Automatic Assistance to Mitigate Rollback Inconsistencies in Collaborative Edits

Saikat Mondal
saikat.mondal@usask.ca

University of Saskatchewan https://orcid.org/0000-0003-1767-6392

Gias Uddin
York University

Chanchal K. Roy
University of Saskatchewan

Research Article

Keywords: Stack Overflow, inconsistent edits, content quality, user study, tool support

Posted Date: January 16th, 2025

DOI: https://doi.org/10.21203/rs.3.rs-5830055/v1

Additional Declarations: The authors declare no competing interests.

Automatic Assistance to Mitigate Rollback Inconsistencies in Collaborative Edits

Saikat Mondal · Gias Uddin · Chanchal Roy

Received: date / Accepted: date

Abstract The success of technical Q&A sites such as Stack Overflow depends on two key factors: (a) active user participation and (b) the quality of the shared knowledge. Stack Overflow introduced an edit system that allows users to suggest improvements to posts (i.e., questions and answers) to enhance the quality of the content. However, users, such as post owners or site moderators, can reject these suggested edits by rollbacks due to unsatisfactory, low-quality edits or violating edit guidelines. Unfortunately, subjectivity bias in determining whether an edit is satisfactory or unsatisfactory can lead to inconsistencies in the rollback decisions. For example, one user might accept the formatting of a method name (e.g., `getActivity()`) as a code term, while another might reject it. Such inconsistencies can demotivate and frustrate users whose edits are rejected. Furthermore, several post owners prefer to keep their content unchanged and even resist necessary edits. As a result, they sometimes roll back necessary edits and revert posts to a flawed version, which violates editing guidelines. The problems mentioned above are further compounded by the lack of specific guidelines and tools to assist users in ensuring consistency in user rollback actions. In this study, we investigate the types, prevalence, and impact of rollback edit inconsistencies and propose a solution to address them. The outcomes of this research are fivefold. First, we manually investigated 764 rollback edits (382 questions + 382 answers) and identified eight types of inconsistent rollback. Second, we surveyed 44 practitioners to assess the impact of rollback inconsistencies. More than 80% of the participants found our identified inconsistency types detrimental to post quality. Third, we de-

Saikat Mondal
Software Research Lab, Department of Computer Science
University of Saskatchewan, Canada
E-mail: saikat.mondal@usask.ca

Gias Uddin
Data Intensive Software Analytics (DISA) Lab, Department of Electrical and Software Engineering
University of Calgary, Canada
E-mail: gias.uddin@ucalgary.ca

Chanchal Roy
Software Research Lab, Department of Computer Science
University of Saskatchewan, Canada
E-mail: chanchal.roy@usask.ca

veloped rule-based algorithms and Machine Learning (ML) models to detect the eight types of rollback inconsistencies. Both approaches achieve over 90% accuracy. Fourth, we introduced a tool, `iEdit`, which integrates these algorithms into a browser extension and assists Stack Overflow users during their edits. Fifth, we surveyed 16 Stack Overflow users to evaluate the effectiveness of `iEdit`. The participants found the tool's suggestions helpful in avoiding inconsistent rollback edits.

Keywords Stack Overflow, inconsistent edits, content quality, user study, tool support

1 Introduction

Stack Overflow is the most popular crowd-sourced technical knowledge-sharing platform. The adoption, growth, and continued success of a crowd-sourced Q&A website like Stack Overflow depend on the (a) active participation of users (e.g., software developers) and (b) quality of the shared knowledge (Bagozzi and Dholakia, 2006; Lakhani and von Hippel, 2003; Parnin et al, 2012). Unlike traditional knowledge-sharing venues and stakeholders (e.g., paid events), users' participation in Stack Overflow is driven by three main factors — (a) opportunity to learn from others (which is free), (b) desire to be part of the community, and (c) recognition from peers within the community. Therefore, the roles of knowledge seekers and providers in Stack Overflow are often fluid and interchangeable. Such elasticity leads to the design of a semi-decentralized system. However, the lack of authoritativeness can raise concerns of trust in the quality of the shared contents (Uddin et al, 2019).

Stack Overflow introduces a collaborative editing system to promote quality by allowing its users to suggest improvements to the posts (Li et al, 2015; Kittur and Kraut, 2008; Dabbish et al, 2012). In particular, such collaborative editing aims to keep posts clear, relevant, and up-to-date. For example, users often edit posts to fix grammatical and spelling mistakes, clarify meanings, and add missing resources (e.g., code snippets) or hyperlinks. However, these suggested edits can be rejected by *rollbacks* by users (e.g., post owners or site moderators) due to unsatisfactory or low-quality edits or violating edit guidelines. A rollback reverts a post to a previous version in the edit history (StackExchange, 2009c) and thus rejects one or more edits. Then, the reverted version appears as the latest item in the edit history. Wang et al. (Wang et al, 2018) manually analyzed 369 rollback edits and identified twelve reasons (e.g., undesired text formatting) behind these rollbacks of suggested edits. The initial goal of our study was to develop classifiers to detect those rollback reasons automatically. However, when we carefully investigated those reasons, we found several *inconsistencies* behind the rollback edits.

In this study, we define a suggested edit as inconsistent if it is evaluated differently by two users. We consider *rollbacks* as *inconsistent* (user-based) when a rollback rejects or accepts (i.e., brings back) inconsistent edits. It should be noted that these rollbacks can occur within the edit history of a single post or across different posts. In addition to user-based inconsistencies, we found that a single rollback rejects multiple accepted edits simultaneously. On the contrary, sometimes, it rejects none. Such an inability to regulate the rollbacks also introduces inconsistencies. We name it system-based inconsistency. We define *rollbacks* as *inconsistent* (system-based) when a rollback rejects multiple accepted edits simultaneously or none. It should be noted that these rollbacks happen in the edit history of a single post (further details are provided in Section 2).

Users edit Stack Overflow posts voluntarily, and these edits should be evaluated according to Stack Overflow's editing guidelines or benchmarks. Consistent judgment is essential for the voluntary participation of the users toward progressing a community-driven site like Stack Overflow. Users get frustrated and demotivated when similar edits are assessed differently, especially those whose edits get rejected. Similarity of edits refers to how closely content, structure, intent, or actions are aligned between two or more edits (further details are provided in Section 2). However, inconsistent practices can hinder users' future engagement and contribution. Moreover, inconsistent decision-making can lead to conflicts, disagreements, and a perception of unfairness among users. It can lead to a loss of confidence in the system. Users cannot trust a platform that is inconsistent and unpredictable. Inconsistent edits can spread confusion and misunderstandings about what decision is accurate.

We found a number of questions posted at MetaStackExchange[1] and MetaStackOverflow[2] that asked for guidance to make decisions when assessing suggested edits. For instance, a question posted on MetaStackExchange (question # 2950) (StackExchange, 2008) to seek opinions on whether users should reject expressions of gratitude and greetings (e.g., Hello). At least $1,124$ users voted up (i.e., score $= 1,124$), and $121K$ users viewed this question. Such high popularity suggests that similar confusion is prevalent among users. Additionally, we found posts that raised concerns about a few more inconsistencies, such as acceptance/rejection of status updates (e.g., EDIT: ...) (StackExchange, 2013b) and signatures (e.g., Regards, Gustavo) (StackExchange, 2013a). These inconsistencies can adversely affect users' engagement by confusing them about editing and rollback guidelines (StackExchange, 2016, 2017, 2018, 2015; StackOverflow, 2015b,a; StackExchange, 2012c,a, 2009b, 2013b, 2012b, 2013a). Given the evidence above, such inconsistencies lead to several adverse outcomes that undermine the integrity of the site.

Most worryingly, inconsistencies can hurt the quality of valuable content on Stack Overflow. In particular, inconsistent rollback edits can lead to incorrect post content. Fig. 1 shows a rollback edit with temporal inconsistency, a system-based inconsistency where multiple accepted edits are rejected by a single rollback. In this example, the rollback reverts an answer from revision #10 (Fig. 1(a)) to 1 (Fig. 1(b)). Thus, it rejects eight (i.e., two to nine) accepted edits simultaneously. In particular, this rollback removed the second line of the code from the code segment, which had been added to remove the left padding from a UITextView. The question submitter requested code to remove all paddings from UITextView. Therefore, the rollback made the solution incomplete, failing to remove the left padding from UITextView. One user thus commented – *"add this line too to remove left padding self.textView.textContainer.lineFragmentPadding = 0;"* (Fig. 1(c)). At least 62 users (i.e., score = 62) acknowledged it as a valuable comment. Unfortunately, the incorrect answer remained live for about nine months (June 29, 2016 – March 28, 2017) before it was edited again to include the second line of code.

[1] https://meta.stackexchange.com

[2] https://meta.stackoverflow.com

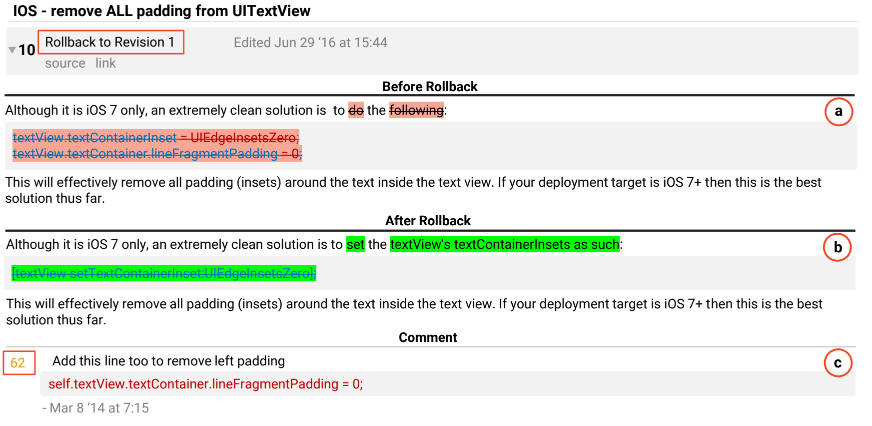


Fig. 1: An example of an inconsistent rollback edit
(https://stackoverflow.com/posts/20269793/revisions)

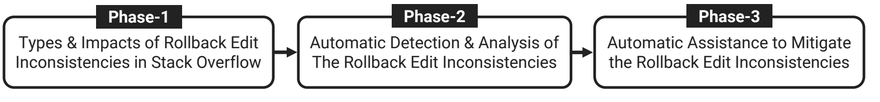


Fig. 2: Phases of our study

The above observations motivated us to shift our focus from developing classifiers to detect rollback edit reasons **to** develop solutions for automated analysis and detection of inconsistencies in Stack Overflow rollback edits. We divided our study into *three* phases (P) as shown in Fig. 2. Across the three phases, we answered a total of *eight* research questions. We summarize the three study phases and their findings as follows.

(P_1) **Types and Impacts of Rollback Edit Inconsistencies in Stack Overflow (3 Research Questions: RQ1 - RQ3).** The motivating scenarios discussed above highlight the presence of inconsistencies in Stack Overflow rollback edits. However, to properly support the Stack Overflow editing process, we need a catalog of all possible inconsistency types. We conduct a qualitative study of 764 rollback edits (382 questions + 382 answers) to identify inconsistent rollback edits. We particularly examine whether (1) each suggested edit that was rolled back also contributed to an accepted edit and (2) rollback rejects multiple accepted edits or none. We manually label the reasons for these inconsistencies. This study produced a catalog of eight inconsistency types under two categories – (1) user-based and (2) system-based inconsistencies. User-based inconsistencies include *presentation* (different presentation styles of similar text or code elements), *gratitudinal* (inconsistent actions of rejecting or accepting expressions of gratitude), *signature* (inconsistent actions of adding or removing signatures, such as user names), *status* (inconsistent actions of rejecting or accepting personal notes or status), *deprecation* (inconsistent actions of rejecting or accepting deprecation notes) and *duplication* (inconsistent actions of rejecting or accepting duplication notes) inconsistencies. System-based inconsistencies include *temporal* (rejecting multiple accepted edits simultaneously) and *structural* (reverting post to the immediate previous version) inconsistencies.

The motivating scenarios also provide exploratory viewpoints on the negative impact of edit inconsistencies. We further sought to capture quantitative evidence, and Stack Overflow user perspectives on the negative impacts of the eight inconsistency types. First, we divided the posts into two groups – (1) posts that suffered from at least one of the eight inconsistencies and (2) posts that did not. Then, we compute several Stack Overflow post popularity metrics (e.g., score, favorite count) for each group. We find that the posts that have undergone inconsistent rollback edits are significantly less popular than those that did not go through such inconsistent edits. Second, we surveyed 44 developers from Stack Overflow to capture their perspectives on the eight inconsistency types. More than 80% of the participants agree that inconsistent rollback edits negatively impact post quality.

(P_2) **Automatic Detection and Analysis of The Rollback Edit Inconsistencies (3 Research Questions: RQ4 - RQ6).** We develop eight rule-based algorithms that achieve 99% overall accuracy in identifying the eight inconsistency types. However, the recall to identify the signature inconsistency is comparatively lower. Upon further investigation, we find that users sometimes add names different from their profile names, which reduces recall. Additionally, we introduce ML models to classify the inconsistent and consistent rollback edits. According to the experiment, our models perform similarly to rule-based algorithms. The precision ranges from 97% to 100% in detecting inconsistency types. Such results confirm the strength of the textual pattern employed in our rule-based algorithms.

We then compute the prevalence of inconsistent edits across the entire September 2019 Stack Overflow data dump. Our analysis reveals that temporal inconsistency is the most frequent system-based inconsistency, whereas presentation and status inconsistencies are the most prevalent among user-based inconsistencies. We also investigate which types of users (e.g., new users) are more likely to commit inconsistent rollback edits. Our findings indicate that nearly all user types are involved in committing inconsistent rollback edits.

(P_3) **Automatic Assistance to Mitigate the Rollback Edit Inconsistencies (2 Research Questions: RQ7 and RQ8).** Our empirical and user studies show that the eight edit inconsistency types are detrimental to the quality of Stack Overflow posts and discourage users from participating in editing. Therefore, it is crucial to alert Stack Overflow users about such inconsistencies while suggesting edits. We thus plan to introduce tool support to identify them automatically. However, before introducing tool support, we conduct a preliminary study to understand its necessity and design requirements. According to the responses, 91% of the participants agreed that such a tool would be beneficial, helping users avoid inconsistent edits. In particular, they expressed interest in a browser plugin that analyzes suggested edits and assists users in avoiding inconsistent edits.

Fig. 3 provides the overview of our tool, `iEdit`. On the client side, we utilize Tampermonkey[3], the most popular userscript manager. It offers an effortless

[3] `https://www.tampermonkey.net`

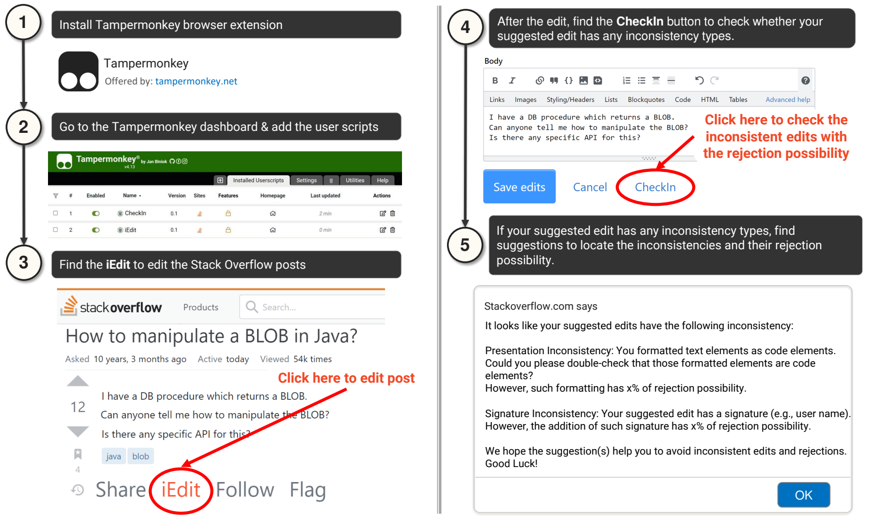


Fig. 3: An overview of the `iEdit` workflow

way to manage userscripts. Tampermonkey is available as a browser extension for nearly all major browsers, including Chrome, Firefox, Safari, Microsoft Edge, and Opera. Tampermonkey enables users to add JavaScripts that modify web pages. However, we add two JavaScripts for the *iEdit* and *CheckIn* interface. iEdit enables users to edit the posts, whereas CheckIn captures the texts before and after edits, along with user information, and transmits them to the server-side application. The server-side application comprises inconsistency detection algorithms. Finally, the tool alerts users and shows the likelihood of rejection if the suggested edits contain inconsistencies.
We then survey 16 developers to assess the effectiveness of `iEdit`. Most users found it easy to install and use with minimal cognitive effort. Moreover, they considered the suggestions from `iEdit` very influential.

The findings from our study can guide (a) **Forum Designers** to improve the edit system, (b) **Forum Users** to shape their edit behavior, and (c) **Researchers** to study collaborative editing.

Contributions. This paper is a significantly extended version of our previous study (Mondal et al, 2021b) that investigated the types of rollback edit inconsistencies in Stack Overflow and their impact on content quality. In the previous study, we answered four research questions (RQ1 - RQ4) and reported four key findings. However, this study extends our previous study in several aspects and answers eight research questions, including four additional questions (RQ5 - RQ8). *First,* we incorporated ML techniques alongside our rule-based inconsistency detection algorithms to assess – (1) the strength of our textual patterns in discriminating between consistent and inconsistent rollback edits and (2) whether machine learning models outperform the rule-based techniques. *Second*, we investigate which types of users, such as new users, are more prone to committing inconsistent rollback edits. *Third*, we survey Stack Overflow users to assess their need for a tool that can automatically detect inconsistent edits. Additionally, we offer several tool support options and ask for their feedback. *Fourth*, we introduce an online, user-friendly tool called `iEdit`, designed to integrate with Stack Overflow's existing edit system. `iEdit` analyzes text before and after edits, along with user information (e.g., name), to identify inconsistent edits. It then alerts users to the types of inconsistencies and their likelihood of rejection. *Finally,* we conducted a survey of developers to assess the effectiveness of `iEdit`.

Structure of the Article. Section 2 provides the background of our study and defines key terms. Section Section 3 explores the catalog of rollback inconsistency types and their impacts. Section 4 presents the rule-based and ML techniques used for detecting inconsistencies, along with their performance and the distribution of inconsistent rollback edits across different user types. Section 5 introduces `iEdit` and evaluates its effectiveness. The reasons for inconsistent rollback edits and the implications of our findings are discussed in Section 6. Section 7 addresses the threats to validity, Section 8 reviews related work, and Section 9 concludes the study.

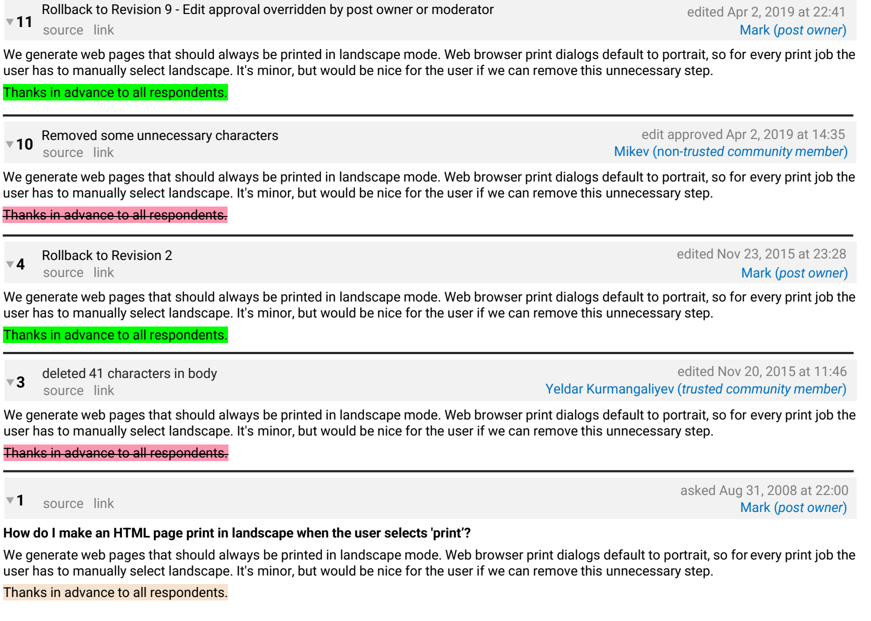


Fig. 4: An example of gratitudinal inconsistency (`https://stackoverflow.com/posts/37162/revisions`).

2 Background

In this section, we provide the background of our study and formally define key terms. The definitions and the discussions form the basis of our study and the development of our `iEdit` tool.

Inconsistent edit. We define a suggested edit as *inconsistent* if it is evaluated differently by two users. Strictly speaking, collaborative editing platforms like Stack Overflow offer guidelines on how to (a) edit a post content (a.k.a., 'suggested edit') and (b) assess the suggested edit. In addition, certain editing principles can be established within the editing community for suggesting an edit and assessing it for acceptance or rejection. However, ignorance of editing guidelines or different interpretations of these principles could lead to inconsistency in suggesting or assessing edits. Furthermore, users' preference to keep the content of their posts unchanged can be another potential reason for such inconsistencies. Broadly speaking, concerns about inconsistency in editing arise when similar edits are judged differently by different evaluators (e.g., Stack Overflow users). For instance, Bob might accept expressions of gratitude (e.g., Thanks), while Alice rejects them. This scenario introduces an inconsistency in what we call 'Gratitudinal Inconsistency.'

Consider a few revisions in the revision history of post ID 37162, as shown in Fig. 4. Mark posted a question and added gratitude, *"Thanks in advance to all respondents"* at the end of the question (revision # 1). Yeldar Kurmangaliyev, a trusted community member, rejected the gratitude (revision #3). However, the post owner brought back the gratitude by rolling back the post to revision #2 (revision #4). Then, Mikev suggested an edit to remove gratitude. The edit got approved by the expert review (revision #10). Surprisingly, the post owner overridden the approved edit to bring back the gratitude (revision #11). Although Stack Overflow guidelines on conducting good edits (StackOverflow, 2015a) instruct users to remove expressions of gratitude, it is clear that users approach this guideline differently. Therefore, we identify edits that accept or reject gratitude as inconsistent.

Similarity of edits. Similarity of edits refers to how closely two or more edits align in terms of structure, intent, content, or action taken.

– *Structure.* How the text or code is organized, such as formatting changes. Presentation inconsistencies were identified based on these structural aspects.
– *Intent.* The purpose behind the edits, such as updating content or addressing any concerns. Inconsistencies related to deprecation, duplication, and status were identified based on these intents.
– *Content.* The specific changes made to the text, such as additions or deletions. Inconsistencies related to gratitude and signature were identified based on these content changes.
– *Actions.* The decision or operation performed during the rollback process, such as the number of revisions rejected. Based on these actions, temporal and structural inconsistencies were identified.

Editing privilege of users. Any registered users of Stack Overflow can suggest edits. However, edits conducted by post owners or trusted community members (i.e., users with a *reputation score* $\geq 2K$) become publicly visible immediately. On the other hand, edits suggested by users with less than a $2K$ reputation are placed in a review queue.

Edit rejection. Suggested edits to the Stack Overflow posts can be rejected in the following two ways.

– *Expert review.* Experts (i.e., trusted community members) review suggested edits that are placed in the review queue. A maximum of three members review each post's suggested edits. After review, they cast either *accept* or *reject* votes to the suggested edits. The suggested edits are rejected and thus not applied to posts if two reject votes are cast. On the contrary, two accept votes are required to apply the edits to the posts.

– *Rollback.* Rollback can reject the suggested edits that were already applied to the posts. In particular, rollback reverts a post to a previous version in the edit history (StackExchange, 2009d) and thus rejects one or multiple revisions (e.g., Fig. 4, revision #11). The rollback action appears to be the most recent item in the edit history. For example, a user rollbacks a post from revision #10 to #9. Then, the content of revision #9 will be the most recent item in the edit history with a new revision #11. However, the suggested edits applied to revision #10 will be rejected by this rollback. Post owners and trusted community members are eligible to roll back posts.

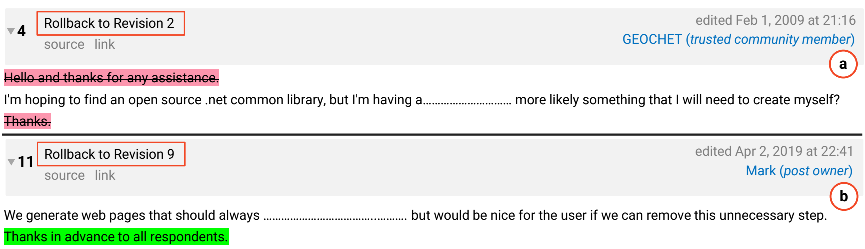


Fig. 5: Gratitudinal inconsistency (Rollback edits were selected from the revision history of posts with IDs 498249 and 37162).

Inconsistent rollback edit. We found two kinds of rollback inconsistency: user-based and system-based rollback inconsistencies. We define *rollbacks* as *inconsistent* with user-based inconsistencies when a rollback rejects or brings back (i.e., accepts) inconsistent edits. For example, Fig. 5 (a) shows a rollback from the revision history of post ID 498249 (StackOverflow, 2008a) that rejects gratitude. On the contrary, Mark rolled back a post from revision #11 to revision #9 (Fig. 5 (b), revision of post ID 37162 (StackOverflow, 2008c)) to bring back gratitude. In this study, we identify both rollbacks as inconsistent since they reject/accept inconsistent edits. Whether the rollback occurs in the same post or a different one, it is identified as an inconsistent rollback edit if it rejects or accepts inconsistent edits. On the other hand, we define *rollbacks* as *inconsistent* with system-based inconsistencies when a rollback rejects multiple accepted edits simultaneously (Fig. 1) or none (Fig. 12).

3 P1: Types and Impacts of Rollback Edit Inconsistencies

In this section, we answer the following research questions:

RQ1. What types of inconsistencies exist in rollback edits on Stack Overflow?

RQ2. Is there a correlation between post quality and inconsistent rollback edits, and do inexperienced users tend to post questions with inconsistent rollback edits?

RQ3. What is the perceived impact of the observed rollback edit inconsistency types?

3.1 What types of inconsistencies exist in rollback edits on Stack Overflow? (RQ1)

As mentioned in Section 1, inconsistencies exist in Stack Overflow rollback edits. This section summarizes the methods used to identify these inconsistent rollback edits and classify their types.

3.1.1 Approach

We conduct a qualitative study of 764 Stack Overflow rollback edits to produce our catalog of inconsistency types. The study is organized into four steps, as outlined below.

(1) **Data Collection.** We downloaded the September 2019 Stack Overflow data dump (StackExchange, 2019b), which was the latest data dump when we started the original study (Mondal et al, 2021b). It contains all the essential information about questions and answers. It stores the history of all the events (e.g., edit body, rollback body, post deleted) for each post. It also includes the date of each event and the user who triggered it.

(2) **Data Preprocessing.** Stack Overflow tracks 38 types of events (StackExchange, 2019a). In this study, we focus on revisions where suggested edits to the body of posts (i.e., $PostHistoryTypeId = 8$) were rejected by rollbacks. Edits can be rejected either by rollbacks or expert reviews. In our September 2019 data dump, out of 26.4 million edits related to the body of posts, 102,289 were rejected solely by rollbacks. We focus on body edits because they contain the majority of the post's content. Our dataset includes 63,071 rollback revisions of questions and 39,218 rollback revisions of answers where rollbacks rejected body edits.

(3) **Random Sampling of Data for Qualitative Analysis.** To achieve a 95% confidence level with a 5% confidence interval (Boslaugh, 2012; Wang et al, 2018), we randomly sampled 382 from 63,071 rollback revisions of questions and 382 from 39,218 rollback revisions of answers. Please note that the original sample size for answers was 380, but we decided to equalize it to 382 for a balanced analysis across both types of revisions. The sample size was computed using the following standard formula: $\frac{Nz^2p(1-p)}{e^2N+z^2p(1-p)}$, where N is the population size (e.g., 63,071), z is the $Z-score$ corresponding to a particular confidence level (e.g., 1.96 for a confidence level of 95%), e is the confidence interval (e.g., 5%), and p is population proportion (e.g., 0.5).

(4) **Qualitative Analysis to Identify Rollback Inconsistencies.** The first two authors of this paper initially analyzed the rollback edit reasons identified by Wang et al. (Wang et al, 2018). During our investigation, we found several inconsistencies, where similar edits were rejected somewhere and also brought back (i.e., accepted) somewhere by rollbacks. We, the two investigators, discuss the

rollback inconsistencies in multiple interactive sessions. We then randomly selected 100 question rollbacks and 100 answer rollbacks from the previously chosen 382 question and 382 answer rollback revisions. These were manually analyzed and labeled for inconsistencies. For a given rollback edit, we meticulously analyzed – (1) the history of all suggested edits of the post before the rollback edit and (2) our list of common actions that are rejected and accepted by rollback edits. A rollback is categorized as inconsistent if it exhibits characteristics similar to an accepted edit. We repeated the labeling process to form higher categories. This manual investigation produced a catalog of eight inconsistency types.

We then measure the agreement using Cohen's Kappa (Cohen, 1968, 1960), which resulted in a value of κ was 0.98, showing almost perfect agreement. We resolve the remaining few disagreements by discussion. Then, the first author independently labeled the remaining samples (282 question revisions and 282 answer revisions) without introducing bias, as the agreement was nearly perfect.

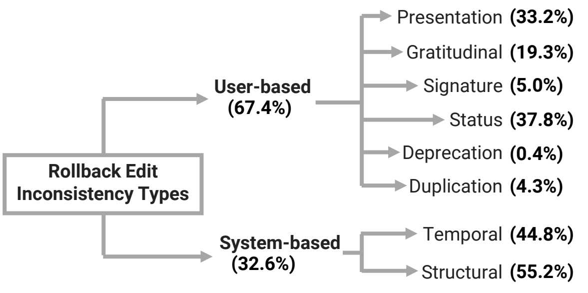


Fig. 6: Rollback inconsistency types and their distribution in our manually analyzed dataset.

3.1.2 Results

Fig. 6 summarizes the inconsistency types. Our manual analysis identifies eight types of inconsistencies under two categories: user-based and system-based. User-based inconsistencies arise due to users' subjective biases when rejecting edits by rollbacks. On the other hand, system-based inconsistencies appear due to the inability of the edit system itself to regulate the rollbacks. Approximately 41% of the edits in our dataset contain one or more inconsistencies. A single rollback may have multiple inconsistencies. Among all the inconsistent rollback edits, 79.8% exhibit a single inconsistency, while 20.2% involve more than one. The details of these rollback inconsistencies are discussed below.

User-based Inconsistencies comprised about 67.4% of all inconsistencies in our analysis (Fig. 6). We discuss the six types of user-based inconsistencies below.

(1) ***Presentation Inconsistency.*** Presentation inconsistency refers to the different presentation styles of similar text or code terms. Consider the example shown in Fig. 7, where the first rollback (StackOverflow, 2011b) (Fig. 7 (a)) was made to put the method names (e.g., `getActivity()`) inside the code tags (`` in markdown denotes a $< code >< /code >$ tag). On the contrary, the second

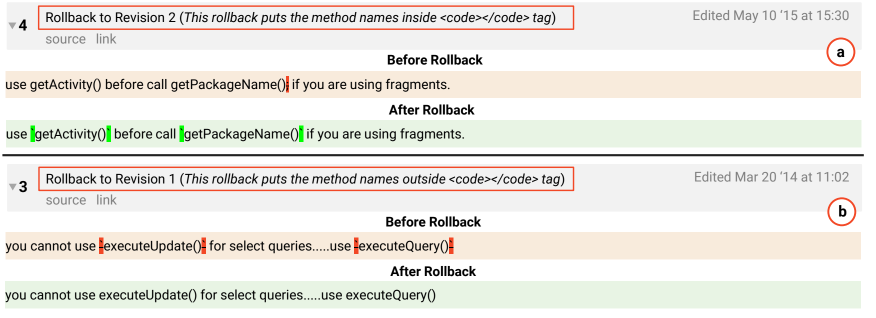


Fig. 7: Presentation inconsistency (Rollback edits were selected from the revision history of posts with IDs 30150411 and 22529397).

rollback (StackOverflow, 2012a) (Fig. 7 (b)) was made to reject a suggested edit where method names (e.g., `executeUpdate()`) were put inside the code tags. Such inconsistent rollbacks certainly confuse the users. In another case, a user named Yi Jiang expressed frustration when his suggested code formatting was rejected, and the answer was reverted to a faulty version. Then, he complained, *"Why did you rollback my edit? Your answer is incorrectly formatted, and you spelled 'position' wrong."* (StackOverflow, 2010). Such presentation inconsistency also includes changing the cases of texts (e.g., uppercase/lowercase), the format of texts (e.g., bold/italic), adding or removing space/newline, creating bullet/number list, replacing acronym by its root words, or bringing the acronyms back, adding links as standard texts/hypertext. Users often reject or bring such styles back by rollback edits.

According to our investigation, formatting and unformatting the inline code terms are the most frequently seen presentation inconsistencies in rollback edits. In this study, we thus only consider them for convenience in processing. Proper and consistent formatting is essential not only for readability but also because it directly impacts searchability, data extraction, and analysis on Stack Overflow, as automated tools depend on correct HTML tags to identify specific content. Improper formatting can lead to missed examples in large-scale Software Engineering (SE) studies and degrade the quality of AI model training. Thus, standardizing formatting and content style is crucial for advancing SE by providing cleaner, more consistent datasets for research and tool development. As shown in Fig. 6, we find that 33.2% of rollback edits have presentation inconsistency among all the user-based inconsistent rollback edits.

(2) ***Gratitudinal Inconsistency.*** Gratitudinal inconsistency refers to the dual viewpoint of rejecting gratitudes. Consider the example, as shown in Fig. 5(a), where the gratitude (e.g., thanks for any assistance) was rejected by a rollback (StackOverflow, 2008a). Conversely, Fig. 5(b) shows a rollback where the gratitude (e.g., thanks in advance to all respondents) was brought back (StackOverflow, 2008c). Such inconsistency even confuses the users who have been editing many posts over the years. To mitigate confusion, they often ask questions and seek opinions from others (StackExchange, 2009b, 2012b). Many users argued that gratitude should be accepted. For example, Toast said, *"If the post has nothing else wrong with it and is just book-ended with "Hi/Thanks" then you can probably pass on the edit."*

(StackExchange, 2009b). However, many others expressed opposite opinions. For example, One user boldly disagreed with accepting gratitudes, *"I've always been against the greetings and salutations."* (StackExchange, 2009b). Thus, such inconsistency needs to be addressed and resolved. According to our investigation, among all the user-based inconsistencies, 19.3% of them have gratitudinal inconsistency (Fig. 6).

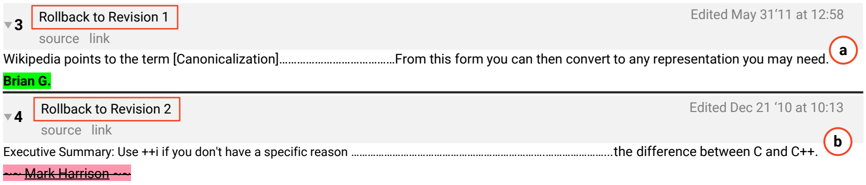


Fig. 8: Signature inconsistency (Rollback edits were selected from the revision history of posts with IDs 280121 and 24904).

(3) ***Signature Inconsistency.*** Users sometimes add their signatures (e.g., name, ID) at the bottom of the posts. In Stack Overflow, every post/edit is signed with a standard user card, which is linked directly to the user page. Many users thus suggest rejecting the signature. For example, one user said, *"If you use an additional signature or tagline, it will be removed to reduce noise in the questions and answers."* (StackExchange, 2009a). On the contrary, some users would like to sign their signatures when they significantly contribute to a post. During the manual investigation, we see that rollback edits rejected signatures in some cases (e.g., Fig. 8(b) (StackOverflow, 2008f)), while such signatures were brought back in some other cases (e.g., Fig. 8(a) (StackOverflow, 2008g)) by rollback edits. As shown in Fig. 6, 5% of user-based inconsistent rollback edits have signature inconsistency.

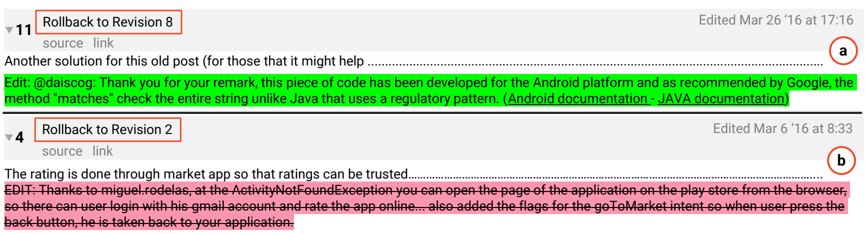


Fig. 9: Status inconsistency (Rollback edits were selected from the revision history of posts with IDs 11024200 and 11270668).

(4) ***Status Inconsistency.*** Users often add status (i.e., personal notes) to clarify others' confusion, append essential messages that were missed during the submission of a post, and acknowledge others' responses. For example, Fig. 9 presents two similar statuses, where users acknowledged the responders and added important notes. One status (e.g., Fig. 9 (a)) was brought back by a rollback (StackOverflow, 2008e). Unfortunately, a rollback rejected another status (e.g., Fig. 9 (b)) (Stack-

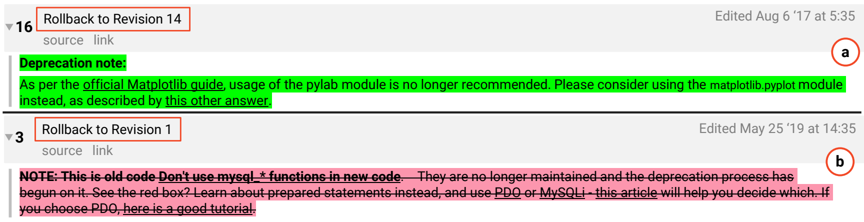


Fig. 10: Deprecation inconsistency (Rollback edits were selected from the revision history of posts with IDs 332311 and 14391452).

Overflow, 2012b). We also find conflicting opinions in many questions posted at meta Stack Exchange ((StackExchange, 2013b, 2012d)). One user said, *"EDIT and UPDATE are rarely needed, nor helpful. For future readers, posts need to be standalone, without any history"* (StackExchange, 2012d). However, some other users do not think that a status update is necessarily harmful. It could draw attention to new information. Such a dual viewpoint of accepting the personal status confuses and frustrates users who suggest edits. Fig. 6 shows that 37.8% of user-based inconsistent rollback edits have status inconsistency.

(5) ***Deprecation Inconsistency.*** Deprecation inconsistency refers to the dual viewpoints of rejecting deprecation notes. Fig. 10 (a) shows that one user brought the deprecation note back by a rollback (StackOverflow, 2008d). On the contrary, such a note was rejected by a rollback (e.g., Fig. 10 (b)) (StackOverflow, 2013b). However, such inconsistency is found infrequently in our selected dataset. Less than 1% user-based inconsistent rollback edits of posts have deprecation inconsistency (Fig. 6).

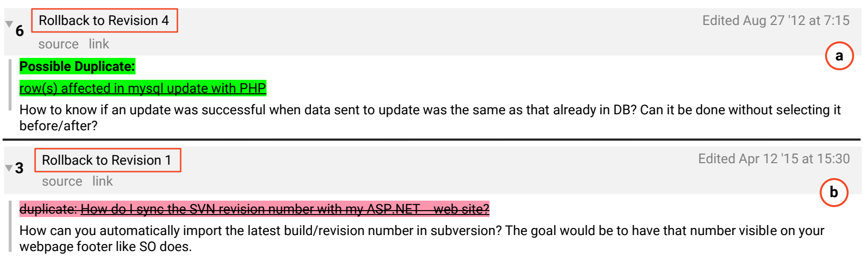


Fig. 11: Duplication inconsistency (Rollback edits were selected from the revision history of posts with IDs 6218171 and 110175).

(6) ***Duplication Inconsistency.*** Duplication inconsistency refers to the dual viewpoints of rejecting duplication notes. As you see in Fig. 11 (b), a duplication note was rejected from the body of a question by a rollback (StackOverflow, 2013a). On the contrary, such a note was brought back by rollbacks (e.g., Fig. 11 (a) (StackOverflow, 2011a)). About 4.3% user-based inconsistent rollback edits (of questions) have duplication inconsistency (Fig. 6).

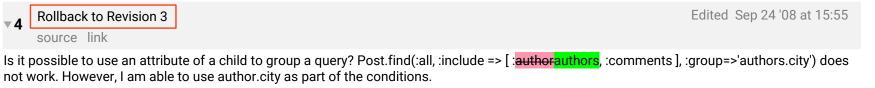


Fig. 12: Structural inconsistency (Rollback edit was selected from the revision history of the post with ID 125523) .

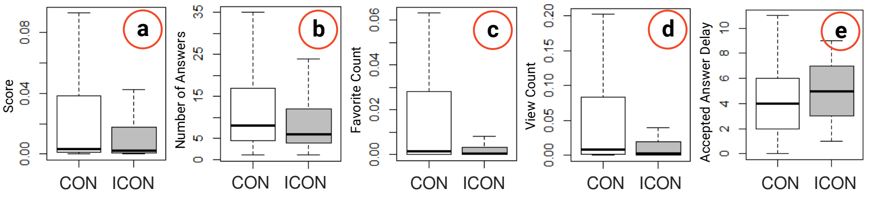


Fig. 13: (a) Score of posts, (b) number of answers, (c) favorite counts, and (d) view counts per question and (e) time delay between question and accepted answer with consistent (CON) vs. inconsistent (ICON) rollback edits

System-based Inconsistencies accounted for around 32.6% of all inconsistencies in our manual analysis (Fig. 6). The system-based inconsistencies are discussed as follows.

(1) ***Temporal Inconsistency.*** Temporal inconsistency arises when already accepted multiple edits get rejected by a single rollback. Hence, it makes previously accepted multiple edits useless. According to our investigation, such inconsistent rollback edits not only ignore the quality of content but also could revert the post to an error-prone revision. Fig. 1 shows that a user rollbacks a revision from 10 to 1 and thus rejects revisions 2 through 9. Unfortunately, the answer was reverted to a faulty version. Temporal inconsistency contributes to 44.8% of all system-based rollback inconsistencies in our dataset.

(2) ***Structural Inconsistency.*** Structural inconsistency reverts a post to the immediate previous revision. Consider the example (StackOverflow, 2008b) shown in Fig. 12, where a rollback reverted a post from revision 4 to revision 3. Revision 4 can be a revised (i.e., edited) version of 3 since there are no revisions to revert in between 3 and 4. According to our manual investigation, about 55.2% of system-based inconsistent rollback edits have structural inconsistency (Fig. 6).

3.2 Is there a correlation between post quality and inconsistent rollback edits, and do inexperienced users tend to post questions with inconsistent rollback edits? (RQ2)

In this section, we investigate the effects of the rollback edit inconsistencies on questions and whether these inconsistencies are associated with questions posted by inexperienced users.

3.2.1 Approach

We separate the posts in our dataset from RQ1 based on whether their revision history has inconsistent rollback. We then analyze the correlation between posts (with inconsistent and consistent rollback edits) and their popularity/quality metrics. In Software Engineering literature, three metrics–average view count, favorite count, and score–are commonly used to measure popularity/quality, while two metrics—the number of answers and time to get an accepted answer–are used to estimate the difficulty of a question (Bagherzadeh and Khatchadourian, 2019; Ahmed and Bagherzadeh, 2018).

We then divide the users who posted questions into four categories based on their reputation score (Calefato et al, 2018; Mondal et al, 2022a). They are – *New* user (score < 10), *Low-Reputed* user ($10 \leq$ score $< 1K$), *Established* user ($1K \leq$ score $< 20K$) and *Trusted* user (score $\geq 20K$). The official Stack Overflow data dump only accumulates the latest reputation scores of the users, which might not be appropriate for our analysis (Mondal et al, 2021a). We thus use the snapshot of users' activities to calculate their reputation while posting questions (Exchange, Accessed on: December 2019,A; Abric et al, 2019). We then examine whether inconsistent rollback edits are associated with questions from inexperienced users.

3.2.2 Results

The findings from the correlation analysis are discussed as follows.

Posts with inconsistent rollback edits are associated with lower popularity scores. The users subjectively evaluate the quality of a post in Stack Overflow through a voting mechanism. Typically, high-quality posts receive more upvotes, while vague, unclear, or inconsistent posts tend to receive downvotes. The net votes (upvotes – downvotes) cast against a given post in Stack Overflow form an evaluation metric called score, which approximates the posts' quality. Fig. 13 (a) shows the box plots of scores for posts with inconsistent versus consistent rollback edits. We normalize the score between 0 and 1. Posts with inconsistent rollback edits tend to have significantly lower scores than those with consistent ones. We use the Mann-Whitney-Wilcoxon test, a non-parametric statistical significance test (Mann and Whitney, 1947), and get a statistically significant p-value (i.e., $p - value = 0.0 < 0.05$). We also examine the effect size using Cliff's delta test (Macbeth et al, 2011), showing a medium effect size, i.e., Cliff's $|d| = 0.37$ (medium) with 95% confidence.

Similar results are observed for the other two popularity metrics, favorite and view counts. Our analysis indicates that posts with inconsistent rollback edits generally have lower favorite counts (Fig. 13(c)) and view counts (Fig. 13(d)) compared to those with consistent rollback edits. These differences are statistically significant (p-value ¡ 0.05) with small effect sizes.

Questions with inconsistent rollback edits tend to have fewer answers. Fig. 13(b) displays the box plots comparing answer counts for questions with consistent versus inconsistent rollback edits. The results suggest that questions with inconsistent rollback edits tend to receive fewer answers on average than those with consistent rollback edits. A Mann-Whitney-Wilcoxon test indicates a significant difference in the number of answers between the two types of questions, with a $p - value < 0.05$. Cliff's $|d| = 0.18$ suggests a small effect size with 95% confidence.

Questions with inconsistent rollback edits tend to take longer to receive an accepted answer. We investigate whether there is a negative correlation between the delay in receiving an accepted answer and the presence of inconsistencies.

To do this, we measure the delay (in minutes) between a question's submission time and the receipt of an accepted answer. Fig. 13(e) shows box plots comparing the delay for questions with consistent versus inconsistent rollback edits. The mean and median delay for questions with inconsistent rollback edits are higher than those with consistent edits. This suggests a potential negative impact of inconsistency on the timeliness of receiving accepted answers. However, this difference is not statistically significant.

Inconsistent rollbacks occur in questions posted by all user types. We analyze to determine if inexperienced users were more likely to post questions that have inconsistent rollbacks. Our analysis shows that questions with inconsistent rollbacks were posted by 13.3% of new users, 68.7% of low-reputation users, and 18% of established users. In contrast, questions with no inconsistent rollbacks were posted by 11.8% of new users, 73.8% of low-reputation users, and 14.4% of established users. These findings suggest that inconsistent rollbacks occur across all user types, indicating no significant difference in experience levels.

3.3 What is the perceived impact of the observed rollback edit inconsistency types? (RQ3)

In Section 3.2, we presented empirical evidence suggesting that inconsistent rollback edits can negatively affect the quality of posts. In this section, we surveyed developers to understand their perceptions of the impact of the eight identified rollback edit inconsistency types.

3.3.1 Approach

We surveyed 44 developers to determine the impact of the eight inconsistency types. We recruited participants with software development experience ranging from 0 to over 15 years and met our constraint (e.g., they must have editing experience of Stack Overflow posts). However, we did not collect data on the number of posts they edited, their rollback edits, or the acceptance/rejection of their edits. This data is not readily available in user profiles, making it difficult for participants to provide accurate information. Additionally, we aimed to gather insights from both experienced and new editors. We recruited 24 participants using a snowball approach, where initial contacts recommended other eligible participants. The remaining 20 participants were selected from 33 interested developers who responded to advertisements on Facebook and LinkedIn and met the specified constraint. The survey asked participants about the impact of inconsistencies on user engagement and the quality of Stack Overflow posts.

User engagement. We presented one example for each of the eight inconsistency types to the participants. Each example was accompanied by a description to ensure participants' understanding of the underlying context. Then, for each inconsistency type, we asked the question to the participants as follows.

> How does inconsistency in rollback editing impact their participation and contribution to sharing knowledge in Stack Overflow? (*Demotivate and frustrate you for suggesting edits/Confuse and discourage you for suggesting edits/It does not bother me*)

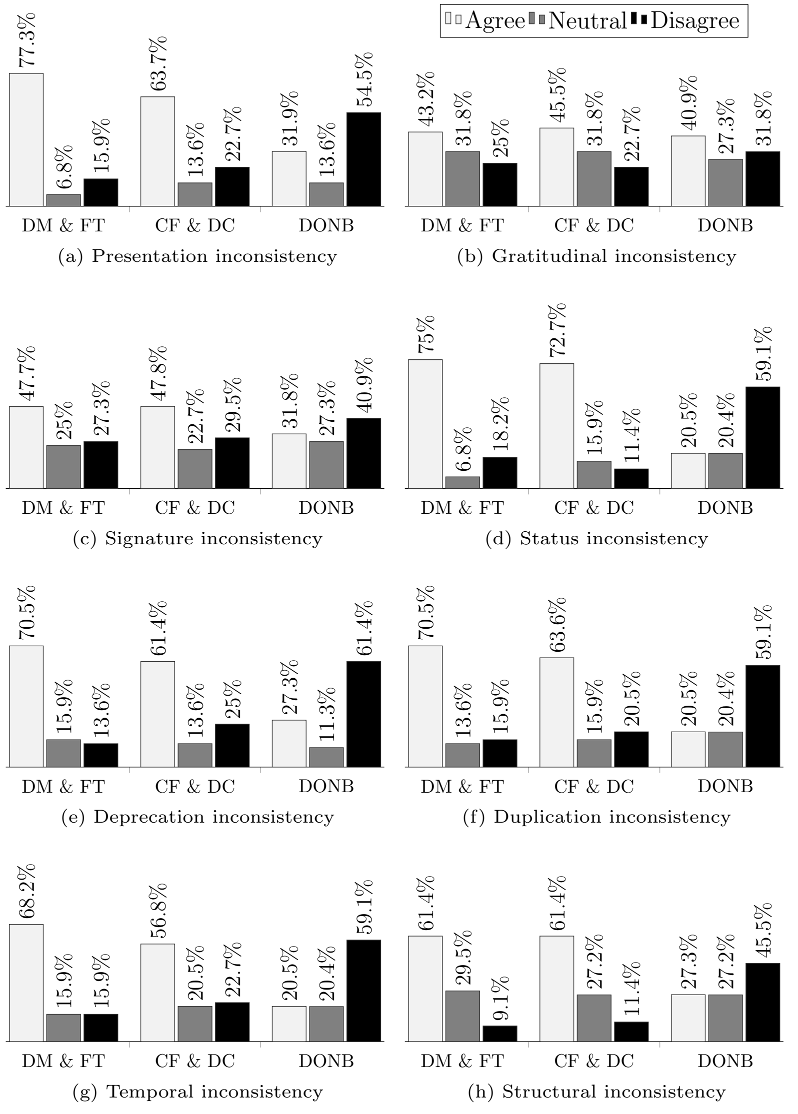


Fig. 14: Impact of the inconsistency types in rollback edits in participation and contribution of users to share knowledge (**DM**: Demotivate, **FT**: Frustrate, **CF**: Confuse, **DC**: Discourage, **DONB**: Does Not Bother).

The participants were asked to provide their agreement/disagreement under each option on a Likert scale: Agree (Strongly Agree, Agree), Disagree (Strongly Disagree, Disagree), and Neutral. We first analyze the findings for each of the inconsistency types. Then, we analyze the results according to the participants' profession (e.g., software developer) and professional experience.

Quality assessment. In our manual analysis, we noticed that inconsistent rollback edits often reject changes that could improve post quality. To validate this observation, we surveyed participants to seek their opinions. Specifically, we asked whether they agreed that inconsistent rollbacks negatively affect the quality of posts with the following question:

> Do you think inconsistencies in rollback edits hurt the quality of the shared content? (*Yes/No*)

3.3.2 Results

The findings on the user engagement of editing posts and their quality are discussed below.

Impacts on user engagement. Fig. 14 (14a – 14f) shows the percentage of agreement and disagreement with each of the three given options for user-based inconsistency types. We see that presentation, status, deprecation, and duplication inconsistencies tend to frustrate and confuse most users, ultimately discouraging them from suggesting edits on Stack Overflow. More than 73% (on average) of the participants agree that such inconsistencies demotivate and frustrate them for suggesting edits in Stack Overflow. Besides, more than 65% (on average) of participants admit that the above inconsistencies confuse and discourage them from suggesting edits. Rollback inconsistencies also highly bother them when contributing to editing to improve posts' quality. Gratitudinal and signature inconsistencies are less likely to be demotivating than the above-mentioned four inconsistencies. However, more than 46% of participants (on average) agree that these inconsistencies demotivate, confuse, and discourage them from suggesting edits. On the contrary, only 26% of the participants (on average) disagree that inconsistencies demotivate them from suggesting edits. Overall, 41% of participants agree that user-based inconsistencies demotivate, frustrate, confuse, and discourage them, which is a substantial portion.

Similar to user-based inconsistencies, about 65% of participants agree that system-based inconsistencies (temporal and structural) demotivate and frustrate them (Fig. 14g and Fig. 14h). In contrast, only about 13% of participants do not agree. About 60% of participants agree that such inconsistencies also confuse and discourage them from suggesting edits and only 17% of participants disagree. Overall, 47% more participants agree that system-based inconsistencies demotivate, frustrate, confuse, and discourage them.

We then analyze the agreement/disagreement according to the survey participants' professions (e.g., software developer, technical lead, academic practitioners). According to our analysis, 62% – 69% participants agree that inconsistent rollback edits demotivate and frustrate, and 55% – 81% participants agree that such inconsistencies confuse and discourage them from suggesting edits. Only 13%–26% of participants disagree that such inconsistencies demotivate, frustrate, confuse, and discourage them from suggesting edits.

We get similar results when analyzing the results based on professional experience. We find that the highly experienced participants (e.g., 15+ years) were frustrated or confused lower (44% – 50%) than less experienced participants. Participants with high experience tend to take a lot more stress and handle diverse jobs daily in their profession. That's why they are not so concerned about such

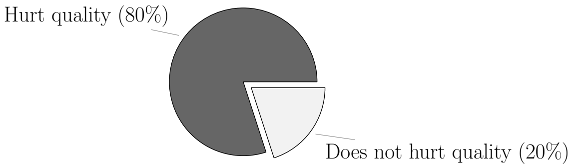


Fig. 15: Impacts of the rollback inconsistency on post quality

inconsistencies. However, more than 56% of them agree that such inconsistent rollback edits bother them.

Impacts on the quality of the posts. According to our analysis, 80% of survey participants agree that inconsistencies in rollback edits hurt the quality of the Stack Overflow posts (Fig. 15).

4 P2: Automatic Detection and Analysis of the Rollback Edit Inconsistencies

In this section, we answer the following research questions:

RQ4. To what extent can rule-based and machine learning techniques detect rollback edit inconsistencies, and which approach is more effective?
RQ5. What is the prevalence and distribution of different types of inconsistencies across all Stack Overflow rollback edits?
RQ6. What types of users perform inconsistent rollback edits on Stack Overflow?

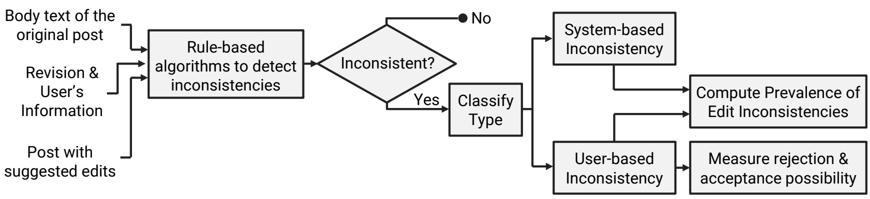


Fig. 16: Detection of rollback edit inconsistencies, their prevalence, and rejection possibility.

4.1 To what extent can rule-based and ML techniques detect rollback edit inconsistencies, and which approach is more effective? (RQ4)

In this section, we discussed the detection of inconsistent edits using both rule-based and machine learning techniques. We then analyzed their effectiveness and performance.

4.1.1 Rule-Based Algorithms

Fig. 16 shows an overview of our inconsistency detection technique using rule-based algorithms. First, we develop eight algorithms to detect the eight rollback inconsistencies (presented in Section 3). Then, we create a corpus to analyze their performance.

4.1.1 (a) Algorithms

We present the algorithms below.

(1) ***Presentation Inconsistency.*** Stack Overflow users are suggested to put inline code terms (i.e., code terms within the textual description) inside the < *code* > .. < */code* > tags and code segments inside the < *code* > under < *pre* > tag. Here, we only consider the presentation inconsistency of inline code terms and thus remove the content under the < *pre* > tag from the body of a post. Now, consider the following terms. TXT_{br}: texts before rollback, TXT_{ar}: texts after rollback, LCT_{br}: list of inline code terms before rollback, and LCT_{ar}: list of inline code terms after rollback. Then, we identify the presentation inconsistency in the following ways.

> IF both LCT_{br} and LCT_{ar} are null OR $LCT_{br} == LCT_{ar}$, THEN there is no presentation inconsistency.

> IF any of the code terms of LCT_{br} are not in LCT_{ar} but in TXT_{ar} (i.e., there is at least one element that was formatted as code terms in the text before rollback but formatted as non-code terms in the text after rollback), THEN there is presentation inconsistency.

> IF any of the code terms of LCT_{ar} are not in LCT_{br} but in TXT_{br}, THEN there is presentation inconsistency.

(2) ***Gratitudinal Inconsistency.*** We find several keywords during the manual investigation that are related to gratitude, such as – *welcome, thanks, sorry, appreciated, thank, ty* (i.e., thank you), *thx, regards,* and *tia* (i.e., thanks in advance). We thus look for these keywords using regular expressions. Then, we attempt to identify the gratitudinal inconsistency in the following ways.

> IF we do not find a keyword match to either TXT_{br} or TXT_{ar} OR we find a keyword match to both TXT_{br} and TXT_{ar}, THEN there is no gratitudinal inconsistency.

> IF we find a keyword match to TXT_{br} but do not find a match to TXT_{br} OR vice versa (i.e., rollback edit either rejects or accepts gratitude), THEN there is gratitudinal inconsistency.

(3) ***Signature Inconsistency.*** We first extract user information, particularly the names of two users: (i) the one who performed the rollback and (ii) the one whose edit was rolled back. We then look for these names using regular expressions.

> IF we do not find a match to any of the two names (full or part) to either TXT_{br} or TXT_{ar} OR we find a match to both TXT_{br} and TXT_{ar}, THEN there is no signature inconsistency.

> IF we find a match to any of the two names (full or part) to TXT_{br} but do not find a match to TXT_{br} OR vice versa (i.e., rollback edit either reject or accept signature), THEN there is signature inconsistency.

(4) ***Status Inconsistency.*** During the manual investigation, we find that statuses are often updated, followed by several keywords, such as – *edit, update, note,* and *ps*. We thus look for these keywords using regular expressions. We apply the same detection technique, as discussed in gratitudinal inconsistency, to decide whether there is status inconsistency.

(5) ***Deprecation Inconsistency.*** To mark an answer that discusses deprecated technology (e.g., API), users add a message followed by keywords *deprecation, deprecate*. We thus look for these keywords using regular expressions. We apply the same detection technique, as discussed in gratitudinal inconsistency, to decide whether there is deprecation inconsistency.

(6) ***Duplication Inconsistency.*** To mark a duplicate question, users add a duplicate message followed by keywords, such as – *duplicate, duplication, related to*. We thus look for these keywords using regular expressions. We apply the same detection technique, as discussed in gratitudinal inconsistency, to decide whether there is duplication inconsistency.

(7) ***Temporal Inconsistency.*** To detect temporal inconsistency, we find the current revision number (say, R_c) and the revision number where the post was reverted by a rollback (say, R_p). We then compute their difference, $d = R_c - R_p$.

> IF $d > 2$, THEN there is temporal inconsistency.

(8) ***Structural Inconsistency.*** To detect structural inconsistency, we find the current revision number (say, R_c) and the revision number where the post was reverted by a rollback (say, R_p). We then compute their difference, $d = R_c - R_p$.

> IF $d == 1$, THEN there is structural inconsistency.

Table 1: Performance of rule-based techniques

Classifier	Inconsistent Rollback			Consistent Rollback			Overall
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Accuracy
Rule-based	1	1	1	1	0.99	0.99	0.99

Table 2: Performance of rule-based algorithms to detect each inconsistency type

Inconsistency	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Sample Count**
Presentation	1	0.98	0.99	0.99	40
Structural	1	1	1	1	29
Temporal	1	1	1	1	42
Gratitudinal	1	1	1	1	31
Status	1	1	1	1	43
Duplication	1	0.94	0.97	0.99	16
Signature and Deprecation	1	0.50	0.67	0.99	5

4.1.1 (b) Performance of the Algorithms

We create an evaluation corpus as follows. We collected all Stack Overflow rollback edits from the September 2019 data dump, the latest dump available during our analysis. This dump contains a total of 102K rollback edits. We used the data dump instead of the Stack Overflow API to ensure the reproducibility of our analysis. We ran each of our eight algorithms on the 102K rollback edits to identify potential inconsistencies. To validate our findings, we randomly selected 400 rollback edits from the dataset – 200 edits where our algorithm detected no inconsistencies and 200 where it found one or more. This sample size is statistically significant, with a 95% confidence level.

To analyze the accuracy of the algorithms, we manually label the sampled 400 rollback edits in a file as follows. (a) **Got**: the inconsistency detected by the algorithm. (b) **Expected**: the actual inconsistency based on our manual analysis. We then create a confusion matrix to analyze the performance of the algorithm as follows. (a) True Positive (TP) = 'got' inconsistency = 'expected' inconsistency (b) False Positive (FP) = ('got' inconsistency $\neq$ 'expected' inconsistency) or ('got' inconsistency but 'expected' no inconsistency) (c) True Negative (TN) = 'got' no inconsistency and 'expected' no inconsistency, and (d) False Negative (FN) = 'got' no inconsistency but 'expected' one or more inconsistency. Using the above matrix, we compute four standard metrics (Precision P, Recall R, F1-score $F1$, and Accuracy A) (Manning et al, 2009) to compute the performance of each algorithm.

Table 1 shows the overall performance of our algorithms in detecting inconsistent and consistent rollback edits. In contrast, Table 2 details their effectiveness in identifying each type of inconsistency. Overall, our algorithms demonstrate high precision and recall, with precision achieving a perfect score of 1 for each algorithm. However, recall is somewhat lower for signature and deprecation inconsistencies. Sometimes, users add names (e.g., nicknames) as signatures different from their Stack Overflow account names. Thus, our algorithm failed to identify these signatures by analyzing the texts of the posts' bodies.

4.1.2 Machine Learning-Based Techniques

The previous section demonstrated that our rule-based algorithms can nearly perfectly identify rollback inconsistency types, as they excel in executing discrete logic. However, ML techniques focus on processing multiple inputs to predict outcomes. To ensure the robustness and scalability of our selected keywords and text patterns, we developed several ML classifiers to assess their effectiveness in classifying both consistent and inconsistent rollback edits and each type of inconsistency. This comparison helps us understand the strengths of both techniques and determine the most effective solution for the problem.

4.1.2 (a) Machine Learning Models

We choose the following four popular machine learning classification techniques with different learning strategies to identify inconsistent rollback edits.

(1) ***Decision Trees (DT).*** Decision Trees is a non-parametric supervised ML technique. Since it is non-parametric, it does not make any assumptions about the underlying data distribution. The intuition behind DT is that simple decision rules are inferred from the dataset features, and the training set is continuously split until all instances of each class are isolated. In particular, this technique employs different heuristics (e.g., information gain, Gini index) to decide which feature to use for the subsequent split of the training set. The frequently used DT are `ID3`, `C4.5`, and `CART`. However, we use `CART` since it performs well on both continuous and categorical features.

(2) ***Random Forest (RF).*** Random Forest is a supervised ensemble learning technique for classification and regression. It consists of many decision trees (i.e., 'forest') and is usually trained with the 'bagging' method. The advantage of such an ensemble technique is that it utilizes a group of weak learners to form a strong learner. In this way, ensemble learners improve the overall performance of single classifiers by combining several classifiers to obtain a new classifier that outperforms every one of them (Polikar, 2006). Moreover, RF is scalable to any number of dimensions with satisfactory performance most of the time. It adds additional randomness to the model while growing the trees. For example, instead of searching for the most important feature while splitting nodes, it searches for the best feature among a random subset of features. Thus, it prevents the overfitting of datasets.

(3) ***eXtreme Gradient Boosting (XGBoost).*** XGBoost is a scalable tree-boosting system that predicts a target class by integrating an ensemble of estimates from a set of more simplistic and weaker models (Chen and Guestrin, 2016). It is a supervised learning algorithm employed for both classification and regression. XGBoost extends gradient-boosted decision trees (GBM) with improved speed and performance. It is relatively faster than other algorithms due to its parallel and distributed computing system. The reasons behind the high performance of XGBoost are its robust handling of various data types, relationships, distributions, and the variety of hyperparameters. In addition, it has built-in cross-validation and a variety of regularizations that help reduce model overfitting.

(4) ***Artificial Neural Network (ANN).*** The working principle of the human brain inspires Artificial Neural Networks that mimic the way biological neurons signal to one another. A multi-layer neural network consists of a large number of units (i.e., neurons) joined together in a pattern of connections. It can classify non-linearly separable sets of instances. ANN depends upon three fundamental aspects - (i) the input and activation functions of the unit, (ii) network architecture, and (iii) the weight of each input connection. The behavior of the ANN is defined by the current values of the weights, given that the first two aspects are fixed. Initially, the weights are set to random values to train the net. Instances of the training set are then repeatedly exposed to the net. The output of the net is compared with

the actual output for instances based on the input values. All the weights are then adjusted slightly to bring the output values of the net closer to the desired output values.

We then select the dataset and model attributes to train and test our models. We choose the manually labeled 764 rollback edits (382 questions + 382 answers) to train our models. Among these rollback edits, 312 have one/multiple inconsistencies, and the remaining 452 do not have such inconsistencies. We use the rule-based algorithms and bag of words model to extract features. We use a binary feature (0/1) in the ML model to represent the presence or absence of inconsistencies, where '1' indicates an inconsistency is present and '0' indicates it is not. These binary features are derived from the inconsistencies detected by our rule-based algorithms. The bag-of-words model is known for its simplicity and effectiveness in text classification and language modeling (Boulis and Ostendorf, 2005; Zhang et al, 2010). We tested the performance of our models on the 400 rollback edits (203 inconsistent + 197 consistent), which was also used to evaluate the performance of the rule-based algorithms.

Table 3: Performance of Machine Learning techniques

Classifier	Inconsistent Rollback			Consistent Rollback			Overall
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Accuracy
DT	0.99	0.85	0.91	0.86	0.99	0.92	0.92
RF	0.96	0.86	0.91	0.87	0.96	0.91	0.91
XGBoost	0.97	0.92	0.95	0.92	0.98	0.95	0.95
ANN	0.90	0.85	0.88	0.86	0.91	0.88	0.88

Table 4: Performance of XGBoost to detect each inconsistency type

Inconsistency	Precision	Recall	F1-Score	Accuracy
Presentation	0.97	0.95	0.96	0.96
Structural	1	1	1	1
Temporal	0.97	0.81	0.88	0.89
Gratitudinal	1	1	1	1
Status	0.97	0.97	0.97	0.98
Duplication	1	0.81	0.90	0.91
Signature and Deprecation	1	0.33	0.50	0.66

4.1.2 (b) Performance Evaluation

Table 3 shows the overall performance of the selected models in classifying inconsistent and consistent edits. According to the experiments, all the models perform well, with over 90% precision for identifying inconsistent edits and overall accuracy ranging from 88% to 95%. Among the models, XGBoost achieves the best performance on our dataset.

Given its superior performance, we utilize XGBoost to classify each inconsistency type. Table 4 shows the performances of the XGBoost classifier in detecting inconsistency types, where precision ranges from 97% to 100%. However, accuracy and recall are lower for signature and deprecation inconsistencies. The potential reasons behind this are the small sample size of these two inconsistency types in our dataset, and users might add names that differ from their profile names.

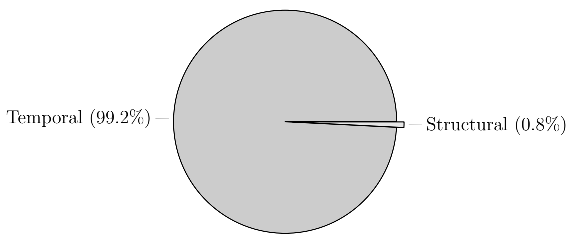
The high performance of machine learning models, which leverage features generated based on rule-based algorithms and bag-of-words, further validates the effectiveness of the identified textual patterns in detecting inconsistent rollback edits. However, the rule-based algorithms are lightweight and fast. Thus, we decided to utilize them for further analysis and integrate them into our online tool.

4.2 What is the prevalence and distribution of different types of inconsistencies across all Stack Overflow rollback edits? (RQ5)

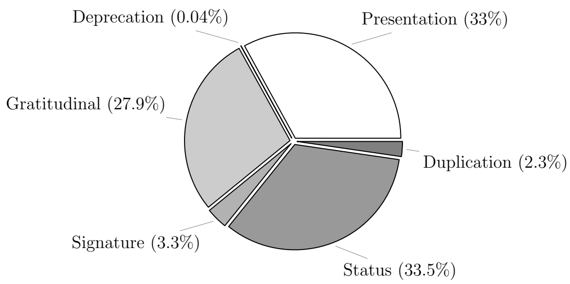
This section attempts to see the prevalence of inconsistent edits on the September 2019 Stack Overflow data dump.

4.2.1 Approach

We collected all the rollback edits from the Stack Overflow September 2019 data dump. Then, we apply our eight inconsistency detection algorithms from Section 4.1 to see the prevalence of inconsistent rollback edits.



(a) System-based rollback edit inconsistencies



(b) User-based rollback edit inconsistencies

Fig. 17: Distribution of rollback edit inconsistencies on the Stack Overflow dump of September 2019.

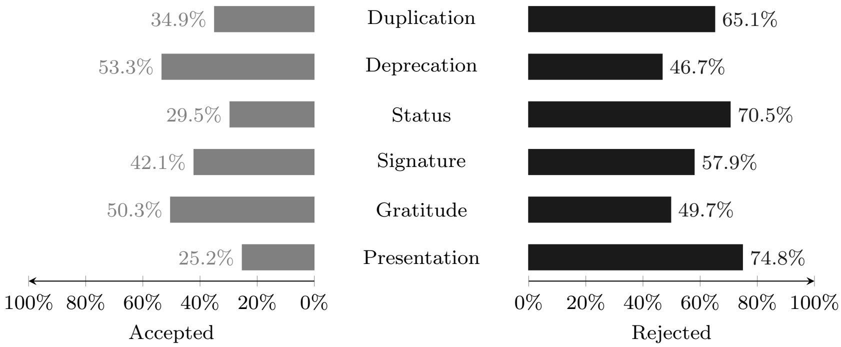


Fig. 18: Accepted vs rejected ratio of rollback inconsistent edits.

4.2.2 Results

According to our experiment, about 40% of rollback edits have one or multiple inconsistencies. Among them, 26.2% of them have system-based inconsistencies, and the remaining 73.8% of them contain user-based inconsistencies. Fig. 17 shows the distribution of rollback inconsistencies (system and user-based) in our dataset. We see that temporal inconsistency is the most frequent system-based inconsistency in the rollback edits. As shown in Fig. 17a, about 99% of the system-based rollback inconsistencies have temporal inconsistency. This high percentage of temporal inconsistency suggests that users frequently perform unnecessary or undesirable edits. Additionally, rejecting multiple accepted edits can also overlook valuable content (e.g., see Fig. 1). In our dataset, 34,602 accepted edits were rejected by 14,479 rollback edits with temporal inconsistency. On average, 2.4 accepted edits were rejected per rollback with temporal inconsistency.

Among user-based inconsistencies (Fig. 17b), status inconsistency is the most prevalent inconsistency. It was seen that 33.5% of the inconsistent rollback edits have status inconsistency. It means that personal messages and acknowledgments are often rejected or brought back by rollback edits. The presentation inconsistency in rollback edits was seen as the second most frequent (i.e., 33%) among all the user-based inconsistencies. New programming languages and updated versions of existing programming languages come with a set of new constructs (i.e., code terms). The unfamiliarity of code terms discussed at Stack Overflow posts and personal preference for formatting code terms could increase presentation inconsistency in rollback edits. Gratitudinal inconsistency is also very frequent (i.e., 27.9%) in Stack Overflow rollback inconsistencies. Stack Overflow authority discourages adding such gratitude with posts. Unfortunately, many users do not follow the rules, which increases rollback edits with gratitudinal inconsistencies.

We also find that some users roll back edits solely to reject or bring back signatures, duplication notes, or deprecation notes without significantly improving post quality. Identifying and discouraging such inconsistent rollbacks is essential to promote genuine contributions. In our dataset, signatures, duplication, and deprecation inconsistencies were observed infrequently.

To provide users with better insights based on revision history, we attempt to see the possibility of rejection for inconsistent edits. In particular, we find the percentage of user-based inconsistent edits rejected or accepted (i.e., brought back) by rollbacks. These findings can help users to decide whether they should include or avoid inconsistencies during editing. Fig. 18 shows the rejected and accepted user-based inconsistencies ratio. Our analysis reveals that rollback edits often reject inconsistent edits. For example, presentation and status inconsistencies are rejected in over 70% of cases. In comparison, duplication inconsistencies are rejected more than 65% of the time, and signature inconsistencies are rejected in about 60% of cases. However, the rejection ratio of gratitudinal and deprecation inconsistencies was similar to the acceptance ratio. Automatic identification of inconsistency types and their rejection/acceptance ratios can assist users in making informed decisions about including or avoiding these inconsistencies in their edits.

Table 5: User types involved in inconsistent rollback edits (**New user** (reputation score < 10); **Low Reputed user** ($10 \leq$ reputation score $< 1K$); **Established user** ($1K \leq$ reputation score $< 20K$); **Trusted user** (reputation score $\geq 20K$))

Rollback by Post-Owner				**Rollback by Non-Post-Owner**	
New	**Low-Reputed**	**Established**	**Trusted**	**Established**	**Trusted**
1,747 (10.7%)	7,397 (45.1%)	5,136 (31.3%)	2,108 (12.9%)	10,891 (52.8%)	9,736 (47.2%)

4.3 What types of users perform inconsistent rollback edits on Stack Overflow? (RQ6)

This section investigates what types of users (e.g., new) conduct inconsistent rollback edits. In particular, we attempt to see whether inconsistent rollback edits are conducted by a specific type of user or all.

4.3.1 Approach

We categorize users into four groups based on their reputation scores – *New*, *Low Reputed*, *Established*, and *Trusted*, following the approach outlined in Section 4.3.1. Next, we examine which user categories are most prone to making inconsistent edits.

4.3.2 Results

Table 5 summarizes the rollback edit inconsistencies based on user types. Low-reputed and established post owners frequently roll back their posts with inconsistencies. For example, about 45% of these rollbacks were committed by low-reputed users. In contrast, new and trusted users infrequently make inconsistent rollback edits. Combining users into two groups—lower reputation (new and low-reputed) and higher reputation (established and trusted)—shows that those with lower reputation scores are responsible for 55.8% of inconsistent rollbacks, while those with higher reputation scores account for 44.2%. We observe similar trends in rollbacks by non-post owners. Established users performed 52.8% of inconsistent rollbacks, and trusted users performed 47.2%. Such findings suggest that inconsistent rollback is not associated with any specific categories of users but all categories. Thus, the inconsistency detection tool could help all the users.

We then examine the relationship between two categorical variables – user type and user count who committed inconsistent rollback edits. To assess the independence of these variables, we apply the *Chi-Squared* statistical test (McHugh, 2013) (see Table 5). The result shows statistically insignificant p-values ($p - values = 0.06, 0.50 > 0.05$) for rollbacks by both post-owners and non-post-owners, suggesting no significant dependency between user type and the occurrence of inconsistent rollback edits.

5 P3: Automatic Assistance to Mitigate the Edit Inconsistencies

Our findings from Section 3 showed that inconsistent rollback edits negatively impact the quality of posts and user engagement. Therefore, these inconsistent edits should be identified and discouraged to – (1) promote the quality of posts and (2) ensure a better user experience. We thus developed algorithms to identify and mitigate those inconsistencies. However, we can assess the actual impact of our proposed algorithms if they can automatically identify inconsistent edits during the editing of a Stack Overflow post.

Furthermore, editing is a time-consuming and largely voluntary activity in Stack Overflow. However, users could benefit from a tool that identifies inconsistent edit types and predicts their likelihood of rejection. We thus focus on introducing an online tool called `iEdit`, which interacts with our algorithms, identifies the inconsistent edit types from the suggested edits, and then assists users with the likelihood of rejection of these inconsistent edits.

To ensure the tool meets user needs, we first surveyed developers to assess the demand for such a tool and to determine its design requirements. We then designed the tool's interface and architecture based on these insights. Finally, we evaluated the tool's effectiveness, focusing on the following two research questions in this section:

RQ7. What design requirements should an interactive tool meet to assist developers in mitigating inconsistencies in their edits?
RQ8. Can the tool effectively assist developers during their edits of Stack Overflow posts?

5.1 What design requirements should an interactive tool meet to assist developers in mitigating inconsistencies in their edits? (RQ7)

The existing editing system of Stack Overflow does not interact with the users, even if the suggested edits have inconsistencies. Thus, the support of interactive and intelligent tools can help address these inconsistencies and enhance the overall quality of the posts.

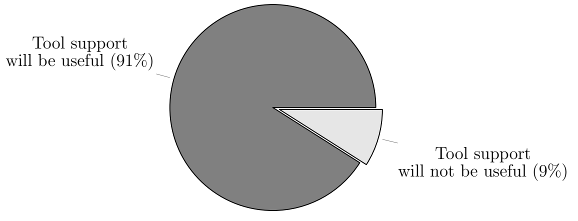


Fig. 19: Practitioners' opinion on the usefulness of tool support.

5.1.1 Approach

To seek developers' opinions on the need for tool support, we asked participants the following question:

> Would tool support be useful for users to avoid inconsistent rollback edits? (*Yes/No*)

We then seek participants' (Section 3.3) opinions on tool support needs. Besides, we offer a few tool support options and employ a 5-point Likert scale to see the participants' consent with the tool options. In particular, we ask the questions as follows.

Q$_1$) What other types of tool support would be helpful? (*Table 6*)
Q$_2$) To what extent do you agree with the following options? (*Table 7*)

The 44 developers surveyed are the same individuals who participated in our study on the impact of rollback edits on Stack Overflow, as detailed in Section 3.3.1. We designed, implemented, and introduced the tool based on their feedback.

Table 6: Participants' recommendation to question: *What other types of tool support would be helpful?*

A tool that forces users to follow certain unified standards while making edits.
Some learning-based tools can learn the different patterns of inconsistency as well as classify them and finally approve only the useful edits.
In stack overflow need a plugin that guides the user to avoid inconsistency.
An API that can be used by Stack Overflow, so when you suggest an edit, you can see the possible inconsistency.

5.1.2 Results

Fig. 19 shows the participants' opinions on the introduction and usefulness of tool support. The survey participants strongly suggest introducing tool support to identify and recommend users with inconsistent edits. Specifically, 91% of the participants agreed that such a tool would be beneficial for interacting with users and helping them avoid inconsistent rollback edits.

Table 7: Assessment of the tool support options

Options	Mean Value (Interpretation)
A tool (e.g., browser plugin) that analyzes rollback edits and suggests users avoid inconsistent rollback edits.	4.3 (Very Influential)
An IDE (e.g., Eclipse) plugin that warns users to avoid inconsistent rollback edits.	3.9 (Influential)
A website that guides users to avoid inconsistent rollback edits.	3.7 (Influential)

We asked participants to provide recommendations on potential tool support options. Table 6 includes some of their practical suggestions, such as a Stack Overflow plugin designed to help users avoid inconsistencies. Table 7 presents our proposed tool options and their evaluations. For instance, one option was "a browser plugin that analyzes rollback edits and suggests users avoid inconsistent rollback edits.". The participants assessed this tool option as very influential (e.g., Likert score $\geqslant 4.21$) (Sözen and Güven, 2019). However, the remaining two options were also estimated as influential ($3.41 \leqslant$ Likert score $\leqslant 4.20$).

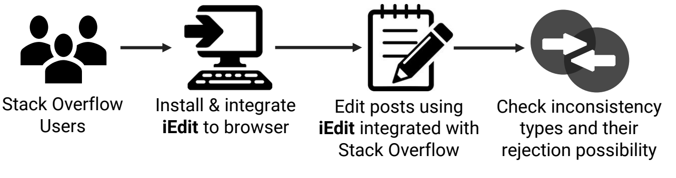


Fig. 20: An overview of the `iEdit` and its usage

Fig. 20 shows the overview of how `iEdit` is introduced. Users of Stack Overflow can easily install and integrate `iEdit` to the Stack Overflow editing system. After installation, users get the `iEdit` interface integrated with the Stack Overflow edit system. Then, they can edit the posts using `iEdit` to improve their quality. Before submitting the edits, users can check whether the edits contain any inconsistencies with their rejection possibility.

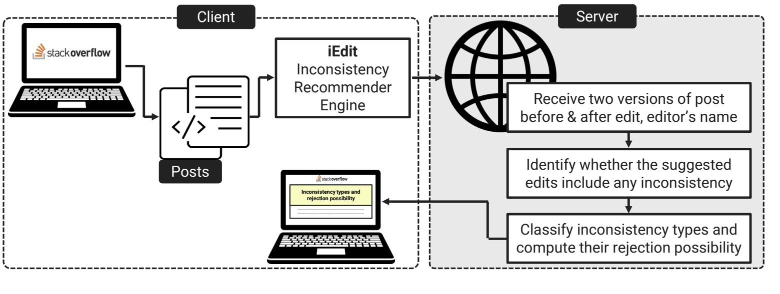


Fig. 21: An overview of the `iEdit` system architecture

Fig. 21 shows an overview of the `iEdit` architecture. `iEdit` has two parts: *client* and *server*. On the client side, users get the `iEdit` interface that comprises two buttons: *iEdit* and *CheckIn*. *iEdit* enables users to edit the Stack Overflow posts. On the other hand, users can check whether the edits have any inconsistency types and their possibility of rejection by clicking the *CheckIn* button. When users click the CheckIn button, the client-side script captures the necessary data to identify the inconsistency types by extracting the text patterns. In particular, it captures – (1) text before edit, (2) text after edit, and (3) the name of the user. Then, it sends this data to the server-side application. Client-side script is written in JavaScript, and server-side application is developed in Java.

However, the server-side application scans the texts and looks for the patterns (e.g., Section 4.1) using regular expressions. Then, it determines the inconsistency types if the texts are matched with one or more target patterns. Finally, the client-side script alerts users about whether the suggested edits contain any inconsistency types and informs the estimated possibility of rejections of those inconsistency types.

5.2 Can the tool effectively assist developers during their edits of Stack Overflow posts? (RQ8)

In this section, we assess the effectiveness of the tool via the real-world usage of the tool by users and attempt to see to what extent can `iEdit` be helpful to users avoid inconsistencies in their edits.

5.2.1 Approach

To analyze the effectiveness of `iEdit`, we select the study participants and design the study as follows.

Table 8
Experience and profession of the participants

Development Experience (Years)			**Profession**			**Editing Experience**	
$\leq$ 2	**3-5**	**9-11**	**Academician**	**SW Developer**	**Research Engineer**	**Yes**	**No**
7 (43.2%)	8 (50%)	1 (6.3%)	11 (68.7%)	2 (12.5%)	3 (18.8%)	12 (75%)	4 (25%)

Study participants. We invited developers who participated in the tool requirement analysis to participate in the survey and encouraged them to share it with their colleagues. We kept most of the previous constraints for recruiting participants but no longer imposed the constraint for prior editing experience on Stack Overflow. Instead, we attempt to see whether users without editing experience feel the tool can help them avoid inconsistencies during their edits. In total, we recruited 16 participants.

Table 8 shows the participants' experience and professions, which include both novice and highly experienced software developers. To evaluate our inconsistency detection tool, we included participants with varying experience levels to see if the tool could help both experienced and inexperienced editors. The participants came from various backgrounds, including academia (e.g., graduate students and faculty members), software industries, and research fields. Notably, 75% of participants had prior experience editing Stack Overflow posts, while 25% did not.

Study design. We conduct an online survey to listen to the recruited participants of the editing experience using our tool `iEdit`. We follow the steps suggested by Kitchenham and Pfleeger (Kitchenham and Pfleeger, 2008), such as setting goals, survey design, developing and evaluating the survey instrument (e.g., questionnaire), and collecting and analyzing data. However, we also consider ethical guidance from the established best practices (Groves et al, 2011; Singer and Vinson, 2002). For example, we take participants' consent before starting the survey. We also confirm that the participants' information must be treated confidentially. Our survey includes different types of questions (e.g., multiple-choice and free-text answers). In the beginning, we explain the purpose of the study and our research goals to the participants. We also inform them of the estimated time (i.e., 15-20 minutes) to complete the survey. Our survey comprises the following parts:

(1) ***Consent.*** In this part, we ask the participants to confirm whether they consent to participate in this survey and agree to process their data.
(2) ***Participants Information.*** In this part, we attempt to collect some information about the participants, such as software development experience, their current profession, organization, country, and editing experience in Stack Overflow posts.
(3) ***`iEdit` Installation.*** This part focuses on the installation complexity (e.g., easy/difficult) of `iEdit`. In particular, we attempt to see whether the `iEdit` installation process is time-consuming and tedious. We upload an installation manual and share the required resources (e.g., UserScripts) with participants to install `iEdit`. Then, they install and integrate our tool with the Stack Overflow editing system. However, a complex installation process might discourage users from using this tool. We thus ask the following question to know participants' experience on `iEdit` installation:
    (a) How complex is the process of installing iEdit in your browser? (Options were: *very easy/easy/moderately difficult/ difficult/very difficult/I cannot install*)
(4) ***Effectiveness Analysis.*** In this section, we attempt to measure the – (1) usefulness of the `iEdit` suggestions and (2) confidence of the participants to follow them. We asked participants to edit several (at least five) Stack Overflow posts. After they made their edits, our tool assisted them by detecting inconsistencies, identifying their types, and providing the associated rejection and acceptance ratios. These insights increased users' awareness of potential inconsistencies, helping to reduce them and, in turn, decreasing the likelihood of their edits being rejected. In particular, we ask the following two questions and employ a 5-point Likert scale (i.e., 1–5) to estimate the participants' consent (Joshi et al, 2015; Vagias, 2006).
    (a) How useful did you find the suggestions from `iEdit`? (*5-point Likert scale*)
    (b) How confident were you to follow the `iEdit` suggestions? (*5-point Likert*)
(5) ***Workload Assessment*** It is essential to assess whether the cognitive workload (e.g., mental demand) of using `iEdit` is low. The NASA Task Load Index (TLX) is one of the most popular techniques for estimating subjective workload (Cao

et al, 2009; Hart and Staveland, 1988). Thus, we leverage NASA TLX (non-weighted) to assess how much effort participants had to exert mentally and physically to use `iEdit` and the standard edit system of Stack Overflow. In particular, participants were asked to rate their scores on an interval scale ranging from low (1) to high (10) in the following six dimensions:

(i) *Mental demand* estimates how much thinking, deciding, or calculating was required to perform the task.
(ii) *Physical demand* measures the amount and intensity of physical activity required to complete the task.
(iii) *Temporal demand* assesses the amount of time pressure involved in completing the task.
(iv) *Effort estimates* how hard the participants have to work to maintain their level of performance.
(v) *Performance estimates* the level of success and satisfaction in completing the task.
(vi) *Frustration level* perceives how insecure, discouraged, secure, or content the participant felt during the task.

(6) ***Suggestions to Improve `iEdit`.*** This section seeks recommendations on how to improve the effectiveness, usefulness, and user experience of `iEdit`. We ask them the question as follows.
   (a) What are your recommendations to further improve `iEdit`? (*Text*)

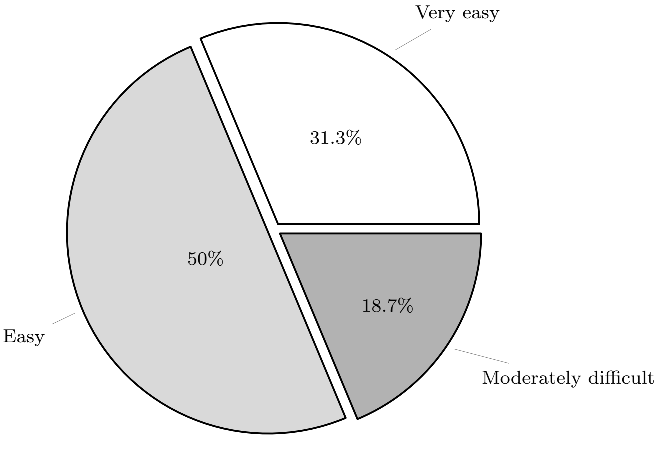


Fig. 22: `iEdit` installation complexity.

5.2.2 Results

Our findings are discussed below.

Ease of iEdit Installation. Fig. 22 summarizes the responses from the participants. We see that 31.3% (5 out of 16) of the participants were able to install our tool *very easily.* Half of the participants found it *easy* to install. The installation process was *moderately difficult* for only three (18.7%) participants. However, none of the participants neither found the installation of `iEdit` difficult nor failed to install it. Such responses ensure that the installation of `iEdit` is not complex at all, and thus, users can easily install and use it to edit Stack Overflow posts to avoid inconsistencies.

Table 9
Effectiveness analysis of `iEdit` and Stack Overflow edit system

Questions	Mean Value	Interpretation
How useful did you find the suggestions from `iEdit`?	4.4	Very Influential
How confident were you to follow the `iEdit` suggestions?	4.3	Very Confident

Usefulness of `iEdit` Suggestions. This section attempts to assess the effectiveness of `iEdit` and the Stack Overflow edit system. Table 9 shows the participants' assessment. We see that participants find the suggestions of `iEdit` *very influential* ($4.21 \leqslant$ score $\leqslant 5.0$) in avoiding the inconsistent edits (Sözen and Güven, 2019). The Likert score (i.e., 4.3) also shows that they were *highly confident* to follow the suggestions given by `iEdit`. When we asked the reason behind their confidence level, one participant responded that *`iEdit` communicates with the rate of approval and rejection, then it becomes easy to take the decision from a statistical point of view.* Such findings indicate that `iEdit` is not only able to provide valuable suggestions but also make users more confident in suggesting edits by avoiding inconsistencies.

Workload Assessment During Edit Task Completion. Fig. 23 shows the box plots of the NASA TLX cognitive workload scores on a scale of *one* (lowest) – *ten* (highest). We see that the median of each index is below five except for performance. Such findings suggest that usage of `iEdit` requires minor mental and physical demands and less effort. However, the median performance value is about eight, which confirms high performance. We then compute the average workload of each participant by summing up their rating of each of the six dimensions (e.g., mental demand) and then dividing it by the number of dimensions. In particular, we use the equation to compute the average workload of each participant as follows.

$$A_{wl} = \frac{1}{D_T}\left[\sum_{i=1}^{D_T} R_i\right] \tag{1}$$

where R_i denotes the rating of i^{th} dimension, D_T represents the total dimensions (here, $D_T = 6$).
Finally, we calculate the average (overall) workload by summing up each participant's average diving by the number of participants. According to the result, the average workload is only 3.6, which guarantees that our tool requires a minimum workload.

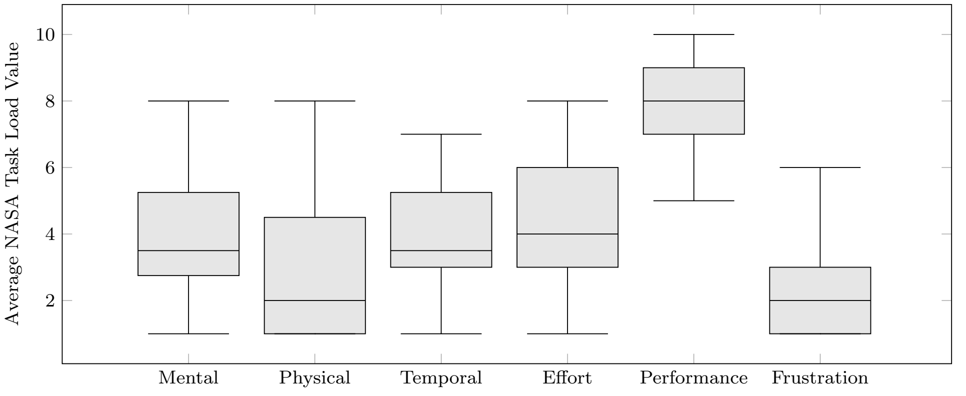


Fig. 23: Cognitive workload using NASA TLX.

Recommendations for `iEdit` Improvements. We analyzed the recommendations of all the participants and summarized them into three categories. We see that participants recommended – (1) improving the Graphical User Interface (GUI), (2) enhancing existing functionalities, and (3) installing the system. We discuss their suggestions below.

- **Improve Graphical User Interface.** Participants suggested improving the graphical user interface by making it more user-friendly and interactive. In particular, they recommended displaying the suggestions from `iEdit` in a suitable place inside the page instead of displaying them as a pop-up window.
- **Enhance Functionality.** Participants recommend that providing suggestions while editing would be a fantastic addition in terms of functionality. They also suggested updating the statistical information (e.g., rejection ratio) frequently so that users can know the current scenario of inconsistent edits. They believe such action could increase reliability. In some cases, `iEdit` shows unexpected behavior, so they suggested conducting more real-time testing.
- **Improve Installation System.** `iEdit` uses Tampermonkey to add userscripts for integrating it into the Stack Overflow edit system. They appreciate it since Tampermonkey is popular, easy to use, and available on all popular browsers. However, they suggest deploying `iEdit` as a standalone browser plug-in in the future.

6 Discussion

6.1 Reasons for Inconsistent Rollback Edits

Inconsistent rollback edits on Stack Overflow can arise from users' intent to preserve original content and the diverse backgrounds and expertise within the community. Variations in editing experience, cultural norms, and evolving community standards also contribute to these inconsistencies. Further details on these factors are provided below.

Preserving Original Content and Self-Reflective Edits. It is crucial to know who performs the edits and inconsistent rollbacks to interpret the data. In our dataset, 62.6% of inconsistent rollbacks were made by the post owner, reflecting either a preference for keeping their original content unchanged or dissatisfaction with the edits. Additionally, the same users made 34.9% of both edits and rollbacks, indicating a process of iterative learning. Among these users, 25.9% were the original post owners. Users might edit and then roll back their changes if they find the edits unnecessary or detrimental to the post's clarity or if feedback from others indicates the edits were not helpful.

Cultural, Contextual, and Communication Challenges. Stack Overflow is a global platform, and users come from diverse cultural and linguistic backgrounds. Cultural, contextual, and communication challenges can contribute to inconsistent edits. For example, users from some countries might prefer to add expressions of gratitude at the beginning or end of their posts, while others might avoid such expressions to keep the content concise and noise-free. These differences in cultural norms, along with varying references, terminology, or linguistic nuances, can lead to variations in how information is presented. Additionally, varying levels of understanding or incomplete knowledge of the topic can result in edits that are not fully aligned with the original content. Miscommunication during collaborative editing or a lack of context can further amplify these inconsistencies, causing changes that do not accurately reflect the intended meaning.

Lack of Awareness and Evolving Standards. Not all users have the same level of editing expertise or experience. Inexperienced or new users might introduce inconsistencies due to unfamiliarity with site guidelines. The editing interface also lacks clear visual cues or warnings about potential inconsistencies, leading to unintentional errors. As the Stack Overflow community evolves, shifts in content quality norms, best practices, and formatting standards can create further inconsistencies. Popular posts that accumulate multiple edits over time may also experience inconsistencies if users do not coordinate their changes according to current norms.

Formatting, Language, and Coding Inconsistencies. Formatting and language preferences can cause inconsistencies in rollback edits. Users often have different ways of formatting, coding styles, and choosing words, which can make the content look uneven. Discrepancies may also arise when users refer to different versions of programming languages, libraries, or frameworks.

Misuse of Editing Powers and Lack of Review. In some cases, users might misuse their editing privileges, leading to inconsistencies or deviations from community guidelines. Additionally, a lack of careful review or oversight can allow edits with varying quality standards to be published, further contributing to inconsistencies.

Interpretation Differences. One of the causes of inconsistent rollback edits is the varying interpretations of questions or answers by different users. Since users may focus on different aspects or angles of the content, their edits can differ in tone, clarity, and emphasis. Subjective topics or opinions, in particular, can lead to varied interpretations, resulting in inconsistent edits that reflect different viewpoints or assumptions about the original intent.

Copy-Pasting Issues. Some users may copy and paste content from other sources to Stack Overflow without verifying its appropriateness for the Stack Overflow context.

6.2 Implications of Study Findings

The findings from our study can guide the following major stakeholders in crowd-sourced platforms: (a) **Forum Designers** to improve the edit system, (b) **Forum Users** to guide their edit behavior, and (c) **Researchers** to study collaborative editing. We discuss the implications below.

Forum Designers. Since content quality is crucial to Stack Overflow's success, improving the current editing system by addressing its identified shortcomings is essential. Specifically, Stack Overflow should focus on reducing prevalent issues like presentation, status, and temporal inconsistencies. As demonstrated, temporal inconsistencies lead to the rejection of thousands of approved edits, which can negatively impact both the quality of shared content and user motivation. These issues may also result in an increased number of suggested or rollback edits, with only a few being meaningful. From a site management perspective, this inefficiency could harm Stack Overflow's performance and overall reputation.

As noted in Sections 1 and 3, inconsistencies in rollback edits can frustrate Stack Overflow users. This frustration, along with a potential decline in content quality, is evident when comparing posts with inconsistent versus consistent rollback edits. Table 10 summarizes four popularity metrics from Fig. 13, showing a significant decrease in views, favorites, scores, and answers for posts affected by inconsistent rollback edits.

Table 10: Average values of popularity metrics across posts with inconsistent vs consistent rollback edits

Metric	Consistent	Inconsistent	Decrease
View	272663.8	196727.3	39%
Favorite	400.0	202.9	97%
Score	693.1	403.3	72%
Answers	18.6	11.1	68%

Therefore, the Stack Overflow authority could consider revising the editing guidelines and communicating these more effectively to users. Additionally, they might regulate rollback actions more effectively. For example, if a user attempts to roll back an edit with temporal inconsistency, Stack Overflow might recommend using a dedicated review channel. Since the platform already utilizes review channels for various decisions, such as closing questions, implementing an additional review channel would be feasible. Furthermore, tools like `iEdit` can assist users in avoiding inconsistencies during their edits. Our user studies on `iEdit` (see Section 5) show that, while simple and intuitive, it is considered highly useful by Stack Overflow users.

SE Researchers. The quality of content on Stack Overflow has been explored in recent studies (Zhang et al, 2018; Ponzanelli et al, 2014a). Such studies traditionally look at the current version of a post to develop tools and techniques, such as assessment of API misuse patterns (Zhang et al, 2018), or producing live API documentation (Subramanian et al, 2014). However, given the availability of editing histories and the fact that high-quality edits can be rejected due to inconsistencies, researchers can use our developed algorithms to pick good-quality

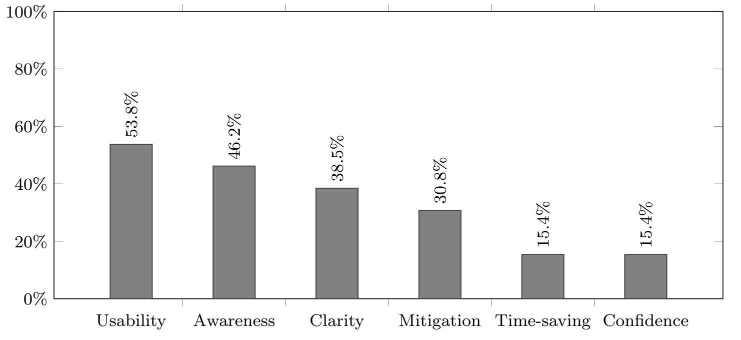


Fig. 24: Aspects of iEdit that Stack Overflow Users Considered Useful During the Survey

suggested edits. Existing ML systems can use additional data aiming to assist developers in the edits of Stack Overflow posts (Chen et al, 2018, 2017).

Our success at designing simple but precise rule-based and ML models to detect inconsistencies shows that it is feasible to develop automated tools that can be incorporated into an interactive virtual assistant like `iEdit`. Our survey of Stack Overflow users finds that the participants considered `iEdit` useful across various aspects. In Fig. 24, we show the useful aspects of the `iEdit` that we observed in the responses of Stack Overflow users. We found the aspects by manually labeling each response of Stack Overflow users based on the underlying theme (i.e., `iEdit` aspect) that the participants mentioned as useful. A response could mention more than one aspect. More than 50% of the responses highlighted the usability of the `iEdit` tool, while around 46% noted that awareness of inconsistencies via `iEdit` could prevent them from introducing inconsistencies in their edits.

Forum Users. Community Q&A forums like Stack Overflow encourage diverse user interactions, making it difficult to enforce strict rules. Our tool, `iEdit`, was developed based on observations from our empirical study that found 29.5% of rollback inconsistencies were related to presentation issues. By highlighting potential inconsistencies during edit suggestions and offering insights on how often these edits are accepted or rejected, `iEdit` helps users understand trends and common issues. Since `iEdit` is built on rules derived from historical analysis of Stack Overflow data, it provides precise suggestions, aiding both new and experienced users in mitigating inconsistencies in their edits.

Additionally, the plugin can track the actions of content owners responsible for approving or rejecting suggested edits. References to previous edits and guidelines can advise the content owner on whether to approve or reject an edit. This plugin can be leveraged to address all six types of user-based inconsistencies in rollback edits. Since approximately 70% of rollback edit inconsistencies are user-based, this system has the potential to significantly decrease these inconsistencies on Stack Overflow. By ensuring more consistent rollbacks, user satisfaction and the overall quality of shared content could be greatly improved.

7 Threats to Validity

External Validity threats relate to the generalizability of our findings. Our study is centered on Stack Overflow, one of the largest and most popular developer forums, to create a catalog of inconsistent rollback edits. We selected statistically representative samples from both question and answer rollback edits, ensuring a 95% confidence level with a 5% margin of error (Boslaugh, 2012). However, these findings may not necessarily apply to other non-technical forums.

Internal Validity threats concern experimenter bias and potential errors in our analysis. Two authors independently labeled the first 200 rollback edits to mitigate bias during our manual labeling process. We then measured agreement using Cohen's Kappa (Cohen, 1968, 1960), achieving a near-perfect agreement (i.e., $\kappa = 0.98$). The distribution of inconsistency types in our manually labeled dataset closely matched that of the entire 102K Stack Overflow rollback edits, where we applied our automated classifier, confirming the reliability of our manual labels. However, since the algorithms were developed using this same dataset, some inadvertent bias might have influenced the analysis of algorithm performance. To address this, we consulted a larger set of unseen data while developing machine learning models, and their performance remained consistent.

Construct Validity threats relate to the difficulty in finding data relevant to identify rollback edits and inconsistencies. Hence, we use revisions of the body of questions and answers from the Stack Exchange data dump, which we think are reasonable and reliable for capturing the reasons and inconsistencies of revisions. However, users can roll back (part of) a change without the rollback button. To investigate this, we manually analyzed 840 revisions (466 questions + 374 answers) from the edit history of 100 samples (50 questions + 50 answers) in our dataset. According to our investigation, 2.8% of revisions for questions involved a partial rollback, while this statistic was 2.4% for answers. However, such percentages might not affect our major findings, and partial rollbacks cannot be identified automatically.

We developed the `iEdit` tool based on feedback from a survey of 44 software developers, including both novice and experienced Stack Overflow users who were experienced with Stack Overflow editing. Their feedback effectively reflects the mood of the Stack Overflow user community. However, future enhancements of `iEdit` could benefit from incorporating input from a broader range of Stack Overflow users. In our tool effectiveness survey, most participants were academics, mainly graduate students who regularly use and edit Stack Overflow posts, providing reliable feedback. The consistency in responses from both academics and other professionals suggests that professional bias did not influence the results.

We followed established literature to design our survey but acknowledged that some questions might have implicit bias. Our survey participants range from novice to experienced and consist mainly of software developers and other related professionals. Such diversity in the survey participants offers validity and applicability to the survey findings. Moreover, the difference between agreement and disagreement on the impact of our inconsistencies is quite large and statistically significant. However, any individual bias in the survey responses should be mitigated via a large sample of 44 users.

8 Related Work

We developed our tool `iEdit` to recommend fixes to suggested edits in Stack Overflow so that Stack Overflow users can avoid committing undesired edits that may lead to the rejection of the edits. As such, our research in this paper belongs to a broader area called 'collaborative editing in social forums'. Related work can broadly be divided into **studies** of collaborative editing systems in crowd-sourced forums and **techniques** to suggest improvements to the editing system.

8.1 Studies of Collaborative Editing Systems

Editing can improve the content quality. As such, it is intuitive that the social Q&A forums offer editing of the post contents. Given social forums can be accessed by many users at the same time, it is a cost-effective measure for the forums to support collaborative editing by allowing their users to do the editing. Indeed, studies show that collaborating editing in social forums and online collaborative knowledge-sharing portals (e.g., Wikipedia) can positively impact the improvement of shared contents (Li et al, 2015; Kittur and Kraut, 2008). The nature of the collaborative editing can be similar across the social forums (e.g., Q&A site) and knowledge portals (e.g., Wikipedia). The research of Li et al. (Li et al, 2015) looked at the adoption of Wikipedia-style collaborative editing into a Q&A site like Stack Overflow. They found that users with good edits are rewarded with positive votes by other users. They analyzed five years of historical editing data from Stack Overflow and found that substantive edits from other users can increase the number of positive votes by 18% for the questions and 119% for answers. This reward can be beneficial for a user who does the edit because the edit may only offer at most 5% improvement over the original post (i.e., the user can be rewarded with mindful but low-cost editing efforts). Indeed, the Stack Overflow reward system can serve as an added influence for users to suggest edits. A recent study by Wang et al. (Wang et al, 2018) in Stack Overflow found that users are motivated to edit more when they are closer to getting a badge.

Collaborative editing systems are common in Wikipedia (Li et al, 2015; Kittur and Kraut, 2008), GitHub code editing (Dabbish et al, 2012), webcasts (Munteanu et al, 2008), scientific contents (Lowry et al, 2005; Calvo et al, 2005), and so on. Studies show that collaborating editing positively impacts the improvement of shared contents (Li et al, 2015; Kittur and Kraut, 2008). A recent study by Wang et al. (Wang et al, 2018) in Stack Overflow found that users are motivated to edit more when they are closer to getting a badge. Indeed, offering incentives such as reputation scores are found to be useful to improve post quality (Li et al, 2015). A similar finding was also observed in webcasts by Munteanu et al. (Munteanu et al, 2008), who tested the effectiveness of engaged users to collaborate in a wiki-like environment to edit/correct transcripts that are produced from webcasts through an automated speech recognition system. Kittur et al. (Kittur and Kraut, 2008) find that the increase in the number of editors does not guarantee the quality of the articles in Wikipedia. Our study findings offer another dimension to the above studies by showing that there are several inconsistencies that are negatively affecting the rollback edit mechanisms and user engagements in Stack Overflow.

The focus of collaborative editing is to improve the quality of the shared content based on user engagement (Agichtein et al, 2008). Chen et al. (Chen et al, 2017) observed that most of the edits in Stack Overflow are small sentence edits. In a follow-up study, Chen et al. (Chen et al, 2018) predicted whether a post needs to be edited. While developing their SOTorrent database, Baltes et al. (Baltes et al, 2018) also observed that the majority of edits in Stack Overflow are relatively small. The quality of the question is important to get an answer: lack of clarity, relatedness, and reproducibility of the problem, as well as the too-short question, could dissuade developers from answering the question (Asaduzzaman et al, 2013; Mondal et al, 2019). The reputation and past activity of an asker could also factor into the likelihood of a question getting resolved (Rahman and Roy, 2015). As such, factors of good questions are investigated, e.g., code-to-text ratio, etc. (Calefato et al, 2018; Duijn et al, 2015). However, depending on the platforms and user characteristics, these factors can vary (Hudson et al, 2015). As such, it is important to detect content quality automatically (Ponzanelli et al, 2014b,a; Ya et al, 2015).

Overall, both Wang et al. (Wang et al, 2018) and Li et al. (Li et al, 2015) conclude that offering incentives as reputation scores is useful to improve post quality within a collaborative editing platform like Stack Overflow. This finding was also observed in other collaborative editing platforms like webcasts (Munteanu et al, 2008) and Wikipedia (Kittur and Kraut, 2008). Munteanu et al. (Munteanu et al, 2008) tested the effectiveness of engaged users collaborating in a wiki-like webcast platform to edit/correct transcripts that are produced from webcasts through an automated speech recognition system. Collaborative editing can be a cost-effective but useful means to improve the quality of the ASR (Automated Speech Recognition) system in webcasts because ASR systems can have an average error rate of 45% - above the accepted threshold of 25%. The field study carried out by the authors in a real lecture environment found that using students to edit the webcast transcript was useful in reducing the error rate. The editing was supported via a webcast extension that engages users to collaborate in a wiki-like manner. Kittur et al. (Kittur and Kraut, 2008) find that The increase in the number of editors does not guarantee the quality of the articles on Wikipedia.

The quality of the question is important to get an answer: lack of clarity, relatedness, and reproducibility of the problem, as well as the too-short question, could dissuade developers from answering to the question (Asaduzzaman et al, 2013; Mondal et al, 2019). The reputation and past activity of an asker could also factor into the likelihood of a question getting resolved (Rahman and Roy, 2015). As such, factors of good questions are investigated, e.g., code-to-text ratio, etc. (Calefato et al, 2018; Duijn et al, 2015). However, depending on the platforms and user characteristics, these factors can vary (Hudson et al, 2015). As such, it is important to detect content quality automatically (Ponzanelli et al, 2014b,a; Ya et al, 2015). Wang et al. (Wang et al, 2018) found that users who make more edits in a short time are likely to get more edits rejected. Thus, bad edits can harm the quality of the content.

Our research on Stack Overflow rollback edits initially started in 2019 to better understand the edit rejection reasons, as reported by Wang et al. (Wang et al, 2018). Through our qualitative analysis of Stack Overflow posts, we also found all the edit rejection reasons reported by Wang et al. (Wang et al, 2018). In addition, we found a few more reasons for edit rejection. While the above papers, including Wang et al. (Wang et al, 2018), focus on analyzing editing mechanisms in collaborative platforms based on empirical studies, our paper focuses on understanding the inconsistencies that may arise due to the editing preferences of different users and developing techniques and tools `iEdit` to assist Stack Overflow users avoid such inconsistencies during their editing behavior. As such, our developed tool `iEdit` can further contribute to supporting the content quality in social forums by assisting users with guidance on improving the quality of their suggested content. Thus, our paper offers complementary viewpoints to the above studies by offering tools and techniques that can facilitate improved edit content in a social Q&A site like Stack Overflow.

8.2 Techniques to Develop to Improve Collaborative Editing Systems

Collaborative editing systems are common in Wikipedia (Li et al, 2015; Kittur and Kraut, 2008), GitHub code editing (Dabbish et al, 2012), webcasts (Munteanu et al, 2008), scientific contents (Lowry et al, 2005; Calvo et al, 2005), and so on. Compared to substantial research on conducting studies on existing collaborative editing systems, we are not aware of much research that focused on developing tools and techniques to improve the systems. This is perhaps due to the fact that currently available collaborative platforms like Wikipedia seem to work well and are hugely popular. In all these platforms, the focus of collaborative editing is to improve the quality of the shared content based on user engagement (Agichtein et al, 2008).

Chen et al. (Chen et al, 2017) observed that most of the edits in Stack Overflow are small sentence edits. While developing their SOTorrent database, Baltes et al. (Baltes et al, 2018) also observed that the majority of edits in Stack Overflow are relatively small. In a follow-up study, Chen et al. (Chen et al, 2018) predicted whether a post needs to be edited. Their approach is based on the concept of 'proactive policy assurance', which assures that a modification to a suggested edit will satisfy the current 'reactive policy assurance' in Stack Overflow, which accepts/rejects based on the matching of existing editing policy after an edit is submitted (i.e., reactive). They developed a deep-learning policy assurance tool to recommend potential mid-level edits to a given post content to post owners or other users. The deep learning model is a CNN (Convolutional Neural Network). In a large-scale experiment, they found that the tool offers good precision, recall, and F1-score (at least 0.7) while suggesting mid-level edits.

As we noted in Section 8.1, our research of this paper started in 2019 to gain a hands-on experience on the edit rejection reasons observed by Wang et al. (Wang et al, 2018). Our initial exploration led to an expansion of the edit rejection reasons and to the submission of a registered protocol report in 2020 (Mondal et al, 2020). In the registered protocol report, we outlined our vision for this paper by offering to develop a Machine Learning (ML) model to automatically detect the reasons for edit rejection. While working on this paper, we observed that some edit rejection reasons could be present both in accepted and rejected edits, resulting in *inconsistencies* in the editing acceptance/rejection process. We reported a catalog of such inconsistencies in our MSR 2021 paper (Mondal et al, 2021b). In our MSR 2021 paper, we also report several rule-based tools that we developed to automatically detect the inconsistencies in Stack Overflow edits. In this paper, we extend our MSR 2021 paper with a tool `iEdit` that can guide developers with suggestions on how not to introduce such inconsistencies during their edits.

9 Conclusions

The editing system in Stack Overflow encourages developers to improve the posted content by suggesting edits. However, rollbacks can reject these suggested edits by rollbacks due to unsatisfactory, low-quality edits or violating edit guidelines. Unfortunately, biases in determining whether an edit is satisfactory or unsatisfactory can result in inconsistencies in rollback decisions. To understand the types and prevalence of such inconsistencies in rollback edits, we manually analyzed 764 rollback edits in Stack Overflow using standard principles of open coding. We found eight types of inconsistencies in the rollback edits. In both empirical and user studies, we find that inconsistencies can negatively impact the popularity and quality of the posts. We develop rule-based algorithms and ML models to detect inconsistencies automatically. Both techniques can detect inconsistent edits with 88%–99% accuracy. We then introduce an online tool called `iEdit` based on the developers' requirements. `iEdit` works with the Stack Overflow edit system to support users in identifying inconsistent edits with their rejection possibility. Next, we survey developers to find the effectiveness of `iEdit`. According to their responses `iEdit` is easy to use with the minimum cognitive workload and capable of offering valuable suggestions. Our findings can guide – (1) *forum designers* to develop an editing system properly, (2) *forum users* to understand inconsistent factors in suggested edits, and (3) *researchers* in software engineering to investigate tools and techniques to guide forum users and designers to handle inconsistent edits properly.

Supporting Data

Supporting Data can be found in our online appendix (Mondal et al, 2022b).

Acknowledgment

This research is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery grants, an NSERC Collaborative Research and Training Experience (CREATE) grant, and two Canada First Research Excellence Fund (CFREF) grants coordinated by the Global Institute for Food Security (GIFS) and the Global Institute for Water Security (GIWS).

References

Abric D, Clark OE, Caminiti M, Gallaba K, McIntosh S (2019) Can duplicate questions on stack overflow benefit the software development community? In: 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR), IEEE, pp 230–234

Agichtein E, Castillo C, Donato D, Gionis A, Mishne G (2008) Finding high-quality content in social media. In: Proceedings of the 2008 International Conference on Web Search and Data Mining, pp 183–194

Ahmed S, Bagherzadeh M (2018) What do concurrency developers ask about?: A large-scale study using stack overflow. In: Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, p Article No. 30

Asaduzzaman M, Mashiyat AS, Roy CK, Schneider KA (2013) Answering questions about unanswered questions of stack overflow. In: Proceedings of the 10th Working Conference on Mining Software Repositories, pp 87–100

Bagherzadeh M, Khatchadourian R (2019) Going big: A large-scale study on what big data developers ask. In: Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ACM, New York, NY, USA, ESEC/FSE 2019, pp 432–442

Bagozzi RP, Dholakia UM (2006) Open source software user communities: A study of participation in linux user groups. Journal of Management Science 52(7):1099–1115

Baltes S, Dumani L, Treude C, Diehl S (2018) Sotorrent: reconstructing and analyzing the evolution of stack overflow posts. In: Proceedings of the 15th International Conference on Mining Software Repositories, pp 319 − 330

Boslaugh S (2012) Statistics in a nutshell: A desktop quick reference. "O'Reilly Media, Inc."

Boulis C, Ostendorf M (2005) Text classification by augmenting the bag-of-words representation with redundancy-compensated bigrams. In: Proc. of the International Workshop in Feature Selection in Data Mining, Citeseer, pp 9–16

Calefato F, Lanubile F, Novielli N (2018) How to ask for technical help? evidence-based guidelines for writing questions on stack overflow. Journal of Information and Software Technology 94:186–207

Calvo RA, O'Rourke ST, Jones J, Yacef K, Reimann P (2005) Collaborative writing support tools on the cloud. IEEE Transactions on Learning Technologies 41:66–99

Cao A, Chintamani KK, Pandya AK, Ellis RD (2009) Nasa tlx: Software for assessing subjective mental workload. Behavior research methods 41(1):113–117

Chen C, Xing Z, Liu Y (2017) By the community & for the community: A deep learning approach to assist collaborative editing in q&a sites. In: Proceedings of the ACM on Human-Computer Interaction, p Article 32

Chen C, Chen X, Sun J, Xing Z, Li G (2018) Data-driven proactive policy assurance of post quality in community q&a sites. In: Proceedings of the ACM on Human-Computer Interaction, p Article 33

Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pp 785–794

Cohen J (1960) A coefficient of agreement for nominal scales. Educational and psychological measurement 20(1):37–46

Cohen J (1968) Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. Psychological bulletin 70(4):213

Dabbish L, Stuart C, Tsay J, Herbsleb J (2012) Social coding in github: transparency and collaboration in an open software repository. In: Proceedings of the

ACM conference on Computer Supported Cooperative Work, pp 37–46
Duijn M, Kucera A, Bacchelli A (2015) Quality questions need quality code: Classifying code fragments on stack overflow. In: Proceedings of the IEEE/ACM 12th Working Conference on Mining Software Repositories, pp 410–413
Exchange S (Accessed on: December 2019) How does reputation work? URL `https://meta.stackexchange.com/questions/7237/how-does-reputation-work`
Groves RM, Fowler J FJ, Couper MP, Lepkowski JM, Singer E, Tourangeau R (2011) Survey methodology
Hart SG, Staveland LE (1988) Development of nasa-tlx (task load index): Results of empirical and theoretical research. In: Advances in psychology, vol 52, Elsevier, pp 139–183
Hudson N, Chilana PK, Guo X, Day J, Liu E (2015) Understanding triggers for clarification requests in community-based software help forums. In: Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing, pp 189–193
Joshi A, Kale S, Chandel S, Pal DK (2015) Likert scale: Explored and explained. CJAST
Kitchenham BA, Pfleeger SL (2008) Personal opinion surveys. In: Guide to advanced empirical software engineering
Kittur A, Kraut RE (2008) Harnessing the wisdom of crowds in wikipedia: quality through coordination. In: Proceedings of the ACM conference on Computer supported cooperative work, pp 37–46
Lakhani KR, von Hippel E (2003) How open source software works: free user-to-user assistance. Journal of Research Policy 32(6):923–943
Li G, Zhu H, Lu T, Ding X, Gu N (2015) Is it good to be like wikipedia?: Exploring the trade-offs of introducing collaborative editing model to q&a sites. In: Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, pp 1080–1091
Lowry PB, Curtis AM, Lowry MR (2005) A taxonomy of collaborative writing to improve empirical research, writing practice, and tool development. Journal of Business Communication 41:66–99
Macbeth G, Razumiejczyk E, Ledesma RD (2011) Cliff's delta calculator: A non-parametric effect size program for two groups of observations. Universitas Psychologica 10(2):545–555
Mann HB, Whitney DR (1947) On a test of whether one of two random variables is stochastically larger than the other. The annals of mathematical statistics
Manning CD, Raghavan P, Schütze H (2009) An Introduction to Information Retrieval. Cambridge Uni Press
McHugh ML (2013) The chi-square test of independence. Biochemia medica 23(2):143–149
Mondal S, Rahman MM, Roy CK (2019) Can issues reported at stack overflow questions be reproduced?: an exploratory study. In: Proceedings of the 16th International Conference on Mining Software Repositories, pp 479–489
Mondal S, Uddin G, Roy CK (2020) Automatic identification of rollback edit with reasons in stack overflow q&a site. In: 36th IEEE International Conference on Software Maintenance and Evolution (ICSME) – Registered Protocol Report, pp 856–856
Mondal S, Saifullah CK, Bhattacharjee A, Rahman MM, Roy CK (2021a) Early detection and guidelines to improve unanswered questions on stack overflow. In:

14th Innovations in Software Engineering Conference (formerly known as India Software Engineering Conference), pp 1–11

Mondal S, Uddin G, Roy CK (2021b) Rollback edit inconsistencies in developer forum. In: 2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR), IEEE, pp 380–391

Mondal S, Rahman MM, Roy CK, Schneider K (2022a) The reproducibility of programming-related issues in stack overflow questions. Empirical Software Engineering 27(3):1–52

Mondal S, Uddin G, Roy CK (2022b) Automatic assistance for rollback edit inconsistencies in stack overflow. URL `https://github.com/saikatmondal/IEdit`

Munteanu C, Baecker R, Penn G (2008) Collaborative editing for improved usefulness and usability of transcript-enhanced webcasts. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp 373–382

Parnin C, Treude C, Grammel L, Storey MA (2012) Crowd documentation: Exploring the coverage and the dynamics of api discussions on stack overflow. Tech. rep., Georgia Tech

Polikar R (2006) Ensemble based systems in decision making. IEEE Circuits and systems magazine 6(3):21–45

Ponzanelli L, Mocci A, Bacchelli A, Lanza M (2014a) Improving low quality stack overflow post detection. In: In Proceedings of the 30th International Conference on Software Maintenance and Evolution, pp 541–544

Ponzanelli L, Mocci A, Bacchelli A, Lanza M (2014b) Understanding and classifying the quality of technical forum questions. In: Proceedings of the 14th International Conference on Quality Software, pp 343–352

Rahman MM, Roy CK (2015) An insight into the unresolved questions at stack overflow. In: Proceedings of the 12h Working Conference on Mining Software Repositories, pp 426–429

Singer J, Vinson NG (2002) Ethical issues in empirical studies of software engineering. TSE

Sözen E, Güven U (2019) The effect of online assessments on students' attitudes towards undergraduate-level geography courses. International Education Studies 12(10):1–8

StackExchange (2008) Meta stack exchange. URL `https://meta.stackexchange.com/questions/2950`, online; Last accessed October 2022

StackExchange (2009a) Are taglines and signatures disallowed? URL `https://meta.stackexchange.com/questions/5029`, online; Last accessed February 2020

StackExchange (2009b) Should 'hi', 'thanks', taglines, and salutations be removed from posts? URL `https://meta.stackexchange.com/questions/2950`, online; Last accessed February 2020

StackExchange (2009c) What is a 'rollback'? URL `https://meta.stackexchange.com/questions/17038`, online; Last accessed February 2020

StackExchange (2009d) What is a 'rollback'? URL `https://meta.stackexchange.com/questions/17038/what-is-a-rollback`, online; Last accessed February 2020

StackExchange (2012a) Editing policy is contradictory and unclear. URL `https://meta.stackexchange.com/questions/138262`, online; Last accessed February 2020

StackExchange (2012b) Is it right to rollback an edit that only removed "thanks a lot!". URL `https://meta.stackexchange.com/questions/146530`, online; Last accessed February 2020

StackExchange (2012c) What are the terms and conditions for editing posts on stack overflow? URL https://meta.stackexchange.com/questions/149839, online; Last accessed February 2020

StackExchange (2012d) When is "edit"/"update" appropriate in a post? URL https://meta.stackexchange.com/questions/127639, online; Last accessed February 2020

StackExchange (2013a) Issues with stack overflow users editing my questions. URL https://meta.stackexchange.com/questions/165463, online; Last accessed February 2020

StackExchange (2013b) What's wrong with putting "edit: ...." in the body of a post? URL https://meta.stackexchange.com/questions/202472, online; Last accessed February 2020

StackExchange (2015) What are some guidelines for editing questions, and how to respond to massive edits? URL https://meta.stackexchange.com/questions/253784, online; Last accessed February 2020

StackExchange (2016) Lack of consistency with offensive posts (multiple communities). URL https://meta.stackexchange.com/questions/274042, online; Last accessed February 2020

StackExchange (2017) Clarify editing guidelines. URL https://meta.stackexchange.com/questions/300211/clarify-editing-guidelines, online; Last accessed February 2020

StackExchange (2018) What is the etiquette for modifying posts? URL https://meta.stackexchange.com/questions/11474, online; Last accessed February 2020

StackExchange (2019a) Database schema documentation for the public data dump and sede. URL https://meta.stackexchange.com/questions/2677

StackExchange (2019b) StackExchage API. URL http://data.stackexchange.com/stackoverflow

StackOverflow (2008a) C# common library. URL https://stackoverflow.com/posts/498249/revisions?page=1, online; Last accessed February 2020

StackOverflow (2008b) Grouping activerecord query by a child attribute. URL https://stackoverflow.com/posts/125523/revisions?page=1, online; Last accessed February 2020

StackOverflow (2008c) How do i make an html page print in landscape when the user selects 'print'? URL https://stackoverflow.com/posts/37162/revisions?page=1, online; Last accessed February 2020

StackOverflow (2008d) How do you change the size of figures drawn with matplotlib? URL https://stackoverflow.com/posts/332311/revisions?page=1, online; Last accessed February 2020

StackOverflow (2008e) How do you compare two version strings in java? URL https://stackoverflow.com/posts/11024200/revisions?page=1, online; Last accessed February 2020

StackOverflow (2008f) Is there a performance difference between i++ and ++i in c++? URL https://stackoverflow.com/posts/24904/revisions?page=1, online; Last accessed February 2020

StackOverflow (2008g) What does the term "canonical form" or "canonical representation" in java mean? URL https://stackoverflow.com/posts/280121/revisions?page=1, online; Last accessed February 2020

StackOverflow (2010) What's wrong with the positioning in this very simple example? (ms ie 8). URL https://stackoverflow.com/questions/3774926, online;

Last accessed February 2020
StackOverflow (2011a) Php/mysql: How to know if update was successful? (when data sent to update was the same as that already in db). URL https://stackoverflow.com/posts/6218171/revisions?page=1, online; Last accessed February 2020
StackOverflow (2011b) put imageview src to hashmap¡string, string¿. URL https://stackoverflow.com/posts/30150411/revisions?page=1, online; Last accessed February 2020
StackOverflow (2012a) I am trying to export mysql data to a file using java, but i am not able to get the table headers. URL https://stackoverflow.com/posts/22529397/revisions?page=1, online; Last accessed February 2020
StackOverflow (2012b) Rate google play application directly in app. URL https://stackoverflow.com/posts/11270668/revisions?page=1, online; Last accessed February 2020
StackOverflow (2013a) How to access the current subversion build number? URL https://stackoverflow.com/posts/110175/revisions?page=1, online; Last accessed February 2020
StackOverflow (2013b) Mysql results issue in php array. URL https://stackoverflow.com/posts/14391452/revisions?page=1, online; Last accessed February 2020
StackOverflow (2015a) How do i make a good edit? URL https://meta.stackoverflow.com/questions/303219, online; Last accessed February 2020
StackOverflow (2015b) Provide more guidelines for reviewing edits. URL https://meta.stackoverflow.com/questions/295319, online; Last accessed February 2020
Subramanian S, Inozemtseva L, Holmes R (2014) Live api documentation. In: Proc. 36th International Conference on Software Engineering, p 10
Uddin G, Baysal O, Guerrouj L, Khomh F (2019) Understanding how and why developers seek and analyze API-related opinions. IEEE Transactions on Software Engineering pp 1–40
Vagias WM (2006) Likert-type scale response anchors. Clemson International Institute for Tourism & Research Development, Department of Parks, Recreation and Tourism Management Clemson University
Wang S, Chen THP, Hassan AE (2018) How do users revise answers on technical Q&A websites? a case study on stack overflow. IEEE Transactions in Software Enginering p 19
Ya Y, Tong H, Xie T, Akoglu L, Xu F, Lu J (2015) Detecting high-quality posts in community question answering sites. Journal of Information Sciences 302(1):70–82
Zhang T, Upadhyaya G, Reinhardt A, Rajan H, Kim M (2018) Are code examples on an online q&a forum reliable? a study of api misuse on stack overflow. In: Proc. 32nd IEEE/ACM International Conference on Software Engineering, p 12
Zhang Y, Jin R, Zhou ZH (2010) Understanding bag-of-words model: a statistical framework. International Journal of Machine Learning and Cybernetics 1(1):43–52