Towards the Use of Machine Learning Algorithms to Enhance the Effectiveness of Search Strings in Secondary Studies

Leonardo Cairo
Universidade Salvador (UNIFACS)
Salvador, Bahia, Brazil
leocairos@gmail.com

Glauco de F. Carneiro
Universidade Salvador (UNIFACS)
Salvador, Bahia, Brazil
glauco.carneiro@unifacs.br

Miguel P. Monteiro
Universidade Nova de Lisboa (UNL)
Lisbon, Portugal
mtpm@fct.unl.pt

Fernando Brito e Abreu
Instituto Universitario de Lisboa /ISCTE-IUL
Lisbon, Portugal
fba@iscte-iul.pt

ABSTRACT

Devising an appropriate Search String for a secondary study is not a trivial task and identifying suitable keywords has been reported in the literature as a difficulty faced by researchers. A poorly chosen Search String may compromise the quality of the secondary study, by missing relevant studies or leading to overwork in subsequent steps of the secondary study, in case irrelevant studies are selected. In this paper, we propose an approach for the creation and calibration of a Search String. We chose three published systematic literature reviews (SLRs) from `Scopus` and applied Machine Learning algorithms to create the corresponding Search Strings to be used in the SLRs. Comparison of results obtained with those published in previous SLRs, show an increase of recall of revisions by up to 12%, with no loss of recall. To motivate future studies and replications, the tool implementing the proposed approach is available in a public repository, along with the dataset used in this paper.

KEYWORDS

secondary studies, machine learning, natural language processing

1 INTRODUCTION

The volume of empirical research in Computer Science is constantly expanding and secondary studies play an important role, becoming an essential reference for any researcher who wants to keep up to date. In this scenario, the decision of which Search String (SS) to use in secondary studies has been shown to be a non-trivial task. The identification of keywords and the rationale behind their arrangement in a SS is neither the hardest step, nor should it be considered the most time consuming. Nevertheless, the quality of a SS will undoubtedly impact the outcome of a secondary study.

In this paper, we propose an approach based on *Text Mining* (TM) and *Machine Learning* (ML) techniques to assist researchers in the creation of a SS. The goal is to suggest terms and their respective arrangements in the scope of a SS, to yield effective search results. The literature provides guidelines to conduct secondary studies, with slightly different suggestions on the number and order of activities.

Kitchenham and Charters summarize the important steps in three main phases [5]: *Review Planning*, *Review Execution* and *Review Reporting*. The definition of the SS is part of the *development of the review protocol* required in the *Review Planning* phase. The same author lists two ways to define a SS in secondary studies [4]: (1) using a subjective approach based on domain knowledge and on experimenter's experience and (2) using an objective elicitation approach, based on Quasi-Gold Standard (QGS) [9] terms, supported by one or more text analysis tools. In this paper, we present an approach following the second principle, based on Text Mining and ML resources, to define an effective SS.

The rest of this paper is organized as follows. Section 2 presents the proposed approach. Section 3 describes the exploratory study conducted to evaluate the proposal and section 4 presents the results. Section 5 summarizes the contributions and outlines some opportunities for future work.

2 PROPOSED APPROACH

Based on the findings of the SLR and following the guidelines of Kitchenham et al. [4], the algorithms *TF-IDF*, *CBOW* and *Skip-Gram* were chosen to define the technique used in the creation and calibration step of the SS. *TF-IDF* (Term Frequency-Inverse Document Frequency) is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. By means of *TF-IDF*, knowledge of the research domain is expanded and improved. The *CBOW* (Continuous Bag of Words) model is a representation of a text as the bag (multiset) of its words, disregarding grammar and even word order, but keeping multiplicity [10]. *Skip-Gram* models [3] are a generalization of n-grams, in which the components (typically words) need not be consecutive in the text under consideration, but may leave gaps that are skipped over.

Figure 1 provides an overview of the process carried out by our proposal. The input is the set of primary studies returned by the SS which, after processing, generates the set of terms and their correlates as output. In the following paragraphs, we provide a short description of the main parts of the figure. Note it is not intended to fully cover all stages of an SLR with search in several repositories. Its purpose is to serve as a proof of concept for the proposed approach. For this purpose, a single well-known repository with an API for querying through the SS is sufficient. Although the proposal can be applied in other repositories, this study focused exclusively on `Scopus`. Once deployed to more than one repository, the issue of duplicate articles and the rules for creating the SS in each repository will have to be addressed. Dealing with multiple repositories is

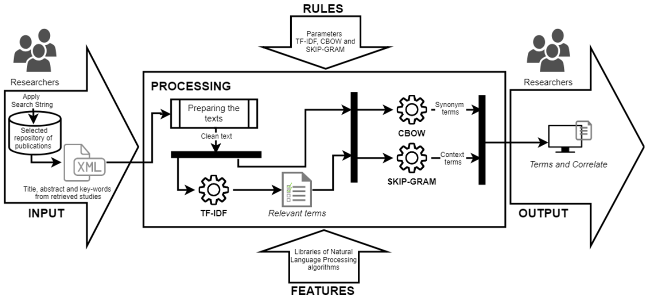

Figure 1: Proposed Approach.

left for future work. We selected `Scopus` due to its robust API, that provides data in an easy handling format. `Scopus` also has a large repository of publications and is used in many Systematic Reviews/Mappings. According to [1], `Scopus` is among the top 7 positions in the ranking of academic search engines. Such ranking takes into account available resources and data reliability.

We developed a tool to support the execution of the proposed approach [1]. It should be noted that the tool's aim is not to fully cover all stages of an SLR with search in multiple repositories, but rather to serve as a proof of concept for the proposed theory. As such, a single repository with a well documented API, such as the one of `Scopus` is sufficient for our study.

According to [1], `Scopus` is among the best in the ranking of search engines for papers published in conferences and journals. Such ranking takes into account the availability of its resources and reliability of the data. For this reason, we decided to use `Scopus`. Moreover, its robust public API provides data from the target studies to enable the analysis we propose in this paper. Another relevant reason for the use of factor was its popularity in the use for Systematic Reviews/Mappings and the large repository of publications it provides. We developed the tool using `Python` aiming to perform analysis of all the references returned from a SS within `Scopus`. The analysis yields a list of words that characterizes all the returned studies. It also suggests words associated to the same context or that are synonyms of the terms present in the SB (through the use of the *Skip-Gram* and *CBOW* of *FastText*, Facebook's ML library).

3 EXPLORATORY STUDY

The goal of this exploratory study is to evaluate the proposal using ML techniques to support creation of SSs in Secondary Computer Studies. To evaluate the proposal, an exploratory study involving 3 SLRs was conducted in `Scopus`. The choice of the exploratory study model was supported by the review findings, being in the model without human interaction. Such choice is independent of prior knowledge of researchers, which could otherwise compromise the impartiality of results. Therefore, the Research Questions (RQ) for this exploratory study are: **RQ1:** *To what extent does the proposed technique support researchers in the definition and refinement of the SS in the context of Secondary Studies?* **RQ1.1:** *Are the suggested terms relevant to the refinement of the SS within the scope of the researcher's review?* **RQ1.2:** *What is the amount of relevant work recovered by the proposed SS compared to the original SS (Recall)?* **RQ2:** *Does the proposed technique contribute to the reduction of effort in the primary study selection activity in the SLR process?* **RQ2.1:** *Do suggested terms improve SS results?* **RQ2.2:** *How much of the irrelevant work missed by the proposed SS compared to the original SS (workload)?*

Metrics for Evaluation. The first metric is the well-known *Recall* [7]. Next, *Workload* is a metric often used to measure workload in Systematic Reviews. Its main objective is to evaluate the reduction of the author's workload. In this work, Workload is evaluated based on the total number of studies that the Author would need to evaluate if using this strategy, versus total amount of work in the original study (e.g., in SS1.5 original work retrieved 112 studies whereas with the proposed approach 106–95% Workload was recovered, i.e., a 5% reduction in workload).

Selected SLRs. In the first stage, an automatic search was performed in `Scopus` to select the revisions that will be replicated using the proposed technique. To carry it out, 3 articles were selected based on the following activities: *Execute search in repository:* automatic search on `Scopus` (performed in February 2018) with the following SS: TITLE-ABS-KEY ("systematic review" OR "systematic literature review"). *Filter results:* Apply filter in the automatic search result to consider just works in Computer Science; publications from 2014; which are Systematic Review/Mapping. *Order results:* Sort results already filtered by publication date (most recent first), followed by number of citations (most cited first). These criteria aim to make results in a replication using the SS are as close as possible to the ones obtained at the time of the original search (there may have been updates in the database). *Record SLRs:* Record the first 3 revisions that meet the following requirements for replication with the technique and tool: (a) Review that used `Scopus` as search repository, once the tool uses it. (b) Relevant publications obtained (title, author, etc.). With the list of publications it is possible to replicate the original classification and thus ensure there is no divergence in selection criteria. (c) When reusing the SS on the current date (same criteria and publication date), the result is consistent with that of the original (similar numbers from `Scopus`). This criterion aims to minimize impact of divergences between works available in `Scopus` on the original study's date versus current date. (d) Result in `Scopus` equal to or less than 1000 publications, since the `Scopus` API is limited that way. The first stage should yield the selected studies and their official data: S1 [8], S2 [6], S3 [2]).

Update Studies Returned by the Original Search String. The second stage is the replication of the selected studies from the first stage. The aim is to update the original quantitative result based on the date of the exploratory study. One possible problem, is that new studies may be returned as a result of applying the SS. Updated results of each of the studies (S1, S2 and S3) were made to conform with the limits of the tool (exclusive search in `Scopus`), to enable a fairer and more accurate comparison. It is necessary to record the total number of primary studies retrieved, not just selected studies. These correspond to the selected studies and discarded studies (not originally selected). Once all studies are retrieved, the workload is computed as well as the total count of selected studies, to yield the proposed approach's Recall. We applied the SS in `Scopus` (February 2018). For example, S2 performed its search in February 2015, so replication applied the filter for publications available up to this date.

[1] https://github.com/slrmltool2019/slrmltool2019

For example, S2 originally found 30 relevant works considering all repositories, 24 of which were found in Scopus. Thus, official and updated result for S2 was a set of 368 studies retrieved of which only 24 were considered relevant (Selected/included).

Aiming at summarizing the official results of the selected systematic literature reviews, we decided to provide the following information. In the case of S1, the authors applied the following original SS: TITLE-ABS-KEY (("agile" OR "scrum" OR "kanban" OR "extreme programming" OR "lean") AND ("hci" OR "hmi" OR "ucd" OR "usability" OR "human" OR "user") AND ("requirements engineering")) AND PUBYEAR > 1994 AND PUBYEAR < 2016. Considering data from Scopus, the authors considered 27 studies as relevant from a total of 112 obtained through the original search string, being excluded 106 studies and included 6. In the case of S2, the authors applied the following original Search String: TITLE-ABS-KEY (("continuous integration" OR "continuous delivery" OR "continuous deployment") AND "software") AND PUBYEAR < 2016.

Considering data from Scopus, the authors considered 30 studies as relevant from a total of 368 obtained through the original search string, being excluded 344 studies and included 24. In the case of **S3**, the authors applied the following original SS: TITLE-ABS-KEY (("recommendation systems" OR "recommendation system" OR "recommender systems" OR "recommender system") AND ("software development" OR "software developer" OR "software engineering")) AND PUBYEAR < 2014. Considering data from Scopus, the authors considered 46 studies as relevant from a total of 264 from the original search string, being excluded 277 studies and included 37.

To conduct the replication of each study, the following activities were performed (all using the developed tool interface) *Execute original SS* - Run the author's original SS inside the tool. The SS was adapted only to restrict the search period. Such restriction aims at making the maximum result compatible with the period in which the author has reviewed it. *Identify selected primary studies* - Within the tool, already with all the works listed by the execution of the automatic search with the author's SS, the studies were classified as "Include" (relevant to the original author) and "Exclude" (irrelevant to the original author). It should be noted here that the classification was exclusively based on the list of studies that the original author listed as relevant, i.e., if the study is in the list of selected author, it is considered as "Include" and if it is not, it is classified as "Exclude". *Record updated results* - In this activity the updated record of the original author's review result was performed. At the end of this step, the updated results of each study were obtained and the tool was already prepared and fed with the original terms used in the author's SS, as well as the official classification in each of the reviews.

Execution of the Proposed Approach. In this stage, the application of the proposed technique on the original revision of each study (S1, S2 and S3) was performed. It is divided into 4 activities and one decision as follows.

Process the SS terms - The developed tool, after performing the search in the Scopus repository, processes all the returned studies generating information that allows the author to produce useful knowledge for the SS creation/calibration.

Evaluate the need to process a new SS - At each processing of a SS, the author must analyze whether the suggested terms are relevant or not for the execution of a new SS. In this exploratory study, all the unique/exclusive terms (only present in only one of the 2 sets - "Include" and "Exclude") proposed by the *TF-IDF* were tested so that the result was not compromised by the experimenter's experience. At that point, the assessment of the need to process a new SS was conditioned by the existence of unique terms that have been suggested and have not yet been processed.

Perform a new search with suggested terms - Having evaluated as relevant the suggested term and as necessary a new search, the author must insert the term in the SS and with that execute a new query in the repository. In this exploratory study, the following sequence of inclusion of the terms in SS was chosen: (a) *TF-IDF* exclusive "Include" term (not present in the *TF-IDF* "Exclude" set). In order to extend the domain of the review (*CBOW*) with possible inclusion of new studies and at the same time refine (*Skip-Gram*) the results. (a.1) Include the unique term of the unique "Include" *TF-IDF* in the appropriate grouping with the "OR" operator. Example: In SS1.1, the term "user" appears only in the TF-Include set and is already present in the author's original SS. If the term was not present it would be appended with the "OR" operator within the second group of the original SS ("hci" OR "hmi" OR "ucd" OR "usability" OR "human" OR "user" OR "single TF-IDF term")). (a.2) Include the term and its correlates (*CBOW*) in the appropriate grouping with the "OR" operator. Example: In SS1.2 along with the term "user" their correlates obtained with the *CBOW* were added with the "OR" operator. (a.3) Include the term and its correlates (*CBOW*) in the appropriate grouping with the "OR" operator and narrowing the results with their context correlates (*Skip-Gram*) with the "AND" and "OR" operators. Example: In SS2.5 in addition to the "status" term obtained by the *TF-IDF* "Include", its *CBOW* correlates (with the "OR" operator) and context correlates (*Skip-Gram*) were included with the operator "AND". (b) Exclusive *TF-IDF* "Exclude" term (not present in the *TF-IDF* "Include" set). In order to restrict the results, eliminating the irrelevant results at the maximum. (b.1) Include the single *TF-IDF* Exclude term as exclusion criterion with the "AND NOT" operator. Example: In SS2.6, the "AND NOT" operator was added along with the exclusive *TF-IDF* Exclude "cloud" in SS. (b.2) Include the term and its correlates (*CBOW*) as exclusion criteria with operator "AND NOT" and "OR". Example: In SS2.7, in addition to the term "cloud" its correlates "services", "loops" and "lifecycles" joined by the "OR" operator. (b.3) Include the term and its correlates (*CBOW*) as an exclusion criterion with "AND NOT" and "OR" operator, tapering the results with context correlates (*Skip-Gram*) with the "AND" operator and "OR". Example: In SS2.8, correlates (*CBOW*) have been included together with the term "cloud" ("cloud" OR "loops" OR "lifecycle") and the interaction of the clustered context terms ("services" OR "paas" OR "architecting").

Record a new result - After executing each SS, a log is generated with the results obtained. The SS executed and the total of relevant and irrelevant studies obtained with the SS are present in this log.

Analyze results - Once the relevant suggestions have been exhausted and the results recorded in the previous activity have been completed, the analyzes were carried out and the comparative results were compared with the original result of each replicated review. Where the Recall (number of relevant studies obtained with the proposed SS against the Official SS) and the WorkLoad (workload generated, that is, the number of total SS studies proposed against the Official SS).

Regarding the terms and correlates selected in this exploratory study for each of the revisions, we observed that for S1, two unique terms have been identified (term present in only one of the "Include"/"Exclude" sets): user with index of 5.079 and story with index of 8.212. For each of these terms, their correlates preceded by the letters S and C are represented. They were generated by *Skip-Gram* and *CBOW* respectively.

In Table 1 we can observe that although the *TF-IDF* disregards the Stop-Words, all words, including Stop-Words in S1 and "toward" in S3, were considered in the processing of *CBOW* and *Skip-Gram*. This is because *CBOW* and *Skip-Gram* do not make restrictions on the words used or even that they be specific to English, i.e., *CBOW* and *Skip-Gram*, unlike *TF-IDF*, is not intended to identify relevant terms. Their purpose is to identify terms that correlate in some way, and with that, all terms are considered. In addition, the table 1 displays the total number of words in the result set of each study/ revision. That is, the total number of words that were obtained by merging Title, Summary, and Keywords from all studies retrieved in each of the revisions (S1, S2, and S3). This total is followed by the unique terms found by the *TF-IDF* with their score, as well as the related terms (preceded by S and C, computers by *Skip-Gram* and *CBOW* respectively). All of this information is relevant and useful for the analysis performed in the following sections.

4 EXPERIMENTAL RESULTS

Quantitative and qualitative analysis was performed based on the total number of studies retrieved (quantitative) versus the total number of relevant studies obtained (Recall, qualitative). Thus, the fewer irrelevant studies retrieved, the better the effectiveness of the technique, since it reduces the human workload in the screening stage (reading titles and abstracts). Figure 2 presents a view of the recall of each generated SS. This allowed us to analyze the performance of the proposal in three different studies, as presented in Table 1. These results show (see Figure 2) that for S3, where the number of processed words is higher, both the recall and the workload have less variation.

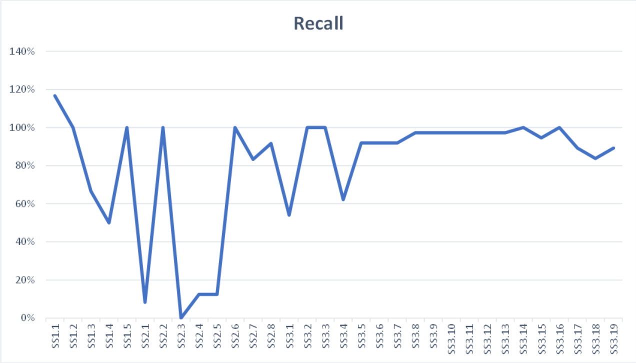

Figure 2: Recall per Search String

Table 1: List of Terms and Respective Correlates.

S	Number of Words	Terms of *TF-IDF*	Correlates
S1	17.961	user (5.079) story (8.212)	['S map', 'S scenarios', 'S users', 'C state', 'C resulted', 'C however'] ['S card', 'S cards', 'S capture', 'C stories', 'C customer', 'C related']
S2	5.949	status (4.040) cloud (8.317)	['S awareness', 'S studio', 'S teams', 'C static', 'C prioritization', 'C statistical'] ['S services', 'S paas', 'S architecting', 'C services', 'C loops', 'C lifecycle']
S3	49.343	code (7.038) service (9.652) networks (9.540) network (9.368) mobile (9.195)	['S undertaken', 'S jdt', 'S clone', 'C conducted', 'C consists', 'C main'] ['S services', 'S standardization', 'S care', 'C services', 'C attacks', 'C lattice'] ['S network', 'S networking', 'S summarization', 'C network', 'C networked', 'C professional'] ['S networks', 'S networked', 'S networking', 'C networks', 'C networked', 'C professional'] ['S mobility', 'S hoc', 'S invasion', 'C toward', 'C optimal', 'C wireless']

We combined the use of three algorithms (*TF-IDF*, *CBOW* and *Skip-Gram*) divided into 2 groups: "Include" and "Exclude", therefore resulting in 6 combinations: (a) **TF-IDF Include** - Using the TF-IDF term exclusively from the "Include" grouping (SS2.1). (b) **TF-IDF Include +** *CBOW* - Using the TF-IDF term exclusively from the "Include" grouping, jointly with the *CBOW* terms (SS2.4). (c) **TF-IDF Include + CBOW + Skip-Gram** - Using the TF-IDF term exclusively from the "Include" grouping, jointly with the *CBOW* and *Skip-Gram* terms (SS2.5). (d) **TF-IDF Exclude** - Using the TF-IDF term exclusively from the "Exclude" grouping (SS2.6). (e) **TF-IDF Exclude + CBOW** - Using the TF-IDF term exclusively from the "Exclude" grouping, jointly with the *CBOW* terms (SS2.7). (f) **TF-IDF Exclude + CBOW + Skip-Gram** - Using the TF-IDF term exclusively from the "Include" grouping, jointly with the *CBOW* and *Skip-Gram* terms (SS2.8).

Discussion of S1 Results In the analysis with the TF-IDF of study S1, it was possible to obtain two words, "user" and "story", which stood out from the others. For this study, by combining TF-IDF with *CBOW* (SS1.1) it was possible to increase the Recall of the original work by 17%. That is, in the original work, because of the SS used by the Author, a study was not captured in `Scopus`. In this same study, it was possible to obtain all the relevant studies (100% of Recall) with a reduction of 6 irrelevant studies (reduction of 5% of the workload), by combining the three algorithms (TF-IDF, *CBOW* and *Skip-Gram*). Finally, with this exploratory study, the approach proposed in this work was shown able to expand the Recall, that is, to improve the coverage of the original revision. And even with a set of few studies (only 112) it was still able to reduce it by 5% (SS1.5). Demonstrating its relevance to the optimization of Recall and Workload.

Discussion of S2 Results In the study of [6] (S2) where in its partial reproduction were identified 368 results in Scopus with a total of approximately 6 thousand words (1/3 of the amount of words obtained in S1). TF-IDF also extracted two words that stood out: "status" and "cloud". The combination of these words featured 8 SSs that reached 100% Recall with up to 12% workload reduction (SS2.6). However, there was a SS that can not be assessed (SS2.3). When applied in the Scopus repository, this SS generated more than 17 thousand results that exceeded the limits of the API, tool and hardware used. Anyway, the combination of the three algorithms had a best result (SS2.8) a Recall of 92% with a reduction of 14% of the workload, meaning that with 1/3 of the words of S1 but with 3 times more studies returned, the approach proposed in this work proved even more relevant and promising for the reduction of the workload of the reviewer. Thus, it was still possible to reduce the author's workload by 14% in this restricted scenario.

Discussion of S3 Results In the study of [2] (S3) 5 words were extracted ("code", "network","network" and "mobile") the 49 thousand more present in the group. The combination of these 5 words generated 19 SS that were able to obtain 100% Recall in 4 of them (SS3.2, SS3.3, SS3.14 and SS3.16) and with a reduction of up to 81% of workload SS3.1). The results for this study were shown to be quite stable, that is, with little variation in Recall and workload. Such stability may be related to the greater number of words existing in relation to the other studies. The lowest workload (91%) with maximum Recall (100%) was obtained by SS3.14. While the best result was SS3.9 and SS3.12, both with Recall of 97% and workload of 86%. Finally, in S3 the approach proposed by this work was shown to be even more stable and coherent through the results obtained. Finalizing this way the cycle of tests with very promising results.

5 CONCLUSIONS

From the discussions of S1, S2 and S3 results, the proposed strategy allows the user to decide on the combination of algorithms that brings the best result for the selection of the primary studies. A new approach was proposed for the creation and calibration of the Search String in SLRs. We facilitated its application by making available a supporting tool to the entire community. Three case studies were used to raise evidence on the validity of the proposed approach. In view of the promising results and the possibility of advancing this line of research, the following future work options were identified: (1) simultaneous processing of multiple repositories, (2) processing in a more performing environment (more specialized and with greater processing power) and (3) evaluation of the proposal with human interaction (controlled experiment).

REFERENCES

Diego Buchinger, Gustavo Andriolli De Siqueira Cavalcanti, and Marcelo Da Silva Hounsell. 2014. Mecanismos de busca acadêmica: uma análise quantitativa. *Revista Brasileira de Computação Aplicada* 6, 1 (2014), 108–120. https://doi.org/10.5335/rbca.2014.3452

Marko Gasparic and Andrea Janes. 2016. What recommendation systems for software engineering recommend: A systematic literature review. *Journal of Systems and Software* 113 (2016), 101–113. https://doi.org/10.1016/j.jss.2015.11.036

David Guthrie, Ben Allison, Wei Liu, Louise Guthrie, and Yorick Wilks. 2006. A closer look at skip-gram modelling. In *Proceedings of the 5th international conference on language resources and evaluation (LRECâĂŹ06)*. European Language Resources Association (ELRA), Genoa, Italy, 1222–1225.

Barbara Kitchenham, David Budgen, and O. Pearl Brereton. 2015. *Evidence-Based Software Engineering and Systematic Reviews*. Vol. 4. CRC press, Boca Raton, Florida, USA.

B Kitchenham and S Charters. 2007. *Guidelines for performing systematic literature reviews in software engineering*. Technical Report. Technical report, ver. 2.3 ebse technical report. ebse.

Eero Laukkanen, Juha Itkonen, and Casper Lassenius. 2017. Problems, causes and solutions when adopting continuous delivery: A systematic literature review. *Information and Software Technology* 82 (2017), 55–79. https://doi.org/10.1016/j.infsof.2016.10.001

Rasmus Ros, Elizabeth Bjarnason, and Per Runeson. 2017. A Machine Learning Approach for Semi-Automated Search and Selection in Literature Studies. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering - EASE'17*. ACM, New York, NY, USA, 118–127. https://doi.org/10.1145/3084226.3084243

Eva Maria Schön, Jörg Thomaschewski, and María José Escalona. 2017. Agile Requirements Engineering: A systematic literature review. *Computer Standards and Interfaces* 49 (2017), 79–91. https://doi.org/10.1016/j.csi.2016.08.011

He Zhang, Muhammad Ali Babar, and Paolo Tell. 2011. Identifying relevant studies in software engineering. *Information and Software Technology* 53, 6 (2011), 625–637. https://doi.org/10.1016/j.infsof.2010.12.010 arXiv:gr-qc/0208024

Yin Zhang, Rong Jin, and Zhi-Hua Zhou. 2010. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics* 1, 1-4 (2010), 43–52.