Research Synthesis Methods, 2018, 9(4):587–601. DOI: 10.1002/jrsm.1317

Discriminating between empirical studies and nonempirical works using automated text classification

Alexis Langlois[1], Jian-Yun Nie[1], James Thomas[2], Quan Nha Hong[3], Pierre Pluye[3]

1. Département d'informatique et de recherche opérationnelle, Université de Montréal, Montréal, Canada
2. EPPI-Centre, University College London Institute of Education, London, UK
3. Family Medicine, McGill University, Montréal, Canada

ABSTRACT

Objective: Identify the most performant automated text classification method (e.g., algorithm) for differentiating empirical studies from nonempirical works in order to facilitate systematic mixed studies reviews.

Methods: The algorithms were trained and validated with 8050 database records, which had previously been manually categorized as empirical or nonempirical. A Boolean mixed filter developed for filtering MEDLINE records (title, abstract, keywords, and full texts) was used as a baseline. The set of features (e.g., characteristics from the data) included observable terms and concepts extracted from a metathesaurus. The efficiency of the approaches was measured using sensitivity, precision, specificity, and accuracy.

Results: The decision trees algorithm demonstrated the highest performance, surpassing the accuracy of the Boolean mixed filter by 30%. The use of full texts did not result in significant gains compared with title, abstract, keywords, and records. Results also showed that mixing concepts with observable terms can improve the classification.

Significance: Screening of records, identified in bibliographic databases, for relevant studies to include in systematic reviews can be accelerated with auto- mated text classification.

Keywords: automated text classification, decision tree, health care, research method, support vector machine, systematic review

1. CONTEXT

Researchers, policymakers, and practitioners are increasingly interested in literature reviews can

be used to justify, design, and interpret results of primary studies. Their growing popularity is mainly due to the increasing interest in evidence-informed decision-making and the need to have rigourous methods to identify and synthesize research. To synthesize research results, preference is given to systematic reviews since they use reproducible methods and are reported in a transparent manner.[1] Systematic reviews are considered epistemologically, methodologically, and practically relevant since they synthesize the best available evidence for a specific question. Moreover, they are increasing in popularity; the growth of the annual number of published systematic reviews largely exceeds that of other types of publications at least since 2010.[2]

Over the past decade, mixed studies reviews have emerged as a new type of systematic review. They apply mixed methods approaches to critically analyse, synthesize, and integrate the findings of empirical studies.[3-5] Moreover, given they combine empirical evidence from qualitative, quantitative, and mixed methods studies, these reviews can provide a rich understanding of complex phenomena. Although empirical research has a clear definition (based directly on observation, experiment, or simulation, rather than on reasoning or theory alone),[6,7] because mixed studies reviews include all types of empirical research designs,[3] search strategies often yield a high number of records to screen (sometimes more than 10 000). In fact, many of these records are totally irrelevant. The high yield means that the screening process is time consuming. Unlike reviews of randomized controlled trials, for example, because systematic mixed studies reviews include all types of design, no term referring to study design can be used to capture them. Empirical research is not referred to as empirical in articles, but rather by study design.

In addition to that, it is estimated that approximately 1.4 million articles are written every year in scientific journals.[8] Estimates also indicate that the entire systematic review process typically takes about 12 months,[9] which may include 1 or 2 months for manual screening of records. This time scale can be problematic for researchers limited in resources. As a result, a high number of irrelevant entries must be filtered. One common practice in systematic reviews is to use highly sensitive search filters to narrow the search for relevant records. The filters (or classifiers) have been developed for a very specific purpose, specific study type design (e.g., randomized controlled trials[10]) or discipline (e.g., primary care[11]). Traditional search strategies in bibliographic databases generally have high sensitivity (i.e., recall in computer science) and specificity for randomized controlled trials but are limited for other types of research study designs.[12] Since mixed studies reviews are interested in several types of designs, these filters cannot be used. Also, several nonempirical works such as opinion letters, commentaries, editorials, reviews, and errata form a group of irrelevant records that are difficult to identify using traditional search filters because they often follow a research paper format (introduction, method, results, and discussion).

El Sherif et al[13] proposed a mixed filter based on Boolean expressions to facilitate the identification of empirical studies for systematic mixed studies reviews. This Boolean filter covers quantitative, qualitative, and mixed methods studies and includes keywords and subject headings for identifying empirical studies and excluding nonempirical works. This filter has shown high sensitivity (89.5%), but its precision and specificity are just over 50%. The task of identifying empirical studies can be cast as a text classification problem since it can be resolved with two classes: relevant (empirical) or irrelevant (nonempirical). Automated text classification is "the activity of labelling natural language texts with thematic categories from a predefined set of data."[14] Also, automated text classification algorithms have the potential to provide users with a confidence or likelihood scale for each prediction. Automated text classification approaches are promising avenues for reducing the burden of screening of thousands of irrelevant records often captured in bibliographic data base searches for systematic reviews. In medical topic-specific searches, it was shown that these methods may reduce screening time by half without any loss of relevant records.[15] Studies about the effectiveness of automated text classification for screening papers in systematic reviews are increasingly being published.[16-19]

Extant research mainly focusses on topic-specific algorithm training where algorithms are conditioned to measure the relationship between a research question and a study. Little is known about the automated identification of potential relevant studies for systematic mixed studies reviews. Indeed, no research has been done to evaluate the performance of automated text classification for reviews based on research methods. Therefore, the objective of this study was to identify the most performant algorithm to distinguish empirical studies from nonempirical works, thereby facilitating the search and filtering of qualitative, quantitative, and mixed methods studies. The objectives of this study cover the following points:

1. Identify the relevant characteristics (i.e., features) of both classes of document (i.e., empirical and nonempirical).
2. Compare the most popular text classification methods with the Boolean "mixed filter."
3. Design a fitted model based on the most efficient algorithm and features.

2. METHODS

2.1 Text Collection

The text collection is a training set of preclassified records that are used to test the algorithms. This text collection consisted of sets of titles, abstracts, and full texts.

2.1.1 Titles and abstracts

In order to train and test the different algorithms, we used several collections. The first contains the 5516 entries extracted from seven journals (covering three areas: medical informatics, public health, and primary care) assembled by the developers of the Boolean mixed filter for evaluating its performance.[13] Second, we reused screened records and results from previous systematic reviews.[20-27] These reviews cover a broad range of topics, from electronic prescription usage and participatory research to dementia and online health care. In total, approximately 10 000 records were gathered. After removing entries with- out abstract or full text, 8050 were included in the final col- lection. The relevant entries (i.e., empirical) were labelled "1," and the irrelevant entries (i.e., nonempirical) were labelled "0." Only titles and abstracts were considered in our initial experimentations. Subsequently, full texts were used for performance comparison. Table 1 shows the final collection distribution.

2.1.2 Full texts

Researchers can obtain full texts automatically from reference management software, provided their institution has access. Thus, we also measured the benefits of incorporating full texts to the classification task. It should be noted that this evaluation is experimental since the availability of such content depends on database subscriptions.

Full texts (PDF) were automatically using EndNote or retrieved manually via Google Scholar. In order to convert PDF files into usable text files, we used Tika,* a content analysis toolkit developed for different document formats. It should be noted that this conversion can be fully automated using the Tika application program interface.

* https://tika.apache.org.

2.2 Datasets

To train the automatic classifiers (algorithms) and, thus, adjust the parameters of their mathematical functions described below, the final collection had to be separated into three datasets: a training set, a validation set, and a test set. The classifiers were tested on the same entries as the Boolean mixed filter using a fourfold cross validation. Therefore, each distinct fold contained 1136 entries for testing, 1000 entries for validation (i.e., optimization), and 5914 entries for training. Entries were selected randomly while keeping the same category ratio (i.e., empirical/nonempirical) between folds.

2.3 Baseline

The algorithms were compared with the Boolean mixed filter[13] as it is the only approach to

distinguish empirical studies from nonempirical works. Developed by librarians and researchers with expertise in systematic mixed studies reviews, this filter consists of a combination of subject headings and keywords associated with randomized controlled trials, nonrandomized and descriptive quantitative studies, and qualitative and mixed methods studies and has been implemented for MEDLINE, an online bibliographic database. With a search engine like the one provided by MEDLINE, it is possible to build complex queries using the Boolean operators AND (i.e., all keywords included), OR (i.e., any keywords included), and NOT (i.e., keywords not included). As such, the filter includes the expression "NOT (letter OR comment OR editorial OR newspaper article).pt." to exclude possible irrelevant publication types (".pt."). Terms associated with relevant methodologies like "case-control," "focus group," and "grounded theory" are combined with the operator OR and searched for in titles and abstracts. To maintain flexibility, some keywords are truncated with the opera- tor "*," allowing the search engine to look for a portion of the words like "random*," "control*," and "evaluation stud*." The Boolean filter and its toolkit are available online.†

† http://toolkit4mixedstudiesreviews.pbworks.com.

2.4 Text characteristics

Automatic text classification relies on features (i.e., characteristics or properties) extracted from the texts. The features we used are terms and concepts as outlined below.

2.4.1 Terms

Terms are stemmed words that we generated as follows. The words composing the abstracts and titles were used to create the initial representation of each record. Terms were determined as follows. First, common words‡ such as "of" and "from" were removed from the documents. Words were then stemmed using the Porter algorithm.[28] The latter is commonly used in natural language processing to standardize singular and plural forms as well as inflected words. An internal representation of a document was then created using the extracted terms as well as their weighting. An example of internal document representation is a vector in the space formed by all the terms. Numerous indexation methods can be used for this.[29] TF-IDF is the most common method for term weighting and it balances the local representativeness of a term within a document and the global discrimination of the term in the whole dataset. It should be noted that this is the technique mostly commonly used in text classification.[30] The values can be calculated as follows:

$$\frac{f_{t,d}}{|d|} \cdot \log\left(1 + \frac{N}{n_t}\right),\ n_t > 0 \tag{1}$$

where $f_{t,d}$ is the frequency of term t in document d, |d| is the length of document d, N is the total number of documents and n_t is the number of documents containing term t.

‡ www.ncbi.nlm.nih.gov/books/NBK3827/table/pubmedhelp.T.stopwords.

2.4.2 Feature selection approaches

Not all of the selected terms may be useful for the task of classification. Thus, to eliminate irrelevant terms and decrease computational load, features were filtered using a feature selection approach.[31] We compared three different feature selection methods: information gain, χ^2 statistic test, and document frequency. Information gain can be translated as the difference between the portion of irrelevant entries considering all features and the portion of irrelevant entries given a specific feature:

$$\mathrm{IG} = \mathrm{H(E)} - H(E|t) \tag{2}$$

where H(E) is the portion of irrelevant entries in the collection E and H(E|t) is the portion of irrelevant entries in E given a feature t.

The χ^2 statistic test method measures the dependency between a term and its category (empirical or nonempirical):

$$x^2(t,c) = \frac{N \times (AD - CB)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)'} \tag{3}$$

where *A* is the number of times term *t* and category *c* co-occur, *B* is the number of times t occurs without *c*, *C* is the number of times c occurs without *t*, *D* is the number of times neither *c* nor *t* occurs, and *N* is the number of documents.

The document frequency method measures the number of times a term *t* occurs in a document (i.e., the text representing a record).

Based on these three calculations, the features obtaining the highest values are selected and used in the classification algorithms. Using our text collection, information gain and χ^2 statistic test generated zero values for terms excluded from the top 8000. As a result, the amount of terms selected for each

measure was set to 8000, accordingly.

2.4.3 Concepts

Many concepts in the Boolean mixed filter[13] are compound words and cannot be captured by single terms. Using a metathesaurus is a simple way to consider complex and potentially important concepts in the indexation process. To this end, we used the Unified Medical Language System (UMLS) that provides a set of possible expressions for each concept, and relationships between concepts.[32] The selection process used a custom script divided in two parts: concepts in UMLS metathesaurus were stemmed and then searched for in the documents. For this task, all the concept identifiers (CUI) listed by UMLS were considered, and their associated names were added in the new set of features. The concept identifiers are located in a rich release format (RRF) file provided with the metathesaurus. In total, 2101 relevant concepts were extracted from the dictionary and added to the vectors.

2.5 Algorithms

Multiple studies have compared traditional text classification approaches for various problems.[14,33,34] Below, we describe these approaches that are strong options for easily exploiting machine learning algorithms for automatic text classification.

2.5.1 K-nearest neighbours (kNNs)

K-nearest neighbour predicts the category of a test document using the most common category of the surrounding documents (i.e., nearest neighbours) in the feature space. K-nearest neighbour is one of the best-known statistical approaches for supervised text classification.[35] Among a set of training documents, the algorithm tries to identify the k closest entries from a test entry x. The majority category of the k entries is then used to classify x following a proximity weighting formula. For a test document x and a distinct training entry v, we used the Euclidian distance to represent the similarity (i.e., proximity) of both entries:

$$sim(x, v) = \left(\sqrt{\sum_{i=1}^{m}(x_i - v_i)^2}\right)^{-1} \quad (4)$$

where x_i and v_i are the ith features of weight vectors x and v, respectively.

The k documents with the highest $sim(x, v)$ values were selected to represent the category of x.

The estimated probabilities of *x* being empirical or nonempirical were calculated as follows:

$$P_0(x) = \frac{1}{\sum_{i=1}^{k} sim(x, V_i)} \sum_{i=1}^{k} g(x, V_i, 0)$$

$$P_1(x) = \frac{1}{\sum_{i=1}^{k} sim(x, V_i)} \sum_{i=1}^{k} g(x, V_i, 1)$$

(5)

where V_i represents the ith nearest neighbour and $P_0(x)$ and $P_1(x)$ represent the likelihood of negative and positive categories, respectively.

Weighting function *g* can be formulated as follows:

$$g(x, V_i, c) = \begin{cases} \frac{1}{sim(x, v)} & if\ y_i = c \\ 0 & else \end{cases}$$

(6)

where y_i is the category of document V_i. The final category 0 else was based on the maximum between $P_0(x)$ and $P_1(x)$.

2.5.2 Naive Bayes

Naive Bayes classifiers are commonly used for auto- mated text classification. Despite the fact that Naive Bayes approaches ignore all dependencies between features, they are still competitive with high-capacity algorithms.[36] Because of this strong assumption, Naive Bayes may identify the winning category with disproportionate probabilities in some cases. Hence, the approach may provide inaccurate estimations but can still be efficient in providing the correct predictions with a large enough dataset. The typical assumption is that continuous data or features (i.e., quantitative data that can be measured) are distributed according to a normal distribution. Two estimators were used for both categories of documents. Training of the classifiers for a document *x* of dimension m was calculated with the following conditional probability:

$$\hat{p}(x|c) = \prod_{i=1}^{m} P(x_i|c)$$

(7)

As stated above, the probability of observing component x_i with category c is modelled as a normal distribution. The final model follows Bayes' formula and choose the category with the highest probability:

$$P(c|x) = \frac{p(x|c)P(c)}{p(x)} \tag{8}$$

where $P(c)$ is the prior likelihood of category c.

2.5.3 Support vector machine (SVM)

Support vector machine can be considered as a representation of entries as points in space, where the greatest possible distance between entries from opposite categories is sought. It is one of the most popular approaches for binary classification. Based on risk minimization, the objective of the algorithm is to find the optimal hyperplane $w^T x + b$ that separate two predefined categories. To address non-linearity, soft margins and higher dimension projections may be considered. We used the LibSVM implementation with a linear kernel to generate our classifier.[37]

2.5.4 Decision trees

Decision trees combine a set of approaches based mainly on rules.[38] They are especially useful for text classification problems since their predictions are easily interpretable. Many versions are exploitable and can be differentiated by their underlying algorithms and pruning techniques. The most common variants for this category of approaches are ID3 and its successor C4.5.[39] We used the latter along with its reduced error pruning (REP) method. C4.5 tries to minimize the entropy (ie, portion of irrelevant entries) of a group of documents by splitting them into two different subsets using a rule generated by discretization. The latter process aims to summarize the behaviour of the features using conditional operators such as >, <, ≤, or ≥. Let E be the initial training set and let E_1 and E_2 be the two subsets resulting from the separation of E using a split based on a feature. Using the entropy of these three sets, the best possible separation rule is defined as the one that provides the highest information gain. This can be calculated as follows:

$$H(E) = -\sum_{c} E_{1,c} \log_2 E_{1,c} - \sum_{c} E_{2,c} \log_2 E_{2,c}$$

$$IG(E) = \left(-\sum_{c} E_{2,c} \log_c \quad E_{2,c} \right) - H(E) \tag{9}$$

where $E_{i,c}$ and E_c are the proportion of documents belonging to category c in E_i and E, respectively.

This process is recursively applied until entropy cannot be further minimized. Pruning is then used to eliminate unnecessary splits based on the predictions of left out documents (i.e., randomly and automatically selected from the training sets before the pruning process).

2.6 Method refinement

To improve the classification results of the approaches mentioned above, we used additional techniques: bagging and booting, feature combination, linear interpolation, and titles. It is important to mention that these techniques do not represent additional distinctive algorithms but can be seen as different ways to enhance the performance of the approaches already presented.

2.6.1 Bagging and boosting

The previous algorithms can be combined and seen as a series of prediction votes (ie, voting techniques). It has been demonstrated that voting techniques have the potential to increase the stability and capacity of traditional algorithms for automated text classification.[40,41] Comparisons have shown appreciable gain of precision using diversified datasets. Since a vote simply corresponds to the aggregation of predictions provided by a group of classifiers, voting techniques can be applied without additional complexity. They can be seen as meta-algorithms since they rely on the predictions of first-level algorithms. For the most part, aggregation represents the average of the predictions generated by high-capacity classifiers. This representation is also referred to as bagging (i.e., bootstrap aggregating). It is also possible to aggregate the predictions of multiple low capacity classifiers or weak learners (ie, boosting). The following formulas describe these two approaches. Assuming an arbitrary training set E separated into j subsets randomly generated with replacement. For each subset E_i, a traditional classifier H_i can be trained. In order to aggregate the predictions for a test document x, the following formula was used:

$$\Pr(x) = \frac{1}{k} \sum_{i=1}^{k} H_i(x)$$

(10)

where $H_i(x)$ is the prediction of the classifier H_i given x.

As for the boosting approach, a first weak learner H_i is trained on dataset E. Prediction results are then memorized in a vector. Subsequently, a second weak learner H_{i+1} is trained on E while making sure misclassified entries from H_i are better categorized. A total of m weak learners are trained iteratively following the same operation. The importance of each learner H is determined by a coefficient α that is based on the error rate of the learner. The error rate often represents the sum of the errors generated by the weak learner. Hence, a learner producing fewer errors will have a greater α value. Similar to the bagging approach, weak learners are then combined to determine the category of a document:

$$\Pr(x) = \sum_{i=1}^{k} \alpha_i H_i(x) \tag{11}$$

where $\alpha_i \geq 0$.

The Adaboost.M1 algorithm was used to represent this approach.[42]

2.6.2 Feature combination

Quantitative research methods rely largely on statistical explanations. Thus, numerical terms represent an important part of the entries implicated in the classification process. For instance, numbers may be observed in the form of percentages, *P* values or quantities. Because the variation of number values should not influence the predictions of the classifiers, a separate *Numbers* feature of documents was generated by merging these particular features. The feature was weighted as follows:

$$x' = (w_1, w_2, \ldots, w_{m-|Q|}, \frac{1}{|d|} \sum_{t \epsilon Q} f_{n,d}) \tag{12}$$

where $f_{n,d}$ is the frequency of a numeric expression n in document d, Q is the group of numeric expressions in document d, and $|d|$ is the length of the document.

Mathematical and statistical symbols are commonly observed in documents containing quantitative research methods. In addition to percentages (%), a large number of texts contain variables (e.g., σ, α, β, and μ), operators (e.g., +, =, ±, <, and >), and fractions or calculus symbols (e.g., ², ³, ½, and ¼). Their occurrences in a document provide additional clues regarding its category. Thus, an

additional *Maths* feature was created and weighted in the same way as the number feature.

Unified Medical Language System provides concept associations such as synonyms. As such, by merging terms based on these relations, features may gain in homogeneity. Thus, we generated additional $Synonym_k$ features combining the weights (i.e., frequencies) of concepts and terms appearing in an observed group of synonym *k*. It is important to mention that number and symbol combinations presented above could have a bigger impact on quantitative methods.

Merging of features was done separately for the three methods. Afterward, an additional evaluation was per- formed using a mix of all combinations (i.e., *Numbers, Maths, and Synonyms*).

2.6.3 Linear interpolation

The different text characteristics described above (i.e., terms and concepts) can be combined during the classification process. Yet, the significance of both types of feature can also be measured in order to grant a greater degree of importance to a specific group of terms or concepts. Smoothing techniques are often used for such evaluation and are particularly popular for natural language models.[43] Linear interpolation (i.e., Jelinek-Mercer's method) is a common approach that allows the com- bination of two different classification models. Specifically, the approach uses a coefficient λ that controls the influence of two separate groups of characteristics (θ_A and θ_B):

$$P(x|\theta) = \lambda P(x|\theta_A) + (1-\lambda)P(x|\,\theta_B) \tag{13}$$

Smoothing is particularly useful for classifying the model based on decision trees (M1). For upper nodes, decision trees are inclined to favour terms that are unrelated to the problem when separating training data (e.g., terms associated with a journal rather than a research method). This phenomenon may affect the generalization of the two categories. Therefore, weights associated with this kind of feature should be penalized. Using the 8000 terms and 2000 concepts previously calculated, let $T \in R^a$ be the weight vector of terms not included in UMLS and $C \in R^b$ be the weight vector of matching concepts for document *x*. Predictions based on linear interpolation and decision trees can be translated as follows:

$$\Pr(x|T,C) = \lambda\left(\sum_{i=1}^{k} P_i(x|T)\right) + (1-\lambda)\left(\sum_{i=1}^{k} P_i(x|C)\right) \tag{14}$$

where $P_i(x)$ represents the probability distribution of document *x* generated by the *i*th tree.

Linear interpolation was tested with λ-values set to 0, 0.25, 0.5, 0.75, and 1.

2.6.4 Titles

Examining terms in the article titles provides important indications regarding the methodology used. To date, in the description of text characteristics, document representations do not differentiate terms from the abstracts and titles. Although term frequency for titles is meaningless, presence and absence indications may be valuable. These features can be represented as simple binary values. Let *title(x)* be the title of document *x*. New features $\alpha_i \in \{0, 1\}$ can be generated as follows:

$$\alpha_i = \begin{cases} 1 \; if \; t_i \in title \; (x) \\ 0 \qquad else, \end{cases} \tag{15}$$

where t_i is the ith term observable in the titles.

By reconsidering the model presented in (14), α components can be merged with vectors T and C. Since concepts are considerably less frequent in titles than regular terms, vectors T were chosen to carry the new features:

$$x_{title} = (x_1, x_2, \dots, x_a, \alpha_1, \alpha_2, \dots, \alpha_l)$$

$$\Pr(x_{title}|T, C) = \lambda \left(\sum_{i=1}^{k} P_i(x_{title}|T) \right) + (1 - \lambda) \left(\sum_{i=1}^{k} P_i(x|C) \right) \tag{16}$$

where *l* is the total number of terms observable in titles.

Terms composing the titles were also evaluated separately in order to measure their capacity to describe the nature of a study.

2.7 Implementation

The approaches were implemented using Weka,[§] an application program interface that provides a collection of several machine learning algorithms. The features were extracted using custom scripts developed in programming language Python. The entries were indexed (i.e., term weighting) in this same language. Once the best method was selected, a more user-friendly and convenient tool[¶] was programmed for researchers. The source code (Java and Python) as well as our original datasets are openly accessible at the same location and can be tested on projects and additional entries, thus, improved. New entries will also be made available over time along with the tool. Otherwise, please do not hesitate to contact the authors for an access to the data.

§ http://www.cs.waikato.ac.nz/ml/weka/. ¶ https://atcer.iro.umontreal.ca.

2.8 Evaluation

Algorithms were directly compared with the Boolean mixed filter (labelled "baseline"). Since sensitivity, precision, specificity, and accuracy were used to evaluate the filter and considered for the new automatic text classifiers (algorithms). The four indices were calculated as follows:

$$\text{Sensitivity} = \frac{TP}{TP+FN}$$
$$\text{Precision} = \frac{TP}{TP+FP}$$
$$\text{Specificity} = \frac{TN}{TN+FP}$$
$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (17)$$

where TP = number of true positives, TN = number of true negatives, FP = number of false positives, and FN = number of false negatives.

3. RESULTS

3.1 Algorithms

A total of 8000 terms exclusively chosen by information gain were kept. Table 2 shows the performance of the six most efficient automatic text classification approaches tested. Note for first-level classifiers that were not improved by bagging and boosting techniques, the associated results are not included in Table 2. The additional method refinement techniques were evaluated separately (see Section 2). Bagging was tested with decision trees, Naive Bayes, kNN, and SVM. Boosting was tested with decision trees, Naive Bayes, and kNN. Most classifiers tended to perform better with nonempirical documents. The decision trees with bagging (M1) approach performed well for empirical entries (>0.8) and increased the accuracy by 31.7% compared with the baseline. Support vector machine (M3) outperformed kNN and Naïve Bayes as well. These results informed the subsequent evaluations that were performed using the two best families of algorithm, that is, the decision trees (with bagging) and SVM.

3.2 Concepts

Figure 1A,B shows the progression of accuracy for decision trees with bagging (M1) and SVM

(M3) when concepts provided by the metathesaurus are added to the weight vectors. A maximum gain of 0.2% can be observed for decision trees. When 2000 concepts are considered in the classification process, precision, sensitivity, and specificity of decision trees with bagging increase by 0.38%, 0.1%, and 0.23%, respectively. As for SVM, accuracy gained 0.5% at 2000 additional concepts. At the same level, precision increased by 1%, sensitivity by 0.2%, and specificity by 0.6%. Table 3 gives an overview of the new performances for both algorithms.

We experimented with different numbers of concepts as features. Figure 1 shows how accuracy changes according to the number of concepts with M1 and M3. Our results indicate the ideal number of concepts for M1 and M3 is, respectively, 2000 and 1200.

To further assess the influence of concepts, we examined the top 50 features (including terms) selected using information gain. Our results indicate that 67% of these features are concepts included in UMLS. This shows that concepts are extensively used by the classification algorithms. The fact that the addition of concepts did not increase performance measures by large margins can be explained by the overlap between terms and concepts: most of these concepts would have been covered by terms if concepts are not used.

Figure 2 shows a list of lemmatized concepts from the initial group of 2000 frequently selected by decision trees. Results indicate that these are multi-word concepts (which are more precise than single words or terms).

3.3 Method refinement

3.3.1 Feature combination

Features were combined following the methods presented in the previous section. Table 4 provides an overview of how the different combinations, using decision trees with bagging and SVM, performed.

Combining synonyms and symbols slightly increased SVM accuracy (+0.09% and +0.02%, respectively). However, most combinations negatively affected decision tree final predictions.

3.3.2 Linear interpolation

Results are shown in table 5. When $\lambda = 0.75$, feature smoothing increased accuracy and precision by 0.16% and 0.5%, respectively. Although concepts from the thesaurus provide substantial support to predictions, regular terms still have a greater impact on the model. Compared with the approach that combined concepts and terms, the smoothing approach is slightly more effective.

3.3.3 Titles

Table 6 lists some examples of predominant design indications that can be observed in titles and associated with a specific category.

Table 7 summarizes the scores of the new model in com- parison with decision trees and bagging without smoothing for abstracts. The best results were obtained when λ is perfectly balanced (0.5). Accuracy increased by 0.07% as opposed to the previous smoothed models, and by 0.23% as opposed to decision trees without smoothing. In total, 137 occurrences of features associated with the titles are exploited by decision trees to split the training set. However, the contribution of these new features is arguable.

3.3.4 Full texts

Based on previous results, we used the classifier based on decision trees with bagging to evaluate its performance using full texts exclusively.

Table 8 shows the gains from abstract to full text representations when concepts are added to the vectors and the three combination approaches are applied. Full text classification is positively influenced by the new features in every case, with the exception of synonyms. Combining the numbers has the greatest positive impact on predictions, with a precision increase of 0.6%. Concepts have a sensitivity gain of 1.23% compared with 0.1% for abstracts.

Table 9 shows the overall performance of the interpolation model (λ = 0.5) for both empirical and nonempirical entries on abstracts and full texts. Although full texts include more detail than abstracts, the final scores for both types of content is similar. When feature combination is active, the most discriminating terms/concepts reported by decision trees (M1) are both involved in full texts and abstracts, which explains the similar results.

4. DISCUSSION

The general observation on the classification algorithms shows that decision trees and bagging perform best, followed by SVM. These three algorithms are clearly better than Naive Bayes and kNN algorithms tested in this study. Moreover, they performed better than the manual Boolean filter (baseline) suggesting they can be used in pace of this filter. An important advantage of automatic classifiers is that they can be trained automatically. Our experiments show that words (terms) are the basic useful features that one can extract and select from abstracts and full texts. Additional features based on numerical and mathematical expressions, as well as concepts, can provide small, but limited, improvements (especially when full texts are used).

Prediction errors generated by decision trees and SVM (M1 and M3) occur with various research methods. Therefore, it is not possible to propose a general solution to improve the classifiers. Additionally, some publication types are often mentioned in both empirical and nonempirical records. For example, “action research” occurs in 246 abstracts of nonempirical works and 384 abstracts of empirical studies. This issue is not uncommon. Our results indicated that predictions for randomized controlled trials are influenced by ambiguous terms like “trial” (475 negative abstracts and 241 positive abstracts). However, most of the prediction errors made by the decision trees with linear interpolation model share common characteristics regarding false positives. Numerous entries labelled as negative and containing empirical research method keywords were incorrectly identified by the algorithm. Meta-analysis and reviews are directly linked to this problem. In our study, it was not unusual to observe co-occurrences of concepts related to opposite classes such as “review” and “con- trolled trial” (325 abstracts), “review” and “cohort study” (176 abstracts), “meta-analysis” and “controlled trial” (133 abstracts), as well as “meta-analysis” and “case-control” (46 abstracts).

False negatives were less common given that letters, editorials, commentaries, and errata were usually correctly identified by both decision trees and SVM. In fact, precision for the negative class was considerably higher (+92%). However, there are a few similarities among false negatives for all the classification methods. More than half of these abstracts did not follow a typical structure with keywords such as “objective,” “results,” and “conclusion”. Also, short abstracts with vague descriptions were often rejected by all the algorithms we tested. Finally, negative concepts are sometimes included in empirical studies. For instance, we observed “review” 293 times in false negatives, “systematic” 121 times, and “analysis” 192 times.

A benefit of using automated text classification methods, other than SVM, for categorizing empirical studies is their ability to provide a confidence score along with the predictions. Even though Naive Bayes and kNN provided irregular distributions for correct and incorrect predictions, decision trees resulted in a relatively coherent model for confidence scores. Regarding feature interpolation, the average disparity between the actual and the predicted classes was 19.21% with a median of 18.3%. In practice, for librarians requiring a reliable confidence scale, these results may be acceptable. To illustrate, a user who chooses to set the confidence threshold of the algorithm to 33% is able to get a greater sensitivity without undue interference on the precision.

Regarding the three methods of feature combination on abstracts, poor overall performance was observed when used on abstracts only. These results can be explained in four ways: insufficient detail in abstracts, ambiguous connotations of numerical terms, uneven distribution of mathematical symbols within categories, and limited coverage of synonyms. An important aspect that is difficult to capture by

number combinations is the variation of meanings associated with particular features. Occurrences of term "2" in different sequences such as "2 years old," "type 2 diabetes," "p = 2," and "2 patients" do not hold the same information. Since documents are very short, this phenomenon may have a negative impact on classification. Furthermore, in this study, merging mathematical and statistical symbols in a single feature did not lead to noticeable improvements in performance. Upon further examination, our data show that symbols have low occurrence frequency per document. In fact, the median of symbol occurrences in empirical documents is close to 1 and nearly 0 for nonempirical articles.

A similar problem can be observed for the approach based on synonym combination. Specifically, a group of synonyms contains only eight concepts, on average, with a low frequency per document. As a result, the scope of each group is considerably reduced. The use of hypernym relations (i.e., broader concepts) proposed by UMLS, for instance, may address this problem. These relations are particularly popular for document and query expansion.[44] Despite the potential impact of synonym combinations on the classification of all three types of research methods (i.e., quantitative, qualitative, and mixed), we were wary of the fact that number and symbol combinations could result in a bias towards quantitative and mixed methods. Nevertheless, the proposed automated text classification for systematic mixed studies reviews is promising as it suggests researchers can use supervised machine learning for screening records. In comparison with manually screening titles and abstracts, combining this method specific automatic text classification method with topic-specific automated text classification could potentially save hours of work by, for the most part, reducing the number of irrelevant records to manually screen. Future work could test this. Provided that reviewers can retrieve full-text publications in an automatic, systematic, and reliable manner, the proposed algorithms may represent an important innovation and transform systematic review processes.

Given the absence of universal access to full-text publications, a combination of abstracts and full texts can be used as training data to enhance the predictions of M1 and M3. Figure 3 presents a possible scenario, illustrating the performance progression according to the ratio of full texts to abstracts in the collection. There is a high correlation between the general performance of our algorithm based on decision trees/bagging and the variation of the full-text ratio. However, sensitivity appears to be negatively affected by the mixture. There is also a decrease of almost every performance index when full-text ratio is relatively small. Alternatively, two distinct classifiers could be used: one for abstracts and one for full texts. In such a scenario, abstracts would need to be automatically differentiated from full texts prior to classification.

Assuming an almost complete availability of full content provided by *Google Scholar*, reviewers would still need an automated tool to extract full texts from the pages listed by the search engine. Such

a tool may require a web crawler[45] and a complete evaluation on a generic data collection. It is important to note that we have not proposed a tool for this type of operation.

The proposed automated text classification (M1) performs very well for excluding nonempirical works (high precision is important for negative class), that is negative sampling. This suggests potential applications for future systematic and nonsystematic reviews. First, in systematic mixed studies reviews, high sensitivity is key. Reviewers seek the entire population of studies (exhaustive search in a comprehensive set of bibliographic databases and in the grey literature) to answer specific questions (qualitative and/or quantitative). For example, "In population P, what is the effectiveness of the intervention I (com- pared to intervention C) regarding the Outcome O?" and "what are the views and the life-experience of end-users and their relatives with regard to the planning, implementation, evaluation and sustainability of intervention I?" Thus, researchers could consider using M1 as an initial screening/filtering procedure to exclude irrelevant documents with high precision. Two independent researchers could then proceed with manual screening to select relevant studies to include in the review. To ensure no relevant studies are lost with the initial automatic text classification step, the process could be completed with citation tracking.[46]

Second, M1 can be of interest in nonsystematic reviews, where an exhaustive search is not required. For example, for theses and dissertations, graduate students do not conduct exhaustive searches of all relevant publications could save time by combining the Boolean mixed filter (high sensitivity) with M1 (high specificity) to obtain a large (good enough) sample of studies. Likewise, sensitivity is not an issue in nonsystematic scoping reviews[47] where reviewers seek a sample of the population of studies to address a large-scope (broad) question. Thus, researchers could consider combining the Boolean mixed filter (high sensitivity) to rule in a large sample of publications and M1 (high specificity) to rule out nonempirical work.

Automated text classification can be easy to use. For example, if made available online, reviewers could export their records from reference management software. The tool would classify records into empirical and nonempirical sets of records. The two sets of records could then be imported to the reference management software. We have built a complete M1 tool (including a user interface) for categorizing records saved in a spreadsheet. The Method Development component of the Quebec-SPOR SUPPORT Unit is currently building and testing a website to disseminate the Automated Text Classification of Empirical Records (ATCER) and a user guide. The user guide will include the abovementioned recommendations for using the algorithm and its limitations. To access the website, please go to https://atcer.iro.umontreal.ca.

5. LIMITATION

One major aspect to consider regarding the results of this study is the limited amount of data used for training the algorithms. Knowing that *PubMed* alone includes at least 420 000 randomized controlled trials and almost 800 000 clinical studies, further tests should be performed to ensure the performance results of the algorithm we report herein are not influenced by our limited data distribution. However, for such tests, the mass extraction of training data from bibliographic databases should be supervised to ensure valid labelling (i.e., empirical vs nonempirical).

Some drawbacks to using decision trees should be noted. The risk of overfitting is high, even with the use of pruning techniques. This occasionally applies when an algorithm is overtrained on a collection that does not represent the full population. For instance, commonly occurring research questions/disciplines and methods can greatly influence the categorization. In addition, decision trees are relatively unstable. In other words, small modifications applied to the training set can lead to very different predictions. Because of these difficulties, feature selection and training must be based on balanced collections with diversified methodologies.

6. CONCLUSION

Automated text classification of empirical studies (vs nonempirical works) is a promising option to use when conducting nonsystematic literature reviews, but further testing is required to verify its performance for systematic reviews. We propose a supervised machine learning algorithm that can facilitate the identification of empirical studies in bibliographic databases (i.e., the search for qualitative, quantitative, and mixed methods evidence) for systematic reviews. This can be used as an alternative or a complement to the Boolean mixed filter. Our results suggest that decision trees can surpass the accuracy of manual queries by at least 30% without influencing sensitivity. More importantly, the presented models obtained very high precision scores (+92%) for nonempirical works and could be used for removing entries rather than selecting studies.

The use of separate features for concepts (extracted from a metathesaurus) and terms in titles moderately increased the performance of our methods. Varying the weights between terms and concepts provided gains as well, especially for precision (+0.5%) when the two groups of features had similar importance. In addition, the combination of features representing numbers, symbols, and synonyms was evaluated but did not enhance results sufficiently to be considered helpful for abstracts. Finally, the use of abstracts in the classification was compared with the use of full texts. Results showed very small gains for specificity and accuracy (≈ 2%) and noticeable gains for precision (≈ 5%) when full texts were employed.

It is important to specify that the nature of a relevant entry may slightly differ according to reviewers' perspectives and chosen topics. Hence, generic training should be followed by adjustment processes based on users' preferences. For example, the proposed classifiers can be improved online when new examples are provided during their utilization. Active learning approaches, which are commonly used to rectify classifier behaviours for automated text classification,[48,49] can also be used. Further research is needed to evaluate the proposed models using a much larger collection, to compare our results with unsupervised machine learning, and to classify empirical records in accordance with the main study designs to facilitate syntheses (i.e., qualitative research, quantitative descriptive, nonrandomized studies, randomized trials, and mixed methods research).

ACKNOWLEDGEMENTS

We would like to thank Reem El Sherif, Genevieve Gore, and Vera Granikov for providing the data used by the Boolean mixed filter and for their valuable input. We would also like to thank Drs Isabelle Vedel and Marie-Pierre Gagnon for supplying additional records used in our collection. This study was supported by the Quebec SPOR SUPPORT Unit (http://unitesoutiensrapqc.ca/english/).

CONFLICT OF INTEREST

The author reported no conflict of interest.

REFERENCES

1. Pluye P, Hong QN, Bush P, Vedel I. Opening-up the definition of systematic literature review: the plurality of worldviews, methodologies and methods for reviews and syntheses. *J Clin Epidemiol*. 2016;73:2-5.
2. Ioannidis J. The mass production of redundant, misleading, and conflicted systematic reviews and meta-analyses. *Milbank Q*. 2016;94(3):485-514.
3. Pluye P, Hong QN. Combining the power of stories and the power of numbers: mixed methods research and mixed studies reviews. *Public Health*. 2014;35(1):29-45.
4. Heyvaert M, Hannes K, Onghena P. *Using Mixed Methods Research Synthesis for Literature Reviews: The Mixed Methods Research Synthesis Approach*. Los Angeles: SAGE Publications; 2016.
5. Souto RQ, Khanassov V, Hong QN, Bush P, Vedel I, Pluye P. Systematic mixed studies reviews: updating results on the reliability and efficiency of the mixed methods appraisal tool. *Int J Nurs Stud*. 2015;52(1):500-501.

6. Porta M, Greenland S, Hernán M, dos Santos Silva I, Last J. *A Dictionary of Epidemiology*. New York: Oxford University Press; 2014.
7. Abbott A. The causal devolution. *Sociol Methods Res*. 1998;27(2):148-181.
8. Björk BC, Roos A, Lauri M. Scientific journal publishing: yearly volume and open access availability. *Inf Res*. 2009;14(1):391.
9. Ganann R, Ciliska D, Thomas H. Expediting systematic reviews: methods and implications of rapid reviews. *Implement Sci*. 2010;5(1):56.
10. McKibbon KA, Wilczynski NL, Haynes RB. Developing optimal search strategies for retrieving qualitative studies in PsycINFO. *Eval Health Prof* . 2006;29(4):440-454.
11. Gill PJ, Roberts NW, Wang KY, Heneghan C. Development of a search filter for identifying studies completed in primary care. *Fam Pract*. 2014;31(6):739-745.
12. Lefebvre C, Manheimer E, Glanville J. Chapter 6: Searching for studies. *Cochrane Handbook for Systematic Reviews of Interventions*. Chichester (UK): John Wiley & Sons; 2008:95-150.
13. El Sherif R, Pluye P, Gore G, Granikov V, Hong QN. Performance of a mixed filter to identify relevant studies for mixed studies reviews. *J Med Libr Assoc*. 2016;104(1):47.
14. Sebastiani F. Machine learning in automated text categorization. *ACM Comput Surv*. 2002;34(1):1-47.
15. O'Mara-Eves A, Thomas J, McNaught J, Miwa M, Ananiadou S. Using text mining for study identification in systematic reviews: a systematic review of current approaches. *Syst Rev*. 2015;4(1):
16. Shemilt I, Simon A, Hollands GJ, et al. Pinpointing needles in giant haystacks: use of text mining to reduce impractical screening workload in extremely large scoping reviews. *Res Syn Meth*. 2014;5(1):31-49.
17. Howard BE, Phillips J, Miller K, et al. Swift-review: a text-mining workbench for systematic review. *Syst Rev*. 2016;5(1):1.
18. Yuanhan M, Kontonatsios G, Ananiadou S. Supporting systematic reviews using LDA-based document representations. *Syst Rev*. 2015;4(1):1.
19. Thomas J, O'Mara-Eves A, McNaught J, Ananiadou S. The potential of text mining to reduce screening workload in systematic reviews: a retrospective evaluation. Better Knowledge for Better Health. Abstracts of the 21st Cochrane Colloquium; 2013.
20. Gagnon MP, Nsangou ÉR, Payne-Gagnon J, Grenier S, Sicotte C. Barriers and facilitators to implementing electronic prescription: a systematic review of user groups' perceptions. *J Am Med Inform Assoc*. 2014;21(3):535-541.

21. Granikov V, El Sherif R, Pluye P. Patient information aid: promoting the right to know, evaluate, and share consumer health information found on the internet. *J Consum Health Internet*. 2015;19(3-4):233-240.
22. Jagosh J, Macaulay AC, Pluye P, et al. Uncovering the benefits of participatory research: implications of a realist review for health research and practice. *Milbank Q*. 2012;90(2):311-346.
23. Jagosh J, Pluye P, Macaulay AC, et al. Assessing the outcomes of participatory research: protocol for identifying, selecting, appraising and synthesizing the literature for realist review. *Implement Sci*. 2011;6(1):1.
24. Khanassov V, Vedel I, Pluye P. Barriers to implementation of case management for patients with dementia: a systematic mixed studies review. *Ann Fam Med*. 2014;12(5):456-465.
25. Khanassov V, Vedel I, Pluye P. Case management for dementia in primary health care: a systematic mixed studies review. *J Clin Interv Aging*. 2014;9:915-928.
26. Khanassov V, Vedel I, Pluye P. Dementia in canadian primary health care: the potential role of case management. *Health Sci Inquiry*. 2014;5(1):74-76.
27. Macaulay AC, Jagosh J, Seller R, et al. Assessing the benefits of participatory research: a rationale for a realist review. *Glob Health Promot*. 2011;18(2):45-48.
28. Porter MF. An algorithm for suffix stripping. *Program*. 1980;14(3):130-137.
29. Salton G, Buckley C. Term-weighting approaches in automatic text retrieval. *Inf Process Manag*. 1988;24(5):513-523.
30. Lan M, Tan CL, Low HB, Sung SY. A comprehensive comparative study on term weighting schemes for text categorization with support vector machines. In: *Special Interest Tracks and Posters of the 14th International Conference on World Wide Web*. New York: ACM Press; 2005:1032-1033.
31. Yang Y, Pedersen JO. A comparative study on feature selection in text categorization. *ICML*. 1997;97:412-420.
32. O Bodenreider. The unified medical language system what is it and how to use it? Tutorial at Medinfo; 2007.
33. Li YH, Jain AK. Classification of text documents. *Comput J*. 1998;41(8):537-546.
34. Yang Y, Liu X. A re-examination of text categorization methods. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; 1999; Berkeley, California USA:42-49.
35. Aha DW, Kibler D, Albert MK. Instance-based learning algorithms. *Mach Learn*. 1991;6(1):37-66.

36. John GH, Langley P. Estimating continuous distributions in Bayesian classifiers. In: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. San Francisco: Morgan Kaufmann Publishers Inc.; 1995:338-345.
37. Chang CC, Lin CJ. LIBSVM: a library for support vector machines. *ACM Trans Intell Syst Technol*. 2011;2:1-27. Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm
38. Mohan V. Decision trees: a comparison of various algorithms for building Decision Trees; 2013.
39. Quinlan JR. *C4. 5: Programs for Machine Learning*. San Fran- cisco: Elsevier; 2014.
40. Breiman L. Bagging predictors. *Mach Learn*. 1996;24(2):123-140.
41. Bauer E, Kohavi R. An empirical comparison of voting classification algorithms: bagging, boosting, and variants. *Mach Learn*. 1999;36(1-2):105-139.
42. Freund Y, Schapire RE. Experiments with a new boosting algorithm. *ICML*. 1996;96:148-156.
43. Zhai C, Lafferty J. A study of smoothing methods for language models applied to information retrieval. *ACM Trans Intell Syst Technol*. 2004;22(2):179-214.
44. Tao T, Wang X, Mei Q, Zhai C. Language model information retrieval with document expansion. In: *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics; 2006:407-414.
45. Shkapenyuk V, Suel T. Design and implementation of a high-performance distributed web crawler. Data engineering. *Proceedings. 18th International Conference on IEEE*. San Jose California: IEEE CS Press; 2002:357-368.
46. Kloda LA. Use Google Scholar, Scopus and Web of Science for comprehensive citation tracking. *Evid Based Libr Inf Pract*. 2007;2(3):87-90.
47. Tricco AC, Lillie E, Zarin W, et al. A scoping review on the conduct and reporting of scoping reviews. *BMC Med Res Methodol*. 2016;16(1):1.
48. Schohn G, Cohn D. Less is more: active learning with sup- port vector machines. In: ICML. Pittsburgh, Pennsylvania USA; 2000:839-846.
49. Tong S, Koller D. Support vector machine active learning with applications to text classification. *J Mach Learn Res*. 2001;2:45-66.

TABLES AND FIGURES

TABLE 1 Summary of the collection

Subcollection	Empirical	Nonempirical	Total
El Sherif et al[13]	2207	3309	5516
Khanassov et al[24-26]	459	214	673
Gagnon et al[20]	33	39	72
Jagosh et al[22,23] and Macaulay et al[27]	613	670	1283
Granikov et al[21]	306	200	506
Total	3618	4432	8050

TABLE 2 Algorithm comparison

Subcollection		Precision	Sensitivity	Specificity	Accuracy, %
Bagging-decision trees	(M1)	0.805	0.853	0.899	88.35
Boosting-decision trees	(M2)	0.776	0.852	0.879	87.01
SVM	(M3)	0.778	0.825	0.884	86.42
Decision trees	(M4)	0.763	0.789	0.878	84.81
kNN	(M5)	0.591	0.365	0.85	68.81
Naïve Bayes	(M6)	0.5	0.981	0.515	66.9
Boolean mixed filter	(Baseline)	0.604	0.895	0.545	56.9

TABLE 3 Performances of decision trees with bagging (M1) and SVM (M3) using 2000 concepts

Algorithm		Precision	Sensitivity	Specificity	Accuracy, %
Bagging-decision trees	(M1)	0.809	0.854	0.9	88.53
SVM	(M3)	0.788	0.827	0.89	86.92

TABLE 4 Performances of decision trees with bagging (M1) and SVM (M3) with feature combination

Type	Classifier	Precision	Sensitivity	Specificity	Accuracy, %
Numbers	M1	0.807	0.849	0.901	88.35 (-0.18)
	M3	0.776	0.833	0.882	86.56 (-0.36)
Symbols	M1	0.785	0.856	0.885	87.55 (-0.98)
	M3	0.788	0.828	0.89	86.94 (+0.02)
Synonyms	M1	0.811	0.834	0.905	88.12 (-0.41)
	M3	0.791	0.826	0.892	87.01 (+0.09)
All	M1	0.788	0.836	0.889	87.19 (-1.34)
	M3	0.777	0.831	0.882	86.51 (-0.41)

TABLE 5 Performances of the model based on interpolation

λ-value	Precision	Sensitivity	Specificity	Accuracy, %
No smoothing	0.809	0.854	0.9	88.53
0	0.803	0.848	0.898	88.14 (-0.39)
0.25	0.812	0.851	0.903	88.6 (+0.07)
0.5	0.813	0.851	0.904	88.66 (+0.13)
0.75	0.814	0.852	0.904	88.69 (+0.16)
1	0.805	0.858	0.898	88.35 (-0.18)

TABLE 6 Design indications in titles

Indication	Frequency	Most Likely Cateogry
Review	403	Nonempirical
Comment	359	Nonempirical
Analysis	265	Nonempirical
Controlled trial	214	Empirical
Systematic review	196	Nonempirical
Qualitative	132	Empirical
Cohort profile	119	Nonempirical
Erratum/corrigendum	95	Nonempirical
Response	87	Nonempirical
Cohort study	79	Empirical
Meta-analysis	76	Nonempirical
Case study	46	Empirical
Opinion/editorial	24	Nonempirical

TABLE 7 Performances of the model based on interpolation with title features

λ-value	**Precision**	**Sensitivity**	**Specificity**	**Accuracy, %**
No smoothing	0.809	0.854	0.9	88.53
0	0.803	0.848	0.898	88.14 (-0.39)
0.25	0.814	0.85	0.905	88.64 (+0.11)
0.5	0.817	0.85	0.906	88.76 (+0.23)
0.75	0.812	0.847	0.903	88.48 (-0.05)
1	0.808	0.853	0.901	88.48 (-0.05)

TABLE 8 Gain (%) provided by additional features for full texts compared with abstracts

Type	Precision	Sensitivity	Specificity	Accuracy, %
		Concepts		
Abstracts	+0.4	+0.1	+0.2	+0.3
Full texts	+0.3	+1.23	+0.1	+0.45
		Number combination		
Abstracts	-0.2	-0.5	+0.1	-0.2
Full texts	+0.6	+0	+0.3	+0.2
		Symbol combination		
Abstracts	-2.4	+0.2	-0.5	-0.98
Full texts	+0.6	+0.1	+0.3	+0.22
		Synonym combination		
Abstracts	+0.2	-2	+0.5	-0.41
Full texts	-0.9	-0.2	-0.6	-0.44
		All combination		
Abstracts	-2.1	-1.8	-1.1	1.34
Full texts	+0	-0.5	+0	-0.2

TABLE 9 Overall performances of the final model

Category	Precision	Sensitivity	Specificity	Accuracy, %
Abstracts (best λ = 0.75)				
Empirical	0.814	0.852	0.904	…
Nonempirical	0.925	0.904	0.852	…
Average	0.87	0.878	0.878	88.7
Full texts (best λ = 0.5)				
Empirical	0.863	0.854	0.933	…
Nonempirical	0.928	0.933	0.854	…
Average	0.896	0.894	0.894	90.71

FIGURE 1 Accuracy of decision trees with bagging (M1) and support vector machine (SVM) (M3) using concepts [Colour figure can be viewed at wileyonlinelibrary.com]

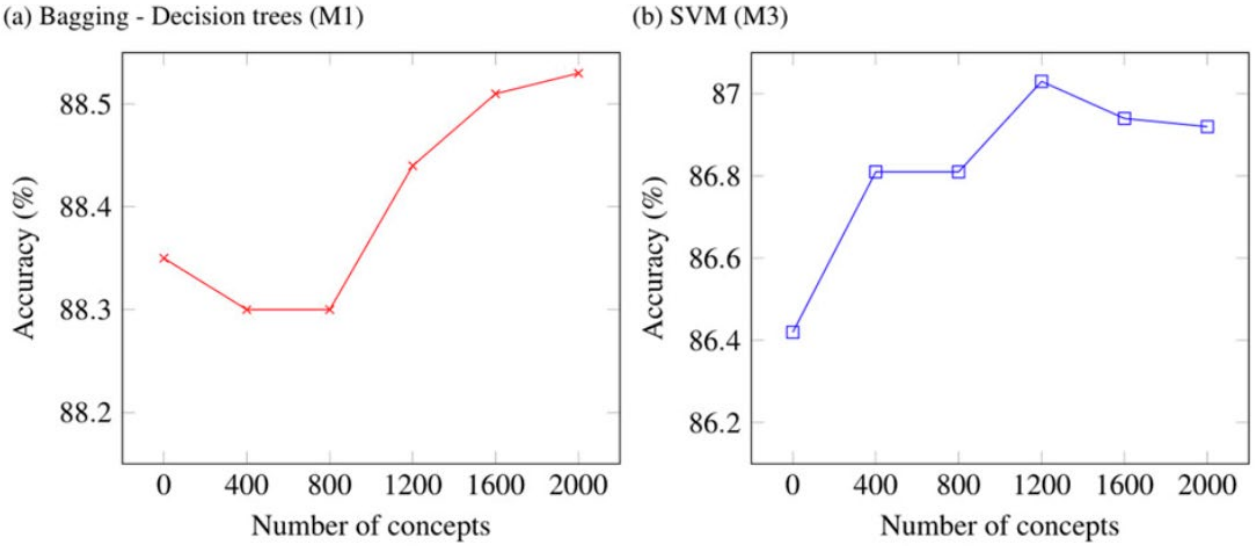

FIGURE 2 Twenty concepts selected by decision trees with bagging (M1)

"participatori research"	"research studi"	"et al."	"health inform"
"action research"	"evid base"	"medic record"	"inclus criteria"
"studi report"	"qualit studi"	"random assign"	"manag practic"
"systematic review"	"health care"	"practic guidelin"	"health outcom"
"studi particip"	"random control trial"	"side effect"	"associ studi"

FIGURE 3 Performances of decision trees with bagging (M1) mixing abstracts and full texts [Colour figure can be viewed at wileyonlinelibrary.com]