Generating Search Strings for Secondary Studies Using Text Mining

Leonardo Fuchs Alves

Generating Search Strings for Secondary Studies Using Text Mining[1]

Leonardo Fuchs Alves

Advisor: *Prof. Bruno Magalhães Nogueira, Ph.D.*
Co-Advisor: *Prof. Francisco José Silveira de Vasconcellos, Ph.D.*

Dissertation delivered to the Faculdade de Computação (FACOM-UFMS) as part of the necessary requirements to obtain the title of Master in Computer Science.

UFMS - Campo Grande
August/2020

[1] This work was supported by CAPES Grant No. 88882.466597/2019-01.

“The mind that opens up to a new idea never returns to its original size.”
– Albert Einstein

Acknowledgements

I would like to express my deepest thanks to my advisors, Bruno Nogueira, and Francisco Vasconcellos. Incredible people who have guided me through these years, contributing everything possible, from ideas for the research to words to keep me motivated. I am grateful for all the challenges that have been proposed to me, and for all the warm discussions that we have had, that have changed infinitely in how I observe the world around me. I thank you for being so helpful and liberal, always allowing me to give an opinion and try to aggregate in some way.

I would also like to extend my deepest gratitude to my girlfriend, Paola. I have the fullest conviction that if I managed to complete this research, it is because of it. There are countless days when she needed to listen, motivate, encourage, or even help me with something. I am incredibly grateful to have you by my side always.

I must also thank my parents, who, since I was born, has strived in a memorable way to provide me with the best education that was possible (and impossible). I also thank all the professors and technicians at the Faculty of Computing (FACOM) at UFMS for their dedication and willingness, making the Graduate Program extremely friendly. Finally, I thank CAPES for providing a scholarship that made this research possible and so rich in detail.

Abstract

A Secondary Study (SS) is an important research method used in several areas. A crucial step in the Conduction phase of an SS is the search of studies. This step is time-consuming and error-prone, mainly due to the refinement of the search string. The objective of this study is to validate the effectiveness of an automatic formulation of search strings for SS. Our approach, termed Search String Generator (SeSG), takes as input a small set of studies (as a Quasi-Gold Standard) and processes them using text mining. After that, SeSG generates search strings that deliver a high F1-Score on the Start Set of a hybrid search strategy. To achieve this objective, we (1) generate a structured textual representation of the initial set of input studies as a bag-of-words using Term Frequency and Document Frequency; (2) perform automatic topic modeling using LDA (Latent Dirichlet Allocation) and enrichment of terms with a pre-trained dense language representation (embedding) called BERT (Bidirectional Encoder Representations from Transformers); (3) formulate and evaluate the search string using the obtained terms; and (4) use the developed search strings in a digital library. For the validation of our approach, we conduct an experiment – using some SS as objects – comparing the effectiveness of automatically formulated search strings by SeSG with manual search strings reported in these studies. SeSG generates search strings that achieve better final F1-Score on the Start Set than searches reported by these SS. Our study shows that SeSG can effectively supersede the formulation of search strings, in hybrid search strategies, since it dismisses the manual string refinements.

Keywords: Search String, Text Mining, Secondary Studies, Systematic Literature Review and Systematic Mapping Study

Resumo

Estudo Secundário (ES) é um importante método de pesquisa utilizado em diversas áreas. Uma etapa crucial na fase de Condução de um ES é a busca de estudos. Esta etapa é demorada e sujeita a erros, principalmente devido ao refinamento da *string* de busca. O objetivo deste estudo é validar a eficácia de uma formulação automática de *strings* de busca para ES. Nossa abordagem, denominada *Search String Generator (SeSG)*, leva como entrada um pequeno conjunto de estudos (um *Quasi-Gold Standard*) e os processa usando mineração de texto. Depois disso, o SeSG gera *strings* de busca que fornecem um alto F1-Score do *Start Set* em estratégias de busca híbridas. Para atingir esse objetivo, (1) geramos uma representação textual estruturada do conjunto inicial de estudos de entrada como uma *bag-of-words* usando Frequência de Termos e Frequência de Documentos; (2) realizamos uma modelagem automática de tópicos utilizando LDA (*Latent Dirichlet Allocation*) e enriquecimento de termos com uma representação de linguagem densa pré-treinada (*embedding*) chamada BERT (*Bidirectional Encoder Representations from Transformers*); (3) formulamos e avaliamos a *string* de busca usando os termos obtidos; e (4) usamos as *strings* de busca desenvolvidas em uma biblioteca digital. Para a validação da nossa abordagem, conduzimos um experimento - usando alguns ES como objetos - comparando a eficácia de *strings* de busca formuladas automaticamente pelo SeSG com *strings* de busca manuais relatadas nesses estudos. O SeSG gera *strings* de busca que alcançam um melhor *F1-Score* do *Start Set* do que as pesquisas relatadas pelos ES. Nosso estudo mostra que SeSG pode substituir efetivamente a formulação de *strings* de busca, em estratégias de busca híbridas, uma vez que dispensa os refinamentos manuais da *string*.

Palavras-chave: *String* de Busca, Mineração de Texto, Estudos Secundários, Revisão Sistemática da Literatura e Estudo de Mapeamento Sistemático

Summary

List of Figures

List of Tables

List of Abbreviations

BERT Bidirectional Encoder Representations from Transformers

DF Document Frequency

GS Gold Standard

LDA Latent Dirichlet Allocation

ML Machine Learning

NLP Natural Language Processing

PS Primary Studies

QGS Quasi-Gold Standard

SE Software Engineering

SeSG Search String Generator

SLR Systematic Literature Review

SMS Systematic Mapping Study

SS Secondary Studies

TF Term Frequency

TM Text Mining

SECTION 1

Introduction

Secondary Studies (SS) are widely used in Software Engineering (SE) to synthesize the knowledge in a given research topic. This method of empirical research has a growing interest in the SE community. This interest increase as indicated by the high number of related published studies (Ali and Usman, 2018) and by the thousands of citations made to the main guides of Systematic Literature Review (SLR) in SE (Kitchenham, 2004; Kitchenham and Charters, 2007; Kitchenham et al., 2015).

An SS is usually grouped into three phases: planning, conducting and documentation. During the planning phase, the need to perform the SS is detailed, and the study protocol is created and validated. This protocol contains, for example, the research questions, the search strategy and the selection criteria of the studies. In turn, the conduction phase aims to identify relevant studies, to select Primary Studies (PS) and to extract and synthesize the data obtained, following the protocol previously formulated. Finally, in the documentation phase, the results obtained during the SS are reported to the community (Kitchenham et al., 2015).

During the conduction of the SS, different types of search strategy can be formulated in order to find relevant studies. Manual search, automated search and snowballing are commonly used search strategies. In manual search, a search is manually performed in journals and conferences. For automated search, a search string is formulated with specific terms related to the research area in question. Then, the search string is used in digital libraries to search for interesting studies. In snowballing, the references and citations of an initial list of relevant studies are analyzed, increasing this list with related studies.

Combining search strategies is also possible. Mourão et al. (2020) proposes the use of hybrid strategies (automated search followed by snowballing) to reduce the effort required during the improvement of the search strategy. This concern with effort is related to the complexity of obtaining an efficient search string. Ideally, a search string must retrieve a high number of relevant studies (high recall) with the smallest number of retrieved studies (high precision). In less structured areas of study (e.g., SE), where relevant concepts may be represented by multiple terms, manual refinement of the search string can become an extremely complicated and time-consuming activity.

In order to facilitate the creation of the search strategy in SS, some effort has been made to assist the search string refinement through Text Mining (TM) (Mergel et al., 2015; Ros et al., 2017; Grames et al., 2019). In general, these studies propose the use of an initial search string in an automated search. Documents retrieved in this initial search are presented to TM techniques in order to extract interesting search terms. These terms are then used in a manual refinement of the search string.

Unlike these approaches, in this work we propose SeSG (Search String Generator), an approach for automatically formulating search strings for SS. The main contribution of SeSG is to automatically generate and refine the search string, minimizing the researchers effort in performing SS. SeSG requires only a list of relevant studies already known (called Quasi-Gold Standard - QGS) as input and does not require an initial search string. Also, SeSG automatically assists the search string refinement by incorporating external terms using pre-trained word embeddings (Turian et al., 2010). These embeddings can find similar words that can improve the search string recall.

SeSG aims to balance precision and recall on a search, particularly in hybrid search strategies. The effectiveness of a search undermines a strict recall (full completeness) which in several topics is not necessary or even difficult to achieve (Kitchenham et al., 2015; Cooper et al., 2018).

To evaluate the effectiveness of the search using the search strings generated by SeSG, an experiment was conducted to compare the proposed approach to automated searches used in SS in the SE area. Our experiments used three different SS and SeSG achieved better F1-Score on the Start Set of a hybrid search than the manual searches.

Considering the advances obtained during this study, we can highlight some as those that can contribute in some way to the field of search strings automation. These are: (1) the investigation of different ways of representing a collection of texts, such as bag-of-words, topics formulation and embeddings; (2) the enrichment of terms present in a search string via BERT; and (3) a strategy for the formation of search strings in an automated way.

SECTION 2

Background

This section presents an overview of SS phases, detailing concepts applied in the generation of search strings. Additionally, we describe a generic TM process aiming to enlighten how it can help researchers automate the search of PS.

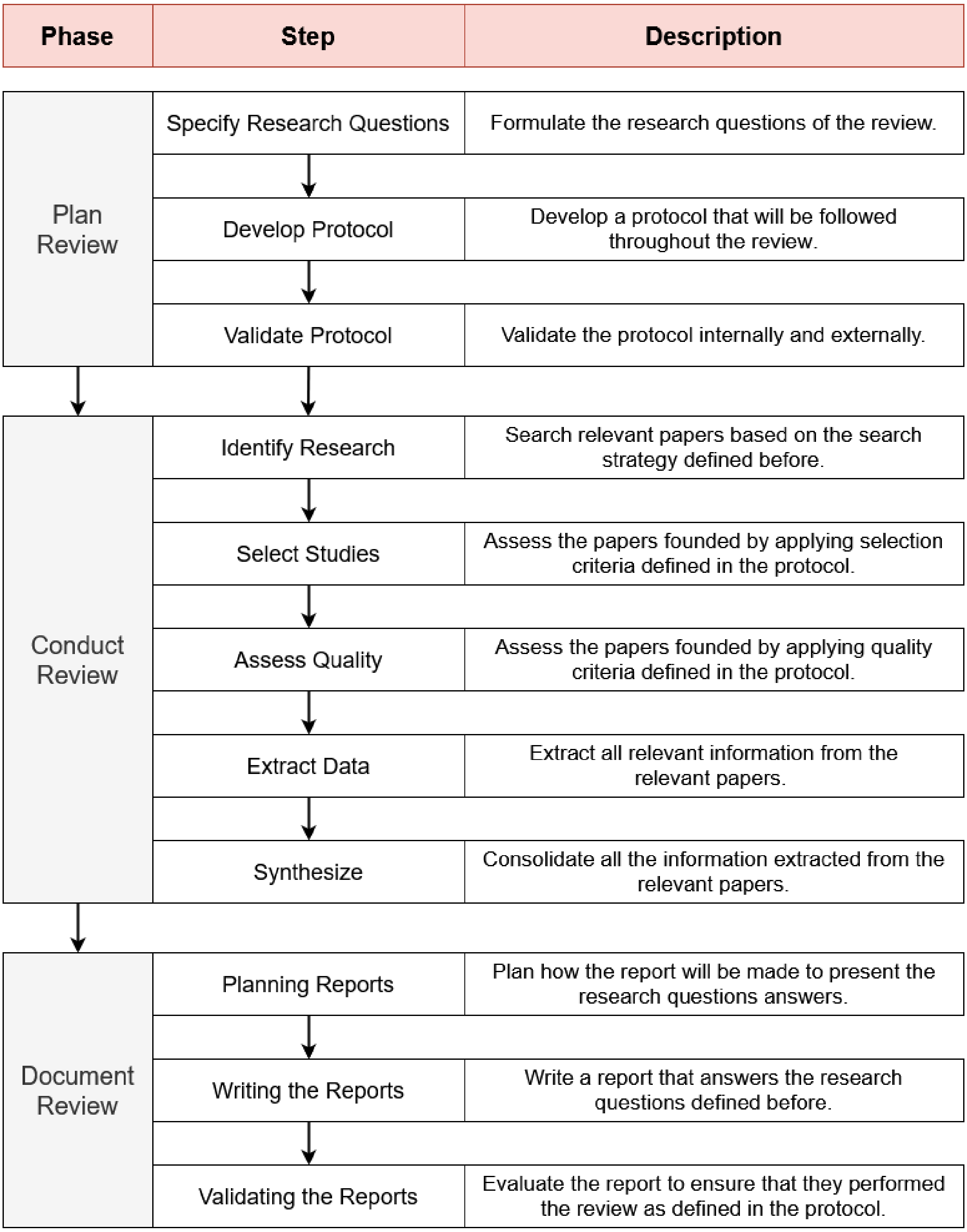
2.1 Secondary Study Process

A SS poses difficulties for experienced and novice researchers. Some guidelines have been published to assist researchers, defining phases and steps of a SS process. Kitchenham et al. (2015) divides the SS process in planning, conducting, and reporting phases, as shown in Figure 2.1.

The steps to be performed in the **Plan Review** phase are:

- **Specify Research Questions:** Research questions direct a SS, providing the foundation for decisions such as which PS will be included, which data should be extracted, and how it will be synthesized.

- **Develop Protocol:** A review protocol is a documented plan that describes the details of the execution of a SS. It is an important document that helps to reduce the likelihood of researcher bias by limiting the influence of researcher expectations. Also, a protocol can be evaluated by other researchers who may provide feedback on the design of a review before its conduction. Lastly, a protocol can aid the writing of a review report. Amongst other fundamental components, a protocol must have a documented search strategy.



Phase	Step	Description
Plan Review	Specify Research Questions	Formulate the research questions of the review.
	Develop Protocol	Develop a protocol that will be followed throughout the review.
	Validate Protocol	Validate the protocol internally and externally.
Conduct Review	Identify Research	Search relevant papers based on the search strategy defined before.
	Select Studies	Assess the papers founded by applying selection criteria defined in the protocol.
	Assess Quality	Assess the papers founded by applying quality criteria defined in the protocol.
	Extract Data	Extract all relevant information from the relevant papers.
	Synthesize	Consolidate all the information extracted from the relevant papers.
Document Review	Planning Reports	Plan how the report will be made to present the research questions answers.
	Writing the Reports	Write a report that answers the research questions defined before.
	Validating the Reports	Evaluate the report to ensure that they performed the review as defined in the protocol.

Figure 2.1: Phases and steps of the SLR Process. Adapted from Kitchenham et al. (2015).

- **Validate Protocol:** A protocol is validated internally and externally. Internal validation includes the test of aspects of the review plan, such as the search strings and the data extraction forms. Also, it is essential to evaluate the protocol by researchers who are external to a review team.

After the Plan Review phase, the SLR process continues in the **Conduct Review** phase. In this phase, the steps to perform are:

- **Identify Research:** As mentioned, an important element of a SS plan is a search strategy that finds the largest possible number of relevant PS to the research questions. A search strategy usually combines complementary search methods. A broadly used method is automated searching using digital libraries and indexing systems. Other methods are manual searching in journals and conference proceedings and snowballing (forward and backward) (Wohlin, 2014).

- **Select Studies:** Once the candidate studies are identified, their relevance to the research questions should be assessed. In this sense, researchers should employ an inclusion and exclusion criteria to filter the set of candidates. Studies that are not relevant to answer the research questions should be excluded.

- **Assess Quality:** During the formulation of the protocol, besides defining inclusion and exclusion criteria for selecting relevant studies, it is also necessary to formulate some measures to evaluate the quality of the selected PS. This assessment's main objective is to assert that the results are complete, valid, and unbiased.

- **Extract Data:** The main objective of this step is to extract the necessary information to answer the research questions.

- **Synthesize:** This step consists of synthesizing studies included in the review.

Finally, Kitchenham et al. (2015) defines the **Document Review** phase, which is responsible for documenting or reporting the study in appropriate ways for the intended audience. This phase has the following steps:

- **Planning Reports:** Planning for final reports should be initiated during the preparation of the review protocol. This step involves specifying potential audiences and deciding what type of document would suit their needs.

- **Writing the Reports:** This step is especially important since the report of a SS will contribute to other researchers. According to Kitchenham et al. (2015), a report should address traceability to provide a clear connection between the research questions and their answers. It must also address repeatability to ensure that the method is defined clearly and in sufficient detail that other researchers could replicate it.

- **Validating the Reports:** It involves internal and external assessment of the quality of reports.

As described, a search strategy is planning in the **Develop Protocol** step and applied in the **Identify Research** step. Usually, the **Validate Protocol** step implies a string refinement to balance the SS completeness target with the effort to identify and select PS.

2.1.1 Challenges in Performing Secondary Studies

The most common problems related to SS research are: (1) the difficulty of performing complex automated searches in digital libraries, (2) the time and effort required to complete the study, (3) the definition of the research protocol, and (4) the evaluation of the quality of PS (Kitchenham et al., 2015).

Imtiaz et al. (2013) analyzed the results of 116 studies and reported that the formulation of the search strategy, planning, and data extraction are the most challenging steps in SS. They state that defining a proper search strategy is one of the most problematic and time-consuming steps in SS, mainly due to the limitations of online databases and the difficulty of creating efficient search strings.

Likewise, Ampatzoglou et al. (2019) conducted a study aiming to identify, categorize, and mitigate the threats to validity, and corresponding mitigation actions, commonly found in SS. From 165 studies, they found that two of the most common threats to validity are building the search string and selecting digital libraries. Solutions usually applied to mitigate the search string problem are using a hybrid strategy with complementary snowballing and the inclusion of synonyms in the search string. To the problem of selecting digital libraries, conventional solutions are the inclusion of the best known digital libraries and the use of search engines or indexers.

Also, Marshall et al. (2014) note that the provision of tools that support the search process is mostly absent. They further argue that this poor support may be a consequence of the difficulties inherent in automated search, and suggest that this field should be developed.

Several studies have analyzed the most considerable difficulties during the SLR or SMS process execution (Babar and Zhang, 2009; Imtiaz et al., 2013; Marshall et al., 2014; Mergel et al., 2015; Mourão et al., 2017; Ampatzoglou et al., 2019). These studies point out a latent difficulty in formulating an efficient search string and manipulating digital libraries to index all necessary searches.

This work aims to automate the formulation of the search string, using TM techniques. Therefore, we further detail relevant concepts about search strategies and completeness assessment of a SS.

2.2 Search Strategies and the Completeness Target of a Secondary Study

An essential step in any SS is formalizing a search strategy to find the most substantial number of relevant studies to address the research questions (Kitchenham et al., 2015).

There are many ways of searching for relevant PS in the literature. One of the widely used search methods is the automated search using digital libraries. Another complementary technique that has widely used in the identification of PS is snowballing, described by Wohlin (2014).

Snowballing is a technique generally used in hybrid strategies, which verifies, from a Start Set, all studies that cited and were cited by each item in this set, in order to identify more relevant studies. Wohlin (2014) stresses the importance of choosing the Start Set for the success of the technique, as the initial set will determine the path followed by the process. Usually, the Start Set is chosen through an automated search.

2.2.1 Automated Literature Search with Search Strings

The automated search with search strings involves using some electronic resources such as digital libraries and indexing systems to search for relevant studies. The researcher must decide which electronic resources intended use and also specify the search strings that will conduct the research.

The definition and refinement of search strings are iterative processes. Search string keywords are usually obtained directly from research questions or terms often used (keywords) in relevant studies. Synonyms of the keywords can also complement the list of terms. The format of search strings, with different keywords and logical combinations, change the returned results, depending on the chosen association (Dieste and Padua, 2007). The logical combination of terms implies a complex balance between a search that finds most relevant studies (i.e., a high recall) and one that does not generate numerous irrelevant studies (i.e., best precision) (Kitchenham et al., 2015).

Each digital library or indexing system (e.g., Scopus, IEEE Xplore, ACM, Web of Science, SpringerLink) has a specific way of accepting the search string formulation. In general, the changes are small, but some adaptations are required to use the search string formulated on different libraries.

Database search always returns studies outside the research context since other areas of knowledge could apply search string terms. Aiming to keep the search in the research context, the snowballing strategy presents efficiency comparable to database search while minimizes the noise from not related areas of concern (Badampudi et al., 2015).

2.2.2 The Snowballing Search

The purpose of the analysis of references and citations is to find distinct relevant studies that cite (forward snowballing) or were cited (backward snowballing) by relevant studies already found. The strategy uses communication between studies to deepen the search. The citation's locus and context usually provide information about the content of the candidate study, and it is practical to get this information from a paper being examined instead of finding the candidate by a database search (Wohlin, 2014).

Snowballing should not necessarily be seen as an alternative to database searches. Different approaches to identifying the relevant literature should be used together in an hybrid approach, seeking to ensure the best possible coverage of the literature. Snowballing is useful to deepen any SS, since relevant papers tend to cite or be cited by other relevant studies on the research topic (Booth, 2016). It is usually used as a secondary method to complement an automated search. In an hybrid approach, the results of an initial search using search strings are used as a Start Set for the snowballing.

An essential requirement of any SS is to outline a search strategy that retrieves as many studies as possible that answer the research questions. Probably the strategy will combine search methods. Indeed, the search strategy aims to achieve an acceptable level of completeness within time and human resources available (Kitchenham et al., 2015).

Within the next section, we look at how to assess the completeness of the set of studies found by a search process.

2.3 Completeness Assessment

In the context of Information Retrieval theory, retrieval effectiveness is mostly measured in terms of precision and recall or by measures based thereon (van Rijsbergen, 1979). Precision is the ratio of the number of relevant documents retrieved to the total number of documents retrieved, and recall (or sensitivity) is the ratio of the number of relevant documents retrieved to the total number of relevant documents (retrieved and not retrieved). These can be calculated as follows:

$$Recall = \frac{R_{found}}{R_{total}} \tag{2.1}$$

$$Precision = \frac{R_{found}}{N_{total}} \tag{2.2}$$

If possible, a search should have a high recall. In short, it should find most of the relevant studies (Kitchenham et al., 2015). A high recall search strategy maximizes the result of relevant studies retrieved; however, it retrieves many irrelevant studies in return.

On the other hand, precision is also essential and desirable because high precision means that the load on reviewers to check studies that become not to be relevant is low. Nevertheless, a high precision strategy decreases the number of retrieved studies with the penalty of missing a good number of relevant ones. Thus, a trade-off is unavoidable between recall and precision. The goal will be to get optimal or at least an acceptable search strategy (Dieste et al., 2009).

In qualitative studies, and for SS where a complete identification of relevant PS is hard to achieve (e.g., in SE), a lower degree of completeness may be tolerated (Dieste et al., 2009). Likewise, Cooper et al. (2018) claim that comprehensive searches are not necessarily an indicator of quality, as previously thought.

Petitti et al. (2000) states that "an optimal search can be defined as a search that strikes a balance between high recall and high precision" (as quoted in Dieste et al. (2009)).

The *"effectiveness measure"* proposed by van Rijsbergen (1979) derives the F-measure (also F-score), which is defined as the harmonic mean of precision (P) and recall (R). Its general formula is shown in Equation 2.3. The F-Score's general formula has a positive real β, where β is chosen so that the recall is considered β times more important than the precision.

For a balanced F-Score, that corresponds to a user who attaches equal importance to precision and recall, the value of β is equal to one, being called F1-Score, as shown in Equation 2.4.

$$F_{\beta}\text{-}Score = (1+\beta^2)\frac{P \times R}{(\beta^2 \times P) + R} \tag{2.3}$$

$$F_1\text{-}Score = 2 \times \frac{P \times R}{P + R} \tag{2.4}$$

An actual conundrum in a recall calculation is that the total number of relevant studies (R_{total}), is not known. Unfortunately, it is impossible to find out the value of this parameter in advance.

A way of assessing completeness is to have a set of studies that are known as relevant studies. This known set can be obtained in several ways, such as through the construction of a Quasi-Gold Standard (QGS). The use of a QGS to assess a search strategy is discussed following.

2.4 Quality Assessment of Search Strategies Using Quasi Gold Standard

A Gold Standard (GS) represents the set of studies (the "golden bullets") that undoubtedly should be part of the review due to their alignment with the inclusion criteria for the SS (Zwakman et al., 2018). The GS has been used to refine search strategies in systematic reviews in other disciplines, such as medicine, clinical research, and social science.

As it is not possible to have GS for most SS in SE, Zhang et al. (2011) introduces the concept of Quasi-Gold Standard (QGS) as a set of known studies on a research topic. The QGS has two fundamental roles in SS: (1) it is a set of documents used to evaluate the search process, and (2) is a source for the refinement of the search strings (Kitchenham et al., 2015).

Figure 2.2 shows how QGS can assist the SS process. A manual search finds the QGS. From this QGS, search terms are extracted and used during the automated search. When observing the automated search result, the researcher decides to redo the automated search or considers it to be good enough. When the researcher considers that the formulated search string is good enough, the relevant studies obtained by the automated search are then used in snowballing. The sum of all these steps then provides the GS.

Kitchenham et al. (2011) proposes an improvement of QGS usage. The authors suggest that the set of known papers should be split into two independent subsets, and one is used to refine search strings. In contrast, the other independent subset should be used to evaluate the effectiveness of the search process.

Identifying search terms to retrieve an unknown set of literature is challenging, particularly for literature that uses varied terminology or is not consistently indexed. Given the challenges in creating search strings from a QGS sample of literature on a topic, TM is considered here as an aid that could

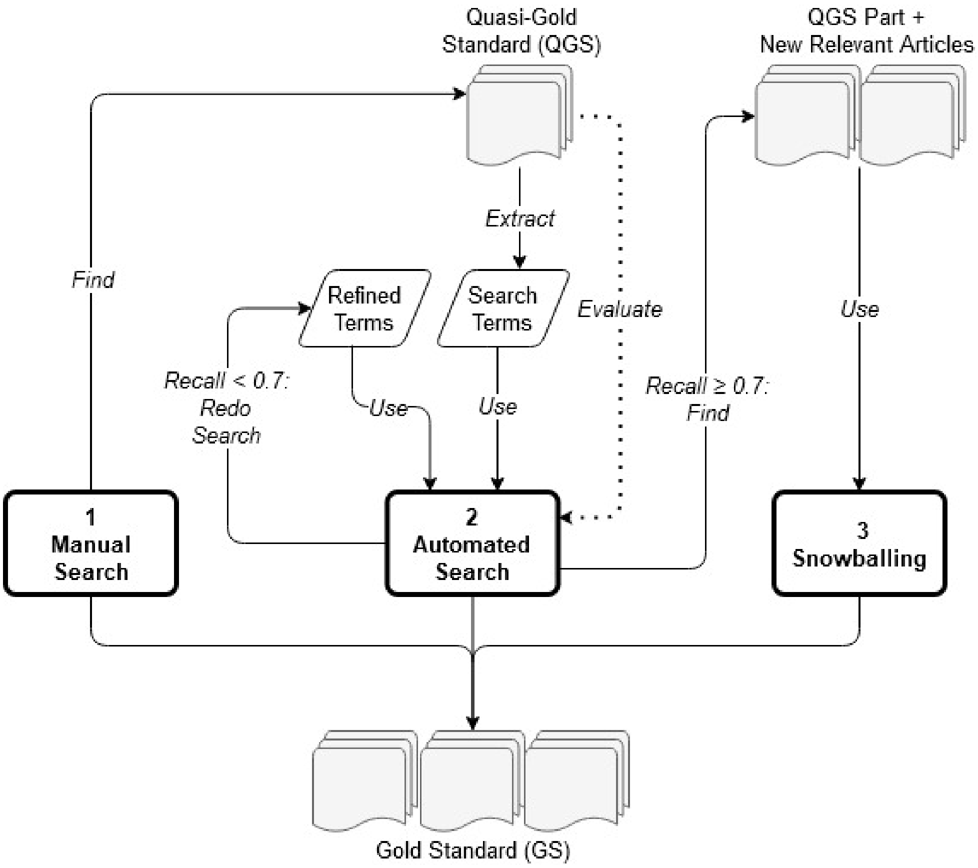


Figure 2.2: The use of QGS to support the search for relevant studies in a hybrid search strategy. Based on Zhang et al. (2011).

contribute to a solution aiming to devise search strategies for topics that wrap a range of conceptual aspects or are described by diverse vocabularies.

2.5 Text Mining

TM can be characterized as a set of techniques and processes to discover new knowledge in texts (Aggarwal and Zhai, 2012). In a scenario where much data is available textually, the TM process emerges as a relevant knowledge management support tool. TM aims to look for patterns and trends in textual documents and be considered a specialization of the Data Mining. While Data Mining operates on predefined databases, TM works with inherently unstructured data (Weiss et al., 2010).

The TM process is developed in a cycle, where at the end of the process, the user acquires knowledge about the analyzed data. This process can be organized according to each application's requirements, and there are several ways to describe it. Based on the work of Fayyad et al. (1996), Weiss and Indurkhya (1998) and Shearer (2000), Rezende (2003) describes the TM process in five steps: (1) problem identification, (2) pre-processing, (3) pattern extraction, (4) post-processing, and (5) knowledge usage, as presented in Figure 2.3.

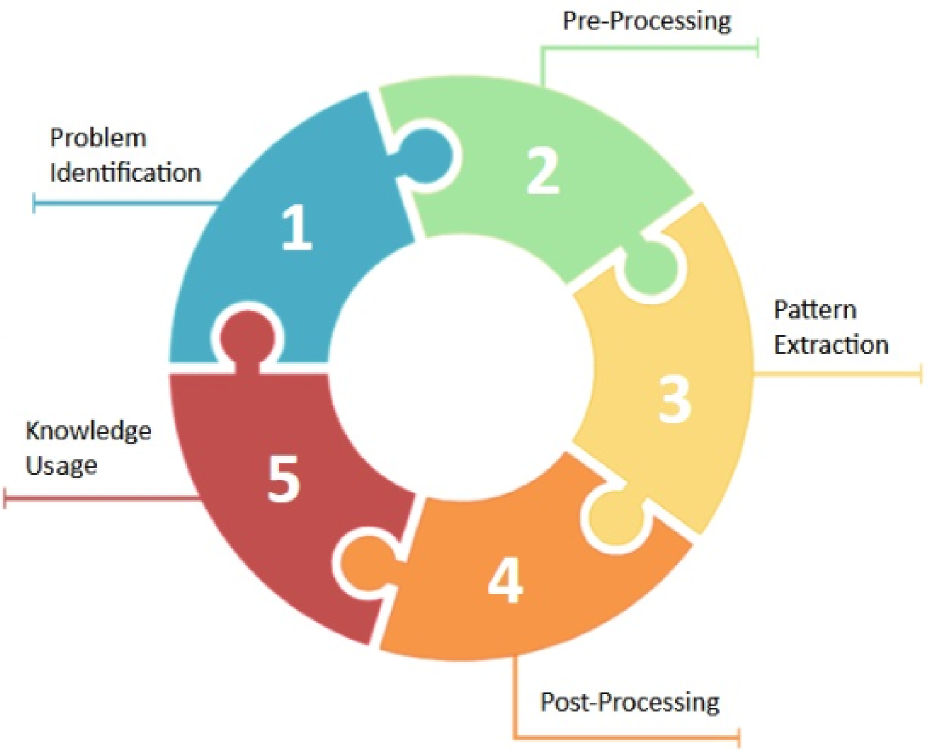


Figure 2.3: Cycle that makes up the TM process and its respective steps. Based on Rezende (2003).

This process is iterative (can be repeated several times to adjust parameters or improve the data selection process for better results in the next iteration) and interactive (based on the interaction users). The steps in the TM process are described as follows. The role of each user class is also superficially presented. Pre-processing, pattern extraction, and post-processing will have more details as they form the process's core.

2.5.1 Problem Identification

At this stage, the domain expert identifies and delimits the scope of the research problem and the collection of texts to be analyzed. This domain expert also examines any prior knowledge in the studied domain that may be beneficial, what is expected at the end of the process and how the results may be used.

Problem Identification is a crucial step since there is no knowledge discovery without the demand for it. Rezende (2003) define four questions that must be answered in this step, adapted from the CRISP-DM model (Shearer, 2000):

1. *What are the main goals of the process?*

2. *What performance criteria are important?*

3. *Should the extracted knowledge be understandable to humans or a black box model is appropriate?*

4. *What should be the relationship between simplicity and precision of the extracted model?*

The decisions that are made in the Problem Identification step guide the subsequent steps in the process.

2.5.2 Pre-Processing

The Pre-Processing step is one of the most important and time-consuming steps. It aims to structure the texts for knowledge extraction. Generally, the analyst should check for features that ensure reliability, non-redundancy, and balance of the text collection.

According to Laguna et al. (2014), pre-processing is performed in five activities that seek to extract the terms that represent the collection of documents. These activities are (1) standardization of text formats; (2) text cleaning; (3) normalization of the words; (4) identification of terms and (5) representation of the textual collection. Such steps are represented in the diagram shown in Figure 2.4.

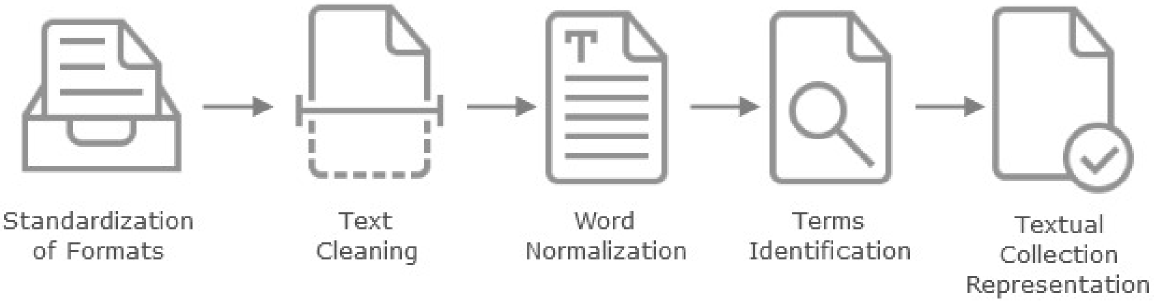


Figure 2.4: Activities developed during the pre-processing step. Based in Laguna et al. (2014).

Since documents can come from various data sources, they can be found in different formats, such as Portable Text Format (PDF) and hypertext. To ensure that all documents are equally accessible and manipulable, formats must be standardized to a plain text format.

All useless content found in standardized documents is removed. Punctuation and mathematical symbols, if they are not useful for the application, are cleared from texts. Terms that do not represent usable knowledge, known as stopwords, must be removed. Examples of stopwords are prepositions, pronouns, studies, interjections, adverbs, among others. In certain domains, domain-specific stoplists are also common, which are sets of words that can be disregarded in the application context.

Subsequently, similarities of meanings are sought among the remaining words, as in cases of morphological variations or synonyms, aiming to normalize the texts' terms. For this, according to Nogueira (2013), it is possible to reduce a word for your root through the stemming processes (Krovetz, 1993), to reduce the words to its lemma through the lemmatization (Arampatzis et al., 1999), substantiate the terms through the substantivation (Gonzalez et al., 2006) or even use dictionaries or thesaurus (Jones and Willett, 1997).

Since words are normalized, Laguna et al. (2014) indicates the use of statistical measures to weight terms in the collection, such as:

- **Boolean**: it assigns the value one to the attribute present in the collection-specific document and zero otherwise;

- **Term Frequency (TF)**: it consists of counting appearances of a given term in a collection document;

- **Document Frequency (DF)**: it counts the number of documents that the term candidate has in the document collection;

- **Term Frequency Inverse Document Frequency (TF-IDF)**: it gives less weight to terms that appear in many documents, making very traditional terms unrepresentative.

The choice of the most appropriate measure depends on the purpose of the application. It also depends on whether each term's representation is calculated in a given document or the entire collection. These metrics can be used to filter the terms in the representation. Relevant terms can be selected by employing a metric threshold (Luhn, 1958).

Such approaches are sufficient to identify simple terms (uni-grams). From these simple terms, it is possible to search the document collection for compound terms (n-grams), consisting of more than one element, but with a single semantic meaning (Manning et al., 2010). The compound terms found in the documents represent the notion of occurrence context since it considers the order of appearance to form the compound terms.

The selected terms are used to form a structured representation for the document collection. The most common structure is the Vector Space Model (VSM) (Salton et al., 1975). In the VSM model, each document is represented as a vector (D_i), and each position of this vector corresponds to a document term (T_j). The vectors for all documents form the attribute-value matrix, as presented in Table 2.1. This matrix is known as bag-of-words. The last column of the matrix corresponds to the document class (C_i) if the process handles labeled data. The A_{ij} cell values are filled using the weight metrics previously discussed.

Table 2.1: Representation of a attribute-value matrix. Each row represents a document and each column represents a term. $D_1 ... D_N$ are the documents, $T_1 ... T_M$ are the terms of the documents, $A_{11} ... A_{NM}$ are the cell values, filled in according to the metric used and $C_1 ... C_N$ are the document class.

Terms Docs	T_1	T_2	...	T_M	Class
D_1	A_{11}	A_{12}	$\dots$	A_{1M}	C_1
D_2	A_{21}	A_{22}	$\dots$	A_{1M}	C_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
D_N	A_{N1}	A_{N2}	$\dots$	A_{NM}	C_N

2.5.3 Embedding-Based Documents Pre-Processing

With the development of more refined techniques of Machine Learning (ML) and Natural Language Processing (NLP), more effective representations emerged that take into account the order of occurrence of the words. Bengio et al. (2003) used neural networks to automatically learn vector representations. This type of representation, formally titled as distributed representations, is currently known as word embeddings or just embeddings.

Turian et al. (2010) explains that word embeddings are real vectors distributed in a multidimensional space induced through unsupervised learning. Each vector dimension of a specific word represents a characteristic of it. This is done to capture semantic, syntactic, or morphological properties of the word in question (Collobert et al., 2011). The number of vector dimensions can differ, and it is generally possible to obtain better representations with more dimensions. However, if this number is too large, the process can become extremely time-consuming.

Collobert and Weston (2008) proposes a model based on a neural network that tries to predict the next term for its context, considering the previous terms. Word representations are learned by adjusting the net weights using the back-propagation algorithm. Thereby, words that occur in equivalent circumstances have similar vectors, and it is the process of training the network that absorbs this notion.

The word embeddings model known as Word2Vec, proposed by Mikolov et al. (2013), popularized this type of representation. The technique used to generate the model follows the same principle as presented in Collobert and Weston (2008), except that it does not use the hidden layer of the network, providing a computationally faster log-linear model. Word2Vec is divided into two models. The first, called CBOW (Continuous Bag-of-Words), receive a window of input words that the network tries to predict the middle word as output. Meanwhile, the other, known as Skip-gram, tries to predict the words in the context window using the middle word as input.

In the model presented, vector representations are generated only for words found in the corpus (set of written texts and oral records in a given language that serves as the basis of analysis), since word-level context windows are vital to the training. This feature may be problematic for applications whose corpus is not the same as the one used for the training of words embeddings, as there may be words that do not have associated vectors.

Although Mikolov et al. (2018) and other studies like Pennington et al. (2014) was instrumental in advancing the field of research, studies on word representations that focus on representations regardless of the learning context are considered obsolete. More recent work focused on context-dependent representations of learning using NLP tasks. For example, ELMo (Peters et al., 2018) uses a bidirectional language model, while CoVe (McCann et al., 2017) uses machine translation to incorporate context information into word representations (Lee et al., 2019).

Another advance in this area is BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019), a contextualized word representation model, which is pre-trained based on a masked language model using bidirectional transformer from and unlabeled text set. BERT uses a masked language model that predicts randomly masked words in a sequence and hence can be used for learning better bidirectional representations. BERT encodes the input tokens as a sum of token embedding segmentation embedding and position embedding. This embedding is thus capable of capturing the semantic and contextual meaning for each word. Currently, BERT achieved state-of-the-art performance in many NLP tasks (Lee et al., 2019).

2.5.4 Pattern Extraction

The pattern extraction step is focused on meeting the objectives set during Problem Identification. That step selects, configures and executes one or more pattern extraction algorithms, usually based on ML. As in Data Mining, the main Pattern Extraction activities are classified as predictive and descriptive (Rezende, 2003).

Predictive activities consist of generalizing past experiences with known answers. These tasks use supervised algorithms, which depends on fully labeled datasets. Supervised algorithms may perform two tasks: classification and regression. Classification refers to the process in which the class attribute has a categorical value. Regression, in turn, symbolizes the process of seeking to predict real-valued classes (Weiss and Indurkhya, 1998).

Descriptive activities consist of identifying behaviors intrinsic to the set of texts. These tasks use unsupervised ML algorithms, where all data is unlabeled. This class of algorithms has two main branches: association rules and clustering.

Association rules are logical relations inferred between correlated data from the text collection, analyzed together, with the central objective of finding associations between collection items (Agrawal and Srikant, 1994).

Data grouping, also known as clustering, seeks to group data according to some measure of similarity between them. So, data in the same group are similar to each other, while data in different clusters are different (Manning et al., 2010). In TM, clustering is usually employed to perform topic modeling. This activity consists of finding sets of similar documents and the most representative words in these sets.

A classic topic modeling technique is Latent Dirichlet Allocation (LDA), a statistical model that allows sets of terms to be described by topics that explain why some parts of the data are similar (Blei et al., 2003). LDA assumes that each document is a mixture of a set of topics, as well as each topic is a mixture of a set of words. LDA has two basic parameters: a prior of document topic distribution α; and a prior of topic word distribution β. By presenting a set o M documents with N words, LDA generates K topics, which can be understood as a set of words. For each topic we can compute $\phi(k)$, the word distribution for topic k, as well as $\theta(m)$, the topic distribution for document m. For each topic k, it is possible to select a set of W words that describe that topic. These words may be the W most relevant words according to $\phi(k)$. In this work, we use LDA to extract topics from a QGS and use those topics to formulate the search string. More details of this process are presented in Section 4.

2.5.5 Post-Processing and Knowledge Usage

After performing the Pattern Extraction, a validation step known as Post-Processing is required. Post-processing is the stage at which users validate their findings. Users must evaluate, for example, the representativeness according to the application context, such as the representativeness of the aggregated knowledge, the novelty contained in the results obtained, and how such knowledge can be used (Nogueira, 2009). In summary, this is the step where findings made in Pattern Extraction are simplified, evaluated, viewed, and documented for the end-user.

After being evaluated and validated in Post-Processing, the knowledge utilization stage consolidates the information obtained. The knowledge is incorporated into an intelligent system or used directly by the end-user to support some decision-making process. If the process has taken place correctly, it can be ensured that knowledge at this point is valid and useful.

Given this background, it is possible to realize that the SS development process is pervasive and that techniques can be used to assist this process as a whole or at least parts of it. In the next session, some works found in the literature that use TM techniques and ML to support the SS process will be presented.

Section 3

Related Work

Ali and Usman (2018) presents a SS that explores conventional approaches to search strategy in SS. Automated search is the most frequently used method. Among these studies, 33% use automated search alone. Besides that, 49% use automated search as their primary search method, supported primarily by a snowballing step, and in one case by a manual search. Only one SLR did not use automatic search as the primary or secondary method, highlighting the importance of search string formulation. Note also that 96% of studies used some search string to find studies that answered the research questions.

Almost half of the studies mentioned in Ali and Usman (2018) do not only use automated search in digital libraries. This shows that SE researchers consider the automated search insufficient to explore the area under study. Similarly, Mourão et al. (2020) compare several hybrid search strategies and state that the use of a search string in Scopus followed by snowballing would provide a high search recall.

Tsafnat et al. (2014) presented a study on the automation of SS. The authors describe in detail the potential for automation in different parts of the SS process. According to the authors, decision support tools for search strategy derivation should suggest data sources, keywords, and search strategies. Some algorithms can also be used to increase the utility (in terms of precision and recall) of user search strings or help users write a better search string.

With the increased use of SS to answer certain research questions, several researchers have created tools that support some part of the SS process. With that, some SS were carried out with the objective of grouping these tools, making it more evident which areas had not yet been fully explored. Marshall and

Brereton (2013), O'Mara-Eves et al. (2015), Feng et al. (2017) and Stansfield et al. (2017) investigate tools or approaches that help the SS process using TM. Many authors reinforce the importance of partial or total automation of the search strategy. However, such studies show that automation's focus is in other stages of the SS, like study selection.

Marshall and Brereton (2013) conducted an SMS to obtain the current state of the art on tools to support SS in SE. The results suggest a predominance of visualization and TM techniques to support study selection, data extraction, and data synthesis stages of the SS process.

Also, O'Mara-Eves et al. (2015) conducted an SLR investigating the use of TM to select studies specifically in systematic reviews. Through this study, it is possible to observe many approaches that address the selection activity of PS. On the other hand, results suggest a lack of effort to formulate search strategies for automated PS selection.

Feng et al. (2017) performed an SLR with a focus on TM techniques and tools that support any activity of the SLR process. The authors report that the current studies offer only two solutions that support the search process, despite its time-consuming and labor-intensive characteristics. These solutions use (i) federated search approaches, providing a unified interface for all data sources; or (ii) search string expansion techniques (Tomassetti et al., 2011; Ghafari et al., 2012; Smalheiser et al., 2014).

Finally, Stansfield et al. (2017) performs a brief review of the literature on TM applications for developing search terms for SLRs. According to their findings, the TM approaches can be used in five ways: (i) to improve search precision; (ii) to identify search terms to improve search recall; (iii) to aid the translation of search strategies across databases; (iv) to search and to screen within an integrated system; and (v) to develop objectively-derived search strategies.

The problem of search string formulation is addressed in a few works on the literature. Ros et al. (2017), for example, explore an approach to support semi-automated search and selection in SLRs. They propose a cyclical process for identifying studies, where decision trees with information obtained from the identified studies form the search string. The idea is that every path from the root of the decision tree to a leaf with a positive classification generates a search string that will be joined later. Making a search string that improves itself with the results obtained seems promising but extremely complicated. Besides, the results presented are initial, made in a proof of concept tool.

Marcos-Pablos and García-Peñalvo (2018) develop a TM and ML technique that acts on a corpus of downloaded abstracts. This technique requires an initial search string. The abstracts collected using this initial search string are projected into a vector space. Then, manually or using Support Vector Machine (SVM), the abstracts are labeled as relevant or irrelevant. With this labeling, new terms are obtained and can be entered into the search string manually.

Grames et al. (2019) produces a quasi-automated method that generates improvements to search strings, based on the use of TM and keyword co-occurrence networks. The approach combines automated and manuals steps, where at the beginning of the process, it is necessary to formulate an initial search string, returning several studies. From these studies, relevant terms are extracted and treated with a document × feature matrix and a co-occurrence network. Finally, terms are provided to the researchers, who must filter them and organize them into groups to provide enhanced search strings.

Lastly, there is an approach provided by Mergel et al. (2015) proposes an iterative method to support the search string refinement. This method originated a tool, called SLR.qub. In practice, SLR.qub receives an initial search string and displays the results of using this search string in the IEEE Xplore digital library. From the returned studies' names, the researchers vote whether the study appears to be relevant or not. The terms of the relevant studies are ordered based on the TF-IDF calculation. These terms are provided to the tool used to refine their initial search string manually.

The use of TM to assist the SS process is not uncommon. However, most studies are concerned with other stages that also require a lot of time and effort, such as selecting studies. Few studies seek to automate part of the formulation stage of the search strategy.

Among the studies that automate part of the search strategy, the vast majority only support the search string refinement. These refinements usually occur after a manual creation of an initial search string. Therefore, they are not intended to provide an initial search string. Also, do not usually worry about possible search strategies that not only include the automated search.

We will then present an approach named SeSG that aims to create an initial search string, without requiring manual refinement. Moreover, the approach is targeted for use in a hybrid search strategy, predicting that a snowballing step will be made after the automated search.

SECTION

4

Search String Generator (SeSG)

As discussed in Section 3, there is a lack of TM approaches that assist in the automatic formulation of search strings. As a consequence of this gap, we present an approach called Search String Generator (SeSG). SeSG aims at reducing the effort spent by researchers in the generation and refinement of search strings to be used in a hybrid search strategy. Using a set of pre-selected studies (called QGS), SeSG generates a search string that can retrieve relevant studies to the research field portrayed in the QGS.

The objective of SeSG is to automatically generate a search string without manual refinements. The lack of manual refinements in the search string can affect the completeness of the search, involuntarily excluding studies that would be relevant to the research questions. For this reason, SeSG is suggested for use in hybrid search strategies. If any relevant study is not achieved from the search using the search string generated by SeSG, snowballing will likely find this missing study.

Besides completeness, other factors make SeSG preferentially (but not exclusively) used in hybrid search strategies. The difficulty of manually obtaining a QGS that represents all the ramifications of the research area in question makes advisable the subsequent use of snowballing. In addition, the selection bias in listing the QGS is also partially removed when using a hybrid strategy. The QGS will have a crucial relationship in the formulation and evaluation of the search string, but it will not radically impact the search completeness.

In an SLR, the refinement of a search string occurs during the protocol development step to assist the identification of PS (Kitchenham et al., 2015). A good search string must present a high recall (percentage of the relevant documents retrieved) and high precision (percentage of recovered documents that are relevant). We employ the harmonic mean of recall and precision, known as the F1-Score (van Rijsbergen, 1979), to evaluate the search results. The F1-Score is calculated comparing the results retrieved in digital libraries using the automatically generated string having the QGS as a parameter.

SeSG is an instance of the TM process presented in Rezende (2003). Figure 4.1 presents this instance, showing that the steps will be carried out in sub-processes. In the next sections, we explain the four core steps of SeSG: (i) Pre-Processing; (ii) Topics Extraction and Enrichment; (iii) Generation of the Search String; and (iv) Application of the Search String in Digital Libraries.

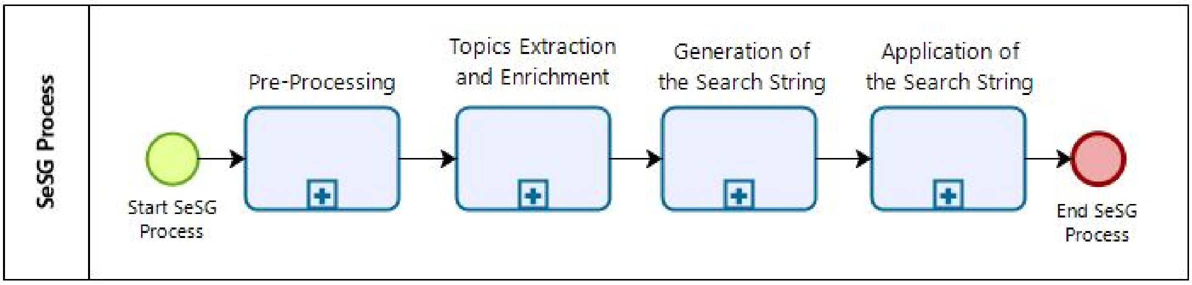

SeSG Process
Start SeSG Process
Pre-Processing
Topics Extraction and Enrichment
Generation of the Search String
Application of the Search String
End SeSG Process

Figure 4.1: The Search String Generator (SeSG) process.

4.1 Pre-Processing: Organizing the Documents and Generating Their Representations

Figure 4.2 presents an expanded view of the Pre-Processing sub-process, showing its activities. In SeSG, the input to the approach is a collection of scientific studies from a particular area of knowledge that represents the area of research to be investigated, known as the QGS. It is a difficult task to define the ideal size of the QGS. We suggest the size of the QGS (QGS_{Size}) to range from 5 to 15 studies from some initial empirical studies. This set of studies may come in a wide variety of forms and should be pre-processed to allow its correct manipulation.

First, the format of the input studies should be standardized. In general, scientific studies come in a PDF format, which is not directly manipulable. Also, different scientific studies may present different textual forms that may turn unfeasible adequate information extraction. In this sense, every study in QGS and its metadata are transformed into plain text.

A word selection process is then carried out by removing stopwords, special characters, and numbers. This task requires a stoplist, which is language-dependent and may be provided as an input to SeSG.

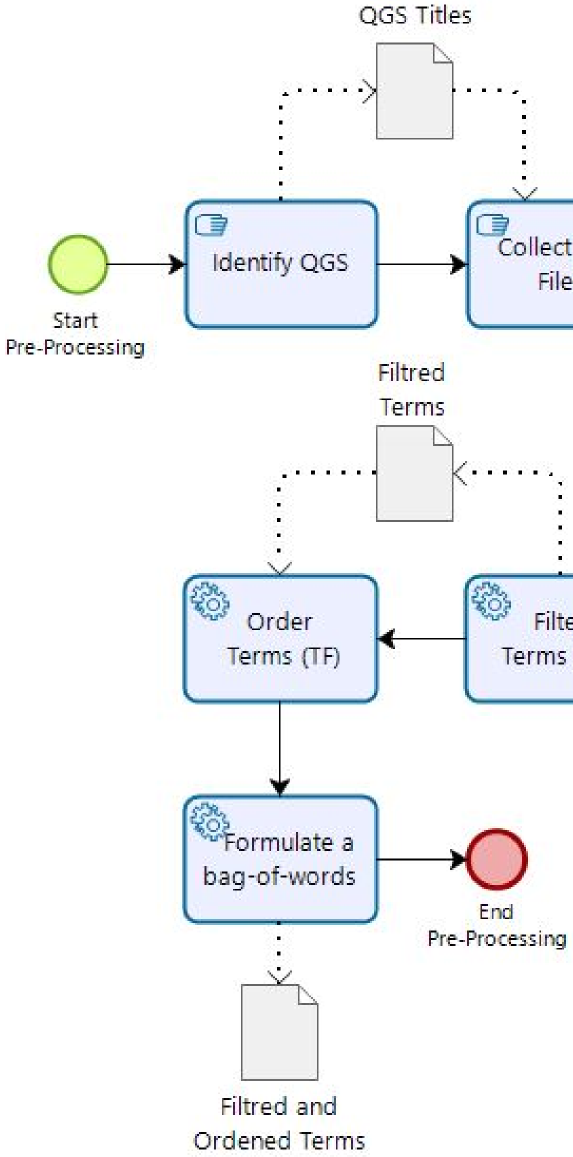


Figure 4.2: The expanded 'Pre-Processing' sub-process of the SeSG approach.

Each word (token) in the collection is considered as a simple term (unigram). From these simple terms, it is possible to search for semantically significant compound terms, also known as n-grams, where n stands for the number of tokens in the word. The n-grams are essential to convey the notion of context of occurrence since it considers the terms' order of appearance. In search string, terms composed by more than three words are unusual. Thus, in our approach, it has been delimited that the number of tokens in an n-gram varies in the range 1 to 3.

Then, infrequent terms may be removed, since they do not carry sufficient information. A simple way to filter those terms is by counting the number of documents a term occurs - the Document Frequency (DF) of a term. In SeSG, the advised range of minimum DF (Min_{DF}) is between 10% and 40% of the number of documents in the QGS and must be empirically tuned. Larger QGS may require larger values of Min_{DF}.

Finally, each selected term is related to each document in the collection through its Term Frequency (TF). This metric counts the number of occurrences of each term in each document. This kind of association is structured as a vector space model, called bag-of-words. The attribute-value matrix representing this vector space model is used as input for the topic extraction, described in the next section.

4.2 Topic Modeling Using LDA and Enrichment of Terms Using BERT

In this step, as described in Figure 4.3, two main activities are carried out to formulate search strings: (i) topic extraction through LDA, which finds the most significant terms that describe the different topics addressed in the QGS; and (ii) the enrichment of the topics, using the BERT word embedding to add similar words to the topics.

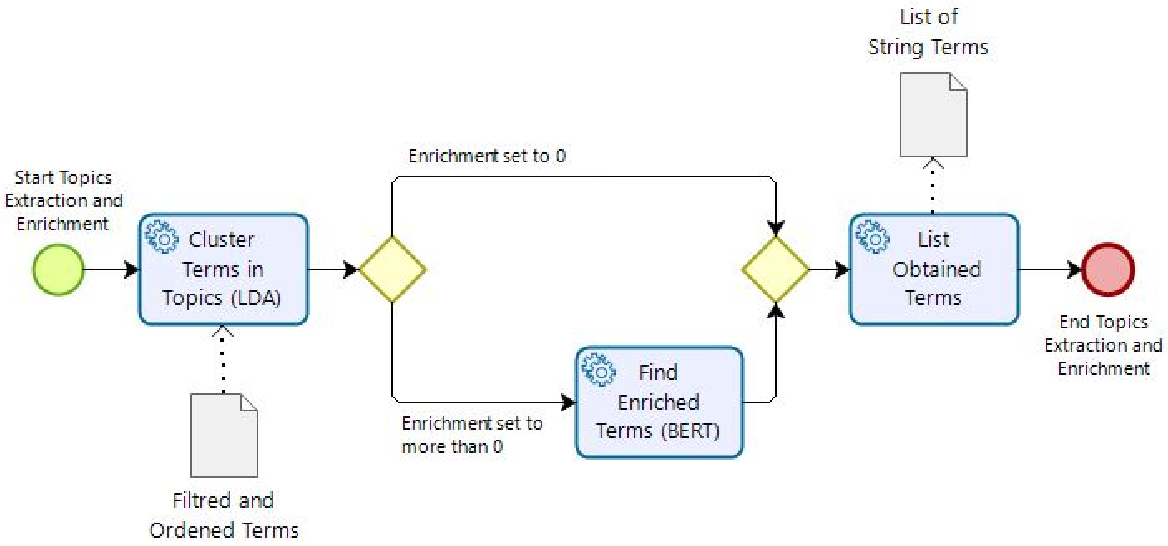


Figure 4.3: The expanded 'Topics Extraction and Enrichment' sub-process of the SeSG approach.

In the first activity, LDA implicitly groups the words in the QGS into a predefined number K of disjoint groups (topics). Then, LDA finds the W most representative terms to each of these groups (topic descriptors), capturing the core subjects addressed in these different groups. The details of this clustering process were discussed in Section 2.5.4.

In the search string generation context, we assume that the K different topics found by LDA should be concatenated with disjunction operators (OR) since these topics represent the different subjects in the QGS. Similarly, the W

different words on the same topic should be concatenated with a conjunction operator (AND), since these are the words that represent that topic. LDA's application requires as input the number of topics desired K and the number of descriptors per topic W, which may be empirically tuned. The broader the research area under investigation is, the more topics and descriptors may be extracted. Both Dirichlet priors α and β are automatically calculated by using $(1/K)$, as suggested by Blei et al. (2003).

Since a small set of representative studies forms the QGS, the vocabulary analyzed by LDA, and consequently, the topic descriptors are limited to words present in these studies. This limitation may impact the recall of the generated search strings since different words are commonly used to represent the same concept. In order to overcome this limitation, in the second activity of the Pattern Extraction step in SeSG we introduce similar words to enrich the topics extracted. For this end, we use the contextualized word representation model called BERT (Devlin et al., 2019).

As discussed in Section 2.5.3, BERT learns a word embedding in which words with similar semantic meaning present similar representation. This high-quality feature fits the objectives of this work, since semantically similar words may be used to expand search strings and perform a broader search. In SeSG, we use a BERT word embedding to incorporate, for each of the W words in the K topics, the S most similar terms to each detected topic.

Word embeddings like BERT encode semantically similar words with similar embeddings. So, similar terms may be found by calculating the cosine distance between the word embeddings. For this process, SeSG requires a pre-trained BERT embedding, like the one made available by Devlin et al. (2019), which was trained on the Wikipedia dataset. Performing a fine tuning, training a BERT embedding for the specific context is also an alternative. However, this would limit the scope of the approach, and therefore this idea was discarded. The parameter S must be defined beforehand, considering that a large value of S may deteriorate the search precision and a small value of S may negatively impact the search recall.

An example of the topic extraction in SeSG can be observed in Figure 4.4. This process takes as input the bag-of-words generated in the Pre-Processing step from an initial set of documents (QGS), considering a Min_{DF} previously established. In this example, LDA is executed using $K = 2$ and $W = 2$, while BERT was used to add $S = 1$ similar word to each topic descriptor.

The enriched topics are then used to generate the search string. In the next section, we detail the generation of the search string in SeSG, which concatenates the topics and their enriched descriptors to formulate the search string.

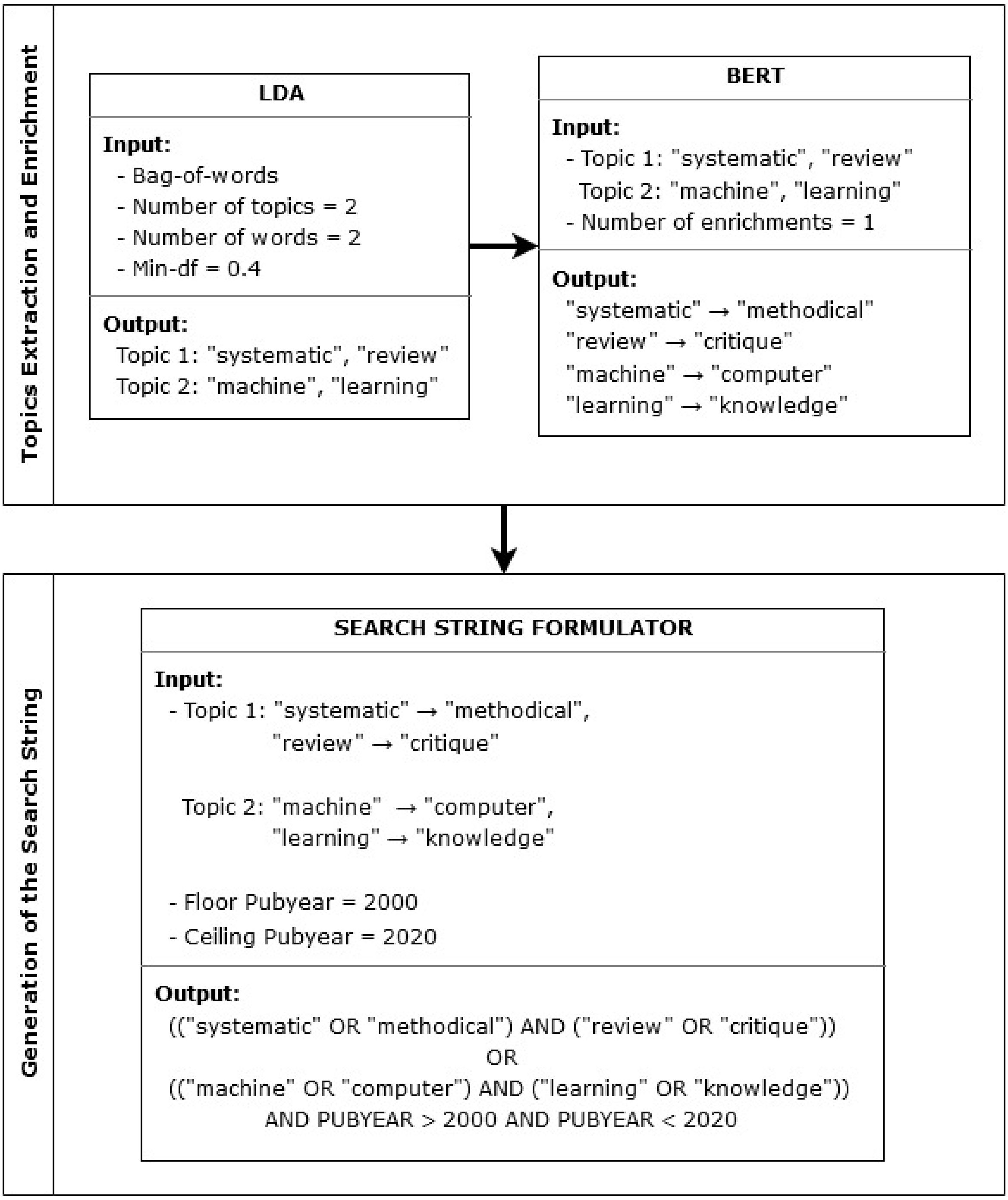

Topics Extraction and Enrichment
LDA
Input:
- Bag-of-words
- Number of topics = 2
- Number of words = 2
- Min-df = 0.4
Output:
Topic 1: "systematic", "review"
Topic 2: "machine", "learning"
BERT
Input:
- Topic 1: "systematic", "review"
Topic 2: "machine", "learning"
- Number of enrichments = 1
Output:
"systematic" → "methodical"
"review" → "critique"
"machine" → "computer"
"learning" → "knowledge"
Generation of the Search String
SEARCH STRING FORMULATOR
Input:
- Topic 1: "systematic" → "methodical",
"review" → "critique"
Topic 2: "machine" → "computer",
"learning" → "knowledge"
- Floor Pubyear = 2000
- Ceiling Pubyear = 2020
Output:
(("systematic" OR "methodical") AND ("review" OR "critique"))
OR
(("machine" OR "computer") AND ("learning" OR "knowledge"))
AND PUBYEAR > 2000 AND PUBYEAR < 2020

Figure 4.4: An example of the Topics Extraction and Enrichment and Generation of the Search String sub-processes of SeSG.

4.3 Generation of the Search String

With the terms obtained in the extraction of topics and the use of enrichment techniques, we have a collection of terms and topics that represent the set of studies provided in the entry (QGS). With this collection in hand, it is necessary to formulate the boolean search string, following the process described in Figure 4.5. This string should follow the standardization that the search engines impose, with the concatenations between terms with "ANDs" and "ORs".

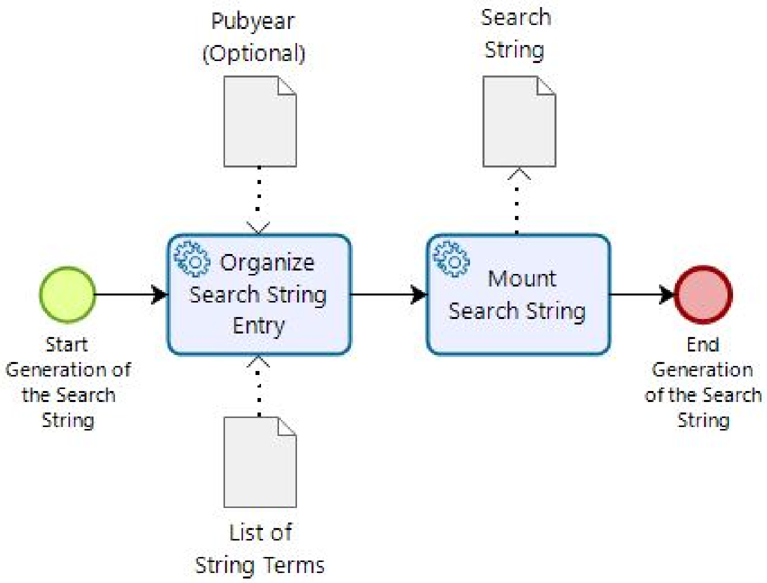


Figure 4.5: The expanded 'Generation of the Search String' sub-process of the SeSG approach.

Most of the digital libraries accept boolean search strings. The reserved word "AND" is used when all the terms connected by it must appear, being able to be distant from one another (e.g., "machine" AND "learning"). On the other hand, the reserved word "OR" is used when at least one of the terms linked must necessarily appear, symbolizing synonyms, alternative spellings or abbreviations (e.g., "process" OR "method").

In Figure 4.6, we illustrate the formulation of the search string shown in the second part of Figure 4.4. In this example, two topics are found with LDA, and each of these topics has two descriptors. Note in the first representation that the topics are concatenated by "ORs" connectors, while "ANDs" concatenate the terms in a topic. Then, each topic descriptor is enriched with its most similar word with an "OR" concatenation.

When formulating the search string, there is an optional parameter that delimits the publication year of the studies retrieved in the search. This parameter is commonly accepted in search engines with the directive *PUBYEAR*

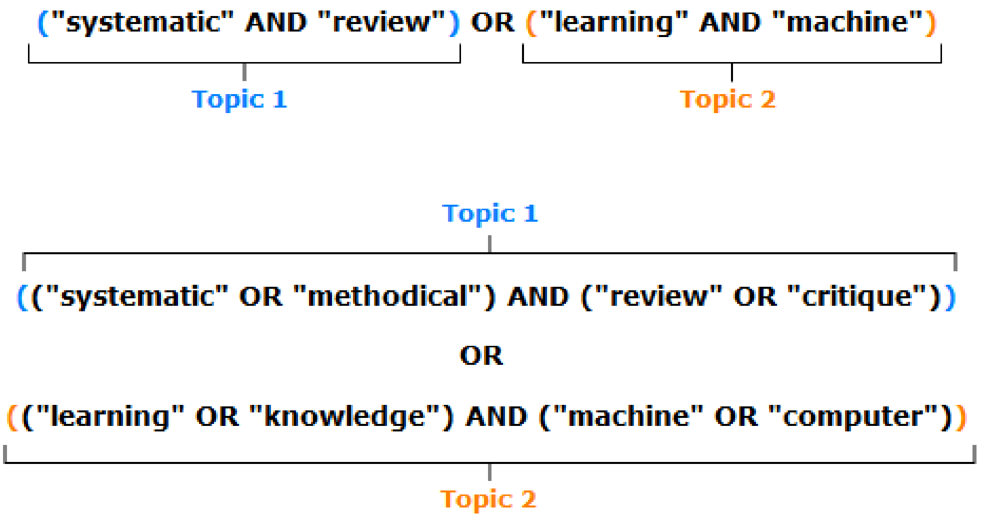

("systematic" AND "review") OR ("learning" AND "machine")
Topic 1
Topic 2
Topic 1
(("systematic" OR "methodical") AND ("review" OR "critique"))
OR
(("learning" OR "knowledge") AND ("machine" OR "computer"))
Topic 2

Figure 4.6: Concept used to formulate the search string, without and with the enrichment of terms.

and is concatenated to the end of the search string with an AND operator. This parameter is expressed as an interval of years (floor and ceiling). For example, suppose that a user would like a research to be limited to studies published after 2010 and before 2015. So, 2009 may be set as the floor and 2016 as the ceiling years. In this case, SeSG adds the constraint *PUBYEAR > 2009 AND PUBYEAR < 2016*. If this limitation is not required, it can be ignored, and SeSG creates the search string using just terms found in the topic modeling step.

After this process, the search string is ready to be used in a search engine. The usage of the search string and the evaluation of the results are described in the next section.

4.4 Application of the Search String in Digital Libraries

Having an initial search string, the automated search may be executed and evaluated, as shown in Figure 4.7. First, the search string generated by SeSG may be introduced in the search field of a digital library. The studies retrieved by this search are considered as a Start Set for a hybrid search, as described in Section 2.2.2.

The completeness assessment of the Start Set in terms of the QGS can be used as a quality measure of the search string assembled. As discussed in Section 2.4, the recall score of the search string in relation to the QGS set is an appropriate form of evaluating the completeness of a search.

According to Zhang et al. (2011), in hybrid search strategies as SeSG, a minimum recall score of 70% in QGS documents is required after the search

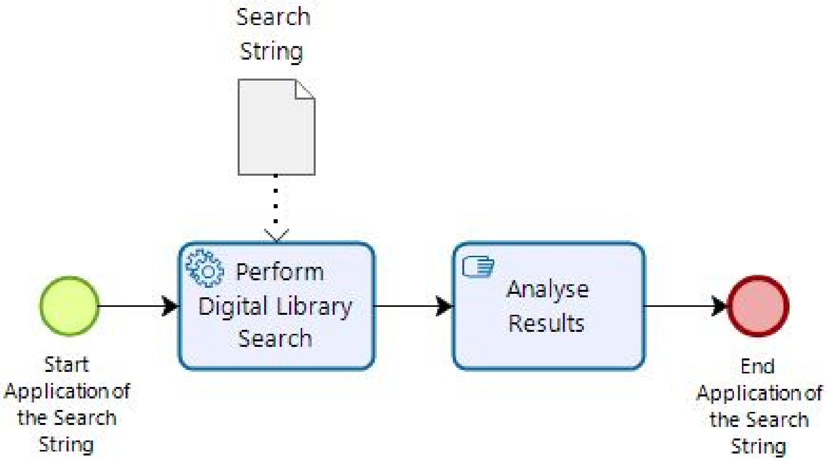

Search String
Perform Digital Library Search
Analyse Results
Start Application of the Search String
End Application of the Search String

Figure 4.7: The expanded 'Application of the Search String' sub-process of the SeSG approach.

in digital libraries using search strings. In SeSG, if the recall achieved in QGS documents after applying the search string generated is bellowing this minimum threshold, the user should go back to previous steps of SeSG and change one or more parameters of the process: the minimum document frequency of a term (Min_{DF}); the number of topics (K); the number of topic descriptors (W); or the number of words for topic descriptor enrichment (S).

On the other hand, if the generated search string achieves a QGS recall above the minimum threshold value, the automated search is concluded, and the user may proceed to the snowballing process. As a part of a hybrid search approach, SeSG does provide an optimized start set, leading to a faster and more efficient snowballing search.

In order to evaluate SeSG, in the next section, we present an experiment with complete hybrid searches using three different Secondary Studies. In these experiments, we evaluate the completeness of searches conducted with SeSG and simulate snowballing processes using Start Sets generated by SeSG.

SECTION

5

Experiment Process

This section presents the process applied to conduct the experimentation. All methodological tasks described in this experiment are based on well-known guidelines about empirical studies in SE (Wohlin et al., 2012; Kitchenham et al., 2015). The process, shown in Figure 5.1, follows the method proposed by Wohlin et al. (2012).

With the objective of the experiment defined, the next task was to plan the experiment. After the planning activity, the operation activity of SeSG was conducted, where automated searches were carried out in Scopus, using the automatically formulated search strings. We used Scopus because this digital library consistently delivered high values for precision (Mourão et al., 2020). Moreover, Scopus provides an API[1] that helps automation of the search. Specifically for SeSG, we applied a Python library, called pyscopus[2], to enable the use of the Scopus API. Finally, we carried out statistical analysis to understand the results obtained in the execution of the experiment.

Although Wohlin et al. (2012) uses the terms “phase” and “steps”, we prefer the hierarchical decomposition presented by Münch et al. (2012) that describes a process subdividing it into activities that, in turn, are subdivided into tasks. While representing a product of the presentation activity, this study describes scoping, planning, operation, and analysis & interpretation activities.

[1] https://dev.elsevier.com/

[2] https://github.com/zhiyzuo/python-scopus

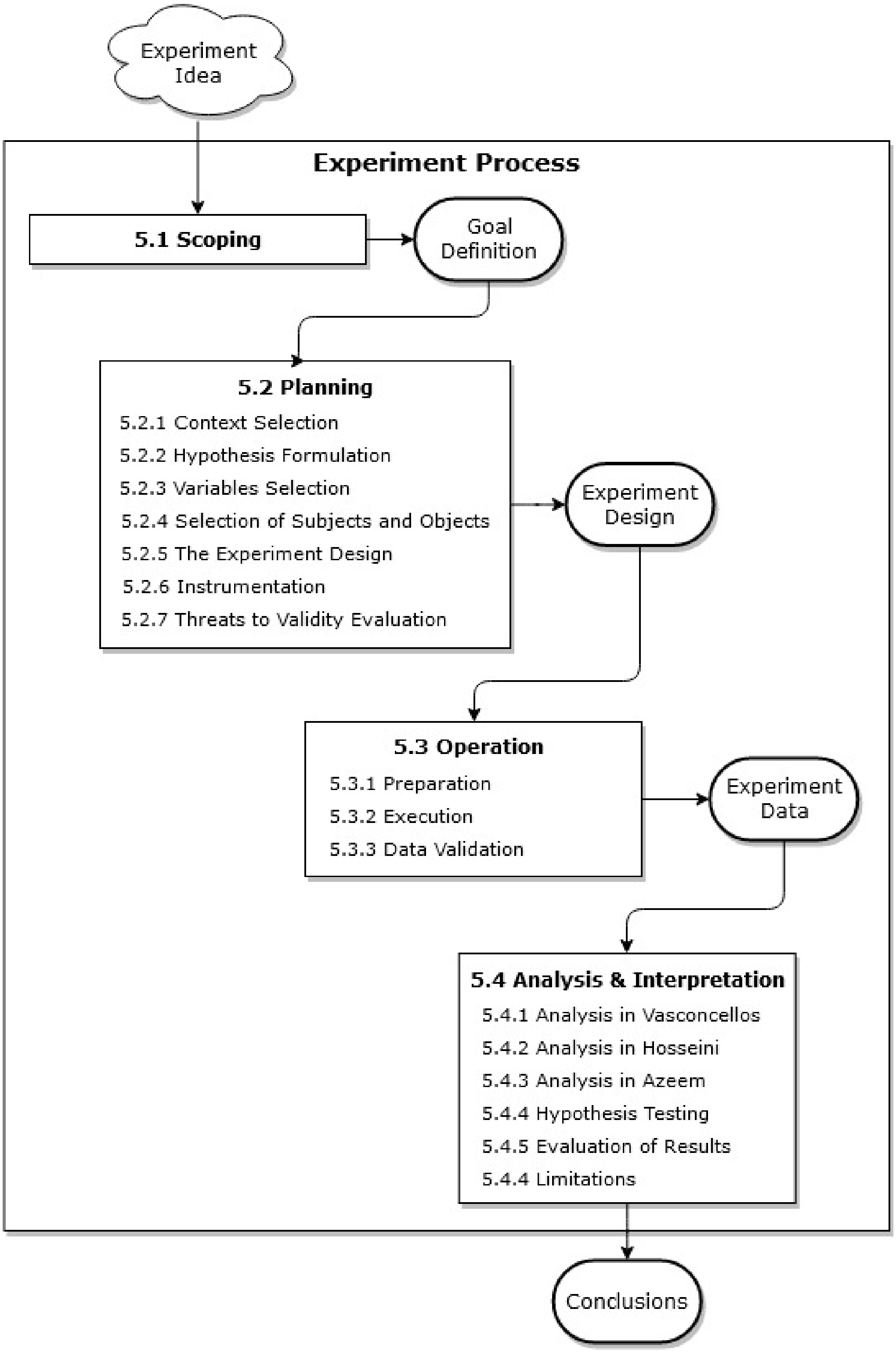


Figure 5.1: Overview of the experiment process. Based in Wohlin et al. (2012). The numbers correspond to sections in this study.

5.1 Scoping

The scope of the experiment is described using the goal template proposed in Wohlin et al. (2012). Our goal is:

Analyze **the results of the automatic search using SeSG**
for the purpose of **evaluation**
with respect to **effectiveness of the search**
from the perspective of **the researchers**
in the context of **secondary studies applying the snowballing strategy.**

5.2 Planning

The Scoping activity determines why the experiment is conducted, while the Planning activity outlines how the experiment is conducted.

According to Wohlin et al. (2012), the Planning activity has seven tasks. Based on the scope of the experiment, the **context selection** characterizes the environment of experimentation. Then, **hypotheses** are formulated, and independent and dependent **variables** are selected. The **selection of objects** and/or subjects is thereafter fulfilled. The **experimentation design** is chosen, and the **instrumentation** is prepared for the practical implementation of the experiment. Finally, an evaluation of the **threats to validity** anticipate factors that may put the results of the experimentation in question.

5.2.1 Context Selection

The experiment context demands online search trials of real SS. The subject is a researcher, and the experiment focuses on a specific scenario in which a hybrid search strategy is pertinent, and strict completeness is not required. The control group is a previously published SS (the object).

5.2.2 Hypothesis Formulation

The hypothesis test aims at verifying whether it is possible to reject a particular null hypothesis, H_0, based on a sample of some statistical distribution. If the null hypothesis is not rejected, nothing can be said about the result. In contrast, if the hypothesis is rejected, it can be said that the hypothesis is false, with a significance level (α) (Wohlin et al., 2012). The level of significance used in this experiment is $\alpha = 0.05$.

An one-tailed hypothesis (Trochim and Donnelly, 2020) was formulated to address the experiment scope described before. The hypotheses adopted for this experiment are as follows:

The **null hypothesis** for this study is:

H_0: As a result of the automatic search using the SeSG approach, there will be either no significant difference in effectiveness (measured as F1-Score on the Start Set) or there will be a significant decrease.

Specifically, the sample mean of F1-Score on the Start Set retrieved by automatically generated search strings - using the SeSG approach ($\mu AF1_{StS}$) - will be less than or equal to the F1-Score on the Start Set obtained by a manually generated search string ($MF1_{StS}$).

$$H_0 : \mu AF1_{StS} \leq MF1_{StS}$$

The null hypothesis is tested against the **alternative hypothesis**:

H_1: As a result of the automatic search using the SeSG approach, there will be a significant increment in search effectiveness.

In other words, **the sample mean of F1-Scores** obtained from a Start Set retrieved by an automatically generated search string using the SeSG approach ($\mu AF1_{StS}$) will be higher than the **F1-Score on Start Set obtained by a manually generated search string** ($MF1_{StS}$).

$$H_1 : \mu AF1_{StS} > MF1_{StS}$$

5.2.3 Variables Selection

In our experiment, the factor under study is the SeSG usage. The control group is an SS previously published, which applied the snowballing with a Start Set constructed by an automated search (a hybrid search). This object (the SS) must document the final set of selected studies (a retrieved Gold Standard - GS), from which we randomly take sets of studies for input to the experimental group (the SeSG usage).

To construct a search string, SeSG takes as input a set of known studies (a subset of the GS) that we typified as a Quasi-Gold Standard (QGS). Thus, the independent variable is the QGS set of studies, randomly selected from the studies present in the GS. Another independent variable is the size of the QGS. We blocked this variable for each tested object, as it affects the result of the experiment, but we are not interested in this effect. The SeSG approach suggests the size of the QGS (QGS_{Size}) within the range of 5 to 15 studies (see Section 4). Particularly, for each object studied in our experiment, we defined that the QGS_{Size} would be at least one third the size of the respective GS (until a maximum of 15 studies).

The dependent variable, derived from the hypothesis, is the F1-Score on the Start Set ($F1_{StS}$). This variable is not directly measurable, and we derive it using the following base metrics:

- GS_{Size}: size of the GS (the total of relevant studies from the object);
- N_{Total}: number of results retrieved from the search on Scopus;
- R_{StS}: number of GS studies retrieved from the search on Scopus (i.e., the relevant studies found in the Start Set).

We obtain the precision ($Precision_{StS}$) and recall ($Recall_{StS}$) of the Start Set, as shown in the Equations 5.1 and 5.2, respectively.

$$Precision_{StS} = \frac{R_{StS}}{N_{Total}} \tag{5.1}$$

$$Recall_{StS} = \frac{R_{StS}}{GS_{Size}} \tag{5.2}$$

With the precision and recall of the Start Set, we calculate the the F1-Score on the Start Set ($F1_{StS}$), using the Equation 5.3.

$$F1_{StS} = 2 \times \frac{Precision_{StS} \times Recall_{StS}}{Precision_{StS} + Recall_{StS}} \tag{5.3}$$

5.2.4 Selection of Subjects and Objects

Our experiment is characterized as technology-oriented because technical treatments are applied to objects (Wohlin et al., 2012). Nevertheless, as the experiment does not intend to study the influence of the subjects, it is recommended a design that sets aside the variability of different subjects (Juristo and Moreno, 2001). One of the authors of the SeSG approach was the only subject throughout the experiment. To minimize this specific threat to validity, we planned the maximum automation of the SeSG usage. The operation is mainly concerned with running automated scripts and collecting data.

To select objects, a search was conducted to find recent SS in the SE area. From 102 SLRs retrieved, we included 44 studies that applied snowballing following a database search as a hybrid strategy. We selected Vasconcellos et al. (2017) as a pilot because we had direct access to the authors. This pilot implied on a search replication of the original study.

Vasconcellos et al. (2017) presents an SS focused on approaches to the strategic alignment of software process improvement. The search strategy is a combination of database search and snowballing, and they assessed the search effectiveness with a QGS composed of ten studies. The database search resulted in 517 studies retrieved from seven digital libraries. The replication, almost three years later, retrieved 1082 results (in 2019). The increase of N_{Total} affects the precision metric and decreases the $F1_{StS}$ as a result.

Also, we decided to select other recently published objects. From the 44 identified SS, we filtered eight studies published in 2019. Then we selected Azeem et al. (2019) because it presents the best recall of the Start Set (see Equation 5.2), and Hosseini et al. (2019) because it presents the higher F1-Score on the Start Set (see Equation 5.3). As we select the objects of the experiment in a non-random way, it may me characterized as a quasi-experiment. For the purpose of simplicity, in this work we use to the word "experiment" to refer to a "quasi-experiment".

Azeem et al. (2019) describes an SLR on ML techniques for code smell detection. The search strategy is a combination of database searches with snowballing, but they did not calculate the search effectiveness. The database search retrieved 2456 studies from six different digital libraries. The Start Set has 12 studies (R_{StS}) and the GS has 15 studies (GS_{Size}).

Hosseini et al. (2019) developed an SLR about cross-project defect prediction. They also applied a search strategy that uses database search and snowballing. The database search resulted in 1889 studies from five different digital libraries. The Start Set has 29 studies (R_{StS}) and the final data set (the GS) contains 46 studies (GS_{Size}).

We did not replicate the studies because we did not have enough information that would allow the replication. However, both studies are recently published, and replication only affects the $F1_{StS}$ due to the increase of N_{Total}. We argue that if we reject the H_0 with the original $F1_{StS}$, we would also reject the H_0 with any subsequent search results.

5.2.5 Experiment Design

The experiment design explains how tests are organized and performed. In this task, we determine how many tests the experiment should perform to ensure that the treatment effect is visible (Wohlin et al., 2012). The type of project used in the experiment design is the one factor - the method used to formulate a search string - with two treatments - the 'manual' approach used in the control group and the SeSG approach used in the experimental group.

The hypothesis has a single value - the $MF1_{StS}$ - for each object since its measure derived directly from the study performed (past evidence). Therefore, the experiment involves comparing a sample mean ($\bar{X}$) with a specified value (μ_0). For this specific case, where the variance is unknown, it is recommended to use the test known as the **one-sample t-test** (Montgomery, 2017). The validity of the test relies on the assumption that the population distribution is at least approximately normal. A Shapiro-Wilk test (Shapiro and Wilk, 1965) on pilot confirmed the data normal distribution.

Moreover, the type II error probability (β) for the t-test depends on the distribution of the statistic test. So we have to choose an adequate sample size (n). We pilot on $n = 10$ to find β smaller than 0.05 (the probability of rejecting H_0 becoming approximately 0.95). We applied the Operating Characteristic Curve presented by Montgomery and Runger (2018) in Appendix A (Chart g) that shows different values of n for the one-sided t-test for a level of significance $\alpha = 0.05$. The Chart plots β for the t-test against a parameter d for various sample sizes n. The parameter d is calculated according to Equation 5.4.

$$d = \frac{|\bar{X} - \mu_0|}{s} \tag{5.4}$$

Table 5.1: The pilot $n = 10$ sample evaluation. $\bar{X}$ is the sample mean of F1-Score on the Start Set retrieved by the n automatically generated search strings, μ_0 is the F1-Score on the Start Set from the object, and s is the sample standard deviation of F1-Score on the Start Set retrieved by the n automatically generated search strings.

	n	$\bar{X}$	μ_0	s	d	β
Vasconcellos (Replicated)	10	0.043	0.027	0.014	1.183	≈ 0.03

The pilot sample size evaluation results are presented in Table 5.1. We derived an approximate $\beta = 0.03$ and conclude that a sample size of $n = 10$ is adequate to provide the desired sensitivity. After the execution of the experiment on each object selected, we tested whether ten trials were sufficient and the results are further detailed.

As mentioned before, we blocked some variables to avoid confounding factors. In this experimentation, the SeSG approach works on a fixed range of parameters that we described in the next section.

5.2.6 Instrumentation

Some parameters of the SeSG approach may oscillate, varying the generated search string produced. After carrying out some initial tests on the pilot object, we proposed value ranges for the approach parameters. These parameters oscillate in a coordinated manner, as shown in Table 5.2. Min_{DF} is the minimum frequency of documents required for the terms found to be valid, K is the number of topics in the search string, W is the number of words in each topic (topic descriptors) of the search string and S is the number of similar words that will be obtained for each word, during the search string enrichment.

Table 5.2: Range of parameters of the SeSG approach. Min_{DF} is the minimum frequency of documents required for the terms found to be valid, K is number of topics that will be obtained during the execution of the LDA algorithm, W number of words that will be obtained during the execution of the LDA algorithm and S is the number of similar words that will be obtained for each word, during the search string enrichment.

Instrument Variables	Value/ Range	Numerical Increment
Min_{DF}	0.1 - 0.4	0.1
K	1 - 5	1
W	5 - 10	1
S	0 - 3	1

In addition to the oscillating parameters of the SeSG approach, other instruments present in the experiment are objects, guides and output measures. The objects of this experiment were defined in Section 5.2.4. Although only one researcher conducted the entire experiment, being the only subject, a guide was created to ensure that the operation was followed during execution on all objects. This guideline contained the steps to be followed during the information extraction from objects, the execution of automated scripts, and the collection of output data.

Measurements in the experiment are performed via data collection. The data collected from the experiment were stored in spreadsheets. The search strings generated by SeSG and used in the tests were stored in text files, for future manual analysis. Co-citation graphs were also automatically generated, showing the relationship between the studies of the GS. In this graph, and edge is added if a study cites or is cited by another study in the GS.

The output spreadsheets containing the results of the tests have two measurements: N_{Total} and R_{StS}. The first measurement, N_{Total}, quantifies the results found in Scopus (chosen digital library) with the automated search. This measurement is compared with the amount of results found by all the digital libraries used in the object study. R_{StS}, on the other hand, represents the amount of GS studies found using the search string provided by SeSG. This value reflects the effectiveness of the search string in retrieving GS studies. In the context of hybrid searches, the evaluation of only the final result may be misleading and may not reflect the real contribution of the automated search. For example, in an extreme case where the co-citation graph of the GS is fully connected, the retrieval of just one study of the GS in the automatic search leads to the retrieval of the entire GS through snowballing.

The other measurements were obtained through specific calculations on the collected data. The dependent variable $F1_{StS}$ is the harmonic mean between the precision and recall of the start set. This variable is used for the mathematical comparison between the research previously carried out by the authors of the objects and the results obtained with the tests. In turn, the $Recall_{Final}$ measurement represents the spectrum comparison of the second stage of the hybrid search strategy, checking if all studies present in GS were found after all the search strategy. The value of $Recall_{Final}$ becomes important since when using only one digital library in the experimental group (Scopus), there is a risk that a relevant study will not found.

5.2.7 Threats to Validity Evaluation

Wohlin et al. (2012) propose a checklist with possible threats to validity as characterized by Campbell and Cook (1979). We use this checklist to describe which threats apply to the experiment and how we planned to mitigate each threat.

Conclusion Validity

The statistical conclusion validity focuses on the relationship between treatment and outcome of the experiment (Wohlin et al., 2012). In this type of threat the validity, we point out three possible threats:

1. *Fishing and the error rate:* To avoid the possibility of "fishing" the best result, we performed experimental tests ten randomized trials on each object with different sets of QGS.

2. *Reliability of treatment implementation:* To minimize this threat, all the tests on different objects, were performed in the shortest possible time (preferably in the same week). This was done to prevent treatments from occurring on different occasions, as the digital library used (Scopus) could change suddenly.

3. *Random irrelevancies in experimental setting:* To reduce the possibility of this threat, we executed all tests without interruption, ensuring similar conditions between the tests. Tests with interruptions were discarded.

Internal Validity

The threat to internal validity symbolizes the issue that the treatment really causes the effect. In this type of threat to the validity, a possible threat was detected:

1. *History:* Although there is a difference in times between the conduction of the groups (the control group was conducted years ago), this difference does not interfere with the analysis. The execution of the control group in a more recent timestamp would worsen $F1_{StS}$, contributing to the rejection of the null hypothesis. We argue that if we manage to refute the null hypothesis with the original data, we would also refute in an eventual replication.

Construct Validity

Construct validity ensures that the treatment reproduces the cause's construct, and the output reproduces the cause of the effect (Wohlin et al., 2012). Two possible construct validity threats were detected:

1. *Mono-operation bias:* If an experiment includes a single subject, it can under-represent the construct, not providing a complete picture of the theory. This experiment has only one subject; however, measures to avoid the subject's direct impact have been taken, mainly to the automation of a major part of the operation activity. A protocol for the experiment's execution was created, seeking to minimize a possible bias originated by the subject.

   Besides, despite the experiment having two independent variables, only one was manipulated (the QGS set), while the other (QGS_{Size}) was fixed. This was done precisely to observe the effect that the variation of the QGS causes on the formulated string, without manipulating the size of the QGS between the tests.

2. *Restricted generalizability across constructs:* The analysis of the effectiveness of the search string is based on the $F1_{StS}$ obtained. However, $Recall_{Final}$ is also relevant, since a search string qualified as efficient may not return certain relevant studies ($Recall_{Final}$ less than 100%). We will then tolerate a lower degree of completeness, as suggested by Dieste et al. (2009).

External Validity

Threats to external validity represent the capacity to generalize the experiment results in the real world. The last threat was detected:

1. *Interaction of selection and treatment*: To avoid this threat, objects with different sizes of GS were selected, addressing different contexts and made by different authors. Despite having, for each object, enough samples to generalize the statements found, through the hypothesis test, the number of objects used in the experiment may not be sufficient to generalize the effectiveness of the SeSG.

After the planning activity of the experiment is complete, it can finally be performed. The following subsection will present the operation of the experiment, showing how its execution was carried out.

5.3 Operation

We will divide the experiment's operational activity into three stages: preparation, execution, and data validation. For this experiment, the preparation activity is where the three selected objects are analyzed, and their relevant information and characteristics are filtered to be used in the execution. In the execution activity, SeSG is performed several times on each object, with pre-established treatment variations. Finally, data validation takes place, where specific data obtained automatically by SeSG are checked manually.

5.3.1 Preparation

Once the studies to be used in the experiment are selected, some information is needed so that the execution is finally running. First, for each object, all items present in the GS must be available. These studies will be further used in the co-citation graph. The metadata of the GS studies, used during the formation of the bag-of-words, are also necessary. Finally, a file with the titles of the GS studies is necessary to make an automated comparison between the studies found in the search and the GS.

In other words, the execution requires PDFs and metadata of the studies present in the respective GS and a list with the names of those studies. Moreover, certain characteristics of the objects must be saved for future measurements, such as the size of the QGS (QGS_{Size}) and the size of the GS (GSS).

5.3.2 Execution

The experiment's execution task is based on running the experiment in the three studies selected ten times, providing a random QGS for each of these runs. If the SS already uses any QGS, SeSG will perform imitating the QGS used by the author. The idea of running ten times aims to reduce the possibility of some bias in selecting random QGS, besides providing a mean of the dependent variables returned. The entire experimental group was run on an Intel(R) Xeon(R) ES-2620 @ 2.00GHz. Some previous tests for delimiting the operating range and checking functionality were performed on an Intel(R) Core(TM) i7-8850U @ 1.80GHz.

A framework for the execution task of the experiment is presented in Figure 5.2. This framework characterizes how the experiment is divided into

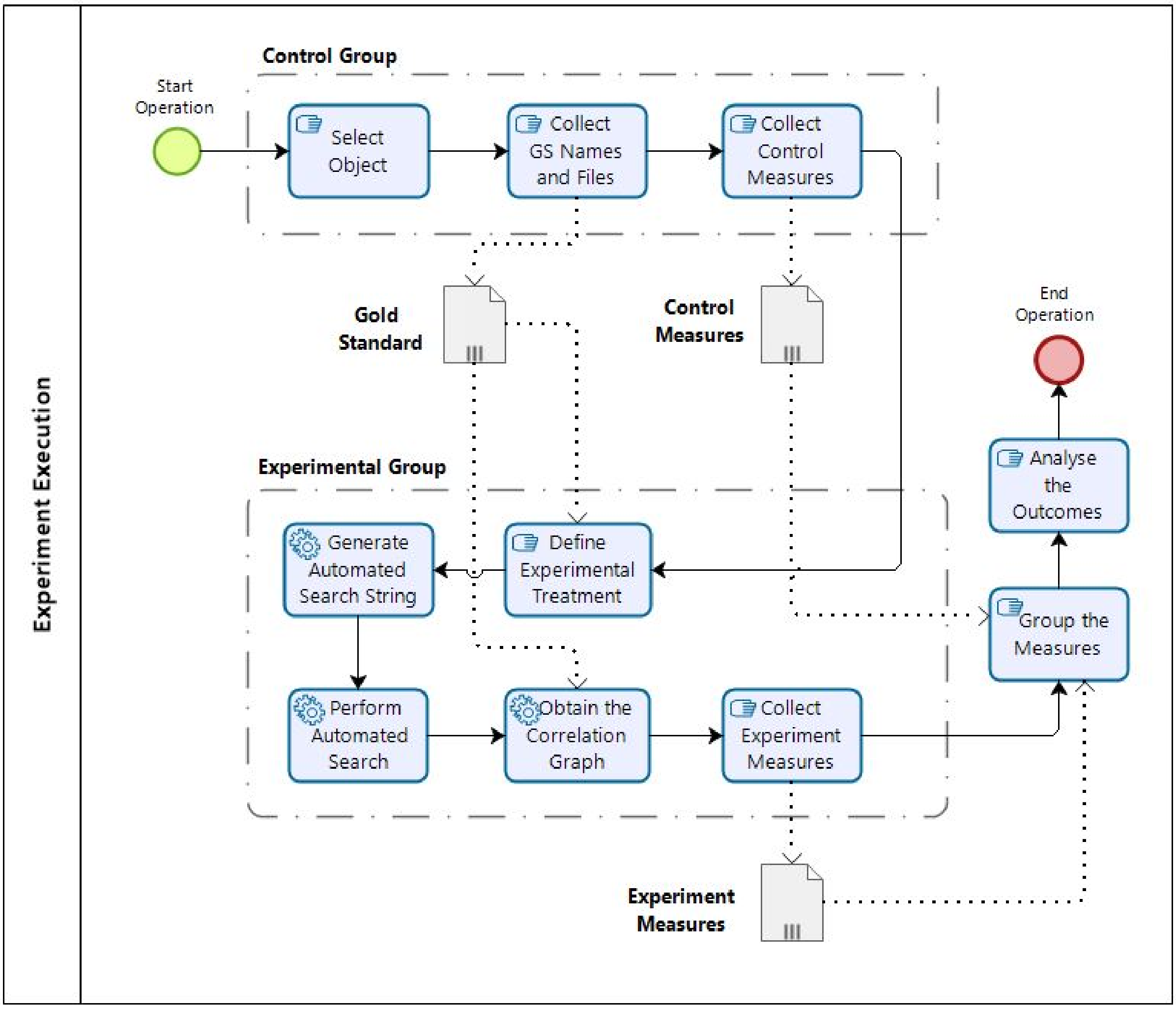


Figure 5.2: The framework for the execution task of the experiment. Based on Kitchenham et al. (2015).

two groups, control, and experiment. In the control group, the objects of the experiment are selected. For each of the selected objects, their GS is extracted, obtaining their names and files. Then, the hybrid search's relevant control measures are collected to allow later comparison between the approaches.

In the experiment group, some actions are different. First, we already have the GS and we use it to formulate a random QGS. With the metadata of this formulated QGS, added to some application parameters, we were able to formulate an automated search string. Using this automated search string in Scopus, we obtain a start set of relevant studies that answer the research questions. From this formulated start set, we assembled a co-citation graph between the GS studies. This graph allows to detect missing GS studies that would be found by snowballing. Similarly, measures are obtained that enable the comparison between groups, and an analysis of the experiment can be formulated.

All the variations in the application parameters are performed on the objects in an automated way via a Python script. Each execution generates two output files. The first file (in .txt format) contains a collection of search strings formulated from the tests. The second file (a worksheet in .csv format) contains the mathematical results themselves, such as the number of results found by the search and the number of GS studies found before and after the co-citation graph. That information found in the second file is mathematically sufficient to get the dependent variable. These results are automatically saved in multiple worksheets and files, named according to the application's parameters.

Besides these output files, a graph is also generated for each oscillation of the application's parameters, representing the connections between the GS studies. The type of vertex in the graph delimits at which stage of the search strategy, the GS study in question was found, whether in base search or snowballing. GS studies that were not found by the SeSG are characterized too in the graph. Since a large part of this output data is generated in an automated way, it is essential to perform data validation.

5.3.3 Data Validation

When data is collected, it is necessary to check if this data is reasonable and collected correctly. In this experiment's case, to ensure data validation, the search strings with the highest effectiveness (which generate the best results) are manually validated in Scopus. Besides, the co-citation graph generated in an automated manner from these most effectively search strings is also manually checked, ensuring data correctness.

5.4 Analysis and Interpretation

After the operation of the experiment, a detailed analysis of the results should be evidenced. For this analysis, the best results obtained by SeSG were examined and compared with the results obtained by the search string developed manually by the objects' authors. For the sake of space limitations, only the most relevant results were exposed in the study. There is a repository on GitHub[3] with more complete results, in addition to the Python code used during the experiment.

During each test, specific parameters that delimit the structure of the search string (Min_{DF}, K, W and S) fluctuate within a range of performance. This oscillation allows search strings of different sizes, having terms with different relevance to be tested. Thereby, for each test, 480 search strings and their respective results were generated. Given that there are 30 tests in total (ten for each object), 14400 search strings are obtained with their respective results.

[3] https://github.com/sesg-creator/SeSG

A mathematical ranking was performed to classify the generated search strings according to their efficiency. For this, each test result was ordered first by the $Recall_{Final}$, followed by the $F1_{StS}$. This order is adopted so that results with highest $Recall_{Final}$ can appear at the top (the search string should return as many GS studies as possible). Then, among the search strings with same $Recall_{Final}$, the most effective is the one with the best $F1_{StS}$.

With this classification, the best results for each test are found. The comparison with the search string results carried out manually formulated by the authors of the objects can then take place. The most efficient test results are shown in Tables 2, 3, and 4. These results can be partially analyzed in the tables, but their analysis can be further developed. The observation of the output graphics and their respective search strings will be used to complement this analysis.

5.4.1 Analysis in Vasconcellos et al. (2017)

The Table 5.3 presents the relevant data obtained through the execution of experiment on the Vasconcellos et al. (2017) object. The first part of the table reports the control group of the experiment. It presents the data obtained during the execution of the original study and the recent replication of the search strategy.

For comparison purposes, the measurements obtained during the replication of the search strategy will be used. In this case, the search string formulated by the authors results in 1082 studies located in different digital libraries. In addition, this search string provides 15 of the 30 studies present in the GS (i.e., the R_{StS} has 15 studies), with an $F1_{StS}$ of 0.027 and a $Recall_{Final}$ of 100%.

The second part of Table 5.3 reports about the experimental group of the experiment. It presents the data obtained during the execution of the SeSG approach in the Vasconcellos et al. (2017) object, as shown in Figure 5.2. The lines with the input set varying from QGS 01 to 10 represent the values of the most efficient search strings in each of the ten tests.

In the case of the Vasconcellos et al. (2017) study, all the tests presented a $Recall_{Final}$ of 97%, while the third test had the best $F1_{StS}$ (0.063), extremely superior to the used in the control group (0.027). Thus, although in some cases the $F1_{StS}$ is lower than the obtained originally (tests 6 and 7). However, the mean $F1_{StS}$ obtained by the tests with SeSG (0.043) is higher than that obtained by the original search string (0.027).

As noted earlier, there was a possibility that certain studies would not be found at all, because they are not indexed by Scopus and not cited by any of the other GS studies. In the case of Vasconcellos et al. (2017) object, this

Table 5.3: Relevant data obtained through the execution of the experiment on the Vasconcellos et al. (2017) object. The table has two parts. The first reports the control group of the experiment. Exclusively on this object, are presented measures of the original and the replicated search strategy, both with the original QGS. The second part shows the experimental group. The most effective results from ten tests (01-10) are reported, each having a different random QGS from the final GS. Lastly, the mean of these ten tests was also displayed. Min_{DF} is the minimum frequency of documents required for the terms found to be valid, K is number of topics that will be obtained during the execution of the LDA algorithm, W number of words that will be obtained during the execution of the LDA algorithm and S is the number of similar words that will be obtained for each word, during the search string enrichment. N_{Total} is the number of results retrieved from the search on Scopus, R_{StS} is the number of GS studies retrieved from the search on Scopus, $F1_{StS}$ is the F-Score of start set, and $Recall_{Final}$ is the final recall.

Control Group

Vasconcellos et al. (2017) / $GS_{Size} = 30$	N_{Total}	R_{StS}	$F1_{StS}$	$Recall_{Final}$
Original Automated Search	517	15	0.054	1.00
Replicated Automated Search	1082	15	**0.027**	**1.00**

Experimental Group

$InputSet$	Min_{DF}	K	W	S	N_{Total}	R_{StS}	$F1_{StS}$	$Recall_{Final}$
QGS 01	0.1	5	8	1	384	12	0.058	0.97
QGS 02	0.2	5	6	1	814	18	0.042	0.97
QGS 03	0.4	5	9	1	474	16	0.063	0.97
QGS 04	0.3	5	8	1	554	16	0.054	0.97
QGS 05	0.2	5	7	1	903	17	0.036	0.97
QGS 06	0.2	5	8	1	1341	18	0.026	0.97
QGS 07	0.1	3	5	2	1321	18	0.026	0.97
QGS 08	0.4	5	7	0	1184	17	0.028	0.97
QGS 09	0.1	5	8	2	665	18	0.051	0.97
QGS 10	0.2	4	7	1	718	18	0.048	0.97
Mean (01-10)	0.2	5	7	1	836	17	**0.043**	**0.97**

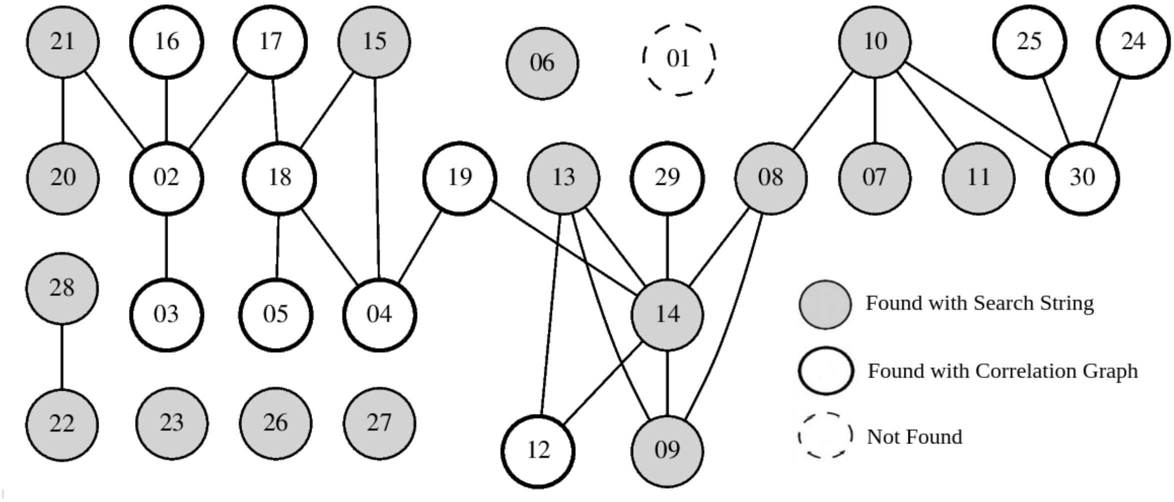


Figure 5.3: Graph representing the third random QGS experiment presented in Table 5.3 from the Vasconcellos et al. (2017) object. Filled vertex represents GS studies found with the use of search string in digital libraries. Vertex with bold outline represents GS studies founded with the correlation between the studies. Dashed vertex represent studies of the GS that would not be found by the hybrid strategy.

problem occurs with a study (Waina (2001)). Despite being a negative point for the search, the missing study in question would not have much impact on the conclusion of the SS. Vasconcellos et al. (2017) reduces the relevance of Waina (2001) in the final result because it is an approach without associated empirical evidence and its influence on other studies is 0.

The interpretation of the experimental tests can be complemented with the analysis of the graphs obtained during the experimental tests. Figure 5.3 shows one of the output graphs obtained during the experimentation in the Vasconcellos et al. (2017) object. The graph explicitly represents the GS of the third test (QGS 03) presented in Table 5.3. This graph has the best $F1_{StS}$ obtained from the results with a random QGS (in this case, the random QGS was formed by the studies 02, 06, 07, 10, 13, 14, 21, 22, 27, 30).

Note that in Figure 5.3, the node enumerated as 01 is dashed, which symbolizes that the search strategy did not find it (because of this the $Recall_{Final}$ showed in Table 5.3 is no more than 97%). Also, the nodes listed as 06, 07, 08, 09, 10, 11, 13, 14, 15, 20, 21, 22, 23, 26, 27, 28 are filled, which represents that all these studies were found with the automated search string formulated by SeSG. The search string used in this particular case is as follows:

TITLE-ABS-KEY(((“improvement” OR “ideal”) AND (“business” OR “objective”) AND (“process” OR “quality”) AND (“process improvement”) AND (“spi”) AND (“software” OR “business”) AND (“model” OR “process”) AND (“goals” OR “thinking”) AND (“software process improvement”)) OR ((“process” OR “quality”) AND (“software” OR “business”) AND (“strategic” OR “design”) AND (“process improve-

ment") AND ("improvement" OR "ideal") AND ("software process") AND ("software process improvement") AND ("goals" OR "thinking") AND ("strategy" OR "model")) OR (("software" OR "business") AND ("software process") AND ("improvement" OR "ideal") AND ("process" OR "quality") AND ("software process improvement") AND ("process improvement") AND ("model" OR "process") AND ("organizations" OR "communities") AND ("based" OR "modeled")) OR (("process" OR "quality") AND ("software" OR "business") AND ("software process") AND ("approach" OR "guide") AND ("development" OR "design") AND ("goals" OR "thinking") AND ("engineering" OR "development") AND ("process improvement") AND ("improvement" OR "ideal")) OR (("measurement" OR "process") AND ("software" OR "business") AND ("engineering" OR "development") AND ("software engineering") AND ("spi") AND ("strategy" OR "model") AND ("strategies" OR "management") AND ("organization" OR "communities") AND ("development" OR "design"))) AND PUBYEAR < 2015

On the other hand, the nodes 02, 03, 04, 05, 12, 16, 17, 18, 19, 24, 25, 29, 30 in Figure 5.3 are bold outline, representing that these studies from GS were found through correlation of the vertices of the graph in question. It is relevant to this graph analysis since the SeSG was modeled to act in hybrid search strategies. This demonstrates that although the search string does not return the entire GS, it will be achieved in the snowballing.

5.4.2 Analysis in Hosseini et al. (2019)

Table 5.4 presents the relevant data obtained from the experiment execution in Hosseini et al. (2019) object. As with Vasconcellos et al. (2017), the first part of the table represents the control group of the experiment. In the Hosseini et al. (2019) object case, the automated search adopted by the authors results in 1889 studies located in different digital libraries. In addition, this search string provides 29 of the 46 studies present in the GS (i.e., the R_{StS} has 29 studies), with an $F1_{StS}$ of 0.029 and a $Recall_{Final}$ of 100%.

The second part of Table 5.4 also reports about the experimental group of the experiment. The lines with the input set varying from QGS 01 to 10 represent the values of the most efficient search strings obtained in each of the ten tests. Analyzing the data obtained during the tests shows that the value of the $F1_{StS}$ fluctuates considerably (between 0.047 and 0.475). However, in all tests, the values of $F1_{StS}$ are higher than those presented by the original search strategy (0.029). This also makes the mean of the $F1_{StS}$ values, presented in the last row of the Table 5.4, higher than that presented by the original automated search (0.043 against 0.027).

As with Vasconcellos et al. (2017) object, the $Recall_{Final}$ obtained during the

Table 5.4: Relevant data obtained through the execution of the experiment on the Hosseini et al. (2019) object. The table has two parts. The first reports the control group of the experiment, presenting the measures of the original search strategy. The second part shows the experimental group. The most effective results from ten tests (01-10) are reported, each having a different random QGS from the final GS. Lastly, the average of this ten tests was also displayed. Min_{DF} is the minimum frequency of documents required for the terms found to be valid, K is number of topics that will be obtained during the execution of the LDA algorithm, W number of words that will be obtained during the execution of the LDA algorithm and S is the number of similar words that will be obtained for each word, during the search string enrichment. N_{Total} is the number of results retrieved from the search on Scopus, R_{StS} is the number of GS studies retrieved from the search on Scopus, $F1_{StS}$ is the F-Score of start set, and $Recall_{Final}$ is the final recall.

Control Group

Hosseini et al. (2019) / $GS_{Size} = 46$	N_{Total}	R_{StS}	$F1_{StS}$	$Recall_{Final}$
Original Automated Search	1889	29	**0.029**	**1.00**

Experimental Group

$InputSet$	Min_{DF}	K	W	S	N_{Total}	R_{StS}	$F1_{StS}$	$Recall_{Final}$
QGS 01	0.4	3	8	0	101	22	0.299	0.96
QGS 02	0.4	2	8	0	38	19	0.452	0.96
QGS 03	0.2	3	8	1	200	20	0.162	0.96
QGS 04	0.1	4	6	3	1018	25	0.047	0.96
QGS 05	0.4	4	10	0	34	11	0.275	0.96
QGS 06	0.1	4	9	1	41	19	0.436	0.96
QGS 07	0.4	4	6	0	252	27	0.181	0.96
QGS 08	0.4	5	7	0	153	26	0.261	0.96
QGS 09	0.1	4	7	3	51	23	0.474	0.96
QGS 10	0.1	5	5	0	363	21	0.102	0.96
Mean (01-10)	0.3	4	7	1	225	21	**0.269**	**0.96**

tests in Hosseini et al. (2019) object did not reach the expected value, being limited to 96%. This limitation is because two studies were not found during the Scopus search stage and the co-citation graph (Pravin and Srinivasan (2012); Wang and Wang (2016)). This symbolizes that these studies are possibly not indexed in Scopus. It also shows that no other GS study references or is referenced by them.

As commented earlier, studies not indexed in Scopus and not referenced by the set that forms the GS in the SS in question cannot be attributed by SeSG. Wang and Wang (2016) was not found because Hosseini et al. (2019) object states that the studies contained in GS were published until 2015, but Wang and Wang (2016) was published in 2016. This information conflict causes that filter in the search string (PUBYEAR < 2016) to remove Wang and Wang (2016) from the search results. Pravin and Srinivasan (2012) was not found because it is in a gray area not reached by Scopus, being indexed only in ProQuest and not cited by any other study besides Hosseini et al. (2019) study.

During the SS quality assessment stage, Hosseini et al. (2019) excluded both studies not founded with SeSG. Wang and Wang (2016) fails during context assessment, while Pravin and Srinivasan (2012) fails in more than one step of the assessment. This does not minimize the severity of two studies not being found but mitigates the impact this could have on a real case.

A graph with the higher $F1_{StS}$ (QGS 09) in the tests is provided. Because it has a larger GS (46 studies) and such studies have much communication, this output graph will not be presented due to its illegibility. This high connectivity between the GS studies shows that the area in question of the study of Hosseini et al. (2019) is extremely consolidated.

However, specific data regarding the graph generated with SeSG from the study of Hosseini et al. (2019) can be presented. The graph has 46 nodes, each representing a GS study, with two dashed nodes (06 and 12). Also, the graph with the higher $F1_{StS}$ result has 23 filled nodes, representing the studies found during the database search step and 21 bold outline nodes representing items correlated to the filled nodes. The search string used in this particular case is as follows:

TITLE-ABS-KEY(((“prediction” OR “.” OR “observed” OR “risk”) AND (“models” OR “##s” OR “designs” OR “systems”) AND (“project” OR “border” OR “country” OR “fault”) AND (“defect” OR “error” OR “cost” OR “model”) AND (“cross” OR “in” OR “within” OR “small”) AND (“cross project”) AND (“projects” OR “elements” OR “samples” OR “countries”)) OR ((“cpdp”) AND (“projects” OR “elements” OR “samples” OR “countries”) AND (“metric” OR “forecast” OR “diagnostic” OR “architecture”) AND (“software” OR “product” OR “manufacturing” OR “vehicle”) AND (“sets”) AND (“defect data”) AND (“data” OR “required” OR “experience”

OR "provided")) OR (("data" OR "required" OR "experience" OR "provided") AND ("defect" OR "error" OR "cost" OR "model") AND ("company" OR "sectional" OR "platform" OR "border") AND ("prediction" OR "." OR "observed" OR "risk") AND ("cc" OR "") AND ("cc data") AND ("wc" OR "")) OR (("prediction" OR "." OR "observed" OR "risk") AND ("data" OR "required" OR "experience" OR "provided") AND ("project" OR "border" OR "country" OR "fault") AND ("defect" OR "error" OR "cost" OR "model") AND ("defect prediction") AND ("cross" OR "in" OR "within" OR "small") AND ("cross project"))) AND PUBYEAR < 2016

As in Vasconcellos et al. (2017), practically half of the GS studies in Hosseini et al. (2019) were obtained by database searching using the search string displayed. Another half will be obtained with the co-citation graph, highlighting the importance of using a hybrid search strategy.

5.4.3 Analysis in Azeem et al. (2019)

Lastly, Table 5.5 presents the data obtained during the execution of the experiment in Azeem et al. (2019) object. This particular case has a much lower GS than previous cases (only 15 studies), using the QGS_{Size} floor in random QGS (only five studies). Thus, perhaps metadata from only five studies may have impacted the quality of SeSG in formulating a bag-of-words that represents the field of study in question.

As done with the other objects, Table 5.5 has two parts. The first part also represents the control group, showing the values used during the SS's original execution. In the case of Azeem et al. (2019) object, the automated search resulted in 2456 studies, located in different digital libraries. Of the GS formed with only 15 studies, 12 are returned in this automated search, while the rest were obtained via snowballing. These measures provides an R_{StS} of 12 and an $F1_{StS}$ of 0.009.

The second part of Table 5.5 also reports about the experimental group. The lines with the input set varying from QGS 01 to 10, representing the values of the most efficient search strings obtained in each of the ten tests. Analyzing the data obtained during the tests, it can be noted that there is a different variation in N_{Total}. As the QGS used for the formulation of the search string is very small (5 studies), the impact of the studies selected to form the QGS is higher than that observed in other objects. This leads to such a considerable variation between the N_{Total}, ranging from 3 results to 2676.

This oscillation obtained in the number of results returned (N_{Total}), directly impacts the $F1_{StS}$. As a result, there is also a big fluctuation between the values obtained by $F1_{StS}$, ranging from 0.007 to 0.608. However, only in one

Table 5.5: Relevant data obtained through the execution of the experiment on the Azeem et al. (2019) object. The table has two parts. The first reports the control group of the experiment, presenting the measures of the original search strategy. The second part shows the experimental group. The most effective results from ten tests (01-10) are reported, each having a different random QGS from the final GS. Lastly, the mean of this ten tests was also displayed. Min_{DF} is the minimum frequency of documents required for the terms found to be valid, K is number of topics that will be obtained during the execution of the LDA algorithm, W number of words that will be obtained during the execution of the LDA algorithm and S is the number of similar words that will be obtained for each word, during the search string enrichment. N_{Total} is the number of results retrieved from the search on Scopus, R_{StS} is the number of GS studies retrieved from the search on Scopus, $F1_{StS}$ is the F-Score of start set, and $Recall_{Final}$ is the final recall.

Control Group								
Azeem et al. (2019) / $GS_{Size} = 15$					N_{Total}	R_{StS}	$F1_{StS}$	$Recall_{Final}$
Original Automated Search					2456	12	**0.009**	**1.00**
Experimental Group								
$InputSet$	Min_{DF}	K	W	S	N_{Total}	R_{StS}	$F1_{StS}$	$Recall_{Final}$
QGS 01	0.3	2	5	2	817	9	0.021	1.00
QGS 02	0.1	3	7	0	6	5	0.476	1.00
QGS 03	0.1	5	5	1	2676	10	0.007	1.00
QGS 04	0.3	1	8	3	656	7	0.020	1.00
QGS 05	0.3	3	8	0	8	7	0.608	0.80
QGS 06	0.3	4	10	0	9	7	0.583	0.80
QGS 07	0.3	5	10	0	3	3	0.333	1.00
QGS 08	0.3	5	8	1	249	10	0.075	1.00
QGS 09	0.1	5	10	0	6	4	0.381	1.00
QGS 10	0.1	4	9	0	3	3	0.333	1.00
Mean (01-10)	0.2	4	8	1	443	7	**0.284**	**0.96**

test (QGS 03), the value of $F1_{StS}$ (0.007) is not higher than that presented by the original search strategy (0.029). This makes the mean of the $F1_{StS}$, presented in the last row of the Table 5.5, much higher than that presented by the original automated search (0.284 against 0.009).

One negative detail that further represents the oscillation between the tests is the $Recall_{Final}$ variation. Contrary to what happens in Tables 5.3 and 5.4, the $Recall_{Final}$ present in QGS 05 and 06 are distinct from the others. This recall oscillation results in a certain subgroup of studies not being reached in all cases, making the search string limited to only part of the GS provided. Figure 5.4 shows what happens. The sub-graph with nodes 07, 14, and 15 was not reached by SeSG in these tests, limiting the final recall obtained.

This segregation of this sub-graph with the other studies of the GS occurs due to the research focus. Nodes 07, 14, and 15 are the only GS studies that seek to investigate the code smell aimed at duplicate codes. Thus, these studies are only related to each other, and if the search string returns none of these studies, they will never be reached through the co-citation graph.

This highlights a threat to the SeSG approach. If the area in question of the SS has particular sub-areas, at least one study of each sub-area must be obtained by the search string. This would ensure that all research sub-areas have been reached. This is exactly what does not happen in the tests with QGS 05 and 06. The lack of completeness in the sub-areas made it possible to exclude part of the GS.

Even with this drawback that two tests presented in Table 5.5 did not reach the expected $Recall_{Final}$, the results are impressive. The evidence that the refinement of the search string does not always have to be so profound. With a hybrid approach, snowballing can find most studies missing without excessive search string refinement.

As previously mentioned, an output from the Azeem et al. (2019) object is shown in Figure 5.4. The GS from this study has a relatively small size compared to the others (15 studies), making graph analysis more visible. In this case, the graph explicitly represents the GS of the second test (QGS 02). This graph was chosen because it presents the best $F1_{StS}$ obtained with a $Recall_{Final}$ of 100%. This specific test was realized with a random QGS formed by studies 01, 02, 06, 11, 15.

Unlike the other objects, in this situation, no node shown in Figure 5.4 is dashed, since the search strategy found all studies present in GS. Besides, five nodes are filled (01, 06, 10, 11, 15), representing that they were found in digital libraries. The remaining ten nodes are bold outline (02, 03, 04, 05, 07, 08, 09, 12, 13, 14), found from the graph correlations. The search string used in this particular case is as follows:

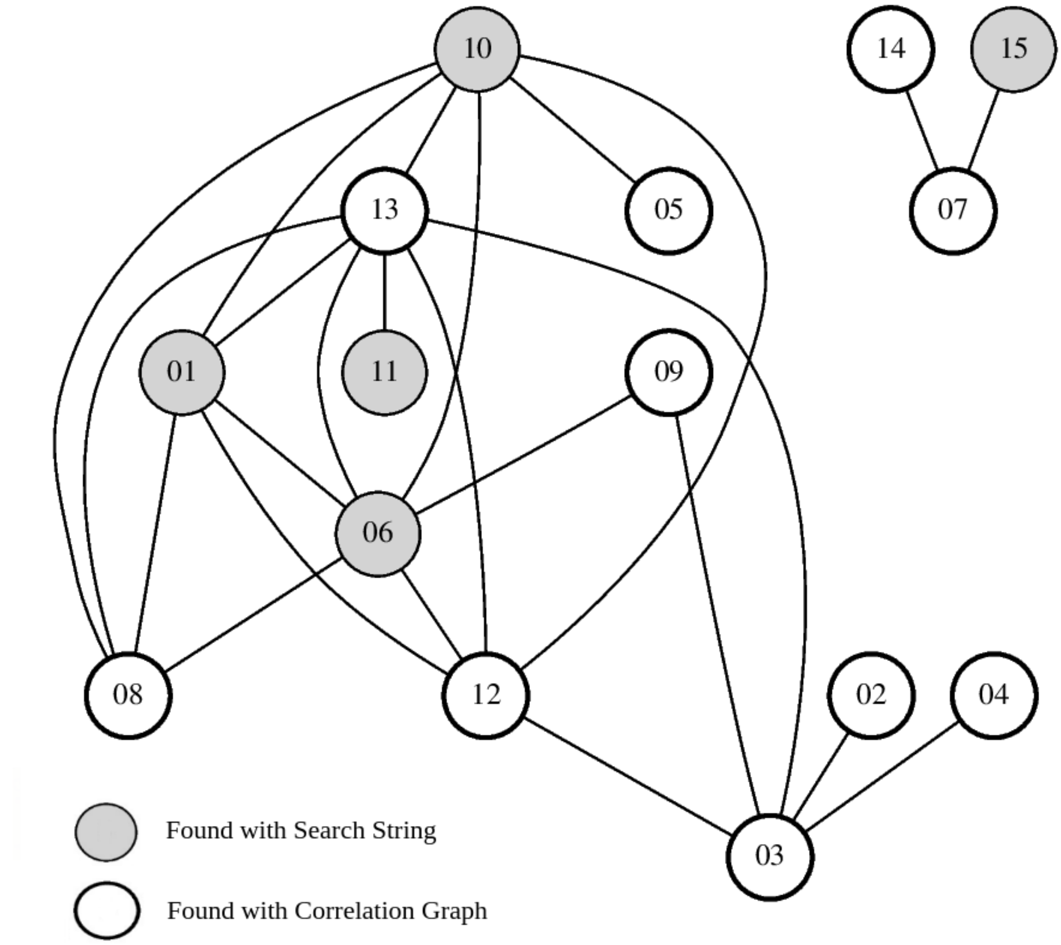


Figure 5.4: Graph representing the second random QGS experiment presented in Table 5.5 from the Azeem et al. (2019) study.

TITLE-ABS-KEY((“code” AND “clones” AND “clone” AND “model” AND “code clones” AND “classification” AND “learning”) OR (“anti” AND “patterns” AND “anti patterns” AND “smurf” AND “systems” AND “software” AND “occurrences”) OR (“design” AND “detection” AND “software” AND “code” AND “smells” AND “different” AND “learning”)) AND PUBYEAR < 2018 AND PUBYEAR > 1999

Note that the search string has the floor and ceiling date filter. This filter is used to match the spectrum reached by the search string that the authors reached at the time of publication, making the comparison fairer. Thus, as the author limited himself to studies from 2000 onwards, the search string was incremented with a date filter seeking to correspond to this limitation.

From all these analyzes, both numerically in the tables and visually through the graphs, it is evident that they are promising despite the results oscillating much. Considering the work that researchers usually have refining a search string, SeSG can minimize the difficulty of finding studies in digital libraries. The time saved that SeSG provides can be used in other steps of the process or to reduce an SS's execution time.

5.4.4 Hypothesis Testing

The analysis was performed to test the hypothesis individually for each object. For each object of the experiment, a check is made if the treatments represent a normal distribution, through the Shapiro-Wilk test (Shapiro and Wilk, 1965). Table 5.6 presenting the values referring to the Shapiro-Wilk test performed on each object of the experiment and their respective results. It is interesting to note that the values used in the samples are in Tables 5.3, 5.4 and 5.5. Moreover, all the objects are normal distributions.

Table 5.6: Values referring to the Shapiro-Wilk test of normality, performed on each of the objects. If p-value is greater than α (0.05), the distribution can be considered normal. n is the sample size, W is the test statistic from Shapiro-Wilk, p-value is the significance probability and α is the level of significance.

	Vasconcellos (Replicated)	Hosseini (Original)	Azeem (Original)
n	10	10	10
Statistic (W)	0.92	0.93	0.87
p-value	0.36	0.52	0.11
α	0.05	0.05	0.05
Conclusion	**Normal**	**Normal**	**Normal**

Since all the objects in the experiment are parametric, the one-sample Student's t-test (Montgomery, 2017) is performed for each object to test the presented hypothesis. To test the null hypothesis that the population mean is equal to a specified value μ_0, the Equation 5.5 is used. Table 5.7 shows all the values used in Equation 5.5 for each object and their respective results. For information, $\bar{X}$ is the sample mean, μ_0 is the F1-Score provided by the object when using the manual search string, s is the sample standard deviation, and n is the sample size.

$$t = \frac{\bar{X} - \mu_0}{\frac{s}{\sqrt{n}}} \tag{5.5}$$

As can be seen in Table 5.7, the null hypothesis (H_0) was rejected in favor of the alternative hypothesis in all objects of the experiment. This emphasizes the premise that the use of the automated search string, formulated by SeSG, is, on mean, better than the manual formulation of a search string.

To analyze the sample size, we applied the technique presented in Section 5.2.5 (Montgomery and Runger, 2018). The results, presented in Table 5.8,

Table 5.7: Values referring to the Student's t-test, performed on each of the objects. If p-value is less than α (0.05), the null hypothesis can be rejected. $\bar{X}$ is the sample mean, μ_0 is the F1-Score provided by the object when using the manual search string, s is the sample standard deviation, n is the sample size, t is the one-sample t-test value, and p-value is the significance probability.

	Vasconcellos (Replicated)	Hosseini (Original)	Azeem (Original)
$\bar{X}$	0.043	0.269	0.284
μ_0	0.027	0.029	0.009
s	0.014	0.149	0.237
n	10	10	10
t	3.741	5.079	3.668
p-value	0.0023	0.0003	0.0025
H_0	**Rejected**	**Rejected**	**Rejected**

shows that the sample $n = 10$ was sufficient to carry out the experiment on all objects, assuring a probability of a type II error less than 0.05.

Table 5.8: Values referring to the probability of a type II error (β) from an O.C. Curve from $\alpha = 0.05$. If β is less than 0.05, H_0 will be rejected with probability at least 0.95. $\bar{X}$ is the sample mean, μ_0 is the F1-Score provided by the object when using the manual search string, s is the sample standard deviation, d is the type II error parameter, and n is the sample size.

	Vasconcellos (Replicated)	Hosseini (Original)	Azeem (Original)
$\bar{X}$	0.043	0.269	0.284
μ_0	0.027	0.029	0.009
s	0.014	0.149	0.237
d	1.183	1.606	1.160
n	10	10	10
β	≈ 0.03	0.00	≈ 0.04

5.4.5 Evaluation of Results

After analyzing the data obtained during the experiment and perform hypothesis testing, some assessments can be made about what it was got. Despite the number of samples used in the hypothesis test for the objects to be small, the p-values obtained have a considerable distance for the level of significance determined. This distance guarantees a certain tranquility to reject the null hypothesis H_0.

The main concern with data analysis is in the $Recall_{Final}$ obtained, where it did not always reach 100% as in the control group. However, this problem was predictable from the moment that the SeSG uses only one digital library, unlike the control groups that employ five or more. Specifically in the Azeem et al. (2019) object, the small number of studies used in the formulation of the QGS possibly had an impact on obtaining relevant terms for the formulation of the search string. As a result, the choice of studies to form the QGS had a more incisive impact on the search, generating large fluctuations between the tests.

It is evident that although the SeSG is an evolving approach, its results are promising. The SeSG behaved satisfactorily on objects with different sized GS (15, 30 and 46), using randomized QGS with proportional sizes (5, 10 and 15). It was also clear that the SeSG must necessarily be used in hybrid search strategies since the completeness when performing only the automated search with the formulated search string is low.

5.4.6 Limitations

Some limitations of the experiment were evident during the process. First, the generalization of the experiments results can be compromised due to the relatively small number of objects in the experiment. Although the objects try to portray the field of action in SE, three objects may not be enough for a complete generalization.

Another limitation is related to the comparisons made. The experiment aims at comparing the efficiency of the search strings formulated with other methods for search strategy formulation. However, no comparison is made with SeSG and other automated or semi-automated search string formulation approach.

Also, the use of the automated search string in only one digital library (Scopus) can be considered a limitation. This usage limitation restricts the search spectrum to only a series of studies indexed by the study base. Although this base of studies is large, it does not imply that all relevant studies on a given topic will be found.

SECTION

6

Conclusion and Future Work

In this study, we presented an approach, called SeSG, that generates search strings for SS using only a list of relevant studies selected by researchers (QGS). In SeSG, the QGS metadata is processed following a TM process and LDA is used used to extract and group relevant terms. Besides, a BERT word embedding improves the search string by adding similar terms to those extracted in the metadata. The proposed approach then assembles the search string with the selected terms and automatically executes it in digital libraries.

To evaluate the efficiency of the SeSG application, an experiment was conducted. Three SS were used as objects of the experiment, all of them having a hybrid search strategy. The experiment results suggest that SeSG provided higher F1-Score on the Start Set than manually generated search strings.

However, the generalizability across the experiment's constructs are still limited. The number of objects used during experimentation is not sufficient to completely generalize the proposed approach. Thus, a possible future work consists in replicating the experiment on a more significant number of objects, seeking to observe the behaviour of SeSG in SS conducted in different ways, with different sizes.

In this study, we did not focus on achieving full completeness in the search, given that our premise was to meet the balance between precision and recall (the use of the F1-Score explicitly this). Due to this fact, in some objects of the experiment the GS was not entirely retrieved. Thus, we point as future work the investigation of different TM techniques for the generation and enrichment of search strings that may establish search strings focusing on completeness (100% final recall). The use of a pre-trained BERT embedding model in a scientific study base can help to find enriched terms that provide the completeness of the search. Finally, we plan to perform experiments using more digital libraries.

References

Aggarwal, C. C. and Zhai, C. X. (2012). *Mining text data.* Springer Science+Business Media.

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB)*, volume 1215, p. 487–499.

Ali, N. B. and Usman, M. (2018). Reliability of search in systematic reviews: Towards a quality assessment framework for the automated-search strategy. *Information and Software Technology*, 99:133–147.

Ampatzoglou, A., Bibi, S., Avgeriou, P., Verbeek, M., and Chatzigeorgiou, A. (2019). Identifying, categorizing and mitigating threats to validity in software engineering secondary studies. *Information and Software Technology*, 106:201–230.

Arampatzis, A., Van Der Weide, T. P., van Bommel, P., and Koster, C. H. (1999). Linguistically-motivated information retrieval. *Encyclopedia of Library and Information Science*, 69:201–222.

Azeem, M. I., Palomba, F., Shi, L., and Wang, Q. (2019). Machine learning techniques for code smell detection: A systematic literature review and meta-analysis. *Information and Software Technology*, 108:115–138.

Babar, M. A. and Zhang, H. (2009). Systematic literature reviews in software engineering: Preliminary results from interviews with researchers. In *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*, p. 346—-355. IEEE Computer Society.

Badampudi, D., Wohlin, C., and Petersen, K. (2015). Experiences from using snowballing and database searches in systematic literature studies. In *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering*, p. 1–10.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Booth, A. (2016). Searching for qualitative research for inclusion in systematic reviews: A structured methodological review. *Systematic reviews*, 5(1):74.

Campbell, D. T. and Cook, T. D. (1979). *Quasi-experimentation: Design and Analysis Issues for Field Settings*. Houghton Mifflin Company.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, p. 160–167.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Cooper, C., Booth, A., Varley-Campbell, J., Britten, N., and Garside, R. (2018). Defining the process to literature searching in systematic reviews: A literature review of guidance and supporting studies. *BMC medical research methodology*, 18(1):85.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 4171–4186.

Dieste, O., Grimán, A., and Juristo, N. (2009). Developing search strategies for detecting relevant experiments. *Empirical Software Engineering*, 14(5):513–539.

Dieste, O. and Padua, A. G. (2007). Developing search strategies for detecting relevant experiments for systematic reviews. In *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*, p. 215–224.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). The kdd process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11):27–34.

Feng, L., Chiam, Y. K., and Lo, S. K. (2017). Text-mining techniques and tools for systematic literature reviews: A systematic literature review. In *2017 24th Asia-Pacific Software Engineering Conference (APSEC)*, p. 41–50.

Ghafari, M., Saleh, M., and Ebrahimi, T. (2012). A federated search approach to facilitate systematic literature review in software engineering. *International Journal of Software Engineering & Applications (IJSEA)*, 3(2):13–24.

Gonzalez, M. A. I., de Lima, V. L. S., and de Lima, J. V. (2006). Tools for nominalization: An alternative for lexical normalization. In *Computational Processing of the Portuguese Language*, p. 100–109.

Grames, E. M., Stillman, A. N., Tingley, M. W., and Elphick, C. S. (2019). An automated approach to identifying search terms for systematic reviews using keyword co-occurrence networks. *Methods in Ecology and Evolution*, 10(10):1645–1654.

Hosseini, S., Turhan, B., and Gunarathna, D. (2019). A systematic literature review and meta-analysis on cross project defect prediction. *IEEE Transactions on Software Engineering*, 45(2):111–147.

Imtiaz, S., Bano, M., Ikram, N., and Niazi, M. (2013). A tertiary study: Experiences of conducting systematic literature reviews in software engineering. In *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*, p. 177–182.

Jones, K. S. and Willett, P. (1997). *Readings in information retrieval*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Juristo, N. and Moreno, A. M. (2001). *Basics of software engineering experimentation*. Springer Science & Business Media.

Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004):1–26.

Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. *Technical Report – Department of Computer Science, University of Durham*.

Kitchenham, B. A., Budgen, D., and Brereton, P. (2015). *Evidence-based software engineering and systematic reviews*. Chapman & Hall/CRC.

Kitchenham, B. A., Li, Z., and Burn, A. J. (2011). Validating search processes in systematic literature reviews. In *EAST*, p. 3–9.

Krovetz, R. (1993). Viewing morphology as an inference process. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 191–202.

Laguna, M. d. S. C., Pardo, T. A. S., and Rezende, S. O. (2014). *Extração automática de termos simples baseada em aprendizado de máquina.* Doctoral thesis in ciências de computação e matemática computacional, Instituto de Ciências Matemáticas e de Computação, University of São Paulo, São Carlos, SP.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. (2019). Biobert: A pre-trained biomedical language representation model for biomedical text mining. *arXiv preprint arXiv:1901.08746.*

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165.

Manning, C., Raghavan, P., and Schütze, H. (2010). Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103.

Marcos-Pablos, S. and García-Peñalvo, F. J. (2018). Information retrieval methodology for aiding scientific database search. *Soft Computing*, p. 1–10.

Marshall, C. and Brereton, P. (2013). Tools to support systematic literature reviews in software engineering: A mapping study. In *2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, p. 296–299.

Marshall, C., Brereton, P., and Kitchenham, B. (2014). Tools to support systematic reviews in software engineering: A feature analysis. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering.* Association for Computing Machinery.

McCann, B., Bradbury, J., Xiong, C., and Socher, R. (2017). Learned in translation: Contextualized word vectors. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, p. 6297–6308.

Mergel, G. D., Silveira, M. S., and da Silva, T. S. (2015). A method to support search string building in systematic literature reviews through visual text mining. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, p. 1594–1601. Association for Computing Machinery.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., and Joulin, A. (2018). Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation.* European Language Resources Association (ELRA).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, volume 2, p. 3111–3119.

Montgomery, D. C. (2017). *Design and analysis of experiments*. John wiley & sons.

Montgomery, D. C. and Runger, G. C. (2018). *Applied statistics and probability for engineers*. John Wiley & Sons.

Mourão, E., Kalinowski, M., Murta, L., Mendes, E., and Wohlin, C. (2017). Investigating the use of a hybrid search strategy for systematic reviews. In *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, p. 193–198.

Mourão, E., Pimentel, J. F., Murta, L., Kalinowski, M., Mendes, E., and Wohlin, C. (2020). On the performance of hybrid search strategies for systematic literature reviews in software engineering. *Information and Software Technology*, p. 106–294.

Münch, J., Armbrust, O., Kowalczyk, M., and Soto, M. (2012). *Software Process Definition and Management*. Springer Publishing Company, Incorporated.

Nogueira, B. M. (2009). *Avaliação de métodos não-supervisionados de seleção de atributos para mineração de textos*. Doctoral thesis in ciências de computação e matemática computacional, Instituto de Ciências Matemáticas e de Computação, University of São Paulo, São Carlos, SP.

Nogueira, B. M. (2013). *Hierarchical semi-supervised confidence-based active clustering and its application to the extraction of topic hierarchies from document collections*. Masters dissertation in ciências de computação e matemática computacional, Instituto de Ciências Matemáticas e de Computação, University of São Paulo, São Carlos, SP.

O'Mara-Eves, A., Thomas, J., McNaught, J., Miwa, M., and Ananiadou, S. (2015). Using text mining for study identification in systematic reviews: A systematic review of current approaches. *Systematic reviews*, 4(1):5.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, p. 1532–1543.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, p. 2227–2237.

Petitti, D. B. et al. (2000). *Meta-analysis, decision analysis, and cost-effectiveness analysis: Methods for quantitative synthesis in medicine*. Number 31. OUP USA.

Pravin, A. and Srinivasan, S. (2012). Detecting of software bugs in source code using data mining approach. *National Journal of System and Information Technology*, 6(1):1–8.

Rezende, S. O. (2003). *Sistemas inteligentes: Fundamentos e aplicações*. Editora Manole Ltda, Barueri, SP.

Ros, R., Bjarnason, E., and Runeson, P. (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*, p. 118–127. Association for Computing Machinery.

Salton, G., Wong, A., and Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Shapiro, S. S. and Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611.

Shearer, C. (2000). The crisp-dm model: The new blueprint for data mining. *Journal of Data Warehousing*, 5(4):13–22.

Smalheiser, N. R., Lin, C., Jia, L., Jiang, Y., Cohen, A. M., Yu, C., Davis, J. M., Adams, C. E., McDonagh, M. S., and Meng, W. (2014). Design and implementation of metta, a metasearch engine for biomedical literature retrieval intended for systematic reviewers. *Health Information Science and Systems*, 2(1):1.

Stansfield, C., O'Mara-Eves, A., and Thomas, J. (2017). Text mining for search term development in systematic reviewing: A discussion of some methods and challenges. *Research Synthesis Methods*, 8(3):355–365.

Tomassetti, F., Rizzo, G., Vetro, A., Ardito, L., Torchiano, M., and Morisio, M. (2011). Linked data approach for selection process automation in systematic reviews. In *15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE 2011)*, p. 31–35.

Trochim, W. M. and Donnelly, J. P. (2020). Research methods knowledge base. (version current as of 27 April 2020).

Tsafnat, G., Glasziou, P., Choong, M. K., Dunn, A., Galgani, F., and Coiera, E. (2014). Systematic review automation technologies. *Systematic reviews*, 3(1):74.

Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, p. 384–394. Association for Computational Linguistics.

van Rijsbergen, C. (1979). Information retrieval. (accessed: 07.07.2020).

Vasconcellos, F. J., Landre, G. B., Cunha, J. A. O., Oliveira, J. L., Ferreira, R. A., and Vincenzi, A. M. (2017). Approaches to strategic alignment of software process improvement: A systematic literature review. *Journal of Systems and Software*, 123:45–63.

Waina, R. (2001). A business goal-based approach to achieving systems engineering capability maturity. In *20th DASC. 20th Digital Avionics Systems Conference (Cat. No. 01CH37219)*, volume 1. IEEE.

Wang, J. and Wang, Q. (2016). Analyzing and predicting software integration bugs using network analysis on requirements dependency network. *Requirements Engineering*, 21(2):161–184.

Weiss, S. M. and Indurkhya, N. (1998). *Predictive data mining: A practical guide*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Weiss, S. M., Indurkhya, N., Zhang, T., and Damerau, F. (2010). *Text mining: Predictive methods for analyzing unstructured information*. Springer Publishing Company, Incorporated, 1st edition.

Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. Association for Computing Machinery.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in software engineering*. Springer Publishing Company, Incorporated.

Zhang, H., Babar, M. A., and Tell, P. (2011). Identifying relevant studies in software engineering. *Information and Software Technology*, 53(6):625–637.

Zwakman, M., Verberne, L. M., Kars, M. C., Hooft, L., van Delden, J. J., and Spijker, R. (2018). Introducing palette: An iterative method for conducting a literature search for a review in palliative care. *BMC Palliative Care*, 17(1):1–9.