Towards a Weighted Voting System for Q&A Sites

Daniele Romano
Software Engineering Research Group
Delft University of Technology
Delft, The Netherlands
Email: daniele.romano@tudelft.nl

Martin Pinzger
Software Engineering Research Group
University of Klagenfurt
Klagenfurt, Austria
Email: martin.pinzger@aau.at

***Abstract*—Q&A sites have become popular to share and look for valuable knowledge. Users can easily and quickly access high quality answers to common questions. The main mechanism to label good answers is to count the votes per answer. This mechanism, however, does not consider whether other answers were present at the time when a vote is given. Consequently, good answers that were given later are likely to receive less votes than they would have received if given earlier.**

In this paper we present a *Weighted Votes (WV)* metric that gives different weights to the votes depending on how many answers were present when the vote is performed. The idea behind *WV* is to emphasize the answer that receives most of the votes when most of the answers were already posted.

Mining the Stack Overflow data dump we show that the *WV* metric is able to highlight between 4.07% and 10.82% answers that differ from the most voted ones.

***Index Terms*—Mining Repositories; Stack Overflow; Q&A Sites; Software Engineering; Metrics; Social Media; Social Coding**

I. Introduction

In the last decade question-answering web sites (Q&A) have become large repositories of knowledge. The key factors of their success are the ease and speed with which users can access valuable knowledge [1]. Among all the Q&A websites, Stack Overflow[1] has become the most popular site to share and look for software development knowledge [2].

In Stack Overflow, and in all the Q&A sites, the voting system is the main means to distinguish high quality answers from low quality ones [3]. Users can up-vote good answers, and down-vote bad answers. As consequence, users looking for good answers can easily focus their attention on answers that get more votes. However, such a voting system has a great disadvantage that can put good quality answers in the background. The count of the votes, on which users rely on, does not take into account the number of answers posted when a vote has been given. Most of the votes could be performed when only few answers to a question have been posted. Hence, the number of votes might not highlight the most valuable answer. As consequence users could be misled.

In this paper we propose a new way to count the number of votes that can overcome this problem. We introduce the *Weighted Votes (WV)* metric that gives different weights to votes depending on the number of answers already posted when a vote is given. The goal of the *WV* metric is to emphasize the answers that receive most of the votes when most of the answers are present.

To analyze the ability of *WV* in highlighting answers different from the most voted ones we have mined the Stack Overflow data and computed the values of *WV* for 4,392,956 answers. The results show that *WV* ranks between 4.07% and 10.82% of the answers higher than the traditional approach. Moreover, we analyzed the extracted data to give an insight into the amount of answers already posted when votes are performed.

The remainder of this paper is organized as follows. In Section II we introduce the *Weighted Votes* metric, we reason about its integration into Q&A sites and discuss the benefits for their communities. Section III presents our study, its results and the process to extract the necessary data. We conclude this paper and draw directions for future work in Section V.

II. The Weighted Votes metric

When a user is looking for the valuable answer to a question of interest she may focus on the most voted answers, especially if the question gets numerous answers. However, the current voting system adopted by Q&A sites is limited to count the number of votes an answer receives along its lifetime. The main limitation of such a system is that most of the votes can be performed immediately after the answer is posted. Hence, they do not take into account the answers posted later.

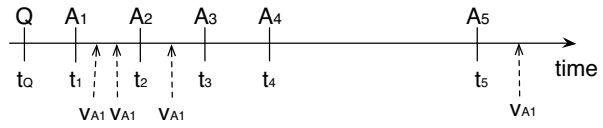
We propose a new way to count votes that takes into account the number of answers to a question already posted when a vote is performed and the total number of answers. We suggest to give different weights to the votes depending on the number of answers already posted when it is given. For an answer *A* to a question *Q* we define the *WeigthedVotes* metric (*WV(A)*) as follows:

$$WV(A) = \sum_{k=1}^{n} \frac{Answers_Q < t_k}{Answers_Q} \quad (1)$$

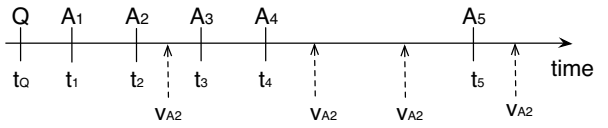
where n is the number of votes given for the answer A; $Answers_Q$ is the total number of answers to Q; t_k indicates the time when the vote k was performed and $Answers_Q{<}t_k$ indicates the number of answers given to A and posted before the vote k was performed.

[1] http://stackoverflow.com

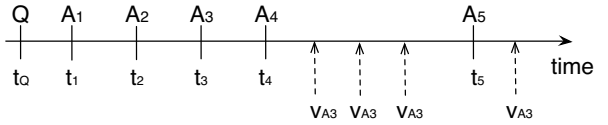
1063-6773/13 $26.00 © 2013 IEEE
DOI 10.1109/ICSM.2013.49



(a) Votes given for answer A_1



(b) Votes given for answer A_2



(c) Votes given for answer A_3

Fig. 1: Timelines showing five answers posted to a question Q and votes given for three answers (A_1,A_2 and A_3)

A. Working Example

Consider the example shown in Figure 1. The example shows a question Q with five different answers posted at different times. Figure 1a, Figure 1b and Figure 1c display respectively the votes given for the answers A_1, A_2 and A_3. According to the current voting system adopted by Stack Overflow all answers A_1, A_2 and A_3 will have the same amount of votes (*i.e.*, 4). As consequence, the user who is looking for the best answer would not know that most of the votes given for answer A_1 had been given without considering the other answers. Precisely, two votes were performed when only the answer A_1 was posted, one vote when the answer A_2 was posted and only one vote when all the answers to Q were posted. On the other hand, the answer A_3 is not emphasized by the number of votes. Even though its votes were performed when four out five answers were posted.

With our metric defined in 1, the ranking of the three answers differ. In fact, when computing the metric values for each answer after the last vote for the answers of question Q has been recorded, we obtain for $WV(A_1)=\frac{1}{5}+\frac{1}{5}+\frac{2}{5}+\frac{5}{5}=\frac{9}{5}=1.8$, $WV(A_2)=\frac{2}{5}+\frac{4}{5}+\frac{4}{5}+\frac{5}{5}=\frac{15}{5}=3.0$ and $WV(A_3)=\frac{4}{5}+\frac{4}{5}+\frac{4}{5}+\frac{5}{5}=\frac{17}{5}=3.4$. Our metric WV clearly highlights the answer, namely in this example A_3, who obtained most of the votes when most of the answers were present.

B. Integration into Q&A sites

The computation of our proposed WV metric can be easily integrated into Q&A sites. The only requirement needed is the ability to update the value of WV for an answer when a new answer is posted. For instance, imagine that an answer A_6 is posted after A_5 in our example shown in Figure 1. In this scenario we should update on the fly the WV values of the other answers. For example, the WC of the answer A_1 would be recomputed as follows: $WV(A_1)=\frac{1}{6}+\frac{1}{6}+\frac{2}{6}+\frac{5}{6}=\frac{9}{6}=1.5$.

C. Impact to the community

Besides the benefits for users looking for good quality answers explained in Section II-A, the WV metric can bring another important advantage for Q&A communities. According to our proposed metric, votes to later answers have a higher weight. This can stimulate people to provide more answers in order to receive votes with higher weights affecting consequently their reputation in the community [4].

III. The Study

In this study we mine the Stack Overflow system in order to measure the values for the WV metric for each answer posted. The *goal* consists in evaluating whether the WV metric highlights different answers compared to the ones higlighted by the number of votes. The *quality focus* is the ability of the WV metric to differentiate the answers with the highest values of WV from the most voted answers. The *perspective* is that of a Q&A site designer who wants to improve its voting system emphasizing votes performed when most of the answers to a question had already been posted. The *context* of this study consists of the latest official dump of the Stack Overflow data that contains all activities performed since July 2008 until August 2012. Among all Q&A sites we decided to mine the Stack Overflow system because it has become the most popular Q&A site for sharing software development knowledge. Moreover, among all the Q&A sites published on the Stack Exchange network[2] it provides the biggest data set for our analysis.

In this paper we answer the following research question:

> *To what extent does the* WV *metric highlight answers different from the answers with the highest number of votes?*

In the following subsections, first we describe the process to extract the data necessary for our analysis. Then we report our results and observations about the extracted data.

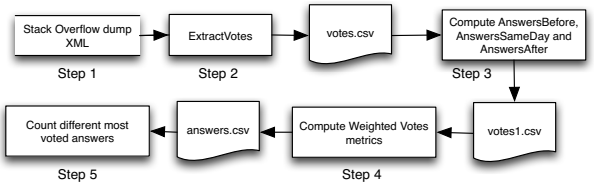


Fig. 2: Process used to extract the data for our analysis.

A. Data Extraction

Figure 2 shows the approach we used to extract the data from the Stack Overflow data dump.

[2]http://data.stackexchange.com

TABLE I: Percentage of answers highlighted by *WV* that differ from the most voted ones for different categories of questions.

	$Answers >= 2$	$Answers = 2$	$Answers = 3$	$Answers >= 4$
Questions (%)	63.96%	29.38%	16.67%	17.92%
WV_{high}	10.31%	5.88%	10.75%	17.17%
WV_{low}	3.21%	2.26%	3.52%	4.48%
Average	6.76%	4.07%	7.13%	10.82%

In the first step we downloaded the data dump in XML format from the Stack Exchange website.[3] The data dump consists of five XML files that store information about the users (*users.xml*), the posts (*posts.xml*), the comments (*comments.xml*), the posts' history (*posthistory.xml*) and the badges *badges.xml*.

In the second step, for each answer contained by *posts.xml* we extract the *up* and *down* votes from the *votes.xml* file. We discarded the votes for answers that have been removed from the database. The output of this step consists of the *votes.csv* file that for each vote contains 1) the *id* of the answer for which the vote has been given, 2) the *id* of the question of the answer and 3) the *creation date* of the vote. In total we extracted 13,700,939 votes of 4,392,956 answers given to 2,421,549 questions.

In the third step, we prepared the data to compute the values for the *WV* metric. To be able to measure the *WV* we needed for each vote k the count of all answers posted before the vote was given ($Answers_Q<t_k$) and total number of answers ($Answers_Q$) given to a question Q. However, differently from the *creation date* of answers, the *creation date* of a vote does not contain the information about hours, minutes and seconds. Its format is in the form month-day-year. As consequence we cannot know if the answers posted on the day when the vote k is given are actually performed before or after the vote. This format is used in all Stack Exchange data dumps and not only for Stack Overflow. For this reason we computed 1) the number of answers posted on the days that precede the day when a vote is given (*AnswersBefore*); 2) the answers posted on the same day (*AnswersSameDay*); and 3) the answers posted on the following days (*AnswersAfter*). These values allow us to estimate the actual value of the *WV* metric as explained in the next step. The output of this step consists of the *votes1.csv* that enriches the *vote.csv* file adding for each vote the values of (*AnswersBefore*), (*AnswersSameDay*) and (*AnswersAfter*).

Since we cannot order the *AnswersSameDay*, in the fourth step we computed the values of two variants of *WV*. We computed the values of WV_{low} and of WV_{high}. Computing WV_{low} we assume that the *AnswersSameDay* have been posted after the vote was given. On the other hand, computing WV_{high} we assume that the *AnswersSameDay* were posted before the vote has been given. In this way WV_{low} and WV_{high} are the lower and upper boundaries of the actual value of *WV*. The values for WV_{high}, WV_{low} and the number of votes for each answer are saved in *answers.csv*.

[3]http://data.stackexchange.com/

In the last step (Step 5 in Figure 2), for each question we compared the ranking of the answers obtained with the *WV* metric and the traditional approach and computed the ratios of answers for which the ranking differed.

B. Results

Table I shows the results obtained. Among all the questions analyzed we report the results of questions with a number of answers greater than two (*Answers*$>=$ 2). They account for 63.96% of all questions. For the questions with only one answer the value for *WV* is equal to the number of votes. Moreover, we report the results for questions with two answers (*Answers=2*), questions with three answers (*Answers=3*) and questions with four or more answers (*Answers>=4*). We chose these values because they represents the median number of answers (*i.e.*, three) and the 75th percentile (*i.e.*, four).

From the results we can state that for the questions with more than two answers (*Answers>=2*) the *WV* metric emphasizes on average 6.76% different answers. In such cases the user can focus on answers that received most of the votes when most of the answers were already posted. For questions with two, three and four or more answers we registered on average respectively 4.07%, 7.13% and 10.82% of different answers highlighted by the *WV* metric.

In conclusion, we can answer our research question stating that the percentage of different answers highlighted by *WV* is 1) between 3.21% and 10.31% for questions with two or more answers, 2) between 2.26% and 5.58% for questions with two answers, 3) between 3.52% and 10.75% for questions with three answers and 4) between 4.48% and 17.17% for questions with four or more answers. On average the *WV* metric highlights a percentage of different answers that ranges from 4.07% to 10.82%.

C. Observations

Besides the *WV*'s ability of highlighting different answers we can make two important observations reading the results shown in Table I.

First, we can notice that the percentage of different answers highlighted with *WV* increases when we consider questions with a higher number of answers. For WV_{high} we registered an increment of $\approx$ 292% (17.17/5.88) between questions with two answers and questions with four or more answers. For WV_{low} we registered an increment of $\approx$ 198% (4.48/2.26) between questions with two answers and questions with four or more answers.

Second, we can notice the difference between the values measured for WV_{high} and WV_{low}. In order to understand

TABLE II: Paired Cliff's delta effect sizes (d) between *AnswersBefore*, *AnswersSameDay* and *AnswersAfter*. The effect size is considered negligible for $d < 0.147$, small for $0.147 \leq d < 0.33$, medium for $0.33 \leq d < 0.47$ and large for $d \geq 0.47$ [5].

Distribution1	Distribution2	Cliff's d
AnswersBefore	*AnswersSameDay*	0.053
AnswersBefore	*AnswersAfter*	0.318
AnswersSameDay	*AnswersAfter*	0.232

this gap we analyzed the difference of the distributions of *AnswersBefore*, *AnswersSameDay* and *AnswersAfter* measured for each vote. We computed the Mann-Whitney p-value for paired samples for each pairs of distributions to test if the distributions were different. For all pairs we registered p-values smaller than 0.01 indicating that the distributions are considered statistically different. Moreover we computed the Cliff's delta effect size (for paired samples) [5] to measure the magnitude of the difference and we report the results in Table II.

The results show that the difference in magnitude between the distribution of answers posted on days before the day when a vote is given (*AnswersBefore*) and the distribution of answers posted on the same day of a vote (*AnswersSameDay*) is negligible (d=0.053<0.147)[5]. The distribution of answers posted after a vote (*AnswersAfter*) is smaller than the distributions of *AnswersBefore* and *AnswersSameDay* because the effect sizes' values (d=0.318 and d=0.232) are considered to be medium [5]. From these results we can state that the distributions of *AnswersBefore* and *AnswersSameDay* are the biggest ones. This explains the difference between the values for WV_{low} and WV_{high} registered in our study.

IV. Related Work

In the last years many studies on Stack Overflow have been presented. The closest to our study has been developed by Schall *et al.* [6]. They analyzed the dynamics of the community activities. As part of this analysis they analyzed the answering behavior per question showing the number of answers per different categories of questions. However, they have not analyzed the voting behavior and the main focus of their work is the mining of expertise.

Among all scientific work about mining Q&A sites, mining expertise from Q&A communities is becoming more and more popular. Many of them propose technique to mine expertise of users in the community, such as [7] [8] [9] [10] [11]. These works propose techniques and approaches to infer the expertise from several variables. Among these variable the number of votes plays a crucial role. The *WV* metric proposed in this paper can help to improve these approaches. For example, it can be used to filter votes given when only one answer is posted.

V. Conclusion and Future Work

In this paper we proposed a *Weighted Votes* metric aimed at highlighting answers that received most of the votes when most of the other answers were already given to a question. Mining the Stack Overflow data dump, we showed that the proposed metric is able to emphasize answers different from the most voted ones. This is particularly useful for users who are looking for high quality answers.

In our future work we plan to further validate and improve this metric. First, we plan to look for data in which the complete timestamp of a vote is registered. This allows us to obtain more precise results avoiding the approximations performed in this study (*i.e.*, the computation of WV_{high} and WV_{low} to estimate the actual value of *WV*).

Second, we plan to perform a qualitative study to test to which extent the number of votes is relevant to users looking for answers. It is particularly useful to investigate if users go through all the answers or if they read only the most voted ones or the accepted ones.

Finally, we plan to perform a qualitative analysis with questionnaires to find out whether the answers highlighted by the *Weighted Votes* metric are considered of better quality compared to the most voted ones.

Acknowledgment

This work has been partially funded by the NWO-Jacquard program within the ReSOS project.

References

[1] C. Treude, F. Figueira Filho, B. Cleary, and M.-A. Storey, "Programming in a socially networked world: the evolution of the social programmer," in *The Future of Collaborative Software Development*, 2012, pp. 1–3.

[2] L. Mamykina, B. Manoim, M. Mittal, G. Hripcsak, and B. Hartmann, "Design lessons from the fastest q&a site in the west," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2011, pp. 2857–2866.

[3] C. Treude, O. Barzilay, and M.-A. D. Storey, "How do programmers ask and answer questions on the web?" in *Proceedings of the International Conference on Software Engineering*, 2011, pp. 804–807.

[4] A. Anderson, D. P. Huttenlocher, J. M. Kleinberg, and J. Leskovec, "Discovering value from community activity on focused question answering sites: a case study of stack overflow," in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2012, pp. 850–858.

[5] R. J. Grissom and J. J. Kim, *Effect sizes for research: A broad practical approach*, 2nd ed. Lawrence Earlbaum Associates, 2005.

[6] D. Schall and F. Skopik, "An analysis of the structure and dynamics of large-scale q/a communities," in *Proceedings of the International Conference on Advances in Databases and Information Systems*, Berlin, 2011, pp. 285–301.

[7] A. Pal, R. Farzan, J. A. Konstan, and R. E. Kraut, "Early detection of potential experts in question answering communities," in *Proceedings of the International Conference on User Modeling, Adaptation, and Personalization*, 2011, pp. 231–242.

[8] N. Raj, L. Dey, and B. Gaonkar, "Expertise prediction for social network platforms to encourage knowledge sharing," in *Proceedings of the International Conference on Web Intelligence*, 2011, pp. 380–383.

[9] B. V. Hanrahan, G. Convertino, and L. Nelson, "Modeling problem difficulty and expertise in stackoverflow," in *Proceedings of the International Conference on Computer Supported Cooperative Work and Social Computing*, 2012, pp. 91–94.

[10] A. Pal, F. M. Harper, and J. A. Konstan, "Exploring question selection bias to identify experts and potential experts in community question answering," *ACM Trans. Inf. Syst.*, vol. 30, no. 2, pp. 10:1–10:28, 2012.

[11] A. Pal, S. Chang, and J. A. Konstan, "Evolution of experts in question answering communities," in *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2012.