RESEARCH

An open-source integrated framework for the automation of citation collection and screening in systematic reviews.

Angelo D'Ambrosio*, Hajo Grundmann and Tjibbe Donker

*Correspondence:
angelo.d.ambrosio@uniklinik-freiburg.de,a.dambrosioMD@gmail.com
Institute for Infection Prevention and Hospital Hygiene, Freiburg University Hospital, Freiburg, Germany
Full list of author information is available at the end of the article

Abstract

Background: The exponential growth of scientific production makes secondary literature abridgements increasingly demanding. We introduce a new open-source framework for systematic reviews that significantly reduces time and workload for collecting and screening scientific literature.

Methods: The framework provides three main tools: 1) an automatic citation search engine and manager that collects records from multiple online sources with a unified query syntax, 2) a Bayesian, active machine learning, citation screening tool based on iterative human-machine interaction to increase predictive accuracy and, 3) a semi-automatic, data-driven query generator to create new search queries from existing citation data sets.

To evaluate the automatic screener's performance, we estimated the median posterior sensitivity and efficiency [90% Credible Intervals] using Bayesian simulation to predict the distribution of undetected potentially relevant records.

Results: Tested on an example topic, the framework collected 17,755 unique records through the citation manager; 766 records required human evaluation while the rest were excluded by the automatic classifier; the theoretical efficiency was 95.6% [95.3%, 95.7%] with a sensitivity of 100% [93.5%, 100%].

A new search query was generated from the labelled dataset, and 82,579 additional records were collected; only 567 records required human review after automatic screening, and six additional positive matches were found. The overall expected sensitivity decreased to 97.3% [73.8%, 100%] while the efficiency increased to 98.6% [98.2%, 98.7%].

Conclusions: The framework can significantly reduce the workload required to conduct large literature reviews by simplifying citation collection and screening while demonstrating exceptional sensitivity. Such a tool can improve the standardization and repeatability of systematic reviews.

Keywords: Systematic review automation; Citation management; Online data collection; Active machine learning; Natural language processing; Bayesian modeling

Background

Scientific production has experienced continuous exponential growth in the last decades [1, 2]. This is especially true for biomedical research, a trend further accelerated by the COVID-19 pandemic, thanks to faster article processing time by publishers and the greater use of preprint databases [3–5]. Consequently, it has become harder for researchers and practitioners to stay up to date on the latest

findings in their field. Secondary research is of paramount importance in this scenario in that it provides valuable summaries of the latest research results; however, it is becoming ever more challenging in terms of time and human resources required [6–9].

The article collection and screening phases of a systematic review are particularly demanding [10]. First, relevant published research must be collected from scientific databases using appropriately built search queries (retrieval phase); secondly, the scientific citations collected must be screened, selecting only those that are relevant to the topic (appraisal phase) [11–13].

Search queries construction is a complex task [14, 15], requiring both expertise in the scientific field of interest and some knowledge of the database query languages. The goal is to obtain a set of results that contains all relevant articles (high sensitivity) while keeping the total number of records low (high specificity), possibly focusing on the first at the expense of the second [15].

If an integrated search tool is not used, manual work is required to download, store and organise the publication data; this approach is complicated by limits to the number of records that can be downloaded at any one time and the need to harmonise different formats and resolve record duplication [16].

The citation screening phase is usually the more resource-demanding task in a systematic review: even with appropriately built search queries, the results may easily range in the tens of thousands, of which just a small fraction are actually relevant [14]. It has been estimated that labelling 10,000 publications can take up to 40 weeks of work and that the average clinical systematic review takes 63 weeks to complete [6, 7, 11]. A consequence of this is that systematic reviews are often already out-of-date by the time they are published [17].

The field of Data Science applied to evidence synthesis and acquisition has greatly maturated in the last years [12, 16, 18]. By applying natural language processing (NLP), it is possible to transform free text into quantitative features, with various levels of abstraction and generalisation [19, 20]; using machine learning, such text-derived data can be used to map and reproduce human judgment, automating the screening of citations [21].

Automation of systematic reviews has made significant improvements in the last years [22–25], and it is possible foreseeable that it will become the standard approach in the field [18], with many solutions already being implemented into commercial or free-to-use tools [see 16, table 1].

This manuscript introduces an open-source, production-ready framework that further contributes to the state-of-the-art in systematic review automation (SRA) and helpers (SRH) tools. We improve the "retrieval phase" by providing a unified framework for the automated collection and management of scientific literature from multiple online sources. For the citation screening (appraisal) phase, we built an active machine learning-based protocol [26, 27], which utilises a Bayesian framework to efficiently identify potentially relevant documents that require human review while automatically screening-out the vast majority of clearly non-relevant ones; the algorithm then requires human review to increase classification accuracy iteratively. Finally, we included a tool to generate new search queries based on an already categorised citation data set, to identify relevant research that manually-made queries may have possibly missed.

We tested the framework in the retrieval and appraisal phases of an example topic of interest to our group: the evaluation of the mathematical modelling of patient referral networks among hospitals and their impact on the diffusion of healthcare-associated pathogenic microorganisms; the protocol is published in [28].
In the Methods, we give an overview of the framework, in the Result, we show the outputs and performance of the framework applied to the example topic, and in the Discussion, we explain the methodological rationale for the different components and features of the framework.

Methods

General description

We built an R [29] based framework to simplify two aspects of systematic literature review: record acquisition and classification. The code used to generate the results is available at https://github.com/AD-Papers-Material/BART_SystReviewClassifier, while an updated and ready to use version of the framework is distributed as an R package at https://github.com/bakaburg1/BaySREn. The framework includes several modules that communicate through intermediate outputs stored in standard formats, making it possible for users to extend the framework or easily integrate it with other tools in their pipeline. See Supplemental Material S1 for an in-depth description of the framework and how to use it.
The tasks carried out by the framework are grouped into "sessions", which comprise obtaining scientific citation data (records) using a search query and then labelling them as relevant ("positive" in the rest of the text) or not ("negative") for the topic of interest with the help of a machine learning engine (Fig. 1). The initial search query should be built using domain knowledge, trying to achieve a high relevant/non-relevant record ratio.
The framework can then generate a new data-driven query from this labelled set to perform a new session to find records possibly missed by the first query.

Record acquisition and initial labelling

We built a set of tools to allow users to automatically search and download citation data from three major scientific databases ("sources"): Pubmed (https://pubmed.ncbi.nlm.nih.gov/), Web Of Science (WOS, https://apps.webofknowledge.com/) and the Institute of Electrical and Electronics Engineers (IEEE, https://ieeexplore.ieee.org/Xplore/home.jsp). The framework handles authorisation management for non-open databases like WOS and IEEE. It is also possible to import previously downloaded records in the framework; this is particularly useful for acquiring records from SCOPUS (https://www.scopus.com/) and EMBASE databases (https://www.embase.com/), for which a comprehensive API interface was not easy to build. An extra manual search was also necessary for Pubmed since the API and the web interface have different rule expansion algorithms and return slightly different results [30]. A short guide on how to set up the framework for each database supported is available in Supplemental Material S3.
The collected records are merged into a single database, resolving duplicates and

different formatting between sources. The records are ordered according to the frequency of the positive query terms (e.g., not preceded by a *NOT* modifier) in the title and abstract ("simple query ordering").
The researcher is then asked to label a subset of records to create the "initial training set" needed to start the automatic classification. We recommend manually labelling the first 250 records (see "hyperparameter optimisation" later). Simple query ordering increases the positivity rate in the initial training set [31], leading to higher sensitivity during automatic classification [32].

Text feature extraction

The framework models the relevance of a record based on the following fields in the citation data: title, abstract, authors, keywords, MeSH terms [33]. A range of Natural Language Processing (NLP) techniques [16, 19, 34] are employed to convert the textual information in these fields into features for machine learning through a bag-of-words approach [16]. Processing of free text fields (title, abstract) includes: tokenisation (i.e., extracting the terms), removal of common stopwords (i.e., sentence components having no semantic value), part-of-speech filtering (only nouns, adjectives, verbs and untagged terms are retained), and lemmatisation of terms (i.e., reduction to their base grammatical form). Text processing for authors, keywords and MeSH terms identifies logical units (e.g., authors' full names, composite keywords) and extracts them.
Terms appearing in less than 5% of the labelled documents are removed from negative records. All terms in the positive set are kept to increase sensitivity at the cost of specificity.
Some terms tend to co-appear in records (non-consecutive ngrams, nc-ngrams), often carrying a particular meaning when they do co-occur. To detect nc-ngrams, we generated a word network representation [35] with edges occurring between terms with a cosine similarity in terms of document co-occurrence > 0.5. We extracted the maximal cliques in the network [36] representing highly correlated groups of terms; these groups are added to the dataset as individual features. Only nc-ngrams comprising a maximum of ten terms are kept.
A second network is built using a co-occurrence threshold of 0.9. In this case, the cliques represent terms that always appear together and can therefore be considered redundant (i.e., they do not need to be considered separately). These terms are merged to increase computational efficiency and reduce overfitting.
The output is a Document-Term Matrix (DTM), with N_d rows representing the records (D_i), N_t terms column for the t_{field} terms (divided by record field) and $0, 1$ values whether $t_{field} \in D_i$. We also enriched the DTM with features referencing the number of terms in each field to help the model scale term importance based on the field length.

Label prediction

We used a Bayesian Additive Regression Trees (BART) machine learning "classification model" [37] [in the implementation of 38] to predict the probability of a record being relevant, given the information coded into the enriched DTM and the initial training set. We set up the BART model to use 2,000 MCMC iterations (after

250 burn-in iterations) and 50 trees; we used a k value of 2 to regularise extreme prediction and let the model use missing fields in the DTM as features [39]. Positive records are oversampled ten times to increase sensitivity [40].
The output is the expected value posterior predictive distribution for each record (PPD in brief) describing the probability of it being relevant (i.e., a positive match). An ensemble of ten models was fitted to improve prediction stability by averaging the PPD between models [41, 42].

To assign the labels, we employed an "active learning" approach [26, 27], where a human reviews a specific subset of predictions made by the machine, which is then retrained on the manually reviewed dataset. This process is carried out iteratively to reduce prediction uncertainty.
Label assignment is done through identification of an "uncertainty zone", the construction of which is possible thanks to the Bayesian nature of BART, which provides full PPDs instead of point-wise predictions for each record.
To describe the process formally, we define:

$$\pi_i = \frac{1}{M}\sum_{j=1}^{M} Pr(L_i = 1|DTM, m_j) \quad \text{(Eq. 1)}$$

as the PPD of a record D_i being relevant (i.e, having a positive label, $L_i = 1$), averaging the PPDs of the ensemble of $M = 10$ models m, and:

$$\begin{aligned} \pi_{i,l} &= \{\pi_i : Pr(\pi_i) = 1\%\} \\ \pi_{i,u} &= \{\pi_i : Pr(\pi_i) = 99\%\} \end{aligned} \quad \text{(Eq. 2)}$$

respectively as the lower and upper boundaries of the 98% quantile interval of π_i (98% Predictive Interval, 98% PrI).
Then we identify the "uncertainty zone" as:

$$U_\pi = [\max \vec{\pi}_u^-, \min \vec{\pi}_l^+] \quad \text{(Eq. 3)}$$

with $\vec{\pi}_u^-$ being the vector of $\pi_{i,u}$ with a negative label and $\vec{\pi}_l^+$ the vector of $\pi_{i,l}$ with a positive label. That is, U_π defines a range of values between the smallest $\pi_{i,l}$ in the set of already labelled positive records L_p and the largest $pi_{i,u}$ related to the negative ones L_n, noting that the two limits can appear in any order.
Consequently, a record D_i will be labelled as positive if:

$$\pi_{i,l} > \max_{\pi \in U_\pi} \pi \quad \text{(Eq. 4)}$$

that is, the record lower 98% PrI boundary should be higher than every value in the uncertainty zone. In other words, for a record to be labelled positive, its PPD

should be within the range of the mixture of PPD of the records previously labelled positive and should not cross the distributions of the negative records.
Conversely, a record is labelled as negative if:

$$\pi_{i,u} < \min_{\pi \in U_\pi} \pi \qquad \text{(Eq. 5)}$$

The remaining records are labelled as "uncertain".

Manual review is then necessary for: 1) uncertain records, 2) positive records (to avoid false positives), and 3) records whose predicted label differs from the existing manual one. The last case helps identify human errors or inconsistent labelling criteria.

The automatic classification and manual review steps alternate in a loop (CR iterations) until no new positive matches are found in four consecutive iterations.

Relevant term extraction

As a measure of feature importance, we computed the "inclusion rate", that is, the proportion of times a term is used in a posterior tree over the sum of total inclusions of all variables [38]. We extracted the terms, the portion of the citation data in which they were used, the average "inclusion rate" among the ensemble models (over 10,000 inclusions) and its ratio over the standard deviation of this inclusion (inclusion stability, IS). For each term, we ran a Poisson regression to get the linear association with a positive label and reported it as Relative Risk (RR) with the number of standard errors as significance index (Statistic); the comparison between the inclusion rate in the BART models and the linear association allows to spot relevant non-linear effects (i.e., the feature is relevant only in association with others). In the Results, we only listed the first 15 terms with IS > 1.5 (in order of inclusion rate), while the first fifty terms, regardless of inclusion stability, are listed in Supplemental Material S2.

New search query generation

We developed an algorithm that generates a new search query to find further relevant publications missed in the initial search, possibly at a reasonable cost to specificity (i.e., a higher number of negative results).
The algorithm is composed of the following steps:

- A partition tree [43] is built between the DTM and 800 samples from the PPD; if a term is present multiple times in the DTM (e.g., both in the title and abstract), it is counted just once, and field term count features are removed. This step generates a list of rules composed by AND/NOT "conditions" made of terms/authors/keywords/MeSH tokens, which together identify a group of records.
- For each rule, negative conditions (i.e., NOT statements) are added iteratively, starting from the most specific one, until no conditions are found that would not also remove positive records.

- The extended set of rules is sorted by positive-negative record difference in descending order. The cumulative number of unique positive records is computed and used to group the rules. Rules inside each group are ordered by specificity.
- The researcher is then asked to review the rule groups and select one or more rules from each group or edit overly specific rules (e.g., citing a non-relevant concept casually associated with a paper, like a numeric value or indicator). It is possible to exclude a group of rules altogether, especially those with the poorest sensitivity/specificity ratio.
- The selected rules are joined together by *OR* statements, defining a subset of records with a sensibly higher proportion of positive records than the original set
- Redundant (i.e., rules whose positive records are already included in more specific ones) and non-relevant rules (i.e., conditions that when removed do not impact sensitivity and specificity) are removed.
- Finally, the rules are re-elaborated in a query that can be used to perform a new citation search.

Because the algorithm is data-driven, it creates queries that effectively select positive records from the input dataset but may be not specific enough when applied to actual research databases. Therefore we added an extra subquery in _AND_that specifies the general topics of our search and narrows the search domain.
The new query was used to initiate a second search session.

Performance evaluation

We trained a simple Bayesian logistic regression (surrogate model) on the reviewed records to evaluate the consistency of the classification model (see Discussion for the theoretical justification). The surrogate model uses as predictor the lower boundary of the 98% PrI of the PPD of the records with weakly regularising, robust priors for the intercept (Student T with $\nu = 3, \mu = 0, \sigma = 2.5$) and the linear coefficient (Student T with $\nu = 3, \mu = 0, \sigma = 1.5$).
The quality of the model was evaluated through the Bayesian R^2 [44], of which we reported the posterior median and 90% Credible Interval [90% CrI]. The R^2 also provides an evaluation of the consistency of the original classification model. Given that this model is conditional only on the BART predictions and not on the DTM, it is characterised by more uncertainty, providing plausible worst-case scenarios.
The surrogate model is then used to generate the predictive cumulative distribution of the number of total positive records in the whole dataset. This distribution allows estimating the expected total posterior "Sensitivity" and "Efficiency" of the classification model in the whole (unreviewed) dataset. Efficiency is summarised by the "Work saved over random" (WSoR) statistic: one minus the ratio between the number of records manually reviewed and those that would be required to find the same number of positives if classification were performed choosing records randomly; this last quantity is estimated through a negative hypergeometric distribution [45] over the predicted number of positive records.
For the number of predicted positive records, sensitivity and efficiency, we reported the "truncated 90% PrI" [trunc. 90% PrI], i.e., the uncertainty interval bounded by

the number of observed total positive records (i.e., there cannot be fewer predicted positive records than observed).

Hyperparameter evaluation

Our classification algorithm has a limited number of hyperparameters:

- Size of the initial training set: 50, 100, 250, 500 records;
- Number of models in the ensemble: 1, 5, 10, 20, 40, 60 repetitions;
- Oversampling rate of positive records: 1x (i.e., no oversampling), 10x, 20x;
- PrI quantiles for building the uncertainty zone: 80%, 90%, 98%;
- Source of randomness between models in the ensemble: MCMC sampling only [46], MCMC plus data bootstrapping [47] of the training set.

To evaluate the hyperparameter effect of performance, we set up a "grid search" [48, 49] on a prelabelled "validation set" derived from the first 1,200 records of the first session dataset. Each hyperparameter combination was tested until four CR iterations were completed with no positive records or until the whole dataset was labelled.

For each combination, a performance score was computed as the product of "Efficiency" (1 minus the ratio of records that required reviewing over the total number of records) and "Sensitivity" (number of positive records found over the total number of positive records). We then used a partition tree [43] to identify homogeneous "performance clusters" of scores given hyperparameter values. For the rest of the study, we used the best hyperparameter set in terms of sensitivity followed by efficiency from the cluster with the highest average score.

Software and data

The framework is built with R v4.0.4 [29]. The R packages required by the framework are listed at https://github.com/bakaburg1/BaySREn/blob/main/DESCRIPTION.

All relevant data necessary to replicate the results is available at https://doi.org/10.5281/zenodo.6323360.

Results

First session

The initial search query for the example topic was:

((model OR models OR modeling OR network OR networks) AND (dissemination OR transmission OR spread OR diffusion) AND (nosocomial OR hospital OR "long-term-care" OR "long term care" OR "longterm care" OR "long-term care" OR "healthcare associated") AND (infection OR resistance OR resistant))

selecting only results between 2010 and 2020 (included). Results were collected from Pubmed, WOS, IEEE, EMBASE and SCOPUS, using the framework tools as described in the Methods and Supplemental Material S1.

The first search session returned a total of 27,600 records, specifically 12,719 (71.6% of the total) records from the EMBASE database, followed by 9,546 (53.8%) from Pubmed, 3,175 (17.9%) from SCOPUS, 2,100 (11.8%) from WOS, and 60 (0.34%) from IEEE (Table 1). There were various degrees of overlapping between

sources, with 38.4% of records being present in more than one database, and EMBASE and IEEE being the databases with the higher uniqueness ratios. The final data set was composed of 17,755 unique records.
The first 250 records (based on "simple query ordering") were categorised manually. Of these 43 (17.2%) were labeled as positive, and 207 (82.8%) as negative.

The categorised records were used to train the Bayesian classification model used to label the remaining records. After seven classification and review (CR) iterations (three resulting in new positive matches and four extra replications to account for stochastic variability), a total of 101 positives matches were found, requiring manual review of 766 records (13.2% positivity rate).
It is noticeable how the number of records that required manual review decreased rapidly between iterations (Table 2), indicating that the engine was converging while the uncertainties were resolved.
This phenomenon is better illustrated in Fig. 1 of Supplemental Material S2. It shows the mixture distribution of the PPDs of the records, specifically for records that were manually reviewed, before and after the classification step: it can be seen how the distribution of uncertain records shrinks (i.e., it becomes concentrated in a shorter probability range) and shifts toward the negative zone as more positive matches are found and reviewed.

We extracted the 15 more relevant terms for the classification model, described as: Term (citation part): Inclusion Rate (Inclusion Stability) [linear Relative Risk, Statistic]:

Patient Transport (Keyword): 61.2 (3.77) [99.1, 21.3], Transfer (Abstract): 57 (3.93) [22.5, 15.4], Network (Title): 56.5 (2.91) [18, 14.2], Network & Patient (Abstract): 54.2 (4.66) [26.3, 15.2], Donker T (Author): 53.5 (4.56) [159, 16.5], Worker (Abstract): 50 (3.33) [0.421, -1.21], Hospitals (Keyword): 49.8 (4.31) [27.8, 16.5], Movement (Abstract): 47.8 (2.7) [27.2, 15], Spread (Title): 46.6 (2.25) [16.2, 12.1], Facility (Abstract): 45 (2.22) [19.6, 14.8], Orange County (Keyword): 44.3 (3.19) [199, 17.2], Conduct (Abstract): 42.6 (3.7) [0.221, -2.57], Patient (Abstract): 42 (3.61) [27.6, 7.23], Perform (Abstract): 41.9 (2.38) [0.342, -2.55], Hospital (Title): 39 (1.95) [12.5, 12.5].

The "&" indicates nc-ngrams, i.e., terms strongly co-occurrent in the documents. The engine was able to pick up the central concept of the research topic, i.e., "patient transport" or "transfer" through a "network" of "facility"ies that facilitates the "spread" of infections, and even one of the authors of this study (Donker T.) as well as the region of interest ("Orange County") of another research group active on the topic of pathogen spreading over hospital networks. Some terms were considered highly relevant by the BART models (e.g., "Worker" in the sixth position out of more than 3800 terms considered), although in a simple linear model, their effect would hardly be significant (statistic: -1.21 s.e.); these are terms that are only relevant in conjunction with other terms but not on their own, highlighting the extra predictive power achieved through the use of advanced, non-linear machine learning.
A more extensive set of terms is presented in Table 1 of Supplemental Material S2.

Second session

The results of the first classification session were used to create a second, data-driven query to perform a more extensive search to find records that may have been missed during the first search session. The resulting query was as follows:

(((Donker T) NOT (bacterium isolate)) OR ((network patient) AND (resistant staphylococcus aureus) NOT (monte carlo) NOT isolation) OR (facility AND (network patient) AND regional NOT hospitals NOT increase NOT (patient transport) NOT (control infection use)) OR ((patient transport) NOT (Donker T) NOT worker) OR (hospitals AND (network patient) NOT (patient transport) NOT regional NOT clinical) OR (facility AND (network patient) NOT hospitals NOT (patient transport) NOT regional NOT prevention NOT medical) OR ((healthcare facility) NOT (Donker T) NOT worker NOT positive) OR (hospitals NOT (network patient) NOT medical NOT environmental NOT outcome NOT global) OR ((network patient) NOT facility NOT hospitals NOT (patient transport) NOT therapy NOT global)) AND ((antimicrobial resistance) OR (healthcare infection))

The final piece *AND ((antimicrobial resistance) OR (healthcare infection)* was added manually to define the search domain better since the algorithm was trained on documents that were all more or less related to these topics.
The generated query also provides a more nuanced understanding of the engine's internal classification logic, and this is helpful to spot possible biases in the model.

The search was done with the same year filter and procedures used in the first session.

The new search produced 107,294 records (Table 1), of which 48,396 (58.6%) from the EMBASE, followed by 28,811 (34.9%) from Pubmed, 17,070 (20.7%) from SCOPUS, 12,956 (15.7%) from WOS, and 61 (0.074%) from IEEE; compared with the first session, the relative weight of EMBASE and Pubmed decreased, while the level of content specificity greatly increased, as it was for SCOPUS. After removal of duplicates, 82,579 unique records were obtained. The newly collected records were joined with those from the first session and duplicates were removed. We obtained 98,371 unique records, with just 1,963 shared records between searches, which equates to 2% of the total. The percentage of records shared by two or more sources dropped to 22%.

Six CR rounds were necessary to complete the second session classification, with just 6 new positive found after reviewing 568 extra records. The first CR iteration required the user to review a substantial number of records (1,273); however, just labelling 275 of them (the suggested 250 plus 25 already labelled for the framework hyperparameter tuning) was sufficient to reduce this number to just 190 in the subsequent round. An evaluation of the convergence (Supplemental Material S2, Fig. 1) showed that, in addition to the dynamics already observed in session 1 (shrinkage and negative shift), a second mode appeared in the mixture distribution of the records to be reviewed, centred in a highly positive zone. The interpretation is that as the number of negative training records increases, the engine becomes

more and more sceptical and even asks to review some records labelled as positive in the initial training set generated during Session 1. This behaviour can rev spot classification errors and inconsistencies. Considering both sessions, 1,333 records were manually reviewed and 107 (8.03%) confirmed positive matches were found.

Again, the evaluation of the inclusion rate of the terms showed that the engine is quite capable of internalising the concepts behind the research topic. A subsample of the relevant terms used by the model in the second session is reported in Table 2 of Supplemental Material S2.

Hyperparameter selection

As described in the methods, hyperparameters were selected by evaluating sensibility and efficiency through a grid search on a validation set of 1,200 manually labelled records. The analysis suggested that the following parameter combination performed best: an initial training set of 250 categorised records with 10x oversampling of positive matches, ten models in the ensemble, no bootstrapping and an uncertainty zone defined by the 98% predictive interval. This combination of parameters was associated with a sensitivity of 98.8% (81 / 82 positive matches found) and an efficiency of 61.5% (462 / 1200 records evaluated). The detailed results of the hyperparameter tuning analysis are reported in Table 3 of Supplemental Material S2. Fig. 2 in Supplemental Material S2 demonstrates that the positive record oversampling rate, the number of ensemble models and the size of the initial training set were the parameters that mainly impacted performance.

Performance evaluation

To evaluate the theoretical performance of the engine, a surrogate Bayesian logistic regression model was trained on the manually reviewed labels using only the lower boundary of the record PPDs as predictor (see the Methods for details). The surrogate model showed the high predictive power of the scores produced by the classification model (Bayesian R2: 98.1% [97.4%, 98.3%] for session 1 and 98.2% [97.6%, 98.3%] for session 2).

Fig. 2 presents the actual and predicted (from the surrogate model) cumulative number of positive matches, ordered by the initial simple ordering query: the median of surrogate models' cumulative predictive distributions matches the actual number of positive records found quite well. It is striking how many more records would have required manual evaluation to find the same number of positive matches without a classification algorithm, with some positive matches found close to the end of the heuristically ordered list of records.

Table 3 shows various performance indexes for both sessions, both descriptive (Total records, Reviewed records, Observed positive matches) and estimated through the surrogate model (Expected efficiency, Predicted positive matches, Expected sensitivity, R^2).

In session 1 we observe an expected total number of positives of 101 [101, 108] for an estimated sensitivity of 100% [93.5%, 100%] and efficiency of 95.6% [95.3%, 95.7%]. In session 2 we observed a drop in expected sensitivity, especially in the lower credibility boundary (97.3% [72.8%, 100%]): as the number of records increases, even

a small probability of being a positive match can, in the worst-case scenario, lead to a relevant number of predicted positive matches (147 in this case). To ensure no obvious positive matches were missed, we evaluated 100 non-reviewed records with the highest median predicted probability and found no additional positive matches.

Discussion

We propose a new integrated framework to help researchers collect and screen scientific publications characterised by high performance and versatility. This framework joins the joining the growing field of systematic review automation (SRA) and helpers (SRH) tools [8, 22, 23, 50]. This framework implements standard approaches and uses ad-hoc solutions to common SRA issues. By freely sharing the tool as an open-source R package and by following a modular design, we sought to adopt some of the so-called Vienna Principles advocated by the International Collaboration for the Automation of Systematic Reviews (ICASR) [18].
The framework consists of four main components: 1) an integrated query-based citation search and management engine, 2) a Bayesian active machine learning-based citation classifier, and 3) a data-driven search query generation algorithm.

The search engine module used by the framework can automatically collect citation data from three well-known scientific databases (i.e., Pubmed, Web of Science, and the Institute of Electrical and Electronics Engineers) and process manually downloaded results from two more sources (SCOPUS, EMBASE). In comparison, most commercial or free SRH tools rely on internal databases (e.g., Mendeley https://www.mendeley.com/) sometimes focusing only on a particular topic [51] or a single external data source [52–54].
Mixing different databases is essential to obtain a more comprehensive view of the literature [55–57]: in our results, 18.7% of the positive matches were found in only one of the different data sources, and no positive record was present in all the sources (data not shown).
The framework online search algorithms are efficient enough to manage tens of thousands of search results, using various solutions to overcome the limitations of citation databases in terms of traffic and download quotas. The results are then automatically organised, deduplicated and arranged by "simple query ordering" in a uniform corpus. The preliminary ordering increases the positivity rate in the initial training set [31].

For the framework's record screening module, we developed an active machine learning protocol [26, 27] based on the best practices from other SRA studies, bringing further improvements at various levels.
The feature extractor module uses modern NLP techniques [19, 20] to transform free text into input data for machine learning. We did not include classical n-grams [58]; rather, we used network analysis to find non-consecutive, frequently associated terms, a generalisation of n-grams that relaxes the term adjacency assumption. This approach can also incorporate term connections across different parts of the records, e.g., terms having a different relevance when associated with a particular

author. The same technique was used with different parameters to merge redundant terms, increasing estimation efficiency and reducing noise.
The use of concurrency network-driven text modelling is not new [35, 59–61] and is a valuable tool to extract semantic information that is not evident in one-word or consecutive n-gram models.

The automatic classification algorithm is based on Bayesian Additive Regression Trees (BART) [37, 38]. Like other boosted trees algorithms [62], the BART method can explore complex non-linearities, perform variable selection, manage missing data while maintaining high predictive power.
However, the Bayesian foundation of the method provides further benefits: lower sensitivity to the choice of hyperparameters, natural regularisation through priors, and, most importantly, predictive distributions as output instead of point-wise predictions [63–65]. By selecting relatively tight prior distributions, we discouraged overly deep trees, long tree sequences, and extreme predicted probabilities, thus reducing the risk of overfitting.
The algorithm runs multiple replications of the model and averages their predictive distributions creating an "ensemble"; this technique has been shown to improve out-of-sample predictive performance [41, 42], as confirmed during the hyperparameter evaluation (Supplemental Material S2). Ensembling reduces the uncertainty in the predictive distribution tails related to the randomness in the MCMC fit [46], generating a shift in the probability mass towards the distribution centre and stabilising it (i.e., reducing variance without impacting bias). On the other hand, simply imposing more robust uninformative priors against extreme predictions would have reduced variance but also shifted the distribution towards a non-decision zone, increasing bias [66].
Since the number of model replications has a significant impact on computation times, we decided to use ten replicas, the lower value after which performance stabilised, as resulted from the evaluation of the hyperparameters (Supplemental Material S2, Fig. 2).
We also investigated whether bootstrapping between replications [47] would improve performance; however, contrary to theory [67], it appeared to be slightly detrimental in our case (Supplemental Material S2, Fig. 2) compared to simple ensembling.

A low proportion of relevant matches (class imbalance) is typical for literature reviews [23, 68, 69], and a strong imbalance between positive and negative records can affect sensitivity [32, 70].
To overcome this problem, we oversampled [40] the positive records ten times before model fitting. The hyperparameter analysis showed that the oversampling rate, together with model ensembling, was the parameter with the most significant impact on performance.
A known risk with positive oversampling is the misclassification of negative records [71]. However, since all predicted positives in our approach are reviewed manually, we are always guaranteed to achieve 100% specificity/positive predictive value: the only price for the increased sensitivity due to oversampling is a larger number of

records to be reviewed.
An alternative to oversampling would be to apply different weights and/or costs to the classes [67, 72], but the BART implementation we used did not have this feature; furthermore, using simple oversampling allows for a broader compatibility with different modelling engines [73, 74].
Finally, sorting the records by query term frequency (simple query ordering) produces a much higher rate of relevant records in the initial training set (17.2%) compared to the overall data (0.11%), which boosts the sensitivity of the model.

One of the key innovations we have introduced is the concept of "uncertainty zone", the implementation of which is possible thanks to the Bayesian foundation of the classification model.
This construct guides the selection of records to be manually reviewed and gets dynamically updated and reduced after each CR iteration, as more uncertain predictions are evaluated (Supplemental Material S2 Fig. 1).
The use of a dynamic uncertainty zone overcomes the usual requirement of dataset-specific hard thresholds in active machine learning and allows to review multiple items at once between iterations [27, 75, 76]. The hyperparameters required by our algorithm are general and non-task-specific, like the PPD intervals underlying the uncertainty zone and the maximum number of iterations without positive matches after which a session is concluded; the evaluation of the classification model hyperparameters shows that the algorithm is robust against variations in these parameters, and we expect the default values to perform well on most datasets.
Since researchers are asked to review both records predicted as surely relevant and those inside the uncertainty zone, this method can be considered as a unifying synthesis of the "certainty" and "uncertainty" paradigms of active learning [27].

We assessed performance as the ability of the screening procedure (automatic classification plus manual review) to find the largest number of relevant records while requiring manual reviewing for as few of them as possible (i.e., sensitivity $\times$ efficiency).
We avoided the classical out-of-sample approaches such as train-test sampling, out-of-bag bootstrapping or cross-validation [77, 78]. Such methods primarily assume that the rate of positivity is the same on average in every possible random subset of the data [79]; this uniformity is broken by how the initial training set and the subsequent reviewed records are selected by the query-based ordering and active learning algorithm, resulting in a lower positivity rate in the unlabelled records (Fig. 2). Moreover, a literature corpus is unique per search query/database combination, and therefore any out-of-sample performance estimate is not replicable since no new data can be acquired related to the current corpus.
To estimate overall sensitivity, we instead applied simple Bayesian regression (surrogate model) to the manually reviewed data to abstract the classification model predictions and generate a maximum entropy [80] estimate of the number of missed positive matches among the unreviewed records in the whole dataset. This simple surrogate model fitted the data very well (R^2 consistently above 97%) using only the lower 98% PrI boundary of the PPDs as predictor, indicating predictive

consistency in the classification model. The posterior predictive distribution of the surrogate model could be used to explore worse case scenarios in terms of sensitivity.

Our framework achieves very high sensitivity by screening only a very small fraction of all records, bringing a meaningful reduction in workload.
Based on the surrogate model, we predicted a predicted median sensitivity of 100% [93.5%, 100%] in the first session (screening 4.29% of records) and of 97.3% [73.8%, 100%] in the second (screening 1.34% of records): efficiency increased significantly in the second session as only a few new positive matches were found; however, given the large number of records, uncertainty about sensitivity increased, as expected.
Both results are above the usual performance in this field [23] and are in line with the average sensitivity of 92% estimated after human-only screening [81]. In one interesting case, the model detected a human-caused misclassification error, demonstrating its robustness and value as a second screener, a role already suggested for SRA tools in previous studies [82–84]. Although "simple query ordering" concentrated most relevant matches in the first 20-25 thousand records, without the tool support, the remaining relevant records would have been missed without manually screening almost the entire dataset.

The model required ˜5-20 minutes per iteration to perform the predictions in session 1 (17,755 documents) and 20-40 minutes in session 2 (98,371 documents) on an eight-core, 2.5 GHz, 16 GB RAM, 2014 laptop; including manual record review, one session required 1-3 days of work, for a total of 1-2 weeks for the whole process (including record collection). This is a considerable time saving compared to the several months typically required for the screening phase of systematic reviews [6, 7, 11]. To our knowledge, the amount of data processed (˜100,000 records) was larger than what is typical of most SRA studies [23, 85], highlighting the scalability of the tool in real-world scenarios.

The last module of our framework is an algorithm for data-driven search query generation. Generating an efficient and effective search query is a complex task [14, 15]; it requires building a combination of positive and negative terms to maximise the number of relevant search results while minimising the total number of records to be reviewed. Our solution combines a sensitivity-driven subquery proposal engine based on concurrent decision trees [86, 87] built on the BART ensemble PPD, with a human review step and an efficiency-driven query builder. The aim is to generate a new search query to help find records missed in the first search session. The generated query did indeed retrieve a few more relevant records not found in session 1 but at the cost of significantly increasing the number of documents.
An interesting aspect of this feature is that it provides a human-readable overview of the classification rules learned by the classification model, showing which combination of terms was particularly relevant and even spotting authors and geographical locations associated with the study topic. The generated query, therefore, served also as a means for machine learning explainability [88, 89], useful for understanding and detecting biases in black-box classification algorithms [90]; explainability is often required or even legally mandatory for high-stake machine learning applications [91, 92].

It is important to note that this process is entirely data-driven. The algorithm is only aware of the "world" defined by the dataset used as input, which is generated by a specific search query focused on a particular topic. Therefore, the new query may not be specific enough when applied to an unbounded search domain and may return an unmanageable amount of irrelevant results. The solution we found was to add another component to the query, specifying the general topic (antimicrobial resistance and healthcare-associated infections) of our research.

As mentioned early, our framework builds on modularity. We have designed so that each module can become fully independent in future iterations; it will be possible for users to add custom features such as citation search and parsing for other scientific databases, alternative text processing algorithms or machine learning modules. We consider such interoperability to be extremely relevant: the main strength of our tool lies in the composition of many independent solutions, such as the idea of Bayesian active machine learning and the exploit of the derived uncertainty in defining the records needing human review.
Each component could benefit considerably from the recent improvements in text mining and machine learning.
For example, the text processing approach based on the "boolean bag-of-words" paradigm is quite simple and could be improved by more nuanced text representations. It might be considered whether feature transformations such as TF-IDF [19, 34] could be advantageous, although we hypothesise that tree-based classification algorithms like BART are robust enough not to require such operations. Instead, it might be worth exploring the application of word embedding: this technique transforms terms into semantic vectors derived from the surrounding text [93–95] and could be used to reduce noise by merging different terms that are semantically similar or enhance signal by distinguishing identical terms with different meaning given the context. Another option would be to employ unsupervised learning models like Latent Dirichlet Analysis and Latent Semantic Analysis, [96–98] or graph-of-word techniques [35, 61] to extract topics that expand the feature space.
Our classification algorithm is applicable with any Bayesian supervised machine learning method that provides full PPDs; therefore, alternative classification models, such as Gaussian Processes, known for their flexibility [99, 100], could be evaluated. It would be even more interesting to test advanced learning algorithms that go beyond the bag-of-words approach and take into consideration higher-level features in the text such as term context and sequences, long-distance term relationships, semantic structures, etc., [95, 101–105], provided that a Bayesian implementation of such algorithms is available (for example **(author?)** [106]).
Finally, a natural improvement would be to provide a graphical user interface to make the framework easy to use also for less technical users.

The field of literature review automation is evolving rapidly, and we anticipate an increasing use of such technologies to address the accelerating pace of scientific production. We believe it is encouraging that a wide variety of tools are being made available to let researchers and policymakers find the approach that best fits their needs.
We contribute to this field with an innovative framework that provides excellent

performance and easy integration with existing systematic review pipelines. The value of this work lies not only in the framework itself, which we make available as open-source software, but also in the set of methodologies we developed to solve various SRA issues and which can also be used to improve existing solutions.

Acknowledgements
We would like to thank Deborah Lawrie-Blum and Fabian Bürkin respectively for English language proofreading and mathematical formalization proof check.

Abbreviations
API: Application Programming Interface; BART: Bayesian Additive Regression Trees; COVID-19: Coronavirus Disease 2019; CR: Classification & Review; DTM: Document-Term Matrix; ICASR: International Collaboration for the Automation of Systematic Reviews; IEEE: Institute of Electrical and Electronics Engineers; IS: Inclusion Stability; MCMC: Monte Carlo Markov Chain; MeSH: Medical Subject Headings; MCMC: Monte Carlo Markov Chain; NLP: Natural Language Processing; PPD: Expected Value Posterior Predictive Distribution; RR: Relative Risk; SRA: Systematic Review Automation; SRH: Systematic Review Helpers; TF-IDF: Term Frequency - Inverse Document Frequency; WOS: Web Of Science; WSoR: Work Saved over Random.

Declarations

Funding
This project was developed under the Joint Programming Initiative on Antimicrobial Resistance (JPIAMR) through the 7th call, project number 01KI1831 funded by BMBF and administrated by DLR Project Management Agency.

Availability of data and materials
The code and the instructions necessary to reproduce the results are available at https://github.com/AD-Papers-Material/BART_SystReviewClassifier. An updated, ready-to-use version of the framework is available at https://github.com/bakaburg1/BaySREn. All relevant data necessary to replicate the results is available at: https://doi.org/10.5281/zenodo.6323360.

Ethics approval and consent to participate
Not applicable.

Competing interests
The authors declare no competing interests.

Consent for publication
Not applicable.

Authors' contributions
Conceptualization: AD, TD; Data curation: AD; Formal analysis & Methodology: AD; Project administration: AD, TD; Software development: AD; Supervision: TD, HG; Writing–original draft: AD, TD; Writing–review & editing: AD, TD, HG. All authors have read and approved the manuscript.

Author details
Institute for Infection Prevention and Hospital Hygiene, Freiburg University Hospital, Freiburg, Germany.

References

1. Larsen, P., Von Ins, M.: The rate of growth in scientific publication and the decline in coverage provided by science citation index. Scientometrics **84**(3), 575–603 (2010)
2. Bornmann, L., Mutz, R.: Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. Journal of the Association for Information Science and Technology **66**(11), 2215–2222 (2015)
3. Aviv-Reuven, S., Rosenfeld, A.: Publication patterns' changes due to the covid-19 pandemic: A longitudinal and short-term scientometric analysis. Scientometrics, 1–24 (2021)
4. Horbach, S.P.: Pandemic publishing: Medical journals strongly speed up their publication process for covid-19. Quantitative Science Studies **1**(3), 1056–1067 (2020)
5. Hoy, M.B.: Rise of the rxivs: How preprint servers are changing the publishing process. Medical reference services quarterly **39**(1), 84–89 (2020)
6. Allen, I.E., Olkin, I.: Estimating time to conduct a meta-analysis from number of citations retrieved. Jama **282**(7), 634–635 (1999)
7. Borah, R., Brown, A.W., Capers, P.L., Kaiser, K.A.: Analysis of the time and workers needed to conduct systematic reviews of medical interventions using data from the prospero registry. BMJ open **7**(2), 012545 (2017)
8. Cohen, A.M., Adams, C.E., Davis, J.M., Yu, C., Yu, P.S., Meng, W., Duggan, L., McDonagh, M., Smalheiser, N.R.: Evidence-based medicine, the essential role of systematic reviews, and the need for automated text mining tools. In: Proceedings of the 1st ACM International Health Informatics Symposium, pp. 376–380 (2010)

9. Bastian, H., Glasziou, P., Chalmers, I.: Seventy-five trials and eleven systematic reviews a day: how will we ever keep up? PLoS medicine **7**(9), 1000326 (2010)
10. Babar, M.A., Zhang, H.: Systematic literature reviews in software engineering: Preliminary results from interviews with researchers. In: 2009 3rd International Symposium on Empirical Software Engineering and Measurement, pp. 346–355 (2009). IEEE
11. Bannach-Brown, A., Przybyła, P., Thomas, J., Rice, A.S., Ananiadou, S., Liao, J., Macleod, M.R.: Machine learning algorithms for systematic review: reducing workload in a preclinical review of animal studies and reducing human screening error. Systematic reviews **8**(1), 1–12 (2019)
12. Tsafnat, G., Glasziou, P., Choong, M.K., Dunn, A., Galgani, F., Coiera, E.: Systematic review automation technologies. Systematic reviews **3**(1), 1–15 (2014)
13. Higgins, J.P., Thomas, J., Chandler, J., Cumpston, M., Li, T., Page, M.J., Welch, V.A.: Cochrane Handbook for Systematic Reviews of Interventions. John Wiley & Sons, ??? (2019)
14. Lefebvre, C., Manheimer, E., Glanville, J., Higgins, J., Green, S.: Searching for studies (chapter 6). Cochrane handbook for systematic reviews of interventions version **510** (2011)
15. Hammerstrøm, K., Wade, A., Jørgensen, A.-M.K., Hammerstrøm, K.: Searching for studies. Education **54**(11.3) (2010)
16. Marshall, I.J., Wallace, B.C.: Toward systematic review automation: a practical guide to using machine learning tools in research synthesis, vol. 8, pp. 1–10. Springer, ??? (2019)
17. Beller, E.M., Chen, J.K.-H., Wang, U.L.-H., Glasziou, P.P.: Are systematic reviews up-to-date at the time of publication? Systematic reviews **2**(1), 1–6 (2013)
18. Beller, E., Clark, J., Tsafnat, G., Adams, C., Diehl, H., Lund, H., Ouzzani, M., Thayer, K., Thomas, J., Turner, T., *et al.*: Making progress with the automation of systematic reviews: principles of the international collaboration for the automation of systematic reviews (icasr). Systematic reviews **7**(1), 1–7 (2018)
19. Ananiadou, S., McNaught, J.: Text Mining for Biology and Biomedicine. Citeseer, ??? (2006)
20. Cohen, K.B., Hunter, L.: Getting started in text mining. PLoS computational biology **4**(1), 20 (2008)
21. Ikonomakis, M., Kotsiantis, S., Tampakas, V.: Text classification using machine learning techniques. WSEAS transactions on computers **4**(8), 966–974 (2005)
22. Ananiadou, S., Rea, B., Okazaki, N., Procter, R., Thomas, J.: Supporting systematic reviews using text mining. Social Science Computer Review **27**(4), 509–523 (2009)
23. 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. Systematic reviews **4**(1), 1–22 (2015)
24. Tsafnat, G., Dunn, A., Glasziou, P., Coiera, E.: The automation of systematic reviews. British Medical Journal Publishing Group (2013)
25. Jonnalagadda, S.R., Goyal, P., Huffman, M.D.: Automating data extraction in systematic reviews: a systematic review. Systematic reviews **4**(1), 1–16 (2015)
26. Settles, B.: Active learning literature survey (2009)
27. Miwa, M., Thomas, J., O'Mara-Eves, A., Ananiadou, S.: Reducing systematic review workload through certainty-based screening. Journal of biomedical informatics **51**, 242–253 (2014)
28. Sadaghiani, C., Donker, T., Andrianou, X., Babarczy, B., De Boer, G., Di Ruscio, F., Cairns, S., Crépey, P., Fortaleza, C.M., Freyler, P., Gagliotti, C., Liljeros, F., Moro, M.L., Oteo, J., Palos, C., Pilarski, G., Reilly, J., Robotham, J., Sá-Leão, R., Simonsen, G.S., Temime, L., Ternhag, A., Grundmann, H., Mutters, N.T.: National Health Care Infrastructures, Health Care Utilization and Patient Movements Between Hospitals - Networks Working to Improve Surveillance: a Systematic Literature Review. (2020). http://www.crd.york.ac.uk/PROSPERO/display_record.asp?ID=CRD42020157987
29. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2021). R Foundation for Statistical Computing. https://www.R-project.org/
30. NCBI Insights : Updated pubmed E-utilities coming in April 2022! U.S. National Library of Medicine. Accessed on 21.10.2021. https://ncbiinsights.ncbi.nlm.nih.gov/2021/10/05/updated-pubmed-api/
31. Wallace, B.C., Small, K., Brodley, C.E., Trikalinos, T.A.: Active learning for biomedical citation screening. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 173–182 (2010)
32. Chawla, N.V., Japkowicz, N., Kotcz, A.: Special issue on learning from imbalanced data sets. ACM SIGKDD explorations newsletter **6**(1), 1–6 (2004)
33. Lipscomb, C.E.: Medical subject headings (mesh). Bulletin of the Medical Library Association **88**(3), 265 (2000)
34. Baeza-Yates, R., Ribeiro-Neto, B., *et al.*: Modern Information Retrieval vol. 463. ACM press New York, ??? (1999)
35. Rousseau, F.: Graph-of-words: mining and retrieving text with networks of features. PhD thesis, Ph. D. dissertation (2015)
36. Eppstein, D., Löffler, M., Strash, D.: Listing all maximal cliques in sparse graphs in near-optimal time. In: International Symposium on Algorithms and Computation, pp. 403–414 (2010). Springer
37. Chipman, H.A., George, E.I., McCulloch, R.E., *et al.*: Bart: Bayesian additive regression trees. The Annals of Applied Statistics **4**(1), 266–298 (2010)
38. Kapelner, A., Bleich, J.: bartmachine: Machine learning with bayesian additive regression trees. arXiv preprint arXiv:1312.2171 (2013)
39. Kapelner, A., Bleich, J.: Prediction with missing data via bayesian additive regression trees. Canadian Journal of Statistics **43**(2), 224–239 (2015)
40. Batista, G.E., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD explorations newsletter **6**(1), 20–29 (2004)
41. Zhou, Z.-H.: Ensemble learning. In: Machine Learning, pp. 181–210. Springer, ??? (2021)
42. Dietterich, T.G.: Ensemble methods in machine learning. In: International Workshop on Multiple Classifier

Systems, pp. 1–15 (2000). Springer
43. Therneau, T., Atkinson, B.: Rpart: Recursive Partitioning and Regression Trees. (2019). R package version 4.1-15. https://CRAN.R-project.org/package=rpart
44. Gelman, A., Goodrich, B., Gabry, J., Vehtari, A.: R-squared for bayesian regression models. The American Statistician (2019)
45. Chae, K.-C.: Presenting the negative hypergeometric distribution to the introductory statistics courses. International Journal of Mathematical Education in Science and Technology **24**(4), 523–526 (1993)
46. Robert, C.P., Casella, G.: Monte Carlo Statistical Methods vol. 2. Springer, ??? (2004)
47. Breiman, L.: Bagging predictors. Machine learning **24**(2), 123–140 (1996)
48. Claesen, M., De Moor, B.: Hyperparameter search in machine learning. arXiv preprint arXiv:1502.02127 (2015)
49. Yang, L., Shami, A.: On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing **415**, 295–316 (2020)
50. Cohen, A.M., Hersh, W.R., Peterson, K., Yen, P.-Y.: Reducing workload in systematic review preparation using automated citation classification. Journal of the American Medical Informatics Association **13**(2), 206–219 (2006)
51. Visser, E.: Performing systematic literature reviews with researchr: Tool demonstration. Technical Report Series TUD-SERG-2010-010 (2010)
52. Thomas, J., Brunton, J.: Eppi-reviewer: software for research synthesis (2007)
53. Poulter, G.L., Rubin, D.L., Altman, R.B., Seoighe, C.: Mscanner: a classifier for retrieving medline citations. BMC bioinformatics **9**(1), 1–12 (2008)
54. Soto, A.J., Przybyła, P., Ananiadou, S.: Thalia: semantic search engine for biomedical abstracts. Bioinformatics **35**(10), 1799–1801 (2019)
55. Bajpai, A., Davuluri, S., Haridas, H., Kasliwal, G., Deepti, H., Sreelakshmi, K., Chandrashekar, D., Bora, P., Farouk, M., Chitturi, N., et al.: In search of the right literature search engine (s). Nature Precedings, 1–1 (2011)
56. Wilkins, T., Gillies, R.A., Davies, K.: Embase versus medline for family medicine searches: can medline searches find the forest or a tree? Canadian Family Physician **51**(6), 848–849 (2005)
57. Woods, D., Trewheellar, K.: Medline and embase complement each other in literature searches. BMJ: British Medical Journal **316**(7138), 1166 (1998)
58. Schonlau, M., Guenther, N.: Text mining using n-grams. Schonlau, M., Guenther, N. Sucholutsky, I. Text mining using n-gram variables. The Stata Journal **17**(4), 866–881 (2017)
59. Violos, J., Tserpes, K., Psomakelis, E., Psychas, K., Varvarigou, T.: Sentiment analysis using word-graphs. In: Proceedings of the 6th International Conference on Web Intelligence, Mining and Semantics, pp. 1–9 (2016)
60. Rousseau, F., Kiagias, E., Vazirgiannis, M.: Text categorization as a graph classification problem. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 1702–1712 (2015)
61. Ohsawa, Y., Benson, N.E., Yachida, M.: Keygraph: Automatic indexing by co-occurrence graph based on building construction metaphor. In: Proceedings IEEE International Forum on Research and Technology Advances in Digital Libraries-ADL'98-, pp. 12–18 (1998). IEEE
62. Hastie, T., Tibshirani, R., Friedman, J.: Boosting and additive trees. In: The Elements of Statistical Learning, pp. 337–387. Springer, ??? (2009)
63. Soria-Olivas, E., Gomez-Sanchis, J., Martin, J.D., Vila-Frances, J., Martinez, M., Magdalena, J.R., Serrano, A.J.: Belm: Bayesian extreme learning machine. IEEE transactions on neural networks **22**(3), 505–509 (2011)
64. Joo, T., Chung, U., Seo, M.-G.: Being bayesian about categorical probability. In: International Conference on Machine Learning, pp. 4950–4961 (2020). PMLR
65. Jospin, L.V., Buntine, W., Boussaid, F., Laga, H., Bennamoun, M.: Hands-on bayesian neural networks–a tutorial for deep learning users. arXiv preprint arXiv:2007.06823 (2020)
66. Hansen, L.K., *et al.*: Bayesian averaging is well-temperated. In: Proceedings of NIPS, vol. 99, pp. 265–271 (2000)
67. Díez-Pastor, J.F., Rodríguez, J.J., García-Osorio, C.I., Kuncheva, L.I.: Diversity techniques improve the performance of the best imbalance learning ensembles. Information Sciences **325**, 98–117 (2015)
68. Sampson, M., Tetzlaff, J., Urquhart, C.: Precision of healthcare systematic review searches in a cross-sectional sample. Research Synthesis Methods **2**(2), 119–125 (2011)
69. Wallace, B.C., Trikalinos, T.A., Lau, J., Brodley, C., Schmid, C.H.: Semi-automated screening of biomedical citations for systematic reviews. BMC bioinformatics **11**(1), 1–11 (2010)
70. Khoshgoftaar, T.M., Van Hulse, J., Napolitano, A.: Comparing boosting and bagging techniques with noisy and imbalanced data. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans **41**(3), 552–568 (2010)
71. Ramezankhani, A., Pournik, O., Shahrabi, J., Azizi, F., Hadaegh, F., Khalili, D.: The impact of oversampling with smote on the performance of 3 classifiers in prediction of type 2 diabetes. Medical decision making **36**(1), 137–144 (2016)
72. Abd Elrahman, S.M., Abraham, A.: A review of class imbalance problem. Journal of Network and Innovative Computing **1**(2013), 332–340 (2013)
73. Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., Herrera, F.: A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **42**(4), 463–484 (2011)
74. Roshan, S.E., Asadi, S.: Improvement of bagging performance for classification of imbalanced datasets using evolutionary multi-objective optimization. Engineering Applications of Artificial Intelligence **87**, 103319 (2020)
75. Laws, F., Schütze, H.: Stopping criteria for active learning of named entity recognition. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pp. 465–472 (2008)

76. Zhu, J., Wang, H., Hovy, E., Ma, M.: Confidence-based stopping criteria for active learning for data annotation. ACM Transactions on Speech and Language Processing (TSLP) **6**(3), 1–24 (2010)
77. Kohavi, R., *et al.*: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Ijcai, vol. 14, pp. 1137–1145 (1995). Montreal, Canada
78. James, G., Witten, D., Hastie, T., Tibshirani, R.: An Introduction to Statistical Learning vol. 112. Springer, ??? (2013)
79. Tashman, L.J.: Out-of-sample tests of forecasting accuracy: an analysis and review. International journal of forecasting **16**(4), 437–450 (2000)
80. Harremoës, P., Topsøe, F.: Maximum entropy fundamentals. Entropy **3**(3), 191–226 (2001)
81. Edwards, P., Clarke, M., DiGuiseppi, C., Pratap, S., Roberts, I., Wentz, R.: Identification of randomized controlled trials in systematic reviews: accuracy and reliability of screening records. Statistics in medicine **21**(11), 1635–1640 (2002)
82. Frunza, O., Inkpen, D., Matwin, S.: Building systematic reviews using automatic text classification techniques. In: Coling 2010: Posters, pp. 303–311 (2010)
83. Bekhuis, T., Demner-Fushman, D.: Screening nonrandomized studies for medical systematic reviews: a comparative study of classifiers. Artificial intelligence in medicine **55**(3), 197–207 (2012)
84. Bekhuis, T., Demner-Fushman, D.: Towards automating the initial screening phase of a systematic review. MEDINFO 2010, 146–150 (2010)
85. Olorisade, B.K., de Quincey, E., Brereton, P., Andras, P.: A critical analysis of studies that address the use of text mining for citation screening in systematic reviews. In: Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering, pp. 1–11 (2016)
86. Blanco-Justicia, A., Domingo-Ferrer, J.: Machine learning explainability through comprehensible decision trees. In: International Cross-Domain Conference for Machine Learning and Knowledge Extraction, pp. 15–26 (2019). Springer
87. Moore, A., Murdock, V., Cai, Y., Jones, K.: Transparent tree ensembles. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 1241–1244 (2018)
88. Bhatt, U., Andrus, M., Weller, A., Xiang, A.: Machine learning explainability for external stakeholders. arXiv preprint arXiv:2007.05408 (2020)
89. Burkart, N., Huber, M.F.: A survey on the explainability of supervised machine learning. Journal of Artificial Intelligence Research **70**, 245–317 (2021)
90. Malhi, A., Knapic, S., Främling, K.: Explainable agents for less bias in human-agent decision making. In: International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems, pp. 129–146 (2020). Springer
91. Bibal, A., Lognoul, M., De Streel, A., Frénay, B.: Legal requirements on explainability in machine learning. Artificial Intelligence and Law **29**(2), 149–169 (2021)
92. Bibal, A., Lognoul, M., de Streel, A., Frénay, B.: Impact of legal requirements on explainability in machine learning. arXiv preprint arXiv:2007.05479 (2020)
93. Turian, J., Ratinov, L., Bengio, Y.: Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 384–394 (2010)
94. Bollegala, D., Maehara, T., Kawarabayashi, K.-i.: Embedding semantic relations into word representations. In: Twenty-fourth International Joint Conference on Artificial Intelligence (2015)
95. Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., Gao, J.: Deep learning–based text classification: A comprehensive review. ACM Computing Surveys (CSUR) **54**(3), 1–40 (2021)
96. Pavlinek, M., Podgorelec, V.: Text classification method based on self-training and lda topic models. Expert Systems with Applications **80**, 83–93 (2017)
97. Chen, Q., Yao, L., Yang, J.: Short text classification based on lda topic model. In: 2016 International Conference on Audio, Language and Image Processing (ICALIP), pp. 749–753 (2016). IEEE
98. Landauer, T.K., Foltz, P.W., Laham, D.: An introduction to latent semantic analysis. Discourse processes **25**(2-3), 259–284 (1998)
99. Jayashree, P., Srijith, P.: Evaluation of deep gaussian processes for text classification. In: Proceedings of the 12th Language Resources and Evaluation Conference, pp. 1485–1491 (2020)
100. Chen, S.-H., Lee, Y.-S., Tai, T.-C., Wang, J.-C.: Gaussian process based text categorization for healthy information. In: 2015 International Conference on Orange Technologies (ICOT), pp. 30–33 (2015). doi:10.1109/ICOT.2015.7498487
101. Cheng, Y., Ye, Z., Wang, M., Zhang, Q.: Document classification based on convolutional neural network and hierarchical attention network. Neural Network World **29**(2), 83–98 (2019)
102. Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., Yu, P.S., He, L.: A survey on text classification: From shallow to deep learning. arXiv preprint arXiv:2008.00364 (2020)
103. Yang, J., Bai, L., Guo, Y.: A survey of text classification models. In: Proceedings of the 2020 2nd International Conference on Robotics, Intelligent Control and Artificial Intelligence, pp. 327–334 (2020)
104. Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: Twenty-ninth AAAI Conference on Artificial Intelligence (2015)
105. Farkas, J.: Document classification and recurrent neural networks. In: Proceedings of the 1995 Conference of the Centre for Advanced Studies on Collaborative Research, p. 21 (1995)
106. Chen, C., Lin, X., Terejanu, G.: An approximate bayesian long short-term memory algorithm for outlier detection. In: 2018 24th International Conference on Pattern Recognition (ICPR), pp. 201–206 (2018). IEEE
107. D'Ambrosio, A., Grundmann, H., Donker, T.: An open-source integrated framework for the automation of citation collection and screening in systematic reviews. arXiv preprint arXiv:2202.10033 (2022)
108. D'Ambrosio, A.: Dataset related to the manuscript: "An open-source integrated framework for the automation of citation collection and screening in systematic reviews". Zenodo (2022). doi:10.5281/zenodo.6323361

Figures

Figure 1. Framework's visual representation.

Figure 2. Observed cumulative number of positive matches (red dots) sorted by simple query ordering. The [trunc. 90% PrI] of the cumulative positive matches estimated by the Bayesian logistic model is shown as a shaded area delimited by the 95% quantile of the PrI and by the observed number of positive matches (light blue lines). A darker blue line represents the median of the PrI.

Tables

Table 1. Distribution of retrieved records by source and session. For each source, we reported the number of records, percentage over the session total (after removing duplicates), and the number of records specific for a source as absolute value and as percentage over the source total. All session shows records after joining and deduplication of the Session 1 and Session 2 data set.

Session	Source	Records	% over total	Source specific records	% over source total
Session1	Total	17,755			
	Embase	12,719	71.6%	6,683	52.5%
	Pubmed	9,546	53.8%	3,457	36.2%
	Scopus	3,175	17.9%	298	9.39%
	WOS	2,100	11.8%	473	22.5%
	IEEE	60	0.34%	29	48.3%
Session2	Total	82,579			
	Embase	48,396	58.6%	40,826	84.4%
	Pubmed	28,811	34.9%	18,021	62.5%
	Scopus	17,070	20.7%	4,908	28.8%
	WOS	12,956	15.7%	2,817	21.7%
	IEEE	61	0.074%	22	36.1%
All Sessions	Total	98,371			
	Embase	59,604	60.6%	46,942	78.8%
	Pubmed	37,278	37.9%	21,371	57.3%
	Scopus	19,353	19.7%	5,181	26.8%
	WOS	14,367	14.6%	3,175	22.1%
	IEEE	108	0.11%	48	44.4%

Table 2. Results of the automatic classification and manual review rounds. The cumulative numbers of positives and negative records and their sum (Total labelled) and percentage over total are shown for each iteration. Also, the number of changes after review and their description is reported."Unlab." indicates unlabelled records marked for review. For each iteration, the number of features used by the engine is also reported. The first row reports the results of the initial manual labelling of records, which served as input for the automatic classification in Iteration 1. In Session 2, the engine uses the labels at the end of Session 1 to classify the newly added records.

Session	Iteration	Positives	Negatives	Total labelled (%)	Unlab. -¿ y	Unlab. -¿ n	Unlab. -¿ *	n -¿ y	Changes	N. features
Session1 (n = 17755)	Initial labelling	43	207	250 (1.41%)	43	207	0	0	250	2,289
	1	93	529	622 (3.5%)	50	322	0	0	372	2,289
	2	100	614	714 (4.02%)	6	86	0	1	93	3,750
	3	101	625	726 (4.09%)	1	11	0	0	12	3,834
	4	101	648	749 (4.22%)	0	23	0	0	23	3,856
	5	101	651	752 (4.24%)	0	3	0	0	3	3,856
	6	101	660	761 (4.29%)	0	9	0	0	9	3,856
	7	101	665	766 (4.31%)	0	5	0	0	5	3,856
Session2 (n = 98371)	1	106	934	1040 (1.06%)	5	270	998	0	1,273	4,729
	2	107	1,123	1230 (1.25%)	1	189	0	0	190	4,729
	3	107	1,176	1283 (1.3%)	0	53	0	0	53	4,733
	4	107	1,200	1307 (1.33%)	0	24	0	0	24	4,729
	5	107	1,209	1316 (1.34%)	0	9	0	0	9	4,729
	6	107	1,226	1333 (1.36%)	0	17	0	0	17	4,729

Table 3. Estimated performance summary. The table reports for each session, the number of reviewed records and the percentage over the total. Also, the posterior expected number of positive records, sensitivity and efficiency (as WSoR) are reported, with their 90% PrI truncated to the observed realisation in the dataset [trunc. PrI] (see. methods). Finally, the logistic model's median Bayesian R^2 [90% CrI] is reported. PrI: Predictive Intervals; CrI: Credibility Intervals.

Indicator	Session 1	Session 2
Total records	17,755	98,371
Reviewed records (% over total records)	766 (4.31%)	1,333 (1.36%)
Expected efficiency (over random) [trunc. 90% PrI]	95.6% [95.3%, 95.7%]	98.6% [98.1%, 98.6%]
Observed positive matches (% over total records)	101 (0.57%)	107 (0.11%)
Predicted positive matches [trunc. 90% PrI]	101 [101, 108]	110 [107, 147]
Expected sensitivity [trunc. 90% PrI]	100% [93.5%, 100%]	97.3% [72.8%, 100%]
Simple Model R^2 [90% CrI]	98.1% [97.4%, 98.3%]	98.2% [97.6%, 98.3%]

Additional Files

S1. Framework description and usage.
Instruction on how to use the framework and reproduce the results in the manuscript.

S2. Additional outputs.
Additional analysis outputs described in the manuscript.

S3. Online search instructions.
Instructions about how to prepare the framework to interact with the online scientific databases to collect records.