

Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Computing and
Information Systems

School of Computing and Information Systems

7-2023

Machine-learning approach to automated doubt identification on stack overflow comments to guide programming learners

Tianhao CHEN

Singapore Management University, thchen.2020@scis.smu.edu.sg

Eng Lieh OUH

Singapore Management University, elouh@smu.edu.sg

Kar Way TAN

Singapore Management University, kwtan@smu.edu.sg

Siaw Ling LO

Singapore Management University, sllo@smu.edu.sg

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research



Part of the [Numerical Analysis and Scientific Computing Commons](#), and the [Programming Languages and Compilers Commons](#)

Citation

CHEN, Tianhao; OUH, Eng Lieh; TAN, Kar Way; and LO, Siaw Ling. Machine-learning approach to automated doubt identification on stack overflow comments to guide programming learners. (2023). *Proceedings of 2023 Pacific Asia Conference on Information Systems, Nanchang, China, July 8-12*. 1-16. Available at: https://ink.library.smu.edu.sg/sis_research/8066

This Conference Proceeding Article is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email cherylids@smu.edu.sg.

Machine-Learning Approach to Automated Doubt Identification on Stack Overflow Comments to Guide Programming Learners

Completed Research Paper

Tian Hao Chen

Singapore Management University
80 Stamford Rd, Singapore 178902
thchen.2020@scis.smu.edu.sg

Eng Lieh Ouh

Singapore Management University
80 Stamford Rd, Singapore 178902
elouh@smu.edu.sg

Kar Way Tan

Singapore Management University
80 Stamford Rd, Singapore 178902
kwtan@smu.edu.sg

Siaw Ling Lo

Singapore Management University
80 Stamford Rd, Singapore 178902
slllo@smu.edu.sg

Abstract

Stack Overflow is a popular Q&A platform for developers to find solutions to programming problems. However, due to the varying quality of user-generated answers, there is a need for ways to help users find high-quality answers. While Stack Overflow's community-based approach can be effective, important technical aspects of the answer need to be captured, and users' comments might contain doubts regarding these aspects. In this paper, we showed the feasibility of using a machine learning model to identify doubts and conducted data analysis. We found that highly reputed users tend to raise more doubts; most answers have doubt in the first comment, and many answers have unsolved doubt in the last comment; high-score and low-score answers are equally likely to contain doubts in comments. Our classifier and findings can provide users with a new perspective on determining answers' helpfulness and allow expert users to easily locate doubts to address.

Keywords: Stack Overflow, Doubt Identification, Text Analytics

Introduction

Stack Overflow is a Community Question Answering (CQA) platform that features a wide range of programming and software development topics. Since Stack Overflow was first established in 2008, it has become one of the essential platforms for professional programmers and enthusiasts to search for answers to various programming problems. With the participation of advanced programmers, Stack Overflow has benefited programming beginners and students. Studies (Bhasin et al. 2021; Lu et al. 2022) have shown that using Stack Overflow has helped learners feel more motivated to participate, gain experiences in collaborative development and learn from others.

While many helpful posts on Stack Overflow exist, the answer quality varies drastically. The quality of user-generated content on a CQA website is diverse due to users' varying expertise and educational backgrounds (Yang et al. 2019). Stack Overflow has some measures implemented to gauge answer quality, for instance, the voting and reputation systems. Each question on Stack Overflow may be voted up or down and receive

multiple answers and comments. Each answer submitted by the community may be voted up or down and receive multiple comments. Each comment may receive votes too although comment voting is less common. Each user on Stack Overflow has a reputation score and may receive badges indicating the contributions.

Although these are useful indicators of the quality of the answers, there are limitations. Given the decentralised nature of Stack Overflow, an incentive system like the voting and reputation system is prone to manipulation, such as voting rings (Mazloomzadeh et al. 2021) where communities form to upvote each other repeatedly. Moreover, the reputation point on Stack Overflow is not always a good indicator of users' expertise (Wang et al. 2021), meaning the answerer's reputation may not represent the answer's quality. According to the Stack Overflow 2022 Developer Survey¹ of 73268 software developers, over 80% of the respondents visit the website at least a few times weekly. However, only around 13% participate at the same frequency, whereby participating means asking, answering, voting, and commenting. This discrepancy between viewing and participation can potentially hinder how accurately the voting score and reputation can gauge the answer quality on the platform, as many answers simply do not get enough attention from users and hence receive very few votes, making it harder for programming learners to differentiate quality answers. Furthermore, the votes are applied to individual posts (i.e., the question and each answer) and hence are unable to capture the quality of the 'stack' (referred to as the combination of the question, answers, and comments) as a whole. The silo nature of votes makes the voting system unable to capture important technical aspects such as compatibility in different environmental settings and changes in the versions of the programming language over time. In such situations, other users (especially inexperienced users) may comment, ask questions, or raise limitations under the comment section within the answer (Zhang et al. 2019), indicating that the answer does not apply in certain situations.

In this paper, we draw inspiration from the concept of 'doubts' (Lo et al. 2021) where 'doubts' can be a question or simply a statement requesting more information and further clarifications. We envision that an answer on Stack Overflow can be 'doubtful' when there exist comments which indicate further questions or seek more information from the community. We aim to identify 'doubts' automatically via text analytics methods to help users (especially programming learners) to find high-quality and helpful answers on top of the existing Stack Overflow metrics such as voting and reputation scores. The doubt indicator proposed in this research serves as an additional feature for Stack Overflow, complementing the existing metrics. By incorporating the doubt indicator, users can have an additional perspective when viewing post comments. We hope that with our new 'doubt' indicator, we can help programming learners better identify quality answers.

This paper shall address the following two research questions:

RQ1. Can machine learning models identify doubts automatically in Stack Overflow comments?

RQ2. What are the relationships between the presence of doubts among comments under the answer post and Stack Overflow's existing voting and reputation metrics?

To classify doubts, our methodology in this research explored both the knowledge-based Sentic Pattern Detection (SPD) model (Lo et al. 2021) where the model was trained using data from a set of student reflection data collected in a course and a pre-trained language model, i.e., RoBERTa (Liu et al. 2019). The SPD model was based on specific vocabulary related to the dataset while a pre-trained model was exposed to a large variety of vocabulary that may not be specific to the dataset. We found interesting results that the pre-trained language model RoBERTa provided more promising results than SPD. We chose RoBERTa because it was trained on an order of magnitude more data than BERT for a longer time, allowing its representations to generalise even better to downstream tasks (Liu et al. 2019). We hope to explore the use of RoBERTa as our initial model to investigate the applicability of 'doubt' identification on Stack Overflow. We will present the findings in detail in the relevant sections of this paper.

Our paper contributes to both technical and business perspectives in two distinct ways. From a technical standpoint, our proposed method aims to establish connections between comments and answers on Stack Overflow, capturing doubts that may arise over time. We utilize machine learning models and present results that are specifically tailored for generalized datasets like those found on Stack Overflow, where the user population is diverse and non-homogeneous, leading to variations in writing styles. From a business

¹ <https://survey.stackoverflow.co/2022/#community>

perspective, our additional indicators on Stack Overflow offer convenient means for programming learners to identify high-quality answers and resolutions to their programming queries. Additionally, these indicators assist expert users in finding answers that require further assistance. We believe that by facilitating the discovery of quality answers, our approach encourages novice programmers by enabling them to achieve small successes in their learning journey. Overall, our contribution extends to both the technical realm by proposing an effective method and the business domain by providing valuable indicators that enhance the Stack Overflow experience for learners and expert users alike.

The remaining sections of this paper are organized as follows: a literature review is presented to provide an overview of current works related to our research topic, followed by a description of the research approach that involves data collection and model training and evaluation. We then discuss the potential application of the doubt indicator in relevant contexts. Finally, we address potential threats to validity and conclude the paper.

Literature Review

Lo et al. (2021) proposed the hybrid Sentic Pattern Detection (SPD) and Machine-Learning (ML) model for doubt identification on students' informal reflections. The study suggests that the proposed hybrid approach has the potential to generalise on all types of data as it does not require any domain knowledge and coupling it with ML models is a promising option to achieve high accuracy in doubt identification. To the best of our knowledge, the paper is the only research on doubt identification. The paper was using Logistic Regression and Dense Neural Networks (DNN) as ML classifiers. We apply SPD and explore a newer pre-trained transformer, i.e., RoBERTa (Liu et al. 2019) in this study. We will compare the performance and select the better-performing model for further discussion and analysis.

Ren et al. (2019) proposed an Open Information Extraction method to extract controversial discussions (an answer being criticised and the critique posts) in Java/Android-tagged question threads on Stack Overflow. They identified controversy in a sentence by calculating the similarity score between the sentence and their sample critique sentences gathered from Stack Overflow java/android-tagged threads. However, our work shall attempt to identify doubts in Stack Overflow comments using the ML approach which does not rely on domain-specific knowledge. Zhang et al. (2021) proposed a random forest classifier to identify informative Stack Overflow comments. The work focuses on decoupling each individual comment from the main thread of the Stack Overflow post. In contrast, our method aims to correlate the presence of doubts within individual comments to account for technical changes that may occur in the thread after an answer has been posted. The focus of this work is to improve Stack Overflow's comment ranking system within a thread so that informative comments can get more attention instead of being hidden away.

There is plenty of related work regarding predicting question/answer quality metrics on CQA websites. For example, Anderson et al. (2012) proposed to predict the long-term value (i.e., page views) of a Stack Overflow question and its answers using a set of features describing various properties of the answering process. Tian et al. (2013) designed several features measuring important aspects of a Stack Overflow answer, which were then learnt by a classifier to predict whether the answer would become accepted. Arai and Handayani (2013) presented a classifier to predict accepted answers using only non-textual features. Yao et al. (2015) proposed a family of algorithms to jointly detect high-quality questions and answers on CQA websites soon after they are posted. Ye et al. (2021) proposed an answer quality evaluation model using a text feature filtering mechanism. These existing works mainly focus on predicting question/answer quality using the body of the question/answer and some other handcrafted features as the input. Our work is not to develop models to predict question/answer quality directly but to explore potential new features that may provide users with a new perspective in finding high-quality and helpful answers, particularly doubts raised in comments under the answer, and how they can correlate with the features commonly discussed in the literature and used as measures for answer quality.

There are also many interesting empirical studies on CQA websites. Zhang et al. (2019) investigated Stack Overflow users' discussions in comments and provided insights into the overall commenting dynamics. Yang et al. (2019) provided a systematic literature review on the key research issues in the areas of CQA expert recommendation. Differing from mentioned empirical studies, our work mines historical questions,

answers, and comments instead and aims to identify clarification required for the question instead of the dynamics.

We hope our proposed method introduces an additional metric that facilitates better decision-making for programmer learners in selecting helpful answers and also allows expert users to quickly locate doubts in comments to add additional resolutions to address the doubts.

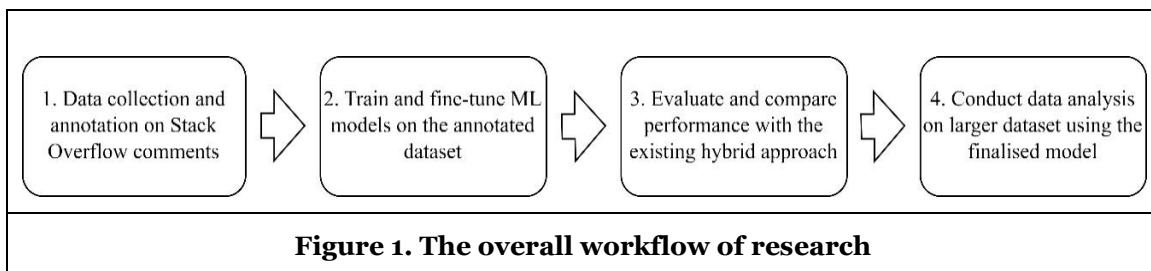
Definitions

A 'doubt' is a statement which potentially is a question or simply a sentence that requires clarifications on a given topic (Lo et al. 2019). In our context, we refer to doubt as a question or statement indicating differing opinions, system configurations, and versions. Some examples of 'doubtful' statements on Stack Overflow are as shown:

- I got an error with the above code. Is it because of python 3? However, `numpy.isnan(float('nan'))` did work. Why would I use `math` instead of `numpy`?
- this solution does have a problem that if `sort()` or other list functions are used, they work on the individual lists within this newly created list
- version 3 of this answer was correct and well formatted. this one (now 7) is wrong again. rolled back as "dont want your edit" while the edits improved the answer.
- Which version of Python was the original answer given in? Just typing `datetime.datetime.now()` in my Python 2.7 interactive console (IronPython hasn't updated yet) gives me the same behavior as the newer example using `print()` in the answer. I haven't successfully replicated what the original answer shows (`datetime.datetime(2009, 1, 6, 15, 8, 24, 78915)`). (Not that I really want to, the `print()` behavior is preferred, but I am curious.)
- The first part "For TensorFlow 2.0 and 2.1..." is not accurate. It's not in the documentation source referenced and I have TF2.0 and when I tested it I got an error. The second part though works on TF2.0 as well as TF2.2+

Research Approach

The research workflow, as illustrated in Figure 1, involved several steps. We initiated the study by collecting and annotating a dataset of Stack Overflow comments. This annotated dataset was utilized to train and fine-tune machine learning models. Subsequently, the performance of our models was evaluated and compared to an existing approach. Finally, we conducted data analysis on a larger dataset using the chosen model, enabling us to extract insights and draw meaningful conclusions. In the subsequent section, we delve into the specifics of the data collection and annotation processes, elaborate on the details of model training and selection, and provide an evaluation of the selected model.



Data Collection

Before implementing data collection, it is necessary to understand the data we are collecting. A Stack Overflow question thread consists of the question post itself and a list of answer posts under the question. Figure 2 shows an example of a Stack Overflow question post². Under the question title, there are indicators such as the creation date, last edit date, and view count of the question. The number to the left of the question body is the voting score of the question, i.e., 207 in Figure 2, which sums up all upvotes and downvotes received by the question, where each upvote/downvote represents +1/-1 on the score respectively. Below the question body, there can be a list of tags for the question, which are keywords that describe the question's topic.

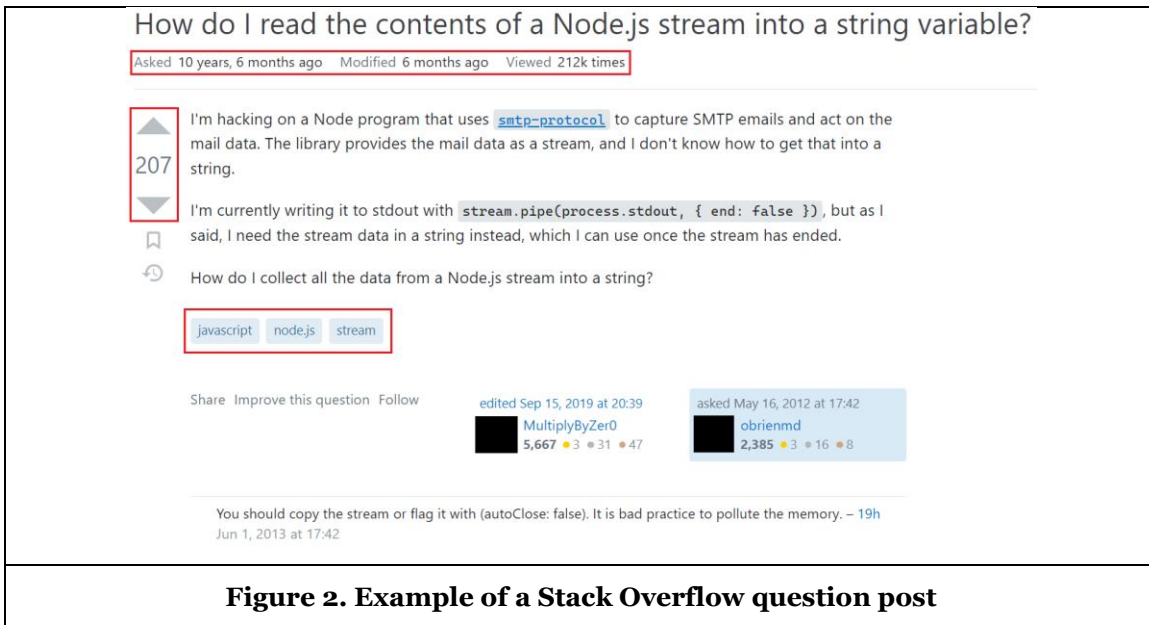


Figure 2. Example of a Stack Overflow question post

Figure 3 shows an example of a Stack Overflow answer post. The number to the left of the answer body is the voting score of the answer, similar to the question's voting score. Under each answer, there can be a list of comments posted by users. Comments can also receive upvotes from users, which are shown on the left of the comment text. The questioner can also mark an answer as accepted, which will be shown as a green tick under the answer's voting score. An accepted answer will be shown at the top of the answer list by default. Users can also gain reputation points by actively participating on the platform, i.e., asking and answering a question, voting, and commenting on any post etc. The answerer's total reputation point is shown as a grey-bolded number under the username, i.e., 357 in Figure 3.

² <https://stackoverflow.com/questions/10623798>



Figure 3. Example of a Stack Overflow answer post

The Stack Overflow comment data are collected from Stack Exchange Data Explorer³. With this open-sourced tool, users can compose and run queries on the public data from the Stack Exchange network. Each row selected from the query represents a single comment under a Stack Overflow answer post. All rows are grouped based on the answer to which the comments belong. We will refer to each of these groups as a stack, which consists of the rows of all comments under a specific answer post. Within each Stack, the rows are sorted based on the comments' creation date time, so we can easily visualise how the discussion took place in real-time, which also helps with annotation later. As Stack Exchange Data Explorer limits the maximum number of returned rows to 50,000. Therefore, separate queries were made to retrieve data with answers' creation dates ranging from January 2020 to August 2022. All data were retrieved on 23 August 2022.

We also derived several criteria when selecting Stack Overflow comments after some preliminary data analyses. The criteria are as follows:

1. Comments must belong to an answer post on Stack Overflow, as we are specifically interested in users' doubts about the answers rather than questions or other types of posts, and the commenting dynamics under questions or answers are likely significantly different.
2. The answer must have at least 10 comments under it. This is to reduce random noises as it is less likely that a meaningful discussion has occurred if there are not enough comments.
3. The question being answered must have at least 1,000 views by the time we retrieve the data. This is to filter out niche questions that not many users care about, which are prevalent on Stack Overflow. For example, many users tend to ask niche questions only to get a quick fix on some rare issues they have in the codes, which provide little value to others. The comment section for the answer is likely to become a live debugging or tutorial session, and we do not want to dump our models with such discussions.

³ <https://data.stackexchange.com/>

Data Annotation

After collecting the comment data, we built our dataset from comments under answers that were created between 2021 and 2022. The dataset contained 10,000 comments and was sent to a third-party TicTag for tagging. Initially, we decided on three classes for every comment in the dataset, 'Doubt', 'No Doubt', and 'Answer'. We used the following criteria to define whether the comment contains doubt:

1. If the comment is a question to the answer's author or another commenter. We only consider a question as doubtful if it is asking for clarification or explanation on a technical topic. For example, in Figure 3, the first comment is doubtful about the answerer's reasoning for the code and the third comment asks for more explanation from another commenter. Therefore, other general questions will be considered as no doubt, e.g., 'what version of python did you use', 'what is the error message', 'can you show me the trace stack' etc.
2. If the comment highlights scenarios that the solution may not work.
3. If the comment mentions getting an error/exception after they tried some suggested codes.
4. If the comment suggests that the answer is obsolete due to version change etc.

For a comment without doubt, it was tagged as 'Answer' if the comment tried to resolve a previously raised doubt; otherwise, it was tagged as 'No Doubt'. During the tagging process, TicTag found that some comments should be tagged as 'answer' but also raised a new doubt in the same comment. Therefore, these comments were put into a new class, 'Both Doubt and Answer'.

The dataset of 10,000 comments went through two rounds of tagging and one round of quality check afterwards. In each tagging round, the comments were assigned to two different taggers who independently labelled them. Subsequently, a round of quality check (QC) was conducted to ensure accuracy at the individual comment level. During this QC process, discrepancies arising from the two rounds of tagging were carefully reviewed and addressed. The QC phase accounted for approximately 20-30% of the data. The quality check was performed collaboratively by the team, involving a collective review and discussion of the cases to ensure consistent and reliable tagging. The taggers were between the age of 18 to 34, and most of them had IT/technical backgrounds. Table 1 shows the number of comments for each class from the finalised dataset.

	Answer	Doubt	No Doubt	Both Doubt and Answer
Number of Comments	4031	2722	2364	883

Table 1. Number of comments for all four classes

Model Training and Evaluation

Sentic Pattern Detection (SPD) Approach

Lo et al. (2021) proposed a hybrid approach for doubt identification which uses a doubt sentic pattern detection (SPD) algorithm and an ML model. Their work included a doubt sentic pattern list extracted from students' feedback on a university course. In this study, only the SPD algorithm is adopted, the ML model from the hybrid approach is not used. The purpose is to assess if using a knowledge-based approach is sufficient for doubt identification in Stack Overflow comments.

The Doubt SPD algorithm is a knowledge-based approach with a dependency relation of common-sense reasoning found in natural language. Examples of doubt sentic patterns are 'does—mean', 'what—differ', where '—' is an arbitrary number of words with a maximum of three words. For example, 'if the assumption has been violated, does that mean the model is useless', 'how similar it is to clustering and what is the different or they are the same'. For our purpose, we investigated each of the doubt sentic patterns separately and added new common doubt sentic patterns observed from the Stack Overflow comments to identify doubts within comments. For example, 'not work' and 'no idea' are common doubt sentic patterns after trying out the answer but did not solve their programming issues.

The results from our SPD model are reported in Table 2.

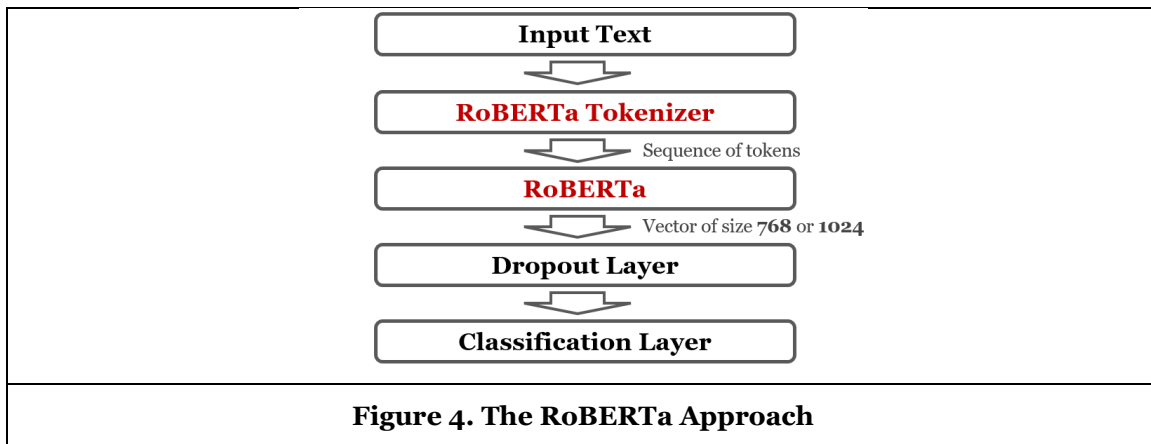
	precision	recall	f1-score	size
Doubt	0.35	0.20	0.26	537
No Doubt	0.72	0.84	0.77	1287
macro avg	0.53	0.52	0.51	1824
weighted avg	0.61	0.65	0.62	1824

Table 2. Classification report of the model using the SPD algorithm

Precision is important in giving our users confidence in using our 'doubt' tag while recall is important in retrieving comments with doubts and identifying quality answers. The F1 score keeps the balance between precision and recall and hence we will be using these metrics to evaluate our models.

RoBERTa-based Machine Learning Approach

We also built a model based on RoBERTa's pre-trained model, which is a robustly optimised pretraining procedure that improved on Bidirectional Encoder Representations from the Transformers (BERT) model (Liu et al. 2019). Two different variations of RoBERTa were tried, 'roberta-base' and 'roberta-large'. The overall architecture of the ML model is shown in Figure 4, where the input text is passed to the RoBERTa tokeniser to generate a sequence of tokens. The sequence is then passed to the RoBERTa model itself which outputs a vector of size 768 for Roberta-base and 1024 for Roberta-large. The output vector will then go through a dropout layer and finally, a classification layer to generate the probability score for each class. The RoBERTa pre-trained model was also used as a word embedding to both Logistic Regression and DNN with one hidden layer as the classifiers, to assess if other classifiers may achieve better results. The approach proposed in Figure 4 has consistently outperformed SPD and thus this configuration is used throughout the paper.



Upon examining the dataset, we decided to exclude the 'Both Doubt and Answer' class as it only has 883 records out of 10,000, and it is a mixture of two other classes, making it difficult for the model to learn and distinguish. The dataset is then split into 70-10-20% for training, validation, and test purposes respectively. The validation set was used to fine-tune the hyperparameters, specifically max learning rate={1e-3, 1e-4, 1e-5, 1e-6}; max sequence length={200, 256, 512}; hidden layer size={16, 32, 64, 128}; dropout probability={0, 0.05, 0.1, 0.2}.

The models were trained using Smith's (2017) one-cycle policy, which utilised cyclical learning rates for training models significantly faster and achieving higher accuracy. Using this policy, most models converged under 10 epochs, from which an early stopping mechanism with the maximum validation

accuracy was used to select the best-performing model. We found that the max learning rate of $1e-5$ usually gives the best performance using the fewest epochs, while max sequence length and dropout probability didn't show a significant impact on the performance. Among the different sizes of the hidden layer, 32 performed slightly better than the rest. The 'roberta-base' variation performed slightly better than 'roberta-large', although 'roberta-large' is more complex and takes significantly longer to train. We also tried to implement additional pre-processing on the comment text, which includes replacing codes and URLs with a placeholder, because these rare words are not easily understood by the model, but it turned out that these additional pre-processing steps did not result in any significant improvement on the models' performance. Table 3 summarises the performance metrics of the finalised model on the test set.

	Three Classes				Two Classes			
	precision	recall	f1-score	size	precision	recall	f1-score	size
Answer	0.70	0.75	0.72	806				
Doubt	0.64	0.79	0.71	537	0.68	0.72	0.70	537
No Doubt	0.59	0.37	0.46	481	0.88	0.86	0.87	1287
macro avg	0.64	0.64	0.63	1824	0.78	0.79	0.78	1824
weighted avg	0.65	0.66	0.65	1824	0.82	0.82	0.82	1824
Table 3. Classification report of the models with three classes (left) or two classes (right) using the RoBERTa model								

As seen in Table 3 (left), the model achieved a decent weighted average F1 score of 0.65. The model also showed significantly worse performance, especially the recall, for the 'No Doubt' class. The main reason for this is that the 'Answer' and 'No Doubt' classes can be very similar, making it difficult for the model to distinguish. The number of 'Answer' records in the dataset was also nearly twice the number of 'No Doubt'. Therefore, the model learned to put a higher probability for 'Answer' while missing out on more 'No Doubt' records, resulting in a worse recall on the 'No Doubt' class.

With the above result, we decided to reduce the number of classes further and see how our model would perform. With all 'Answer' records re-labelled as 'No Doubt', new models are trained using only binary classes, i.e., 'Doubt' or 'No Doubt'. The summarised performance metrics of the finalised model on the test set are shown in Table 3 (right).

The model showed significantly better weighted average F1 score of 0.82 with only two classes. The AUC score of the model was 0.86, which was great given the amount of data we were training on. However, it still had a slightly worse performance on the 'Doubt' class than the 'No Doubt' class, mainly due to the imbalance of the two classes in the dataset.

Our system was able to train on the training set of 6,382 instances in an average of 2 minutes and 24 seconds for one epoch. We utilized an NVIDIA GeForce RTX 3080 with 16GB of graphical memory for training. After training for 10 epochs, our best-performing model achieved its highest validation accuracy in the 5th epoch. In terms of inference, our best model took 27 seconds to generate prediction results for the test set of 1,824 instances. These results indicate that our model has demonstrated excellent efficiency in both training and inference.

Model Selection

Based on our results in Tables 2 and 3, we found that the SPD was not able to accurately predict doubtful statements. In terms of precision, recall and F1-Score, the SPD appears to favour the comments without a doubt. Since doubt identification is the key to identifying quality answers, we evaluated that the RoBERTa model with two classes performed the best.

We seek to understand the reasons why the SPD model did not perform as well. One reason is that Stack Overflow comments are much more varied in terms of topic and style of language than student feedback on

how they express doubt. For example, it is very common for 'Doubt' comments to suggest that something is not working, hence we checked for the sentic pattern 'not work' and its equivalent, but this pattern turned out to be prevalent in 'No Doubt' comments as well, as some users may add explanations and fixes after they mentioned that something did not work. We believe that more feature engineering and text pre-processing need to be done to continue exploring sentic patterns. For instance, a more sophisticated approach is required to capture the discourse structure for sentic patterns to work on a rich knowledge base like the Stack Overflow comments (Poría et al. 2014).

For our study, we find the performance of the model using RoBERTa with two (binary) classes satisfactory. For subsequent analyses, we will be analysing based on the model using RoBERTa with binary classes.

Result Analyses and Discussions

Research Question 1

RQ1. *Can machine learning models identify doubts automatically in Stack Overflow comments?*

Based on the results in Table 3, we can see that the RoBERTa-based ML model is able to achieve high accuracy in identifying doubt in Stack Overflow comments, with an overall accuracy of 0.82 using two classes, 'Doubt' and 'No Doubt'. The performance of the 'roberta-base' variation performed slightly better than 'roberta-large' on our dataset. However, additional pre-processing, such as replacing code snippets and URLs with placeholders, does not result in significant improvement in the model's performance.

Research Question 2

RQ2. *What are the relationships between the presence of doubts among comments under the answer post and Stack Overflow's existing voting and reputation metrics?*

To answer RQ2, we performed further analyses on a new dataset using our selected model, i.e., the RoBERTa model with binary classes. The Stack Overflow comments were selected using the same criteria as stated in the 'Data Selection' Section, but the answers' creation date ranged from 2016 to 2020. All data were retrieved on 1 November 2022, including users' reputation data, which will be used for analysis later. The dataset consisted of 354,471 comments in total. Our model selected is the RoBERTa model with binary classes. We hope our findings from our RQ2 will serve as guiding principles for the design of inclusion of 'doubt' indicators.

Correlation of Doubts with User Reputation

We seek to explore the relationships between users' doubt comments and their reputations. The user reputation on Stack Overflow is highly skewed, where most of the users have less than 10,000 reputation points, while some users can reach more than 1,000,000. Therefore, using a conventional linear scale did not reveal much information. We decided to use a log-10 scale on user reputation and found that it follows a normal-like distribution, as shown in Figure 5, which is a log-scaled histogram of user reputation with 100 bins.

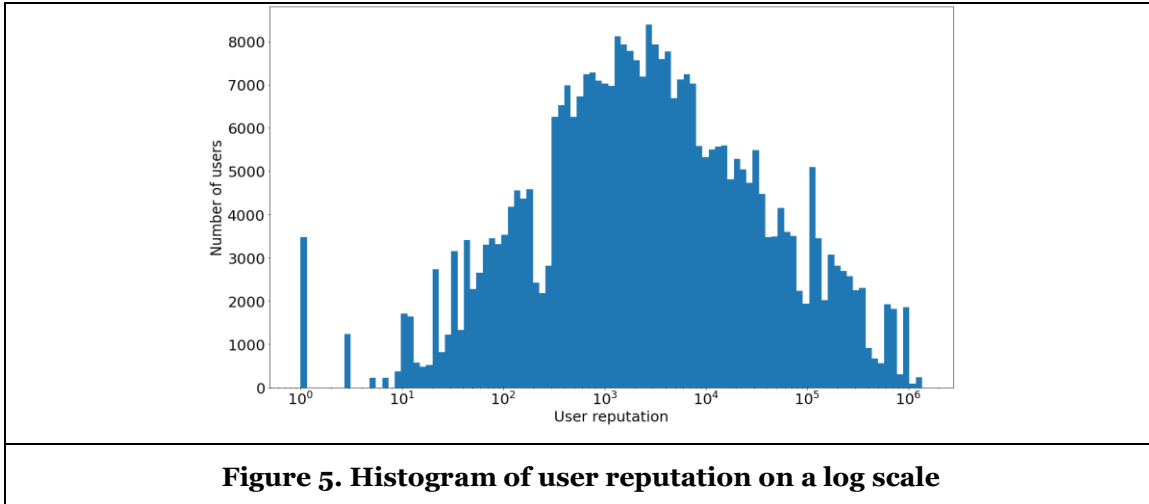
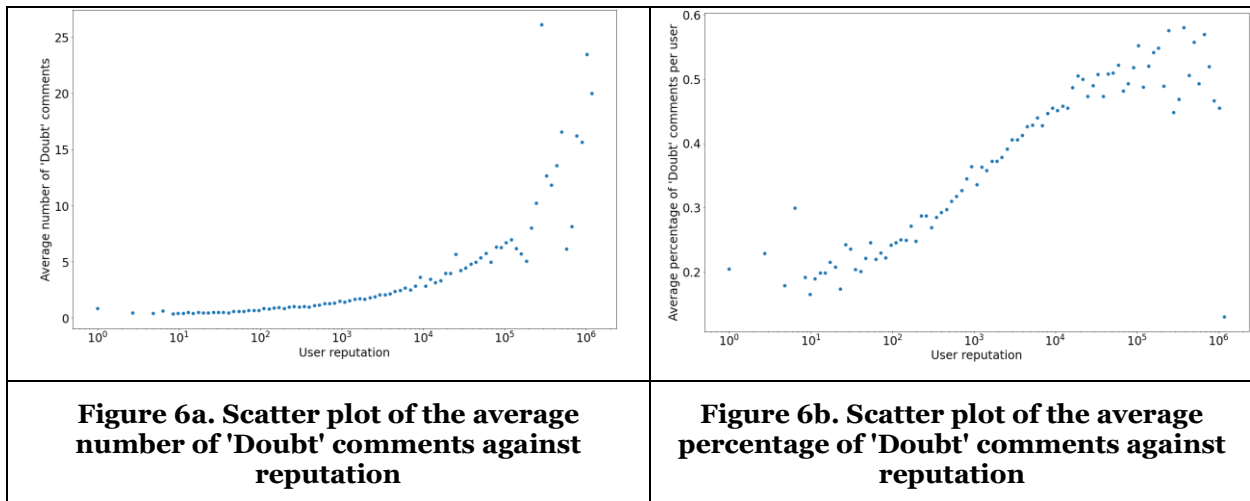


Figure 6a shows the average number of 'Doubt' comments posted by the users for each reputation range, using the same scale and bins as Figure 5. The plot shows an exponential-like trend, suggesting that highly reputed users raise significantly more doubts in their comments than less reputed users.



We also calculated the percentage of 'Doubt' comments out of all comments made by each user and took the average for all users in each reputation range. Figure 6b shows this average percentage against user reputation. The graph shows that the average percentage of 'Doubt' comments has a highly notable linear correlation with user reputation on a log-10 scale, with a Pearson's correlation coefficient of 0.878. This suggests that on average, users with higher reputations have a higher tendency to raise doubts in their comments.

Our findings from Figures 6a and 6b are counterintuitive as we may expect that inexperienced programmers, who are more likely to be less reputed users, tend to raise more doubts in their comments. One explanation for this observation could be that highly reputed users are very likely to be experienced and knowledgeable in the areas they comment on, so they tend to be more critical of the answer or others' comments, and raise doubt to correct, improve or extend the answer (Zhang et al. 2019). We recommend paying more attention to 'doubtful' comments from highly reputed users as they tend to provide additional value to the answer. We proposed that we could have a doubt indicator (along with a user reputation indicator) on each comment to draw the attention of programmer learners.

Position of Doubtful Comments

We were particularly interested in the position of doubtful comments and hence we ran a series of experiments to investigate. Figure 7 shows the distribution of the position where the first doubt is raised in an answer's comment section if there exists a doubtful comment. Position 1 means the first comment contains doubt, 2 means a doubt is first raised in the second comment and so on.

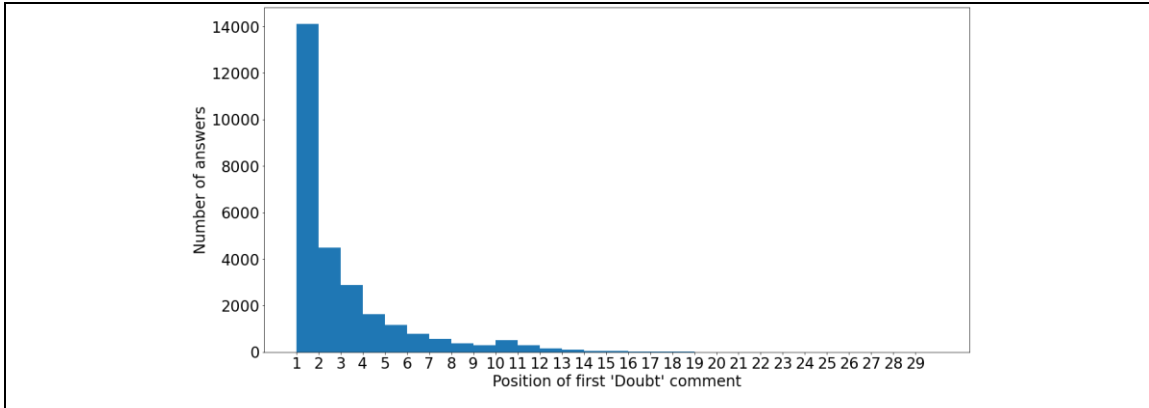


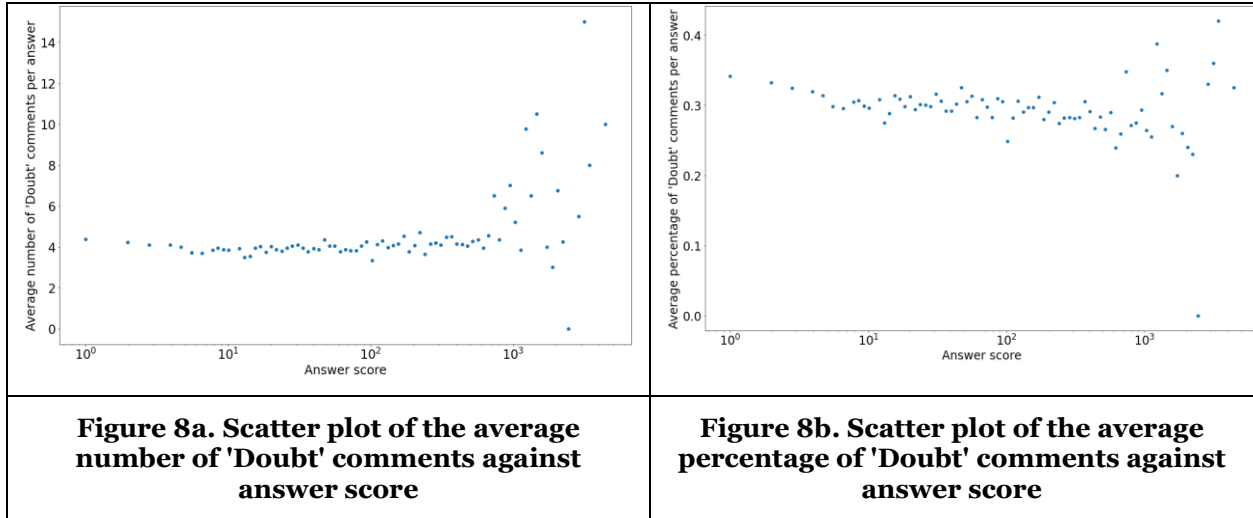
Figure 7. Histogram of the position of the first doubtful comment

As seen from Figure 7, for most of the comment sections below their respective answers, the first comment contains doubt. Doubts raised at the start of a comment section will get more attention from users as early comments will be seen by users first. Therefore, users such as programming learners can extract crucial information without going through the entire comment section by reading the initial few comments to get more comprehensive solutions to the answers.

On the other hand, we also found that among the 27,534 answers' comment sections, 4,748 (or 17.2%) of them contain doubt in the latest comment. Since we used answers that were created at least two years ago, we think that there should have sufficient time to provide resolutions by the community. When we did the same analysis on the first batch of data, where the answers were created in 2021 or 2022, 24.7% of the answers' comment section contained doubt in the last comment. This observation indicated that there are unresolved doubts raised in the latest comments which went unnoticed by the community. As part of our aim to help programming learners find quality answers that could attest to the time (perhaps due to version or environment changes), it is important to provide a mechanism to conveniently locate answers with outstanding doubt within an answer.

Correlation of Doubts with Answer Score

We would also like to find out how doubts in the comment section correlate with the answers' voting score, simply referred to as the answer score. Figure 8a shows the average number of 'Doubt' comments under an answer against the answer score using a log-10 scale with 100 bins, as the answer score distribution is also very right-skewed. From Figure 8a, answers across different answer score intervals all have a similar number of doubtful comments, while only the highly scored answers show significantly more fluctuations, likely due to insufficient samples in that range, hence the large variance.





The percentage of 'Doubt' comments out of all comments under each answer is then calculated and the average percentage for all answers in each answer score interval is taken, shown in Figure 8b. As seen from the figure, highly scored answers have slightly less percentage of doubtful comments as compared to answers with lower scores. However, the linear correlation is insignificant as Pearson's correlation coefficient is only -0.236.

Our findings suggest no significant linear correlation between the answer score and the presence of doubt in comments. This finding means that if given doubt indicators in comments, they can be treated independently from the answer score. This also means that vice versa, answers with high scores have the same likelihood of containing doubts in the comment section as answers with low scores.

Potential Application of Doubt Indicator

We envision implementing the doubt indicators in Stack Overflow, as demonstrated in our example in Figure 9. We can include 'Doubt' indicators in the Stack Overflow post to assist the user in identifying comments with doubts. Some of the possible design principles are as follows:

1. Comments with doubt are shown with a pink background.
Since there is no relationship between the answer score and doubtful comments, we propose a background colour change (in pink) to indicate doubtful comments.
2. Comments that are not shown but contain doubts are indicated with an icon .
Since not all comments are shown to the user, we propose an expert-needed indicator to guide experts to the doubtful comment.
3. Comments with doubt by a reputed user are indicated with an icon .
This indicator can help draw programmer learners' attention to comments made by reputed users.

We acknowledge that a comprehensive user experience (UX) evaluation should be conducted as part of future work.

Use the `+` operator to combine the lists:

```
listone = [1, 2, 3]
listtwo = [4, 5, 6]

joinedlist = listone + listtwo
```

Output:

```
>>> joinedlist
[1, 2, 3, 4, 5, 6]
```

Share Improve this answer Follow

edited Jun 6, 2022 at 2:02 answered Nov 12, 2009 at 7:07

Mateen Ulhaq 23.3k ●16 ●89 ●132 Daniel G 65.9k ●7 ●42 ●42

5262

Doubt submitted by a reputed user

157 does this create a deep copy of listone and appends listtwo? – Daniel F Apr 19, 2012 at 12:34

205 @Daniel it will create a new list with a shallow copy of the items in the first list, followed by a shallow copy of the items in the second list. Use `copy.deepcopy` to get deep copies of lists. – Daniel G Apr 19, 2012 at 14:51

299 another useful detail here: `listone += listtwo` results in `listone == [1, 2, 3, 4, 5, 6]` – ricknagy Jan 29, 2014 at 16:14

21 @br1ckb0t will that change what listone is pointing at? So: `list3 = listone` `listone+=listtwo` Is list3 changed as well? – MikeH Feb 19, 2014 at 5:01

8 @Pygmalion That is not Python3 specific, but specific to how NumPy arrays handle operators. See the answer by J.F. Sebastian in the answer by Robert Rossney for concatenating NumPy arrays. – 153957 Apr 16, 2015 at 11:42

Show 8 more comments

Comments with doubt in pink background

More doubts in the rest of the comments for experts to address

Figure 9. A possible design of 'Doubt' indicators on Stack Overflow

Threats to Validity

During data collection, the comments were intentionally selected using the abovementioned criteria, potentially threatening external validity as the model trained on the resultant dataset may not generalise well on all Stack Overflow comments. We selected the data to reduce random noises by filtering out discussions with less value for our model to learn. Another reason is that we did not have enough resources and time to reliably tag a larger dataset of more than 10,000 comments.

Our proposed design of "Doubt" indicators on Stack Overflow remains to be validated. Although our proposed designs are based on our dataset evaluation, we have not experimented with this design with the actual users. We intend to evaluate this design with novice learners in programming in our future work.

Conclusion

In this research, we proposed an approach to automate doubt identification on Stack Overflow comments. Our results show that a pre-trained machine learning model using the RoBERTa algorithm can provide good precision, recall and F1-scores to classify comments into binary classes indicating the presence of 'doubt'. With our model and further experiment with an additional dataset, we found interesting correlations between 'doubts' and the existing voting and reputation scores. Our results showed that users with higher reputation scores tend to raise more doubt than less reputed users; most Stack Overflow answers have doubt raised in the first few comments, and a significant portion of answers has unsolved doubt in the last comment. Using these correlations as guiding principles, we designed and proposed additional metrics on Stack Overflow to guide programming learners to find quality answers; and to help experts identify posts whose answers may need improvements or revision. We hope our proposed metrics help learners gain small successes in their learning journeys.

For future work, we will embark on a validation project to sharpen our design of the doubt indicators on Stack Overflow and collect empirical results on the usefulness of the indicators to novice programmers in solving their programming issues. We can also evaluate other large language models and evaluate their respective performances in identifying doubts on Stack Overflow comments.

References and Citations

- Anderson, A., Huttenlocher, D., Kleinberg, J., & Leskovec, J. (2012, August). Discovering value from community activity on focused question answering sites: a case study of stack overflow. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 850-858).
- Arai, K., & Handayani, A. N. (2013). Predicting quality of answer in collaborative Q/A community. *International journal of advanced research in artificial intelligence*, 2(3), 21-25.
- Bhasin, T., Murray A. and Storey M. (2021). Student Experiences with GitHub and Stack Overflow: An Exploratory Study, IEEE/ACM 13th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), pp. 81-90, Madrid, Spain.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimised bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Lo, S. L., Tan, K. W., & Ouh, E. L. (2021). Automated doubt identification from informal reflections through hybrid sentic patterns and machine learning approach. *Research and Practice in Technology Enhanced Learning*, 16(1), 1-24.
- Lo, S. L., Tan, K. W., & Ouh, E. L. (2019). Do my students understand? Automated identification of doubts from informal reflections. *Proceedings of The 27th International Conference on Computers in Education (ICCE)*, Kenting, Taiwan.
- Lu, Y., Mao, X., and Zhou, M., and Zhang, Y., and Li, Z., Wang, T., and Yin G., and Wang, H. (2022). Motivation Under Gamification: An Empirical Study of Developers' Motivations and Contributions in Stack Overflow, in *IEEE Transactions on Software Engineering*, vol. 48, no. 12, pp. 4947-4963.
- Mazloomzadeh, I., Udin, G., Khomh, F., & Sami, A. (2021). Reputation gaming in stack overflow. *arXiv preprint arXiv:2111.07101*.
- Poria, S., Cambria, E., Winterstein, G., & Huang, G. B. (2014). Sentic patterns: Dependency-based rules for concept-level sentiment analysis. *Knowledge-Based Systems*, 69, 45-63.
- Ren, X., Xing, Z., Xia, X., Li, G., & Sun, J. (2019, November). Discovering, explaining and summarising controversial discussions in community q&a sites. In 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE) (pp. 151-162). IEEE.
- Smith, L. N. (2017, March). Cyclical learning rates for training neural networks. In 2017 IEEE winter conference on applications of computer vision (WACV) (pp. 464-472). IEEE.
- Tian, Q., Zhang, P., & Li, B. (2013). Towards predicting the best answers in community-based question-answering services. In *Proceedings of the International AAAI Conference on Web and Social Media (Vol. 7, No. 1, pp. 725-728)*.

- Wang, S., German, D. M., Chen, T. H., Tian, Y., & Hassan, A. E. (2021, September). Is reputation on Stack Overflow always a good indicator for users' expertise? No!. In 2021 IEEE International Conference on Software Maintenance and Evolution (ICSME) (pp. 614-618). IEEE.
- Yang, Z., Liu, Q., Sun, B., & Zhao, X. (2019). Expert recommendation in community question answering: a review and future direction. *International Journal of Crowd Science*.
- Yao, Y., Tong, H., Xie, T., Akoglu, L., Xu, F., & Lu, J. (2015). Detecting high-quality posts in community question answering sites. *Information Sciences*, 302, 70-82.
- Ye, W., Chen, Y., Ma, G. F., & Yang, X. H. (2021, April). Answer Quality Evaluation of Social Q&A Platforms Based on Feature Filtering. In 2021 IEEE 6th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA) (pp. 245-249). IEEE.
- Zhang, H., Wang, S., Chen, T. H., & Hassan, A. E. (2019). Reading answers on stack overflow: Not enough!. *IEEE Transactions on Software Engineering*, 47(11), 2520-2533.
- Zhang, H., Wang, S., Chen, T. H., & Hassan, A. E. (2021). Are comments on Stack Overflow well organised for easy retrieval by developers?. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 30(2), 1-31.