

Retrieving Curated Stack Overflow Posts from Project Task Similarities

Glauca Melo

David R. Cheriton

*School of Computer Science
University of Waterloo
Waterloo, Canada
gmelo@uwaterloo.ca*

Toacy Oliveira

PESC/COPPE

*Universidade Federal
do Rio de Janeiro
Rio de Janeiro, Brazil
toacy@cos.ufrj.br*

Paulo Alencar

David R. Cheriton

*School of Computer Science
University of Waterloo
Waterloo, Canada
palencar@uwaterloo.ca*

Don Cowan

David R. Cheriton

*School of Computer Science
University of Waterloo
Waterloo, Canada
dcowan@uwaterloo.ca*

Abstract—Software development depends on diverse technologies and methods and, as a result, software development teams often need to handle issues in which team members are not experts. To address this lack of expertise, developers typically rely on information obtained from web-based Q&A sites such as Stack Overflow, a popular platform to find solutions to specific technology-related problems. However, access to these Q&A websites is currently not explicitly integrated with software development projects. Therefore, software developers often need to search for solutions to similar and recurring issues multiple times. This lack of integration not only hinders the reuse of the knowledge obtained but also compels developers to perform repeated searches for recurring problems. In this paper, we investigate an approach that explicitly associates project tasks with Stack Overflow posts that have already been curated by developers, and use project task similarities to investigate the possibility to suggest curated Stack Overflow posts. Precision and accuracy were 71.60% and 77.78%, respectively. We also found indications that project task elements such as the process activity, influence accuracy and precision if attempting to reuse curated Stack Overflow posts.

Index Terms—Software engineering, stack overflow, project task, text similarity.

I. INTRODUCTION

Software development is a knowledge-intensive collaborative activity [1]. Currently, development changes with a variety of technologies in use. Therefore, new knowledge must be constantly gathered and software engineers need to engage in tasks that are related to knowledge management, such as learning, capturing, and reusing collaborative knowledge during a software project [2]. During the execution of a software development project, knowledge and expertise from developers are vital for the project to succeed [2].

Software development is usually an integrated system of code editors and debuggers [3] in which developers interact to build a software product. To acquire external support (e.g., code snippets), developers frequently switch between the development environment and browsers [4]. In other words, developers must leave the development environment, reason about relevant and accurate terms for searches, open a browser, verify the results of the search, check if the source is reliable, and only then, transfer the knowledge obtained to the software

[5] [4]. Such activity usually occurs more than once, as software projects are large-scale and iterative. We refer to this effort of tapping into sources of support, reasoning about the help needed and choosing among the vast available content, as *curation*. We use curation inspired by humanities, and which is defined as:

”Select, organize, and present (online content, merchandise, information, etc.), typically using professional or expert knowledge.”

Essential sources of knowledge (information software developers use for support while working) are question and answer (Q&A) websites. A famous Q&A website for software engineers is Stack Overflow (SO) [6] [7] [8] [9]. Although SO is widely used during software development [6], there are still issues about explicitly associating the tasks performed during development and the knowledge obtained from a SO post. The lack of integration of the support often needed by software developers with the development project is identified by researchers as an open issue [10] [11]. Some works have proposed solutions to integrate SO with the software project through text overlap, mainly focusing on issues (bugs, exceptions) [11] [12].

Research indicates that developers’ expertise largely contributes to the success of software projects [2]; however, current approaches to selecting SO posts that use text overlap do not consider developers’ expertise. Integrating curated SO posts with software development projects could help developers avoid performing curation multiple times, redundantly. Additionally, integrating SO posts can convey other benefits, such as keeping relevant information into the project, avoiding searches of the same information, helping less experienced developers know how experts are working and reducing workflow interruptions [4].

In this paper, we investigate the possibility to reuse curated SO posts based on project task similarity by performing an investigation on task contexts and submitting these contexts to a similarity retrieval model. Precision and accuracy (the most common metrics identified among our related works) were collected. Two research questions guide the evaluation of the implemented model: RQ1 investigates the precision and accu-

racy of our proposal, facilitating a comparison among other works. RQ2 compares different task contexts to understand how different combinations can influence SO post reuse.

This paper is structured as follows. Section I presents an introduction of the discussed subject. Section II presents the study and implementation conducted to investigate the association of Stack Overflow posts with project tasks. After, Section III evaluates the study and implementation. Conclusion, discussion, and future work are presented in Section IV.

II. STUDY ON RETRIEVING CURATED STACK OVERFLOW POSTS

Curation is the act of searching and selecting useful SO posts for a given problem during software development. The process of curating SO posts is as follows. First, the developer that has a problem or that needs support creates a search string that may retrieve satisfactory results. Second, the search string is submitted to SO. Third, SO executes the search according to internal algorithms and lists the results. Finally, the developer selects one (or a set of) SO post that can be used as support. Each of these steps can be executed repeatedly until developers are satisfied with the results listed and choose a SO post that meets their needs. Once a solution is chosen, curation is over. If results are not helpful, the developer changes the string, exploring new terms that can indicate useful results.

Frequently, after a solution from a SO post is selected, the post from where the solution was extracted is not associated with the project task and, therefore, cannot be reused by developers dealing with similar or recurring tasks. A growing body of literature recognizes the importance of associating external knowledge with the development [13] [11] [14] [10] [15]. Stack Overflow is the source of support considered in this work, and this support is presented in the form of curated SO posts. Through the identification of similar project tasks, SO posts that were once used to support a project task during its solution could be automatically associated with project tasks with similar context(s), therefore making developers aware of this SO post previously used by another developer. After a preliminary investigation [16], this work was further developed. We researched what comprises the context elements of a project task and implemented a process model using the investigated context in Rapidminer¹.

A. Project Task Context

Project Task Context is a set of elements that compose a project task. Investigating possible project task context elements is essential to contribute to clarifying what information from project tasks are available for similarity comparisons. For completeness purposes, we have taken advantage of both academic and industrial perspectives, including project management tools that support project tasks in software development. It is important to consider other sources in grounding research other than formal literature in software engineering [17], allowing a broader theoretical aspect and bringing practical insights to the work.

¹rapidminer.com

From the academic perspective, after an *ad-hoc* literature review, we found that software engineering is knowledge-intensive due to its dynamism and the massive amount of technology used activity [1] [18]. According to Lindvall and colleagues [18], software engineering has two types of knowledge associated with the project: technical and business domain information. Our work does not consider application domain information; it only considers technical information because it aims to be agnostic to business characteristics. We propose a context element that captures technological information about project tasks (e.g., tags). A tag is a piece of information related to an element. In this case, the elements are any technical information directly related to the task that can characterize the task. Each project has a specific context in respect to product [19], such as technical characteristics that every task will inherit necessarily, indicating the need for a project tag.

From an industrial perspective, we analyzed project management tools, by verifying the default elements each tool has for project tasks. In this *ad-hoc* analysis, we concluded that some of the project task context elements identified in the literature were also reported in software development tools that support project workflows. The analyzed tools were JIRA², Trello³ and Redmine⁴, which are broadly used. After performing an analysis of both literature and project management tools contents, we present a list of the identified context elements in Table I.

TABLE I
CONTEXT ELEMENTS.

Element	Description	Source
Project/Board	The name of the project that tasks belongs to	Redmine, Trello JIRA
Project Tag	Tags related to the project	Lindvall et al. [18]
Process	Process information that can be associated with the task	Lindvall et al. [18]
Title	The title of the task	Redmine, Trello JIRA
Description	The description of the task	Redmine, Trello JIRA
Category	A classification used to divide tasks into different niche	Redmine JIRA
Task Tag	Tags related to the task	Lindvall et al. [18], JIRA, Trello

B. Study Implementation

After project task context elements were investigated in both literature and project management tools, we propose an implementation to obtain the similarities between project task context elements. We implemented a process model that is prepared to receive as input a dataset containing project tasks associated with SO posts, and retrieves similarity indexes from pairs of project tasks and evaluates if the SO posts are the same between project tasks with a high degree of similarity.

²atlassian.com/Jira

³trello.com

⁴redmine.org

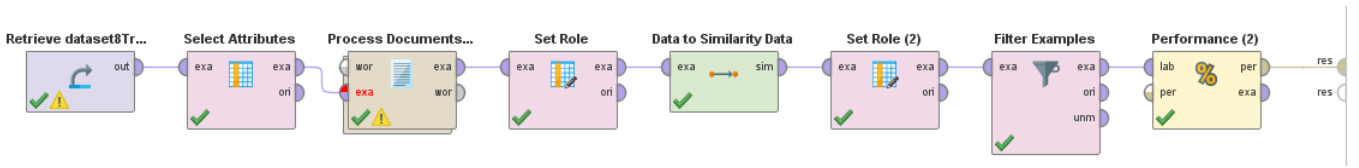


Fig. 1. RapidMiner process model.

RapidMiner (RapidMiner Studio version 8.2) was used to implement a similarity process model. RapidMiner is a compelling data science platform, requiring a small learning curve to be used, widely adopted in the academic field [20]. The implemented RapidMiner model is illustrated in Figure 1. The model loads a dataset of project tasks, pre-processes the text and executes a Jaccard algorithm [21] to retrieve text similarities. More details can be found in the work of [22] and the source code for the implemented model can be found on GitHub (github.com/glauciams/task2stackRapidMiner).

III. EVALUATION

Based on the guidelines proposed by Shani et al. [23] we verify the effectiveness of using task similarities to reuse curated SO posts, through well-established metrics: precision and accuracy. The research questions for the evaluation are:

RQ1: What are the precision and the accuracy metrics for the collected sample? It is essential to verify these metrics to gather quantitative results while ascertaining the effectiveness of considering similar project tasks to reuse curated SO posts. The metrics precision and accuracy were chosen after being the most common metrics identified in the works that relate to ours.

RQ2: What are the impacts in precision and accuracy when different context elements are combined? We evaluate different project task context combinations because task contexts can vary in each project. A project can maintain records of processes and another project might not, for example. Given this variation, it is essential to understand the impacts of different project task context combinations.

Hypothesis: Project task similarity can provide helpful suggestions for curated Stack Overflow posts. We test this hypothesis by verifying whether similar tasks (similarity above 50%) share the same Stack Overflow posts.

Controlling Variables: Considering this study uses only one dataset, having fixed controlled variables is not a concern. We propose a study considering different variable combinations to analyze the effects of the absence or presence of variables on precision and accuracy.

A. Executing the implemented process model

To execute the implemented process model, a dataset with project tasks has to be loaded to the RapidMiner process model. The selected dataset was gathered from a company in Brazil that has been developing software products for more than 20 years and has a total of 30 employees. The software development projects in this company follow agile guidelines, and the project tasks are managed with the support of a project

management tool. We were able to gather 25 project tasks with associated SO posts to each of the 25 tasks. The software developers and managers of the company provided a dataset of project tasks (context elements and SO posts associated with each task).

B. Results, Discussion and Threats to Validity

After executing the evaluation of the dataset, precision and accuracy are calculated, using the Jaccard algorithm. A confusion matrix is generated by the execution of the RapidMiner process model, which supports the extracted results.

Answering RQ1, the accuracy for the given dataset with the elements identified in Section II is 77.78%, and the precision mean 71.60%. Other works with similar characteristics present different precision and accuracy results. Rahman et al. [24] report 11% of precision and 88% of accuracy. The work of Wang et al. [25] reports 62% of precision.

Answering RQ2, Table II presents the context attributes' combination selected, the precision and accuracy extracted for each combination, and changes made in each combination, as it is not easy to perceive from the attribute list which attributes were selected and which were not in the combination.

TABLE II
RQ2 RESULTS: CONTEXT COMBINATIONS

Precision	Accuracy	Combination changes
71.60%	77.78%	Current Work - RQ1
61.17%	70%	Included Iteration
48.01%	54.69%	Removed Process
69.81%	66.67%	Removed Project Tags and Task Tags
45.64%	37.12%	Removed Title and Description
40.87%	36.57%	Removed Title
45.64%	37.12%	Removed Description
54.76%	38.28%	Removed TaskTags
61.46%	74.47%	Removed ProjectTags
49.01%	41.18%	Removed Category
61.46%	74.47%	Removed Project
77.78%	85%	Removed Project and Project Tags
71.67%	54%	Removed Category, Process, Project, Project Tags and Task Tags
13.51%	22.69%	Removed Category, Process, Project, Title and Description

According to the model proposed by Wohlin et al. [26], the **internal threats** to the validity of this evaluation are the small sample size, which can produce distortions in the expected result and conclusions. A strategy to mitigate this problem can include the use of more extensive or various samples. The lack of another sample can also be characterized as an **external threat** because it can jeopardize the generalization of the study results. As **construction threats**, we can cite the

selection of methods and definitions of measures, which was performed according to the related work, but with no other indication of exactitude or representativeness. Finally, the fact that the same developers can be involved in the study might indicate **conclusion threats**, given the possibility the text of project tasks can be standardized, in case they were written by the same person.

IV. CONCLUSIONS

Identifying similar project tasks during software development could decrease the curation effort of developers, when in need of external support. We performed a study to identify project task context elements, propose an implementation of a process model and evaluate the proposal. The implementation aims to verify if similar project tasks are associated with the same SO posts curated by developers. Precision and accuracy were 71.60% and 77.78%, respectively. We also conclude that there is a significant variation of precision and accuracy when project task context elements are added or removed. When considering the project task context elements initially identified in Section II, precision and accuracy are the highest among all combinations. Results also indicate that the hypothesis of the evaluation Project task similarity can provide useful suggestions of curated SO posts is correct, as the prediction and accuracy are as high as 70%. The model built in RapidMiner can be used with other datasets with similar characteristics, indicating the study is replicable.

Future work might involve the development of a recommendation tool using the proposed strategy, allowing the incorporation of rating mechanisms for given suggestions. The tool was not developed in this work, as we intended to focus on the study of the reuse of curated SO posts using task similarity. A deeper understanding of the role of project task context elements should be pursued, allowing the definition of weights for specific contexts.

ACKNOWLEDGMENT

The authors thank the Natural Sciences and Engineering Research Council of Canada (NSERC), the Emerging Leaders in the Americas Program (ELAP) and MITACS.

REFERENCES

- [1] C. Di Ciccio, A. Marrella, and A. Russo, "Knowledge-intensive processes: characteristics, requirements and analysis of contemporary approaches," *Journal on Data Semantics*, vol. 4, no. 1, pp. 29–57, 2015.
- [2] S. Vasanthapriyan, J. Tian, and J. Xiang, "A survey on knowledge management in software engineering," in *Software Quality, Reliability and Security-Companion (QRS-C), 2015 IEEE International Conference on*. IEEE, 2015, pp. 237–244.
- [3] R. Minelli, A. Mocchi, and M. Lanza, "I know what you did last summer - an investigation of how developers spend their time." Piscataway: The Institute of Electrical and Electronics Engineers, Inc. (IEEE), May 1, 2015, p. 25.
- [4] A. N. Meyer, L. E. Barton, G. C. Murphy, T. Zimmermann, and T. Fritz, "The work life of developers: Activities, switches and perceived productivity," *IEEE Transactions on Software Engineering*, vol. 43, no. 12, pp. 1178–1193, 2017.
- [5] L. Ponzanelli, A. Bacchelli, and M. Lanza, "Leveraging crowd knowledge for software comprehension and development." IEEE, Mar 2013, pp. 57–66.
- [6] L. Mamykina, B. Manoim, M. Mittal, G. Hripcsak, and B. Hartmann, "Design lessons from the fastest qa site in the west." ACM, May 7, 2011, pp. 2857–2866.
- [7] S. Fumin, W. Xu, S. Hailong, and L. Xudong, "Recommendflow: Use topic model to automatically recommend stack overflow qa in ide," in *International Conference on Collaborative Computing: Networking, Applications and Worksharing*. Springer, 2016, pp. 521–526.
- [8] X. Liu, B. Shen, H. Zhong, and J. Zhu, "ExpSol: Recommending online threads for exception-related bug reports," in *2016 23rd Asia-Pacific Software Engineering Conference (APSEC)*. IEEE, 2016, pp. 25–32.
- [9] T. P. Sahu, N. K. Nagwani, and S. Verma, "An empirical analysis on reducing open source software development tasks using stack overflow," *Indian Journal of Science and Technology*, vol. 9, no. 21, 2016.
- [10] T. Wang, G. Yin, H. Wang, C. Yang, and P. Zou, "Linking stack overflow to issue tracker for issue resolution." ACM, Nov 17, 2014, pp. 11–14.
- [11] L. Ponzanelli, A. Bacchelli, and M. Lanza, "Seahawk: Stack overflow in the ide." IEEE Press, May 18, 2013, pp. 1295–1298.
- [12] M. Rahman and C. K. Roy, "Surfclipse: Context-aware meta-search in the ide," in *2014 IEEE International Conference on Software Maintenance and Evolution*, 2014, pp. 617–620, iD: 1.
- [13] D. Correa and A. Sureka, "Integrating issue tracking systems with community-based question and answering websites," in *2013 22nd Australian Software Engineering Conference*. IEEE, 2013, pp. 88–96.
- [14] L. Ponzanelli, G. Bavota, M. Di Penta, R. Oliveto, and M. Lanza, "Mining stackoverflow to turn the ide into a self-confident programming prompter." ACM, May 31, 2014, pp. 102–111.
- [15] P. Kochhar, "Mining testing questions on stack overflow," in *Proceedings of the 5th International Workshop on Software Mining*. ACM, 2016, pp. 32–38.
- [16] G. Melo, U. Telemaco, T. Oliveira, P. Alencar, and D. Cowan, "Towards using task similarity to recommend stack overflow posts," in *Avances en Ingenieria de Software a Nivel Iberoamericano, CibSE 2018*, 2018, pp. 199–211.
- [17] V. Garousi, M. Felderer, and M. Mäntylä, "The need for multivocal literature reviews in software engineering: complementing systematic literature reviews with grey literature," in *Proceedings of the 20th international conference on evaluation and assessment in software engineering*. ACM, 2016, p. 26.
- [18] M. Lindvall and I. Rus, *Knowledge management for software organizations*, ser. Managing software engineering knowledge. Springer, 2003, pp. 73–94.
- [19] F. M. Santoro, P. Brézillon, and R. M. De Araujo, "Context dynamics in software engineering process," in *International Conference on Computer Supported Cooperative Work in Design*. Springer, 2006, pp. 377–388.
- [20] N. Schlitter, J. Lässig, S. Fischer, and I. Mierswa, "Distributed data analytics using rapidminer and boinc," in *Proceedings of the 4th Rapid-Miner Community Meeting and Conference (RCOMM 2013)*, 2013, pp. 81–95.
- [21] P. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining*, pearson internat. ed. ed. Boston ; Munich [u.a.]: Pearson Addison Wesley, 2006.
- [22] G. Melo, "Retrieving curated stack overflow posts of similar project tasks," Master's thesis, Universidade Federal do Rio de Janeiro, 2018. [Online]. Available: <https://www.cos.ufrj.br/index.php/pt-BR/publicacoes-pesquisa/details/15/2881>
- [23] G. Shani and A. Gunawardana, *Evaluating Recommendation Systems*, 1st ed., ser. Recommender Systems Handbook. Boston, MA: Springer US, 2011, pp. 257–297.
- [24] M. Rahman, S. Yeasmin, and C. K. Roy, "Towards a context-aware ide-based meta search engine for recommendation about programming errors and exceptions." IEEE, Feb 2014, pp. 194–203.
- [25] T. Wang, G. Yin, H. Wang, C. Yang, and P. Zou, "Automatic knowledge sharing across communities: a case study on android issue tracker and stack overflow," in *Service-Oriented System Engineering (SOSE), 2015 IEEE Symposium on*. IEEE, 2015, pp. 107–116.
- [26] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science Business Media, 2012.