Please cite the Published Version

Heuristics for improving the rigour and relevance of grey literature searches for software engineering research
PREPRINT

Austen Rainer

School of Electronics, Electrical Engineering and Computer Science

Queen's University Belfast, UK

a.rainer@qub.ac.uk

Ashley Williams

Department of Computer Science and Software Engineering

University of Canterbury, NZ

ashley.williams@pg.canterbury.ac.nz

2018

Background: Software engineering research has a growing interest in grey literature (GL). Aim: To improve the identification of relevant and rigorous GL. Method: We develop and demonstrate heuristics to find more relevant and rigorous GL. The heuristics generate stratified samples of search and post–search datasets using a formally structured set of search keywords. Conclusion: The heuristics require further evaluation. We are developing a tool to implement the heuristics.

Keywords: grey literature review, search engines, reasoning, quality criteria

Contents

List of Figures

List of Tables

1 Introduction

There is increasing interest in software engineering research in the value of online grey literature. For example, Garousi and his colleagues [2] have investigated the development of multi-vocal literature reviews (MLRs) to incorporate grey literature into systematic literature reviews (SLRs).

Using existing online search engines to search for and select the better–quality grey information is challenging as such engines can return irrelevant articles (false positives) or not return relevant articles (false negatives). These challenges impact the effectiveness and efficiency of conducting online searches e.g. as part of a MLR or a Grey Literature Review (GLR) or a Rapid Review [3, 4]. Search engines:

- typically use a keyword–based search query, and such queries do not necessarily allow for finer-grained searching, or for more sophisticated lexical features such as grammatical structures;

- are likely to optimise the search results to the searcher, based on the search engine's history of prior searches by that searcher; and

- are likely to maintain their own topic models to determine relevance of results e.g. whether an online article relates to software testing

Additionally, the size of the World Wide Web is unknown (even unknowable) and therefore it is difficult to assess the extent to which the search results are representative of a wider online 'population' of results.

Whilst keyword–based online search engines are challenging to use for systematic grey literature searches, they are also the only general–purpose search mechanism currently available to conduct such searches. We are therefore looking for ways to help researchers improve the effectiveness of their online searches using existing search engines.

In this paper, we propose and briefly demonstrate heuristics to improve the relevance and rigour of grey literature searches for software engineering research. In essence, our heuristics promote stratified positive and negative sampling and subsequent filtering based on topic keywords and quality–criteria keywords. Whilst the heuristics have contributed to recent research [5] they are explicitly discussed for the first time in the current paper.

2 Overview to the heuristics

To provide an overview to our heuristics we enumerate their elements:

1. The heuristics assumes the researcher already has a set of topic–related keywords that she or he intends to use for online searches. For example, Garousi *et al.* [6] conducted a MLR of when and what to automate in software testing. One of their search strings was <decision automated software testing>.

2. In addition to keywords, search engines sometimes allow configuration of advanced search settings e.g. restricting the searches to a date range, a file type, or a natural language. These configuration options can be used to implement some inclusion and exclusion criteria e.g. to only search for PDF documents written in English and dated in the years 2008 – 2018. Other exclusion and inclusion criteria must often be applied after the searches have been conducted, but of course the researcher can only then exclude or include on the basis of the results of the search(es).

3. We propose that, where appropriate, the researcher distinguishes types of topic keywords. With the Garousi *et al.* [6] example, the researchers could potentially distinguish between the more generic topic of automated software testing and the more specific topic of decision–making related to automated software testing, and could construct and manage two distinct sets of keywords for each of these types. As another example, one could distinguish between a topic (such as software testing) and some perspective on that topic e.g. the experience of practitioners (suggesting keywords for experience) or between topic and empirical studies relating to that topic (suggesting keywords for empirical studies). Again, in these contrasting examples, the researcher can manage these types of keywords separately.

4. We also propose that the researcher consider introducing (some) quality criteria as additional keywords to be used in the keyword–based searches. Although the constraints of keyword–based search engines limit the quality criteria that can be implemented with keywords, there are still some quality criteria that can be considered.

   a) One type of quality criteria relates to reasoning, and there is therefore the opportunity to include *reasoning indicators* as keywords in searches. Our previous research [7] has identified a number of reasoning indicators with high precision but low recall.

   b) A second type of quality criteria that could be implemented as keywords relates to *identity* e.g. keywords based on particular institutions or individuals that were associated with high(er) quality articles in a specified context. We recognise that there are validity threats to choosing particular institutions or individuals. Discussion of these threats is beyond the scope of this short paper, but see Garousi *et al.*'s inclusion of *authority* in their quality assessment checklist [8] and also the AACODS checklist [9].

5. In addition to proposing the combination of topic–oriented and quality–oriented keywords, we also propose that researchers:

   a) conduct a stratified sampling of searches based on set–theoretic combinations of these keywords (see Table 2); and further

   b) conduct searches that include *negative* searches i.e. searching for the negation of a keyword.

   Effectively, the researcher is conducting stratified positive and negative sampling based on topic keywords and quality–criteria keywords.

6. Following the conduct of stratified searches, the researcher then has stratified samples of search *results*. These samples then allow the researcher to investigate the properties of the samples, and to compare these samples using measures relevant to the research being conducted e.g. distance metrics or information metrics. (As an aside, the stratified samples can potentially be used as datasets suitable for machine classifiers.)

7. We recognised earlier that some quality–criteria are not suitable for implementation as keywords. One example is quality of writing, another example is presence of citations. These kinds of quality-criteria can be applied to the post–search samples e.g. to examine the quality of writing, or the presence of citations in each of the samples returned from the search results. The logic of stratified sampling still applies, with layers of search–produced samples and, within those layers, sub-layers of post-search sub–sampling.

We recognise that our proposal potentially introduces additional effort for the researcher to conduct an increased number of searches. There is the need for automation of these searches. We are developing a tool[1] to automatically conduct the stratified searches.

3 A brief demonstration of the heuristics

In previous research [5], we were interested in exploring whether practitioners cited other sources (i.e. researchers or practitioners) when they wrote online articles about their experiences of software testing. We derived three sets of keywords for the online searches:

- Keywords relating to the topic of software testing

- Keywords relating to the articulation of professional experience

- Keywords relating to the presence of reasoning

In forming our sets of keywords, we distinguish between quality (the reasoning keywords) and two aspects of topic. Our keywords are summarised in Table 1.

As standard search engines do not support the searching for articles containing citations, we were not able to search on the presence of URL citations e.g. anchor tags. Our searching of URL citations therefore occurred as a post–search process.

We used the Google Custom Search API[2] so that we could automate the Google searches. Prior research in software engineering (e.g. [10]) has tended to use manual searching of Google. To collect the data for the current study, we performed daily searches for all nine search sets over a continuous four-week (28–day) period between October and November 2017. The Google Custom Search API places a limit of 100 free

---

[1]https://github.com/zedrem/coast
[2]https://developers.google.com/custom-search/

searches per day. We therefore executed 10 searches per search set per day with nine independent search engines. Each search returns 10 pages of results. Each page contains 10 results. We ran the queries 10 times a day to attempt to smooth the (proprietary) variation of results from the Google Custom Search API. Overall we retrieve 1,000 results per day for each set.

Our reasoning indicators were derived from a review of prior research. There is little prior research on searching for experience online (but see [11, 12]), so we constructed a basic set of keywords to search for experience. The validity of the experience keywords is not central to the demonstration of our heuristics here: we use the keywords as a proof of concept.

Table 1: Keywords for topic, reasoning and experience

Criteria	Keywords
Topic	software AND testing
Reasoning	but, because, for example, due to, first of all, however, as a result, since, reason, therefore
Experience	i, me, we, us, my, experience, experiences, experienced, our

We structured our searches, using set theory, to ensure the full coverage of combinations of keywords. These searches are summarised in Table 2.

Search sets S1 and S9 are special cases. Formally, search set S1 contains the universe of (other, potential) online content and should therefore be included for completeness of evaluation. Practically, we do not have the resources to adequately search the universe of online content (or even Google's indices of the universe of online content). We anticipate that we would have in S1 a sparse and unpredictable dataset. Accepting these constraints, we constructed a random sample of search queries (with query length between two and five keywords) for search S1. We then complemented search set S1 with a more constrained search set, S9. Search set S9 is defined as the set of all articles relating to "software engineering" excluding those articles referring to "testing".

Ideally, we want the search engine to find online content that contains reasoning and experience relating to software testing. The search resulting in set S6 targets that ideal content. We conduct the other sets of searches to allow us to evaluate the quality of content in S6. For example, search S3 is intended to find online content that contains reasoning and experience, but where the content is not about software testing.

With the samples of results from the searches, we performed semi–automated analyses of the articles for the presence of citations. Table 3 presents an indicative selection of our sub-sampling. We were then able to select particular sub–samples of relevance to our research and analyse those qualitatively (see [5] for more detail). By comparing different samples, we are able to establish a relative 'size' of citations to research. For example, with S6 we see that developers cite *Developer authorities* four times as much as they cite *Research*.

Table 2: Logic for each set of searches and resulting datasets
(T=Topic; R=Reasoning; E=Experience; !=logical not)

Search set	T	R	E	!T	!R	!E
S1				•	•	•
S2		•		•		•
S3		•	•	•		
S4			•	•	•	
S5	•	•				•
S6	•	•	•			
S7	•		•		•	
S8	•				•	•
S9				○	•	•

Table 3: Percentage of source articles that cite an external URL (see [1] for further information)

Category of cited URL	Search set of citing article								
	1	2	3	4	5	6	7	8	9
Peer–reviewed research e.g. IEEE Xplore	0%	0%	1%	0%	1%	1%	0%	1%	1%
Education e.g. .edu domains	2%	2%	3%	2%	3%	3%	1%	4%	12%
Developer authorities e.g. MSDN	0%	0%	1%	0%	2%	4%	2%	1%	1%
Developer Q&A e.g. StackOverflow	0%	0%	0%	0%	0%	1%	1%	0%	0%
Repository e.g. GitHub	0%	1%	1%	0%	4%	2%	1%	2%	1%

4 Discussion and conclusion

We have proposed, and briefly demonstrated, heuristics to improve the rigour and relevance of grey literature searches that use keyword–based search engines. In essence, the heuristics generate stratified samples of search and post–search datasets using a formally structured set of search keywords (and 'negative' keywords). The heuristics therefore allow the generation of a more comprehensive set of searches and a more complete coverage of the 'search space' being queried.

With regards to relevance, we use an example briefly considered earlier in the paper to illustrate our argument. We noted earlier that one of Garousi *et al.* [6] search strings was <decision automated software testing>. The heuristics could, with this example, support the construction of a more comprehensive and detailed set of search queries relating to decision–making in automated software testing, and therefore help the researcher find a larger number of (more) relevant search results.

With regards to rigour, we again draw on Garousi *et al*, who proposed a quality checklist for MLRs [8]. The checklist suggests the basis of quality criteria to use in the post–search filtering. Using an implementation of Garousi *et al.*'s checklist in the post–search filtering should help to improve the rigour of the articles to be analysed by the researcher.

The heuristics require additional time and effort to construct, and also to apply e.g.

through conducting more searches and more complex searches. For these reasons, we are developing software tools to assist with the online searches, the download of search results, and the post–search filtering of the results. We believe the resulting stratified samples are advantageous because they give the researcher more flexibility in selecting the articles to analyse. As a result, the researcher can both *reduce* their effort, because the researcher can be more *selective* in the articles they study, and be more *effective* in their effort, because the researcher can work with the higher–quality set of articles.

Although we have used the heuristics in our research [5], we have yet to formally evaluate the heuristics. The heuristics are explicitly discussed for the first time in the current paper. We have previously evaluated the reasoning indicators [7] to identify those with high precision.

In further research, we plan to evaluate our heuristics, for example in relation to Garousi *et al.*'s quality checklist [8], and further develop software tools to implement the heuristics.

References

[1] A. Rainer, A. Williams, Technical report: Do software engineering practitioners cite research on software testing in their online articles? a structured search of grey data (April 2018).
URL https://www.researchgate.net/publication/324706645

[2] V. Garousi, M. Felderer, M. V. 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.

[3] B. Cartaxo, G. Pinto, S. Soares, Towards a model to transfer knowledge from software engineering research to practice, Information and Software Technology 97 (2018) 80 – 82. doi:https://doi.org/10.1016/j.infsof.2018.01.001.
URL http://www.sciencedirect.com/science/article/pii/S0950584918300028

[4] B. Cartaxo, G. Pinto, S. Soares, The role of rapid reviews in supporting decision-making in software engineering practice, in: Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018, EASE'18, ACM, 2018, pp. 24–34. doi:10.1145/3210459.3210462.
URL http://doi.acm.org/10.1145/3210459.3210462

[5] A. Williams, Do software engineering practitioners cite research on software testing in their online articles?: A preliminary survey., in: Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018, EASE'18, ACM, 2018, pp. 151–156. doi:10.1145/3210459.3210475.
URL http://doi.acm.org/10.1145/3210459.3210475

[6] V. Garousi, M. V. Mäntylä, When and what to automate in software testing? a multi-vocal literature review, Information and Software Technology 76 (2016) 92–117.

[7] A. Williams, Using reasoning markers to select the more rigorous software practitioners' online content when searching for grey literature, in: Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018, EASE'18, ACM, 2018, pp. 46–56. doi:10.1145/3210459.3210464.
URL http://doi.acm.org/10.1145/3210459.3210464

[8] V. Garousi, M. Felderer, M. V. Mäntylä, Guidelines for including the grey literature and conducting multivocal literature reviews in software engineering, arXiv preprint arXiv:1707.02553.

[9] J. Tyndall, Aacods (authority, accuracy, coverage, objectivity, date, significance) checklist (2010).
URL http://dspace.flinders.edu.au/dspace/

[10] O. Dieste, N. Juristo, Systematic review and aggregation of empirical studies on elicitation techniques, IEEE Transactions on Software Engineering 37 (2) (2011) 283–304.

[11] V. Jijkoun, M. de Rijke, W. Weerkamp, P. Ackermans, G. Geleijnse, Mining user experiences from online forums: an exploration, in: Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media, Association for Computational Linguistics, 2010, pp. 17–18.

[12] K. Inui, S. Abe, K. Hara, H. Morita, C. Sao, M. Eguchi, A. Sumida, K. Murakami, S. Matsuyoshi, Experience mining: Building a large-scale database of personal experiences and opinions from web documents, in: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01, IEEE Computer Society, 2008, pp. 314–321.