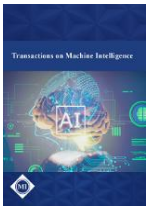
Transactions on Machine Intelligence 4(4) (2021) 238-244

ISSN Online: 2821-1693

Transactions on Machine Intelligence

Journal Homepage: https://tmachineintelligence.ir/index.php/TMI



Finding the Potential Accepted Answer on Stack Overflow: A Text Mining Approach

M. Jamshidiyan Tehrani [1,*], P. Arjomand [2], S. Haghighat [2]

[1] Faculty of Informatics, Università della Svizzera Italiana, Lugano, Switzerland
[2] Department of Computer Engineering, Salman Farsi University of Kazerun, Taleghani, Kazerun, 73175-457, Fars, Iran

ARTICLE INFO

Article History:
Received 17 June 2021
Received in revised form 28 August 2021
Accepted 23 December 2021
Available online 25 December 2021

Keywords:
Stack Overflow, Data Mining, Text Mining, Machine-Learning, Sentiment Analysis

ABSTRACT

Stack Overflow serves as a widely-used, community-driven platform where developers seek assistance with programming-related issues. While the platform allows users to post questions and receive multiple answers, a significant portion of these questions do not culminate in an accepted solution. This lack of a clearly identified best answer often results in confusion for both the original poster and future visitors, as well as increased time spent navigating through numerous responses. To address this challenge, we present a method for automatically identifying the most promising answer among unaccepted ones. Our approach involves the application of text mining techniques to extract 13 informative features from a large dataset comprising 15,464 questions, 37,275 answers, and 72,025 comments. These features capture various textual, structural, and user-related aspects of the posts. The extracted data are then used to train machine learning models aimed at predicting the answer most likely to be accepted. The study focuses solely on English-language content available on Stack Overflow. The proposed method demonstrates promising performance, achieving an overall accuracy of 71% and an F1 score of 70%. These results suggest that automated answer recommendation can significantly enhance the user experience by reducing ambiguity and improving the efficiency of information retrieval on Q&A platforms.

1. INTRODUCTION

Q&A forums serve as platforms for users to exchange knowledge and swiftly resolve software-related issues. These platforms typically consist of three fundamental components: users, questions, and answers. Additional features such as answer scoring, tagging, question categorization, and user credibility are often incorporated to enhance functionality [1].

Stack Overflow has transformed from a basic Q&A site into a vast social community where individuals of varying expertise levels collaborate to address programming challenges [2]. It has profoundly influenced how programmers learn, communicate, and collectively develop content repositories for future reference [3-5]. Due to its extensive

* Corresponding Author: masoud.jamshidiyantehrani@usi.ch
Faculty of Informatics, Università della Svizzera italiana, Lugano, Switzerland.

http://dx.doi.org/10.47176/TMI.2021.238

use, it has become an integral part of the software development ecosystem, with developers increasingly relying on it for their daily programming needs. Moreover, users on other platforms, such as mailing lists and GitHub, actively encourage participants to refer to Stack Overflow posts for solutions [6].

The site's growing significance can be attributed to four key factors:

- Users can find multiple high-quality answers to questions on nearly every programming language, tool, framework, and software [7].
- If the desired information is not available, users can create a post themselves and receive answers promptly [8].
- Virtual rewards, such as reputation points and badges, incentivize users to contribute more [9].
- The rich interface enables users to display their expertise to potential recruiters [10].

These elements contribute to the site's transparency. Each user has a dedicated profile page that aggregates their contributions and achievements on the site.

When a user poses a question, all users, including the questioner, can submit answers. If an answer precisely addresses the questioner's issue, they can approve it by marking it as the accepted answer. Each accepted answer indicates that it directly resolved the questioner's problem, thereby assisting others with similar issues in finding solutions more efficiently. Some questions receive multiple proposed answers but lack an accepted answer [11]. In these cases, the questioner may not have found a satisfactory solution among the answers. Alternatively, the questioner might have forgotten to mark an answer as the accepted one or may not know how to do so. This situation can lead others with similar questions to lose trust in the existing answers and spend more time searching for a solution. Therefore, predicting the accepted answer can help resolve these issues and assist the questioner in identifying the best response to their query. Stack Overflow itself can also use this approach to suggest the most appropriate answer to viewers. Additionally, any automatic question-answering system [12] can utilize this method to enhance its effectiveness.

2. CONTRIBUTION

In this study, we introduce a method for identifying and predicting potential accepted answers on Stack Overflow by employing machine learning models and text-mining techniques. Our approach utilizes 13 features, a reduction from the 52 features used in state-of-the-art methods [13]. This simplification enhances processing speed and reduces computational costs without compromising performance.

The features are derived from basic text mining techniques applied to the English texts of questions, answers, and their associated comments on Stack Overflow. Among these features, we include sentiment analysis of user responses. These extracted features are then input into machine learning models for analysis.

By focusing on a streamlined set of features, our approach aims to efficiently predict potential accepted answers, thereby improving the user experience on Stack Overflow.

3. RELATED WORK

Researchers have increasingly leveraged text analysis and machine learning to enhance the quality and utility of question-and-answer (Q&A) platforms and online reviews. Yazdaninia, Mohamad, David Lo, and Ashkan Sami explored the dynamics of questions lacking accepted answers on such platforms. Their study focused on analyzing question characteristics to predict the likelihood of receiving an accepted response. The core objective of their work was to empower questioners to refine their inquiries, thereby increasing the probability of obtaining high-quality answers. They proposed that adjustments—such as phrasing questions more clearly, enhancing readability, or selecting relevant tags—could significantly improve outcomes. To support this, they developed an innovative online tool designed to evaluate questions and recommend specific enhancements, offering a practical solution for users aiming to optimize their queries.

In a complementary investigation, Diyanati, Ahmad, et al. ([14]) examined user expertise on Stack Overflow, introducing a novel approach termed "comment mining." This method assesses an individual’s expertise by assigning

scores based on the presence of positive and negative terms within comments on their questions and answers. A cumulative score is then calculated to reflect the user's proficiency. The findings indicate that this technique effectively gauges expertise to a satisfactory degree, providing valuable insights for selecting features that better represent user competence. This approach underscores the potential of sentiment-driven analysis in evaluating online contributions.

Similarly, Naghashzadeh, Mahshid, et al. ([11]) confirmed the prevalence of unanswered questions in Q&A forums, with a specific focus on MATLAB-related platforms. Their analysis revealed a notably high proportion of questions without accepted answers, particularly in Simulink and the three most widely used MATLAB toolboxes: image processing, signal processing, and computer vision. This high rate of unresolved queries highlights a critical challenge in technical forums and emphasizes the need for strategies to improve question quality and response rates.

In a different domain, Pan and Zhang ([15]) investigated how the comprehensiveness of online reviews influences their perceived helpfulness among customers. They measured comprehensiveness by counting the product attributes mentioned in reviews and analyzed its interplay with review valence (positive or negative sentiment). Employing a word-level bigram analysis, they extracted attributes from review texts and assessed their impact on helpfulness votes. Their results suggest that reviews discussing a greater number of attributes tend to garner more helpfulness votes, with this effect being amplified in negative reviews. This finding points to the moderating role of valence, where detailed critiques appear to resonate more strongly with readers seeking informative feedback.

Expanding the application of text analysis, Garner, Benjamin, et al. ([15]) utilized sentiment analysis to explore tourists' comments, shedding light on how memorable travel experiences are formed. Their study demonstrated the power of machine learning, specifically text mining, in uncovering insights related to consumer happiness and well-being. By analyzing sentiment in travel-related feedback, they illustrated how positive and negative emotions shape perceptions of memorable experiences, offering a deeper understanding of customer satisfaction in tourism.

Collectively, these studies underscore the transformative role of machine learning and text mining in dissecting user-generated content. Whether enhancing Q&A interactions or decoding consumer sentiments in reviews, these methodologies provide robust tools for improving platform functionality and user experience. The integration of such techniques not only refines the analysis of comments and reviews but also informs strategies to elevate engagement and satisfaction across diverse online contexts.

4. DATA PREPROCESSING

The dataset utilized in this study is a subset of the one employed in [13], albeit with a reduced sample size. It comprises 15,464 questions, 37,275 answers, and 72,025 comments. Initially, the raw data included numerous non-English characters, necessitating a preprocessing step to eliminate these elements and ensure analytical consistency. Following this, stopwords—frequently occurring words such as "a," "the," "is," and "are"—were removed. These terms, which typically offer minimal informational value, were excluded manually to prevent the unintended deletion of specific terms critical to our feature set. The nature of these preserved "special words" will be elaborated upon in the subsequent feature extraction section. Finally, all remaining non-English words, such as proper names, were also filtered out to refine the dataset further.

5. FEATURE EXTRACTION

Text mining techniques were employed to extract features for classifying answers, with each feature derived exclusively from questions and comments directly associated with a specific answer. A total of 13 features were extracted from the textual data, and their inclusion in our analysis is justified by their relevance to predicting accepted answers. These features, along with the rationale for their selection, are detailed below.

1. **Number of Answers**: This feature quantifies the total answers provided to a question. When a question receives only a single answer, that response is highly likely to be the accepted one, making this a critical predictor in our model.

2. **Word Count Metrics**: The next trio of features measures the word count in related questions, answers, and comments. The underlying assumption is that lengthier text increases the likelihood of an answer being accepted. For instance, a detailed question might elicit a comprehensive response, or an accepted answer could attract extended commentary as users elaborate on its correctness, thereby elevating its perceived value.

5. **Number of Comments**: This feature captures the count of comments linked to an answer, distinct from the word count of those comments. Separating these metrics acknowledges that some comments may be concise, consisting of brief sentences, yet their frequency could still signal an answer's significance.

6. **Answer Sentiment**: Sentiment analysis was applied to assess the emotional tone of answers, sourced from the Senti4SD tool ([16]). While most answers are expected to exhibit neutral sentiment, variations—positive or negative—can reflect the writer's confidence. A positive sentiment may indicate assurance, potentially increasing the answer's appeal to the questioner and its likelihood of acceptance.

7. **Presence of "Thanks"**: As a binary feature, this indicates whether the word "Thanks" or its variants (e.g., "thanks," "thank," "tanks," "tnx") appears in an answer's comments. The presence of such gratitude often signifies appreciation for a correct or helpful response, enhancing the probability that it is the accepted answer.

8. **Negative Maker Words Count**: This feature evaluates the frequency of negative maker words (e.g., "don't," "isn't," "wouldn't," etc.) within comments to gauge their sentiment. Comments containing these words might express dissatisfaction, such as an answer failing to address a user's needs, thereby reducing its likelihood of being accepted. Our manually curated list includes 38 variations of negative terms, such as "dont," "isnt," "couldnt," and "not." To account for comment length, the count is normalized by dividing it by the total word count of the answer's comments.

9. **Presence of "But"**: Building on the previous feature, this binary indicator assesses whether "But" appears in comments. Its presence can counteract the effect of negative maker words, potentially reversing a sentence's sentiment and thus influencing the perceived utility of the answer.

10. **Customized Sentiment Analysis**: Two additional sentiment-based features were developed for answers and comments, leveraging the Bing Liu Opinion Lexicon ([17-18]). For comments, we constructed a corpus from all comments across answers, calculating the frequency of positive (Ps(y)) and negative (Ng(y)) words by dividing their occurrences by the corpus's total word count. For each answer, we summed the Ps(y) values from its comments and subtracted the Ng(y) values; a positive result indicates a positive sentiment, a negative result suggests negativity, and zero denotes neutrality. This score was normalized by the word count of the answer's specific comments. For answers, a similar process was applied using all answers as the corpus, ensuring a tailored sentiment measure independent of comment length.

$$ps(y) = \frac{\text{word(positive)}}{\text{w(comments)}} \tag{1}$$

$$Ng(y) = \frac{\text{word(negetive)}}{\text{w(comments)}} \tag{2}$$

$$Sentiment = \frac{\sum Ps(y) - \sum Ng(y)}{\text{W(Answer's Comments)}} \tag{3}$$

12-13. Document Similarity: The similarity between an answer and its corresponding question, or between an answer and its associated comments, serves as an indicator of their relevance to one another. High relevance suggests that the answer is likely correct and valuable, particularly when the question-answer pairing is closely aligned, increasing the probability of the answer being accepted. To quantify this similarity, we utilized WordNet ([19-20]), a lexical database, and applied the Wu-Palmer similarity metric ([21]). For each word in the first document (e.g., the question), we computed its Wu-Palmer similarity score with every word in the second document (e.g., the answer) based on their first synonyms, summing these values across all word pairs. To mitigate the influence of document

length, the total similarity score was normalized by dividing it by the combined word count of both documents. This process was implemented to assess two distinct relationships: the similarity between questions and answers, and the similarity between answers and comments, providing a robust measure of contextual alignment in each case.

6. LABELING

Each answer that is included in the accepted answers' dataset is labeled 1, and the rest are labeled 0. The classifiers here are trying to predict the answer that is capable of being the accepted one.

7. CLASSIFICATION RESULTS

Machine learning models that presented us with the best results are KNN with K = 8, Entropybased Random Forest with 100 trees, AdaBoost with 100 ensembles, entropy-based Decision Tree, RBF kernel SVM, and Naive Bayes classifier. The results in table 1 show that ensemble methods have better accuracy with AdaBoost gaining 71% accuracy, 70% F1, 71% Recall, and 70% Precision.

Table 1. Classification Results

Method	Accuracy	F1	Recall	Precision
KNN	63%	61%	63%	62%
SVM	65%	64%	65%	64%
Naive Bayes	67%	66%	67%	66%
Decision Tree	62%	62%	62%	62%
AdaBoost	71%	70%	71%	70%
Random Forest	69%	69%	69%	69%

8. CONCLUSION AND FUTURE WORK

Our proposed method revealed that by leveraging only English text extracted from posts, an accuracy of 71% can be achieved through the application of machine learning ensemble models and text mining features. This finding underscores the potential of text mining techniques alone to substantially address the challenge of predicting accepted answers. However, many questions on Stack Overflow receive multiple responses, complicating the task of identifying the optimal or potentially accepted answer with greater precision using solely textual data. The incorporation of deep learning models, the expansion of the dataset, or the inclusion of additional Stack Overflow attributes—such as the author's reputation or upvote counts—could further enhance predictive accuracy beyond the current reliance on text. The critical role of an accepted answer in Stack Overflow posts highlights the significance of this issue, motivating continued research and refinement of our approach in future investigations.

Transparency Statement

The data supporting this study are available upon reasonable request to the corresponding author, subject to ethical and confidentiality considerations.

Acknowledgments

We would like to express our gratitude to all individuals who contributed to this project.

Declaration of Interest

The authors declare that they have no competing interests.

Funding

This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

REFERENCES

[1] Faisal, M. S., et al. (2019). Expert ranking techniques for online rated forums. Computers in Human Behavior, 100, 168–176. https://doi.org/10.1016/j.chb.2018.06.013

[2] Anderson, A., et al. (2012). 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). https://doi.org/10.1145/2339530.2339665

[3] Begel, A., et al. (2013). Social networking meets software development: Perspectives from GitHub, MSDN, Stack Exchange, and TopCoder. IEEE Software, 30(1), 52–66. https://doi.org/10.1109/MS.2013.13

[4] Singh, V., et al. (2009). Users of open source software—How do they get help? In Proceedings of the 42nd Hawaii International Conference on System Sciences (pp. 1–10). IEEE. https://doi.org/10.1109/HICSS.2009.259

[5] Storey, M.-A., et al. (2010). The impact of social media on software engineering practices and tools. In Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research (pp. 359–364). https://doi.org/10.1145/1882362.1882435

[6] Vasilescu, B., et al. (2014). How social Q&A sites are changing knowledge sharing in open source software communities. In Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (pp. 342–354). https://doi.org/10.1145/2531602.2531659

[7] Parnin, C., et al. (2012). Crowd documentation: Exploring the coverage and the dynamics of API discussions on Stack Overflow. Georgia Institute of Technology, Tech. Rep, 11.

[8] Mamykina, L., et al. (2011). Design lessons from the fastest Q&A site in the west. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 2857–2866). https://doi.org/10.1145/1978942.1979366

[9] Deterding, S., et al. (2011). Gamification: Using game-design elements in non-gaming contexts. In CHI'11 Extended Abstracts on Human Factors in Computing Systems (pp. 2425–2428). https://doi.org/10.1145/1979742.1979575

[10] Capiluppi, A., et al. (2012). Assessing technical candidates on the social web. IEEE Software, 30(1), 45–51. https://doi.org/10.1109/MS.2012.169

[11] Naghashzadeh, M., et al. (2021). How do users answer MATLAB questions on Q&A sites? A case study on Stack Overflow and MathWorks. In Proceedings of the 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER) (pp. 559–563). IEEE. https://doi.org/10.1109/SANER50967.2021.00059

[12] Pundge, A. M., et al. (2016). Question answering system, approaches and techniques: A review. International Journal of Computer Applications, 141(3), 1–8. https://doi.org/10.5120/ijca2016909587

[13] Yazdaninia, M., et al. (2021). Characterization and prediction of questions without accepted answers on Stack Overflow. In Proceedings of the 2021 IEEE/ACM 29th International Conference on Program Comprehension

(ICPC) (pp. 1–11). IEEE. https://doi.org/10.1109/ICPC52881.2021.00015

[14] Diyanati, A., et al. (2020). A proposed approach to determining expertise level of Stack Overflow programmers based on mining of user comments. Journal of Computer Languages, 61, 101000. https://doi.org/10.1016/j.cola.2020.101000

[15] Pan, Y., & Zhang, J. Q. (2011). Born unequal: A study of the helpfulness of user-generated product reviews. Journal of Retailing, 87(4), 598–612. https://doi.org/10.1016/j.jretai.2011.05.002

[16] Calefato, F., et al. (2018). Sentiment polarity detection for software development. In Proceedings of the 40th International Conference on Software Engineering (pp. 1–12). https://doi.org/10.1145/3180155.3182519

[17] Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 168–177). https://doi.org/10.1145/1014052.1014073

[18] Hu, M., & Liu, B. (2004). Mining opinion features in customer reviews. In Proceedings of the 19th National Conference on Artificial Intelligence (pp. 755–760). AAAI Press.

[19] Fellbaum, C. (1998). WordNet: An electronic lexical database. MIT Press. https://doi.org/10.7551/mitpress/7287.001.0001

[20] Miller, G. A. (1995). WordNet: A lexical database for English. Communications of the ACM, 38(11), 39–41. https://doi.org/10.1145/219717.219748

[21] Wu, Z., & Palmer, M. (1994). Verb semantics and lexical selection. In Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics (pp. 133–138). https://doi.org/10.3115/981732.981751