Automatic Identification of Rollback Edit with Reasons in Stack Overflow Q&A Site

Saikat Mondal
University of Saskatchewan, Canada
Email: saikat.mondal@usask.ca

Gias Uddin
University of Calgary, Canada
Email: gias.uddin@ucalgary.ca

Chanchal K. Roy
University of Saskatchewan, Canada
Email: chanchal.roy@usask.ca

***Abstract*—*Context.* The online technical Q&A site, Stack Overflow has changed the way software developers look for solutions. As such, content quality in Stack Overflow is paramount. Users in Stack Overflow can suggest improvement to a post (i.e., answer or question) by suggesting edits to the post.**

***Problem.* Recent research shows that a large number of suggested edits in Stack Overflow are being rejected by rollbacks due to undesired edits or violating edit guidelines. Such a scenario could hurt the quality of the shared content, frustrate, and demotivate users.**

***Objective.* This paper aims at improving the Stack Overflow edit system by automatically identifying the rollback edits with the potential reasons. This study also plans to introduce an online tool namely *EditEx*. EditEx can guide Stack Overflow users during their editing of a post by highlighting the potential causes of rollback.**

***Method.* With a view to understanding the rollback edit reasons, we conducted a case study by manually investigating 777 rollback edits in Stack Overflow. This study produced a catalog of 19 rollback edit reasons. We then identify a suite of predictor variables that could be useful to develop a machine learning-based classifier to automatically identify rollback edits. Next, we plan to investigate rule-based natural language techniques (e.g., ngrams, POS tagging, dictionary-based, etc.) to determine the possible rollback reason. We also plan to develop an online tool namely EditEx that can automatically guide Stack Overflow users during their editing of a post by highlighting potential causes of rollback. We offer details of an empirical study to assess the accuracy of the classifiers and a user study to assess the effectiveness of the online tool.**

***Index Terms*—Stack Overflow, Rollback edits, Classification Model, User Study**

I. Introduction

The adoption, growth, and continued success of an online question and answering (Q&A) site such as Stack Overflow depend on two major factors: participation of users and quality of the shared knowledge [2, 16, 21]. In Stack Overflow, the edit system is introduced to promote quality by allowing users to communicate on the quality of the questions and answers through editing. In particular, collaborative editing helps to keep questions and answers clear, relevant, and up-to-date. For example, users edit questions and answers to fix grammar and spelling mistakes, clarify the meaning, and add related resources or hyperlinks. Unfortunately, a number of suggested edits in Stack Overflow get rejected by *rollbacks* because of undesired editing (e.g., it does not satisfy the post owner), or violating edit guidelines [20]. Rollback means reverting a post (i.e., question or answer) to a previous version in the edit history [9]. The reverted version then appears as the most recent item in the edit history.

Manually identification of undesired edits or such edits that violate editing guidelines of Stack Overflow wastes a lot of community time and effort. For example, when one user raised such an issue of classifying undesired edits at meta.stackexchange, another user responded, *"It takes time to read and parse through those questions when I am trying to spend my time more efficiently reading through the actual question and figuring out how to appropriately answer it"* [8]. At least 921 users support it by casting upvotes. It indicates that the identification of undesired edit reasons manually that could be rejected by rollbacks is killing users' valuable time and also irritating them. On the other hand, users who suggested edits and later get rejected by rollback become frustrated because many users (especially novices) are not aware of editing guidelines. Besides, the current editing system of Stack Overflow does not support users with automatic suggestions that discourage undesired editing. Therefore, a study on automatic identification of rollback edit reasons is warranted to support users of Stack Overflow.

Realizing the need for an automated tool, some users have started writing personal scripts to identify undesired edits programmatically. For example, one user wrote a script to identify and remove greetings (e.g., hello, dear) automatically while reviewing suggested edits [8]. Such a scenario clearly shows the demand for a system that identify the potential undesired edit reasons given the recent focus on the quality of contents shared in Stack Overflow [32, 30, 13, 24, 1, 19], the suite of tools and techniques developed to detect and recommend quality posts [22, 23, 31, 11, 17, 5]. However, we are not aware of any existing edit assistance system that automatically identifies rollback edits with reasons to support the current editing system of Stack Overflow.

The edits in Stack Overflow were subject to a recent study by Wang et al. [28]. They analyzed 369 rollback edits of answers and identified 12 reasons (e.g., undesired text formatting, emotional sentence addition/removal). We extended their study manually analyzing a total of 777 rollback edits (382 questions + 395 answers). We found a total of 19 rollback edit reasons, seven of which were not reported by Wang et al. [28]. The newly derived seven reasons for rollback edits are status update, gratitude add/remove, greetings add/remove,

TABLE I: Rollback edit reasons

Reason	Description
Undesired Text Formatting	Changes of the font size, text cases (uppercase/lowercase), emphasizing text making them bold/italic, adding or removing space/newline, creating bullet/number list, and formatting text term as code term or vice versa.
Undesired Text Add/Remove	Addition of texts that have less or no impact on the quality of questions or answers, or removal of important texts.
Undesired Text Change	Changes of the sentence structures (*e.g.,* simple, complex), tenses (*e.g.,* present, past), voices (*e.g.,* active, passive), rewording, interchanging contractions by root words, acronyms/abbreviations by elaborations and vice versa.
Incorrect Text Change	Rewording with incorrect terms, grammatical and spelling mistakes, incorrect changes in software versions, or specifications.
Undesired Code Formatting	Modification of code indentation (e.g., adding or removing spaces and newlines), add/remove line numbers, splitting/merging code segments, changes in text cases (e.g., *"select"* to *"SELECT"* in a SQL query).
Undesired Code Add/Remove	Unwanted code statements are added or important statements are removed.
Undesired Code Change	Refactoring (e.g., variable renaming), changing APIs, and editing comments.
Incorrect Code Change	Changes datatype of variables, function return types, function arguments, arithmetic expressions.
Status Update	Status refers to the personal notes of post owners that are added to clarify confusion, append important messages that were missed during the submission time of their questions or answers, and also to acknowledge users' responses.
Emotion Add/Remove	Addition/removal of emotional words/sentences/emoticon.
Gratitude Add/Remove	Addition/removal of thanksgiving sentences (e.g., thank you, cheers!).
Greetings Add/Remove	Addition/removal of greeting/salutations (e.g., hello, hi, dear).
Undesired Reference Modification	Addition of inactive hyperlinks, inappropriate images, or diagrams with questions or answers. On the contrary, the removal of essential references or unreasonably modify them.
Signature Add/Remove	Addition/removal of user name, ID, links of personal sites.
Partial Acceptance	Revision is rolled back, but part of the changes are still accepted. Then, the accepted changes are included in later revisions.
Deprecation Note	Addition/removal of deprecation notes inside the body of an answer.
Duplication Note	Addition/removal of duplication notes inside the body of a question.
Introduce Spam	Deface the question or answer to promote product or service, insert garbage texts.
Other	An asker asked a question inside the answer, add a solution inside the question, interchange the position of texts.

signature add/remove, deprecation note, duplication note, and introduce spam. Table I summarizes the rollback edit reasons.

With a view to assisting Stack Overflow users, this paper aims at investigating the development of tools and techniques that can automatically offer insights about the 19 rollback edit reasons in Table I. We report the design of machine learning models based on textual features to automatically detect rollback edits, and rule-based natural language techniques (e.g., ngrams, POS tagging, dictionary-based, etc.) to determine the possible rollback reason. We introduce the design of web-based tool that we are currently developing based on the rule-based rollback reasons classification technique. The tool is called *EditEx*, which automatically highlights potential rollback edit reasons in the suggested edit of user, before he submits the edit for review. We present the details of a user study that we aim to conduct for analyzing the effectiveness of the tool.

II. Research Questions

This paper aims at assisting users in Stack Overflow by offering them automated guidance and support during their editing of post. We formulate two major research questions:

RQ1. To what extent do our classifiers predict the rollback edits with the potential reasons?

Identification of undesired editing reasons that cause rollbacks is important to promote quality editing. Such automatic identification could save users time and effort from differentiating the undesired and accepted edits manually. Specifically, we attempt to exploit cues in the textual contents of suggested edits to build a suite of classifiers that can automatically determine the rollback edits with potential rollback reasons of the suggested edit.

We define the following null hypothesis:

H_1: *The accuracy of our developed classifiers is not better than a random classifier with 50% accuracy.*

RQ2. To what extent can EditEx be helpful to users avoid rollback in their edits?

The actual impact of our developed classifiers can be assessed, if the classifiers can automatically guide a user during this editing process of a Stack Overflow post. Editing is a time consuming and largely voluntary activity in Stack Overflow. Therefore, efforts should be made to assist editors with a tool that can recommend them with fixes to their edits. The focus of the tool will be to reduce the likelihood of rollback of the suggested edits. The effectiveness of the tool can be assessed via the real-world usage of the tool by users in actual Stack Overflow contents and environment.

We define the following null hypothesis:

H_2: *Our EditEx tool is not effective to help users avoid rollback edits.*

III. Rejected Edit Prediction Model (RQ1)

A. Study Materials

We download the September 2019 data dump of Stack Overflow from the Stack Exchange site [10]. Our data dump contains a total of 116,473 rollback edits (72,159 questions + 44,314 answers) of body. We use a confidence level of 95% with a confidence interval of 5% [3] to calculate the statistically significant random sample size for rollback edits of questions and answers. Therefore, we create one sample for questions and another sample for answers. In particular, we randomly sampled (1) 382 from 72,159 rejected question revisions, and (2) 395 from 44,314 rejected answer revisions for manual analysis. The first two authors used an open card sorting approach to label the reasons for each of the 777 rollback edits [14]. We manually identify the reasons of rollback for each edit in the dataset. This process has produced a list of 19 rollback edit reasons as shown in Table I. We use this dataset for our evaluation. We share this manually analyzed dataset in our online appendix[1].

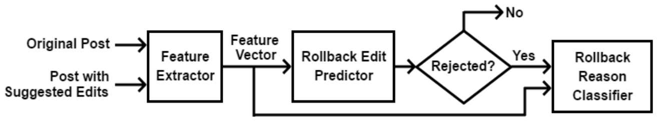


Fig. 1: Classification technique of rollback edits with reasons.

B. Proposed Models

Fig. 1 shows the proposed classification technique. Our classification pipeline includes three components. The inputs to the *Feature Extractor* are the original post and the suggested edits to it. The output is a list of feature vectors based on the values of the predictor variables as follows.

- **Editing Distance.** This variable represents the edit distance between two subsequent revisions. We will measure such distance using *Levenshtein* distance [29].
- **Emotion.** This variable represents whether editing is due to add or remove emotion. We will use *EmoTxt* to find emotion [4].
- **Status.** This variable represents whether editing adds or removes personal status. We will use a regular expression-based or heuristic-based approach to find the value of this predictor. We plan to encode such a predictor using three values: 1 for addition, -1 for removal, and 0 for no existence. A similar approach will be applied to find the values of the following predictors.
- **Gratitude.** This variable represents whether editing adds or removes gratitude.
- **Greetings.** This variable represents whether editing adds or removes greetings.
- **Reference.** This variable represents whether editing modifies references.
- **Signature.** This variable represents whether editing adds or removes signatures.
- **Deprecation.** This variable represents whether editing adds or removes deprecation notes.
- **Duplication.** This variable represents whether editing adds or removes duplication notes.

The *Rollback Edit Predictor* takes as input the feature vectors and outputs a dichotomous variable “Rejected”. The value of “Rejected” Is 1, if the predictor determines that the suggested edit will most likely be rejected and it is 0 otherwise. If the value is 1 for the rejected variable, we then take as input the corresponding feature vector into another classifier *Rollback Reason Classifier*, which outputs the likely reasons for rollback.

For the Feature Extractor, we will apply a suite of existing techniques or our heuristic-based approaches to generate the feature vectors. For the Rollback Edit Predictor, we will investigate both generative and ensemble machine learning classifiers, such as Random Forest, XGBoost and so on. We will pick the classifier with the best performance as the final predictor. We will train and test the performance of the classifiers on a manually created benchmark dataset. A total of three human coders will be used to produce the benchmark. Following standard practices in such a benchmark creation process [26], the authors will discuss a subset of the sample and produce labels. This ensures that all authors agree on a common set of guidelines. The first author will then manually label a subset of the benchmark. The second/third author will then manually label the same subset. The labels between the first author and the other authors will be compared using Cohen’s kappa (k) metric. If the agreement is not substantial (k>= 0.61), the first author will revisit the disagreements and will modify the labels again. The first author then picks another subset of the benchmark and the authors repeat the above process. This process is repeated until the agreements between the authors are at least substantial. At every stage, the external expert will be consulted to resolve the disagreements. Once a substantial agreement is reached, the first author then labels the rest of the benchmark. For the Rollback Reason Classifier, we will investigate rule-based natural language techniques (e.g., ngrams, POS tagging, dictionary-based, etc.) to determine the possible rollback reason. We report the accuracy of the techniques in each component using three standard information retrieval metrics such as – precision, recall, and F1-score.

C. Accuracy Analysis

To see whether our model identifies the potential rollback reasons accurately, we will manually label statistically sig-

[1] https://figshare.com/articles/Dataset/12408782

nificant edits (rollback + accepted) from testing samples in a file as follows. **Got**: the reasons detected by the model. **Expected**: the actual reasons based on our manual analysis. We then create a confusion matrix to analyze the performance of the proposed model as follows. *True Positive (TP).* 'got' reasons = 'expected' reasons, *False Positive (FP).* 'got' reasons $\neq$ 'expected' reasons or 'got' reasons but 'expected' no reasons, *True Negative (TN).* 'got' no reasons and 'expected' no reasons, and *False Negative (FN).* 'got' no reasons but 'expected' one or more reasons.

Using the above matrix, we will determine four standard metrics (Precision P, Recall R, F1-score F, and Accuracy A) to compute the performance of the proposed model [18].

IV. A Recommendation Engine to Suggest Fixes to Suggested Edits To Avoid Rollback (RQ2)

To support the current editing system of Stack Overflow, we aim to develop a web-based tool that identifies the potential rollback edit reasons. The tool is called EditEx. We aim to share EditEx with Stack Oveflow users as a browser plug-in. Upon installation of the plug-in in a user browser, the tool recommends potential fixes to the edits suggested by the user in Stack Overflow. Based on what users are editing, the system will automatically extract the edit reasons. EditEx consults our developed Rollback Reason Classifier that identifies the potential reasons if the edit is classified as rollback edit (Fig. 1). It will get the potential reasons from that classifier. Then, it will mine some patterns using regular expressions to detect the sentences/keywords and then highlights the potential contents in the suggested edit that could induce a rollback of the edit. For example, if the potential reason of rollback is gratitude add/remove, EditEx will look for several keywords related to gratitude such as welcome, thanks, sorry, appreciated, thank, ty (i.e., thank you), thx, regards, and tia (ı.e., thanks in advance) using regular expressions. It will highlight the sentence accordingly. Figure 2 shows a partial view of the proposed web-based system. The first block in 2 (left most) shows the contents of a post that a user would like to edit. The middle block shows the suggested edits to the post in green. The 'Suggest Me' button will be shown to the user, whenever he starts to edit the post. Upon click, the 'Suggest Me' button will analyze the suggested edits and match those with potential rollback edit reasons. The reasons will be detected based on the classifiers we developed. The tool then highlights the potentially susceptible textual contents that could lead a rollback of the suggested edit. The tool also suggests the potential fixes of the highlight (see the right most block in Fig. 2).

We aim to investigate the effectiveness of EditEx by determining how the tool can help a Stack Overflow user during his editing of Stack Overflow posts. We do this by studying actual Stack Overflow users in real-world editing tasks. We discuss the design of the study below.

A. Participants

We will adopt a snowball approach to recruit our study participants. First, we will recruit a list of participants who have a sufficient amount of editing experience in Stack Overflow. This list of users will be collected based on personal contacts. We will then ask the participants to recommend other users with similar editing experience. Each participant should have completed at least 100 edits prior to our recruitment. This is to ensure that the participant has sufficient experience with the Stack Overflow edit system. Each participant will be trained using a coding guide to learn about our developed tool. Each participant will be properly informed of the editing tasks.

B. Execution Plan

We divide the participants into two groups:

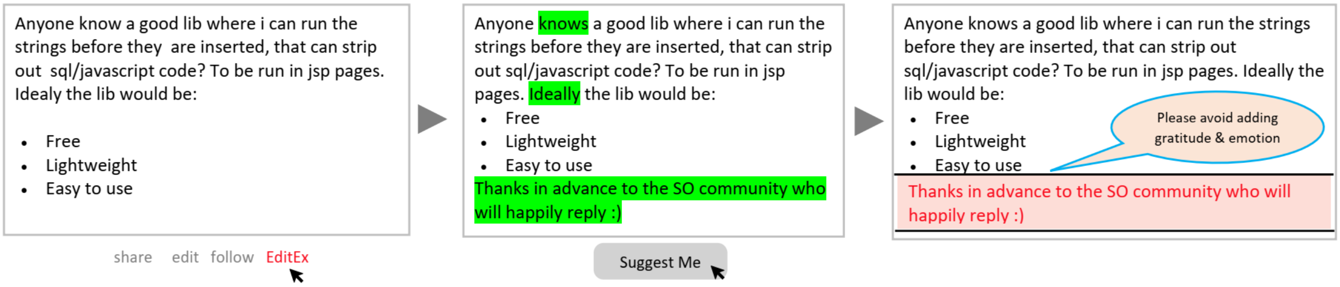
- **Treatment.** Each participant in this group will be assisted in their editing of Stack Overflow posts by our developed EditEx tool. The participant will also have access to the standard Stack Overflow edit system.
- **Control.** Each participant in this group will edit a Stack Overflow post by using the standard Stack Overflow edit system only.

Each participant (control and treatment) will be given links to the official Stack Overflow editing guidelines. All the editing tasks will follow the editing guidelines. We plan to recruit at least 30 participants (15 for the treatment group and 15 for the control group) for our study. Each participant will be asked to edit ten posts. So, each group (treatment/control) will edit at least 150 posts. A post can be an answer or a question. We will ask the participants to edit posts based on the identified reasons that cause rollbacks. For example, both the treatment and control groups will be asked to add a simple 'thank you' note to the suggested edit. In this case of the treatment group, the EditEx tool will warn them against adding such gratitudinal notes, which the control group will not be getting such guidance. To also mitigate individual bias, both control and treatment groups will make a particular type of suggestions (e.g., gratitudinal) to the same user. This means that we will pick two questions/answers to a given user. We will give one to the control group and another to the treatment group. Both groups will decide on whether to add gratitudinal-type suggestions into the edits. We will also ask the participants to edit posts arbitrarily that may cover the false positives. That is, participants will edit posts that are not related to any particular rollback edit reasons to limit the bias of this study. We will monitor the status of the suggested edit for 15 days after the suggestion is submitted by our study participant. This threshold of 15 days is picked based on our analysis on the average/median number of days it takes to decide on a suggested edit. We will record three types of status updates for each suggested edit: Rejected, Approved, Undecided. After the completion of the 10 edit tasks, each participant will be invited to complete a short survey. The survey questions are as follows:

Q1. How confident are you with your suggested edit?

Q2. What challenges did you face while using the Stack Overflow edit system?

Q3. (Only for the treatment group) Did the use of EditEx tool help you make better editing suggestion?



Anyone know a good lib where i can run the strings before they are inserted, that can strip out sql/javascript code? To be run in jsp pages. Idealy the lib would be:

- Free
- Lightweight
- Easy to use

Anyone knows a good lib where i can run the strings before they are inserted, that can strip out sql/javascript code? To be run in jsp pages. Ideally the lib would be:

- Free
- Lightweight
- Easy to use

Thanks in advance to the SO community who will happily reply :)

Anyone knows a good lib where i can run the strings before they are inserted, that can strip out sql/javascript code? To be run in jsp pages. Ideally the lib would be:

- Free
- Lightweight
- Easy to use

Please avoid adding gratitude & emotion

Thanks in advance to the SO community who will happily reply :)

Fig. 2: Proposed tool interface.

Q4. (Only for the treatment group) Would you use EditEx tool to help you edit in Stack Overflow from now on?

In addition, each participant will be also asked to assess the complexity and effort required for the editing tasks using the NASA Task Load Index (TLX) [12]. NASA TLX assesses the subjective workload of subjects. After completing each task, we asked each subject to provide their self-reported effort on the completed task through the official NASA TLX log engine at nasatlx.com. Each subject will be given a login ID, an experiment ID and task IDs, which they will use to log their effort estimation for each task.

C. Analysis Plan

For each participant, we count the total number of suggested edits that got rollback. Based on that, we compute the average number of rollback edits per user for both control and treatment groups. If the average number is lower for treatment group, that gives us a preliminary validation that our tool is useful. To further confirm the usefulness, We will find whether the difference of rollback edits between treatment and control groups is statically significant using the Mann Whitney U test, a non-parametric statistical significance test. P-value from the Mann Whitney U test can inform whether an effect exists. However, the p-value will not reveal the size of the effect. So, we plan to employ Cliff's delta to see how substantially different the two groups are.

We compute the efforts spent by each participant using the NASA TLX values as reported by the participants. We analyze the following five dimensions in the TLX metrics for each task under each setting: 1) *Frustration Level.* How annoyed versus complacent the developer felt during the coding of the task? 2) *Mental Demand.* How much mental and perceptual activity was required? 3) *Temporal Demand.* How much time pressure did the participant feel during the coding of the solution? 4) *Physical Demand.* How much physical activity was required. 5) *Overall Performance.* How satisfied was the participant with his performance? Each dimension is reported in a 100-points range with 5-point steps. A TLX 'effort' score is automatically computed as a task load index by combining all the ratings provided by a participant. Because the provided TLX scores were based on the judgment of the participants, they are prone to subjective bias. Detecting outliers and removing those as noise from such ordinal data is a standard statistical process [25]. By following Tukey, we only considered values between the following two ranges as valid: 1) Lower limit: First quartile - 1.5 * IQR 2) Upper limit: Third quartile + 1.5 * IQR . Here IQR stands for 'Inter quartile range', which is calculated as: IQR $= R3 - R1$. R1 and R3 stand for the first and third quartile, respectively. We then compare the efforts spent by participants between the control and treatment group by computing both average efforts as well as the statistical significance and effect size. While the above approach removes the outliers from the TLX scores, we will also compare the scores without removing the outliers to report the effect of the outliers on the overall TLX scores.

We analyze the survey responses of the participants as follows. The Q1 in Section IV-B has five confidence scales: -2, -1, 0, 1, 2. A value 2 denotes very confident. A value of -2 denotes not confident at all. We will compare the reported confidence values between treatment and control groups, similar to analysis of efforts. The responses to Q2 (i.e., challenges they faced) will be open-ended, i.e., participants can write anything they want. We will use open coding to analyze and report the responses, following previous studies [15, 27]. The responses to Q3 and Q4 will be binary, i.e., yes or no. We will report those by simply providing the percentages of participants responding to a yes/no.

V. Implications of Study Findings

The developed classifiers from our study can be used to automatically detect rollback edit reasons in Stack Overflow. This then can extend current tools and techniques that predominantly use contents from suggested edits to recommend editing suggestions (e.g., see the works of Chen et al. [6, 7]). The tool EditEx, once properly developed can help developers to use it alongside the current Stack Overflow edit system. This can help the reduction of rollback edit reasons in Stack Overflow and can improve the overall satisfaction of Stack Overflow users. In the long term, the tool can promote better contents, because users will be more motivated. Such high quality contents then can offer better content and recommendation support for tools and techniques that focus on the quality of contents shared in Stack Overflow [32, 30, 13, 24, 1, 19], the suite of tools and techniques developed to detect and recommend quality posts [22, 23, 31, 11, 17, 5].

References

[1] Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. Finding high-quality content in social media. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 183–194, 2008.

[2] Richard P. Bagozzi and Utpal M. Dholakia. Open source software user communities: A study of participation in linux user groups. *Journal of Management Science*, 52(7):1099–1115, 2006.

[3] Sarah Boslaugh. *Statistics in a nutshell: A desktop quick reference*. "O'Reilly Media, Inc.", 2012.

[4] Fabio Calefato, Filippo Lanubile, and Nicole Novielli. Emotxt: a toolkit for emotion recognition from text. In *2017 seventh international conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pages 79–80, 2017.

[5] Fabio Calefato, Filippo Lanubile, and Nicole Novielli. How to ask for technical help? evidence-based guidelines for writing questions on stack overflow. *Journal of Information and Software Technology*, 94:186–207, 2018.

[6] Chunyang Chen, Zhenchang Xing, and Yang Liu. 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*, page Article 32, 2017.

[7] Chunyang Chen, Zhenchang Xing, and Yang Liu. 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*, page Article No. 32, 2017.

[8] Stack Exchange. Should 'hi', 'thanks', taglines, and salutations be removed from posts?, 2009. URL https://meta.stackexchange.com/questions/2950/. Online; Last accessed February 2020.

[9] Stack Exchange. What is a 'rollback'?, 2009. URL https://meta.stackexchange.com/questions/17038/what-is-a-rollback. Online; Last accessed February 2020.

[10] Stack Exchange. StackExchage API, 2019. URL http://data.stackexchange.com/stackoverflow.

[11] F. Maxwell Harper, Daphne Raban, Sheizaf Rafaeli, and Joseph A. Konstan. Predictors of answer quality in online q&a sites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 865–874, 2008.

[12] S. G. Hart and L. E. Stavenland. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. pages 139–183, 1988.

[13] Nathaniel Hudson, Parmit K. Chilana, Xiaoyu Guo, Jason Day, and Edmund Liu. Understanding triggers for clarification requests in community-based software help forums. In *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing*, pages 189–193, 2015.

[14] William Hudson. *The Encyclopedia of Human-Computer Interaction*, chapter Card Sorting. The Interaction Design Foundation, 2 edition, 2013.

[15] Oleksii Kononenko, Olga Baysal, and Michael W. Godfrey. Code review quality: How developers see it. In *Proc. 38th International Conference on Software Engineering*, pages 1028–1038, 2016.

[17] Lei Li, Daqing He, Wei Jeng, Spencer Goodwin, and Chengzhi Zhang. Answer quality characteristics and prediction on an academic q&a site: A case study on researchgate. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1453–1458, 2015.

[16] Karim R Lakhani and Eric von Hippel. How open source software works: free user-to-user assistance. *Journal of Research Policy*, 32(6):923–943, 2003.

[18] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *An Introduction to Information Retrieval*. Cambridge Uni Press, 2009.

[19] Saikat Mondal, Mohammad Masudur Rahman, and Chanchal K. Roy. Can issues reported at stack overflow questions be reproduced?: an exploratory study. In *Proceedings of the 16th International Conference on Mining Software Repositories*, pages 479–489, 2019.

[20] Stack Overflow. How do i make a good edit?, 2015. URL https://meta.stackoverflow.com/questions/303219/how-do-i-make-a-good-edit. Online; Last accessed February 2020.

[21] Chris Parnin, Christoph Treude, Lars Grammel, and Margaret-Anne Storey. Crowd documentation: Exploring the coverage and the dynamics of api discussions on stack overflow. Technical report, Georgia Tech, 2012.

[22] Luca Ponzanelli, Andrea Mocci, Alberto Bacchelli, and Michele Lanza. Understanding and classifying the quality of technical forum questions. In *Proceedings of the 14th International Conference on Quality Software*, pages 343–352, 2014.

[23] Luca Ponzanelli, Andrea Mocci, Alberto Bacchelli, Michele Lanza, and David Fullerton. Improving low quality stack overflow post detection. In *Proceedings of the IEEE International Conference on Software Maintenance and Evolution*, pages 541–544, 2014.

[24] Mohammad Masudur Rahman and Chanchal K. Roy. An insight into the unresolved questions at stack overflow. In *Proceedings of the 12h Working Conference on Mining Software Repositories*, pages 426–429, 2015.

[25] John W. Tukey. *Exploratory Data Analysis*. Pearson, 1st edition, 1977.

[26] Gias Uddin and Foutse Khomh. Automatic mining of opinions expressed about apis in stack overflow. *IEEE Transactions on Software Engineering*, 2019.

[27] Gias Uddin, Olga Baysal, Latifa Guerrouj, and Foutse Khomh. Understanding how and why developers seek and analyze API-related opinions. *IEEE Transactions on Software Engineering*, pages 1–40, 2019.

[28] Shaowei Wang, Tse-Hsun (Peter) Chen, and Ahmed E. Hassan. How do users revise answers on technical Q&A websites? a case study on stack overflow. *IEEE Transactions in Software Engineing*, page 19, 2018.

[29] Wikipedia. Levenshtein distance, 2020. URL https://en.wikipedia.org/wiki/Levenshtein_distance. Online; Last accessed February 2020.

[30] Yuan Ya, Hanghang Tong, Tao Xie, Leman Akoglu, Feng Xu, and Jian Lu. Want a good answer? ask a good question first! Technical report, arXiv preprint arXiv:1311.6876, 2013.

[31] Yuan Ya, Hanghang Tong, Tao Xie, Leman Akoglu, Feng Xu, and Jian Lu. Detecting high-quality posts in community question answering sites. *Journal of Information Sciences*, 302 (1):70–82, 2015.

[32] Tianyi Zhang, Ganesha Upadhyaya, Anastasia Reinhardt, Hridesh Rajan, and Miryung Kim. Are code examples on an online q&a forum reliable?: a study of api misuse on stack overflow. In *In Proceedings of the 40th International Conference on Software Engineering*, pages 886–896, 2018.