Identifying Textual Features of High-Quality Questions: An Empirical Study on Stack Overflow

Qing Mi[†], Yujin Gao[‡*], Jacky Keung[†], Yan Xiao[†], Solomon Mensah[†]
[†]Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong
[‡]School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
Email: {Qing.Mi, yanxiao6-c, smensah2-c}@my.cityu.edu.hk, paulgyj@bit.edu.cn, Jacky.Keung@cityu.edu.hk

***Abstract*—*Background:* Stack Overflow (SO) is a programming-specific Q&A website that serves as a valuable repository of software engineering knowledge. For SO members, formulating a good question is the first step towards eliciting satisfactory responses. *Aims:* To guide SO members on how to make a good question, we conduct an empirical study using the publicly available *Stack Overflow Data Dump* for the period of 2008-2016. *Method:* We first choose 25 features along 5 dimensions to represent the textual characteristics that we are interested in. Making use of the Boruta algorithm, we then capture all features that are either strongly or weakly relevant to the question quality. *Results:* The results show that the number of tags and code snippets are the most discriminative features, whereas there is only a weak correlation between the question quality and the sentiment-related factors. Based on the empirical evidence, we provide useful and usable suggestions to SO members on how to optimize their questions. *Conclusions:* We consider that our findings will provide SO members with a better understanding of the patterns behind high-quality questions, this is to support effective and efficient utilization of Q&A websites as the ultimate goal.**

***Index Terms*—Stack Overflow, Q&A website, textual feature, Boruta algorithm, empirical software engineering**

I. Introduction

Stack Overflow (SO) is one of the most popular question and answer (Q&A) websites, an important community for novices and experienced software developers to share their knowledge and advance their careers. Recently, an increasing amount of software engineering research has been dedicated to the study of SO data. Barua et al. [1] applied Latent Dirichlet Allocation (LDA) to discover the main discussion topics and trends from the SO posts. Zou et al. [2] presented an interrogative-guided re-ranking approach to refine search results based on the SO question-answer pairs. Yao et al. [3] observed a strong quality correlation between SO questions and the associated answers.

For software developers, knowing what makes a good question is the first step towards effective and efficient utilization of Q&A websites. Given limited understanding in this regard, we perform an exploratory analysis to identify the characteristics of good questions based on the publicly available *Stack Overflow Data Dump*[1]. Note that we focus solely on textual factors so as to make our empirical findings general enough to be applicable to other Q&A websites, and also because questioners have great control over them. Specifically, we make the following main contributions:

- An empirical study is performed using SO data for the period of 2008-2016 to characterize high-quality questions according to a set of selected features ranging from simple structural aspects to complex readability metrics.
- Based on the empirical findings, we provide a number of practical suggestions to guide SO members on how to optimize their questions, this is to maximize the utility of Q&A websites as the ultimate goal.

The rest of the paper is organized as follows. Section II describes the design of the empirical study and Section III provides the empirical results. In Section IV, we discuss the threats to validity. Section V presents the related work and Section VI concludes this paper.

II. Research Design

We begin by constructing the two comparison groups (i.e., *High-Quality* and *Low-Quality* questions) according to the *Voting Score* of the user posts. After that, 25 features along 5 dimensions are chosen in aspects ranging from text length to sentiment strength. Making use of the Boruta algorithm, we capture all features that are in some circumstances relevant to the question quality.

A. Dataset Construction

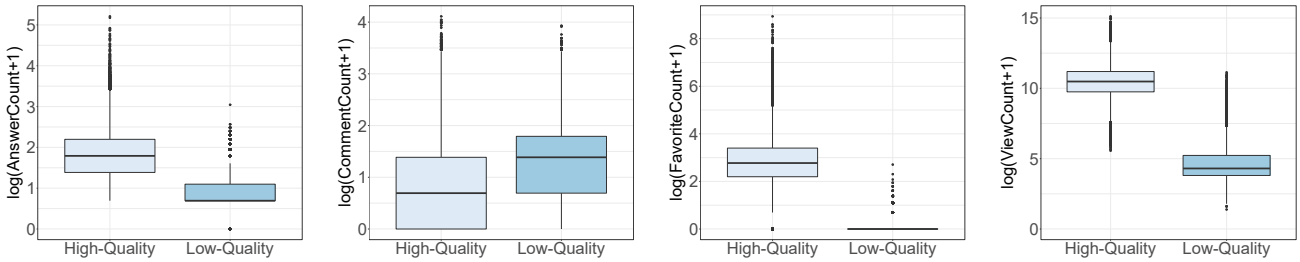
The raw dataset is the publicly available *Stack Overflow Data Dump* for the period of 2008-2016, which is composed of several XML-formatted files under the Creative Commons license [1]. For our purposes, we concentrate on *Posts.xml* that contains 12.35M questions and 19.78M answers.

To construct the two comparison groups, we begin with a definition of the question quality. Actually, there are several statistics that can be used as the criterion (e.g., the number of favorites or views). Analogous to prior studies [3], [4], [5], we employ the *Voting Score* of the user posts, which is the difference between upvotes and downvotes that are given by SO members. Essentially, the more helpful the user post, the higher the voting score.

After ranking all questions based on their voting scores, we extract the top 1% with accepted answers as the *High-Quality* group, while the bottom 1% without accepted answers as the *Low-Quality* group. The average voting scores for the two comparison groups are 84.84 and -3.96 respectively.

*Corresponding author.
[1]https://archive.org/details/stackexchange

978-1-5386-3681-7/17 $31.00 © 2017 IEEE
DOI 10.1109/APSEC.2017.77



(a) AnswerCount (b) CommentCount (c) FavoriteCount (d) ViewCount

Fig. 1. Boxplots (on Logarithmic Scale) Comparing High-Quality Questions with Low-Quality Questions

It is noteworthy that although we specify the *Voting Score* as the indicator of the question quality, there is actually a broader sense of the term *High-Quality*, which also represents more answers (Figure 1a), fewer comments (Figure 1b),[2] more favorites (Figure 1c), and more views (Figure 1d) to a certain extent as compared to the *Low-Quality* group.

B. Feature Selection and Extraction

Unlike previous work, we focus on textual features of user posts mainly because they can be easily improved by SO members with appropriate actions. In other words, we intentionally exclude some widely used factors from this study, even though they are highly correlated with the question quality. For instance, we do not involve community-related aspects (e.g., the reputation of the questioner and the number of acquired badges). Although they are good quality indicators [4], [6], it is impossible for SO members to raise their reputation scores or obtain many badges within a short time to "optimize" their questions.

Our hypothesis is that there must be some characteristics behind good questions, such as clear structure, prominent topic, and easy-to-understand content. As shown in Table I, we carefully choose 25 features along 5 dimensions to capture the textual factors that we are interested in. The selection rationale and the extraction approach are detailed as follows:

1) Size: A set of descriptive statistics is adopted to reveal the basic features of the user posts, for instance, the number of paragraphs, sentences, words, letters, and their ratios. To support text-based analyses, we first combine the title and the content of each question into one single document. After that, we remove all non-text elements (e.g., images). The output serves as the basis for the entire feature extraction process.

2) Element: Because organized information (e.g., lists) could effectively attract viewers' attention and significantly release their comprehension burden, we expect that the presence of certain elements in the user posts correlates closely with the question quality. Thus, we count the number of these elements using HTML tags, for instance, <ul> and <ol> for *Lists*, and for *EmphTexts*.

[2]Comments are essentially messages sent to original posters asking for correction or clarification about their posts. Therefore, the *Low-Quality* group tends to involve more comments as compared to the *High-Quality* group.

3) Readability: Readability refers to a human judgment on how easy it is to understand a text, which is one of the most frequently assessed textual features. In the literature, a wide variety of readability formulas is provided to determine the reading level of a text. Since there is no definitive answer as to which one is the best, it is preferable to involve multiple statistics to keep the result valid and generic [7]. As shown in Table I, we adopt a set of widely used readability metrics in this study (e.g., the Flesch-Kincaid Grade Level [8] and the SMOG Grading [9]). To calculate them, we employ *koRpus*, an R-Package for text analysis. The output is represented in terms of *Reading Grade Level*, which can be interpreted as *easy* for <6, *average* for 7-9, and *difficult* for >9. A lower readability could be an indicator of poor question quality.

4) Lexical Diversity: Lexical Diversity (LD) can be defined as the range of different word stems used in a text [10], which is an important measurement of text difficulty. The lower the value of LD, the higher the quality of the corresponding question. Given that *MTLD* (the Measure of Textual Lexical Diversity), *HDD* (the Hypergeometric Distribution D), and *Maas* (the Maas index) all appear to be able to capture unique LD information, it is advised to use them together [10]. Therefore, we apply the *lex.div* function in the *koRpus* R-Package to calculate these indices, with a greater value indicating a higher diversity.

5) Sentiment: Given that emotions usually have a great influence on human actions and decisions, we assume that the presence of sentiment-related words could significantly affect the question quality. Thus, we adopt SentiStrength [11] (a sentiment analysis tool specialized in dealing with short, even informal text) to determine the polarity of sentiment (i.e., positive, negative or neutral) in a piece of text fragment, which works mainly by assigning a quantitative score to sentiment-related words to estimate the overall sentiment strength. Considering that even short texts can express both positivity and negativity [12], the output of each sentence is organized as a dual tuple (p, n) in 5 point scale:

- p: 1 (not positive) to 5 (extremely positive)
- n: -1 (not negative) to -5 (extremely negative)

Following the similar approach as given in Jongeling et al. study [13], we calculate the document-level result as the sum of the maximum p and the maximum n, which is considered

TABLE I
TEXTUAL FEATURES CONSIDERED IN THIS STUDY[a]

Dimension	Feature	Description	High-Quality		Low-Quality	
			Avg.	Max.	Avg.	Max.
Size	Paragraphs	The number of paragraphs in the question.	4.76	68.00	4.03	83.00
	Sentences	The number of sentences in the question.	7.38	1068.00	5.70	1696.00
	Words	The number of words in the question.	92.09	2659.00	68.36	2984.00
	Letters	The number of letters in the question.	409.50	12190.00	293.40	22260.00
	AvgParaLen	The average number of sentences per paragraph.	1.54	405.50	1.49	341.70
	AvgSntcLen	The average number of words per sentence.	13.83	192.50	15.97	1166.00
	AvgWordLen	The average number of letters per word.	4.40	26.57	4.28	90.77
Element	Lists	The number of ordered/unordered lists in the question.	0.11	40.00	0.04	16.00
	EmphTexts	The number of emphasized/strong texts in the question.	0.47	84.00	0.20	44.00
	Links	The number of hyperlinks in the question.	0.38	164.00	0.14	16.00
	CodeSnippets	The number of code snippets in the question.	2.12	71.00	1.48	300.00
	Tags	The number of tags that describe the topic of the question.	3.00	5.00	2.44	5.00
Readability[b]	ARI	$Automated\,Readability\,Index = 0.5 \times \frac{W}{St} + 4.71 \times \frac{C}{W} - 21.43$	6.21	104.90	6.72	566.40
	SMOG	$Simple\,Measure\,of\,Gobbledygook = 1.043 \times \sqrt{W_{3Sy} \times \frac{30}{St}} + 3.1291$	9.72	27.37	9.26	34.94
	Flesch	$Flesch\,Reading\,Ease = 206.835 - 1.015 \times \frac{W}{St} - 84.6 \times \frac{Sy}{W}$	7.97	16.00	7.87	16.00
	GunningFog	$Gunning\,Frequency\,of\,Gobbledygook = 0.4 \times \left(\frac{W}{St} + \frac{100 \times W_{3Sy}}{W}\right)$	9.07	77.26	9.34	466.40
	FleschKincaid	$Flesch\,Kincaid\,Grade\,Level = 0.39 \times \frac{W}{St} + 11.8 \times \frac{Sy}{W} - 15.59$	7.05	71.73	7.44	451.00
	FORCAST	$FORCAST = 20 - \frac{W_{1Sy} \times \frac{150}{W}}{10}$	9.48	16.67	9.30	18.33
	ColemanLiau	$Coleman\,Liau = 5.88 \times \frac{C}{W} - 29.6 \times \frac{W}{St} - 15.8$	7.53	127.80	6.85	514.10
Lexical Diversity[c]	Maas	$The\,Maas\,Index = \frac{\log N - \log V}{\log^2 N}$	0.22	0.61	0.22	0.67
	MTLD	The average number of sequential words in a text that maintain a certain TTR value.	59.21	1774.00	52.24	5300.00
	HDD	For each lexical type in a text, the probability of encountering any of its tokens in a random sample of 42 words.	30.76	41.74	29.26	42.00
Sentiment	PosScore	The maximum positive sentiment strength.	1.80	5.00	1.75	5.00
	NegScore	The maximum negative sentiment strength.	-1.71	-1.00	-1.55	-1.00
	SentiScore	The document-level sentiment strength.	0.09	4.00	0.19	4.00

[a] In this table, Average/Avg. refers to the arithmetic mean.
[b] St stands for the number of sentences, W for the number of words, C for the number of characters, Sy for the number of syllables, W_{1Sy} for the number of words with exactly one syllable, W_{3Sy} for the number of words with at least three syllables.
[c] N stands for the number of tokens, V for the number of types, TTR for the classic type-token ratio.

positive for $p + n > 0$, *negative* for $p + n < 0$, and *neutral* for $p + n = 0$.

C. Analysis Method

The analysis process can be divided into four steps as described below.

Step1: Correlation Analysis

We first perform a variable clustering analysis to detect collinearity between features making use of the *varclus* function in the *Hmisc* R-Package. The result is represented as a hierarchical overview. For the highly correlated variables (i.e., with an absolute correlation of 0.7 or higher [14]), we reserve only one from each pair.

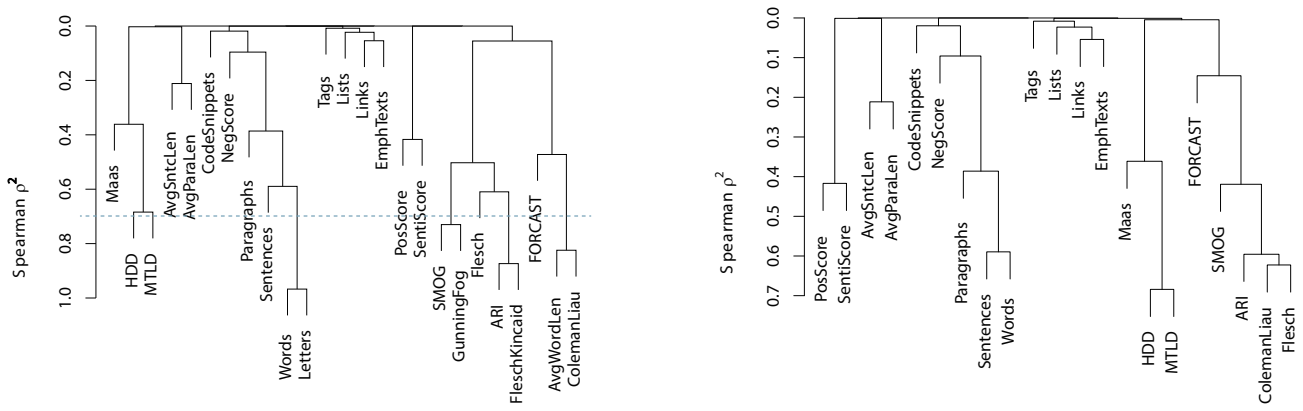
Step2: Redundancy Analysis

To remove redundant features, we apply the *redun* function in the *Hmisc* R-Package to determine to what extent each variable can be predicted from the remaining ones in a stepwise fashion. At each step, the most predictable variable is dropped. The process continues until no variable can be predicted with an adjusted R^2 at least at 0.8.

Step3: Select All-Relevant Features

Considering that the objective of this study is to investigate the textual characteristics of high-quality questions, we intend to identify all features that are either strongly or weakly relevant to the question quality, rather than finding a possibly compact subset of features that yields a minimal error on a chosen classifier. Thus, we employ the Boruta algorithm [15] (an all-relevant feature selection method) to capture all-relevant features carrying usable information, which is more reasonable and applicable in the context of this study as compared to the traditional minimal-optimal feature selection methods. The Boruta algorithm performs a top-down search to iteratively test whether the original feature's importance is significantly higher than random probes, the detailed procedure is introduced as follows:

1) Add shadow attributes (i.e., shuffled copies of all vari-



(a) The Hierarchical Overview before Variable Deduction

(b) The Hierarchical Overview after Variable Deduction

Fig. 2. The Result of the Correlation Analysis

ables) to the given dataset.

2) Apply a *Random Forest* classier on the extended dataset and record the maximum Z-Score (i.e., mean decrease accuracy) obtained among shadow attributes, which is denoted by *MZSA*.
3) Attributes that have importance significantly lower than *MZSA* are classified as *unimportant* and removed permanently, whereas attributes that have importance significantly higher than *MZSA* are classified as *important*.
4) The process is repeated either until all attributes are judged to be confirmed or rejected, or a predefined limit of iterations is reached. The remaining attributes without a decision are claimed as *tentative*.

The Boruta algorithm is implemented with the help of the *Boruta* R-Package. The confidence level is set as 0.99 and the maximal number of importance source runs is set as 100.

Step4: Rank Features by Importance

To determine which features are most influential to the question quality, we employ Z-Score that computed by dividing the average loss of accuracy among trees in the forest by its standard deviation, which is an intrinsic measure that can be used as the feature importance.

III. Empirical Findings

A. Results

The result of the correlation analysis is shown in Figure 2. It can be observed that four pairs of features have a correlation larger than 0.7: 1) *Words* and *Letters*; 2) *SMOG* and *GunningFog*; 3) *ARI* and *FleschKincaid*; 4) *AvgWordLen* and *ColemanLiau*. We randomly remove one from each pair, namely *Letters*, *GunningFog*, *ARI*, and *AvgWordLen*. During the redundancy analysis, we further remove *PosScore* and *FleschKincaid* that can be represented by other variables.

With the remaining 19 features, we perform the Boruta algorithm. After 11 iterations, Boruta confirms all 19 features as *important* that are capable of discriminating the question quality, while none are deemed as *unimportant* or *tentative* during the Boruta run. As shown in Figure 3, all-relevant features proved by Boruta are ranked along the Y-Axis according to the (normalized) Z-Score in descending order, with a higher Z-Score (X-Axis) indicating a greater importance.

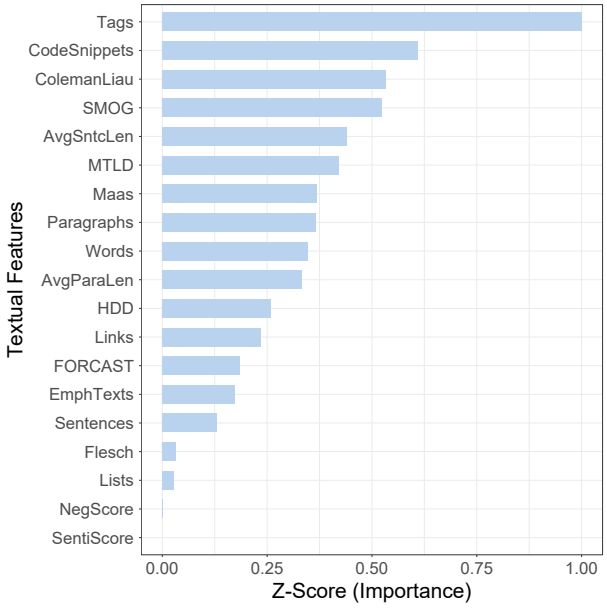


Fig. 3. Rank Features by Importance

B. Implications

The results show that 19 of the 25 selected features have relation with the question quality (Figure 3). The number of tags is identified as the most discriminative one. As a means of sorting questions into specific, well-defined categories, tags do play a critical role in connecting domain experts with questions they are able to answer. Moreover, our analysis emphasizes the importance of the presence of code snippets. The finding agrees with Calefato et al.'s work [16], which is rational as

Stack Overflow (SO) is essentially a programming-specific Q&A website.

Contrary to our expectations, there is only a weak correlation between the question quality and the sentiment-related factors. Taken together with Table I, the statistics (i.e., the average and maximum values of the textual features) suggest that strong negativity is rare. The sentiment expressed in the user posts tend to be mildly positive. Additionally, it can be observed that the *High-Quality* group involves more paragraphs, sentences, and words than the *Low-Quality* group, this is because good questions usually provide enough details. However, the value of *AvgSntcLen* (i.e., the average number of words per sentence) is lower in the *High-Quality* group than that of the *Low-Quality* group, which indicates the necessity of keeping sentences short to help other SO members to grasp the main idea of the user posts quickly.

Based on these empirical findings, we propose a checklist as a guideline for SO members to optimize their questions, which is listed in Table II. The Rel. column represents the relationship between the *High-Quality* group and the *Low-Quality* group. A plus character indicates that the *High-Quality* group has higher value on the feature than the *Low-Quality* group, and a minus character indicates the opposite trend. A set of suggestions is provided in Table II and prioritized according to the feature importance as given in Figure 3.

TABLE II
CHECKLIST FOR MAKING A HIGH-QUALITY QUESTION

Dimension	Feature	Rel.	Suggestion
Element	Tags	+	Include all relevant tags.
	CodeSnippets	+	Provide code snippets.
	Links	+	Present well-sourced facts.
	EmphTexts	+	Make the content skimmable with
	Lists	+	emphasized texts and lists.
Readability	ColemanLiau	+	Check the content readability.[a]
	SMOG	+	Use simple language (if possible,
	FORCAST	+	aim for a readability degree below
	Flesch	+	the 10th-grade level).
Size	AvgSntcLen	-	Keep sentences short.
	Paragraphs	+	Break the content into paragraphs.
	Words	+	Provide a relatively detailed
	AvgParaLen	+	description, yet keep the content
	Sentences	+	to the point.
Lexical Diversity	MTLD	+	Examine the lexical diversity.[a]
	Maas	-	Aim for a low value as the metric
	HDD	+	is an indicative of text difficulty.
Sentiment	NegScore	-	Detect the sentiment polarity.[b]
	SentiScore	-	Use neutral wording.

[a] https://ripley.psycho.hhu.de/koRpus
[b] http://sentistrength.wlv.ac.uk

Table II enumerates all textual features that are proven to be associated with the question quality. We encourage SO members to give careful consideration to these factors while structuring and stating their questions. However, it should be noticed that association does not imply causation, by no means do we claim that a SO question satisfies the checklist in Table II will be absolutely high-quality.

IV. THREATS TO VALIDITY

There are several factors limiting the results of this study.

First, we only consider the dataset from Stack Overflow. It is unclear whether the same findings suitable for all Q&A websites. Further research is necessary to verify the generalization of our conclusions on other communities (e.g., Yahoo! Answers[3]).

Second, we adopt the voting score as the indicator of the question quality. However, some questions attract high scores simply because their topics are interesting. To mitigate this threat, we intend to incorporate controls of these influential factors (e.g., topic and user popularity) in our future work. In addition, the voting score may not be able to represent exact quality. Further study is required to explore other metrics such as question utility [17], [18].

Besides, we choose textual features from different aspects that intuitively seem to have some effect on the question quality. Admittedly, we may have overlooked some representative factors. This threat to validity could be mitigated by empirical investigations extending to additional features.

Another possible threat resides in the use of SentiStrength (a publicly available tool for sentiment analysis) [11]. SentiStrength is initially designed for estimating the sentiment strength of short, informal English text in social web contexts such as MySpace[4], which might not be applicable for domains like software engineering [19]. Further research is needed to address this issue.

V. RELATED WORK

Stack Overflow (SO) is a valuable and indispensable repository of programming-specific knowledge, which has attracted much attentions from academic researchers and industrial practitioners in software engineering community.

Many studies have mined SO data to provide interesting insights. Bazelli et al. [20] explored the personality traits of SO users using the Linguistic Inquiry and Word Count (LIWC). They found that the top, medium, and low reputed authors differed in Neuroticism, Extroversion, Openness, Agreeableness, and Conscientiousness. Asaduzzaman et al. [21] focused on the characteristics of unanswered questions and built models to predict how long a question would remain unanswered. Dalip et al. [22] proposed a learning to rank (L2R) approach for ranking answers in SO. The method outperformed the best baseline with gains of up to 21% in NDCG.

In particular, a lot of research efforts have been directed towards the quality prediction problem. Yang et al. [23] found that the number of editing actions on a question is a significant indicator of the question quality. Correa et al. [24] developed a predictive framework to detect the probability of a question to be deleted. Yao et al. [18] proposed a family of algorithms to jointly predict the voting scores of questions and answers.

[3] https://answers.yahoo.com
[4] https://myspace.com

Duijn et al. [5] analyzed the quality of a SO question based on the code fragments involved in the question. The algorithm classified questions as either good or bad with an accuracy of approximately 80%. Jiarpakdee et al. [6] built prediction models to determine if a question is likely to get no answer using textual, community-based and affective features. Rather than for prediction, our investigation aims to suggest factors that SO members need to take note of while structuring and stating their questions.

The most related research to ours is by Ponzanelli et al. [4] who identified misclassified posts in the review queue using different SO-specific, readability, and popularity metrics. Similar to the prior work we too examine the impact different features of a question have on its quality. However, we consider a different goal. Our study attempts to provide a practical guideline for SO members to optimize their questions. Accordingly, we rule some factors out (e.g., popularity metrics in Ponzanelli et al.'s work such as the number of acquired badges) and employ features from additional dimensions that have not been investigated before, for instance, lexical diversity and sentiment-related factors. As a result, we do obtain some new findings in this study.

VI. Conclusions and Future Work

To identify the textual features of good questions on Q&A websites, we performed a comparative analysis between *High-Quality* and *Low-Quality* questions using Stack Overflow (SO) data for the period of 2008-2016. The main contributions of this study are summarized as follows:

- An empirical study was conducted to provide insights into the textual characteristics of high-quality questions.
- A set of practical suggestions was presented for guiding SO members on how to optimize their questions.

Our work is continuing to investigate additional features that can potentially affect the question quality. Once completed, the development of a prototype tool will begin with the goal of quantitatively assessing the quality of a certain question and providing targeted suggestions on what should be done to make the question better, in an attempt to maximize the utility of Q&A websites. We release our dataset and the corresponding R code to enable critical or extended analyses.[5]

VII. Acknowledgments

This work is supported in part by the General Research Fund of the Research Grants Council of Hong Kong (No. 11208017 and 11214116), and the research funds of City University of Hong Kong (No. 7004683).

References

[1] A. Barua, S. W. Thomas, and A. E. Hassan, "What are developers talking about? an analysis of topics and trends in stack overflow," *Empirical Software Engineering*, vol. 19, no. 3, pp. 619–654, 2014.

[2] Y. Zou, T. Ye, Y. Lu, J. Mylopoulos, and L. Zhang, "Learning to rank for question-oriented software text retrieval (t)," in *Proceedings of the 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE Computer Society, 2015, pp. 1–11.

[5]https://github.com/CityU-QingMi/StackOverflow

[3] Y. Yao, H. Tong, T. Xie, L. Akoglu, F. Xu, and J. Lu, "Want a good answer? ask a good question first!" *arXiv preprint:1311.6876*, 2013.

[4] L. Ponzanelli, A. Mocci, A. Bacchelli, M. Lanza, and D. Fullerton, "Improving low quality stack overflow post detection," in *2014 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2014, pp. 541–544.

[5] M. Duijn, A. Kučera, and A. Bacchelli, "Quality questions need quality code: Classifying code fragments on stack overflow," in *Proceedings of the 12th Working Conference on Mining Software Repositories*. IEEE Press, 2015, pp. 410–413.

[6] J. Jiarpakdee, A. Ihara, and K.-i. Matsumoto, "Understanding question quality through affective aspect in q&a site," in *Proceedings of the 1st International Workshop on Emotion Awareness in Software Engineering*. ACM, 2016, pp. 12–17.

[7] S. Badarudeen and S. Sabharwal, "Assessing readability of patient education materials: current role in orthopaedics," *Clinical Orthopaedics and Related Research*, vol. 468, no. 10, pp. 2572–2580, 2010.

[8] R. Flesch, "A new readability yardstick." *Journal of applied psychology*, vol. 32, no. 3, p. 221, 1948.

[9] G. H. Mc Laughlin, "Smog grading-a new readability formula," *Journal of reading*, vol. 12, no. 8, pp. 639–646, 1969.

[10] P. M. McCarthy and S. Jarvis, "Mtld, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment," *Behavior research methods*, vol. 42, no. 2, pp. 381–392, 2010.

[11] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment strength detection in short informal text," *Journal of the Association for Information Science and Technology*, vol. 61, no. 12, pp. 2544–2558, 2010.

[12] R. Berrios, P. Totterdell, and S. Kellett, "Eliciting mixed emotions: a meta-analysis comparing models, types, and measures," *Frontiers in psychology*, vol. 6, 2015.

[13] R. Jongeling, S. Datta, and A. Serebrenik, "Choosing your weapons: On sentiment analysis tools for software engineering research," in *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2015, pp. 531–535.

[14] Y. Tian, M. Nagappan, D. Lo, and A. E. Hassan, "What are the characteristics of high-rated apps? a case study on free android applications," in *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2015, pp. 301–310.

[15] M. B. Kursa and W. R. Rudnicki, "Feature selection with the boruta package," *Journal of Statistical Software*, vol. 36, no. 11, pp. 1–13, 2010.

[16] F. Calefato, F. Lanubile, M. C. Marasciulo, and N. Novielli, "Mining successful answers in stack overflow," in *Proceedings of the 12th Working Conference on Mining Software Repositories*. IEEE Press, 2015, pp. 430–433.

[17] Y.-I. Song, C.-Y. Lin, Y. Cao, and H.-C. Rim, "Question utility: A novel static ranking of question search," in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, 2008, pp. 1231–1236.

[18] Y. Yao, H. Tong, T. Xie, L. Akoglu, F. Xu, and J. Lu, "Detecting high-quality posts in community question answering sites," *Information Sciences*, vol. 302, pp. 70–82, 2015.

[19] F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli, "Sentiment polarity detection for software development," *Empirical Software Engineering*, Sep 2017.

[20] B. Bazelli, A. Hindle, and E. Stroulia, "On the personality traits of stackoverflow users," in *2013 29th IEEE International Conference on Software Maintenance (ICSM)*. IEEE, 2013, pp. 460–463.

[21] M. Asaduzzaman, A. S. Mashiyat, C. K. Roy, and K. A. Schneider, "Answering questions about unanswered questions of stack overflow," in *2013 10th IEEE Working Conference on Mining Software Repositories (MSR)*. IEEE, 2013, pp. 97–100.

[22] D. H. Dalip, M. A. Gonçalves, M. Cristo, and P. Calado, "Exploiting user feedback to learn to rank answers in q&a forums: a case study with stack overflow," in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2013, pp. 543–552.

[23] J. Yang, C. Hauff, A. Bozzon, and G.-J. Houben, "Asking the right question in collaborative q&a systems," in *Proceedings of the 25th ACM conference on Hypertext and social media*. ACM, 2014, pp. 179–189.

[24] D. Correa and A. Sureka, "Chaff from the wheat: characterization and modeling of deleted questions on stack overflow," in *Proceedings of the 23rd international conference on World wide web*. ACM, 2014, pp. 631–642.