Information and Software Technology 98 (2018) 59–73

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

A comparison of automated training-by-example selection algorithms for Evidence Based Software Engineering

Edgar E. Hassler[a], David P. Hale*[,b], Joanne E. Hale[b]

[a] Department of Computer Information Systems, Appalachian State University, Boone, NC 28608, USA
[b] Department of Information Systems, Statistics, and Management Sciences, Culverhouse College of Commerce, The University of Alabama, Tuscaloosa, AL 35401, USA

ARTICLE INFO

Keywords:
Research infrastructure
Evidence Based Software Engineering
Systematic Literature Review
Systematic Mapping Studies
Culling
VSM
LSA
Recall
Precision
Document selection

ABSTRACT

Context: Study search and selection is central to conducting Evidence Based Software Engineering (EBSE) research, including Systematic Literature Reviews and Systematic Mapping Studies. Thus, selecting relevant studies and excluding irrelevant studies, is critical. Prior research argues that study selection is subject to researcher bias, and the time required to review and select relevant articles is a target for optimization.
Objective: This research proposes two training-by-example classifiers that are computationally simple, do not require extensive training or tuning, ensure inclusion/exclusion consistency, and reduce researcher study selection time: one based on Vector Space Models (VSM), and a second based on Latent Semantic Analysis (LSA).
Method: Algorithm evaluation is accomplished through Monte-Carlo Cross-Validation simulations, in which study subsets are randomly chosen from the corpus for training, with the remainder classified by the algorithm. The classification results are then assessed for recall (a measure of completeness), precision (a measure of exactness) and researcher efficiency savings (reduced proportion of corpus studies requiring manual review as a result of algorithm use). A second smaller simulation is conducted for external validation.
Results and conclusions: VSM algorithms perform better in recall; LSA algorithms perform better in precision. Recall improves with larger training sets with a higher proportion of truly relevant studies. Precision improves with training sets with a higher portion of irrelevant studies, without a significant impact from the training set size. The algorithms reduce the influence of researcher bias and are found to significantly improve researcher efficiency.

To improve recall, the findings recommend VSM and a large training set including as many truly relevant studies as possible. If precision and efficiency are most critical, the findings suggest LSA and a training set including a large proportion of truly irrelevant studies.

1. Introduction

Evidence Based Software Engineering (EBSE) is growing in importance and impact, aiding the maturity of the Software Engineering discipline by driving systematic, structured analysis, synthesis, and interpretation of empirical evidence [1]. Within EBSE, Systematic Literature Reviews (SLRs) and Systematic Mapping Studies (SMS) depend on the effective search and selection of primary research studies. While study search algorithms have been the focus for numerous researchers [2–7], study selection algorithms have received comparatively less attention [8]. Effective and efficient study selection (that is, selecting relevant studies and excluding irrelevant studies) is critical to the success of EBSE.

One of the main barriers in conducting SLRs and SMSs is the time-commitment associated with the selection of studies from the extensive set of results returned as part of the search stage of the protocol. In addition to the time required for the selection of studies to be included in the review, a common problem is the consistent application of the inclusion/exclusion criteria of the review [9]. While inconsistences in the application of the inclusion/exclusion criteria are resolved through the discussions among researchers, such inconsistences result in the expenditure of additional time to resolve the inconsistences and re-accomplish the selection process. This additional researcher time and effort may present a significant barrier to the undertaking of such research [10].

The contribution of this research is the validation of automated training-by-example classifier algorithms that are computationally simple, do not require extensive training or tuning, and ensure inclusion/exclusion consistency while reducing researcher study selection time expenditure. The algorithms are evaluated using a series of simulations.

* Corresponding author.
E-mail addresses: hassleree@appstate.edu (E.E. Hassler), dhale@ua.edu (D.P. Hale), jhale@cba.ua.edu (J.E. Hale).

https://doi.org/10.1016/j.infsof.2018.02.001
Received 1 April 2017; Received in revised form 3 January 2018; Accepted 1 February 2018
Available online 06 February 2018
0950-5849/

The remainder of this paper is organized as follows. Section 2 provides an overview of the selection process and associated metrics along with a summary of previous work. Section 3 details the conceptual research model and research questions. The Monte-Carlo Cross-Validation simulation approach is described in Section 4. Section 5 discusses the research findings. Section 6 provides a confirmatory case study to provide additional evidence that extends external validity. Section 7 includes the discussion, implications and limitations of the research. Finally, Section 8 concludes the research and describes future work.

2. Background

To frame and ground this research, this section describes the study selection process used by EBSE researchers, the metrics used to assess the success of the selection process, current tools and methods used to support study selection, as well as the new proposed automated training-by-example classifier algorithms.

2.1. EBSE process

EBSE research studies follow the following process [1,11]:

1. **Planning.** During this phase, the research objectives and questions are defined and the protocol is created. The protocol includes sources of primary research studies, search methods and keywords, study inclusion (relevant) and exclusion (irrelevant) criteria, study quality criteria, a data extraction form, and a data synthesis strategy.
2. **Execution**. During this phase, primary studies are obtained, evaluated, and analyzed.
   a. **Search Execution**. During this Execution step:
      i. First, during Initial Selection, primary studies are identified, collected, and organized in the document ***corpus.***
      ii. During **Selection Execution** (also called here **Study Selection**), studies are evaluated according to the inclusion and exclusion criteria, and **classified** as either **relevant** (included) or **irrelevant** (excluded). If quality criteria dictate, this is followed by additional review of the corpus to remove studies deemed below quality thresholds.
      iii. In Selection Review, the corpus is reviewed to minimize incorrect exclusion of relevant studies.
   b. **Information Extraction.** During this Execution step, relevant information is extracted from those studies classified as included.
   c. **Analysis & Synthesis.** During this phase, the results of the included studies are analyzed and synthesized to accomplish the original research objectives and questions. This phase is highly variant between SLR and SMS research projects.
3. **Documentation.** During this final phase, a report detailing the results and findings is prepared. The report provides a transparent, repeatable account of the study, explicates the results, and provides discussion around meaning, implications and limitations.

This research focuses on improving the **Selection Execution** step (2b above; called **Study Selection** hereafter in this research). This is often accomplished in a series of steps designed to progressively reduce the corpus down to the truly relevant studies and includes [12,13]:

a. Starting with corpus study titles and the predefined inclusion criteria, studies are removed that are clearly irrelevant to the research topic.
b. Abstracts of the remaining studies from step ***a*** are examined, removing studies that are clearly irrelevant to the research topic.
c. Using the set of included studies from step ***b***, the full text of the studies is screened again against the inclusion/exclusion criteria.

In many cases, researchers combine steps a and b above. Two or more researchers conduct these steps independently, with classification disagreements resolved by agreed-upon method, such as consensus or voting. The research then progresses to Information Extraction as discussed above.

2.2. Study selection metrics

The goal of Study Selection is to retain the relevant studies found during the search process while excluding irrelevant studies. Consequently, the effectiveness of the selection algorithm can be measured by the level of True Positive (TP) and True Negative (TN) classifications (studies correctly classified as include and exclude, respectively), False Positive (FP) errors (including a study that should be rejected) and False Negative (FN) errors (excluding a study that should be included).

The traditional proportions of each type of error (FP or FN) with respect to the total number of studies classified provide a general indicator of classifier performance, but may be misleading due to the asymmetry between the portions of relevant and irrelevant documents typically found in the corpus [13]. Therefore, alternative measures the field of information retrieval, recall and precision, are used here.

An indicator of the lack of FN errors is defined by the *Recall* statistic [13]:

$$Recall = \frac{TP}{TP + FN} \tag{1}$$

Thus, recall measures the percent of relevant studies that are retrieved, and is a measure of completeness. A maximum recall measure of 1.0 signifies the inclusion of all relevant studies, with recall measures less than one indicating increased FN errors.

An indicator of FP errors is found in the measure of precision [13]:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Thus, precision measures the percentage of retrieved documents that are relevant, and is an indicator of exactness. A maximum precision measure of 1.0 signifies the exclusion of all irrelevant studies, with measures less than one indicating an increase in the number of FP errors.

Research process efficiency is also an important metric to assess, as required research effort may be an impediment to the undertaking of such research and may reduce the accuracy of results due to fatigue. Using the approaches proposed in this research, the researcher analyzes a subset of the search corpus (hereafter referred to simply as a corpus) to train the classifier, deciding which studies within the training set are relevant (included) and irrelevant (excluded). Once the classifier is applied, the researcher completes the Study Selection step with only the algorithm-Included studies, manually removing those included erroneously by the classifier.

In this context, researcher efficiency savings are obtained by reducing the number of irrelevant studies that must be read by the researcher, by reducing the proportion of studies in the algorithm training set and reducing classifier FP errors. This is consistent with prior research which argues that the time required to review articles is a target for optimization [see 14,15]. Therefore, this research defines the efficiency savings as the percentage of study inclusion / exclusion decisions that the researcher must make in training the selection algorithm and within the classified include set.

$$Efficiency\ Savings\ = 1 - \frac{TSS + FP}{Corpus} \tag{3}$$

where:

- TSS = Size of the algorithm training set
- Corpus = the number of studies in the corpus, which is also equal to

Table 1
Hypothetical exemplars of classifier efficiency savings.

Exemplar description	# Studies in training set	# Studies automatically classified	# Irrelevant studies mis-classified	Efficiency savings
High efficiency savings – no irrelevant studies misclassified as relevant by the algorithm. Only additional studies manually reviewed are from the training set.	10	100–10 = 90	0	1–(10 + 0)/100 = 0.90 ***90% efficiency savings.***
Low efficiency savings – from large training set, and poor classification results.	50	100–50 = 50	40	1–(50 + 40)/100 = 0.10 ***10% efficiency savings***
Moderate efficiency savings. Manual review of all studies in the training set plus those irrelevant studies incorrectly included by the selection algorithm	10	100–10 = 90	20	1–(10 + 20)/100 = 0.70 ***70% efficiency savings***
No savings – from a totally ineffective classification of irrelevant studies and no relevant studies to be automatically classified. No improvement over completely manual selection process.	10	100–10 = 90	all	1–(10 + 90)/100 = 0 ***0% efficiency savings***

$$TP + TN + FP + FN$$

This measure of efficiency savings and this research does not account for the researcher time required to run the automated classifier.

The following hypothetical exemplars in Table 1 provide insights into relative meaning of the algorithm efficiency savings metric. In each case, the corpus holds a total of 100 studies.

2.3. Study selection support

Efforts to support study selection span several technologies. In the medical domain, linear systems that combine feature values assigned to studies are tested and found to provide a work savings in most cases of between 10% and 30% with a 95% recall rate [13].

Zhong [16] created an iterative process framework for refining the study inclusion and exclusion criteria; the refined criteria are then used to classify the next set of candidate studies. This process was found to significantly improve recall, but not precision.

2.3.1. Document clustering techniques

Document clustering techniques [17,18] group sets of documents based on the similarity of topics covered and are used for improving information retrieval effectiveness, organizing search results, and creating document taxonomies. Document clustering begins with the Bag of Words representation of each study or document, which consists of the document's unordered list of terms (or words), disregarding grammar [19,20]. Each Bag of Words representation is then translated into an N-dimensional vector in which each cell in the vector equates to the frequency count of a term used in the document. The Vector Space Model (VSM) represents a common algorithm basis for processing, comparing, and visualizing the document vector. VSM is proposed here as an automated study selection / classifier algorithm, and is further discussed in Section 2.4.

Hierarchical models [21] produce a tree of document clusters (a dendrogram), with a single, all-inclusive cluster at the top, intermediate clusters underneath, and individual singleton clusters at the bottom. The resulting dendrogram provides a corpus taxonomy or hierarchical index.

Partitional models (including VSM) build un-nested, single-level clusters of related studies [21]. One common clustering method is the K-means algorithm, in which a centroid is calculated representing the mean or median point of a study cluster. From this, the primary document topic(s) are determined, which must be assigned meaning by the researcher. Document clustering allows the researchers to focus on the core set of primary topics and reduces distractions from irrelevant terms.

2.3.2. Visual Text Mining

Visual Text Mining (VTM) is emerging as a popular and effective technique to support SLRs [22], building on and supporting the natural human power of visual information processing. VTM combines text mining methods (extracting patterns and knowledge from unstructured textual data) and data visualization tools (such as maps and graphs) to support interactive data exploration. [22]. Visual Text Mining is proposed as both a means of supporting the selection of studies during SLRs [14,15,23,24] and for the validation of the selection process [25].

The VTM tool Revis, introduced by Felizardo et al. [23], supports the entire SLR process, from planning to execution and reporting. Relevant here is the support of the Study Selection process, accomplished through document maps (clustering related documents in 2-dimensional space using content such as title, abstracts, and keywords), edge bundles (hierarchical tree visualization showing primary studies as nodes and citations among nodes as links) and citation networks (showing primary studies as central circles surrounded by and linked cited references, illustrating shared references among primary studies within the corpus). Through an original and a replicated experiment,

VTM was found to reduce the study selection time (regardless of researcher experience) and improve selection effectiveness for experienced researchers [22].

A related study explores the use of VTM during Selection Review within the Selection Execution step, conducted to validate classification decisions and reduce false negatives when the observed quality criterion demands [25]. The empirical case study found that content and citation maps reduced the time required to complete selection review, with no impact on accuracy.

Similarly, Malheiros et al. [15] proposes the use of Visual Text Mining to support SLRs through the Projection Explorer (PEx) tool. PEx includes document content maps (based on VSM or the Normalized Compression Distance model). Visualization pre-processing is also automated, including processing study abstracts, removing stop words, and calculating term frequencies. Their research found that VTM aided researchers in identifying erroneously classified and similar studies. VTM increased study selection efficiency and allowed the researchers to broaden their search algorithms to create a larger corpus, because the tool quickened the identification of irrelevant studies.

The State of the Art Through Systematic Review (StArt) tool [26] provides support for each SLR stage by organizing an SLR and automating many repetitive SLR tasks. Researchers import into StArt the SLR protocol and search results. StArt automatically computes each study's relevancy score based on similarity of its meta-data to the stored protocol; if studies with higher scores are not relevant, refinement of the search string is indicated. The Vector Processing Model is used to cluster related studies. The tool includes a robust set of data visualizations, including charts illustrating the percentage of studies included and excluded, trends in topic evolution over time, as well as content and citation maps.

The Visual Text Mining methods discussed here (such as content and citation maps) provide valuable imagery for the researcher, emphasizing study and corpus information critical to the EBSE task at hand [14,23,27]. Within the Study Selection task, VTM facilitates effective and efficient inclusion and exclusion decision making [25]. While this approach shows promise in assisting researchers in study selection, it is still time consuming and requires specialized training to be effective with the outcome remaining subject to researcher bias [see 15]. The decision to include or exclude a study still rests solely with the researcher through a manual, visualization supported, review of the study metadata or full text.

2.3.3. Automated document classification

Building on the foundation of visual and text-based EBSE support tools, automatic document classification methods are emerging. These techniques identify document features and/or patterns and use this information to automatically classify documents. The goal of these methods, as is the case in this research, is to effectively identify relevant documents while reducing the amount of time demanded of the researcher to manually review and exclude irrelevant documents [28–30].

A set of automated classifiers are ontology based, where an ontology is "an explicit specification of a shared conceptualization," [31] a single, standardized, collective representation of vocabulary and terms relevant to a central concept. Sun et al. [32] built a general empirical ontology, SLRONT, using structured abstracts of known relevant studies. SLRONT was then extended to a more detailed ontology, COSONT, for a cost estimation SLR. The empirical results of automated selection using COSONT was promising. The COSONT inclusion and exclusion selection decisions did not vary from the manual counterpart, but manual review and decision -making time was eliminated. However, significant time was required to build the ontology, the time tradeoff of which was not taken into consideration. Not specifically created for EBSE, Song et al. [28] created an ontology-based classifier for the categorization of web pages. Their algorithm categorized web page text by creating a VSM (further discussed in Section 2.4) based on weighted feature and category vectors.

Several automated classifiers are based on Bayesian models. The use of factorized complement naïve Bayes classification with weight engineering in systematic drug class reviews have provided an estimated workload reduction of 33.5% on average [33]. The Bayesian-based classifier proposed as both a means of supporting the selection of studies during SLRs and for the validation of the selection process. While this approach shows promise in assisting researchers in the selection process, it is still time consuming and requires specialized training to be effective with the outcome remaining subject to researcher bias.

Focusing on document clustering generally rather than EBSE specifically, Nigam et al. [30] first trains a classifier with known labeled documents. Probabilistically-weighted class labels are then assigned to each unlabeled document. A revised classifier is then trained by iterating through all the documents in the corpus, both the originally labeled and the formerly unlabeled. Classifier parameters are identified locally maximize the likelihood of all labeled and unlabeled documents.

The Linked Data Bayesian classifier proposed by Tomassetti et al. [34] starts by forming the Bag-of-Words (BoW) model of known relevant studies. For each of these studies, the BoW model is enriched by linking the keywords to DBpedia abstracts, a Resource Description Framework (RDF) repository where information stored in Wikipedia is represented as structured data. Each study in the corpus is then classified using its associated enriched BoW model and a Naive Bayes classifier. Those included by the Bayes classifier are then manually read and either included or excluded. The model is rebuilt using the revised Included set, and the process is repeated iteratively with the remaining corpus studies until no further studies are included. The case study resulted in a work load reduction of 20% and a recall of 100%.

Another class of automated classifiers are machine learning based. Although not specifically designed for EBSE research, Bloehdorn and Hotho [29] proposed the AdaBoost machine learning algorithm for text classification. Their approach utilizes Boosting, which combines multiple weak algorithms (that individually perform only slightly better than guessing) to form a powerful ensemble classifier. Their work found that Boosting Algorithms scale well to a large number of dimensions when used with binary feature representations and the appropriate ontology.

Wallace, Trikalinos, Lau, Brodley and Schmid [35] employ Support Vector Machines in combination with supervised learning and estimate the method reduces the number of studies to be reviewed by 50% with no loss of recall. Ramesh and Sathiaseelan [36] proposed advanced multi class instance selection as a means for improving the efficiency of a support vector machine. The Advanced Multi Class Instance Selection based support vector machine (AMCISSVM) algorithm is compared with Multi Class Instance Selection (MCIS) and Neighborhood Property based Pattern Selection (NPPS) algorithms. The AMCISSVM is found to outperform these algorithms with respect to classification accuracy, ratio of selected instances and time consumption.

Baysian and Machine Learning classifiers have been shown to significantly improve Study Selection. However, both types of classifiers require extensive training and tuning of the classifiers, which reduces the efficiency benefits, and impedes the undertaking of these important studies.

2.4. Proposed algorithms

The goal of this research is to identify and evaluate automated classifier algorithms that are computationally simple, do not require extensive training or tuning, and ensure inclusion/exclusion consistency while reducing researcher study review and selection time expenditure. The proposed algorithms use a *training-by-example* approach, in which the researcher presents several studies that exemplify what is to be included and excluded, and the automated classifier then selects similar studies. Two potential related bases for the construction of such a classifier are VSM and Latent Semantic Analysis (LSA), a

member of the VSM family of algorithms.

VSM–based algorithms have been frequently used in search protocols (as discussed earlier in this paper) and writing analysis/test grading [19,20,37–40], but to our knowledge not for automatic classification during study selection. Thus, this research represents a new application for these tools. These chosen algorithms are algebraic, analyzing the frequency of terms occurring within a study, and comparing the similarity of term patterns found across studies.

Probabilistic Latent Semantic Analysis (PLSA) utilizes joint probability in a statistical model for co-occurrence data to associate each observation with an unobserved (latent) class variable [41]. PLSA is a two level, generative model for the documents in the corpus that models each document as a probability distribution on a fixed set of topics. Latent Dirichlet Allocation (LDA) extends PLSA with the addition of a Dirichlet prior distribution for document topic and word topic distributions [42]. LDA is a three level model that models the probability of topics in a document and the probability of a word given a topic. This research does not examine the PLSA or LDA models due to the fixed topic assumptions, as the number of topics within a corpus generated by a keyword search is unknown and computationally complex/expensive to identify.

2.4.1. The Vector Space model

The VSM algorithm begins with the Bag of Words representation of each study or document, which consists of the document's unordered list of terms (or words), disregarding grammar [19,20]. Each Bag of Words is then translated into an N-dimensional vector in which each cell in the vector equates to the frequency count of a term used in the document [39].

The collection of document vectors – in the form of column vectors – are then aggregated to form a matrix in which the columns represent the studies in the corpus and the row vectors represent the frequency of terms utilized in the corpus for each study [19,38,39]. Thus each study is represented in t dimensions, where t is the number of unique terms in the entire corpus. This matrix is referred to as the *term-document matrix* for the corpus.

In the original VSM indexing application, vector lengths of the term-document matrix are normalized to one, yielding a projection of each study onto the unit sphere [39]. Documents with similar terms appear closer together than those with dissimilar indexed terms. To produce optimal clustering of the document projections, weighting is applied to each term in the term-document matrix [39]. Optimal term weighting methods vary based on the corpus being indexed, but generally take the form:

$$term_{i,j} = LWF(i,j) \times GWF(i) \tag{4}$$

where:

- *LWF(i,j)* is a local weight function expressing the weight of term i in study j, and
- *GWF(i)* is a global weight function expressing the weight of term i across all documents [43].

The selection of weighting functions is needed to tune the algorithm for optimal retrieval. For a detailed discussion of term weighting the reader is referred to Nakov, Popova and Mateev [43].

Queries to retrieve documents based on a set of terms are accomplished by constructing a query vector in t dimensions, applying the appropriate weighting, normalizing in relationship to the space, and projecting the vector into the document space [19,39]. The distance between the projected query vector and other corpus vector projections are then utilized to identify documents which are similar to the original query vector [39].

2.4.2. Latent Semantic Analysis

Deerwester, Dumais, Furnas, Landauer and Harshman [20] proposed the LSA extension to VSM to improve performance with regard to synonymy, polysemy and compound terms. In the case of synonymy, the search terms entered by a user may be expressed in different terms with similar meanings within the corpus. In the same manner, polysemy – where a term has multiple semantic meanings – generates cases where the user intends one meaning of the term while corpus studies utilize alternative meanings. In the case of compound terms, if each word in the compound term is treated in isolation, the resulting matches may be due simply to co-occurrence of the words in a study rather than their use as a compound term in the document. In all cases, the retrieved results do not meet users expectations of the search [20].

Similar to VSM, Deerwester et al.'s process begins with the assembly of a weighted term-document matrix. Next, in place of the vector length normalization process, Singular Value Decomposition (SVD) of the term-document matrix is undertaken [19,20] such that:

$$X = T_0 S_0 D_0' \tag{5}$$

where:

- X is the original term-document matrix,
- T_0 is a matrix of orthonormal, singular column vectors representing the terms in X,
- S_0 is a diagonal matrix of the singular values of X arranged in descending value order, and
- D_0 is a matrix of orthonormal, singular column vectors representing the documents in X [19].

The SVD of the term-document matrix characterizes the semantic dimensionally of the studies represented by the term-document matrix [19,38,44]. By zeroing out singular values of S_0 below a certain threshold, the amount of semantic noise is reduced [19,20], thus allowing a more meaningful match to queries. The number of dimensions to retain is part of the tuning process of the algorithm with 100 to 400 dimensions typically providing best performance [44].

After zeroing out the smaller singular values in S_0, the three matrices of the decomposition are multiplied as indicated in EQ (5) to form a least-squares estimate of the original matrix known as the semantic space. This reduction in semantic dimensionality collapses the weighting of synonyms causing each of the terms to be weighted equally [38]. The polysemy issue is partially resolved as the weighting of such terms are conditioned by other words in the document [20]. Similarly, compound terms – and the individual terms within them – are weight conditioned by surrounding terms in the document.

When utilized for indexing, queries are constructed by forming a query vector and scaling the resulting vector into the semantic space such that:

$$D_q = X_q' T S^{-1} \tag{6}$$

where:

- D_q is the resulting scaled query vector,
- T and S are the matrices from the SVD used to create the semantic space of the corpus, and
- X_q' is the query vector of interest.

The scaled query vector is then compared to the corpus studies by a similarity measure [20,38]. Common similarity measures are [40]:

1. The Cosine of the angle formed by two vectors;
2. Pearson's Correlation;
3. Spearman's Rho; and
4. The cross-product of the vectors.

As compared to VSM, LSA provides additional factors to address synonymy, polysemy and compound terms. Limiting LSA is its added

complexity, and researcher training required to effectively apply the algorithm. A question addressed in this research is the extent to which the added factors and complexity result in improved algorithm effectiveness.

In addition to indexing, LSA has also been applied in areas similar to the study selection task explored in this research. After training with a large corpus of English text, LSA performance on standardized tests for synonym identification is found to be equivalent to the performance of United States college applicants from non-English speaking countries [38].

LSA judgments of word relatedness and sorting show better than chance performance, and in many cases mimic human performance [38]. Similar results have been found in subject matter knowledge testing with multiple choice questions. Additionally, LSA has been utilized to model lexical semantic priming for lexical decisions concerning polysemy [45].

An application of particular relevance to this research is the use of LSA for the assessment of essay content and quality [37,38,46]. In this application, a set of essays that have previously undergone human scoring are scaled into a semantic space constructed from a corpus of associated material and utilized for LSA training. New essays are then processed by transforming them into a document vector scaled into the semantic space and comparing the new document with the training set [37,38,46]. Similarity of meaning is then assessed by the closest match in the training set based on the Cosine measure [37]. Similarity of content is assessed either as the sum of similarities between the new essay and all essays in the training set [37], or as the length of the vector [46]. This approach is used as the basis of the LSA algorithm tested in this research.

3. Research model

This research hypothesizes that through training-by-example, VSM and LSA based classifiers are effective and efficient in the selection of previously unseen studies. In addition, the amount of information needed to determine the inclusion / selection is limited to a subset of the metadata commonly found in citation downloads – specifically the title and abstract of each study. In support of the outlined objectives, the research questions addressed here are:

RQ1: How effective are the proposed VSM and LSA algorithms in identifying relevant studies within a search corpus to be included in a research project?
RQ2: How effective are the proposed VSM and LSA algorithms in excluding irrelevant studies within a search corpus?
RQ3: How much researcher efficiency is gained by the use of the proposed VSM and LSA algorithms?

These research questions are further refined in Section 3.3.

3.1. Modified Study Selection process

This research proposes a modified study selection process, relying on trained automated algorithms to execute the researchers' inclusion and exclusion criteria. Because both SLRs and SMSs begin with the same literature search steps, the algorithms can be equally applied to SLRs and SMSs.

- The manual Study Selection step (as outlined in Section 2.1) is conducted ONLY for a subset of the document corpus, to be used as a classification algorithm training set.
- The VSM or LSA classification algorithm is then run against the remaining studies in the corpus, resulting in a set of included (classified as relevant) studies.
- The researchers then review the set of studies automatically classified/included as relevant, saving the manual researcher effort that would have been required to review the studies excluded/automatically classified as irrelevant.

Through this process, training-by-example provides a simple, intuitive means for researchers to train the classifier, but does so at the cost of researcher time and efficiency because every study within the training set must be manually evaluated (rather than automatically evaluated and classified by the algorithm).

The automated classifiers (VSM and LSA) are conducted on study meta-data (titles, abstracts) that are readily available from literature databases, rather than the full study text. This is a further time savings, as the researchers no longer need to download full text files of the entire corpus. Only those studies in the training set plus those classified as Included by the algorithm require full text download.

3.2. Algorithm tuning

Algorithm tuning is the process of adjusting algorithm parameters to fit the specific problem context in which it is being applied; tuning is used to improve performance (in this case, recall and precision) or to make the algorithm easier to train. Three factors that can be used to tune the performance of VSM and LSA algorithms are [38,40,43,44]:

1. Preprocessing of the text.
2. Weighting functions applied to the term frequencies.
3. The similarity measure employed for comparisons.

Preprocessing factors include stop word filtering, stemming, restrictions based on local or global term frequency counts, and the use of controlled vocabulary [40,44]. Stop word filtering is the exclusion of common words that appear with high frequency in a language such as articles (e.g. in English – a, an, and the). Stemming is the process of reducing a word to its root form. Each of these options influence the terms produced for inclusion in the term vector of a document. Restrictions based on a minimum/maximum appearance of a term within a document, or across the document set, may increase/decrease the number of terms included in the term vector. Likewise, the use of a controlled vocabulary restricts the terms included in the corpus vectors. To follow the principle of minimal tuning and researcher training, this research elects to employ two of these preprocessing options: stop word filtering and stemming. Based on prior research, these options are found to be the most effective in reducing noise in the dataset [40,44,47].

Weighting functions – the transformation of term frequencies – have been shown to influence the performance of VSM and LSA algorithms [43,44]. In investigating the use – and absence – of local and global weighting functions Nakov, Popova and Mateev [43] find the use of weighting functions to be beneficial. Of those tested, the combination that consistently resulted in superior performance is the use of the Log function for local weighting and Shannon's Entropy function for global weighting. Based on their findings, this research elects to use a local log weighting in combination with globally based entropy for term weighting. These weighting functions can be utilized in statistical tools such as R without additional programming or parameterization, in keeping with the principle of minimal tuning.

In addition to these three factors, the performance of the LSA algorithm is further impacted by the selected dimensionality [40,44]. In this research two options are explored:

1. Retain all dimensions from the decomposition. This is equivalent to a simplified VSM treatment of the term vectors and will provide a basis for comparisons.
2. Retain the number of dimensions at which the sum of the dimensions divided by the total of all dimensions meets or exceeds 0.5. This is the default setting for the R statistical package. Given the goal of assessing minimally-tuned models, this default approach was used in this research.

As discussed in Section 2.4.2, there are four primary methods of comparing vectors. The Cosine of the angle formed by the vectors has been previously found to provide the most robust results [48], thus it is employed for comparison of vectors in this research.

3.3. Algorithm variations

The evaluated *training-by-example* classifiers first require providing exemplars of truly relevant and irrelevant studies from the corpus, classified manually by the researcher. This section describes the VSM and LSA algorithm variations explored in this research:

1. The size of the training set manually evaluated by the researcher,
2. The controlled proportion of relevant studies within the training set,
3. The vocabulary used by the classifier, and
4. The number of dimensions in the LSA final vector space.

The exploration of these algorithm variations further refine the research questions set out earlier in this section.

3.3.1. Size of training set

Increasing the size of the training set is expected to improve classifier accuracy, but does so at the expense of researcher efficiency. The relative tradeoffs are explored through the refined research questions (RQ):

RQ1a: How does the training set size impact the recall performance of the VSM and LSA algorithms?
RQ2a: How does the training set size impact the precision performance of the VSM and LSA algorithms?
RQ3a: How does the training set size impact the researcher efficiency performance of the VSM and LSA algorithms?

3.3.2. Controlling the proportion of relevant studies in the training set

Training provides the classifier with examples of truly relevant and irrelevant studies from the corpus. These examples can be selected randomly from the corpus, or the proportion of truly relevant / irrelevant studies can be controlled. This is accomplished by selecting relevant studies from a corpus subset known to contain a high proportion of relevant studies; for example, from a related conference proceedings or journal special issue. Controlling the proportion of relevant and irrelevant studies may bias the selection algorithms, and is explored through these research questions:

$RQ1_b$: How does controlling the proportion of relevant and irrelevant studies in the training set size impact the recall performance of the VSM and LSA training-by-example algorithms?
$RQ2_b$: How does controlling the proportion of relevant and irrelevant studies in the training set size impact the precision performance of the VSM and LSA training-by-example algorithms?
$RQ3_b$: How does controlling the proportion of relevant and irrelevant studies in the training set size impact the researcher efficiency performance of the VSM and LSA training-by-example algorithms?

3.3.3. Vocabulary

The VSM and LSA algorithms are executed against the *term-document matrix*, with column vectors representing the studies in the corpus and the row vectors representing the frequency of terms utilized in the corpus for each study [19,38,39]. Thus, the vocabulary of terms drives the results of the algorithm. The vocabulary can be set by the studies in the training set or by all the studies in the corpus. The impact of vocabulary size on the selection algorithms is explored through these research questions:

$RQ1_c$: How does the vocabulary size impact the recall performance of the VSM and LSA algorithms?
$RQ2_c$: How does the vocabulary size impact the precision performance of the VSM and LSA algorithms?
$RQ3_c$: How does the vocabulary size impact the researcher efficiency performance of the VSM and LSA algorithms?

3.3.4. Dimensionality

The LSA algorithm is a form of factor analysis [38]. It collapses synonyms into a common term, thus reducing the number of dimensions with minimal loss of information. Similarly, compound terms utilized repeatedly in similar contexts will be weighted the same and therefore collapse into a singular term. In both cases, the distance between documents in higher dimensional space will be reduced. For the VSM algorithm there is no dimensional reduction, thus the number of dimensions will be equal to the size of the vocabulary. The impact of dimensionality on the LSA selection algorithms is explored through these research questions:

$RQ1_d$: How does the number of retained dimensions impact the recall performance of the LSA algorithms?
$RQ2_d$: How does the number of retained dimensions impact the precision performance of the LSA algorithms?
$RQ3_d$: How does the number of retained dimensions impact the researcher efficiency performance of the LSA algorithms?

These research questions are explored through the empirical research as described in the following section.

4. Primary simulation studies

Evaluation of the VSM and LSA training-by-example classifier algorithms is accomplished through a series of repeated independent selection simulations. Algorithm parameters (as set out in Section 3.3) are varied to address the posed research questions that assess the algorithms' impact on recall, precision, and researcher efficiency savings.

4.1. Dataset

The corpus (i.e., dataset) for this research is comprised of the literature search results of a systematic mapping study conducted in the area of Evidence Based Software Engineering for the period from 2004 to 2014 [8]. In the study, a broad search string was utilized to query five electronic literature databases, providing comprehensive coverage of work completed during the time period concerning the development of EBSE. This dataset was selected to construct the corpus for this research as 1) it was the result of a well-documented, actual search and selection process; 2) it is vetted for inclusion/exclusion by active researchers in the domain; and 3) it embodies additional semantic complexities in the form of closely related topics that are difficult to clearly delineate – similar to that found in conducting SLRs. A total of 5979 unique reported studies were selected to construct the corpus.

The publications selected for inclusion in the corpus analysis consist of 543 studies in the dataset and is available through Researchgate [49]. This subset is used to represent the full set of truly relevant (TP) studies in the corpus.

The remainder of the dataset, 5436 studies, represents publications returned by the search query, but ultimately excluded from the analysis. These are also available through Researchgate [50]. This subset is used to represent the full set of truly irrelevant studies (TN) in the corpus.

4.2. Procedure

Each of the proposed classifiers is programmed in R, following the logic outlined in this section and is available through Researchgate [51]. The two algorithms are then independently repeatedly applied through random subsampling within the research dataset. In each

simulation run, a subset of studies is randomly chosen from the document corpus, on which the VSM and LSA algorithms variations (as set out in Section 3.3) are trained. The remainder of the corpus studies are then automatically classified by the trained algorithms.

4.2.1. Classifier algorithms

To construct the VSM classifier, the title and abstract of each study are parsed to form a document vector consisting of the word frequencies found in the text. Next, term vectors from studies identified for training and previously marked as included or excluded, along with term vectors of studies to be classified, are aggregated to form a term-document matrix. Term vectors in need of classification are compared to each term vector in the training set to find the closest match, representing a k of 1 in the k-nearest-neighbor classifier [46].

If the closest match for the term vector to be classified is a term vector associated with a study marked for inclusion, then the study represented by the term vector to be classified is marked for inclusion. If the closest match for the term vector to be classified is a term vector associated with a study marked for exclusion, then the study represented by the term vector to be classified is marked for exclusion. This process continues until all term vectors in need of classification are marked as either included or excluded.

To construct the LSA classifier, an addition is made to the base VSM classifier. After the initial parsing to form document vectors for each study, term vectors from studies identified for training and previously marked as included or excluded are aggregated to form a term-document matrix. The term-document matrix is then submitted to LSA to form the semantic space for comparisons. After construction of the semantic space, the term vectors of the studies to be classified are scaled into the semantic space as previously noted. Following the scaling process, selection for inclusion or exclusion of unknown studies proceeds as described for the VSM classifier.

4.2.2. Algorithm software

The proposed classifier algorithms are implemented utilizing version 3.1.1 of the R statistical package and the associated LSA add-in. The LSA add-in provides the parser and associated functionality required to convert individual text files into term vectors and term-document matrices. The add-in also contains functions for scaling and measurement.

Each classifier is implemented in the form of an R-script which reads individual files containing the title and abstract of one study from a directory on disk. Each file is parsed to form a document vector, which in turn is processed in accordance with the implemented algorithm. To simplify the execution of simulated algorithm trials, key parameters are implemented via variables utilized as constants with values assigned at the beginning of each script. The results of each trial are recorded individually with the aggregated results exported as a comma separated variable file.

4.2.3. Simulations

This research uses Monte-Carlo Cross-Validation to simulate and assess the VSM and LSA algorithm variations. Using Monte-Carlo Cross-Validation [52], a researcher randomly selects (without replacement) a fraction of data to form a training set, assigning the remainder of the points to a test set. This process is repeated, randomly generating new training and test partitions each time.

An alternative the Monte-Carlo approach is the k-fold cross-validation technique, in which the dataset is divided into k equally-sized mutually-exclusive “folds,” with one fold serving as the test set and the remaining k-1 folds to form the training set. This process is repeated, with each fold used once as the training set. The Monte-Carlo method was chosen over the k-fold technique to meet the research objective of identifying algorithms that reduce the number of corpus studies that must be manually reviewed (thus improving researcher efficiency). Using the k-fold technique, the majority of corpus studies (k-1 folds) would be included in the training set, all of which must be manually reviewed, thus minimizing the use and efficiency benefit of the algorithm. For example, a 10-fold cross validation approach would result in 90% (1 – 1/10) of the corpus studies included in the training set (with complete manual review). A 2-fold cross validation approach would result in 50% (1 – ½) of the corpus studies included in the training set (with complete manual review).

In this research, the publication corpus represents the domain of possible classifier inputs. For a given set of algorithm parameters, 100 independent trials are conducted. For each trial, algorithm parameters are set as outlined in Table 2. A study training set is selected randomly from the corpus, and the algorithms are trained from that subset of studies. The trained algorithms are then used to automatically classify as included or excluded the remainder of the publication corpus, and measures of recall, precision, and efficiency savings are obtained as outlined in Section 2.2. Results are then aggregated across all 100 trials with identical parameters.

4.3. Measures and GLM model

Following the completion of each trial, the include/exclude decision of the classifier is compared to the ground truth set by the original decision made in the SMS for each of the publications. For each trial, the number of TP and TN classifications (i.e., the correct inclusions and

Table 2
Simulation parameters
From model variations, Section 3.3.

Parameter	Studied values	Comments
MODEL: Classifier Algorithm	2 Algorithms: VSM, LSA	Reflected as a group classifier in the GLM.
TSS: Percentage of corpus studies randomly selected for the classifier training set	5 settings: 1%, 5%, 10%, 20%, 30%	Each simulation run will randomly select a set proportion of the studies for algorithm training. The remainder will be automatically selected or excluded.
TSS Squared	From TSS value	Added to reflect the identified non-linear relationship between Training Set Size and the dependent variables Recall and Precision.
TSS Cubed	From TSS value	Added to reflect the identified non-linear relationship between Training Set Size and the dependent variables Recall and Precision.
REL-INCL: Percent of Truly Relevant Studies in the classifier training set	4 settings: Uncontrolled, 10%, 20%, 30% of corpus	Logit transformation performed, as recommended for percentage values [53].
VOCAB: Vocabulary Size	2 settings: Training Set, Entire Corpus	Training set vocabulary determined within each simulation trial
DIM: Dimensions	2 settings for LSA: 1. All, 2. (Sum all dim) / (Total of all dim) meets or exceeds 0.5 1 setting for VSM: all dimensions	Dimensions set within each simulation trial

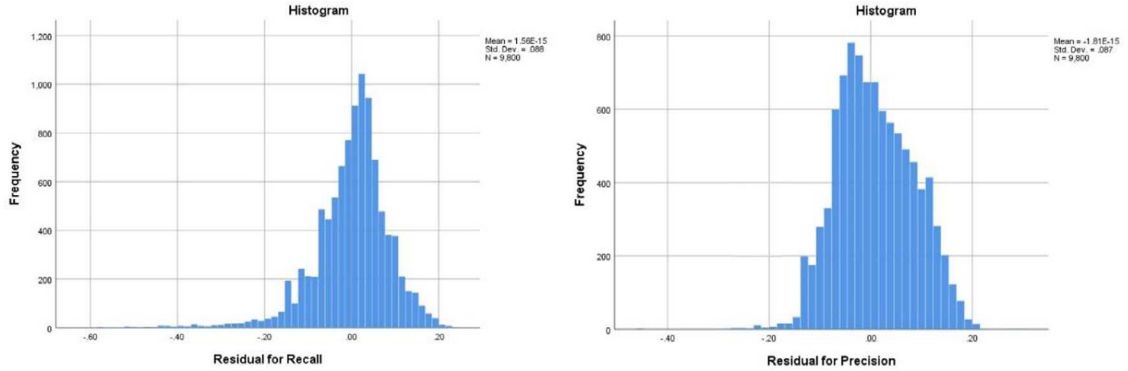

Fig. 1. Recall and precision residual histograms.

correct exclusions), FN errors, and FP errors are recorded. From these base data, recall, precision, and researcher efficiency savings are then calculated.

A multivariate general linear model (GLM) was then created to assess the impact of the model parameters (as described in Table 2) on the dependent variables of Recall and Precision. Within the GLM class of models lies regression, ANOVA, factor analysis, cluster analysis, and most empirical analysis techniques used in applied statistical research. The tested model can be expressed as:

$$Y = XB + U_1 \quad (7)$$

where:

- Y is the two-dimensional matrix of $[Y_r, Y_p]$,
- Y_{ri} is the *i*th observation of model Recall,
- Y_{pj} is the *j*th observation of model Precision,
- X is the design matrix of explanatory variables (MODEL, T-SIZE, T-SIZE-SQ, T-SIZE-CUBED, REL-INCL, VOCAB, and DIM),
- B is the matrix containing the parameter estimates for X, and
- U is the error matrix.

5. Results

Results associated with each of the three main research questions and the related sub-questions are found in Tables 3–6 and detailed in this section. We first address the model assumptions, research questions related to algorithm recall performance, followed by precision and researcher efficiency (discussed together because of their interdependence). We conclude by examining the relative performance of algorithm type (VSM versus LSA). All analysis is completed using IBM SPSS for 64 bit Windows, Version 24.

5.1. GLM assumptions

Prior to exploring the GLM results, it is important to determine the extent to which critical statistical model assumptions are valid. Important to the GLM model are the assumptions of residual normality, equality of variance, as well as the validation of outliers or influential individual observations.

Table 3
Descriptive Statistics.

Model		Mean	Std. deviation	N
Recall	LSA	.79579828423	.141516504262	4899
	VSM	.81965707824	.122598206503	4899
	Total	.80772768123	.132925379309	9798
Precision	LSA	.34367693648	.134688127738	4899
	VSM	.30139215351	.128169070044	4899
	Total	.32253454500	.133151730199	9798

The assumption of residual normality is tested using the Kolmogorov–Smirnov statistic. The residuals for Recall ($p = 0.0023$) and Precision ($p = 0.0003$) showed statistically significant variation from normality. Because this can be the result of high sensitivity from the large sample size [54], the residual histogram plots were examined (see Fig. 1). Both histograms appear approximately normal, supporting use of the model [55]. Further support is found in examination of the residuals' skewness (Recall, −1.294; Precision, −0.345) and kurtosis (Recall, 3.280; Precision, −0.574). For large sample sizes such as used in this research, Gravetter and Wallnau [56] recommend acceptable limits of |2| for skewness and |7| for kurtosis. Both of these normality conditions are met here.

The assumption of homoscedasticity (equal variances) is tested using the Breusch–Pagan test. Results of the Breusch–Pagan test indicate unequal variances ($p = 0.0181$). Although GLM Type-I errors have been found to be robust against moderate violations of this assumption, particularly when group sizes are equal [57], as is the case here, robust standard errors were used to correct for the discovered heteroscedasticity [58].

Cook's Distance was used to identify influential observations and predictor outliers. Notable observations were inspected and determined to be valid; no data were removed as a result.

5.2. Effect size

In this research, effect size is measured by partial eta squared (η_p^2), the proportion of the total variance in a dependent variable that is associated with a unit change in an independent variable when the effects of other independent variables and interactions are removed [59]. Following Cohen's [60] benchmarks, partial eta squared values of 0.0099, 0.0588, and 0.1379 are interpreted as small, medium, and large effect sizes, respectively.

5.3. RQ1: algorithm recall performance

$RQ1_a$ asks:

How does the training set size impact the recall performance of the VSM and LSA algorithms?

Within this research, the linear (TSS) and non-linear (TSS-SQ and TSS-Cubed) regression predictor variables are all statistically significant predictors of recall, as shown in Table 4. The linear TSS coefficient is positive; the associated η_p^2 of 0.086 shows that TSS alone explains 8.6% of variation in algorithm recall performance, once the impact of other variables is accounted for. The squared term (TSS-SQ) is associated with a negative β and η_p^2 of 0.039. The cubed term (TSS-C) is associated with a positive β and η_p^2 of 0.025.

Thus, increasing the training set size significantly improves classifier recall, across all examined algorithm variations. Pooling the impact of the linear and non-linear effects, training set size accounts for

Table 4
Tests of between subjects effects.

Source		Type III sum of squares	Df	Mean square	F	Sig.	η_p^2	Noncent. parameter	Observed Power[c]
Corrected model	Recall	111.024[a]	7	15.861	2501.190	0.000	.641	17,508.332	1.000
	Precision	116.399[b]	7	16.628	2841.269	0.000	.670	19,888.883	1.000
Intercept	Recall	649.360	1	649.360	102,402.961	0.000	.913	102,402.961	1.000
	Precision	22.176	1	22.176	3789.134	0.000	.279	3789.134	1.000
Dims	Recall	.410	1	.410	64.608	.000	.007	64.608	1.000
	Precision	.215	1	.215	36.722	.000	.004	36.722	1.000
Vocab	Recall	.009	1	.009	1.469	.226	.000	1.469	.228
	Precision	.570	1	.570	97.378	.000	.010	97.378	1.000
logitPctIncl	Recall	104.444	1	104.444	16,470.584	0.000	.627	16,470.584	1.000
	Precision	91.523	1	91.523	15,638.417	0.000	.615	15,638.417	1.000
TSS	Recall	5.854	1	5.854	923.137	.000	.086	923.137	1.000
	Precision	.122	1	.122	20.902	.000	.002	20.902	.995
TSS-SQ	Recall	2.525	1	2.525	398.200	.000	.039	398.200	1.000
	Precision	.076	1	.076	13.056	.000	.001	13.056	.951
TSS-CUBED	Recall	1.583	1	1.583	249.705	.000	.025	249.705	1.000
	Precision	.003	1	.003	.579	.447	.000	.579	.119
Model	Recall	1.664	1	1.664	262.407	.000	.026	262.407	1.000
	Precision	3.294	1	3.294	562.775	.000	.054	562.775	1.000
Error	Recall	62.081	9790	.006					
	Precision	57.296	9790	.006					
Total	Recall	6565.555	9798						
	Precision	1192.966	9798						
Corrected Total	Recall	173.105	9797						
	Precision	173.695	9797						

[a] R Squared = 0.641 (Adjusted R Squared = 0.641).
[b] R Squared = 0.670 (Adjusted R Squared = 0.670).
[c] Computed using alpha = 0.05.

approximately 15% of the variation in algorithm recall performance, considered a large effect size. The significant and negative TSS-SQ β-term tells us that improvement dissipates as the training set size increases.

Recognizing that within a given training set size the proportion of truly relevant and truly irrelevant studies can be controlled, $RQ1_b$ asks:

How does controlling the proportion of relevant and irrelevant studies in the training set size impact the recall performance of the VSM and LSA algorithms?

In this research, we examine algorithms with an uncontrolled percentage of relevant studies as well as training sets with 1, 5, 10, 20, and 30% relevant studies. As shown in Table 4, the associated log-transformed predictor variable (log-PCT-INCL) is a statistically significant predictor of recall, with a positive β. Thus, increasing the percentage of truly relevant studies in the training set significantly improves the algorithm recall performance. The associated η_p^2 is 0.627, meaning that the percent of relevant studies in the training set has a large effect on the classifier recall variation.

For a given classifier, the vocabulary set used to classify studies can be extracted from the training set or from the entire corpus of studies. Thus, $RQ1_c$ asks:

How does the vocabulary size impact the recall performance of the VSM and LSA algorithms?

In this research, the vocabulary size (VOCAB) did not significantly impact classifier recall performance, with a significance level of 0.226 (see Table 4).

An LSA classifier algorithm can be modified by the number of semantic dimensions represented (used to compare a study for inclusion

Table 5
Parameter estimates.

Dependent variable		β	Std. error	t	Sig.	95% Confidence interval		η_p^2	Noncent. Parameter
						Lower bound	upper bound		
Recall	Intercept	.901	.003	302.204	0.000	.895	.906	.903	302.204
	DIMS	−2.589E-05	3.221E-06	−8.038	.000	−3.220E-05	−1.958E-05	.007	8.038
	Vocab	−1.327E-07	1.095E-07	−1.212	.226	−3.472E-07	8.191E-08	.000	1.212
	logitPctIncl	.258	.002	128.338	0.000	.254	.262	.627	128.338
	TSS	.001	1.316E-05	30.383	.000	.000	.001	.086	30.383
	TSS-SQ	−3.631E-07	1.820E-08	−19.955	.000	−3.988E-07	−3.274E-07	.039	19.955
	TSS-CUBED	1.053E-10	6.667E-12	15.802	.000	9.228E-11	1.184E-10	.025	15.802
	[Model = LSA]	−.037	.002	−16.199	.000	−.041	−.032	.026	16.199
	[Model = VSM]	0[a]							
Precision	Intercept	.137	.003	47.950	0.000	.132	.143	.190	47.950
	DIMS	1.875E-05	3.094E-06	6.060	.000	1.269E-05	2.482E-05	.004	6.060
	Vocab	1.038E-06	1.052E-07	9.868	.000	8.316E-07	1.244E-06	.010	9.868
	logitPctIncl	−.242	.002	−125.054	0.000	−.246	−.238	.615	125.054
	TSS	5.782E-05	1.265E-05	4.572	.000	3.303E-05	8.261E-05	.002	4.572
	TSS-SQ	−6.316E-08	1.748E-08	−3.613	.000	−9.743E-08	−2.890E-08	.001	3.613
	TSS-CUBED	4.875E-12	6.405E-12	.761	.447	−7.680E-12	1.743E-11	.000	.761
	[Model = LSA]	.052	.002	23.723	.000	.047	.056	.054	23.723
	[Model = VSM]	0[a]							

Table 6
Algorithm Average Performance.

Training set size (% of Corpus)	% Relevant studies in training Set	Efficiency savings		Avg recall		Avg precision	
		VSM	LSA	VSM	LSA	VSM	LSA
OVERALL	0.747	0.775	0.820	0.796	0.301	0.344	
1	UC	***0.887***	***0.918***	***0.591***	***0.494***	0.365	0.403
1	10	0.868	0.905	0.684	0.578	0.344	0.392
1	20	0.798	0.856	0.804	0.757	0.275	0.342
1	30	0.710	0.793	0.896	0.873	0.223	0.286
5	UC	0.860	0.878	0.724	0.699	0.413	0.458
5	10	0.858	0.877	0.733	0.708	0.406	0.454
5	20	0.796	0.830	0.847	0.833	0.309	0.360
5	30	0.740	0.790	0.901	0.887	0.246	0.296
10	UC	0.827	0.840	0.713	0.700	0.441	0.483
10	10	0.821	0.834	0.745	0.732	0.429	0.469
10	20	0.765	0.792	0.865	0.852	0.312	0.357
10	30	0.716	0.755	0.913	0.899	0.232	0.274
20	UC	0.739	0.749	0.746	0.737	0.462	0.499
20	10	0.737	0.747	0.753	0.747	0.449	0.488
20	20	0.691	0.710	0.868	0.858	0.286	0.323
20	30	0.649	0.678	0.917	0.910	0.159	0.188
30	UC	0.652	0.660	0.745	0.739	***0.479***	***0.513***
30	10	0.649	0.657	0.759	0.752	0.459	0.495
30	20	0.610	0.625	0.870	0.862	0.226	0.258
30	30	***0.572***	***0.596***	***0.918***	***0.903***	***0.014***	***0.017***

Notes: UC = Uncontrolled. Italics Used to highlight best and worst performing models.

or exclusion). The associated $RQ1_d$ asks:

How does the number of semantic dimensions impact the recall performance of the LSA algorithms?

Increasing the number of dimensions (DIMS) has a statistically significant impact on LSA algorithm recall variance (p-value = 0.000; see Table 4). However, the impact of dimensionality within this research is less than Cohen's [60] benchmark for small impact, with a η_p^2 of only 0.007 (0.7% of recall variance is explained by the number of semantic dimensions).

Two *training-by-example* classifier algorithms are explored in this research – the VSM, and the LSA algorithm. In evaluating these two algorithms, RQ1 asks:

Which classifier algorithm (VSM or LSA) results in higher recall performance?

As illustrated in Table 3, the average algorithm recall is 0.81 across all simulated algorithm trials. On average, the VSM class of algorithm outperformed the LSA algorithms (with a VSM average recall of 0.82 compared to a LSA average recall of 0.80). Thus, within this research, the recall levels of the VSM classifier consistently out-performed the LSA classifier across all examined variations. However, the effect size is small; with a η_p^2 of only 0.026, the type of classifier algorithm accounts for only 2.6% of the variation in recall performance.

5.4. RQ2: algorithm precision performance

$RQ2_a$ asks:

How does the training set size impact the precision performance of the VSM and LSA algorithms?

Within this research, the linear (TSS) and squared (TSS-SQ) regression variables are statistically significant predictors of precision, while the cubed variable (TSS-Cubed) is not statistically significant (see Table 4). The linear TSS coefficient is positive, with an associated η_p^2 of only 0.002. The squared TSS (TSS-SQ) coefficient is negative, with an associated η_p^2 of only 0.001.

Thus, the results show that increasing the training set size does improve algorithm precision. While the impact is statistically significant, the practical impact is very small (once taking into account other factors). Pooling the impact of the linear and squared effects, training set size accounts for only 0.3% of the variation in precision, less than Cohen's benchmark [60] for small impact size.

For a given size of a corpus training set, researchers can control for the proportion of truly relevant and irrelevant studies. Thus, $RQ2_b$ asks:

How does controlling the proportion of relevant and irrelevant studies in the training set size impact the precision performance of the VSM and LSA algorithms?

Using the same approach as previously described, we examine algorithms with an uncontrolled percentage of relevant studies as well as training sets with 1, 5, 10, 20, and 30% relevant studies. As shown in Table 4, the associated log-transformed predictor variable (log-PCT-INCL) is a statistically significant predictor of precision, with a negative β. Thus, increasing the proportion of truly relevant studies in a training set of a specific size significantly reduces the classifier precision.

The associated η_p^2 of 0.615 means that the percentage of truly relevant studies included in the classifier training set determines 61.5% of the variance in classifier precision – considered a large impact and the highest-impact variable in the regression.

Examining the impact of vocabulary size (set by the training set or the entire corpus) on precision, $RQ2_c$ asks:

How does the vocabulary size impact the precision performance of the VSM and LSA algorithms?

The larger vocabulary (from the entire corpus of studies) improves precision performance at a statistically significant level (p = 0.000; see Table 4). However, the associated η_p^2 of 0.010 indicates that the impact is small; vocabulary size explains only 1% of the variation in classifier precision performance.

Examining the impact of dimensionality on LSA classifier precision, $RQ2_d$ asks:

How does the number of semantic dimensions impact the precision performance of the LSA algorithms?

Increasing the number of semantic dimensions (DIMS) improves precision performance at a statistically significant level (p = 0.000; see Table 4). However, the variable's η_p^2 is only 0.004; thus the impact of dimensionality is less than Cohen's [60] benchmark for small effect size, explaining only 0.4% of variation in LSA algorithm precision.

Comparing the precision of the VSM and LSA algorithms, RQ2 asks:

Which classifier algorithm (VSM or LSA) results in higher precision performance?

As illustrated in Table 3, the average algorithm precision is 0.32 across all simulated algorithm trials. On average, the LSA class of algorithms outperformed the VSM algorithms (with a LSA average precision of 0.34 compared to a VSM average precision of 0.31). Thus, within this research, the LSA algorithms result in consistently and significantly higher levels of precision, across all examined algorithm variations. The effect size is moderate; moving from a VSM to LSA classifier algorithm accounts for 5% (η_p^2 of 0.054) of the variation in precision, all other modeled effects removed.

5.5. RQ3: algorithm researcher efficiency performance

Automated study selection of a search corpus allows a researcher to examine fewer studies manually. Thus, with acceptable levels of recall, all automated training-by-example classifier algorithms aid the researcher by reducing the number of studies manually reviewed (thus, improving efficiency). Because both the LSA and VSM algorithms (with equivalent parameters) are based on the same training set size, researcher efficiency (reduction in studies manually reviewed and

classified) is very similar. The difference between the two results from differences in precision (irrelevant studies misclassified that must then be read and removed manually). Because the precision levels of the LSA algorithm consistently out-perform the VSM algorithm across all examined variations, a similar result is thus found with respect to researcher efficiency. As shown in Table 6, tested LSA algorithms improved researcher efficiency by an average of 77.5% over a completely manual process, while the tested VSM algorithms improved efficiency by 74.7%.

$RQ3_a$ asks:

How does the training set size impact the researcher efficiency performance of the VSM and LSA algorithms?

In this research, there were 5,979 studies in the document corpus, with training sets of 1%, 5%, 10%, 20% and 30%. The impact of training set size on researcher efficiency is illustrated by the research results shown in Table 6.

A larger training set size means that the researcher must manually review a larger number of studies. Thus, increasing training set size reduces researcher efficiency, as illustrated in Table 6. With a training set of 1%, the LSA algorithms resulted in efficiency savings of 79–92%, and the VSM algorithms led to a 71–89% improvement. When the training set was increased to 30% of the corpus, the equivalent savings decreased to 60–66% and 57–65% respectively.

This effect is balanced by an algorithm's precision. As precision improves (holding the training set size constant), the researcher reviews fewer irrelevant studies incorrectly included by the classifier, as illustrated in Table 6. Within the 1% training set algorithms, as LSA precision grew from 28.6% to 40.3% (as a result of more irrelevant documents in the training set), researcher efficiency savings correspondingly grew from 85.6% to 91.8%. Also within the 1% training set algorithms, as VSM precision grew from 27.3% to 36.5%, researcher efficiency savings grew from 71.4% to 88.7%.

$RQ3_b$ asks:

How does controlling the proportion of relevant and irrelevant studies in the training set size impact the researcher efficiency performance of the VSM and LSA algorithms?

The efficiency impact of the proportion of truly relevant studies within a training set of a specified size operates through precision. For a given training set size, a more precise algorithm will improve researcher efficiency by reducing the number of truly irrelevant studies that must be manually reviewed after the automated selection is completed, as illustrated in Table 6. Thus, paralleling the findings from RQ2, researcher efficiency is increased as the proportion of truly irrelevant studies in a training set is increased.

$RQ3_c$ asks:

How does the vocabulary size impact the researcher efficiency performance of the VSM and LSA algorithms?

The impact of vocabulary size on researcher efficiency operates mathematically through precision. Since the analysis of $RQ2_c$ found that the corpus-derived vocabulary improves precision, using the corpus rather than the smaller training set alone will also improve researcher efficiency. Because the $RQ2_c$ research findings determined that vocabulary size impact on precision is small, so is its impact on researcher efficiency.

$RQ3_d$ asks:

How does the number of semantic dimensions impact the researcher efficiency performance of the LSA algorithms?

The impact of LSA dimensionality on researcher efficiency operates mathematically through precision. Since the analysis of $RQ2_d$ found that increasing the number of semantic dimensions improves LSA algorithm precision at a statistically, but not practically, significant level, the same relationship holds for the impact of the number of LSA semantic dimensions on researcher efficiency.

6. SLR test case study

The primary dataset used to assess the performance of the proposed training-by-example classifiers is a large document corpus (of 5979 studies) obtained to support an SMS with the broad research goal of understanding the EBSE research domain. As discussed in Section 5, results show promise. Questions remain, however, regarding external validity. Will the algorithms prove to be efficient and effective when applied to different EBSE research contexts? To address this important question, the two proposed algorithms are applied to a focused SLR and the results compared to the original SMS-based simulation.

The SLR test case study uses a published SLR dataset in the area of software requirements change [61]. For testing purposes here, a subset of Bano at al.'s [61] document corpus was extracted. The corpus subset is comprised of 190 studies randomly selected from Bano et al.'s [61] literature search of six major databases (ACM portal, IEEE xplore, Science Direct, Citreseerx, Springer link, EI Compendex).

Of the 190 studies, 5 are TP and 185 are TN. These 190 studies are supplemented by an additional 3 TP studies published since 2010 (independently evaluated and judged as TP by three experienced researchers), resulting in a test corpus of 193 total studies, 8 TP and 185 TN. The primary SMS simulation is based on a relatively large (5979 documents) corpus with 9.1% TP studies, while the SLR test case study is based on a relatively small (193 documents) corpus with 4.1% TP studies.

The VSM and LSA algorithms are then applied to the test SLR corpus using the same method as discussed for the primary SMS dataset. The results of the test SLR corpus simulations are summarized and compared to the original SMS-based results in Table 7. Four scenarios were tested: a training set size of 2 studies (1% of the corpus; 1 TP, 1 TN); a training set size of 15 (8% of the corpus; 5 TP, 10 TN); a training set size of 30 (16% of the corpus; 5 TP, 25 TN), and a training set size of 50 (26% of the corpus; 5 TP and 45 TN). Vocabulary and dimensionality were held constant.

Table 7
SLR case study algorithm average performance.

Training set size (% of Corpus)	% Relevant studies in training Set	Efficiency savings		Avg recall		Avg precision	
		VSM	LSA	VSM	LSA	VSM	LSA
1	50	0.258	0.266	0.986	0.986	0.057	***0.059***
8	33	0.401	0.614	***1.000***	***0.993***	***0.033***	0.066
16	17	***0.510***	***0.643***	***1.000***	0.980	0.059	0.112
26	10	0.426	0.512	***0.997***	***0.963***	***0.089***	***0.142***
Original SMS – OVERALL	0.747	0.775	0.820	0.796	0.301	0.344	
Original SMS –Poorest Measures	0.572	0.596	0.591	0.494	0.014	0.017	
Original SMS –Best Measures	0.887	0.918	0.918	0.903	0.479	0.513	

Note: Italics used to highlight best and worst performing models.

For this SLR test case study, both VSM and LSA algorithms resulted in nearly perfect recall. Over the 100 simulated trials, the lowest observed average recall is 96.3% and most scenarios resulted in average recall of better than 99%. This result represents a significant improvement over the primary SMS-based simulation.

The SLR test case study results show that LSA algorithms perform better in precision. LSA algorithms resulted in precision measures ranging from 5.9% to 14.2%, while VSM algorithms resulted in precision measures ranging from 3.3% to 8.9%. These results represent a significant degradation of precision when compared to the primary SMS-based simulation.

For this SLR test case study, the researcher efficiency savings from LSA algorithms are consistently better than their VSM counterparts. LSA algorithms resulted in researcher efficiency savings ranging from a low of 26.6% to a high of 64.3%, while VSM algorithms resulted in researcher efficiency savings ranging from a low of 25.8% to a high of 51.0%. Across all scenarios, researcher efficiency savings are less than in the primary SMS-based simulations.

In the SLR test case study, as in the primary SMS-based simulation, precision improves with larger training sets containing a higher portion of TN studies; the limited number of scenarios do not allow the determination of whether this is due to training set size or the proportion of TN studies in the training set.

Researcher efficiency savings is maximized at moderate (8% and 16%) sizes of the training set. That is, the case study results indicate that the training set size exhibits a point of diminishing returns with regard to researcher efficiency, due to: the increase in researcher time required to review and classify studies within the larger training sets; and decreasing rates of improvement in algorithm precision (and associated number of FP) resulting from the additional training.

7. Discussion and implications

This research provides insight into significant means by which EBSE researchers can improve their Study Selection results while reducing the amount of manual review effort required and with minimal algorithm tuning. While the primary goal of this research is to reduce researcher effort during study selection, the importance of recall must not be overlooked. This is evidenced by the extant literature specifically examining the outcome of the search process [3,5,6]. Simply stated, recall during the selection process is as critical as it is in the search process. The selection process is dependent on the search process data (data coupling) and, in turn, the outcome of the analysis/synthesis process is dependent on the selection process data.

The level of recall required for a given investigation may vary based on trade-off among various concerns. As previous research has noted [see 11,62] the recall required for a SLR may be considerably higher than what is needed for the typical SMS. Similarly, when there is risk to life or high financial risks associated with the topic, a higher level of recall is warranted. Other concerns may include time, infrastructure and financial or other resource limitations.

With respect to the performance of the classifiers in selecting/including TP studies (recall), this research found that:

- VSM was consistently better than LSA (RQ1).
- For both selection algorithm classes, increasing the size of the training set (ranging here from a low of 1% to a high of 30%) significantly improved recall performance ($RQ1_a$).
- Algorithm recall was most improved by forcing the inclusion of a larger proportion of truly relevant studies in the training set (ranging here from uncontrolled to 10%, 20% or 30% - $RQ1_b$).
- Neither vocabulary size nor dimensionality resulted in any practically significant algorithm recall performance differences ($RQ1_c$, $RQ1_d$).
- In the second SLR test case study conducted for external validity, both VSM and LSA algorithms resulted in nearly perfect recall.

With respect to the performance of the classifiers in excluding TN studies (precision), this research found that:

- LSA was consistently better than VSM (RQ2).
- Increasing the size of the training set does not significantly improve algorithm precision (RQ2a).
- Algorithm precision was most improved by forcing the inclusion of a larger proportion of truly irrelevant studies in the training set (ranging here from uncontrolled to 90%, 80% or 70% - $RQ2_b$); more important than the size of the training set, it is critical to train the algorithm on what NOT to include as much as what TO include.
- Neither vocabulary size nor dimensionality resulted in any practically significant algorithm precision performance differences ($RQ2_c$, $RQ2_d$).
- In the second SLR test case study conducted for external validity, results show that LSA algorithms perform better in precision. Across all algorithm variations, precision is lower in the small SLR case study when compared to the primary SMS-based simulation.

In the primary SMS-based simulations, tested LSA algorithms improved researcher efficiency by an average of 77.5% over a completely manual process, while the tested VSM algorithms improved efficiency by 74.7%. In the second SLR-based case study, LSA algorithms resulted in researcher efficiency savings ranging from a low of 26.6% to a high of 64.3%, while VSM algorithms resulted in researcher efficiency savings ranging from a low of 25.8% to a high of 51.0%. Thus, for the EBSE research examined here, the automated classifier algorithms significantly improved researcher efficiency over the traditional manual methods; more improvement is seen in SMS-based study selection characterized by a larger corpus and a greater percentage of TP studies.

7.1. Comparison to alternatives

The research results presented in this manuscript compared the VSM and LSA automated classifiers to the traditional manual EBSE methods, rather than other proposed improvement methods such as those discussed in Section 2.3. Felizardo et al. [22] propose a Visual Text Mining tool that enables the researchers to quickly visualize the similarities and differences among corpus studies. They conducted an SLR case study using 15 subjects, 7 of which used manual selection and 8 used the proposed Revis tool. Their case study utilized 20 studies in the training set and 37 in test set (with 23 TP and 14 TN in the test set). To enable equivalent comparison, their measures were translated to the measures used in this research.

The VTM approach resulted in 62.0% recall, 81% precision, and 22% researcher efficiency savings. To create a roughly equivalent test case, we utilized a single LSA algorithm scenario with a training set of 20 (2 TP and 18 TN). The resulting 100 simulated study selections were characterized by a recall of 98.8%, precision of 55.7% and researcher efficiency savings of 29.7. Thus in this single comparison, the proposed automated LSA training-by-example algorithm outperformed VTM in recall and researcher efficiency savings, while the VTM approach proposed by Felizardo et al. [22]. resulted in higher precision.

7.2. Implications

The findings of this research carry several implications for the EBSE community. This research demonstrates the feasibility of utilizing the VSM and LSA algorithms for the purpose of screening academic publications for content based on known examples of interest. The outcome of such an algorithm is considerable time savings and a more consistent application of criteria regarding what is of interest and what is not. This will reduce the time required of researchers reduce the information overload for practitioners seeking guidance from academic literature.

Likewise, the application of the VSM or LSA classifier will ease the maintenance and updating of SLRs moving forward. By leveraging the

inclusion/exclusion decisions made in previous versions of an SLR, the time required for selection of relevant articles from the continuous stream of academic publications is drastically reduced. Thus researchers can produce updated SLRs more frequently with less effort. Because of the reduced effort required, broader search criteria may be used, creating a larger corpus of candidate documents.

In the context of a SMS such as the one from which the testing dataset for this research is drawn, the results of this research are remarkable with regard to the work savings versus accuracy tradeoff. As mapping studies are designed to map out a domain of research and are therefore more tolerant of lower recall statistics, the performance of the VSM classifier is acceptable. One must consider what level of recall is truly necessary in a given situation. If the risk of injury or loss is a major consequence, additional tuning to achieve higher levels of recall may be required.

Many EBSE research efforts are driven primarily by the need for classifier recall, with researchers willing to sacrifice precision and efficiency. In such cases, the findings of this research lead to the following recommendations:

1. The VSM should be used rather than the competing LSA classifier.
2. The size of the classifier training set should be as large as practically allowed.
3. The training set should include as many truly relevant studies as possible; a journal special issue or a focused research workshop proceedings may be fitting sources of the training set.

Other EBSE research efforts may be more exploratory in nature, and may be driven by the need for precision and researcher efficiency. In such cases, the findings of this research lead to the following guidance:

1. The LSA classifier should be used rather than the competing VSM.
2. The training set should include a large proportion of truly irrelevant studies.
3. The absolute size of the training set can be kept small, thus improving researcher efficiency while having little detrimental impact on precision. This will, however, result in a cost to classifier recall.

If recall, precision and researcher efficiency are all important in equal measure, it is recommended that the training set be comprised of 50% relevant studies (perhaps from a journal special issue or specialized workshop / conference) and 50% irrelevant studies. This balance can result in an effective algorithm without increasing training set size (and, thus, decreasing researcher efficiency).

Vocabulary and dimensionality have been found here to have little practical influence on classifier performance. Thus, it is recommended that classifier defaults be used, in keeping with the goal of minimal algorithm tuning.

7.3. Limitations

Generalization of the results of this research must be approached with caution due to several factors. A significant threat to external validity is the usage of a single SMS-based case study to validate the proposed algorithms. This threat to external validity is mitigated by conducting a small second simulation based on a more focused SLR-based dataset. Future research will test the VSM and LSA automated classifiers on additional SLRs and SMSs to further address this threat.

The dataset utilized in testing is from a broad SMS in which there are several disjoint, but interacting primary topics of interest. Thus the classifier is seeking evidence of one or more topics from among a group of topics of interest in order to include a study. This variety of content may contribute to the classifier matching a study that may have been excluded without the presence of a syntactically similar, yet semantically dissimilar topic. Based on what is known of the LSA algorithm and the comparison measure utilized in this research, one can also speculate that as the number of semantic categories of interest is reduced, the recall and precision performance of the classifiers will improve. Further testing with additional datasets from a variety of topic areas is needed to understand and address this concern.

Similarly, the impact of vocabulary (terms) surrounding a given topic of interest may impact classifier recall and/or precision. The breadth of topics contained in the 5,979-study SMS corpus (see Section 4.1) – which includes 291 SLR studies – is believed to minimize this risk, however additional research is needed to fully understand the impact of vocabulary with regard to topic.

Finally, bias in the original selection of publications for the SMS may have impacted the articles included and thus the training sets and results of this research. The potential bias was minimized by expert review of the protocol, review of the SMS publication selection by two independent researchers, followed by conducting a test/retest sample. However, one must cognizant of the risk regardless.

8. Conclusion and future work

This research demonstrates the feasibility of utilizing VSM family algorithms to automate the EBSE Study Selection process. This allows for considerable savings of researchers' time, as well as more consistent application of inclusion and exclusion criteria. Future research will focus on improving the recall and precision of the classifier through a better understanding of the algorithms in the VSM family.

The use of a VSM or LSA based classifier also opens new avenues of approach to continued maintenance of an SLR. It provides a means by which publications can be scanned in real-time to identify the need for updating based on either a given mass of publications accumulating or a disruptive shift in research findings. With this capability however, come new questions. Will the domain of interest remain confined within the original constraints, or will it slowly expand as studies are added to the collection? Is a periodic "intervention" needed to keep the selection of studies confined to criteria defined by the research, or should it be allowed to grow organically? Further research is needed to understand how entropy and scope-creep may factor into the evolution of a publication set over time, especially when the classifier immediately adds studies identified for inclusion into its training set.

The EBSE topic may impact automated classifier algorithm performance. Where there are significant levels of synonymy and polysemy, where a new topic is emerging from multiple or diverse reference disciplines, the importance of dimensionality may increase and the performance of LSA over VSM may improve. Future research will explore the performance impact of these factors.

Future research will expand upon the generalizability of the results through the testing of the classifiers in additional SLR and SMS research contexts. This will assist in development of theory as to how and why the VSM family of algorithms work, the boundaries in which it works, and allow for empirically founded tuning of the classifier algorithm. Future research will explore the performance impacts of related probabilistic models, including Probabilistic LSA and Latent Dirichlet Allocation.

Acknowledgments

We acknowledge partial support from the United States National Science Foundation under grant NSF-1305395.

References

[1] K. Petersen, S. Vakkalanka, L. Kuzniarz, Guidelines for conducting systematic mapping studies in software engineering: An update, Inf. Softw. Technol. 64 (2015) 1–18.

[2] O. Dieste, O.A.G. Padua, Developing search strategies for detecting relevant experiments for systematic reviews, Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on, 2007, pp. 215–224.

[3] Z. He, M.A. Babar, B. Xu, L. Juan, H. Liguo, An empirical assessment of a systematic search process for systematic reviews, Evaluation & Assessment in Software Engineering (EASE 2011), 15th Annual Conference on, 2011, pp. 56–65.
[4] S. Jalali, C. Wohlin, Systematic literature studies: database searches vs. backward snowballing, Empirical Software Engineering and Measurement (ESEM), 2012 ACM-IEEE International Symposium on, 2012, pp. 29–38.
[5] B. Kitchenham, P. Brereton, M. Turner, M. Niazi, S. Linkman, R. Pretorius, D. Budgen, The impact of limited search procedures for systematic literature reviews – A participant-observer case study, 2009 3rd International Symposium on Empirical Software Engineering and Measurement, ESEM 2009, October 15, 2009–October 16, 2009, IEEE Computer Society, Lake Buena VistaFLUnited States, 2009, pp. 336–345.
[6] B. Kitchenham, Z. Li, A. Burn, Validating search processes in systematic literature reviews, 1st International Workshop on Evidential Assessment of Software Technologies, EAST 2011, in Conjunction with ENASE 2011, June 8, 2011–June 11, 2011, SciTePress, BeijingChina, 2011, pp. 3–9.
[7] H. Zhang, M.A. Babar, On searching relevant studies in software engineering, Proceedings of the 14th international Conference on Evaluation and Assessment in Software Engineering (EASE), 2010.
[8] E.E. Hassler, Advancing Evidence-Based Practice in System Development: Providing Juried Knowledge to Software Professionals, The University of Alabama, Tuscaloosa, AL, 2014.
[9] J.C. Carver, E. Hassler, E. Hernandes, N.A. Kraft, Identifying barriers to the systematic literature review process, Empirical Software Engineering and Measurement, 2013 ACM/IEEE International Symposium on, 2013, pp. 203–212.
[10] A. Al-Zubidy, J.C. Carver, D.P. Hale, E.E. Hassler, Vision for SLR tooling infrastructure: prioritizing value-added requirements, Inf. Softw. Technol. 91 (2017) 72–81.
[11] B. Kitchenham, S. Charters, Guidelines for Performing Systematic Literature Reviews in Software Engineering, Keele University and Durham University, 2007.
[12] E. Hassler, J.C. Carver, N.A. Kraft, D. Hale, Outcomes of a community workshop to identify and rank barriers to the systematic literature review process, Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, ACM, 2014, p. 31.
[13] A.M. Cohen, W.R. Hersh, K. Peterson, P.-Y. Yen, Reducing workload in systematic review preparation using automated citation classification, J. Am. Med. Inform. Assoc. 13 (2006) 206–219.
[14] K.R. Felizardo, N. Salleh, R.M. Martins, E. Mendes, S.G. MacDonell, J.C. Maldonado, Using visual text mining to support the study selection activity in systematic literature reviews, Empirical Software Engineering and Measurement (ESEM), 2011 International Symposium on, 2011, pp. 77–86.
[15] V. Malheiros, E. Hohn, R. Pinho, M. Mendonca, A visual text mining approach for systematic reviews, Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on, 2007, pp. 245–254.
[16] Q. Zhong, Supporting study selection of systematic literature reviews in software engineering with text mining, Information Processing Science, University of Oulu, 2017.
[17] Y. Zhao, G. Karypis, U. Fayyad, Hierarchical clustering algorithms for document datasets, Data Min. Knowl. Discov. 10 (2005) 141–168.
[18] M. Steinbach, G. Karypis, V. Kumar, A comparison of document clustering techniques, KDD Workshop on Text Mining, Boston, 2000, pp. 525–526.
[19] M.W. Berry, S.T. Dumais, G.W. O'Brien, Using linear algebra for intelligent information retrieval, SIAM Rev. 37 (1995) 573–595.
[20] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman, Indexing by latent semantic analysis, J. Am. Soc. Inf. Sci. 41 (1990) 391–407.
[21] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surveys (CSUR) 31 (1999) 264–323.
[22] K.R. Felizardo, S.R. Souza, J.C. Maldonado, The use of visual text mining to support the study selection activity in systematic literature reviews: a replication study, Replication in empirical software engineering research (RESER), 2013 3rd International Workshop On, IEEE, 2013, pp. 91–100.
[23] K.R. Felizardo, S.G. MacDonell, E. Mendes, J.C. Maldonado, A systematic mapping on the use of visual data mining to support the conduct of systematic literature reviews, J. Softw. 7 (2012) 450–461.
[24] K. Romero Felizardo, S.R.S. Souza, J.C. Maldonado, The use of visual text mining to support the study selection activity in systematic literature reviews: a replication study, Replication in Empirical Software Engineering Research (RESER), 2013 3rd International Workshop on, 2013, pp. 91–100.
[25] K.R. Felizardo, G.F. Andery, F.V. Paulovich, R. Minghim, J.C. Maldonado, A visual analysis approach to validate the selection review of primary studies in systematic reviews, Inf. Softw. Technol. 54 (2012) 1079–1091.
[26] A.M. Fernandez-Saez, M.G. Bocco, F.P. Romero, SLR-TOOL: a tool for performing systematic literature reviews, 2010 5th International Conference on Software and Data Technologies (ICSOFT 2010), 22-24 July 2010, INSTICC Press, SetubalPortugal, 2010, pp. 157–166.
[27] K.R. Felizardo, M. Riaz, M. Sulayman, E. Mendes, S.G. MacDonell, J.C. Maldonado, Analysing the use of graphs to represent the results of systematic reviews in software engineering, Software Engineering (SBES), 2011 25th Brazilian Symposium on, 2011, pp. 174–183.
[28] M.-H. Song, S.-Y. Lim, S.-B. Park, D.-J. Kang, S.-J. Lee, Ontology-based automatic classification of web pages, Appl. Soft Comput. Technol. (2006) 483–493.
[29] S. Bloehdorn, A. Hotho, Boosting for text classification with semantic features, WebKDD, Springer, 2004, pp. 149–166.
[30] K. Nigam, A.K. McCallum, S. Thrun, T. Mitchell, Text classification from labeled and unlabeled documents using EM, Mach. Learn. 39 (2000) 103–134.
[31] T. Gruber, Ontology, in: L. Liu, M.T. Özsu (Eds.), Encyclopedia of Database Systems, Springer-Verlag, 2008.
[32] Y. Sun, Y. Yang, H. Zhang, W. Zhang, Q. Wang, Towards evidence-based ontology for supporting systematic literature review, 16th International Conference on Evaluation & Assessment in Software Engineering (EASE 2012), IET, 2012, pp. 171–175.
[33] S. Matwin, A. Kouznetsov, D. Inkpen, O. Frunza, P. O'Blenis, A new algorithm for reducing the workload of experts in performing systematic reviews, J. Am. Med. Inform. Assoc. 17 (2010) 446–453.
[34] F. Tomassetti, G. Rizzo, A. Vetro, L. Ardito, M. Torchiano, M. Morisio, Linked data approach for selection process automation in systematic reviews, Evaluation & Assessment in Software Engineering (EASE 2011), 15th Annual Conference on, 2011, pp. 31–35.
[35] B. Wallace, T. Trikalinos, J. Lau, C. Brodley, C. Schmid, Semi-automated screening of biomedical citations for systematic reviews, BMC Bioinform. 11 (2010) 55.
[36] B. Ramesh, J. Sathiaseelan, An advanced multi class instance selection based support vector machine for text classification, Proc. Comput. Sci. 57 (2015) 1124–1130.
[37] T. Kakkonen, E. Sutinen, Automatic assessment of the content of essays based on course materials, Information Technology: Research and Education, 2004. ITRE 2004. 2nd International Conference on, 2004, pp. 126–130.
[38] T.K. Landauer, P.W. Foltz, D. Laham, An introduction to latent semantic analysis, Discourse Process. 25 (1998) 259–284.
[39] G. Salton, A. Wong, C.S. Yang, A vector space model for automatic indexing, Commun. ACM 18 (1975) 613–620.
[40] F. Wild, C. Stahl, Investigating unstructured texts with latent semantic analysis, in: R. Decker, H.-J. Lenz (Eds.), Advances in Data Analysis, Springer, Berlin, Heidelberg, 2007, pp. 383–390.
[41] T. Hofmann, Probabilistic latent semantic analysis, Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc., 1999, pp. 289–296.
[42] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[43] P. Nakov, A. Popova, P. Mateev, Weight functions impact on LSA performance, EuroConference RANLP'2001 (Recent Advances in NLP), 2001.
[44] P. Nakov, E. Valchanova, G. Angelova, Towards deeper understanding of the LSA performance, In Proc. Recent Advances in Natural Language Processing, Citeseer, 2003.
[45] R. Till, E. Mross, W. Kintsch, Time course of priming for associate and inference words in a discourse context, Memory Cognit. 16 (1988) 283–298.
[46] G.K. Chung, H.F. Neil, Methodological Approaches to Online Scoring of Essays, University of California, Los Angeles, 1997.
[47] N. Evangelopoulos, X. Zhang, V.R. Prybutok, Latent semantic analysis: five methodological recommendations, Eur. J. Inf. Syst. 21 (2012) 70–86.
[48] F. Wild, C. Stahl, G. Stermsek, Parameters driving effectiveness of automated essay scoring with LSA, Proceedings of the 9th CAA Conference, Loughborough, 2007.
[49] D. Hale, E. Hassler, J. Hale, Primary Study Publications Included as Being Related to Se Slr Studies, (2017) ResearchGate.net.
[50] D. Hale, E. Hassler, J. Hale, Primary Study Publications Excluded as Not Being Related to Se Slr Studies, (2017) ResearchGate.net.
[51] D. Hale, E. Hassler, J. Hale, r, VSM & LSA Primary Study Classifier Random Sample Generato, (2017) ResearchGate.net.
[52] R.R. Picard, R.D. Cook, Cross-validation of regression models, J. Am. Statis. Assoc. 79 (1984) 575–583.
[53] D.I. Warton, F.K. Hui, The arcsine is asinine: the analysis of proportions in ecology, Ecology 92 (2011) 3–10.
[54] G.D. Garson, Testing Statistical Assumptions, Statistical Associates Publishing, Asheboro, NC, 2012.
[55] J. Fox, Applied Regression Analysis and Generalized Linear Models, Sage Publications, 2015.
[56] F. Gravetter, L. Wallnau, Essentials of Statistics for the Bahavioral Sciences, 8th ed., Wadsworth, Cengage Learning, Belmont, CA, 2014.
[57] G. Van Belle, Statistical Rules of Thumb, John Wiley & Sons, 2011.
[58] D. Hoechle, Robust standard errors for panel regressions with cross-sectional dependence, Stata J. 7 (2007) 281.
[59] J.T. Richardson, Eta squared and partial eta squared as measures of effect size in educational research, Educ. Res. Rev. 6 (2011) 135–147.
[60] J. Cohen, Statistical Power Analysis For the Behavioral Sciences, Revised ed., Academic Press, New York, 1977.
[61] M. Bano, S. Imtiaz, N. Ikram, M. Usman, M. Niazi, Technical Report of Systematic Literature Review For Causes of Requirement Change, Keele University, Keele, UK, 2010.
[62] K. Petersen, N.B. Ali, Identifying strategies for study selection in systematic reviews and maps, Empirical Software Engineering and Measurement (ESEM), 2011 International Symposium on, 2011, pp. 351–354.