Accessibility Challenges Mobile App Developers Encounter on Stack Overflow

Amila Indika · Christopher Lee · Haochen Wang · Justin Lisoway

Abstract—**This study explores the accessibility design challenges faced by mobile developers by analyzing Stack Overflow posts related to the accessibility design of Android and iOS applications. The investigation focuses on three key aspects: factors influencing the growth of mobile accessibility questions, characteristics of such questions, and the challenges inherent in mobile accessibility development. The methods of this study involve exploring the growth in mobile accessibility questions through a per-year analysis, and various statistical measures are employed to delineate their characteristics. Topic modeling techniques, specifically Latent Dirichlet Allocation, were utilized to identify developers' challenges. Findings revealed a substantial increase in accessibility questions in the early 2010s, potentially attributed to the introduction of laws and regulations, alongside the emergence of new accessibility features like voiceover and talkback. Furthermore, data analysis indicates that accessibility questions typically receive responses within two days and tend to outnumber corresponding answers. Additionally, integrating new accessibility features into mobile applications, particularly in user interface design and the quest for practical solutions to mobile accessibility issues, was identified as a core challenge mobile developers encounter.**

I. Introduction

Mobile devices are ubiquitous in modern life. Approximately seven billion people adorn their pockets with a mobile device for vital digital applications (apps) and services [1]. The proliferation of smartphones has expanded the scope of mobile phone usage beyond traditional communication, with apps now being extensively utilized in diverse sectors such as finance, food, travel, health & fitness, retail, and entertainment [2]. The extensive utilization of smartphones has resulted in billions of hours of app usage and a surge in app downloads across these domains [2]. Thus, given the multifaceted utility of smartphones across various domains, it is essential to recognize that individuals with a wide range of abilities use these devices.

According to the World Health Organization (WHO), approximately 1.3 billion individuals, constituting roughly 16% of the global population, grapple with significant disabilities [3]. Consequently, given the widespread adoption of smart mobile devices and the substantial percentage of disabled individuals, the imperative arises to create accessible mobile apps that extend services to people with disabilities. Hence, there is a significant body of medical and technological research dedicated to exploring the use of mobile devices by individuals with disabilities, including those with visual impairments [4][5], hearing impairments [6], motor disabilities [4][7], senior citizens [8], and individuals with other forms of disabilities [9], among others.

Moreover, a substantial body of research in the realm of mobile accessibility has focused on the analysis of real-world mobile applications, aiming to identify accessibility issues [10–13] and assess compliance with mobile accessibility guidelines [14]. The emphasis on analyzing real-world mobile applications underscores the significance of mobile accessibility as a prominent concern. However, it is noteworthy that many of these research studies are confined to examining Android apps. Furthermore, studies have endeavored to create taxonomies or guidelines that offer a broader perspective on the mobile accessibility landscape through the analysis of mobile applications [15–17]. Nevertheless, some of these efforts are characterized by incompleteness [16], a lack of recent updates [17], or reliance on smaller-scale analyses [15].

A. Motivation

Despite the presence of numerous laws and regulations in the United States [18][19], as well as global initiatives [19], and accessibility guidelines such as the Web Content Accessibility Guidelines (WCAG) [20][21], Android accessibility guidelines [22], iOS accessibility guidelines [23] to guarantee accessibility for mobile apps, mobile accessibility issues are evident to persist [11, 24, 17, 13, 25]. Additionally, researchers have identified a need for more awareness among mobile developers and technical challenges as significant factors contributing to the inaccessibility of mobile applications [14][13]. *Therefore, this study seeks to examine the challenges faced by mobile app developers in developing mobile accessibility from their perspective.*

Determining the challenges encountered by mobile app developers in developing accessible mobile apps will be achieved by analyzing accessibility issues discussed on the software question-and-answer platform Stack Overflow. A survey conducted in 2023, encompassing over 90,000 professional software developers, revealed that Stack Overflow [26] is utilized by over 98% of developers surveyed [27]. The survey underscores Stack Overflow's status as one of the software development community's most widely employed question-and-answer platforms. As such, Stack Overflow will be used based on the plethora of data, approximately 24 million questions [28] and 35 million answers, totaling 59 million questions and answers [26].

The scope of this study's specific inquiry into accessibility focuses on Android and iOS, stemming from the significance of these systems. In this context, the dominant market share of 99.3% collectively held by the mobile operating systems Android and iOS [29] underscores the importance of this study.

Both respective app stores share a volume of over 5 million apps [30]. Therefore, the focus of this study will exclude other mobile operating systems due to the lack of a mature developer community, low adoption rates, or discontinuation of said operating systems.

While previous research has delved into Stack Overflow mining for mobile accessibility [24, 31, 32], it is essential to note that some of these studies are constrained to the Android platform [24][31]. Others focus on broader hot topics instead of explicitly targeting mobile accessibility[32], and a few have limited their dataset to less than 1,000 Stack Overflow discussion threads, which is relatively small [24]. Consequently, a notable research gap exists in the literature concerning large-scale Stack Overflow analysis aimed at understanding the challenges developers face in mobile accessibility, encompassing both Android and iOS mobile apps. *As a result, this study seeks to address this gap by conducting extensive large-scale Stack Overflow mining to shed light on the overarching challenges related to mobile accessibility for Android and iOS.*

B. Goals & Research Questions (RQs)

The goal of this study is to investigate the challenges mobile app developers face in developing accessible apps to catalyze solutions for those challenges by providing analysis. To accomplish the goal, the following research questions are proposed. The initial research question involves quantitative analysis, the second involves quantitative and qualitative, and the third involves qualitative analysis.

> **RQ1:** What are the attributes related to the growth of mobile accessibility questions?

The first research question aims to understand better the evolution of mobile accessibility questions and answers over time. This inquiry provides valuable insights into the engagement of active mobile developers, the promptness of responses to questions, and the growth trajectory of popular accessibility features.

Quantitative aspects will be analyzed temporally. Aspects include the growth of specific tags, changes in the number of questions and answers, changes in average response time, and improvements, stagnation, or deterioration in growth. To accomplish this, an investigation into Stack Overflow's mobile accessibility-related posts yearly will be conducted.

> **RQ2:** What are the characteristics of mobile accessibility questions?

The second research question seeks to uncover the characteristics of mobile accessibility questions posed by the community. The method of answering this question entails an analysis of various aspects, including the volume of questions asked and answered, prevalent tags, and the level of developer interaction with the questions.

The statistical characteristics in question involve quantitative analysis of the following: popular topics, number of tags, number of questions, number of answers, number of comments, response time, frequency of answered questions, and ask-answer ratio.

> **RQ3:** What are the challenges associated with mobile accessibility development?

The third research question offers an opportunity to categorize and comprehend the diverse array of mobile accessibility challenges developers discuss. It also allows insights into identifying the most formidable mobile accessibility challenges within these categories and their corresponding answer rates.

The third research question will be answered by sorting potential topics via the Latent Dirichlet Allocation (LDA) [33] technique. After that, a statistically significant representative sub-sample of topics will be taken to verify its validity. Furthermore, another statistically significant sub-sample of popular tags will be taken to investigate potential patterns in the posts.

C. Contribution

To our knowledge, there is a lack of focus on analyzing mobile accessibility questions in Stack Overflow. However, there are a few preexisting works in this domain for which to verify and improve. The sheer volume of this study's analyzed data should sufficiently distinguish it from similar papers in this area of research. Furthermore, the initial literature review revealed that numerous studies have concentrated on mobile accessibility concerning Android apps. However, there exists a noticeable gap in the research landscape regarding mobile accessibility, encompassing both Android and iOS apps. Hence, this study would fill that gap.

II. Related Works

There are multiple related studies preceding this research. The following related works provide a more insightful understanding of this research field. The table in Appendix A summarizes each related work and their respective research questions.

A. Mobile Accessibility

*1) **Mobile application analysis**:* In their study, Ross et al. [11] examined a data set consisting of 9,999 free Android apps, focusing on analyzing them with seven distinct accessibility barriers that impact individuals with impairments. While their research contributes valuable insights into prevalent accessibility issues within Android apps, it is essential to note that the data set used in their study was not initially collected for accessibility analysis. Furthermore, their research needs a comparative analysis across various mobile operating systems like iOS and Android. In contrast, this study takes a broader approach by investigating the landscape of mobile accessibility across multiple operating systems, encompassing both iOS and

Android. It is worth mentioning that this study's methodology differs significantly from that of Ross and colleagues. Rather than conducting a direct analysis of mobile apps to assess accessibility issues, this study's approach involves an exploration of discussion posts related to mobile accessibility on Stack Overflow. This exploration enables insights into the challenges faced by developers and the community in the context of mobile accessibility.

Another study by Ross et al. [10] assesses 5,753 Android mobile apps to identify prevalent mobile accessibility issues. Their study highlighted the widespread presence of accessibility barriers, particularly those related to labels and images. However, it is essential to note that their research focuses on a specific subset of inaccessibility barriers. In contrast, this study's investigation takes a broader approach, encompassing a more comprehensive examination of mobile accessibility issues.

Yan and Ramachandran [12] examined 479 Android apps, employing the IBM Mobile Accessibility Checker (MAC) to identify accessibility issues within their Graphical User Interfaces (GUI). Their study introduced coverage metrics designed to assess the effectiveness of automated tools in detecting accessibility issues. Their primary focus lies in the domain of automated tools for evaluating mobile accessibility, a distinction from this study's direction. Furthermore, it is essential to note that their study focuses explicitly on GUI-related challenges in mobile accessibility. In contrast, this study takes a more comprehensive approach, encompassing a broader examination of mobile accessibility issues across various facets, extending beyond GUI-specific concerns.

Di Gregorio et al. [14] asserts that despite the growing interest in mobile accessibility, developers lack adequate mechanisms to tackle these issues. Their empirical study delves into the extent to which existing mobile accessibility standards and guidelines are put into practice by developers. A survey follows the empirical study that gauges the perspective of actual mobile app developers about mobile accessibility. Their research findings highlight a deficiency in practical guidelines for implementing universal design, regardless of the existence of mobile accessibility guidelines. Furthermore, they conclude that developers lack awareness regarding several types of disabilities and grapple with technical challenges when incorporating accessibility features into mobile apps. It is worth noting that this study differs from theirs in several vital aspects. While their focus is solely on Android apps, this study extends to encompass mobile accessibility for both Android and iOS platforms. Additionally, compared to this study, they did not employ data mining on Stack Overflow. Instead, they adopted a coding strategy to evaluate the implementation of mobile accessibility in Android apps.

Acosta-Vargas et al. [34] highlights a notable gap in awareness among mobile developers regarding accessibility issues. Their approach involved subjecting the top 10 most popular mobile apps, as per PCMAG[1], to an evaluation based on the Web Content Accessibility Guidelines 2.1 (WCAG 2.1). Through a combination of manual and automated reviews, they demonstrated instances of non-compliance with the recommended accessibility standards set forth by the World Wide Web Consortium. It is crucial to note that their research differs from this study. Unlike this study's direction, which does not involve the review of actual mobile apps, Acosta-Vargas and his team conducted a detailed examination of these apps to assess their adherence to accessibility guidelines.

Alshayban et al. [13] conducted a comprehensive study examining over 1000 Android apps to assess the prevalence of 11 distinct accessibility issues. Additionally, they administered an online survey to Android developers to gain insights into their perceptions regarding mobile accessibility. Their research reveals a significant challenge disabled users face in identifying suitable accessibility-supported apps among popular apps with high ratings. This difficulty arises due to the relatively small proportion of disabled users. The study's findings point to the creation of a substantial number of accessibility issues attributed to developers' lack of awareness, additional costs associated with ensuring accessibility, and insufficient support from management. However, compared with their research, this study does not feature an analysis of mobile apps, highlighting differing focuses and objectives.

Summary: Studies in this section have primarily concentrated on examining real-world mobile applications to identify accessibility barriers. Notably, many studies have focused exclusively on Android app accessibility. In comparison, this study approach does not involve the direct analysis of real mobile applications. Instead, the insights into the landscape of mobile accessibility are acquired by mining Stack Overflow discussions.

*2) **Automated accessibility detection and repair tools**:* Alotaibi et al. [35] examined the automation of repairs for size-based mobile accessibility issues. It is important to note that this study differs significantly from theirs. Unlike Alotaibi and colleagues, this study does not incorporate any refactoring or automated repair processes for the mobile accessibility issues identified. Instead, this study's primary objective summarizes the prevailing mobile accessibility challenges developers encounter.

Chen et al. [36] introduced Xbot, an automated accessibility testing tool designed for Android apps, featuring User Interface exploration capabilities. Their study included experiments involving 2,270 Android apps, revealing that approximately 89% of these apps exhibited accessibility issues when assessed with the Xbot tool. It is important to note that Xbot's scope is limited to Android apps and does not offer automated repair functionalities. Additionally, the tool's findings are specific to Android apps and do not offer a comprehensive overview of mobile accessibility in general.

Eler et al. [37] introduced the MATE automated accessibility testing tool for Android, designed to automatically generate tests and employ dynamic code analysis to uncover mobile accessibility issues. Their research showcased the superior performance of MATE in comparison to static checkers such as Android Link [38], as well as testing frameworks like Espresso [39] and Roblectric [40], particularly in the

[1]https://www.pcmag.com/article/362295/the-100-best-android-apps

realm of detecting mobile accessibility issues. While their study contributes to automated test generation for mobile accessibility, it is essential to note that this study takes a distinct path. Unlike Eler and colleagues, this study does not delve into automated test generation for mobile accessibility.

Summary: The related work discussed in this section primarily revolved around automated test generation, automated detection, and refactoring of mobile accessibility issues. Notably, many of these studies focused on automating accessibility solutions for Android applications. In contrast, this study does not explore accessibility automation or detection. Instead, this study focuses on exploring the challenges articulated by mobile developers regarding mobile accessibility.

*3) **Evaluation of accessibility detection tools**:* Mateus et al. [41] systematic mapping study delves into the realm of accessibility evaluation methods prevalent in both web and mobile apps. Their primary focus is examining mobile accessibility evaluation techniques, encompassing automated tools, user evaluations, and expert inspections. Notably, their study does not center on exploring actual mobile accessibility issues. In contrast, this study pursues a distinct path. Rather than assessing evaluation methods, this study focuses on directly investigating and summarizing mobile accessibility challenges. Hence, a notable divergence exists between their research and this study's focus and objectives.

Silva et al. [25] surveyed the landscape of automatic accessibility evaluation tools spanning Android, iOS, and Windows mobile operating systems. Their study offers an exhaustive compilation of these automated tools. It highlights a critical finding: collectively, these tools address only approximately 13% of the accessibility guidelines outlined by the British Broadcasting Corporation (BBC) and the World Wide Web Consortium (W3C). While their research aligns with the exploration of automated mobile accessibility evaluation tools, it is crucial to emphasize that this study pursues a distinctly different objective and research direction.

Summary: Research endeavors in this area have centered on mobile accessibility evaluation and associated tool development. In contrast, this study takes a distinct path by delving into developers' perspectives and challenges concerning mobile accessibility.

*4) **Taxonomy of accessibility issues**:* Khalajzadeh et al. [15] developed three categories for accessibility and usability issues in mobile apps. They surveyed 1,200 app reviews and 1,200 issue comments, categorized them into App Usage, Inclusiveness, and User Reaction, and then developed a machine-learning model to do this task in the future. Although this study do not intend to automate classification, the LDA-generated topics will be compared to their taxonomy in tackling RQ3. However, it is important to note that only 12 projects with 2400 data points were reviewed in this study, whereas our study takes on a far larger data set. Thus, our taxonomy related to RQ3 will be more complex as it involves more data. Furthermore, this study focuses on the search for accessibility issues, whereas this study has a broader approach aiming at all human-centric issues.

In their study, Mascetti et al. [16] shed light on a distinct facet of mobile accessibility, focusing on Cross Platform Development Frameworks (CPDF) like React Native and Xamarin. They contend that despite developers recognizing the importance of implementing mobile accessibility, they encounter challenges due to the insufficient support for such functionalities within CPDF. Much like this study, which aims to unveil the mobile accessibility hurdles faced by developers, Mascetti and colleagues also compile a catalog of accessibility features offered by iOS and Android. However, it is essential to note that the authors caution against regarding this list as a formally comprehensive inventory of accessibility issues. While both studies address mobile accessibility concerns, this study critically diverges from theirs. Specifically, this study does not delve into CPDF and the associated mobile accessibility challenges.

Ballantyne et al. [17] undertook the task of assembling a comprehensive set of 92 distinct guidelines aimed at assessing mobile accessibility. They then applied these guidelines to evaluate the top 25 most popular mobile apps on the Google Play Store. Their findings suggest that while mobile apps demonstrate accessibility at the system level, issues persist at the usage level due to suboptimal design and content choices. The resemblance between their research and this study is worth noting, as both explore commonly discussed mobile accessibility issues among developers. However, there are noteworthy distinctions. Unlike this study, Ballantyne and colleagues conduct in-depth reviews of mobile apps, and their focus is confined solely to the Android operating system.

Summary: Specific papers within this context dedicates to creating taxonomies for mobile accessibility issues, which aligns closely with this study's ultimate research goal of offering a comprehensive overview of the mobile accessibility landscape among developers. This study also aims to provide a taxonomy-based analysis of Stack Overflow posts.

B. Stack Overflow Mining for Mobile Accessibility

Vendome et al. [24] undertook similar research, utilizing Stack Overflow mining techniques to collect a corpus of 810 discussion threads. In addition, they conducted a study involving mining 13,817 GitHub Android repositories. The study aimed to ascertain whether developers utilized Assistive Technologies (AT), specifically Accessibility Application Programming Interfaces (Accessibility-APIs) and assistive content generation techniques for GUIs. However, a notable distinction exists between their and this study's approaches. Vendome and colleagues relied on Stack Overflow tags for gathering discussion posts, whereas this study's method employs a generic SQL query to filter posts related to mobile accessibility. Moreover, this study encompasses both iOS and Android discussion threads without restricting the focus solely to Android, in contrast to their approach. Like this study's methodology,

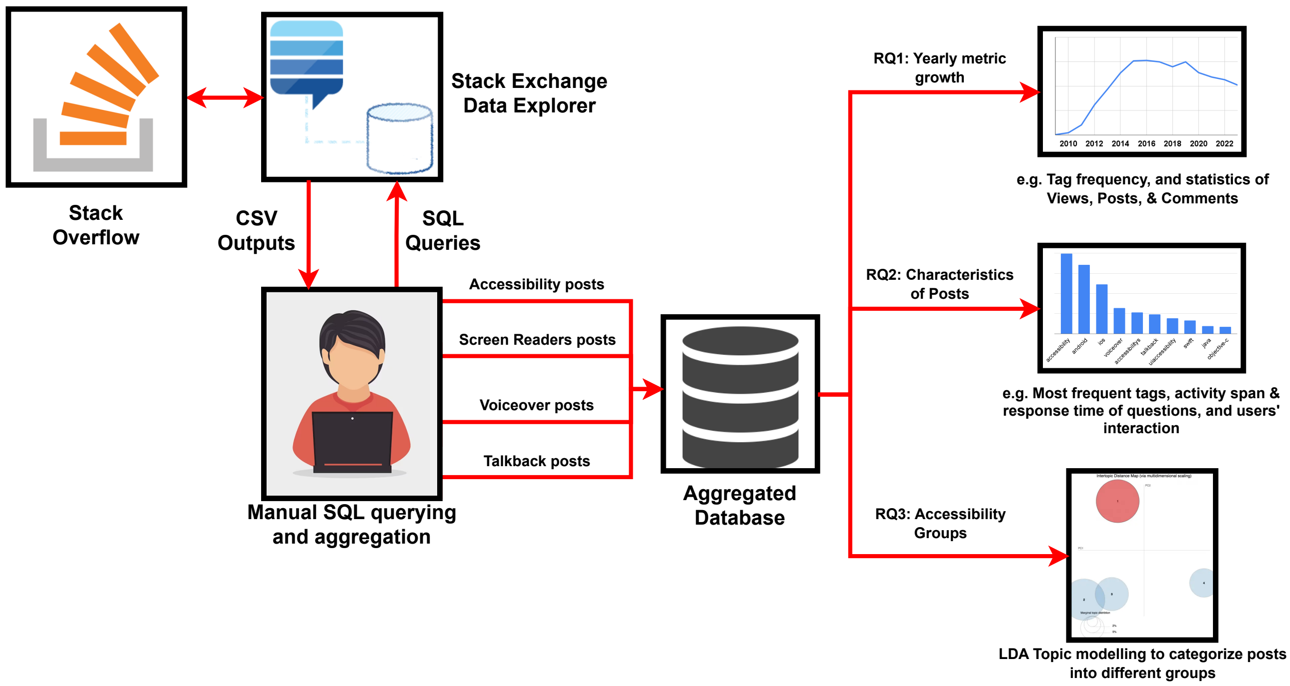

Fig. 1. Outline of the method used in mining Stack Overflow and the analysis of Research Questions

they also manually review Stack Overflow posts, selecting a statistically significant sample. Additionally, they construct a taxonomy to categorize various accessibility-related aspects within Android apps. However, it is worth mentioning that their study does possess certain limitations. It is constrained to the realm of Android accessibility aspects, relies on a relatively small pool of Stack Overflow discussion threads, and lacks generalizability to the broader domain of mobile accessibility.

Ma et al. [31] conducted a study addressing dark mode and its associated accessibility concerns within Android apps. Their approach involved data mining Stack Overflow posts related to accessibility and dark mode. They also analyzed over 6000 Android apps available on the Google Play Store, the official marketplace for Android apps. To categorize developers' challenges regarding dark mode, they employed LDA techniques to create a taxonomy based on Stack Overflow post data. Additionally, they summarized mobile accessibility issues specific to Android apps in dark mode. This study bears similarities to theirs by delving into the realm of mobile accessibility within the context of both iOS and Android apps. However, this study differs in certain respects. Unlike Ma et al., this study does not intend to scrutinize real-world apps. Furthermore, this study focuses on a broader spectrum of mobile accessibility issues rather than being confined solely to dark-mode apps.

Fontao et al. [32] performed similar research to this study by mining 1,568,377 Stack Overflow technical questions related to Android, iOS, and Windows Phone platforms. They then separated the discussions into categories that included hot topics, "what" and "How to" questions, and unanswered hot topics using the Latent Dirichlet Allocation algorithm to process the text. Finally, they produced ten strategies to support developer governance. Compared to their research, this study does not examine the Windows platform, as it is a less popular mobile platform than iOS or Android. However, similar to their research, this study also employs the LDA algorithm for analysis. Furthermore, this study may undertake a smaller data set but will focus more on the discussions in the data set. In this study, only the discussions about mobile app accessibility development for iOS and Android are focused upon, and other hot-topic areas are not.

Summary: Among the publications listed, several have involved the mining of Stack Overflow, albeit with a predominant focus on Android. For instance, Fontao et al. [32] explored general hot topics rather than exclusively addressing mobile accessibility. This study shares several similarities with these studies via the employment of SQL queries for Stack Overflow mining and LDA topic modeling for taxonomy creation. However, the distinguishing feature lies in the scope of this study's investigation, as a large-scale analysis of mobile accessibility is conducted by mining posts related to both Android and iOS.

III. Methodology

A. Goal

This study's dominant goal is to identify accessibility design challenges mobile app developers discuss on Stack Overflow. To achieve the goal, the three research questions of this study will be answered and analyzed according to the methodology stated in this section.

B. Proposed Methodology

Figure 1 shows a high-level overview of the proposed methodology. The following sections will elaborate on the individual steps within each methodology stage. This study follows the methodology outlined in "An Empirical Study on Refactoring Trends and Topics in Stack Overflow" by Peruma et al. [42]. In their research, Stack Overflow posts were sourced from the SOTorrent dataset [43]. That study filtered posts relevant to 'refactor' based on tags and titles. In addition, the study employed techniques such as LDA topic modeling for data analysis.

Likewise, this study's methodology will bear several similarities. Essential resemblances include the analysis of Stack Overflow posts related to mobile accessibility and utilizing techniques like LDA topic modeling. However, this study opts to retrieve Stack Overflow data using SQL queries from the Stack Exchange Data Explorer [44] instead of relying on SOTorrent, primarily to ensure the temporal relevance of the data. Furthermore, a similar approach will be adopted to examine the tags and titles of Stack Overflow posts during the data mining process for mobile accessibility-related content. This approach is favored to avoid false positives when filtering using the body of Stack Overflow posts, as shown by Peruma et al. [42]. Additionally, this study focuses on mining mobile accessibility posts pertinent to the Android and iOS mobile operating systems.

The study leverages *SQL* programming language for mining of Stack Overflow posts, *Python* programming language for in-depth analysis, and *Shell Scripting* for efficient data aggregation. Additionally, various Python packages and libraries, including *numpy, matplotlib, sqlite3, wordcloud, datetime, and statistics*, were instrumental in the analysis process. The complete source code for the study can be accessed through the designated GitHub repository [45].

C. Querying

The initial step of this study's methodology involves retrieving data from the Stack Exchange Data Explorer [44]. Each author of this study formulates an individual SQL query to extract pertinent information from the Stack Exchange Data Explorer, which is subsequently cross-examined to select the most effective query. After querying from the Stack Exchange Data Explorer, it was observed that the keyword *'accessibility'* generated a diverse range of posts related to mobile accessibility. To further enhance the dataset, with additional accessibility-related posts not captured by the *'accessibility'* keyword, a snowball technique was employed, identifying additional keywords by closely examining the accumulated Stack Overflow posts. Through this process, it is determined that the keywords *'screen readers'*, *'talkback'*, and *'voiceover'* specifically generated more accessibility-related posts concerning Android and iOS mobile applications. Subsequently, SQL queries were executed using these newly identified keywords to gather the relevant data.

However, during this exploratory phase, experiments revealed that the higher complexity of SQL queries often caused the Stack Exchange Data Explorer to time out. Consequently, the decision was made to run separate SQL queries for each identified four keywords to ensure data collection without time out. An illustrative sample of the SQL query generated for extracting accessibility-related posts is provided in Figure 2. This SQL query serves as a template and is modified into variants to incorporate the different tags contributed by other authors.

```sql
WITH
-- Get Post Ids
postIds AS (
  SELECT
    Posts.Id
  FROM
    Posts
  WHERE
    (
      Posts.Tags LIKE '%accessibility%'
    )
    AND (
      Posts.Tags LIKE '%<android%'
      OR Posts.Tags LIKE '%<ios%'
    )
    OR (
      LOWER (Posts.Title) LIKE '% accessibility %'
      AND (
        LOWER (Posts.Title) LIKE '% android %'
        OR LOWER (Posts.Title) LIKE '% ios %'
      )
    )
),

-- Get Questions
questions AS (
  SELECT
    Posts.*
  FROM
    Posts
    INNER JOIN postIds ON Posts.Id = postIds.Id
),

-- Get Answers
answers AS (
  SELECT
    Posts.*
  FROM
    Posts
    INNER JOIN postIds ON Posts.ParentId = postIds.Id
)

-- Join Questions and Answers
SELECT
  *
FROM
  questions
UNION
SELECT
  *
FROM
  answers
```

Fig. 2. The SQL query used to extract accessibility-related Stack Overflow posts

As illustrated in Figure 2, the querying process entails several key steps, including:

- Extracting the post IDs associated with Stack Overflow questions related to accessibility, employing filters based on post titles and tags.
- Retrieving the questions and their corresponding answers for the extracted post IDs.
- Merging the questions and answers to create a comprehensive dataset encompassing Stack Overflow posts discussing the accessibility of Android and iOS applications.

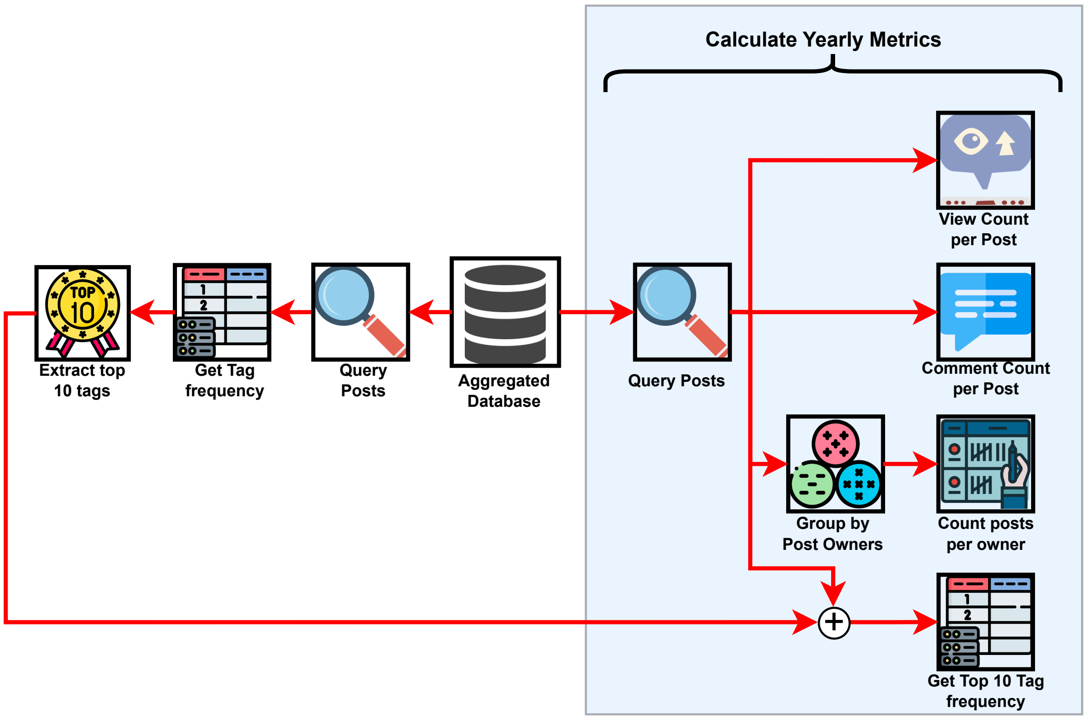

Fig. 3. Overview of the Research Question 1 procedure

Next, the data extracted from the Stack Exchange Data Explorer was saved as a Comma Separated Values (CSV) file for each identified keyword. These CSV files were then consolidated into a unified local SQLite database using a shell scripting approach. The initial bash script was created for Linux and MacOS and then converted into a batch file for Windows. During this process, only distinct Stack Overflow posts, based on ID, were selected for aggregation into the database.

D. Dataset

The consolidated SQLite database of Stack Overflow posts comprises **6,371 Stack Overflow posts** encompassing **3,022 questions** and **3,349 answers**. Each record within the database corresponds to a specific Stack Overflow post featuring 23 distinct fields, the comprehensive details of which are elaborated upon at the Meta Stack Exchange [46]. Some of the essential fields utilized to analyze the research questions include, but are not limited to, Tags, ViewCount, AnswerCount, PostCount, CommentCount, and Score.

E. RQ1 Procedure

Research Question 1 (RQ1) entails an analysis of the growth of mobile accessibility questions, with an overview of the procedure outlined in Figure 3. Concerning understanding the temporal evolution of mobile accessibility challenges, a quantitative investigation into the queried accessibility design Stack Overflow questions and answers on the local database was conducted for several proposed metrics. The metrics include the *growth rate of the top ten accessibility tags, changes in comment count per post, changes in post count per user, and changes in view count per post by year.*

- To obtain the change in the top ten tags, a selection of all tags was obtained via a query to the local database. Then, the frequency distribution was extracted. Subsequently, the distribution was sorted to obtain the *Top Ten Tags*. After that, the database was queried to obtain all tags by year and filtered using the *Top Ten Tags*. To examine the change, the yearly results were analyzed for frequency to obtain a count and are plotted by year.
- To obtain the change in view counts per post by year, a selection of *'ViewCount'* by year was queried.
- To obtain the change in comment count per post by year, a selection of *'CommentCount'* by year was queried.
- To obtain post count per user, a count of posts who share the same *'OwnerUserId'* was obtained by year.

The results of the last three queries were evaluated via mean, median, standard deviation, variance, min, max, and total. Only relevant metrics were retained concerning the research questions.

F. RQ2 Procedure

An overview of the RQ2 procedure is provided in Figure 4. Research Question 2 (RQ2) combines quantitative and qualitative analysis to explore the characteristics of mobile accessibility questions and answers for several proposed metrics. These proposed metrics are as follows:

Per Post:

- To gain insight into tags related to accessibility, a selection of *'Tags'* was queried, and the frequency of tags was extracted.

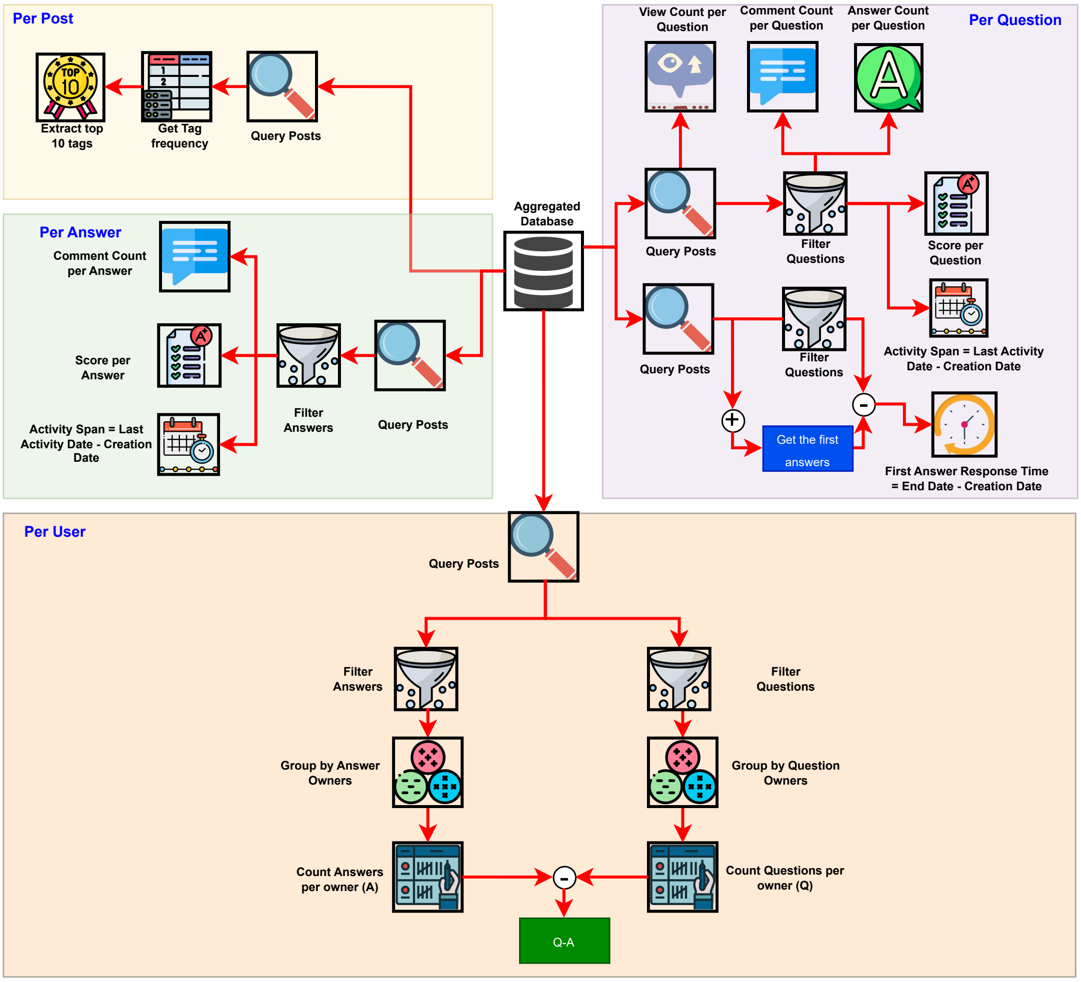

Fig. 4. Overview of the Research Question 2 procedure

Per Question:

- To obtain view count per post, a selection of *'ViewCount'* was queried. *Per posts* is equivalent to *per question* for this query as view count only applies to question posts, not answer posts.
- To obtain the first answer response time (in days) per question, a selection of *'CreationDate'* and *'ParentId'* was performed and treated as the *EndDate*. In the same query, a selection of *'CreationDate'* was obtained where the *EndDate*'s *'ParentId'* matched the *'Id'*. Only the *EndDate* closest to the *'CreationDate'* is retained and excludes other results that share the same *'ParentId'*.
- To obtain response time per question for all posted answers, even the responses after the first response, a selection of *'CreationDate'* and *'ParentId'* was performed and treated as the *EndDate*. In the same query, a selection of *'CreationDate'* was obtained where the *EndDate*'s *'ParentId'* matched the *'Id'*.
- To obtain the answer count per question, a selection of *'AnswerCount'* was queried.
- To obtain score per question, a selection of *'Score'* was queried.
- To obtain comment count per question, a selection of *'CommentCount'* was queried.
- To obtain the activity span (in days) per question, a difference of *'CreationDate'* and *'LastActivityDate'* was queried.

Per Answer:

- To obtain score per answer, a selection of *'Score'* was queried.
- To obtain comment count per answer, a selection of *'CommentCount'* was queried.
- To obtain the activity span (in days) per answer, a difference of *'CreationDate'* and *'LastActivityDate'* was queried.

Per User:

- To obtain knowledge of whether Stack Overflow users post more answers or questions, a Q minus A metric was introduced where Q represents questions, and A represents answers. To obtain the relevant information, a query involving counting posts per *'OwnerUserId'* where *'PostTypeId'* equals 1 for questions is utilized. To gather the answer count per user, a count of posts per *'OwnerUserId'* where *'PostTypeId'* equals 2 was performed. For each user, the number of answers asked was subtracted from the number of questions, which

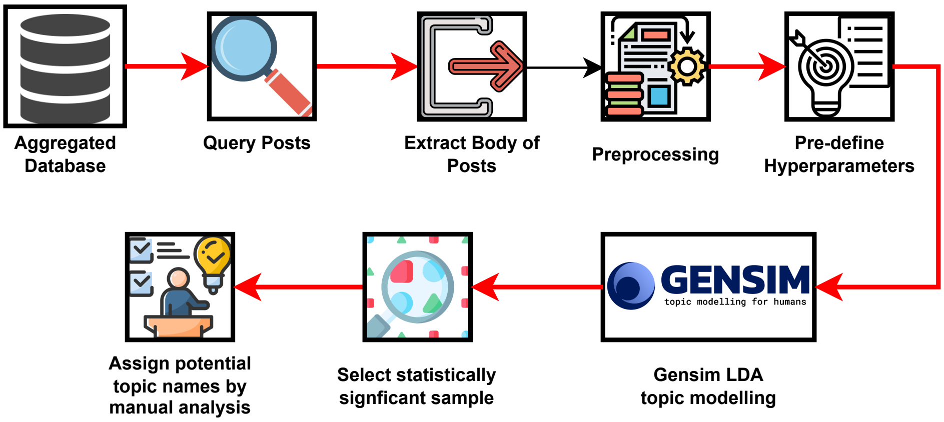

Fig. 5. Overview of the Research Question 3 procedure

resulted in the Q minus A metric. If Q minus A exceeds 0, the user posts more questions than answers. If Q minus A equals 0, then the users' question and answer are the same. If Q minus A is less than 0, the user answers more questions than they post.

- To perform a more insightful investigation about questions per user, a count of questions asked per user was extracted from the database.
- Similarly, regarding answers per user, a count of answers asked per user was extracted from the database.

For the queries that did not extract a frequency distribution, an analysis of the mean, median, standard deviation, variance, min, max, and total was calculated.

For the qualitative aspect, the frequency of tags was taken and classified manually into categories. For example, tags such as *screen-readers* is categorized under *Visual Impaired* and *popupmenu* and *android-progressbar* are classified as *ui*. This will allow extrapolation of possible focus in specific categories.

G. RQ3 Procedure

Research Question 3 (RQ3) focuses on a qualitative analysis where the broader challenges of mobile accessibility are investigated. The summary of the RQ3 procedure is illustrated in Figure 5. The categorization of the challenges related to mobile accessibility development involves employing LDA (Latent Dirichlet Allocation) topic modeling to analyze the collected data. This natural language processing technique clusters the data into distinct groups based on similarities, providing insights into the mobile accessibility challenges.

The implementation of the LDA model will be conducted using Gensim [47], an open-source Python library for topic modeling. Before feeding the Stack Overflow post bodies into the model, the data will undergo preprocessing involving standard NLP (Natural Language Processing) techniques, including the removal of trailing and leading empty spaces, expansion of contracted words, elimination of HTML tags and URLs, exclusion of non-alphanumeric characters and punctuation, tokenization, conversion of text to lowercase, and removal of standard and custom stop words. Custom stopwords were eliminated by identifying frequently occurring unigrams, bigrams, and trigrams. Subsequently, irrelevant custom stop-words were manually removed for this study.

To ensure accurate results, it is crucial to predefine specific hyperparameters, such as the number of topics, data size, and training iterations. Determining the ideal number of topics for the LDA models involved running models with topics ranging from 2 to 10 for each unigram, bigram, and trigram. From these models, the optimal number of topics was selected through manual inspection, focusing on achieving the ideal segregation of word tokens (unigrams, bigrams, trigrams) within topics. Following the grouping by the LDA model, the statistically significant samples are manually selected from each group for in-depth analysis. The manual analysis will facilitate the identification of potential accessibility challenges corresponding to each topic group.

IV. Results and Discussion

To identify accessibility design challenges mobile app developers encounter on Stack Overflow, the results from the methods used to answer the research questions will be analyzed.

> RQ1: What are the attributes related to the growth of mobile accessibility questions?

As shown in figs. 6 to 9 the early 2010s witnessed a remarkable surge in discussions concerning mobile accessibility. Upon closer examination, several probable factors emerged, contributing to this rapid expansion. Notably, in 2009, Apple unveiled Voiceover for the iPhone 3GS [48]. Concurrently, Google's eyes-free project initiated that year facilitated speech-enabled eyes-free Android applications [49], eventually integrated into Android 4.0 (Ice Cream Sandwich) as Talkback in 2011 [50]. This Android 4.0 version introduced various accessibility features, such as activating accessibility options through a clockwise rectangle touch gesture [50]. Moreover, Android 4.0 introduced the Explore-by-touch mode, enabling users to touch the screen and receive spoken content feedback [50]. Notably, it included a system-wide font size adjustment, beneficial for visually-impaired

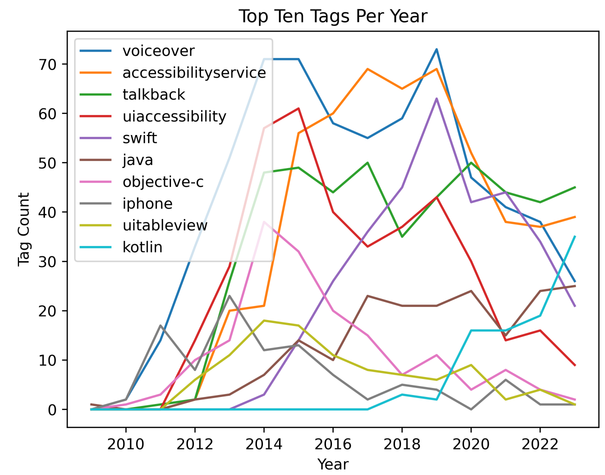

Fig. 6. Change in top ten tags per year with accessibility, ios, and Android tags removed

users [50]. Subsequently, Android Jelly Bean (4.1, 4.2, and 4.3) incorporated additional accessibility enhancements. For instance, version 4.1 introduced accessibility focus [51, 52], allowing users to navigate focusable objects using vertical or horizontal swipes and activating them with a double tap. Complex gestures like haptics were also introduced [51, 52]. Android 4.2, in 2012, expanded these features, enabling screen magnification through triple-tapping and zoom-and-pan with two fingers, alongside new accessibility feedback for Braille devices [53, 54].

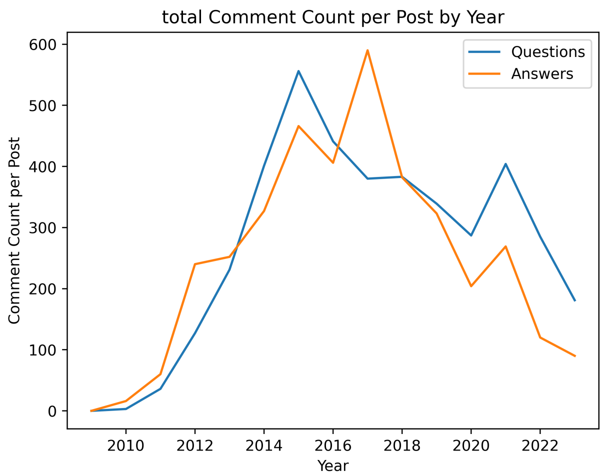

Fig. 7. Yearly variation of Total Comment Count Per Post

Simultaneously, various regulations and laws were implemented, significantly impacting mobile app accessibility during this period. Notably, the Web Content Accessibility Guidelines (WCAG) 2.0 in 2008 [20] offered recommendations for enhancing web content accessibility. Similarly, the EN 301 549 European Standard [55] aimed to align ICT products and services in Europe with these accessibility requirements.

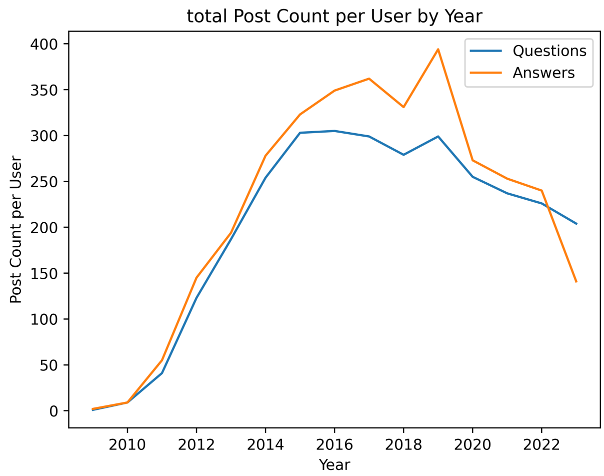

Fig. 8. Yearly variation of Total Post Count Per User

Considering these factors, we posit a correlation between the surge in accessibility-related inquiries from 2010 to 2015 and the establishment of accessibility regulations, technological advancements in accessibility features, and the launch of accessibility tools during that period.

No normalization was applied to the data (e.g., using methods to compare accessibility posts and all Stack Overflow posts per year to accurately represent the potential change in Stack Overflow's growth contextually). It is apparent in the initial years of 2009 and 2010, where samples collected were limited (i.e., below ten), and the perceived exponential growth of metrics can be misleading if considering the industry contextually instead of solely on Stack Overflow posts.

The limited presence of mobile accessibility-related posts in 2009 and 2010 can be attributed to Stack Overflow's inception in 2008 [56]. Stack Overflow was still emerging, hosting fewer than 3.7 million questions and answers. The count for the number of posts from 2010 and earlier was obtained through the SQL query:

```sql
SELECT COUNT(*) FROM Posts
WHERE YEAR(CreationDate) <= 2010
```

One intriguing observation to note is the drop in mobile accessibility inquiries post-2016, as depicted in Figure 6. The Level Access survey in 2022 [57] sheds light on this trend. Among 1030 participants, 62.7% reported receiving accessibility training, indicating a rising awareness and educational focus on accessibility. The rising focus could lead to fewer queries from developers, empowered with better understanding. The report also reveals that 91% of respondents utilize free accessibility tools, signaling increased resources dedicated to accessibility testing, potentially diminishing the need for related questions. Furthermore, there is an evident shift in development practices, notably proactive inclusion of accessibility considerations in early planning, design, and definition stages by User Experience Designers and development

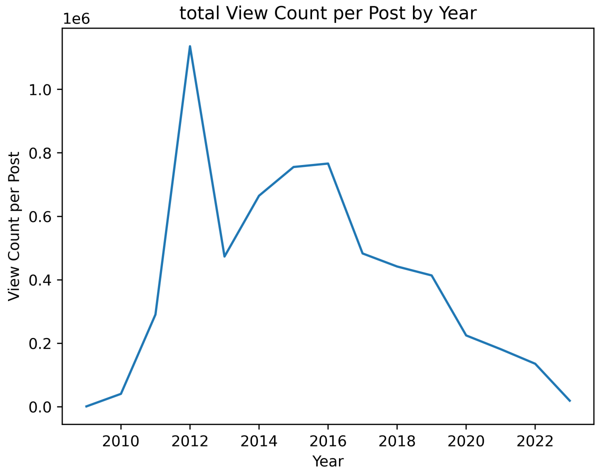

Fig. 9. Yearly variation of Total View Count Per Post

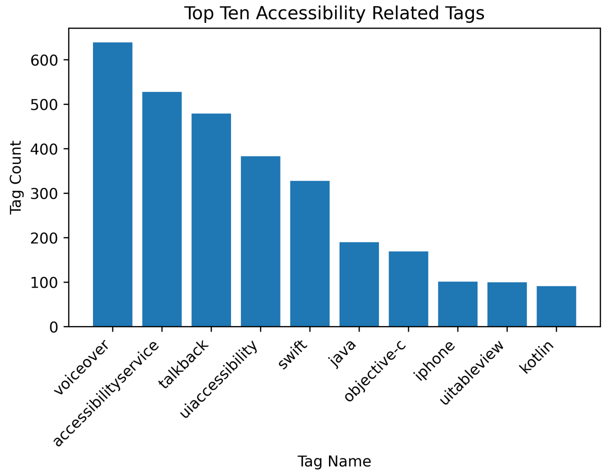

Fig. 10. Top Ten Tags excluding accessibility tag and operating systems tags (android and ios)

teams [57]. This shift likely contributed to the relative decrease in accessibility inquiries.

Considering these factors, the maturity of available information, augmented accessibility testing resources, a growing developer community with enhanced awareness and education, and the evolution of development practices collectively support the observed decline in accessibility-related queries post-2016.

While not explicitly addressing mobile accessibility, the WebAIM report assessing website accessibility [58] highlights that over 96.3% of website home pages fail to meet WCAG guidelines. This observation leads us to speculate that similar trends might exist concerning mobile accessibility despite the maturity of resources on mobile accessibility.

Figure 7, depicting the total comment count per post, and Figure 8, showing the total post count per user, exhibit a parallel upward trend from 2010 to 2016, followed by a decline in subsequent years. These trends align with the variation in tag popularity per year, as illustrated in Figure 6. We conjecture that the rationale behind the initial surge and subsequent decline in the total comment count per post and the total post count per user mirrors that of the previously discussed case.

Moreover, Figure 9 illustrates a peak in the total view count per post in 2012, succeeded by a decline in subsequent years. This early surge aligns with the inception of accessibility tools, laws, and regulations between 2009 and 2014. Additionally, this spike might be attributed to Stack Overflow's emergence as a Q&A platform for developers, potentially contributing to the increased traction of accessibility-related questions on the platform.

RQ2: What are the characteristics of mobile accessibility questions?

Due to the scope of this research focused on accessibility and mobile applications, all tag results from queries inherently pertain to that scope. Therefore, when analyzing RQ1, we removed the top three tags, Accessibility, Android, and iOS, to reflect meaningful tags. From Figure 10, there is a significant interest in screen readers evident from the following tags; *voiceover* an iOS screen reader, *uiaccessibility* which works in conjunction with *voiceover*, *talkback* an Android screen reader, and *accessibilityservice* which works in association with *talkback*. Additionally, *accessibilityservice* and *uiaccessibility* provide additional functionality beyond interaction with screen readers. Adjacent to the tags mentioned earlier, languages for iOS development: *swift* and *objective-c* are shown. As for Android, *java* and *kotlin* are used.

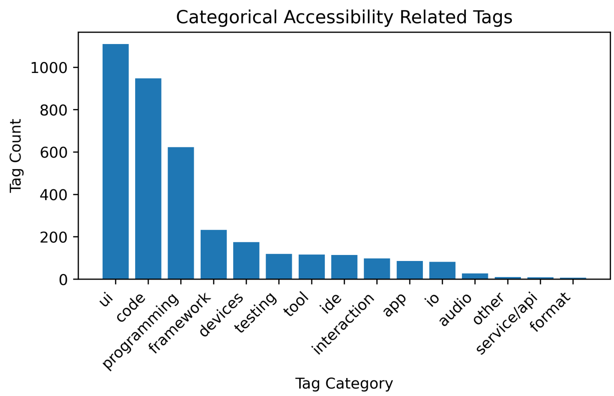

Fig. 11. Categories of tags with accessibility and operating systems removed

To gain further insights, Figures 11 and 12 were created in which tags were manually grouped (by hand) into categories. Figure 11 shows the categorized breakdown of all tags except operating systems and direct accessibility-related classifications obscured. The following will explain briefly

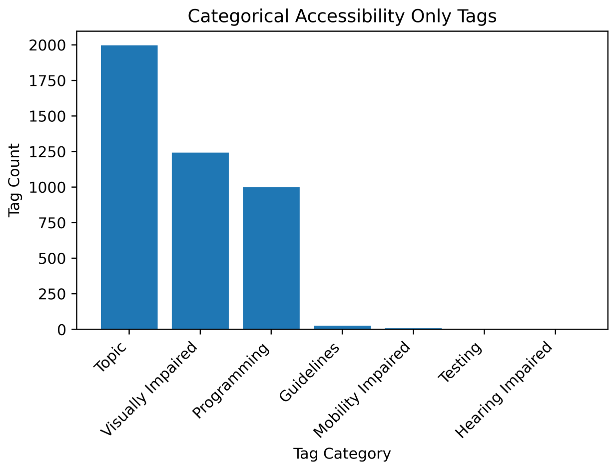

Fig. 12. Focused insight on the category of tags relating only to accessibility

some of the categories for general clarification: *ui* represents tags associated with UI elements, *other* are not categorized tags such as *jailbreak* or *hebrew*, and *format* relates to file formats.

The accessibility categories directly associated with accessibility can be found in Figure 12, where *Topic* relates to tags such as *accessibility* and *accessible*. *Visually Impaired* category represents tags that aid in visual impairments, which are inclusive of screen readers, *magnification*, and *text-to-speech*. Similarly, *Mobility Impaired* is associated with functionality to automated gestures, and *Hearing Impaired* contains *closed-captions*. *Guidelines* provide information on laws and accessibility guidelines. Lastly, *Testing* encompasses testing tools.

As shown in Figure 12, *Visually Impaired* and *Programming* categories reflect the top tags found in Figure 10 and numerically contribute as a significant portion towards to categorical counts. Figure 13 represents a statistical analysis of a few chosen metrics. Considering both quantitative and qualitative analysis, the following conclusions can be extrapolated. Using the median as the primary metric due to resilience against outliers:

- Expected answers for questions can be posted within two days.
- Generally, one answer is associated with a question.
- There is a significant lack of comments per post (questions and answers).
- Posts tend to receive a limited number of up-votes (score) of the posts (questions and answers).
- Given any author, on median, are more likely to post more questions than answers.
- Posts focus on visual impairments rather than other disabilities dis-proportionally (see Figure 12).

RQ3: What are the challenges associated with mobile accessibility development?

Several LDA models were generated, ranging from 2 to 10 topics for unigrams, bigrams, and trigrams. However, to simplify the presentation, only the results for the LDA model output with four topics for bigrams were showcased, which effectively categorize bigram tokens. The detailed results are in Appendix B. Utilizing the pyLDAvis Python package [59], the figs. 14 to 17 present the visualized categorized topics alongside associated word tokens for each topic.

In this visualization, the parameter lambda (λ) within the range of 0 to 1 adjusts the relevance of word tokens within specific topics [60]. Lower λ values emphasize tokens particular to a topic, while higher values may include tokens found in multiple topics. In their original paper on LDA visualization, Sievert and Shirley [60] suggest that λ values ranging from 0.3 to 0.6 offer improved categorization, with 0.6 yielding optimal results in their user study. Consequently, we experimented with λ values within this range. After trial and error, we settled on a λ value of 0.3 for our specific case.

Observing the visualization in figs. 14 to 17, red bar plots indicate the presence of relevant word tokens within each topic.

For instance, Figure 14 highlights bigram word tokens predominantly containing ”talkback” like *talkback_message, problem_talkback, suppress_talkback, talkback_announcement*, and *google_talkback*, suggesting a correlation with <u>talkback-related issues</u> in topic 1.

Similarly, Figure 15 showcases bigram tokens such as *screen_reader, page_swipe, braille_display, braille_board, wireless_keyboard, onscreen_keyboard*, and *braille_keyboard*, implying topic 2's association with <u>mobile accessibility solutions for visually impaired users</u>.

In Figure 16, words like *ui_accessibility, date_picker, hiding_buttons, mmmm_yyyy*, and *lock_screen* suggest topic 3 may address <u>user interface or frontend-related issues</u>. Notably, longer bigrams appear, possibly due to code segments on Stack Overflow or low-level accessibility functionalities in Android and iOS.

Lastly, Figure 17 displays words like *snippets_objc, crash_ios, voiceover_switch, switch_voiceover*, and *objc_swift*, indicating a connection to objective-c, swift programming languages, and iOS voiceover functionality in topic 4, potentially addressing <u>voiceover-related issues</u>.

Additionally, it is essential to acknowledge that we manually assign these topic names using the tokens. Therefore, altering the λ value and obtaining a distinct set of tokens for each topic might yield different types of topic names.

While we have demonstrated word token categorization for four bigram topics, further subdivisions and additional insights could emerge with more topics and higher n-grams like four-grams and five-grams.

V. Threats to Validity

The study's scope was confined to extracting data solely from Stack Overflow, prompting considerations about its generalizability. While Stack Overflow is a go-to platform for developer queries, it primarily addresses programming questions without specifically emphasizing accessibility design. For

name	average	median	stdev	min	max	total
View Count per Post	1995.31	714.00	7595.86	7.00	340916.00	6029821.00
All Responses, Response Time in Days per Question	322.25	7.49	647.88	0.00	4303.01	1074716.05
First Response, Response Time in Days per Question	100.37	1.32	321.09	0.00	3737.42	220403.22
Answer Count per Question	1.11	1.00	1.10	0.00	13.00	3349.00
Score per Question	2.61	1.00	6.66	-5.00	245.00	7895.00
Comment Count per Question	1.34	0.00	2.11	0.00	17.00	4054.00
Activity Span in Days per Question	423.73	12.55	774.95	0.00	4370.83	1096618.19
Score per Answer	2.51	1.00	8.30	-8.00	291.00	8402.00
Comment Count per Answer	1.12	0.00	1.96	0.00	23.00	3745.00
Activity Span in Days per Answer	257.92	0.85	565.38	0.00	3684.91	234191.24
Q Minus A per Author	-0.08	1.00	4.83	-218.00	11.00	-327.00

Fig. 13. Statistics of gathered Stack Overflow posts

deeper insights into accessibility-related inquiries, developers might frequently go to more specialized forums like Google groups for Android accessibility [61] or Apple developer groups for iOS accessibility [62]. Moreover, alternative platforms for software discussions exist, such as GitHub Discussions [63], and dedicated sub-communities within broader discussion platforms like Reddit [64] and Quora [65]. These avenues potentially serve as viable alternatives for software developers seeking answers to mobile accessibility questions beyond Stack Overflow.

Additionally, this study's scope was confined to Android and iOS, prompting a need to consider its applicability across all mobile operating systems. Furthermore, this study excluded other mobile operating systems such as Huawei's EMUI, Amazon's Fire OS, LG Electronics's LG UX, and Google's ChromeOS. Nevertheless, it is vital to highlight that statistics regarding the mobile operating system market reveal Android holding 70.5% and iOS holding 28.8%, summing up to 99.3% [29]. Android and iOS accounts for a substantial portion of mobile applications, reinforcing the relevance of this study.

The "snowball" technique employed during the data collection to identify keywords for querying Stack Overflow posts relies on the our perception. Keywords noticed during our initial query for Android and iOS accessibility issues were then used in further queries. However, this approach may have missed other relevant keywords that should have been included in the queries and resultant data.

In RQ1, while introducing new accessibility features, laws, and regulations and the emergence of StackOverflow in the early 2010s align with the growth of accessibility tags, it is essential to consider the potential influence of additional external factors on this growth.

When sorting tags into categories in RQ2, it should be noted that some tags were ambiguous or applicable to multiple categories. For example, the cocoa-touch framework is a touch-testing framework that applies both touch and testing. As a result, only one category was assigned to the cocoa-touch framework. The assignment is done by the order of relevancy depicted in the code. However, since it is categorized manually, incorrect categorization is possible due to human error and subjectivity.

The selection of custom stopwords for removal in RQ3 was performed manually. Consequently, the possibility exists that stopwords were overlooked during the manual selection process. Furthermore, there were no established criteria for custom stopword selection. Therefore, the choice of custom stopwords could be prone to bias. In RQ3, exploring more LDA topics could have provided more refined subcategories. Due to computational limitations, our exploration was confined to topics ranging from 2 to 10. Furthermore, in RQ3, assigning names to LDA topic groups relied on observed tokens, a process subject to the λ parameter and the authors' subjective perception of identified keywords.

VI. Conclusion

In summary, this study delves into three research questions, yielding the following key findings:

> **RQ1:** What are the attributes related to the growth of mobile accessibility questions?

RQ1 Takeaway: Various factors, including the enactment of accessibility regulations and laws and the inception of new accessibility features by major vendors such as Google and Apple, influence the growth of accessibility questions.

> **RQ2:** What are the characteristics of mobile accessibility questions?

RQ2 Takeaway: Accessibility questions on Stack Overflow demonstrate distinctive traits. They typically receive responses within two days and usually garner at least one answer per query. Any unique author, on median, is more likely to post questions then answer them. However, in total there are more answers than questions. Moreover, these queries tend to lack comments and upvotes.

> **RQ3:** What are the challenges associated with mobile accessibility development?

RQ3 Takeaway: The advent of new accessibility functionalities like voiceover and talkback has presented significant challenges for mobile developers regarding implementation. Additionally, challenges are evident in user interface design, and developers encounter difficulties finding practical solutions for mobile accessibility issues.

VII. Future Work

Future work for this study could involve validating the analyzed results with developers through user studies and questionnaires. Additionally, conducting a manual analysis on a statistically significant sample drawn from the collected questions could offer comparisons with the current analysis, shedding light on potential insights and validating research findings.

In addition, exploring external factors influencing the growth of accessibility questions warrants further investigation. Also, expanding the search space for LDA topic count and considering higher n-grams, such as four-grams and five-grams, presents an avenue for deeper analysis.

Moreover, other potential research directions involve automating manual steps in our methodology, such as removing custom words and employing automated snowball querying techniques for data gathering.

VIII. Acknowledgment

The technical support and advanced computing resources from the University of Hawaii Information Technology Services – Cyberinfrastructure, funded in part by the National Science Foundation MRI award #1920304, are gratefully acknowledged.

References

[1] "Forecast number of mobile users worldwide from 2020 to 2025," https://www.statista.com/statistics/218984/number-of-global-mobile-users-since-2010/, accessed: 2023-09-26.

[2] "50 mobile app statistics for app growth in 2022," https://clearbridgemobile.com/stats-for-mobile-app-growth-and-success/#, accessed: 08-Oct-2023.

[3] "Disability," https://www.who.int/news-room/fact-sheets/detail/disability-and-health, accessed: 03-Oct-2023.

[4] S. K. Kane, C. Jayant, J. O. Wobbrock, and R. E. Ladner, "Freedom to roam: a study of mobile device adoption and accessibility for people with visual and motor disabilities," in *Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility*, 2009, pp. 115–122. [Online]. Available: https://doi.org/10.1145/1639642.1639663

[5] S. K. Kane, J. O. Wobbrock, and R. E. Ladner, "Usable gestures for blind people: understanding preference and performance," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2011, pp. 413–422. [Online]. Available: https://doi.org/10.1145/1978942.1979001

[6] S. S. Nathan, A. Hussain, and N. L. Hashim, "Studies on deaf mobile application," in *AIP Conference Proceedings*, vol. 1761, no. 1. AIP Publishing, 2016. [Online]. Available: https://doi.org/10.1063/1.4960939

[7] M. Naftali and L. Findlater, "Accessibility in context: understanding the truly mobile experience of smartphone users with motor impairments," in *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility*, 2014, pp. 209–216. [Online]. Available: https://doi.org/10.1145/2661334.2661372

[8] E. C. Almao and F. Golpayegani, "Are mobile apps usable and accessible for senior citizens in smart cities?" in *Human Aspects of IT for the Aged Population. Design for the Elderly and Technology Acceptance: 5th International Conference, ITAP 2019, Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, July 26-31, 2019, Proceedings, Part I 21*. Springer, 2019, pp. 357–375. [Online]. Available: https://doi.org/10.1007/978-3-030-22012-9_26

[9] J. Dekelver, M. Kultsova, O. Shabalina, J. Borblik, A. Pidoprigora, R. Romanenko *et al.*, "Design of mobile applications for people with intellectual disabilities," *Communications in Computer and Information Science*, vol. 535, pp. 823–836, 2015.

[10] A. S. Ross, X. Zhang, J. Fogarty, and J. O. Wobbrock, "Examining image-based button labeling for accessibility in android apps through large-scale analysis," in *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*, 2018, pp. 119–130. [Online]. Available: https://doi.org/10.1145/3234695.3236364

[11] ——, "An epidemiology-inspired large-scale analysis of android app accessibility," *ACM Transactions on Accessible Computing (TACCESS)*, vol. 13, no. 1, pp. 1–36, 2020. [Online]. Available: https://doi.org/10.1145/3348797

[12] S. Yan and P. Ramachandran, "The current status of accessibility in mobile apps," *ACM Transactions on Accessible Computing (TACCESS)*, vol. 12, no. 1, pp. 1–31, 2019. [Online]. Available: https://doi.org/10.1145/3300176

[13] A. Alshayban, I. Ahmed, and S. Malek, "Accessibility

issues in android apps: state of affairs, sentiments, and ways forward," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, 2020, pp. 1323–1334. [Online]. Available: https://doi.org/10.1145/3377811.3380392

[14] M. Di Gregorio, D. Di Nucci, F. Palomba, and G. Vitiello, "The making of accessible android applications: an empirical study on the state of the practice," *Empirical Software Engineering*, vol. 27, no. 6, p. 145, 2022. [Online]. Available: https://doi.org/10.1007/s10664-022-10182-x

[15] H. Khalajzadeh, M. Shahin, H. O. Obie, P. Agrawal, and J. Grundy, "Supporting developers in addressing human-centric issues in mobile apps," *IEEE Transactions on Software Engineering*, vol. 49, no. 4, pp. 2149–2168, 2022. [Online]. Available: https://doi.org/10.1109/TSE.2022.3212329

[16] S. Mascetti, M. Ducci, N. Cantù, P. Pecis, and D. Ahmetovic, "Developing accessible mobile applications with cross-platform development frameworks," in *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility*, 2021, pp. 1–5. [Online]. Available: https://doi.org/10.1145/3441852.3476469

[17] M. Ballantyne, A. Jha, A. Jacobsen, J. S. Hawker, and Y. N. El-Glaly, "Study of accessibility guidelines of mobile applications," in *Proceedings of the 17th international conference on mobile and ubiquitous multimedia*, 2018, pp. 305–315. [Online]. Available: https://doi.org/10.1145/3282894.3282921

[18] J. Lazar, "The potential role of us consumer protection laws in improving digital accessibility for people with disabilities," *U. Pa. JL & Soc. Change*, vol. 22, p. 185, 2019.

[19] J. Lazar, D. F. Goldstein, and A. Taylor, *Ensuring digital accessibility through process and policy*. Morgan kaufmann, 2015.

[20] "Web content accessibility guidelines 2.0. (dec 2008)," https://www.w3.org/TR/WCAG20/, 2008.

[21] "Web content accessibility guidelines (wcag) 2.1," https://www.w3.org/TR/WCAG21/, 2023.

[22] "Android accessibility," https://developer.android.com/guide/topics/ui/accessibility, 2023.

[23] "Apple accessibility," https://developer.apple.com/accessibility/, 2023.

[24] C. Vendome, D. Solano, S. Liñán, and M. Linares-Vásquez, "Can everyone use my app? an empirical study on accessibility in android apps," in *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2019, pp. 41–52. [Online]. Available: https://doi.org/10.1109/ICSME.2019.00014

[25] C. Silva, M. M. Eler, and G. Fraser, "A survey on the tool support for the automatic evaluation of mobile accessibility," in *Proceedings of the 8th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion*, 2018, pp. 286–293. [Online]. Available: https://doi.org/10.1145/3218585.3218673

[26] "Stack overflow," https://stackoverflow.co/, accessed: 2023-10-1.

[27] "Stack overflow developer survey 2023," https://survey.stackoverflow.co/2023/#community-stack-overflow-site-use, accessed: 03-Oct-2023.

[28] "Stack overflow questions," https://stackoverflow.com/questions, accessed: 03-Oct-2023.

[29] "Global marketshare held by mobile opertaing systems since 2009," https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/, accessed: 2023-09-26.

[30] "Number of apps avaible in leading app stores," https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/, accessed: 2023-09-26.

[31] S. Ma, C. Chen, H. Khalajzadeh, and J. Grundy, "A first look at dark mode in real-world android apps," *ACM Transactions on Software Engineering and Methodology*, 2022. [Online]. Available: https://doi.org/10.1145/3604607

[32] A. Fontao, B. Ábia, I. Wiese, B. Estácio, M. Quinta, R. P. d. Santos, and A. C. Dias-Neto, "Supporting governance of mobile application developers from mining and analyzing technical questions in stack overflow," *Journal of Software Engineering Research and Development*, vol. 6, pp. 1–34, 2018. [Online]. Available: https://doi.org/10.1186/s40411-018-0052-6

[33] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003. [Online]. Available: https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf?ref=https://githubhelp.com

[34] P. Acosta-Vargas, L. Salvador-Ullauri, J. Jadán-Guerrero, C. Guevara, S. Sanchez-Gordon, T. Calle-Jimenez, P. Lara-Alvarez, A. Medina, and I. L. Nunes, "Accessibility assessment in mobile applications for android," in *Advances in Human Factors and Systems Interaction: Proceedings of the AHFE 2019 International Conference on Human Factors and Systems Interaction, July 24-28, 2019, Washington DC, USA 10*. Springer, 2020, pp. 279–288. [Online]. Available: https://doi.org/10.1007/978-3-030-20040-4_25

[35] A. S. Alotaibi, P. T. Chiou, and W. G. Halfond, "Automated repair of size-based inaccessibility issues in mobile applications," in *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2021, pp. 730–742. [Online]. Available: https://doi.org/10.1109/ASE51524.2021.9678625

[36] S. Chen, C. Chen, L. Fan, M. Fan, X. Zhan, and Y. Liu, "Accessible or not? an empirical investigation of android app accessibility," *IEEE Transactions on Software Engineering*, vol. 48, no. 10, pp. 3954–3968, 2021. [Online]. Available: https://doi.org/10.1109/TSE.2021.3108162

[37] M. M. Eler, J. M. Rojas, Y. Ge, and G. Fraser, "Automated accessibility testing of mobile apps," in *2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST)*. IEEE, 2018, pp. 116–126. [Online]. Available: https:

//doi.org/10.1109/ICST.2018.00021
[38] “Android lint,” https://developer.android.com/studio/write/lint, accessed: 02-Oct-2023.
[39] “Espresso,” https://developer.android.com/training/testing/espresso, accessed: 02-Oct-2023.
[40] “Robolectric,” https://robolectric.org/, accessed: 02-Oct-2023.
[41] D. A. Mateus, C. A. Silva, A. F. de Oliveira, H. Costa, and A. P. Freire, “A systematic mapping of accessibility problems encountered on websites and mobile apps: A comparison between automated tests, manual inspections and user evaluations,” *Journal on Interactive Systems*, vol. 12, no. 1, pp. 145–171, 2021. [Online]. Available: https://doi.org/10.5753/jis.2021.1778
[42] A. Peruma, S. Simmons, E. A. AlOmar, C. D. Newman, M. W. Mkaouer, and A. Ouni, “How do i refactor this? an empirical study on refactoring trends and topics in stack overflow,” *Empirical Software Engineering*, vol. 27, no. 1, p. 11, 2022. [Online]. Available: https://doi.org/10.1007/s10664-021-10045-x
[43] S. Baltes, L. Dumani, C. Treude, and S. Diehl, “Sotorrent: reconstructing and analyzing the evolution of stack overflow posts,” in *Proceedings of the 15th international conference on mining software repositories*, 2018, pp. 319–330. [Online]. Available: https://dl.acm.org/doi/10.1145/3196398.3196430
[44] “Query stack overflow,” https://data.stackexchange.com/stackoverflow/query/new, accessed: 04-Oct-2023.
[45] “Stackoverflow-mobileaccessibility,” https://github.com/iSQARE-Lab/StackOverflow-MobileAccessibility/tree/main, accessed: 07-Nov-2023.
[46] “Database schema documentation for the public data dump and sede,” https://meta.stackexchange.com/questions/2677/database-schema-documentation-for-the-public-data-dump-and-sede/2678#2678, accessed: 07-Nov-2023.
[47] “gensim 4.3.2,” https://pypi.org/project/gensim/, accessed: 11-Dec-2023.
[48] “Apple announces the new iphone 3gs-the fastest, most powerful iphone yet,” https://www.apple.com/newsroom/2009/06/08Apple-Announces-the-New-iPhone-3GS-The-Fastest-Most-Powerful-iPhone-Yet/, accessed: 08-Dec-2023.
[49] “Eyes-free,” https://code.google.com/archive/p/eyes-free/, accessed: 08-Dec-2023.
[50] “Ice cream sandwich: android developers,” https://developer.android.com/about/versions/android-4.0-highlights#simple, accessed: 08-Dec-2023.
[51] “Android 4.1 apis: android developers,” https://developer.android.com/about/versions/android-4.1#A11y, accessed: 08-Dec-2023.
[52] “Jelly bean: android developers,” https://developer.android.com/about/versions/jelly-bean#android-4.1, accessed: 08-Dec-2023.
[53] “Android 4.2 adds gesture typing, wireless tv display, multiple user support on tablets, and more,” https://www.theverge.com/2012/10/29/3569244/android-4-2-new-features-miracast-gesture-keyboard-multiple-users-photo-sphere, accessed: 08-Dec-2023.
[54] “Jelly bean: android developers,” https://developer.android.com/about/versions/jelly-bean#android-4.2, accessed: 08-Dec-2023.
[55] “Etsi en 301 549 - v1.1.2,” https://www.etsi.org/deliver/etsi_en/301500_301599/301549/01.01.02_60/en_301549v010102p.pdf, accessed: 08-Dec-2023.
[56] “Five years ago, stack overflow launched. then, a miracle occurred.” https://stackoverflow.blog/2013/09/16/five-years-ago-stack-overflow-launched-then-a-miracle-occurred/, accessed: 08-Dec-2023.
[57] “2022 state of digital accessibility - levelaccess.com,” https://www.levelaccess.com/wp-content/uploads/2022/10/2022_State_of_Digital_Accessibility_Report.pdf, accessed: 08-Dec-2023.
[58] “The webaim million 2023 report on the accessibility of the top 1,000,000 home pages,” https://webaim.org/projects/million/, accessed: 08-Dec-2023.
[59] “pyldavis 3.4.1,” https://pypi.org/project/pyLDAvis/, accessed: 11-Dec-2023.
[60] C. Sievert and K. Shirley, “Ldavis: A method for visualizing and interpreting topics,” in *Proceedings of the workshop on interactive language learning, visualization, and interfaces*, 2014, pp. 63–70.
[61] “Eyes-free programming and development,” =https://groups.google.com/g/eyes-free-dev/about, accessed: 11-Dec-2023.
[62] “Accessibility: Apple developer forums,” https://developer.apple.com/forums/tags/accessibility, accessed: 11-Dec-2023.
[63] “Github discussions documentation,” https://docs.github.com/en/discussions, accessed: 11-Dec-2023.
[64] “Reddit,” https://www.reddit.com/, accessed: 11-Dec-2023.
[65] “Quora - a place to share knowledge and better understand the world,” https://www.quora.com/, accessed: 11-Dec-2023.

Appendix A
Related Work Papers

Ref.	Summary	Research Questions
[11]	This article performs a large-scale analysis of free Android apps to count accessibility barriers & related factors. They rank and categorize their results.	None Stated
[24]	After observing developers seldom use Accessibility APIs, this study focuses on the developer's perspective by investigating Stack Overflow. Lessons to guide future developers were distilled from this.	RQ1: What accessibility aspects are discussed by developers in Stack Overflow?
[17]	This study examined accessibility guidelines and evaluated the top 25 Google Play Store apps. Results show that while system-level accessibility features are present, there is a significant need for improvement in app design & content to enhance accessibility.	None Stated
[13]	This paper examines 1000 Android apps to uncover accessibility issues and subsequently conducts a developer survey to pinpoint the underlying causes of these accessibility challenges. The study underscores the prevalent issue of mobile accessibility unawareness among developers and emphasizes the lack of support from organizational management as a contributing factor to these issues. Additionally, the research challenges the assumption that app popularity and user ratings reliably indicate the level of accessibility features implemented in a mobile app.	RQ1: How prevalent are accessibility issues in Android apps? RQ2: What are the most common types of accessibility issues? Are specific categories of apps more susceptible to accessibility issues than others? RQ3: How does accessibility evolve in Android apps? RQ4: Do the same developers tend to create similar types of accessibility issues? RQ5: What are the underlying reasons for developing apps with accessibility issues RQ6: Do apps developed by large and well-known companies have better inaccessibility rate than other apps? RQ7: Do developers perceive all accessibility issues equally? RQ8: What accessibility issues do users complain about? RQ9: Do accessibility issues have any association with app ratings?
[25]	This survey paper offers a detailed overview of automated accessibility evaluation tools, explaining their abilities. The authors emphasize a lack of support for WCAG accessibility guidelines in these tools as a common deficiency.	None Stated
[10]	In this study, the authors examined 5,753 Android apps to identify accessibility problems related to labels on image-based buttons. Their findings show significant accessibility issues in both label and image-based buttons within these apps.	None Stated
[12]	This study analyzed 479 Android apps using the Mobile Accessibility Checker (MAC) to find accessibility issues. Also, the research introduced two new accessibility metrics: the Inaccessible Element Rate (IAER) and the Accessibility Issue Rate (AIR).	None Stated
[14]	These researchers explore current practices and guidelines surrounding how developers implement accessibility features in Android apps. They find that developers ignore most guidelines.	RQ1: How are existing accessibility guidelines implemented in mobile applications? RQ2: What are the developer's take on implementing accessibility guidelines in mobile applications?
[34]	The examination of ten popular mobile apps from the Google Play Store using WCAG 2.1 guidelines reveals that many apps in this study fail to meet the minimum standards outlined by WCAG 2.1.	None Stated

Continued on next page

TABLE I – continued from previous page

Ref.	Summary	Research Questions
[35]	This paper looks at sized-based accessibility issues. They present and evaluate a novel automated approach for repairing such issues that successfully resolve 99 percent of issues.	RQ1: How effective is our approach in repairing SBIIs in Android applications? RQ2: How long does it take for our approach to generate repairs for SBIIs in Android applications? RQ3: How does our approach impact the visual appeal of Android applications after applying the selected repair?
[36]	This publication introduces Xbot, an automated tool designed for mobile page exploration. Xbot is capable of conducting accessibility tests and automatically identifying accessibility issues. The study's results indicate that Xbot outperforms existing tools, such as Google Monkey, particularly in automated accessibility issue detection.	RQ1: Can Xbot outperform the existing methods on app page exploration and issue collection when conducting accessibility testing? RQ2: What is the overall severity status of app accessibility at the issue level for closed-source and open-source apps at the issue level? RQ3: What are the in-depth relations between the accessibility issue types and app category, GUI component? RQ4: What are the quantitative characteristics of specific issues such as text or image contrast issues? RQ5: How many accessibility issues have been fixed during app version updates?
[37]	This paper introduces a novel automated test generation technique known as Mobile Accessibility Testing (MATE), which is designed to detect accessibility issues dynamically. Empirical results demonstrate that MATE outperforms several state-of-the-art accessibility detection tools.	RQ1: How effective is MATE's state abstraction? RQ2: How does MATE compare to static analysis with Android Lint? RQ3: How does MATE compare to the accessibility testing frameworks relying on existing tests? RQ4: Can MATE find accessibility flaws that cannot be avoided by enabling Android's assistive features?
[41]	The study performed a meta-analysis on 38 accessibility issues. They conclude that automated assessments are inadequate in covering these problems and should be supplemented with user tests and expert-based inspections.	RQ1: Among the problems identified in accessibility evaluations, what are issues found by any combinations of methods? RQ2: What are the benefits and limitations of each method for evaluating accessibility on the web? RQ3: What are the benefits and limitations of each method for evaluating accessibility on mobile platforms?
[15]	This study aims to define and categorize human-centered problems in mobile apps. They developed an ML model to do future classification based on their taxonomy.	RQ1: What end-user human-centric issues typically manifest in mobile apps? RQ2: Can we accurately and automatically classify end-user human-centric issues from developer discussions and app reviews? RQ3: Do practitioners perceive that the automated classification of end-user human-centric issues in apps is useful?
[16]	This team investigates how CPDFs support mobile apps that cater to visually impaired people. They find many shared accessibility APIs in iOS and Android, but most are unavailable in popular CPDFs.	None Stated
[31]	This article attempts to understand Android apps' "dark mode" provision. They look at 300 discussion threads and 6000 apps, then distill key lessons for Android mobile developers.	RQ1: What dark mode aspects are discussed by developers in Stack Overflow? RQ2: What are dark mode features found in real-world Android applications? RQ3: How well do app development support tools aid dark vs. light mode development?
[32]	This study mines over 1.5 million Stack Overflow discussions and uses LDA to help categorize the text. They associate ten core strategies to support developer governance.	RQ1: What can be understood from the three main MSECOs (i.e. Android, iOS, and Windows Phone) based on technical questions at Stack Overflow?

Appendix B
LDA model outputs with four topics for bi-grams

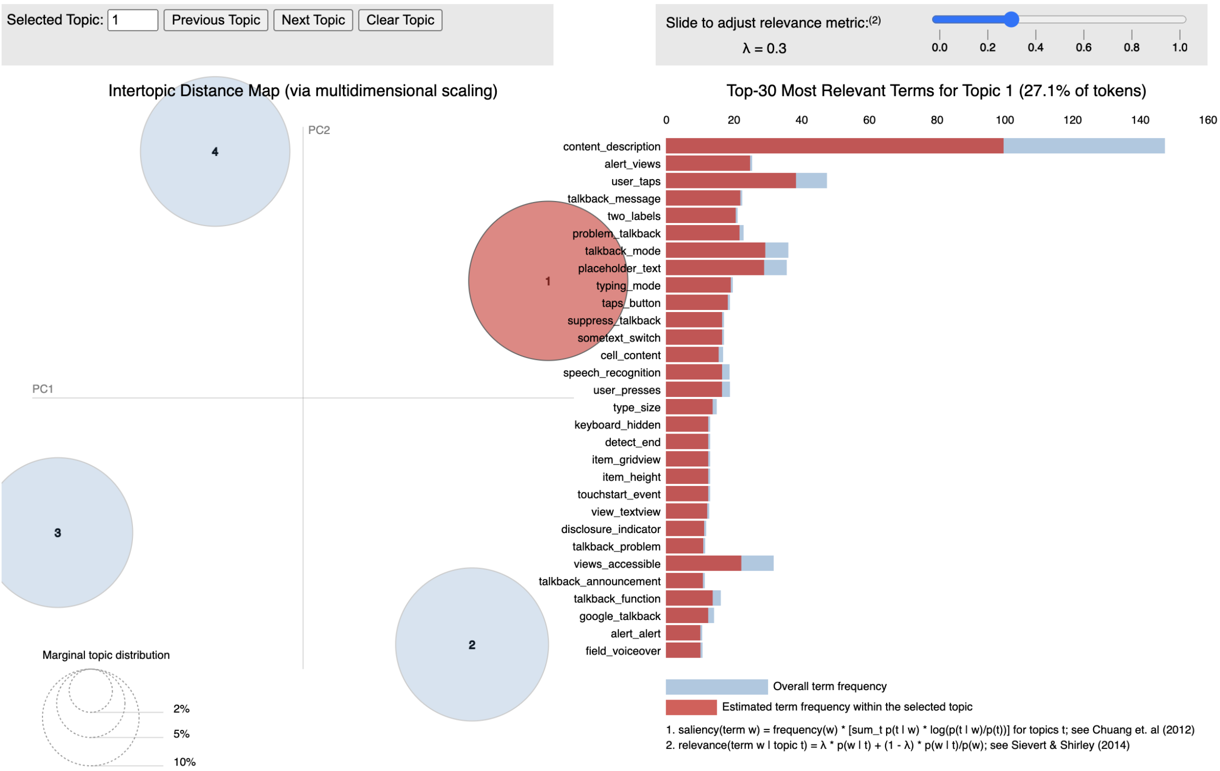

Fig. 14. Topic 1 of LDA model output

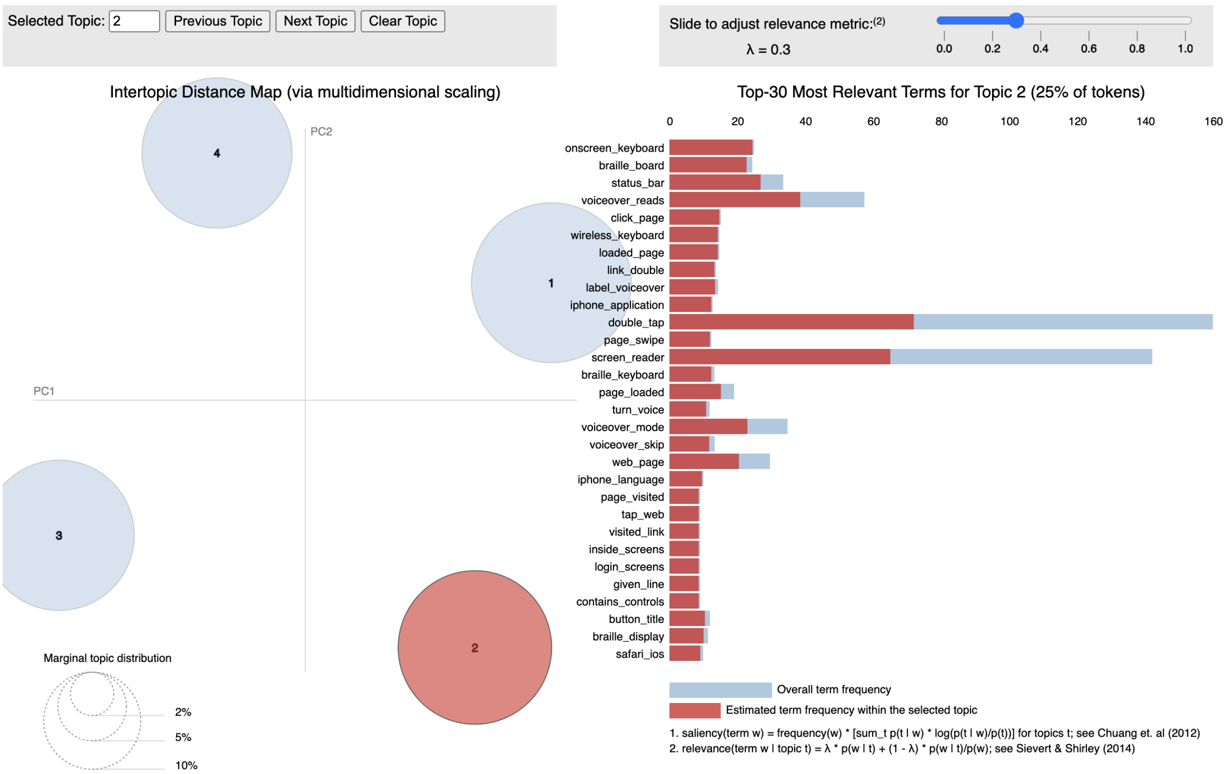

Fig. 15. Topic 2 of LDA model output

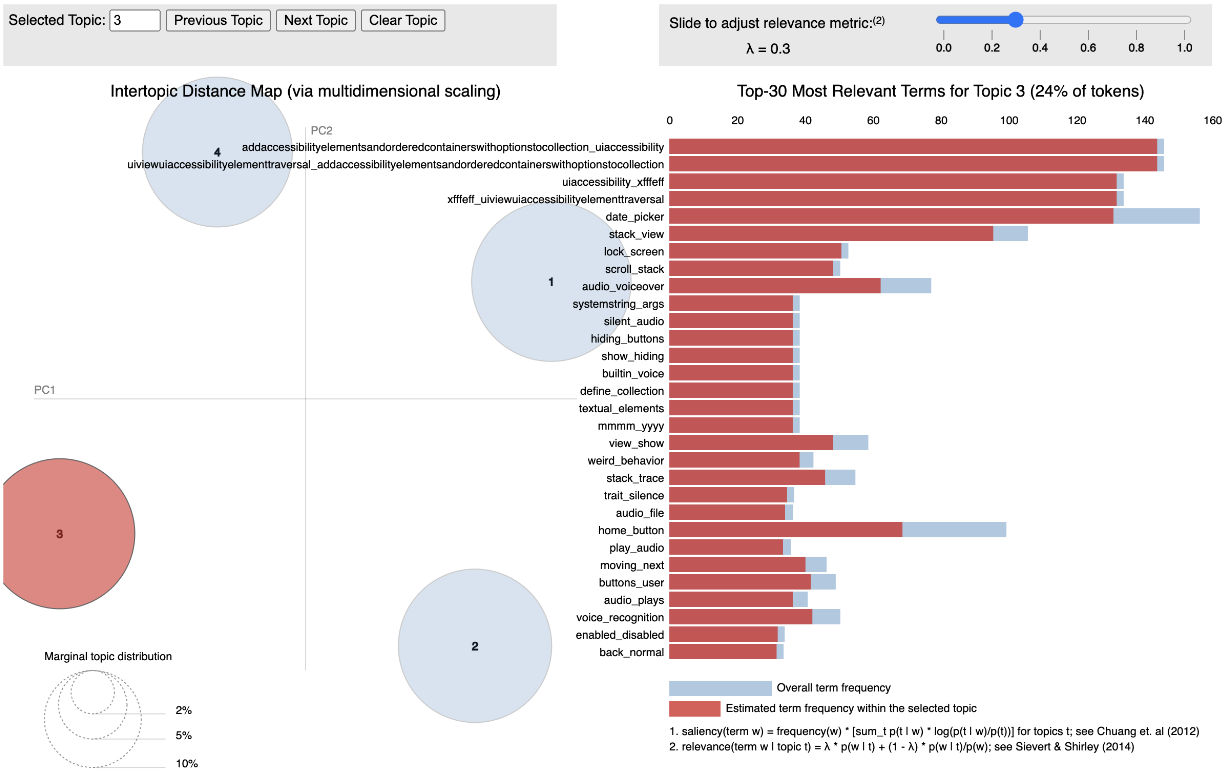

Fig. 16. Topic 3 of LDA model output

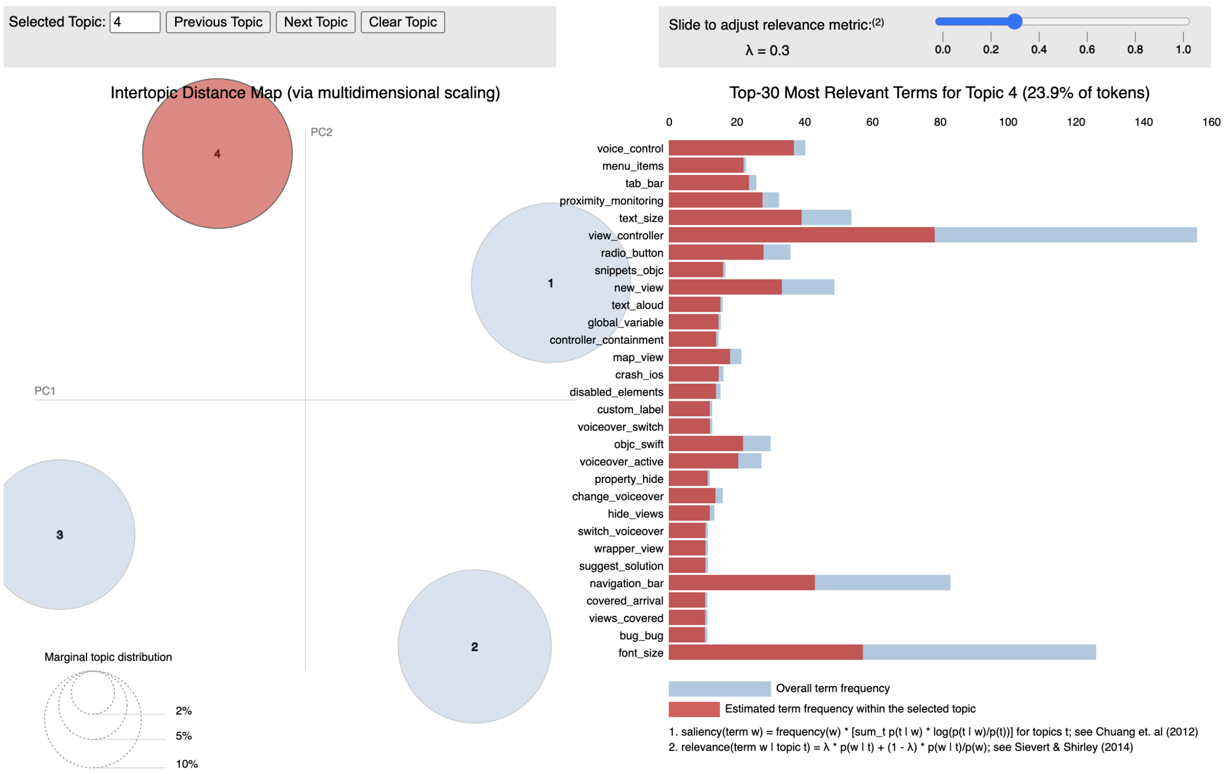

Fig. 17. Topic 4 of LDA model output