Fetch Data of a Day (Questions/Answers)
No
Save Data as JSON
Has API limit been reached?
No
Has all data been fetched?
Yes
Write Merged Data to Output File
Merge Questions and Answers
Powered By Visual Paradigm Community Edition

Fig. 1. Activity Diagram Data Collection

To further clarify the process, the activity diagram in Figure 1 visually represents the steps involved in collecting data from the Stack Exchange API, handling rate limits, and merging questions with their answers. This diagram provides a high-level overview to help understand the workflow discussed in the previous subsections.

3.6 Dataset Structure and initial statistics

For this study, we analyzed datasets containing SO questions and answers from two distinct one-year periods: April 1, 2021, to April 1, 2022, and April 1, 2023, to April 1, 2024.

The dataset for the period 2021-2022, contains a total of 1,488,716 questions and 1,652,359 answers. This dataset occupies 6.59 GB of storage. In contrast, the dataset for the period April 1, 2023, to April 1, 2024, includes 839,924 questions and 644,815 answers, with a total size of 3.74 GB. The significant difference in the volume of data between these periods provides a preliminary indication of changing dynamics on the platform, potentially influenced by the adoption of generative AI tools like ChatGPT similar to the decreasing pattern of using SO discussed in the paper [27] where they compare ChatGPT's performance in answering programming questions already answered from SO.

The data structure depicted in Table 4 begins with the tags field, which is an array containing tags related to the question. Tags help categorize the question and improve its searchability. Following this, the owner field is an object that provides detailed metadata about the user who posted the question, including their reputation, user

Algorithm 1 Incremental Data Fetching from Stack Exchange

```
Require: Start date, end date, site, API key, output directory, requests per day
function TO_UNIX_TIMESTAMP(date_str)
    return Unix timestamp from date_str
function FETCH_QUESTIONS/ANSWERS(start_time, end_time, site, api_key, page_size)
    Initialize questions/answers list
    has_more ← True
    page ← 1
    while has_more do
        Set up API request parameters
        Handle HTTP and SSL errors with retries
        Fetch data from API and parse JSON response
        Append fetched items to questions/answers list
        has_more ← Check if more pages are available
        Increment page
        Check for backoff period and wait if necessary
    return questions/answers
function FETCH_DATA_INCREMENTALLY(start_date, end_date, site, api_key, output_dir, requests_per_day)
    start_time ← TO_UNIX_TIMESTAMP(start_date)
    end_time ← TO_UNIX_TIMESTAMP(end_date)
    batch_start_time ← start_time
    while True do
        daily_requests ← 0
        while daily_requests < requests_per_day do
            batch_end_time ← batch_start_time + 86400    ▷ 1 day in seconds
            questions/answers ← FETCH_QUESTIONS/ANSWERS(batch_start_time, batch_end_time, site, api_key)
            if questions/answers is None or length of questions/answers is 0 then
                break
            Save questions/answers to JSON file
            Increment daily_requests
            batch_start_time ← batch_end_time
            if daily_requests ≥ requests_per_day then
                break
        Wait for 24 hours before continuing
function LOAD_AND_CONCATENATE_JSON_FILES(output_dir)
    Initialize data_frames list
    for each file in output_dir do
        if file ends with .json then
            Read JSON file into DataFrame and append to data_frames
    return concatenated DataFrame
                                                ▷ Main Execution
Define start_date, end_date, site, api_key, output_dir, requests_per_day
Create output_dir if it does not exist
Call FETCH_DATA_INCREMENTALLY(start_date, end_date, site, api_key, output_dir, requests_per_day)
combined_df ← LOAD_AND_CONCATENATE_JSON_FILES(output_dir)
```

Table 3. Quality of studies using the DARE criteria [38]

Papers List						
Study	Article type	QA1	QA2	QA3	QA4	Total score
[54]	RP	Y	P	Y	Y	3.5
[45]	RP	Y	P	Y	Y	3.5
[66]	RP	Y	P	Y	Y	3.5
[12]	RP	Y	P	Y	Y	3.5
[52]	RP	Y	P	Y	Y	3.5
[27]	RP	Y	P	Y	Y	3.5
[28]	RP	Y	Y	Y	Y	4
[49]	RP	Y	Y	Y	Y	3.5
[26]	RP	Y	Y	Y	Y	4
[47]	RP	Y	Y	Y	Y	4
[14]	RP	Y	P	Y	Y	3.5
[68]	RP	Y	Y	Y	Y	4
[34]	RP	Y	P	Y	Y	3.5
[39]	RP	Y	Y	Y	Y	4
[48]	RP	Y	P	Y	Y	3.5
[63]	RP	Y	P	Y	Y	3.5

Algorithm 2 Merge SO Questions and Answers

```
Require: Question directories, answer directory, output file path
function READ_JSON_FILES_IN_BATCHES(directory, batch_size)
    Read JSON files from directory in batches
function WRITE_TO_OUTPUT_FILE(data_batch, output_file)
    Write data batch to output file
function PROCESS_DATA(question_dirs, answer_dir, output_path)
    Initialize dictionary for questions
    for each question directory in question_dirs do
        Read questions in batches
        Store questions in dictionary
    Read answers in batches
    Match answers to questions in dictionary
    Write matched data to output file in batches
                                                ▷ Main Execution
Define question directories, answer directory, and output file paths
Call process_data for each time period
```

ID, user type, acceptance rate, profile image URL, display name, and a link to their SO profile.

The is_answered field is a boolean value indicating whether the question has received an accepted answer. The view_count field shows the total number of views the question has accumulated, reflecting its popularity or difficulty. If the question has an accepted answer, the accepted_answer_id field will contain the ID of that answer.

The answer_count field provides the number of answers the question has received, and the score field represents the net score of the question, calculated as the difference between upvotes and downvotes. The last_activity_date and creation_date fields are timestamps indicating the last activity on the question and the date it was created, respectively. Additionally, the last_edit_date field shows when the question was last edited.

Each question is uniquely identified by the question_id, and the content_license field specifies the licensing of the question content. The link field provides a direct URL to the question on SO, while the title and body fields contain the title and detailed description of the question.

Lastly, the answers field is an array of objects, each representing an answer to the question. Each answer object includes user information, whether the answer is accepted, its score, and relevant timestamps.

Field	Description
`tags`	An array of tags associated with the question, used for categorization and searchability.
`owner`	An object containing information about the user who posted the question, including reputation, user ID, user type, acceptance rate, profile image URL, display name, and link to their SO profile.
`is_answered`	A boolean indicating whether the question has an accepted answer.
`view_count`	The number of times the question has been viewed.
`accepted_answer_id`	The ID of the accepted answer for the question, if any.
`answer_count`	The total number of answers posted for the question.
`score`	The score (upvotes minus downvotes) of the question.
`last_activity_date`	The timestamp of the last activity on the question (e.g., an edit or a new answer).
`creation_date`	The timestamp of when the question was originally posted.
`last_edit_date`	The timestamp of the last edit made to the question.
`question_id`	The unique identifier for the question.
`content_license`	The content license under which the question is published, indicating the usage rights.
`link`	The URL link to the question on SO.
`title`	The title of the question, summarizing the issue or topic.
`body`	The detailed body of the question, often including descriptions, code snippets, and images.
`answers`	An array of objects, each containing details about the answers provided to the question, including user information, acceptance status, score, timestamps, answer ID, and question ID.

Table 4. Data Structure of SO JSON Entries

3.7 Data processing

Based on the selected papers from Chapter 2, our data cleaning approach closely followed established methodologies, applied to both the body and title of questions. Initially, we converted HTML entities to their corresponding characters using the 'html.unescape' method. This step was essential to ensure the text's readability and consistency. Next, we removed URLs from the text using regular expressions to eliminate any irrelevant web addresses that could interfere with the analysis. A significant portion of the questions and answers on SO included code snippets, which, while useful for human readers, do not contribute meaningfully to topic models. As noted by previous studies [50], code content can obscure the primary textual data that these models analyze. Therefore, we removed code by identifying and stripping content within '<code>' and '<pre><code>' tags. Additionally, we removed all remaining HTML tags (e.g., <a href="...">) to ensure only the textual content was retained.

For both LDA and BERTopic, we processed the text by removing common English-language stop words such as "an", "the", and "was". These words do not help create meaningful topics and are often removed in text analysis to enhance the model's performance. In the BERTopic documentation [18] this is the only necessary preprocessing step for allowing the model to generate more accurate and relevant topics from the raw text. For LDA, we further applied the Porter stemming algorithm, as employed by [12]. Stemming reduces words to their base or root form, which aids in grouping similar words together during topic modeling.

After titles and bodies of the questions and answers are preprocessed, we append only the accepted answer to the corresponding question rather than including all answers. We also remove all questions with a negative score or without an accepted answer. This approach aligns with several studies that emphasize the importance of focusing on high-quality content for topic modeling. For instance, Study [54] filters out low-quality posts by removing questions with negative scores or those without an accepted answer, ensuring that only the most relevant and high-quality content is analyzed. Similarly, Study [52] focuses on questions with a score of 1 or more and an accepted answer, prioritizing high-quality content for analysis. In contrast, including all answers might introduce noise and reduce the clarity of the extracted topics, as not all answers may be equally informative or accurate. Therefore, appending only the chosen answer by the person who posted the question is a more effective approach to ensure the quality and accuracy of the data used in topic modeling. The new number of documents (question and its answers form a document) for the first period of time is 568677 and 212735 for the 2023-2024. These documents are going to be used for the topic modelling methods.

3.8 Topic modeling

In this paper, we apply topic modeling to extract discussion topics from SO posts. Topic modeling is an advanced technique in NLP that leverages unsupervised learning to identify and summarize key themes within large text datasets. This approach does not rely on pre-existing tags, training data, or predefined categories. Instead, it utilizes word frequencies and their co-occurrences within the documents to uncover latent topics.

The primary function of topic modeling is to group frequently co-occurring words into sets of topics, thereby revealing the underlying themes in the text corpus. This method has proven effective across various fields [25], enabling the automatic organization and analysis of vast collections of unstructured text. Unlike supervised learning methods that require labeled training datasets, topic modeling independently generates thematic annotations, making it a powerful tool for text analysis. By identifying common keywords or phrases and grouping them into topics, this technique provides insights into the primary themes characterizing a collection of documents.

To better understand the various topic modeling techniques applied in this study, the following flowchart in Figure 2 outlines the step-by-step process involved in each approach.Each method has its own specific sequence of steps, from generating embeddings to clustering topics and extracting keywords. These are going to be better explained in the following subsections. LDA method constructs a document-term matrix and applies Gibbs sampling to infer topics, while BERTopic involves UMAP for dimensionality reduction and HDBSCAN for clustering. LLaMA-3, KeyBERT, and POS tagging work on top of BERTopic, as they are additional techniques applied after BERTopic has generated the initial topics. In the case of BERTopic + LLaMA-3, the model leverages LLaMA for efficient topic summarization. KeyBERT extracts keywords using cosine similarity, while POS tagging identifies key parts of speech to filter potential keywords based on grammatical structure.

3.8.1 *Latent Dirichlet Allocation.*

LDA is a probabilistic topic modeling algorithm designed to uncover hidden thematic structures within a corpus of text. Unlike linear discriminant analysis, LDA represents topics as probability distributions over words in the corpus and documents as probability distributions over these topics. This method leverages the document-term matrix to generate topic distributions, which are lists of keywords with corresponding probabilities. The fundamental assumption is that words frequently co-occurring in documents are likely to belong to the same topic.

LDA works by finding sets of words that tend to appear together in the documents. For example, a topic with words like "gene," "sequence," "mutation," and "genome" would likely relate to genetics, while another topic with words like "market," "investment," "stock," and "finance" would relate to economics. The algorithm can also identify that a document contains multiple topics, such as both genetics and economics. This allows for a nuanced understanding of the document's content without requiring any manual tagging or pre-existing labels.

The mathematical foundation of LDA involves the use of Gibbs sampling, an iterative process for topic assignment. The Gibbs sampling algorithm repeatedly updates the topic assignments for each word in the corpus, refining the probability distributions over multiple iterations. The principal components of Gibbs sampling include two main ratios such as the probability of topic t in document d, calculated based on the number of words in d that belong to t and the probability of word w belonging to topic t, determined by the occurrences of w in t across the corpus.

The probabilities are represented as:

$$P(z_i = t \mid z_{-i}, w) = \frac{n_{m,t}^{-i} + \alpha}{\sum_{t'=1}^{T} (n_{m,t'}^{-i} + \alpha)} \cdot \frac{n_{t,w_i}^{-i} + \beta}{\sum_{v'=1}^{V} (n_{t,v'}^{-i} + \beta)}$$

where $n_{m,t}^{-i}$ is the number of words in document d assigned to topic t, α is the Dirichlet prior for the document-topic distributions, n_{t,w_i}^{-i} is the number of times word w is assigned to topic t in the entire corpus, β is the Dirichlet prior for the topic-word distributions, T is the number of topics, and V is the vocabulary size.

LDA implementation in our study uses the MALLET (MAchine Learning for LanguagE Toolkit) which applies the Gibbs sampling algorithm, similar to the implementation of [12]. Mallet uses Gibbs Sampling which is more precise than Gensim's faster and online Variational Bayes [3].

The number of topics, denoted as K, is a critical parameter influencing the granularity of the discovered topics. Larger values of K yield more detailed topics, while smaller values produce broader, more general topics. Optimal K values vary by dataset and research goals. For instance, studies have used different K values: 100 topics merged into 12 clusters [54], 40 topics for medium granularity [12], 50 topics based on coherence scores [52], 15 topics validated through coherence scores and manual inspection [28], 5 based on coherence scores and manual inspection [49]. We will run our experiments using the values 5,15,30,40,50,75 and 100 for K.

LDA can analyze text using uni-grams (single words) or n-grams (sequences of adjacent words). Bi-grams, in particular, enhance text analysis quality by capturing more context. For example, in the context of medical records, the text "heart attack symptoms" can be split into uni-grams ("heart," "attack," "symptoms") and bi-grams ("heart_attack," "attack_symptoms"). We will use both uni-grams and bi-grams. Bi-grams shown to increase the performance for topic modeling in [37]. The output of LDA includes a set of topics, each defined as a distribution over words, and topic membership vectors for each document, indicating the proportion of words in the document from each topic. The top words in a topic provide insights into its nature.

3.8.2 *BERTopic.*

BERTopic is an advanced topic modeling technique that utilizes transformer-based embeddings and sophisticated clustering algorithms to identify latent topics within text corpora. Unlike traditional topic models such as LDA, which rely on word co-occurrence patterns and probabilistic distributions, BERTopic

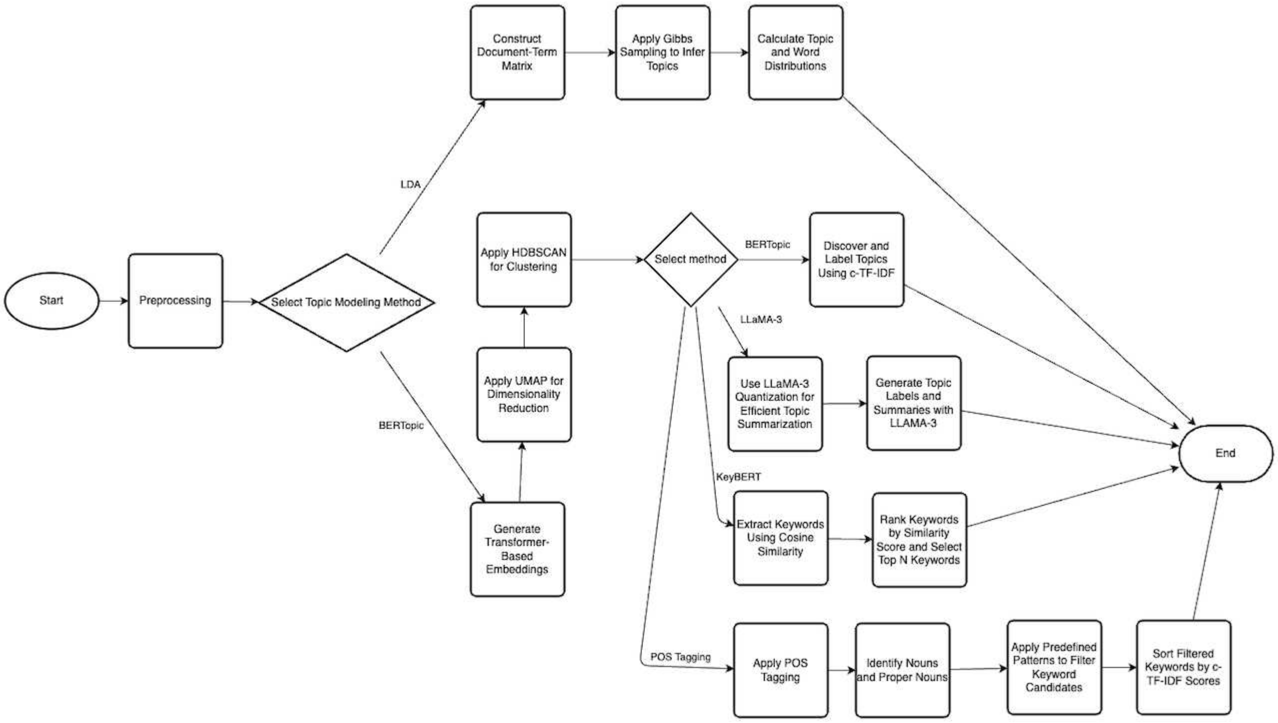

Start
Preprocessing
Select Topic Modeling Method
LDA
Construct Document-Term Matrix
Apply Gibbs Sampling to Infer Topics
Calculate Topic and Word Distributions
BERTopic
Generate Transformer-Based Embeddings
Apply UMAP for Dimensionality Reduction
Apply HDBSCAN for Clustering
Select method
BERTopic
Discover and Label Topics Using c-TF-IDF
LLaMA-3
Use LLaMA-3 Quantization for Efficient Topic Summarization
Generate Topic Labels and Summaries with LLAMA-3
KeyBERT
Extract Keywords Using Cosine Similarity
Rank Keywords by Similarity Score and Select Top N Keywords
POS Tagging
Apply POS Tagging
Identify Nouns and Proper Nouns
Apply Predefined Patterns to Filter Keyword Candidates
Sort Filtered Keywords by c-TF-IDF Scores
End

Fig. 2. Topic modeling flow diagram

leverages pre-trained language models to capture deeper semantic relationships between words and documents. This allows in theory, for the generation of more coherent and contextually meaningful topics. For instance, [11, 12, 54] utilized BERTopic to filter discussions, prune low-quality posts, and clean textual content, resulting in more precise and meaningful topics. Another study highlighted the advantages of using BERTopic for analyzing large datasets, finding that it produces more readable and coherent topics [45]. The effectiveness of BERTopic in capturing trends and relationships within text data has been validated across various research works, underscoring its capability to enhance topic modeling tasks [26, 49, 66] as reviewed in 2.

BERTopic integrates several advanced machine learning techniques to effectively discover topics within a corpus of documents. The key components of BERTopic include transformer-based embeddings, UMAP for dimensionality reduction, and HDBSCAN for clustering [16]. Initially, the Sentence Transformer model "BAAI/bge-small-en" is employed to encode the documents into dense vector representations. This embedding process is accelerated using GPU support, ensuring efficient handling of our large datasets. We conducted our experiments using Google Colab, leveraging an A100 GPU with 83.5 GB of high RAM and 40 GB of RAM memory. This setup provided the computational power necessary to efficiently process and analyze large text datasets using BERTopic and BERTopic quantized with LLAMA-3-8B LLM. UMAP (Uniform Manifold Approximation and Projection) [18] is employed to reduce the high-dimensional embeddings into a lower-dimensional space. This step is crucial for visualizing and clustering the data while preserving both global and local structures. Key parameters for UMAP include n_components, which determines the number of dimensions to reduce the data to; n_neighbors, which specifies the number of neighboring points used in the manifold approximation; and min_dist, which sets the minimum distance between points in the low-dimensional space. HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) [18] is a robust clustering algorithm capable of identifying clusters of varying densities and effectively handling noise in the data. Key parameters for HDBSCAN include min_samples, which defines the minimum number of samples in a cluster, and gen_min_span_tree, which indicates whether to generate the minimum spanning tree for cluster hierarchy. For example in paper [26] the UMAP parameters were fine-tuned with the number of nearest neighbors and components set to 5, and a minimum distance of 0.0, while HDBSCAN used a minimum cluster size of 5 with all other parameters left at their default values. To ensure reproducibility they seeded all of their experiments with random seed of 42. For our experiments we have used a minimum cluster size of 200 for the 2023-2024 as the dataset is smaller and 400 for 2021-2022. Both periods produced in the first instance between 120 and 140 topics. To ensure that the number of topics are the same as LDA for having an equal comparison we reduced the number of topics using the fuction 'nr_topics' stated in the documentation of BERTopic [17] that used Agglomerative Clustering on the cosine distance matrix of the topic c-TF-IDF or semantic embeddings.

3.8.3 BERTopic quantized with LLAMA-3-8B LLM. The last in our comparison of topic modeling methods is the BERTopic quantized with LLAMA-3-8B model. Several studies have explored the integration of LLMs like GPT and LLaMA with topic modeling techniques. For instance, [27] investigates the impact of ChatGPT on programmer help-seeking behavior and uses GPT-2 for generating answers. Similarly, [47] compares the capabilities of LLMs like ChatGPT 3.5 and LLaMA-2 in generating high-quality answers compared to human-generated answers on SO. Another study, [63], introduces PromptTopic, a novel approach to topic modeling that leverages LLMs like ChatGPT and LLaMA to integrate word and sentence semantics for a more holistic topic modeling experience. Despite these advancements, none of the studies reviewed have combined BERTopic with quantized LLMs for topic modeling and extracting topics from questions and answers. The unique integration of BERTopic with a quantized LLM, such as LLaMA-3-8b, can potentially enhance topic modeling performance by providing more coherent, contextually accurate topics and to improve topic quality. Moreover, most of the studies using traditional topic modeling methods like LDA required manual labeling of topics. This was necessary because the topics generated were often too similar, making it difficult to distinguish between them without human intervention [54],[52],[28],[49]. This manual process is time-consuming and prone to inconsistencies, highlighting the need for more advanced and automated approaches.

The implementation leverages the transformer-based Llama model, which is known for its robust natural language processing abilities [58]. We initialize the Llama model with a specific configuration for 4-bit quantization using the BitsAndBytes library [1]. This quantization process involves loading the model in 4-bit precision, utilizing normalized float 4 (nf4) quantization type, applying a second quantization layer (double quantization), and performing computations in bfloat16 precision. This setup ensures efficient and effective model performance on available GPU resources. The Llama model is loaded using the AutoModelForCausalLM class from the Transformers library, and it is set up for text generation tasks with parameters such as low temperature (0.1) and a maximum of 500 new tokens, which helps in generating precise and concise outputs. Similar to the BERTopic, for the embedding model, we use the SentenceTransformer model "BAAI/bge-small-en" to encode the documents into embeddings, capturing semantic meanings and relationships. The Llama model generates topic labels and summaries for each cluster. It uses the embeddings to understand the context and content of the documents within a cluster. By analyzing the entire documents and their embeddings, Llama can generate these labels. The embeddings serve as a reference to ensure that the generated labels are contextually relevant and semantically accurate. To generate topic labels using LLaMA 3, we opted for a custom command format due to the issue of repeated labels when using predefined examples as used in [17]. More specifically we observed that many documents ended up with the same label as the examples provided. This redundancy suggested that the examples might have biased the model, leading to less accurate and more subjective labels. This approach allowed LLaMA 3 to generate contextually relevant and unique short labels for each topic as depicted in the Results Section 4, based solely on the provided documents and keywords.

```
system_prompt = """
<s>[INST] <<SYS>>You are a helpful,
    respectful and honest assistant for
    labeling topics.<</SYS>>"""

main_prompt = """
[INST]
I have a topic that contains the following
    documents:
[DOCUMENTS]

The topic is described by the following
    keywords: '[KEYWORDS]'.

Based on the information about the topic
    above, please create a short label of
    this topic. Make sure you to only return
    the label and nothing more.
[/INST]
"""

combined_prompt = system_prompt + main_prompt
```

3.8.4 KeyBERT and POS. In addition to the previous models we have also added KeyBERT and POS [16], because they have been easy to incorporate into the representation_model which is configured in BERTopic, to use different approaches based on the specific requirements and nature of the data. As you're going to see in the Results Section 4 these methods performed better than traditional approaches like simple BERTopic and LDA but not as good as BERTopic quantized with Llama-3. KeyBERT is an advanced keyword extraction method that leverages the powerful embeddings from BERTopic to identify the most relevant keywords and phrases from a corpus of documents. Unlike traditional keyword extraction methods that rely on frequency-based techniques, KeyBERT captures the contextual and semantic relationships between words. For each topic discovered by BERTopic, KeyBERT identifies the most representative documents. It does this by sampling several documents from each topic cluster and calculating their c-TF-IDF scores. Then, for each word or phrase within a document, KeyBERT calculates the cosine similarity between the embeddings of individual keywords or phrases extracted from the documents and the document embedding. This step ensures that the extracted keywords are not only frequent but also semantically relevant to the overall document context. The keywords are ranked based on their similarity scores. The top N keywords with the highest scores are selected as the most representative of the document's content.

On the other side POS tagging, is a fundamental NLP technique used to annotate words in a text with their corresponding part-of-speech tags, such as nouns, verbs, adjectives, etc. This process is used for understanding the grammatical structure of the text and identifying key syntactic elements. One common approach is to use statistical models trained on large annotated corpora that everage algorithms like Hidden Markov Models (HMMs). Nouns and proper nouns, often identified through POS tagging, can be prioritized as potential keywords. The candidate keywords are filtered through predefined patterns and sorted by their c-TF-IDF values.

3.8.5 Metrics. In evaluating the effectiveness of topic modeling, two crucial metrics are commonly used: topic coherence [26], [47], [68] and topic diversity [49], [63].

Topic coherence [20] measures how semantically related the words within a single topic are. In simpler terms, it evaluates whether the words that make up a topic actually belong together and make sense as a group. High topic coherence indicates that the words are closely related and form a coherent theme, which is crucial for the interpretability of the topics. Coherence is typically calculated using statistical measures that assess the pairwise similarity between the words in a topic, often leveraging external resources such as word embeddings or co-occurrence statistics from large corpora. For instance, if a topic includes words like "python," "coding," "programming," and "software," a high coherence score would suggest these words naturally belong together in the context of programming. Several studies in the related work, such as those by [26], [47], and [68], emphasize the importance of topic coherence. They have employed various techniques to ensure that the topics generated are coherent and meaningful, often using manual validation to confirm the coherence scores.

Topic diversity [20], on the other hand, measures the extent to which the topics cover a broad range of themes. It assesses whether the model captures a wide variety of distinct topics or if it redundantly generates similar topics. High topic diversity indicates that the topics span a broad spectrum of different themes, making the topic model more valuable for exploring diverse aspects of the data. In practical terms, it means that the topics generated by the model should not overlap significantly and should provide unique and distinct insights into the data. Studies such as [49] and [63] have highlighted the importance of topic diversity in their analyses. These studies often utilize metrics like the proportion of unique words across all topics to gauge diversity. Ensuring high topic diversity is crucial for applications where a comprehensive understanding of different themes and trends within the data is required. For instance, in analyzing discussions on SO, high topic diversity would mean covering a wide range of programming languages, tools, and development practices, providing a richer and more informative overview.

Additionally, we have also included topic share, topic relationships, and topic trends over time. These metrics have been used in studies such as [12], [54], and [48]. Besides their predefined goal we have also used them to create plots on the already tagged questions from SO for initial analysis.

Topic share measures the proportion of posts that contain a specific topic z_k. This helps in understanding the relative popularity of a topic across all posts. The mathematical formula for topic share is given by:

$$\text{share}(z_k) = \frac{1}{|D|} \sum_{d_i \in D} \mathbf{1}(d_i, z_k \geq \delta) \tag{1}$$

where D is the set of all posts in our dataset, and $\mathbf{1}(d_i, z_k \geq \delta)$ is an indicator function that equals 1 if the topic z_k in post d_i is above a threshold δ.

Topic relationships investigate the relationship between topics found in questions and their corresponding answers. This metric quantifies the influence of divergent answer topics with respect to the question topics. The formula for topic relationships is:

$$\text{rel}(z_q, z_a) = \sum_{d_i \in Q, A(d_i)} \theta(d_i, z_q) \times \theta(d_i, z_a) \tag{2}$$

where Q is the set of all question posts and $A(d_i)$ is the set of all answers related to question d_i. $\theta(d_i, z_q)$ and $\theta(d_i, z_a)$ are the topic distributions for the question and answer, respectively.

Topic trends over time analyze the temporal dynamics of topics. The impact metric assesses the prevalence of a topic z_k in a specific month m:

$$\text{impact}(z_k, m) = \frac{1}{|D(m)|} \sum_{d_i \in D(m)} \theta(d_i, z_k) \tag{3}$$

where $D(m)$ is the set of all posts in month m, and $\theta(d_i, z_k)$ is the topic distribution for post d_i.

4 RESULTS

In this section, we present our findings and systematically address each research question along with its subquestions. Each subsection delves into specific aspects of the research, for more detailed comparisons and examination of the data from 2021-2022 and 2023-2024 periods.

4.1 RQ1 - How has the overall activity on SO changed from April 1, 2021, to April 1, 2022, compared to April 1, 2023, to April 1, 2024?

The following research questions aim to investigate the overall activity and trends on SO over two distinct time periods: April 1, 2021, to April 1, 2022, and April 1, 2023, to April 1, 2024. By comparing these periods, we seek to understand how the platform's usage has evolved, focusing on various aspects such as tag frequency, question and answer dynamics, user engagement, and response times based. The data used in this analysis is exactly the data extracted from our JSON files with the two periods of time, and the tags were not processed using topic modeling, being the tags pre-defined by users on SO.

4.1.1 Are there noticeable trends differences based on the fre- quency of SO tags? For answering these questions, we have made a grouped bar chart showing the average usage of the most used tags in the SO questions in Figure 3. Some key observations for example are the dominance of Python, which remains the most used tag in both periods. However, there is a noticeable overall decrease in the average count of usage for all tags, with Python's count dropping from over 18,000 to about 8,000, and similar reductions observed for JavaScript and Java, being inline with the overall decrease in usage of SO. The inclusion of Flutter in the 2023-2024 data points to a growing interest in mobile and cross-platform development. Developers are increasingly turning to tools like Flutter to create applications that

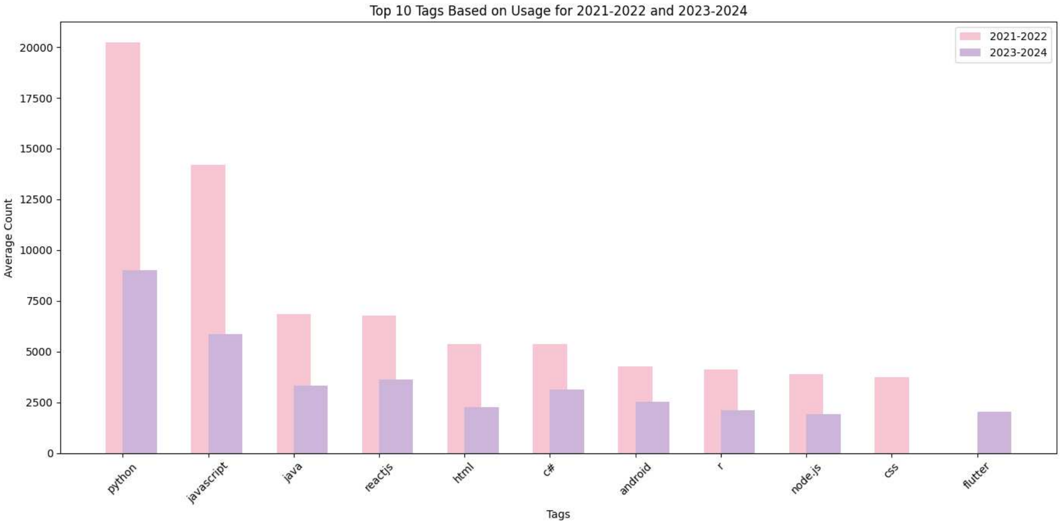


Fig. 3. Top 10 tags based on average usage

can run on multiple platforms efficiently, highlighting the demand for streamlined development processes across both Android and iOS.

Next, we made four graphs depicting the "impact" of SO tags to provide a perspective on the shifting interests. The graphs showing the top decreasing tags impact over time in Figures 4 and 5 reveal a decline in some dominant technologies. The variations in impact percentages are not dramatic but rather subtle of almost 0.8% for Python and 0.05% for JavaScript in the later period analysed. These two and HTML decline in percentages on Y-axis might indicate a saturation of available information and solutions related to these technologies.Alternatively, it could reflect a maturation where the foundational questions have largely been addressed, where we can see a decrease in questions asked.

On the other hand in Figures 6 and 7 we can see an increase of newer technologies such as Flutter, Next.js, and React-Native. The growth in the impact of these tags underscores maybe an interest in cross-platform development capabilities, server-side rendering, and native mobile app development using familiar web technologies. The rise in these discussions, particularly in a time of overall decrease in total content volume, highlights their growing relevance. This gradual increase could be indicative of a steady but slow adoption or increasing curiosity rather than a sudden change in popularity.

4.1.2 Has there been a significant change in the number of questions posted per month? In this subquestion we are focusing on the number of questions posted per month for the two different periods, visible in Figure 8. At the beginning of April 2021, there were approximately 135,000 questions posted per month. By the end of March 2024, this number had decreased to around 60,000 questions per month. This substantial decrease highlights a significant shift in user engagement on the platform over these three years. The plot for 2021-2022 shows a clear overall declining trend in the number of questions posted per month. Starting in April 2021 at over 130,000 questions, the number steadily decreases, reaching around 110,000 questions by March 2022. The monthly percentage changes highlight this decline, with notable drops in certain months. For instance, from April to May 2021, there is a small decrease of about -0.74%, while from May to June 2021, the decline is more pronounced at approximately -2.99%.

In contrast, the 2023-2024 period starts at a lower baseline of around 70,000 questions in April 2023 but shows more volatility throughout the year. The number of questions peaks in July 2023 at approximately 85,000 before experiencing a sharp decline towards the end of the year, bottoming out around 55,000 in November 2023. This period also displays significant monthly percentage changes. For example, from May to June 2023, there is a substantial increase of about 16.90%, indicating a burst of activity possibly driven by specific events or new technological trends as explained in the later Discussion chapter 5. To determine if these observed changes are statistically significant, we employed an independent two-sample T-test [65] to compare the mean percentage changes between the two periods. The resulted p-value was less than 0.05, more exactly 0.037, suggesting that there is a significant difference in the question posting patterns between the two periods.

4.1.3 How has the number of answers per question evolved over time? To understand how the number of answers per question has evolved over time, we made a plot Figure 9 containing both periods from April 2021 to March 2024. From April 2021 to around mid-2021, the average number of answers per question remained relatively stable, hovering around 1.1. This indicated a consistent level of engagement where most questions were receiving at least one answer, with some getting more. From late 2021 onwards, there is a gradual decline in the average number of answers per question. By early 2023, this average drops below 1.0, indicating that many questions were not getting answered or were getting fewer answers overall. The sharp decline starting around March 2023, where the average plummets from above 1.0 to around 0.7 by mid-2023 was the most strong one. This sharp drop suggests that fewer users are answering questions.

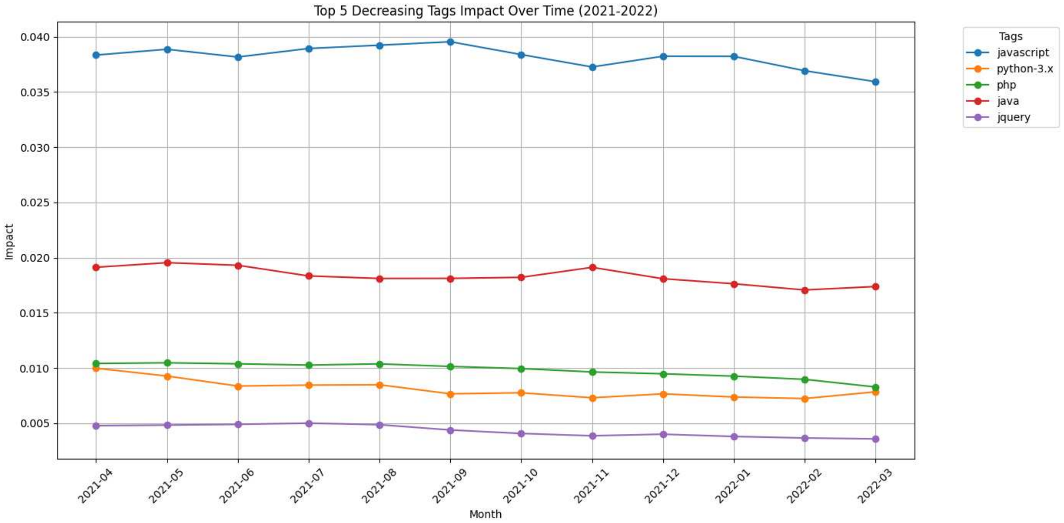


Fig. 4. Top 5 decreasing tags 2021-2022

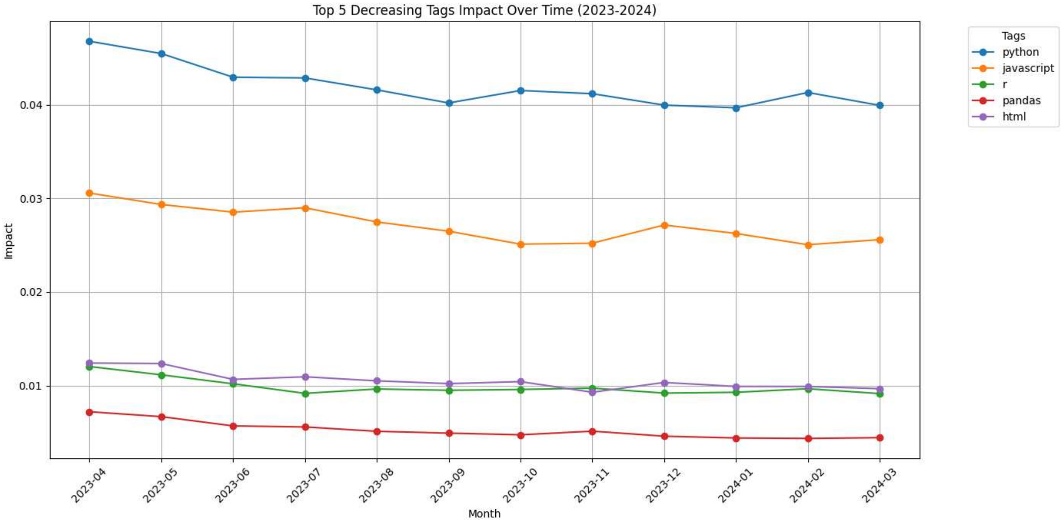


Fig. 5. Top 5 decreasing tags 2023-2024

To gain deeper insights into the number of answers each question receives on SO, we have also plotted the distribution of answer counts for the periods 2021-2022 and 2023-2024 in Figure 10. These distribution graph help visualize how many questions receive zero, one, two, or more answers. For the period 2021-2022 single-answer questions dominate this period, with slightly over 300,000 questions receiving one answer. The number of questions receiving two or more answers decreases steadily, with fewer than 100,000 questions receiving three answers. There is also a noticeable long tail, with some questions receiving up to 76 answers. Although rare, these questions might contain popular topics that attract extensive community input.

In contrast, the distribution for 2023-2024 shows a higher relative number of unanswered questions, exceeding 100,000. This increase in unanswered questions could reflect the decline in community engagement seen also in the other graphs. Compared to the previous period fewer questions received a single answer. The decline in the number of questions receiving two or more answers is more pronounced in this period, with fewer than 10,000 questions receiving three answers, and the distribution tails off quickly with a maximum of up to 31 answers per question. In terms of percentages and numbers, for 2021-2022, over 20% of the questions posted received no answers, about 20% received exactly one answer, and questions receiving two answers account for roughly 10%, with rapidly decreasing numbers for three or more answers. For

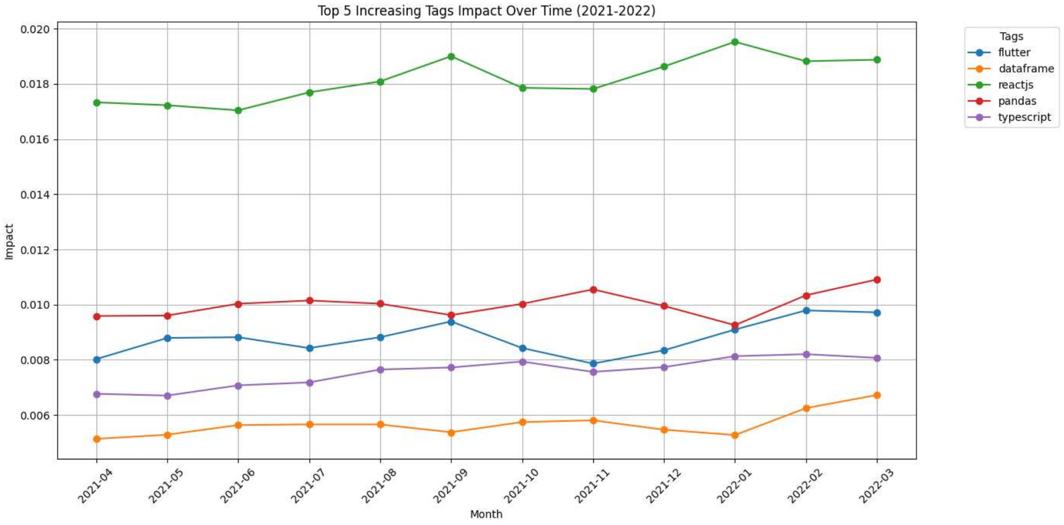


Fig. 6. Top 5 increasing tags 2021-2022

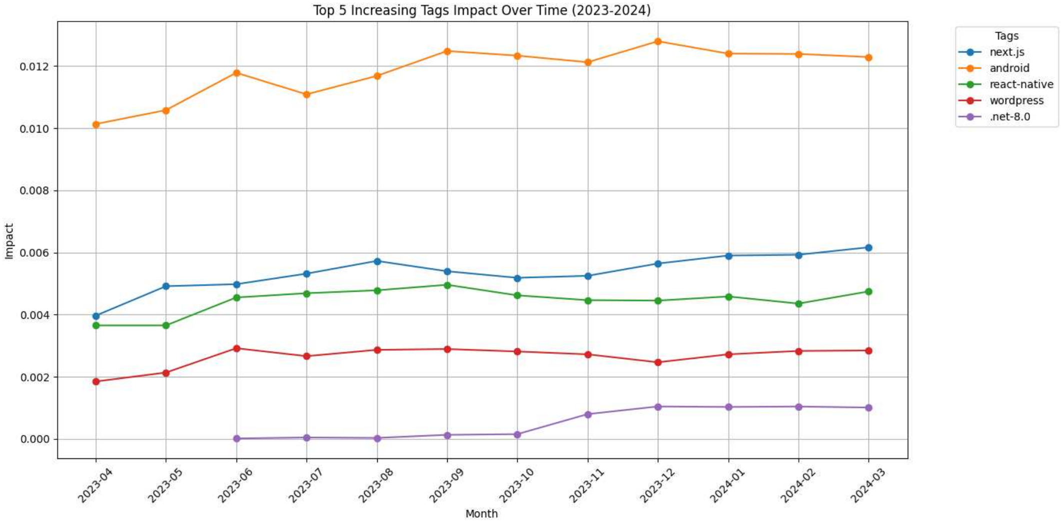


Fig. 7. Top 5 increasing tags 2023-2024

2023-2024, the percentage of unanswered questions has increased, potentially exceeding 30%, while the proportion of questions receiving one answer has decreased to around 15%. The trend of fewer multi-answer questions continues, with less than 5% of questions receiving three answers.

4.1.4 How has the average time to receive an accepted answer evolved over time ? In the following histogram 11, it illustrates the time taken for questions to receive an accepted answer on SO, focusing on the first 10 hours (600 minutes) for the periods 2021-2022 and 2023-2024. The pink bars represent the data from 2021-2022, while the purple bars represent the data from 2023-2024. For the period 2021-2022, there is a significant spike in the number of questions receiving an accepted answer within the first few minutes (30 to 35 minutes). The initial peak shows over 12,000 questions being resolved almost immediately. This trend continues, with a steep decline observed as the time increases. The majority of questions receiving an accepted answer do so within the first 100 minutes. The number of questions that get an accepted answer gradually diminishes beyond this point. For the period 2023-2024 there is still a peak in the number of questions receiving an accepted answer shortly after posting, this peak is not as pronounced as in the previous period. The initial peak in 2023-2024 is around 7,000 questions, significantly lower than the 2021-2022 period. A closer comparison reveals that in the 2021-2022 period, approximately 90% of questions that received an accepted answer did so within the first 200 minutes. In the 2023-2024

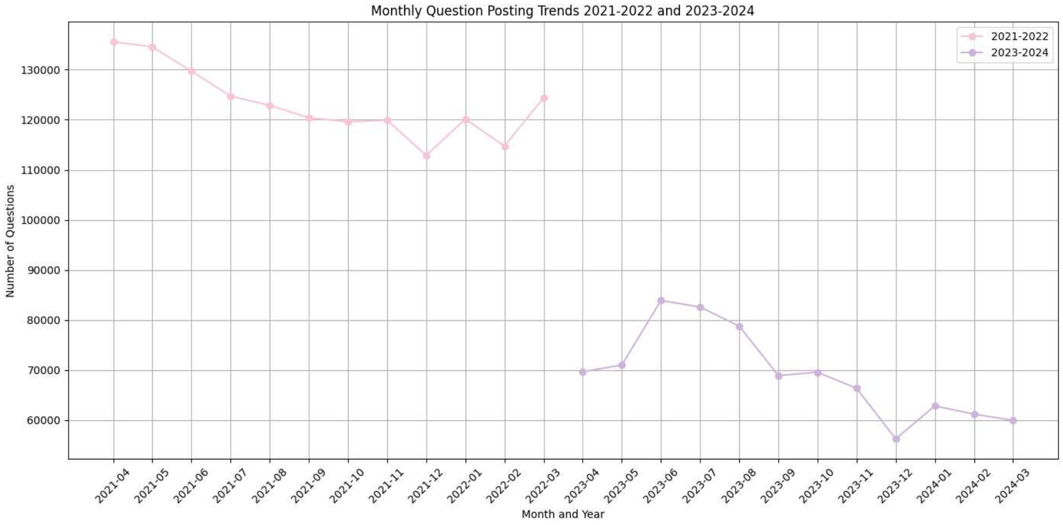


Fig. 8. Number of questions per month

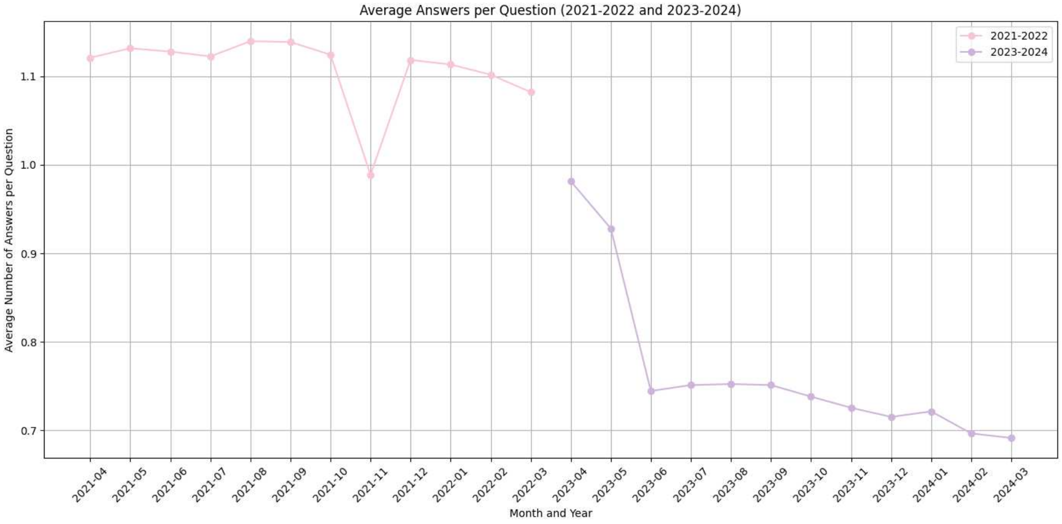


Fig. 9. Average number of answers for questions

period, this percentage is notably lower, around 70%. It is important to note that beyond the 10-hour mark, the tail of the distribution extends up to approximately 375 days. The number of questions receiving an accepted answer remains consistent between 0 and 2000 for each day up to 375 days. However, this tail was not included in the main focus of the analysis, as the most significant interactions occur within the first 10 hours. Overall, the comparison between the two periods indicates a decline in the speed at which questions receive accepted answers, reflecting the community disengagement in answering questions on SO.

4.1.5 How has the average time to receive an accepted answer evolved over time ? For answering this subquestion we've provided this graph in Figure 12 that illustrates the monthly average view counts per question on SO for both periods of time. During the period from April 2021 to March 2022, the average view count per question remained relatively stable, hovering around 1000 views per question. The minor fluctuations observed in the average view count suggest typical variations in user activity without any notable changes in overall engagement. By April 2023, the average view count had decreased to approximately 500, representing a 50% reduction. From April 2023 to March 2024, the average view count dropped from about 500 to below 200, marking a further decline of approximately 60% over this period. These significant decreases in the average view count per question indicates a notable drop of SO users.

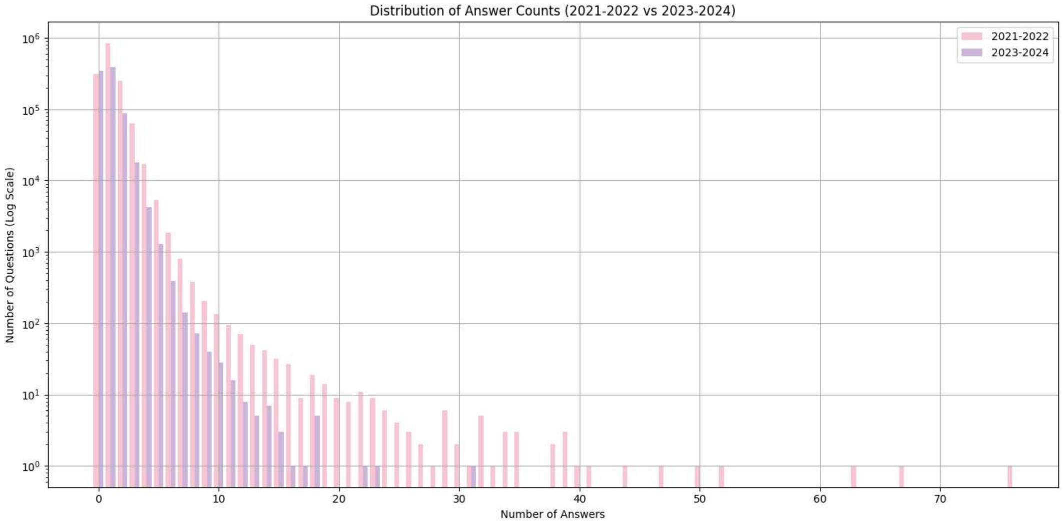


Fig. 10. Distribution of answer counts

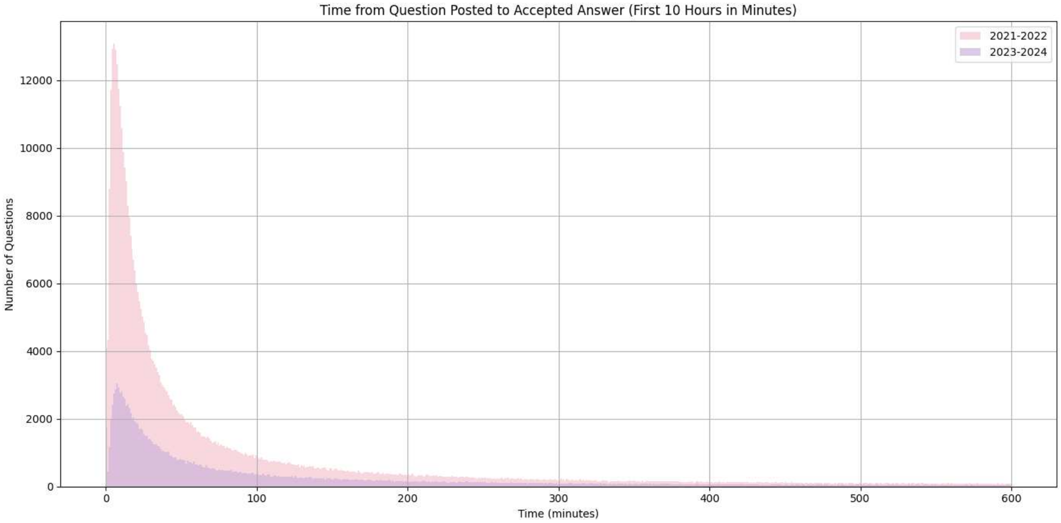


Fig. 11. Waiting time for receiving an accepted answer

4.1.6 What is the proportion of questions with accepted answers changed over time? The pie charts in Figure 13 illustrate the proportion of questions with accepted answers on SO for the periods 2021-2022 and 2023-2024. The first pie chart represents the data from 2021-2022, while the second pie chart represents the data from 2023-2024. For the period 2021-2022, 42.3% of questions had accepted answers, while 57.7% did not. This indicates a moderately high level of user engagement, with a substantial portion of questions being resolved through community interactions. In contrast, the period 2023-2024 shows a marked decrease in the proportion of questions with accepted answers. Only 28.2% of questions had accepted answers, while a significant 71.8% did not. This represents a notable decline in the effectiveness of community engagement in providing accepted answers to questions. The decrease of 14.1 percentage points (from 42.3% to 28.2%) suggests that fewer questions are being resolved satisfactorily or less people engage in giving answers and help to the community as suggested in [45].

4.2 RQ2 - Has the introduction of ChatGPT influenced the types of questions asked on SO?

4.2.1 Are there noticeable trends in the frequency of SO tags related to AI, LLMs, and specifically ChatGPT before and after its introduction? In the first two subquestion we want to explore and check if tags related to AI, LLMs, and specifically ChatGPT exist on SO and how the frequency of tags related to this field changed before and after its introduction. The tags analyzed were chosen

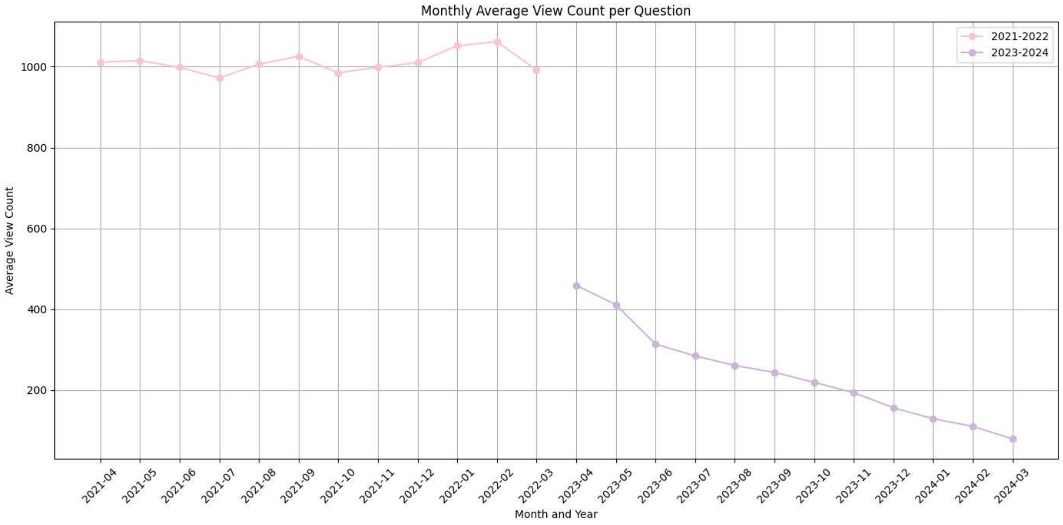


Fig. 12. Average View Count for both periods of time

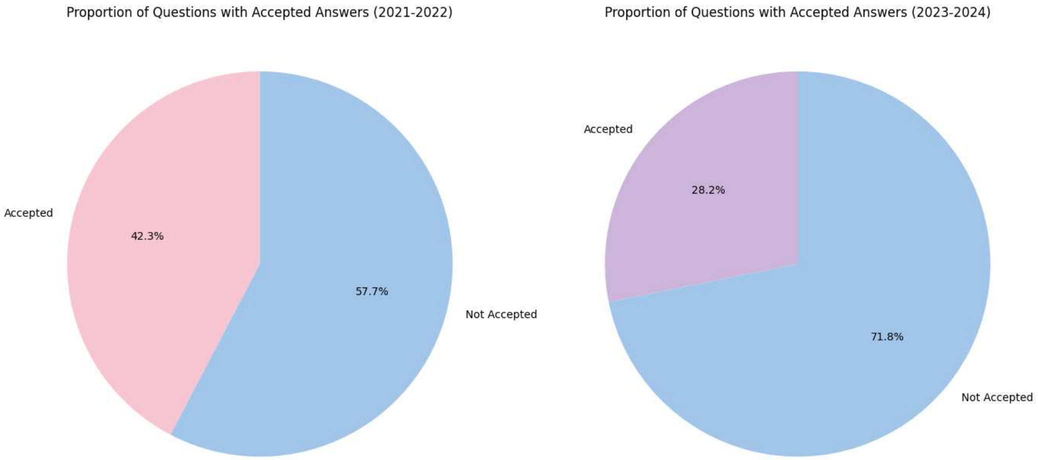


Fig. 13. Proportion of questions with accepted answers changed over time

based on their popularity from all unique tags found related to this field. Specifically, we focused on tags such as ’chatgpt’, ’openai’, ’large-language-model’, ’gpt-3’, ’gpt-4’, ’llm’, ’tensorflow’, ’pytorch’, ’machine-learning’, ’deep-learning’, ’nlp’, ’transformer’, ’language-model’, ’ai’, ’artificial-intelligence’, and ’neural-networks’. The first set of plots illustrates the popularity of the selected tags over time, comparing the periods from 2021-2022 in Figure 14 and 2023-2024 in Figure 15. The tag ’chatgpt’ saw a significant rise from nearly no occurrence in 2021-2022 to approximately 50 questions per month in 2023-2024. Similarly, the tag ’openai’ increased from around 50-60 questions per month to 300-400 questions per month, reflecting a 500% to 600% increase. The ’large-language-model’ tag also surged from minimal occurrences to about 150-200 questions per month. Tags like ’gpt-3’ and ’gpt-4’ showed noticeable increases, with ’gpt-3’ rising by 50% to 100% and ’gpt-4’ emerging to around 10-20 questions per month. In contrast, more traditional AI tags such as ’machine-learning’ and ’deep-learning’ experienced slight declines, with ’machine-learning’ decreasing by approximately 20% and ’deep-learning’ by about 25%. These trends highlight a interest and shift towards newer AI technologies, particularly those related to ChatGPT and OpenAI, while interest in traditional AI and machine learning tags has either stabilized or slightly dropped.

4.2.2 Are questions tagged with AI/ChatGPT-related tags more likely to receive answers compared to other tags? The second set of plots compares the monthly answer rates for questions tagged with AI/ChatGPT-related tags versus other tags for both periods Figure 16.

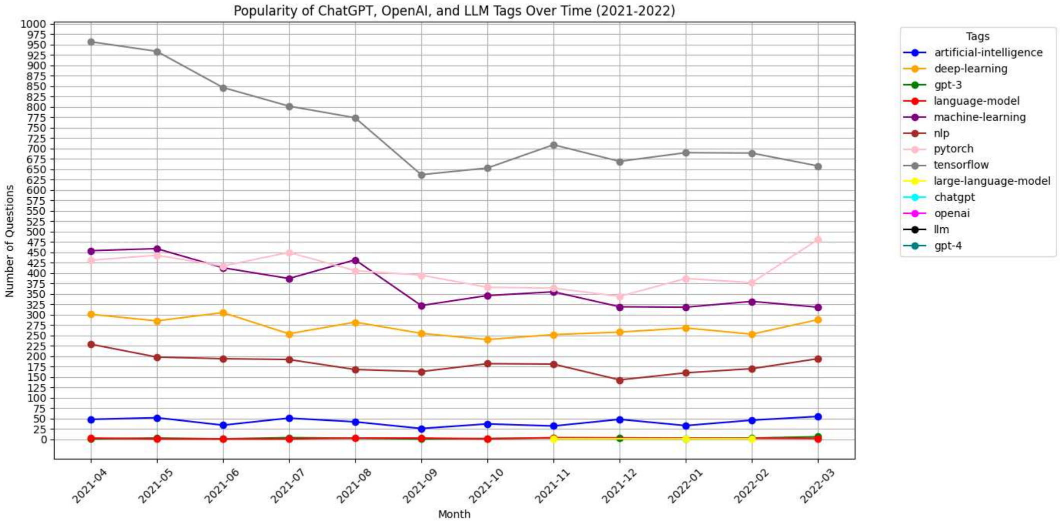


Fig. 14. Popularity of AI/ChatGPT-related tags 2021-2022

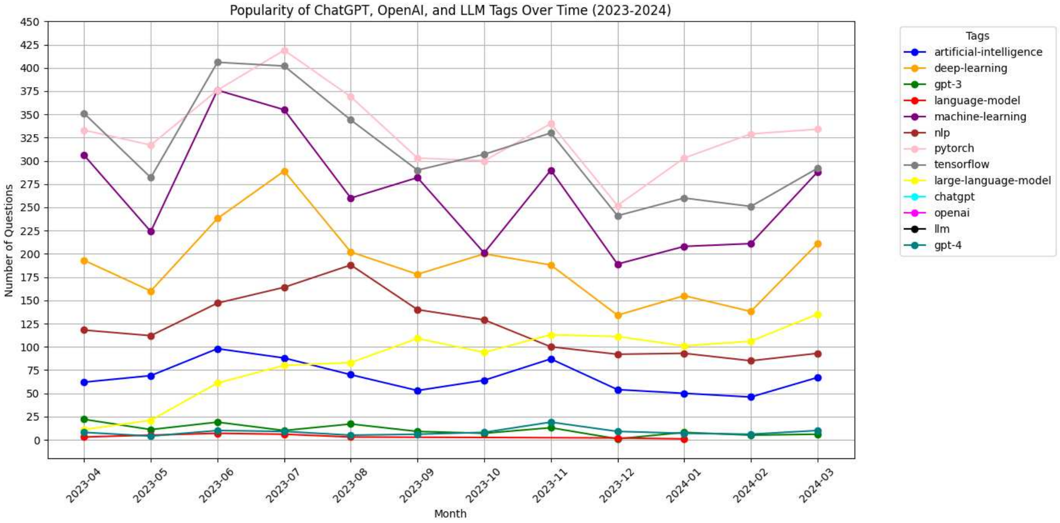


Fig. 15. Popularity of AI/ChatGPT-related tags 2023-2024

We can say that changes exist in the answer rates for AI/ChatGPT-related tags compared to other tags. During the 2021-2022 period, the answer rate for AI/ChatGPT-related tags decreased by approximately 7.14%, while other tags saw a smaller decline of around 3.33%. In the 2023-2024 period, the answer rate for AI/ChatGPT-related tags experienced a more pronounced decrease of about 35%, compared to a 32% decrease for other tags. Despite the reduction of questions and answers volume on SO, the decrease in answer rates for AI/ChatGPT-related tags is comparable to the overall decline, suggesting that the challenges in providing timely responses to AI-related queries may have another reason. An example can be the complex discussions that require more expertise and resources to address adequately.

4.3 How has the grammatical structure and syntactic complexity of messages changed over time?

To address the sub-question about changes in grammatical structure over time, we employ POS tagging and dependency parsing. We use spaCy to tag parts of speech in the messages, identifying components such as nouns, verbs, and adjectives. Dependency parsing is then applied to analyze the grammatical structure of the sentences, revealing the relationships between words and phrases within the sentences.

Next, we proceed with feature extraction, where we extract metrics such as sentence length to provide a basic measure of sentence

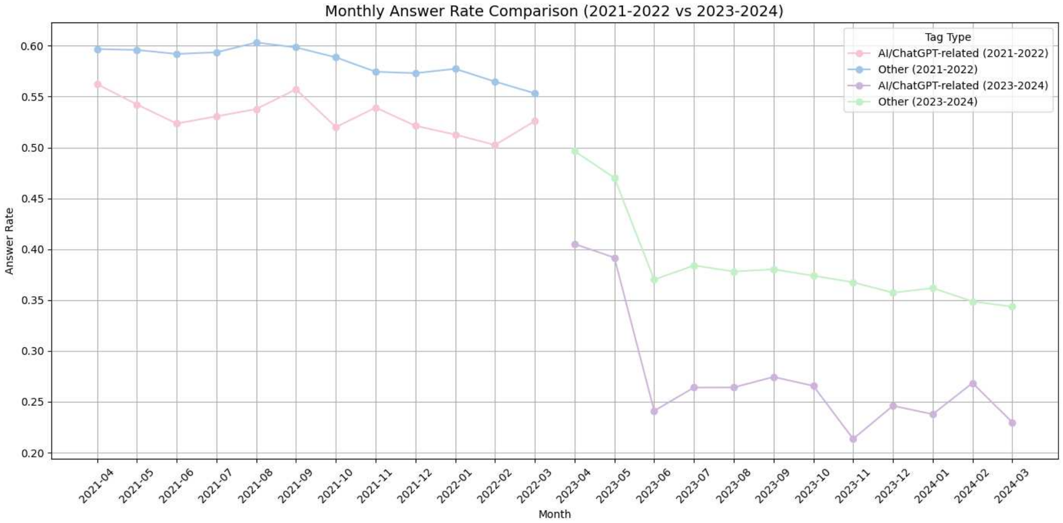


Fig. 16. Answer rate of AI/ChatGPT-related tags vs other tags 2021-2022

complexity. We also examine the distribution of POS tags to understand the text composition in terms of parts of speech and syntactic structures, such as dependency trees, which offer a detailed view of sentence construction. For measuring syntactic complexity, we consider the depth of dependency trees to provide an indication of how deeply nested the syntactic structures are.

Table 5 provide the results based on POS tagging and dependency parsing on the SO posts with selected answers. For 2021-2022, the POS tagging analysis revealed the most frequent tags were nouns, verbs, and punctuation. Specifically, there were 20,053,596 nouns, 14,496,216 verbs, and 12,895,549 punctuation marks. This shows that there is a substantial use of descriptive elements and actions, with a significant amount of structural punctuation. Pronouns and determiners also had high counts, highlighting the presence of referencing and specifying elements in the text. In comparison, the 2023-2024 period showed a decrease in overall counts due to the smaller dataset size, but the distribution remained similar. The most common tags were 8,822,127 nouns, 6,308,473 verbs, and 5,710,373 punctuation marks. Despite the smaller dataset, the relative frequencies suggests a stable pattern. The dependency parsing for 2021-2022 revealed the most frequent dependency roles were punctuation ('punct'), determiners ('det'), and nominal subjects ('nsubj'). Specifically, there were 13,376,028 instances of punctuation, 9,928,393 determiners, and 9,176,854 nominal subjects. This indicates structurally complex text with clear subject-predicate relationships and significant detail provided by determiners. For 2023-2024, the pattern persisted with 5,940,284 instances of punctuation, 4,412,985 determiners, and 3,921,126 nominal subjects. This is also in line with the smaller number of documents. What is interesting is the the average sentence length that increased from 206.67 in 2021-2022 to 241.91 in 2023-2024. This 17% increase suggests that posts in the latter period contained more detailed or complex sentences, potentially reflecting more elaborate explanations or queries.The average tree depth also increased from 52.61 to 56.73 in 2023-2024, indicating more complex sentence structures with deeper nested relationships. The users might have more elaborate questions and answers due to the increasing complexity of programming languages and technologies discussed similar to the findings of [66]. On the other hand, the increase could be attributed to users being inspired by the detailed and comprehensive answers generated by AI models.

Table 5. Comparison of Key Metrics for 2021-2022 and 2023-2024

Metric	**2021-2022**	**2023-2024**
Total Documents	568,677	212,735
Nouns (NOUN)	20,053,596	8,822,127
Verbs (VERB)	14,496,216	6,308,473
Punctuation (PUNCT)	12,895,549	5,710,373
Pronouns (PRON)	10,110,043	4,222,841
Determiners (DET)	10,010,028	4,445,298
Adpositions (ADP)	9,360,127	4,081,892
Adjectives (ADJ)	5,202,300	2,311,899
Proper Nouns (PROPN)	4,605,483	2,155,709
Adverbs (ADV)	4,208,522	1,846,795
Conjunctions (CCONJ)	2,918,677	1,287,301
Subordinating Conjunctions (SCONJ)	2,917,935	1,258,703
Numbers (NUM)	1,716,527	778,359
Symbols (SYM)	366,794	168,430
Interjections (INTJ)	143,006	53,970
Average Sentence Length	206.67	241.91
Average Tree Depth	52.61	56.73

4.4 How unique are the stems and vocabulary used in messages over time?

For this subquestion, we utilized stemming and lemmatization techniques to reduce words to their base or root forms. Stemming, specifically using the Porter Stemmer, was applied to the corpus of SO questions and answers. This transformation helped identify the core vocabulary by stripping words down to their simplest forms, for a precise comparison of lexical diversity. We assessed the uniqueness of stems by counting the number of distinct stems in the corpus for each period to determined how varied the vocabulary was and how it changed over time. We calculated two key metrics: the Type-Token Ratio (TTR) and the Shannon Diversity Index. TTR is a measure of lexical diversity calculated as the ratio of the number of unique words (types) to the total number of words (tokens) in a text [43] [22]. A higher TTR indicates a more diverse vocabulary. The Shannon Diversity Index [23] [57], considers both the abundance and evenness of word usage, providing a more comprehensive measure of lexical diversity compared to TTR alone. It measures the entropy in the distribution of word frequencies. It accounts for how many different words are used and how evenly the frequencies of those words are distributed.

By examining the Table 6 we can see the that the number of unique stems decreased from 1,569,401 in 2021-2022 to 856,457 in 2023-2024. This reduction, amounting to a decrease of approximately 45.4%, indicates a substantial narrowing of the vocabulary. The possible reasons for this could be the increased reliance on automated tools like ChatGPT, which might generate more standardized and less varied language compared to human contributors. On the other hand, the TTR increased from 0.0192 in 2021-2022 to 0.0242 in 2023-2024. This 26% increase suggests a higher proportion of unique words relative to the total word count in the latter period. This could indicate that despite the lower absolute number of unique stems, the texts in 2023-2024 are more lexically diverse relative to their length. It suggests that users might be employing a more varied vocabulary in their posts, even if the overall number of unique stems is lower. The Shannon Diversity Index, also saw a slight increase from 10.7321 to 10.8079 suggesting a more even distribution of word usage in the 2023-2024 period. No single word overwhelmingly dominates the text, enhancing the richness of the vocabulary. The overall result based on all of the 3 findings indicate a shift towards more concise, focused communication, possibly influenced by the increasing use of AI tools like ChatGPT that can generate clear and coherent text efficiently.

Metric	2021-2022	2023-2024
Unique Stems	1,569,401	856,457
Type-Token Ratio (TTR)	0.0192	0.0242
Shannon Diversity Index	10.7321	10.8079

Table 6. Comparison of Lexical Diversity Metrics between 2021-2022 and 2023-2024

4.5 RQ3 - How do different topic modeling methods (LDA, BERTopic, KeyBERT, POS and BERTopic Quantized with LLaMA 3 - 8B) compare in terms of coherence scores and topic diversity?

The aim of this research question is to find out which method out of the ones enumerated perform the best based on Topic Coherence and Topic Diversity. For this we're using the Topic Quality, a composite measure combining coherence and diversity, essential in evaluating the overall performance of topic models. The formula for topic quality, as defined in the paper [15] is:

$$\text{Topic Quality} = \text{Coherence} \times \text{Diversity} \tag{4}$$

Starting with the results from the period 2021-2022 Table 8, LDA exhibited a gradual increase in coherence scores as the number of topics increased, peaking at 0.539 for 40 topics before slightly declining. The topic diversity for LDA also improved with the number of topics, reaching a maximum of 0.985 at 100 topics. BERTopic, although it started with lower coherence scores (0.379 for 5 topics), showed a significant improvement as the number of topics increased, achieving a coherence score of 0.6286 at 100 topics. However, BERTopic's topic diversity was relatively lower, with a maximum of 0.6343 at 100 topics. KeyBERT consistently performed well in coherence, with the highest score of 0.7332 at 40 topics, and maintained high topic diversity across various topic counts, peaking at 0.9571 for 15 topics. The POS model demonstrated strong performance in both coherence and diversity, achieving a coherence score of 0.6914 and a diversity score of 0.966 for 5 topics. The best performer was BERTopic quantized with LLaMA 3 - 8B, which achieved the highest coherence score of 0.9264 at 50 topics and maintained a respectable topic diversity, peaking at 0.8852 for 5 topics.

In the period 2023-2024 Table 9, similar trends were observed. LDA's coherence scores ranged from 0.3933 for 5 topics to 0.5406 for 50 topics, with topic diversity reaching 0.976 at 100 topics. BERTopic showed improvement in coherence, peaking at 0.6127 for 100 topics, with a corresponding diversity score of 0.6565. KeyBERT maintained its strong performance with the highest coherence score of 0.7705 for 5 topics and high diversity scores, the highest being 0.9613. POS continued to perform well, with a coherence score of 0.7117 and a diversity score of 0.966 for 5 topics. BERTopic quantized with LLaMA 3 - 8B again excelled, achieving the highest coherence score of 0.9123 for 50 topics and a significant diversity score of 0.8741 for 50 topics.

When examining the Topic Quality in Table 7, it becomes evident that BERTopic quantized with LLaMA 3 - 8B consistently provided the highest topic quality across both periods. In the 2021-2022 period, the highest topic quality score was 0.7891 at 50 topics, while in the 2023-2024 period, the highest topic quality score was 0.7969 also at 50 topics. This model demonstrated the best balance between coherence and diversity, making it the most effective method for extracting high-quality topics from our data.

4.6 RQ4 - What are the main topics of discussion on SO in the specified periods, and how have these topics shifted over time?

4.6.1 How do the topics extracted using various topic modeling methods (e.g., LDA, BERTopic, KeyBERT, POS, and BERTopic quantized with LLaMA 3) compare in appearance and content? For answering this subquestion a number of different visualization were made per method to have an overview of how the models worked and what would the topics would look like compared to the pre-defined tags of SO data.

To visualize the different topics generated by the LDA model, we utilized pyLDAvis [46] which is a library particularly useful

Table 7. Topic Quality for 2021-2022 and 2023-2024

Model	**Num. Topics**	**2021-2022 Topic Quality**	**2023-2024 Topic Quality**
LDA	5	0.3012	0.2596
LDA	15	0.3939	0.3565
LDA	30	0.4919	0.4635
LDA	40	0.5040	0.4857
LDA	50	0.4895	0.5091
LDA	75	0.4399	0.4453
LDA	100	0.3983	0.4184
BERTopic	5	0.1421	0.1309
BERTopic	15	0.0804	0.1066
BERTopic	30	0.1499	0.1652
BERTopic	40	0.2029	0.1915
BERTopic	50	0.2421	0.2416
BERTopic	75	0.3298	0.3529
BERTopic	100	0.3987	0.4020
KeyBERT	5	0.5271	0.7407
KeyBERT	15	0.6955	0.6503
KeyBERT	30	0.6366	0.5950
KeyBERT	40	0.6184	0.5802
KeyBERT	50	0.6044	0.5692
KeyBERT	75	0.5660	0.5217
KeyBERT	100	0.5312	0.5149
POS	5	0.3404	0.6874
POS	15	0.5645	0.5654
POS	30	0.5355	0.5202
POS	40	0.5122	0.5013
POS	50	0.5021	0.4825
POS	75	0.4526	0.4495
POS	100	0.4197	0.4337
BERTopic (Quantized with LLaMA 3 - 8B)	5	0.7397	0.7130
BERTopic (Quantized with LLaMA 3 - 8B)	15	0.7338	0.7602
BERTopic (Quantized with LLaMA 3 - 8B)	30	0.7787	0.7949
BERTopic (Quantized with LLaMA 3 - 8B)	40	0.7851	0.7955
BERTopic (Quantized with LLaMA 3 - 8B)	50	**0.7891**	**0.7969**
BERTopic (Quantized with LLaMA 3 - 8B)	75	0.7410	0.7452
BERTopic (Quantized with LLaMA 3 - 8B)	100	0.6878	0.7055

for interactive topic model visualization, allowing for an in-depth analysis of the topics and word clouds. One limitation of LDA is that it does not inherently provide a mechanism to rank topics based on certain criteria. Therefore, the pyLDAvis library came in hand for this analysis. Using the relevance metric slider, which adjusts the weight of term relevance based on the formula:

$$\text{Relevance}(w \mid t) = \lambda \cdot p(w \mid t) + (1 - \lambda) \cdot \frac{p(w \mid t)}{p(w)}$$

where $p(w \mid t)$ is the probability of term w given topic t, and $p(w)$ is the marginal probability of term w, allowed us to identify more relevant or more unique terms for each topic by varying λ from 0 to 1. Moreover, the saliency metric was used to find terms that are both frequent and specific to the topic, as depicted in the formula:

$$\text{Saliency}(w) = \text{frequency}(w) \cdot \left[\sum_t p(t \mid w) \cdot \log \left(\frac{p(t \mid w)}{p(t)} \right) \right]$$

where $p(t \mid w)$ is the probability of topic t given term w, and $p(t)$ is the marginal probability of topic t.

An important to note before showing the results is the impact of the high topic diversity in LDA, particularly when splitting into a larger number of topics. While the highest diversity was observed at 100 topics, this resulted in many of the initial topics being filled with data that is not particularly useful, such as common words like "would," "like," "use," "example," and "need." These simple words do not provide meaningful insights into the actual topics discussed on SO. This highlights a limitation of LDA. In the next presented Figures 17, 18 we have used the results from splitting in 5 topics

Table 8. Coherence Scores and Topic Diversity for 2021-2022

Period	Model	Num. Topics	Coherence Score	Topic Diversity
2021-2022	LDA	5	0.4183	0.72
	LDA	15	0.4765	0.8266
	LDA	30	0.5367	0.9166
	LDA	40	0.539	0.935
	LDA	50	0.5099	0.96
	LDA	75	0.4509	0.976
	LDA	100	0.4044	0.985
	BERTopic	5	0.379	0.375
	BERTopic	15	0.3761	0.214
	BERTopic	30	0.4141	0.3620
	BERTopic	40	0.4473	0.4538
	BERTopic	50	0.4785	0.5061
	BERTopic	75	0.5703	0.5783
	BERTopic	100	0.6286	0.6343
	KeyBERT	5	0.5548	0.95
	KeyBERT	15	0.7268	0.9571
	KeyBERT	30	0.7241	0.8793
	KeyBERT	40	0.7332	0.8435
	KeyBERT	50	0.7240	0.8346
	KeyBERT	75	0.7050	0.8040
	KeyBERT	100	0.6914	0.7686
	POS	5	0.4399	0.775
	POS	15	0.6812	0.8285
	POS	30	0.6902	0.7758
	POS	40	0.6914	0.7410
	POS	50	0.6893	0.7285
	POS	75	0.6797	0.6662
	POS	100	0.6725	0.6242
	BERTopic (Quantized with LLaMA 3 - 8B)	5	0.8354	0.8852
	BERTopic (Quantized with LLaMA 3 - 8B)	15	0.8431	0.8701
	BERTopic (Quantized with LLaMA 3 - 8B)	30	0.8996	0.8659
	BERTopic (Quantized with LLaMA 3 - 8B)	40	0.9162	0.8567
	BERTopic (Quantized with LLaMA 3 - 8B)	50	0.9264	0.8521
	BERTopic (Quantized with LLaMA 3 - 8B)	75	0.9061	0.8181
	BERTopic (Quantized with LLaMA 3 - 8B)	100	0.8981	0.7659

in order to see at a high level which topics were discussed in the 2 different periods and if there are any differences. Both pictures present the Topic 5 from both periods of time.

In 2021, the topics were diverse and covered various aspects of software development. Topic 1 was heavily focused on memory management, plotting, and CPU-related terms, indicating a strong emphasis on performance optimization and data visualization using tools like ggplot. Topic 2 was dominated by terms related to data manipulation in pandas, suggesting significant attention to data processing and analysis tasks. Topic 3 centered around service-oriented architecture with terms like request, service, and client, reflecting a focus on backend services, cloud integration, and API management. Topic 4 revolved around frontend development with terms such as component, button, and HTML, indicating a focus on user interface design and interactive elements. Finally, Topic 5 related to file and directory management, with terms like files, folder, and install, which suggests a focus on system-level scripting and automation tasks.

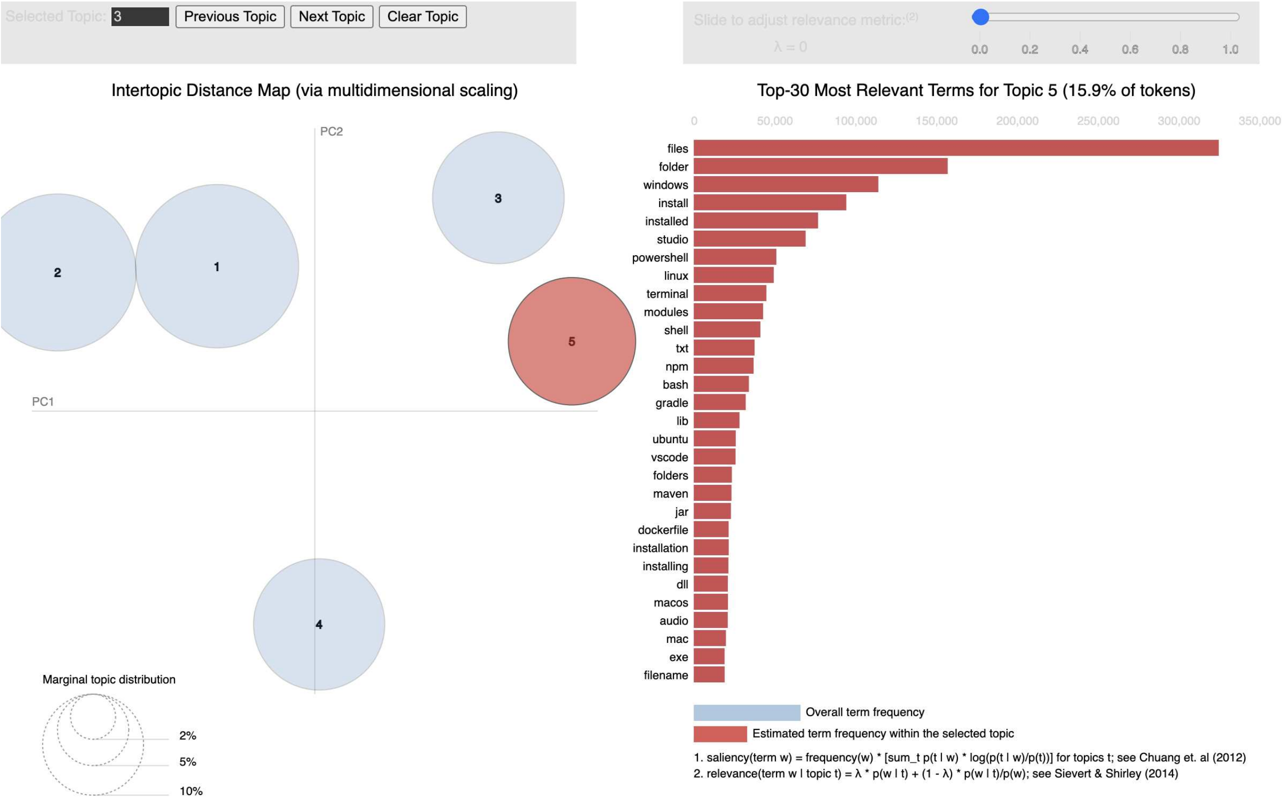
In 2024, the topics showed some shifts in focus areas while maintaining certain consistencies. Topic 1 continued to emphasize memory management and compiler-related tasks, indicating a shift towards understanding low-level programming concepts and optimizing code execution. Topic 2 focused on cloud services and deployment, with terms like azure, service, and folder, reflecting the increasing importance of cloud infrastructure and deployment practices. Topic 3 still emphasized data manipulation, similar to 2021, but with a greater focus on structured data and database operations. Topic 4 again focused on frontend development, but with a slightly different emphasis on components and navigation, suggesting evolving trends in web development frameworks. Lastly, Topic

Table 9. Coherence Scores and Topic Diversity for 2023-2024

Period	Model	Num. Topics	Coherence Score	Topic Diversity
2023-2024	LDA	5	0.3933	0.66
	LDA	15	0.4818	0.74
	LDA	30	0.52286	0.8866
	LDA	40	0.5179	0.9375
	LDA	50	0.5406	0.942
	LDA	75	0.46189	0.964
	LDA	100	0.4287	0.976
	BERTopic	5	0.3572	0.3666
	BERTopic	15	0.3829	0.2785
	BERTopic	30	0.4168	0.3965
	BERTopic	40	0.4342	0.4410
	BERTopic	50	0.4734	0.5102
	BERTopic	75	0.5571	0.6337
	BERTopic	100	0.6127	0.6565
	KeyBERT	5	0.7705	0.9613
	KeyBERT	15	0.6795	0.9571
	KeyBERT	30	0.6741	0.8827
	KeyBERT	40	0.6697	0.8666
	KeyBERT	50	0.6642	0.8571
	KeyBERT	75	0.6455	0.8081
	KeyBERT	100	0.6503	0.7919
	POS	5	0.7117	0.966
	POS	15	0.6596	0.8571
	POS	30	0.6617	0.7862
	POS	40	0.6721	0.7461
	POS	50	0.6626	0.7285
	POS	75	0.6711	0.6702
	POS	100	0.6688	0.6484
	BERTopic (Quantized with LLaMA 3 - 8B)	5	0.8451	0.8439
	BERTopic (Quantized with LLaMA 3 - 8B)	15	0.8579	0.8864
	BERTopic (Quantized with LLaMA 3 - 8B)	30	0.8901	0.8924
	BERTopic (Quantized with LLaMA 3 - 8B)	40	0.9087	0.8748
	BERTopic (Quantized with LLaMA 3 - 8B)	50	0.9123	0.8741
	BERTopic (Quantized with LLaMA 3 - 8B)	75	0.9132	0.8162
	BERTopic (Quantized with LLaMA 3 - 8B)	100	0.9028	0.7820

5 concentrated on graphical plotting and visualization, with terms like color, plot, and axis, indicating a continuous interest in data visualization but with more advanced graphical representations.Some key observations from this comparison include a noticeable increase in cloud and deployment-related topics in 2023-2024 compared to 2021-2022, reflecting the growing adoption of cloud infrastructure which later on is present also from the topics extracted with BERT and Llama. There is a consistency in the focus on frontend development, but the specific technologies and frameworks have evolved. Both periods show a strong focus on data-related topics, but the tools and techniques have become more advanced in 2024. The emphasis on service-oriented architecture and backend services persists, highlighting the ongoing importance of these areas in software development. In addition to using PyLDAvis as mentioned in the beginning we have also used word clouds. Word clouds provide a visual representation of the most frequent words within a topic, with the size of each word corresponding to its probability of occurrence within that topic being and easy method for visualisations. For instance, the word cloud generated for Topic 36 19 highlights terms such as "plot," "model," "graph," "axis," and "layers," which are prominently associated with data visualization and machine learning.

For visualizing topics extracted using BERTopic, including those labeled by KeyBERT, POS, and Llama-3, we utilized the library 'datamapplots' [61]. This library provides advanced visualization capabilities, enabling us to create detailed datamapplots that effectively display the clustering of topics within the document space. To make the visualizations interpretable, we reduced the high-dimensional embeddings into a 2D space. Datamapplots offers a variety of visualization techniques, allowing us to see not only the frequency and



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

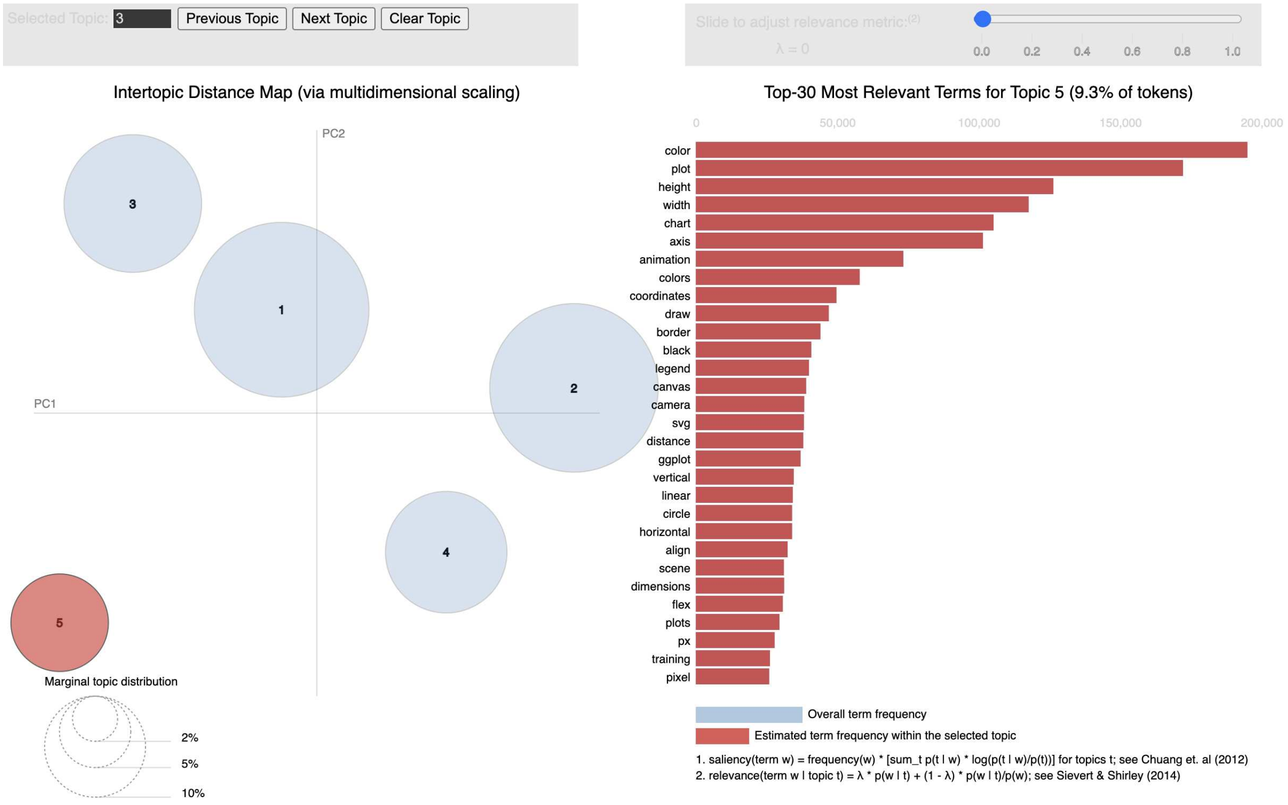
Fig. 17. pyLDAvis LDA topics 2021-2022

distribution of topics but also the relationships and overlaps between them. An example of a datamappplot is depicted in Figure 20.We're using the top 10 topic labels generated by the LLaMA model, but usually they are getting cutted as there are longer than 3-4 words. For a better visualisation of the topics, they can be checked in Table 10. The same datamapplot can be generated with custom labels, including those from KeyBERT, POS, or BERTopic, enhancing the clarity and relevance of the visualized topics as in Figure 21. In these visualizations, each cluster represents a topic, with points signifying individual documents. The clusters are color-coded and labeled based on the identified topics, enabling a clear visual differentiation of topics.

4.6.2 How do the top 10 topics identified in the two periods differ, and what might these differences indicate? To address the subquestion we have utilized the BERTopic method from the BERTopic API [17], specifically the Dynamic Topic Modeling (DTM) visualization feature to analyze and visualize topics over time. We have used the resulted BERTopic labels for the topics names inside the plots as there is no option in selecting the Llama repesentations, only if there are custom labels made before plotting. n the 2021-2022 period Figure 22, the most frequent topic was "dataframe_column_columns_data," starting with a frequency of around 4000 and gradually declining to approximately 3000. This topic maintained the highest frequency throughout the period, indicating consistent interest in data manipulation using dataframes, likely with Pandas in Python. Other prominent topics included "the_css_to_it," "table_query_column_the," and "component_react_the_type," each maintaining a frequency of around 3000 to 2000, indicating steady interest in CSS, SQL queries, and React components, respectively. Topics like "aws_to_the_docker" and "flutter_view_the_widget" also showed significant frequency, reflecting interest in cloud computing and mobile development.

For the 2023-2024 period Figure 23, the topic "dataframe_column_columns_pandas" remains highly frequent but with a lower starting frequency of around 1400, declining to about 800 which is in line with the number of questions selected for this analysis. This suggests a continued but reduced interest in data manipulation topics compared to the previous period. React and Flutter were popular in both periods, but the emphasis on React with hooks and advanced TypeScript features in 2023-2024 suggests evolving practices and deeper adoption of these technologies as seen also later the in topics created with Llama-3. Other key topics in this period include "pointer_template_type_function," "plot_axis_legend_ggplot," and "type_typescript_types_generic," each maintaining a frequency of around 400 to 200

4.6.3 How do the topic labels and discussions on SO look before and after the advent of ChatGPT, and what insights can be drawn from these comparisons? For answering the last research sub-question,



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Fig. 18. pyLDAvis LDA topics 2023-2024

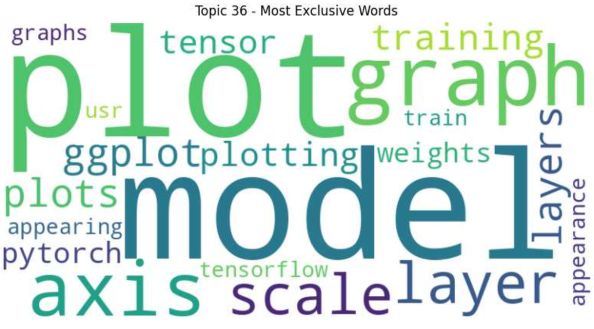


Fig. 19. Word Clouds

we have chosen to focus on 50 topics because the Topic Quality 7 was the highest for both periods of time, making it the most informative and representative sample for this analysis. By comparing the LLaMA representations in the next two tables 10 11, which are highly coherent we could have a get a better and deeper understanding of the real problems that exist and which the users of SO are seeking for help. We can observe new trends and technologies that gained prominence in the later period. For example, topics related to newer JavaScript frameworks (e.g., Vite, React with hooks) and advancements in machine learning frameworks (e.g., PyTorch) show that users started to shift their questions towards more modern development practices. Furthermore, topics such as "Terraform AWS," "Azure Functions," and "Docker Configuration and Troubleshooting" are more prevalent in 2023-2024, reflecting the growing importance of cloud infrastructure and DevOps practices. The growing discourse around these domains underscores a noteworthy transition towards the deployment and administration of programs in a scalable, effective, and automated manner, all of which are indispensable to contemporary software development and operations [10]. In terms of programming languages and frameworks, there is a noticeable increase in topics related to modern front end frameworks like React and Flutter. Additionally, topics on Python data manipulation, PyTorch model training, and SQL optimization highlight an increased focus on data-driven development and machine learning applications. This trend might show users interests towards the data science domain, reflecting the broader industry's movement towards leveraging data for insights, automation, and predictive capabilities. Discussions on Git, branch management, and continuous integration (CI) tools have become more prevalent, reflecting an increased emphasis on collaborative development and automation. There are also more detailed discussions on mobile development issues, particularly with Flutter and Android, in 2023-2024, indicating a deeper dive into mobile-specific challenges. The topics during 2021-2022, suggest that there was a significant focus on data manipulation with pandas, web development with JavaScript and React Native, SQL

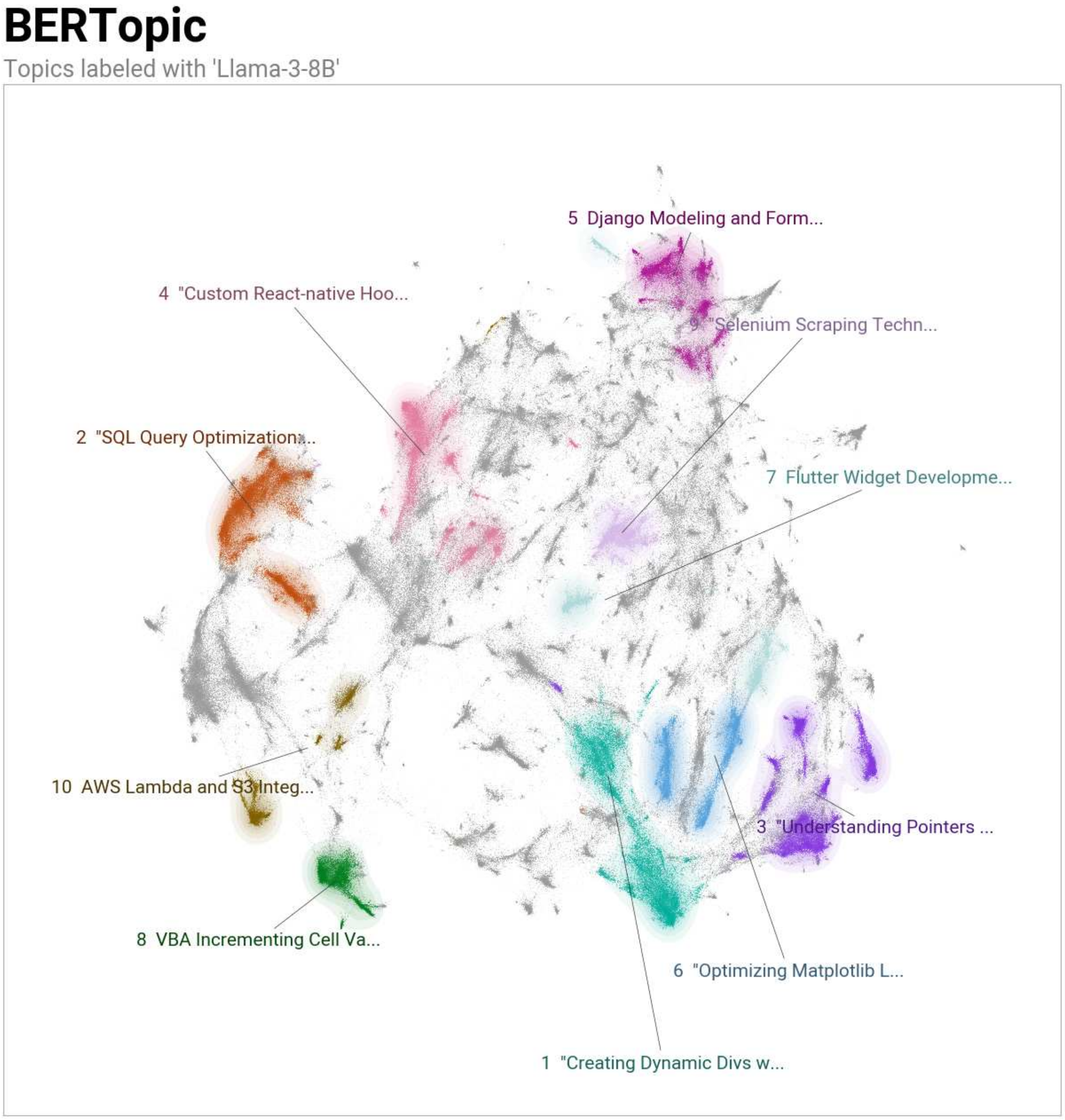


Fig. 20. Datamapplot Llama-3

optimization, web frameworks like Django, and cloud integration with AWS. While some of these topics remain relevant also in the second period studied, the prominence of specific technologies and frameworks in 2021-2022 indicates the trends and challenges faced by developers during that period of time.

5 DISCUSSION

The overall activity on SO has notably declined, with the number of questions posted per month dropping from approximately 135,000 to around 60,000, reflecting a broader reduction in user engagement. This decline is also evident in the usage of popular tags like Python, which saw its usage fall from over 18,000 to about 8,000. However, new tags such as R and Flutter entered the top 10 in the latter period, signaling a shift in developer interests towards emerging technologies, particularly those associated with data science and mobile development. Despite the overall drop in activity, the complexity and depth of discussions have increased, as evidenced by the longer average sentence length and more intricate syntactic structures. This suggests that while fewer questions are being posted, those that are posted tend to be more detailed and complex, possibly due

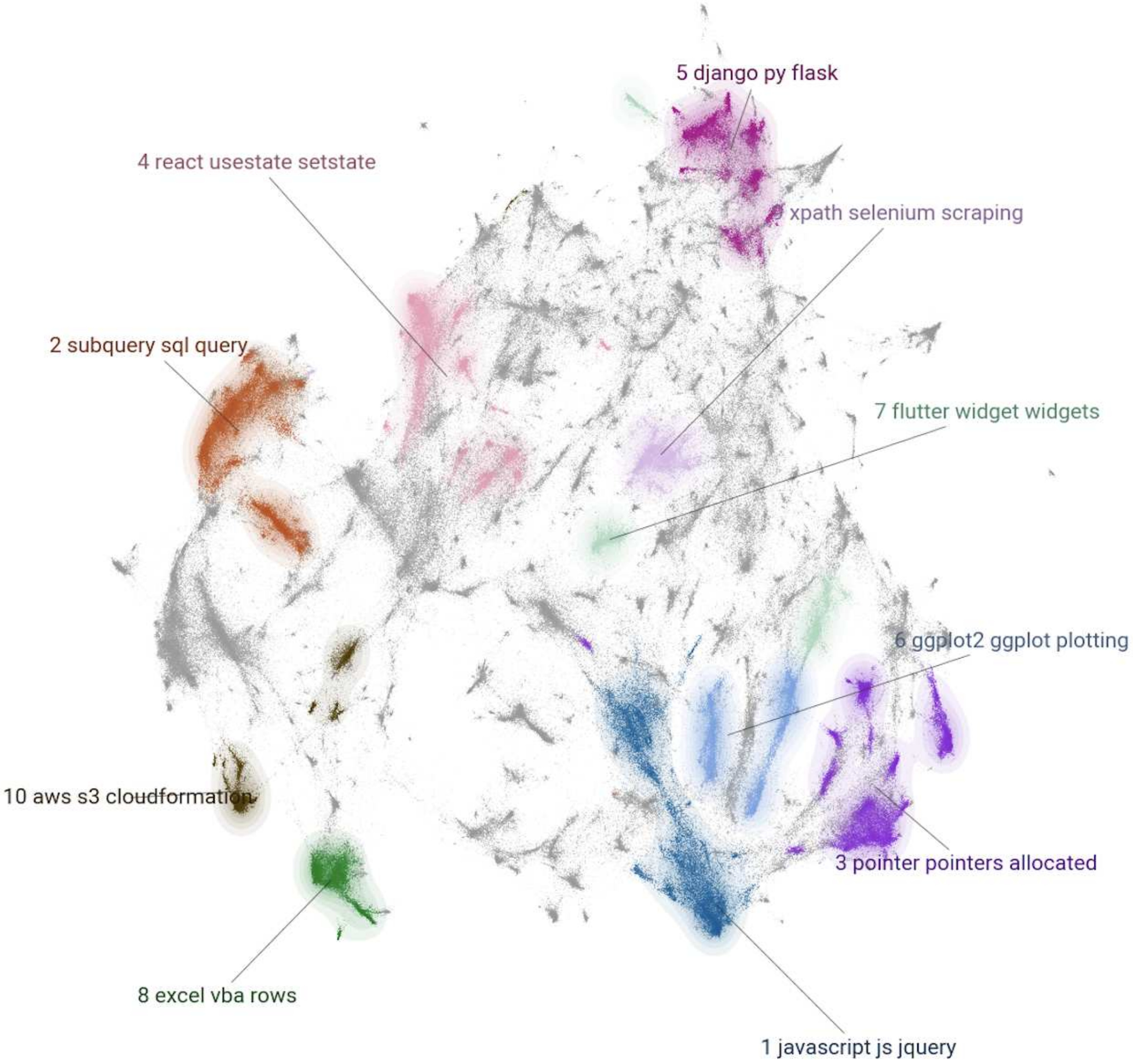


Fig. 21. Datamapplot KeyBERT

to the increasing sophistication of the technologies being discussed. The decline in user engagement is further underscored by the reduction in the number of answers per question and the longer response times for receiving accepted answers. The percentage of questions with accepted answers dropped from 42.3% to 28.2%, and the average view count per question decreased from about 1000 to below 200, indicating a significant decrease in community interaction. This trend might reflect a saturation of easily answerable questions and a shift towards more complex issues that require deeper expertise. The influence of AI technologies, particularly ChatGPT, has also become more pronounced, with tags related to AI and LLMs seeing solid increases. This reflects a growing interest in and reliance on AI tools, which is reshaping the types of questions asked on the platform. One interesting pattern in most of the created graphs is the drop between May and June 2023. In June 2023, tags related to LLMs saw growth Figure 15 , but the monthly answer rate Figure 9 and average view counts Figure 12 on SO dropped significantly. While these have dropped, the number of questions increased Figure 8. After a deeper search about what could have happened in those two months, ChatGPT's popularity peaked in May 2023, with 1.8 billion web visits, largely driven by its prominence during Google's developer conference, Google I/O [59]. Even though ChatGPT's popularity dipped in June and July [60], despite the release of its mobile apps on iOS and Android, it seems that users may have shifted their problem-solving habits, preferring to search for solutions on ChatGPT rather than SO. Meanwhile, in April 2023, SO began moderating its approach to AI tools, and by July, they launched OverflowAI, a generative AI in SO ecosystem to help developers and users access relevant information

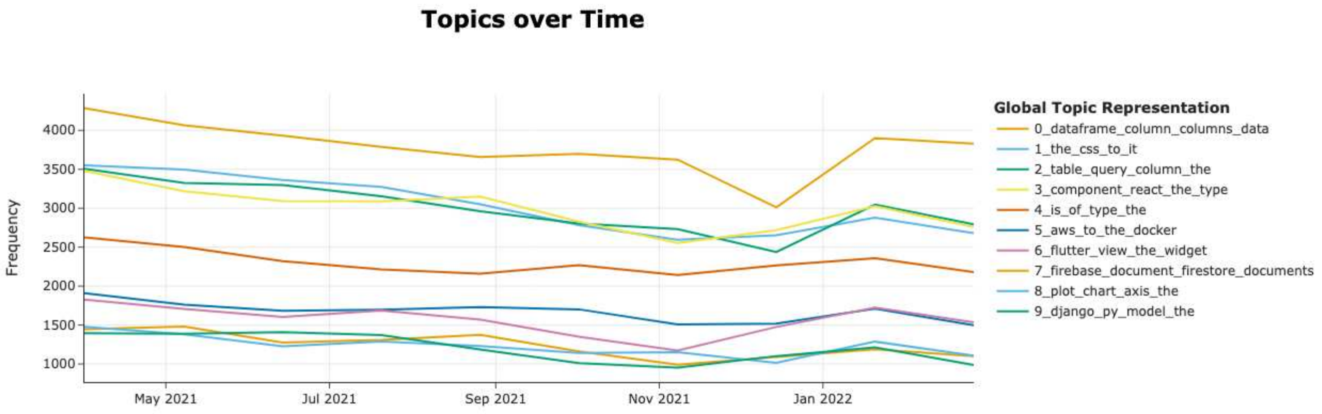


Fig. 22. Topics over time 2021-2022

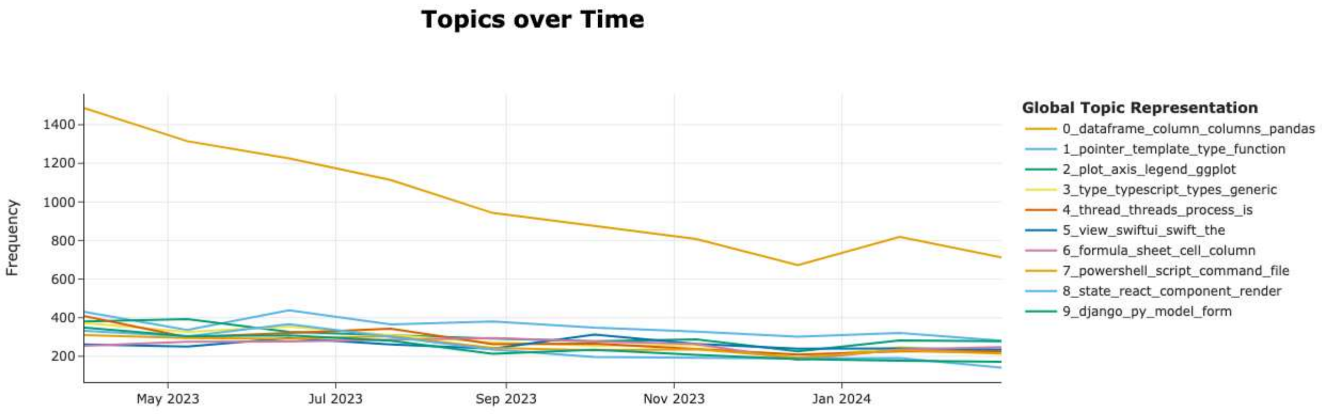


Fig. 23. Topics over time 2023-2024

more efficiently. This move acknowledged the declining influence of SO, as developers increasingly turned to ChatGPT, GitHub Copilot, and other generative AI tools. SO searches have dropped by 50% since early 2023, even with a temporary boost following the announcement of OverflowAI. The small boost in July can be seen in the case of number of average answers per question Figure 9 and let's say a stable/easily decreasing patter of number of questions per month Figure 8. The rise of AI-powered tools like ChatGPT has been a significant factor in the 35-50% traffic decline on SO stated also by [6]. Despite launching OverflowAI to stay competitive, it remains to be seen if it can regain traction in a world where generative AI is becoming the go-to resource for programming assistance.

The comparison of topic models between the two periods reveals a clear shift in the focus of discussions. In the earlier period, topics centered around data manipulation with tools like Pandas, web development with JavaScript and React Native, and SQL optimization. By 2023-2024, the focus had shifted towards cloud services, modern JavaScript frameworks like Vite, and machine learning with PyTorch, indicating the evolving priorities and challenges faced by developers. The highest quality topics were extracted using BERTopic quantized with LLaMA 3 - 8B, which provided a high Topic Quality.

6 FUTURE STEPS

Future research can enhance the methodology used in this thesis by focusing on several key areas. Preprocessing techniques could be refined to better capture the intricacies of technical discussions. Instead of removing all code snippets, selectively retaining relevant parts could improve the relevance of topic extraction, particularly for discussions where code is a crucial context. Preserving domain-specific terminology would also ensure that the analysis reflects the detailed nature of specialized fields like software development.

In the area of topic modeling, experimenting with other advanced LLMs or adjusting quantization methods could result in more precise topic identification. Fine-tuning models specifically for technical or domain-specific content would yield more reliable and focused topics. For example, reducing the minimum cluster size for HDBSCAN could enable the discovery of more granular topics, as currently the number of topics are limited compared to the actual number of topics. Increasing the number of topics (K) in LDA would provide a more detailed topic breakdown but requires careful consideration of computational demands.

Incorporating sentiment analysis alongside topic modeling, as discussed in the papers [52] and [27], could add another layer of understanding to user interactions. Analyzing both the sentiment behind questions and answers, as well as the topics being discussed, could provide deeper insights into community behavior and the nature of the conversations happening on platforms like Stack Overflow.

Improving visualization techniques is another area for future work. Current tools like pyLDAvis and word clouds are useful but can be expanded. Developing custom visualization tools or incorporating more interactive, advanced visualizations could offer a clearer view of topic evolution and relationships. This would provide a more intuitive way for users and platform administrators (SO) to explore trends and dynamics within large datasets, improving decision-making for content organization.

7 CONCLUSION

This thesis has explored the significant decline in user engagement on SO, alongside shifts in the nature of interactions on the platform. As SO struggles with a reduced user base and slower response times, the growing influence of LLMs, such as ChatGPT, has become evident. AI-driven tools have begun to alter how users seek information, often bypassing traditional Q&A forums like SO. This trend reflects a broader transformation in developer problem-solving approaches, where reliance on generative AI is becoming more prevalent. Through empirical analysis, it became clear that while user activity has decreased. The questions asked on SO now tend to focus more on advanced technologies, particularly in areas like AI, machine learning, and modern web development frameworks. Despite this shift towards more sophisticated content, the platform's existing methods for categorizing and tagging content have struggled to keep pace. This thesis introduced advanced topic modeling methods to offer a solution for more efficient and accurate content organization. These models have shown their potential in automating the discovery of emerging trends and categorizing vast amounts of data, which could greatly enhance SO's ability to serve its users. By adopting a high quality model like the BERTopic fined tuned with LLama-3, the platform can improve its search functionalities, refine its content tagging system by maybe automatically tagging new users' questions, and better adapt to evolving technologies and user behaviors.

To conclude, the thesis provides a foundation for how platforms like SO can address their current challenges. Implementing automated topic modeling and improving content organization can lead to a more dynamic and user-friendly experience. These approaches not only ensure that the platform remains relevant in the age of LLMs but also lay the groundwork for a more adaptive, future-proof system capable of maintaining high-quality knowledge sharing in a rapidly changing technological landscape.

REFERENCES

[1] Bitsandbytes.
[2] Chatgpt.
[3] Lda mallet.
[4] Quora.
[5] Reddit.
[6] The impact of chatgpt on stack overflow pros and cons, Feb 2024.
[7] Ames, M., and Naaman, M. Why we tag: motivations for annotation in mobile and online media. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2007), pp. 971–980.
[8] Asaduzzaman, M., Mashiyat, A. S., Roy, C. K., and Schneider, K. A. Answering questions about unanswered questions of stack overflow. In *2013 10th Working Conference on Mining Software Repositories (MSR)* (2013), pp. 97–100.
[9] Atwood, J., and Spolsky, J. Stack exchange api, 2008.
[10] Azad, N., and Hyrynsalmi, S. Devops critical success factors — a systematic literature review. *Information and Software Technology 157* (2023), 107150.
[11] Bajaj, K., Pattabiraman, K., and Mesbah, A. Mining questions asked by web developers. In *Proceedings of the 11th Working Conference on Mining Software Repositories* (New York, NY, USA, 2014), MSR 2014, Association for Computing Machinery, p. 112–121.
[12] Barua, A., Thomas, S. W., and Hassan, A. E. What are developers talking about? an analysis of topics and trends in stack overflow. 619–654.
[13] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners, 2020.
[14] del Rio-Chanona, M., Laurentsyeva, N., and Wachs, J. Are large language models a threat to digital public goods? evidence from activity on stack overflow, 2023.
[15] Dieng, A. B., Ruiz, F. J. R., and Blei, D. M. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics 8* (2020), 439–453.
[16] Grootendorst, M. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794* (2022).
[17] Grootendorst, M. P. Bertopic.
[18] Grootendorst, M. P. Tips tricks.
[19] Gupta, M., Li, R., Yin, Z., and Han, J. Survey on social tagging techniques. *SIGKDD Explor. Newsl. 12*, 1 (nov 2010), 58–72.
[20] He, J. Uva, May 2011.
[21] Heitz, L., Rozgonyi, K., and Kostic, B. *AI in Content Curation and Media Pluralism.* 01 2021.
[22] Hess, C. W., Ritchie, K. P., and Landry, R. G. The type-token ratio and vocabulary performance. *Psychological Reports 55*, 1 (1984), 51–57.
[23] Jarvis, S. Capturing the diversity in lexical diversity. *Language learning 63* (2013), 87–106.
[24] Jeff Atwood, J. S. Stack overflow.
[25] Jelodar, H., Wang, Y., Yuan, C., and Feng, X. Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey.
[26] Jung, H. S., Lee, H., Woo, Y. S., Baek, S. Y., and Kim, J. H. Expansive data, extensive model: Investigating discussion topics around llm through unsupervised machine learning in academic papers and news. *PLOS ONE 19* (05 2024), 1–18.
[27] Kabir, S., Udo-Imeh, D. N., Kou, B., and Zhang, T. Is stack overflow obsolete? an empirical study of the characteristics of chatgpt answers to stack overflow questions, Feb 2024.
[28] Khan, A. A., Khan, J. A., Akbar, M. A., Zhou, P., and Fahmideh, M. Insights into software development approaches: Mining qa repositories, 2023.
[29] Kochhar, P. S. Mining testing questions on stack overflow. pp. 32–38.
[30] Liukko, V., Knappe, A., Anttila, T., Hakala, J., Ketola, J., Lahtinen, D., Poranen, T., Ritala, T.-M., Setälä, M., Hämäläinen, H., et al. Chatgpt as a full-stack web developer. In *Generative AI for Effective Software Development.* Springer, 2024, pp. 197–215.
[31] Mamykina, L., Manoim, B., Mittal, M., Hripcsak, G., and Hartmann, B. Design lessons from the fastest qa site in the west. pp. 2857–2866.
[32] Mann, H. B. Nonparametric tests against trend. *Econometrica 13* (1945), 245.
[33] Meldrum, S., Licorish, S. A., and Savarimuthu, B. T. R. Exploring research interest in stack overflow – a systematic mapping study and quality evaluation, 2020.

[34] Naghshzan, A., and Ratte, S. Enhancing api documentation through bertopic modeling and summarization, Aug 2023.

[35] Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., and Mian, A. A comprehensive overview of large language models, 2024.

[36] Nielsen, A. Nielsen, a.p.: Understanding dynamic capabilities through knowledge management. journal of knowledge management 10(4), 59-71. *J. Knowledge Management 10* (07 2006), 59–71.

[37] Nokel, M., and Loukachevitch, N. Accounting ngrams and multi-word terms can improve topic models.

[38] of York, U. Centre for reviews and dissemination.

[39] Papoutsoglou, M., Kapitsaki, G. M., German, D., and Angelis, L. An analysis of open source software licensing questions in stack exchange sites, 2021.

[40] Porter, M. F. *An algorithm for suffix stripping*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997, p. 313–316.

[41] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners.

[42] Ray, P. P. Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems 3* (2023), 121–154.

[43] Richards, B. Type/token ratios: What do they really tell us? *Journal of child language 14*, 2 (1987), 201–209.

[44] Rungruangjit, W., and Charoenpornpanichkul, K. What motivates consumers' continued usage intentions of food delivery applications in post-covid-19 outbreak? comparing generations x, y and z. *Journal of Asia Business Studies 18* (11 2023).

[45] Shan, G., and Qiu, L. Examining the impact of generative ai on users' voluntary knowledge contribution: Evidence from a natural experiment on stack overflow, May 2023.

[46] Sievert, C., and Shirley, K. E. Ldavis: A method for visualizing and interpreting topics, Jun 2014.

[47] Silva, L. D., Samhi, J., and Khomh, F. Chatgpt vs llama: Impact, reliability, and challenges in stack overflow discussions, 2024.

[48] Singh, S., and Kotian, R. Is stack overflow overflowing with questions and tags.

[49] Son, J., and Kim, B. Trend analysis of large language models through a developer community: A focus on stack overflow. *Information 14*, 11 (2023).

[50] Stephen W., T. Mining software repositories with topic models - school of ..., Feb 2012.

[51] Sulír, M., and Regeci, M. Software engineers' questions and answers on stack exchange. In *2022 IEEE 16th International Scientific Conference on Informatics (Informatics)* (2022), pp. 304–310.

[52] Sulír, M., and Regeci, M. Software engineers' questions and answers on stack exchange | ieee conference publication | ieee xplore, Jun 2023.

[53] Tahmooresi, H., Heydarnoori, A., and Aghamohammadi, A. An analysis of python's topics, trends, and technologies through mining stack overflow discussions, 2020.

[54] Tahmooresi, H., Heydarnoori, A., and Aghamohammadi, A. An analysis of python's topics, trends, and technologies through mining stack overflow discussions, Apr 2020.

[55] Tan, C.-M., Wang, Y., and Lee, C.-D. The use of bigrams to enhance text categorization. *Information Processing Management 38* (07 2002), 529–546.

[56] Thomas, S. W. Mining software repositories using topic models. In *2011 33rd International Conference on Software Engineering (ICSE)* (2011), pp. 1138–1139.

[57] Torene, S., Follmann, A., Teague, T., Chang, P., and Howald, B. Automated hashtag hierarchy generation using community detection and the shannon diversity index, with applications to twitter and parler. *International Journal of Semantic Computing 16*, 04 (2022), 473–496.

[58] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. Llama: Open and efficient foundation language models, 2023.

[59] Trenholm, R. Chatgpt's popularity drops, say new figures, Feb 2024.

[60] Trock, D. Chatgpt is declining in popularity: Here's what's taking its place, Jul 2023.

[61] TutteInstitute. Tutteinstitute/datamapplot: Creating beautiful plots of data maps.

[62] Vaghela, D. Social media impact on website ranking.

[63] Wang, H., Prakash, N., Hoang, N. K., Hee, M. S., Naseem, U., and Lee, R. K.-W. Prompting large language models for topic modeling, 2023.

[64] Wikipedia contributors. Jaccard index — Wikipedia, the free encyclopedia, 2024. [Online; accessed 14-July-2024].

[65] Xu, M., Fralick, D., Zheng, J. Z., Wang, B., Tu, X. M., and Feng, C. The differences and similarities between two-sample t-test and paired t-test, Jun 2017.

[66] Xue, J., Wang, L., Zheng, J., Li, Y., and Tan, Y. Can chatgpt kill user-generated qa platforms?, May 2023.

[67] Zhang, B., Liang, P., Zhou, X., Ahmad, A., and Waseem, M. Practices and challenges of using github copilot: An empirical study.

[68] Zhao, Z., Chen, Y., Bangash, A. A., Adams, B., and Hassan, A. E. An empirical study of challenges in machine learning asset management. *Empirical Software Engineering 29*, 4 (June 2024).

[69] Zou, J., Xu, L., Guo, W., Yan, M., Yang, D., and Zhang, X. An empirical study on stack overflow using topic analysis. In *Proceedings of the 12th Working Conference on Mining Software Repositories* (2015), MSR '15, IEEE Press, p. 446–449.

Table 10. LLaMA Topics for 2021-2022

Topic Labels
Dataframe Reshaping Unique Column Elements Row Selection Pandas MatchingFiltering and Efficient Updates
Creating Dynamic Divs with JavaScript and HidingShowing Input Fields through Button Clicks
SQL Query Optimization Combining Two Select Statements and Handling NULL Values
Understanding Pointers to 2D Arrays in C
Custom Reactnative Hooks and Initial State Issues
Django Modeling and Form Creation
Optimizing Matplotlib Line Plots with Comma Separated Values and Shared Legend
Flutter Widget Development Changing Text Using Lists Random Background Images Understanding GlobalKeys and Form Validation
VBA Incrementing Cell Value on Paste and Finding Last Row with Specific Criteria in Excel
Selenium Scraping Techniques and Challenges
AWS Lambda and S3 Integration Issues and Solutions
Laravel Resource Controller with Model Class and Implicit Binding
Training KerasPyTorch models with custom validation strategies and saving weights for future use
Managing State and Lifecycle in SwiftUI Handling Reference Types and Updating Views
Vue Component Mutation and Alternatives
Firebase Connection Error and User Collection Creation in Flutter with Security Rules
Powershell Troubleshooting How do you want to open this file Error with SelectString and Windows 11 Update
Discord Bot Error Handling and Command Implementation
Passing Data between Recycler View Button and Fragment in Kotlin
Understanding TypeScripts Union vs Intersection Resolution in Conditional Types
MongoDB Query Techniques Excluding Documents Insertion with Upsert Finding Duplicates Replacing Null Values and Aggregation Examples
NPM JS Webpack Node Package Module Management Angular Build Error Node Version Conflict Global Update Reinstallation ESLint
TS JSON Modules Run My App This File In To And
Regex Patterns for String Manipulation and Matching
JavaScript Array Object Manipulation Matching IDs Filtering Converting
Python package installation troubleshooting and best practices for data science and scientific computing environments using Conda and virtual environments
Android Gradle Plugin Upgrade Required for Flutter Build Errors
Merging Audio Clips with Video using AVMutableComposition
CMake Library Project with Optional Executable Structuring and Building
Troubleshooting HTTP Requests in ASPNET Core Web API with Postman and Swagger
WPF Data Binding to User Control Error Converting Binding to String
Numpy Array Manipulation Techniques
Docker Image Building and Running Issues
Kubernetes Horizontal Scaling with Multiple Pods on Same Node
Customizing Chartjs Diagrams Parsing JSON Data Triangle Shapes User Input Grouped Data and Adding Horizontal Lines
Azure Blob Storage Triggered Copy to ADLSgen2 Data Lake with Path Extraction via Python Function or Data Factory
PySpark Column Manipulation Techniques
Generic Method Parameter Bivariance in Kotlin and Type Inference from Implementing Interfaces
Git branch management and commit tracking
JSON parsing and mapping in various languages
Pygame Collision and Camera Movement for Game Development
Azure AD B2C User Flow Authentication with Claims and Tokens
Troubleshooting Tkinter Window Issues Freezing No Widgets and Error Messages
Creating a Standalone Java Executable Without an Installer and JRE
Promises and Asynchronous Functionality in JavaScript
Python Class Variables and Decorators for Reusability
VS Code Preventing Outside File Memory Configuring Context Menu Behavior
Rust Lifetime Confusion and Type Erasure in Implementing TicTacToe Board Functionality
SSL Certificate Configuration forNET 5 API with External Certificate
Angular Form Component Validation with Builtin Directives and Dynamic Input Fields

Table 11. LLaMA Topics for 2023-2024

Topic Labels
Function to select first nonNA value from variable subset in R
C Pointer Concepts and Behavior Analysis
Creating customized bar plots with ggplot2 and labeling axes and legends
TypeScript Generics and Correlated Unions for Stronger Type Safety
Optimizing Multiple Thread Pools for CPUBound Tasks and RealTime Requirements
Optimizing SwiftUI List View Reusability and Passing Data Between View Models in SwiftUI
Excel Dynamic Formulas for Filtering and Searching Large Datasets
Optimizing Class Sharing in PowerShell Scripts
Optimizing React Component Rendering with State Variables and Hooks
Optimizing Django File Downloads and Handling Form Submissions Securely
Efficiently finding session durations using SQL partitioning
Efficient Array Manipulation with NumPy and SymPy Constraints
Regex Matching and String Manipulation Techniques for Log Analysis and Data Parsing
Terraform AWS Setting up Assume Role for Multiple Accounts and Creating AWS Parallel Cluster Instances
Flexbox Height and Width Adjustments for Responsive Design
Pytorch Model Training and Evaluation with Flexible Time Axis Data
NPM Vite Typescript Webpack ImportExport Confusion
Python Package Installation in Virtual Environments and Resolving Import Errors
FirebaseFirestore Testing with Large Data Sets and Authentication Rules
Conditional Setting of Include Paths in CMake for CrossPlatform Builds
Java Generics Issue with Eitherconverge Method
Selenium Element Locating Issues and Scraping Techniques
Flutter Layout Issues Ignoring Layout on Bottom Nav Bar Wrapping Multiple Chips AppBar Leading Unwanted Margins in Columns
Laravel Error Unable to Update Data in Controller and Redirect to Dashboard Without Showing Errors
Flutter Async Calls and Refreshing State with Riverpod Provider
Kafka Consumer Corruption during Restart
Understanding and Handling UTC Information in Datetime Formats and Pandas Functions
Implementing generic traits with lifetime issues and avoiding explicit type annotations in Rust
Python Dictionary Manipulation with Loops and List Operations
MVVM Implementation and Binding Issues inNET MAUI and WPF
Managing Async Data and State Logic in Vuejs with Pinia and Composables
JavaScript Saving Checkbox Status and Rendering Persistent State
Java 17 vs 11 dependency conflict in Spring Boot project with Tomcat 9
MongoDB Optimizing Array Updates and Projections
Customizing Animations and Updating Data Points in Chartjs Line Charts
Merging Namespaced XML Files with XSLT 20
Docker Configuration and Troubleshooting
Blazor Component Passing Objects vs Value Types Debugging Null Parameters
Unity Keeping Enemy Continuously Shooting Towards Moving Player Using SpriteKit
Discord Bot Message Handling and Integration with Telegram
VS Code Customizing Syntax Highlighting for Embedded Languages
Git Branch Merge Conflicts and Unstaged Changes
Azure Functions Configuring Logging and Monitoring
Array Filtering and Mapping Objects in JavaScript
Swagger Configuration for Polymorphic Request Bodies in ASPNET Core Web API
Combining Selenium and Tkinter Windows A Guide
Python Type Annotations for Function Returning Class Decorator and Metaclass Singleton Pattern
JSON Deserialization with Custom Default Values and Handling Missing Properties
Resolving Compatibility Issues betweenNET Framework andNET Core Projects