Manual Verification Results of the Detection of Similar Projects

December 27, 2021

1 Description of this material

To test the effects of our methods on detecting similar projects, we need participants to decide whether the query projects are similar to the target project. Since this process may introduce personal bias, we ask four researchers to to evaluate the effectiveness of our detection of similar projects. The detailed process of our evaluation is shown in Section 2.1, and we discuss the results of our evaluation in Section 2.2, Then, the limitations of our detection of similar projects are discussed in Section 3.

2 Our manual evaluation process

First, randomly select 1000 samples, including queried projects and projects that are considered as similar to queried projects by our automatic method. Second, assign the 1000 samples to four Participants (i.e., the four researchers in our study). Third, for each sample, offer the four participants the URLs of the GitHub repositories and therefore the participants can access the code, authors and contributors, README file, related links and other information. Finally, according to the information, We ask the question about the relevance (see Table 1) of each retrieved repository to the query projects. (This relevance division is often used to evaluate the similarity of selecting similar items, e.g., [2, 1]).

2.1 Evaluating the effects of the manual verification

We follow the evaluation method provided by [[2, 1] to evaluate the effects of the manual verification, that is, SuccessRate, Confidence and Precision.

(1) **SuccessRate**: if a retrieved project is rated as 4 or 5 by a participant, we consider this project to be successfully detected by our automatic method.

Table 1: The relevance of the retrieved repository to the query repository

Level	**Description**
Highly Irrelevant (rating 1)	The participant finds that there is absolutely nothing in common between the retrieved and query repositories.
Irrelevant (rating 2)	The participant finds that the two repositories have little in common.
Neutral (rating 3)	The participant finds that the two repositories are marginally relevant.
Relevant (rating 4)	The participant finds that the two repositories are similar on a number of aspects.
Highly Relevant (rating 5)	The participant finds that the retrieved and query repositories are similar in most aspects, and even some parts may be identical.

Table 2: An example of the results of two queries

Project	**Participant1**	**Participant2**	**Participant3**	**Participant4**
Query 1	Precision for query 1: 0.250 (2/8)			
Retrieved 1	3	2	5	3
Retrieved 1	3	1	5	3
Query 2	Precision for query 2: 0.667 (8/12)			
Retrieved 2	4	3	4	2
Retrieved 2	4	5	4	
Retrieved 2	5	4	4	3
Confidence	Median:4 Mean:3.8	Median:3 Mean:3	Median:4 Mean:4.4	Median:3 Mean:2.6
Confidence (ALL)	Median:3 Mean:3.45			
SuccessRate	0.600 (3/5)	0.400 (2/5)	1.000 (5/5)	0.000 (0/5)
SuccessRate (ALL)	0.500(10/20)			

Then we use precision[1] to evaluate the effectiveness of our method. We evaluate SuccessRate for four participants (i.e., labeler in this process) and the overall SuccessRate for all participants. An example is shown in the Table 2.

(2) **Confidence**: Median and mean confidences are defined as the median and mean relevance degrees that participants give to all retrieved repositories recommended by a system. An example is shown in the Table 2.

Consider that the personal bias of different participants, we calculate the consistency of assessment by the Fleiss kappa coefficient.

[1]Precision is defined as the proportion of relevant and highly relevant repositories (i.e., whose final relevance degrees are equal to or greater than 4) among the recommendations that our method generates for a query.

Table 3: Results of the manual verification

Measure	**Participant1**	**Participant2**	**Participant3**	**Participant4**
Confidence	Median:5 Mean:4.45	Median:5 Mean:4.47	Median:4 Mean:4.26	Median:4 Mean:4.20
Confidence (ALL)	Median:4 Mean:4.345			
SuccessRate	0.967	0.962	0.811	0.786
SuccessRate (ALL)	0.881			

2.2 Results of the manual verification

Our results contain 76 query projects and 924 retrieved projects, and the overall Kappa value of the four participants considering all queries is 0.72, which indicates substantial agreement between the participants. The results of the confidence and SuccessRate are shown in Table 3. The confidence box plot is shown in Figure 1 and the precision box plot is shown in Figure 2.

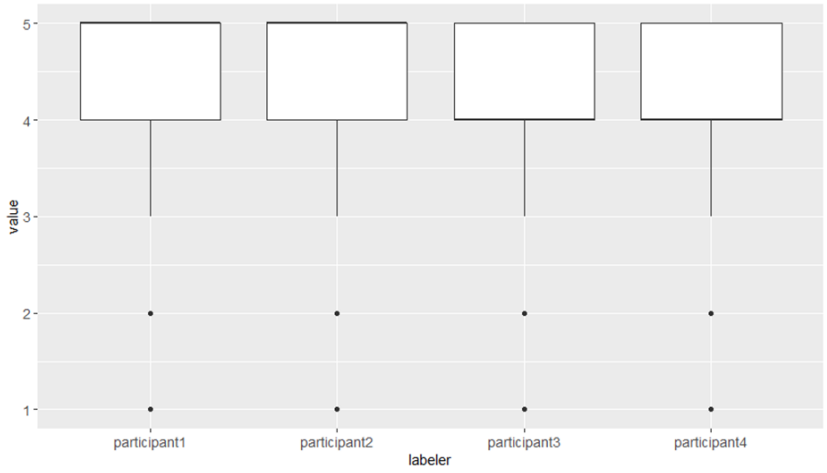

Figure 1: The box plot of confidence

From the results in Table 3, we can see that the mean confidence score of our method is 4.345 and the median score of our method is 4. In addition, the SuccessRate of our method is 0.881, which means that the retrieved projects selected by our method are likely to be similar to the query projects.

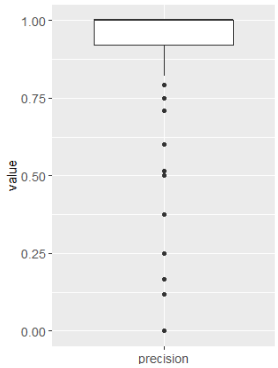

Figure 2: The overall box plot of precision

3 Limitation

Although our method performs well in finding similar projects. We notice two potential limitations of our method. First, we mainly use the similarity between descriptions of projects to find similar projects. However, not all projects describe their project content in the description. As a result, our method will miss this part of projects for a query project. Hence, the similar projects selected by our method are basically the projects with very similar descriptions to the query project. Second, two projects with similar descriptions do not mean that they have similar functions. For example, two projects have description "wow addon" are projects that develop addons to a famous online game wow. The one can be a "quest addon" and the other can be a "chat addon". However, from the precision box plot in Figure 1, we find that this kind of problems do not appear frequently, which means that this threat is limited.

References

[1] Nafi, K.W., Roy, B., Roy, C.K., Schneider, K.A.: A universal cross language software similarity detector for open source software categorization. Journal of Systems and Software **162**, 110491 (2020)

[2] Zhang, Y., Lo, D., Kochhar, P.S., Xia, X., Li, Q., Sun, J.: Detecting similar repositories on github. In: Proceedings of the 24th International Conference on Software Analysis, Evolution and Reengineering (SANER), pp. 13–23. IEEE (2017)