nature machine intelligence

ARTICLES

https://doi.org/10.1038/s42256-020-00287-7

Check for updates

OPEN

An open source machine learning framework for efficient and transparent systematic reviews

Rens van de Schoot[1]✉, Jonathan de Bruin[2], Raoul Schram[2], Parisa Zahedi[2], Jan de Boer[3], Felix Weijdema[3], Bianca Kramer[3], Martijn Huijts[4], Maarten Hoogerwerf[2], Gerbrich Ferdinands[1], Albert Harkema[1], Joukje Willemsen[1], Yongchao Ma[1], Qixiang Fang[1], Sybren Hindriks[1], Lars Tummers[5] and Daniel L. Oberski[1,6]

To help researchers conduct a systematic review or meta-analysis as efficiently and transparently as possible, we designed a tool to accelerate the step of screening titles and abstracts. For many tasks—including but not limited to systematic reviews and meta-analyses—the scientific literature needs to be checked systematically. Scholars and practitioners currently screen thousands of studies by hand to determine which studies to include in their review or meta-analysis. This is error prone and inefficient because of extremely imbalanced data: only a fraction of the screened studies is relevant. The future of systematic reviewing will be an interaction with machine learning algorithms to deal with the enormous increase of available text. We therefore developed an open source machine learning-aided pipeline applying active learning: ASReview. We demonstrate by means of simulation studies that active learning can yield far more efficient reviewing than manual reviewing while providing high quality. Furthermore, we describe the options of the free and open source research software and present the results from user experience tests. We invite the community to contribute to open source projects such as our own that provide measurable and reproducible improvements over current practice.

With the emergence of online publishing, the number of scientific manuscripts on many topics is skyrocketing[1]. All of these textual data present opportunities to scholars and practitioners while simultaneously confronting them with new challenges. Scholars often develop systematic reviews and meta-analyses to develop comprehensive overviews of the relevant topics[2]. The process entails several explicit and, ideally, reproducible steps, including identifying all likely relevant publications in a standardized way, extracting data from eligible studies and synthesizing the results. Systematic reviews differ from traditional literature reviews in that they are more replicable and transparent[3,4]. Such systematic overviews of literature on a specific topic are pivotal not only for scholars, but also for clinicians, policy-makers, journalists and, ultimately, the general public[5–7].

Given that screening the entire research literature on a given topic is too labour intensive, scholars often develop quite narrow searches. Developing a search strategy for a systematic review is an iterative process aimed at balancing recall and precision[8,9]; that is, including as many potentially relevant studies as possible while simultaneously limiting the total number of studies retrieved. The vast number of publications in the field of study often leads to a relatively precise search, with the risk of missing relevant studies. The process of systematic reviewing is error prone and extremely time intensive[10]. In fact, if the literature of a field is growing faster than the amount of time available for systematic reviews, adequate manual review of this field then becomes impossible[11].

The rapidly evolving field of machine learning has aided researchers by allowing the development of software tools that assist in developing systematic reviews[11–14]. Machine learning offers approaches to overcome the manual and time-consuming screening of large numbers of studies by prioritizing relevant studies via active learning[15]. Active learning is a type of machine learning in which a model can choose the data points (for example, records obtained from a systematic search) it would like to learn from and thereby drastically reduce the total number of records that require manual screening[16–18]. In most so-called human-in-the-loop[19] machine-learning applications, the interaction between the machine-learning algorithm and the human is used to train a model with a minimum number of labelling tasks. Unique to systematic reviewing is that not only do all relevant records (that is, titles and abstracts) need to seen by a researcher, but an extremely diverse range of concepts also need to be learned, thereby requiring flexibility in the modelling approach as well as careful error evaluation[11]. In the case of systematic reviewing, the algorithm(s) are interactively optimized for finding the most relevant records, instead of finding the most accurate model. The term researcher-in-the-loop was introduced[20] as a special case of human-in-the-loop with three unique components: (1) the primary output of the process is a selection of the records, not a trained machine learning model; (2) all records in the relevant selection are seen by a human at the end of the process[21]; (3) the use-case requires a reproducible workflow and complete transparency is required[22].

Existing tools that implement such an active learning cycle for systematic reviewing are described in Table 1; see the Supplementary Information for an overview of all of the software that we considered (note that this list was based on a review of software tools[12]).

[1]Department of Methodology and Statistics, Faculty of Social and Behavioral Sciences, Utrecht University, Utrecht, the Netherlands. [2]Department of Research and Data Management Services, Information Technology Services, Utrecht University, Utrecht, the Netherlands. [3]Utrecht University Library, Utrecht University, Utrecht, the Netherlands. [4]Department of Test and Quality Services, Information Technology Services, Utrecht University, Utrecht, the Netherlands. [5]School of Governance, Faculty of Law, Economics and Governance, Utrecht University, Utrecht, the Netherlands. [6]Department of Biostatistics, Data management and Data Science, Julius Center, University Medical Center Utrecht, Utrecht, the Netherlands. ✉e-mail: a.g.j.vandeschoot@uu.nl

Table 1 | Existing tools that implement an RITL-based active learning cycle for systematic reviewing

Name	Machine learning algorithms	Active learning features	Privacy policy
Abstrackr[58]	Classifier: SVM. Model inputs: user-provided keywords (relevant/irrelevant with degree of confidence); citations. Feature extraction: TF–IDF. Label options: relevant; borderline; irrelevant.	Query strategy: uncertainty-based; certainty-based; random sampling. Balance strategy: aggressive undersampling. Active learning starts after: a reasonable representation of the minority class has been labelled (ref. [14]). Retraining: asynchronous. Stopping: when the model predicts none of the remaining abstracts to be relevant.	GDPR notification: "We do not have a limit on how long we retain your account information and/or data.". "We do not share any information with third parties.".
ASReview[27]	Classifier: NB; SVM; DNN; LR; LSTM-base; LSTM-pool; RF. Model inputs: piece of text (for example, title and abstract). Feature extraction: Doc2Vec; embedding IDF, TF–IDF, sBERT. Label options: relevant; irrelevant.	Query strategy: uncertainty-based; certainty-based; random sampling; mixed sampling. Balance strategy: simple (no balancing); dynamic resampling (double and triple); undersampling. Active learning starts after: one label. Retraining: asynchronous. Stopping: currently left to the reviewer.	The software does not have access to user data, as the program runs locally.
Colandr[59]	Classifier: SVM with SGD learning. Model inputs: user-provided key terms and citation (abstract, title, keywords). Feature extraction: Word2Vec Label options: include; exclude.	Query strategy: certainty-based. Balance strategy: reweighting. Active learning starts after: 100 inclusions and 100 exclusions. Retraining: every 30 abstracts. Stopping: is left to the reviewer.	No terms and conditions available. The Colandr team was contacted and they ensured the user can remove data any time. In the future, user data will be used to improve Colandr but only if granted permission from the project owner.
FASTREAD[60]	Classifier: SVM. Model inputs: title and abstract. Feature extraction: TF–IDF. Label options: relevant; irrelevant.	Query strategy: uncertainty sampling; Certainty sampling. Users are allowed to switch between active learning types after thirty inclusions. Balance strategy: mix of weighting and aggressive undersampling. Active learning starts after: one relevant abstract is retrieved (through querying random abstracts). Retraining: every ten abstracts. Stopping: the number of relevant abstracts is estimated by semi-supervised learning.	The software does not have access to user data as the program runs locally.
Rayyan[61]	Classifier: SVM. Model inputs: user-provided key terms and citation (title and abstract). Feature extraction: unigrams, bigrams, MeSH terms. Label options: include; exclude; maybe.	Query strategy: Rayyan predicts a relevancy of a citation on a five-star scale. The user can order citations by their predicted relevancy. Balance strategy: unknown. Active learning starts after: unknown. Retraining: unknown; "as the user is labelling citations". Stopping: when there are no more citations to be labelled or when the model can no longer be improved.	Rayyan terms of service: 3.1: "Rayyan, may use any User data and information to evaluate and improve its performance and expand its services.". 3.4: "This Agreement is governed by the laws of the State of Qatar. By accessing this Rayyan website you consent to these terms and conditions and to the exclusive jurisdiction of the Qatar courts in all disputes arising out of such access.". 9.2.2: "Rayyan does not own User Content. The User retains the copyright of their Content. …".
RobotAnalyst[62]	Classifier: SVM. Model inputs: title; abstract; topic model proportions. Feature extraction: TF–IDF L2 normalized (title); BOW for abstract; LDA for topic model proportions. Label options: included; excluded; undecided.	Query strategy: uncertainty-based; certainty-based. Balance strategy: none. Active learning starts after: a manually labelled 'initial batch' of abstracts, randomly sampled or obtained through a focused search. Retraining: when to retrain is left to the user. Possible after every labelled citation Stopping: is left to the reviewer, however at least a sequence of excluded citations is necessary.	Not available.

An overview of those tools that implemented active learning and describe what machine learning algorithms have been implemented, which active learning features are available and information about privacy policy. As a starting point we used the systematic review[12] that describes machine learning-aided software tools for systematic reviewing. In Supplementary Table 1 we provide an overview of all tools found by Harrison and colleagues and indicate which open-source tools implemented machine learning and/or active learning. Note that we added FASTREAD, RobotAnalyst and ASReview to the overview, which were not described by Harrison and co-workers. Machine learning, the kind of machine learning model used; active learning, how active learning is implemented; privacy policy, quotes from privacy policy are given (if available) to indicate possible concerns; SVM, support vector machine; TF–IDF, term frequency–inverse document frequency; NB, naive Bayes; DNN, dense neural network; LR, logistic regression; LSTM, long short-term memory; RF, random forests; Doc2Vec, document to vector; embedding IDF, embedding inverse document frequency; sBERT, sentence bidirectional encoder representations from transformers; SGD, stochastic gradient descent; Word2Vec, words to vector; BOW, bag of words; LDA, latent dirichlet allocation; GDPR, general data protection regulation; MeSH, medical subject headings.

However, existing tools have two main drawbacks. First, many are closed source applications with black box algorithms, which is problematic as transparency and data ownership are essential in the era of open science[22]. Second, to our knowledge, existing tools lack the necessary flexibility to deal with the large range of possible concepts to be learned by a screening machine. For example, in systematic reviews, the optimal type of classifier will depend on variable parameters, such as the proportion of relevant publications in the initial search and the complexity of the inclusion criteria used by the researcher[23]. For this reason, any successful system must allow for a wide range of classifier types. Benchmark testing is crucial to understand the real-world performance of any machine learning-aided system, but such benchmark options are currently mostly lacking.

In this paper we present an open source machine learning-aided pipeline with active learning for systematic reviews called ASReview. The goal of ASReview is to help scholars and practitioners to get an overview of the most relevant records for their work as efficiently as possible while being transparent in the process. The open, free and ready-to-use software ASReview addresses all concerns mentioned above: it is open source, uses active learning, allows multiple machine learning models. It also has a benchmark mode, which is especially useful for comparing and designing algorithms. Furthermore, it is intended to be easily extensible, allowing third parties to add modules that enhance the pipeline. Although we focus this paper on systematic reviews, ASReview can handle any text source.

In what follows, we first present the pipeline for manual versus machine learning-aided systematic reviews. We then show how ASReview has been set up and how ASReview can be used in different workflows by presenting several real-world use cases. We subsequently demonstrate the results of simulations that benchmark performance and present the results of a series of user-experience tests. Finally, we discuss future directions.

Pipeline for manual and machine learning-aided systematic reviews

The pipeline of a systematic review without active learning traditionally starts with researchers doing a comprehensive search in multiple databases[24], using free text words as well as controlled vocabulary to retrieve potentially relevant references. The researcher then typically verifies that the key papers they expect to find are indeed included in the search results. The researcher downloads a file with records containing the text to be screened. In the case of systematic reviewing it contains the titles and abstracts (and potentially other metadata such as the authors's names, journal name, DOI) of potentially relevant references into a reference manager. Ideally, two or more researchers then screen the records's titles and abstracts on the basis of the eligibility criteria established beforehand[4]. After all records have been screened, the full texts of the potentially relevant records are read to determine which of them will be ultimately included in the review. Most records are excluded in the title and abstract phase. Typically, only a small fraction of the records belong to the relevant class, making title and abstract screening an important bottleneck in systematic reviewing process[25]. For instance, a recent study analysed 10,115 records and excluded 9,847 after title and abstract screening, a drop of more than 95%[26]. ASReview therefore focuses on this labour-intensive step.

The research pipeline of ASReview is depicted in Fig. 1. The researcher starts with a search exactly as described above and subsequently uploads a file containing the records (that is, metadata containing the text of the titles and abstracts) into the software. Prior knowledge is then selected, which is used for training of the first model and presenting the first record to the researcher. As screening is a binary classification problem, the reviewer must select at least one key record to include and exclude on the basis of background knowledge. More prior knowledge may result in improved efficiency of the active learning process.

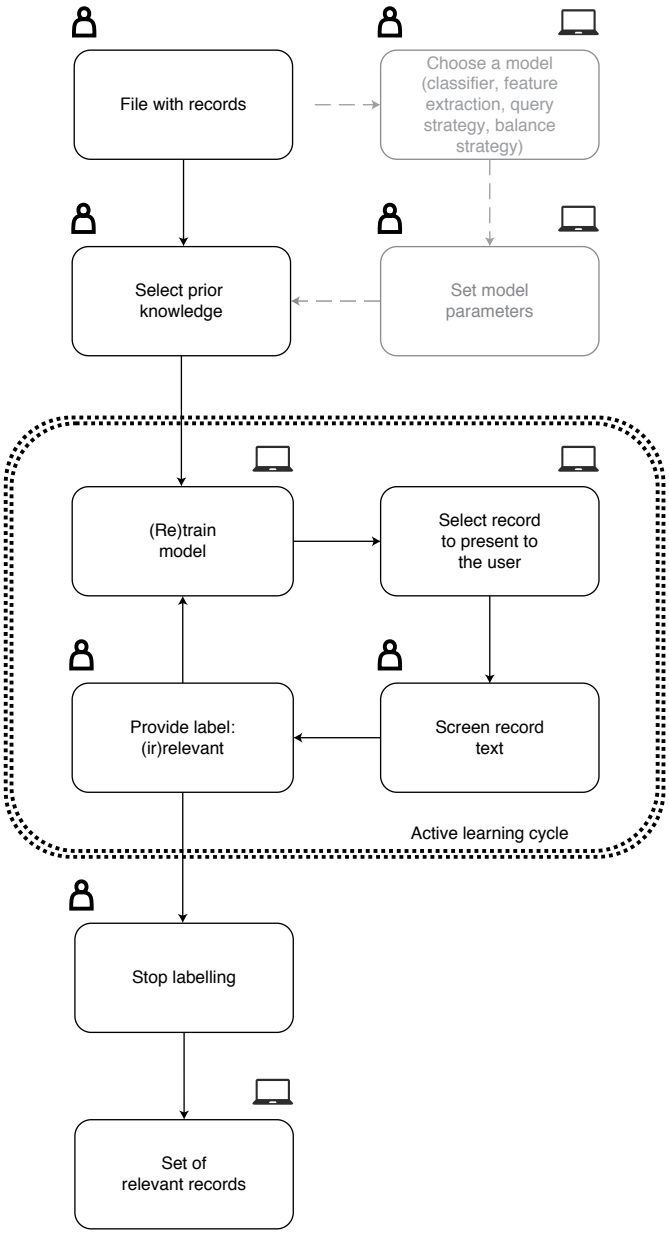

Fig. 1 | Machine learning-aided pipeline for ASReview. The symbols indicate whether the action is taken by a human, a computer, or whether both options are available.

A machine learning classifier is trained to predict study relevance (labels) from a representation of the record-containing text (feature space) on the basis of prior knowledge. We have purposefully chosen not to include an author name or citation network representation in the feature space to prevent authority bias in the inclusions. In the active learning cycle, the software presents one new record to be screened and labelled by the user. The user's binary label (1 for relevant versus 0 for irrelevant) is subsequently used to train a new model, after which a new record is presented to the user. This cycle continues up to a certain user-specified stopping criterion has been reached. The user now has a file with (1) records labelled as either relevant or irrelevant and (2) unlabelled records ordered from most to least probable to be relevant as predicted by the current model.

Table 2 | Implemented classifiers, feature extraction techniques, query strategies and balance strategies available in ASReview

Classifier	Feature extraction	Query strategy	Balance strategy
Naive Bayes (default)	TF–IDF (default)	Certainty-based sampling (default)	Dynamic resampling (double and triple) (double = default)
Support vector machine	Embedding-IDF	Uncertainty-based sampling	
Neural network	Sentence BERT	Random sampling	Undersampling
Logistic regression	Doc2Vec	Mixed sampling (for example, 5% random/95% certainty-based)	Simple (no balancing)
LSTM-base	Embedding LSTM		
LSTM-pool			
Random forests			

Note that not all combinations are possible. For example, the NB classifier cannot handle a feature matrix with negative values and thus cannot be combined with Doc2Vec; LSTM-base and LSTM-pool classifiers exclusively work with embedding LSTM feature extraction and vice versa. Technical details are described in our documentation[28].

This set-up helps to move through a large database much quicker than in the manual process, while the decision process simultaneously remains transparent.

Software implementation for ASReview

The source code[27] of ASReview is available open source under an Apache 2.0 license, including documentation[28]. Compiled and packaged versions of the software are available on the Python Package Index[29] or Docker Hub[30]. The free and ready-to-use software ASReview implements oracle, simulation and exploration modes. The oracle mode is used to perform a systematic review with interaction by the user, the simulation mode is used for simulation of the ASReview performance on existing datasets, and the exploration mode can be used for teaching purposes and includes several preloaded labelled datasets.

The oracle mode presents records to the researcher and the researcher classifies these. Multiple file formats are supported: (1) RIS files are used by digital libraries such as IEEE Xplore, Scopus and ScienceDirect; the citation managers Mendeley, RefWorks, Zotero and EndNote support the RIS format too. (2) Tabular datasets with the .csv, .xlsx and .xls file extensions. CSV files should be comma separated and UTF-8 encoded; the software for CSV files accepts a set of predetermined labels in line with the ones used in RIS files. Each record in the dataset should hold the metadata on, for example, a scientific publication. Mandatory metadata is text and can, for example, be titles or abstracts from scientific papers. If available, both are used to train the model, but at least one is needed. An advanced option is available that splits the title and abstracts in the feature-extraction step and weights the two feature matrices independently (for TF–IDF only). Other metadata such as author, date, DOI and keywords are optional but not used for training the models. When using ASReview in the simulation or exploration mode, an additional binary variable is required to indicate historical labelling decisions. This column, which is automatically detected, can also be used in the oracle mode as background knowledge for previous selection of relevant papers before entering the active learning cycle. If unavailable, the user has to select at least one relevant record that can be identified by searching the pool of records. At least one irrelevant record should also be identified; the software allows to search for specific records or presents random records that are most likely to be irrelevant due to the extremely imbalanced data.

The software has a simple yet extensible default model: a naive Bayes classifier, TF–IDF feature extraction, a dynamic resampling balance strategy[31] and certainty-based sampling[17,32] for the query strategy. These defaults were chosen on the basis of their consistently high performance in benchmark experiments across several datasets[31]. Moreover, the low computation time of these default settings makes them attractive in applications, given that the software should be able to run locally. Users can change the settings, shown in Table 2, and technical details are described in our documentation[28]. Users can also add their own classifiers, feature extraction techniques, query strategies and balance strategies.

ASReview has a number of implemented features (see Table 2). First, there are several classifiers available: (1) naive Bayes; (2) support vector machines; (3) logistic regression; (4) neural networks; (5) random forests; (6) LSTM-base, which consists of an embedding layer, an LSTM layer with one output, a dense layer and a single sigmoid output node; and (7) LSTM-pool, which consists of an embedding layer, an LSTM layer with many outputs, a max pooling layer and a single sigmoid output node. The feature extraction techniques available are Doc2Vec[33], embedding LSTM, embedding with IDF or TF–IDF[34] (the default is unigram, with the option to run *n*-grams while other parameters are set to the defaults of Scikit-learn[35]) and sBERT[36]. The available query strategies for the active learning part are (1) random selection, ignoring model-assigned probabilities; (2) uncertainty-based sampling, which chooses the most uncertain record according to the model (that is, closest to 0.5 probability); (3) certainty-based sampling (max in ASReview), which chooses the record most likely to be included according to the model; and (4) mixed sampling, which uses a combination of random and certainty-based sampling.

There are several balance strategies that rebalance and reorder the training data. This is necessary, because the data is typically extremely imbalanced and therefore we have implemented the following balance strategies: (1) full sampling, which uses all of the labelled records; (2) undersampling the irrelevant records so that the included and excluded records are in some particular ratio (closer to one); and (3) dynamic resampling, a novel method similar to undersampling in that it decreases the imbalance of the training data[31]. However, in dynamic resampling, the number of irrelevant records is decreased, whereas the number of relevant records is increased by duplication such that the total number of records in the training data remains the same. The ratio between relevant and irrelevant records is not fixed over interactions, but dynamically updated depending on the number of labelled records, the total number of records and the ratio between relevant and irrelevant records. Details on all of the described algorithms can be found in the code and documentation referred to above.

By default, ASReview converts the records's texts into a document-term matrix, terms are converted to lowercase and no stop words are removed by default (but this can be changed). As the document-term matrix is identical in each iteration of the active learning cycle, it is generated in advance of model training and stored in the (active learning) state file. Each row of the document-term matrix can easily be requested from the state-file. Records are internally identified by their row number in the input dataset. In oracle mode, the record that is selected to be classified is retrieved from the state file and the record text and other metadata (such as title and abstract) are retrieved from the original dataset (from the file or the computer's memory). ASReview can run on your local computer, or on a (self-hosted) local or remote server. Data (all records and their labels) remain on the users's computer. Data ownership and confidentiality are crucial and no data are processed or used in any way by third parties. This is unique by

comparison with some of the existing systems, as shown in the last column of Table 1.

Real-world use cases and high-level function descriptions

Below we highlight a number of real-world use cases and high-level function descriptions for using the pipeline of ASReview.

ASReview can be integrated in classic systematic reviews or meta-analyses. Such reviews or meta-analyses entail several explicit and reproducible steps, as outlined in the PRISMA guidelines[4]. Scholars identify all likely relevant publications in a standardized way, screen retrieved publications to select eligible studies on the basis of defined eligibility criteria, extract data from eligible studies and synthesize the results. ASReview fits into this process, particularly in the abstract screening phase. ASReview does not replace the initial step of collecting all potentially relevant studies. As such, results from ASReview depend on the quality of the initial search process, including selection of databases[24] and construction of comprehensive searches using keywords and controlled vocabulary. However, ASReview can be used to broaden the scope of the search (by keyword expansion or omitting limitation in the search query), resulting in a higher number of initial papers to limit the risk of missing relevant papers during the search part (that is, more focus on recall instead of precision).

Furthermore, many reviewers nowadays move towards meta-reviews when analysing very large literature streams, that is, systematic reviews of systematic reviews[37]. This can be problematic as the various reviews included could use different eligibility criteria and are therefore not always directly comparable. Due to the efficiency of ASReview, scholars using the tool could conduct the study by analysing the papers directly instead of using the systematic reviews. Furthermore, ASReview supports the rapid update of a systematic review. The included papers from the initial review are used to train the machine learning model before screening of the updated set of papers starts. This allows the researcher to quickly screen the updated set of papers on the basis of decisions made in the initial run.

As an example case, let us look at the current literature on COVID-19 and the coronavirus. An enormous number of papers are being published on COVID-19. It is very time consuming to manually find relevant papers (for example, to develop treatment guidelines). This is especially problematic as urgent overviews are required. Medical guidelines rely on comprehensive systematic reviews, but the medical literature is growing at breakneck pace and the quality of the research is not universally adequate for summarization into policy[38]. Such reviews must entail adequate protocols with explicit and reproducible steps, including identifying all potentially relevant papers, extracting data from eligible studies, assessing potential for bias and synthesizing the results into medical guidelines. Researchers need to screen (tens of) thousands of COVID-19-related studies by hand to find relevant papers to include in their overview. Using ASReview, this can be done far more efficiently by selecting key papers that match their (COVID-19) research question in the first step; this should start the active learning cycle and lead to the most relevant COVID-19 papers for their research question being presented next. A plug-in was therefore developed for ASReview[39], which contained three databases that are updated automatically whenever a new version is released by the owners of the data: (1) the Cord19 database, developed by the Allen Institute for AI, with over all publications on COVID-19 and other coronavirus research (for example SARS, MERS and so on) from PubMed Central, the WHO COVID-19 database of publications, the preprint servers bioRxiv and medRxiv and papers contributed by specific publishers[40]. The CORD-19 dataset is updated daily by the Allen Institute for AI and updated also daily in the plugin. (2) In addition to the full dataset, we automatically construct a daily subset of the database with studies published after December 1st, 2019 to search for relevant papers published during the COVID-19 crisis. (3) A separate dataset of COVID-19 related preprints, containing metadata of preprints from over 15 preprints servers across disciplines, published since January 1st, 2020[41]. The preprint dataset is updated weekly by the maintainers and then automatically updated in ASReview as well. As this dataset is not readily available to researchers through regular search engines (for example, PubMed), its inclusion in ASReview provided added value to researchers interested in COVID-19 research, especially if they want a quick way to screen preprints specifically.

Simulation study

To evaluate the performance of ASReview on a labelled dataset, users can employ the simulation mode. As an example, we ran simulations based on four labelled datasets with version 0.7.2 of ASReview. All scripts to reproduce the results in this paper can be found on Zenodo (https://doi.org/10.5281/zenodo.4024122)[42], whereas the results are available at OSF (https://doi.org/10.17605/OSF.IO/2JKD6)[43].

Datasets. First, we analysed the performance for a study systematically describing studies that performed viral metagenomic next-generation sequencing in common livestock such as cattle, small ruminants, poultry and pigs[44]. Studies were retrieved from Embase ($n=1,806$), Medline ($n=1,384$), Cochrane Central ($n=1$), Web of Science ($n=977$) and Google Scholar ($n=200$, the top relevant references). After deduplication this led to 2,481 studies obtained in the initial search, of which 120 were inclusions (4.84%).

A second simulation study was performed on the results for a systematic review of studies on fault prediction in software engineering[45]. Studies were obtained from ACM Digital Library, IEEExplore and the ISI Web of Science. Furthermore, a snowballing strategy and a manual search were conducted, accumulating to 8,911 publications of which 104 were included in the systematic review (1.2%).

A third simulation study was performed on a review of longitudinal studies that applied unsupervised machine learning techniques to longitudinal data of self-reported symptoms of the post-traumatic stress assessed after trauma exposure[46,47]; 5,782 studies were obtained by searching Pubmed, Embase, PsychInfo and Scopus and through a snowballing strategy in which both the references and the citation of the included papers were screened. Thirty-eight studies were included in the review (0.66%).

A fourth simulation study was performed on the results for a systematic review on the efficacy of angiotensin-converting enzyme inhibitors, from a study collecting various systematic review datasets from the medical sciences[15]. The collection is a subset of 2,544 publications from the TREC 2004 Genomics Track document corpus[48]. This is a static subset from all MEDLINE records from 1994 through 2003, which allows for replicability of results. Forty-one publications were included in the review (1.6%).

Performance metrics. We evaluated the four datasets using three performance metrics. We first assess the work saved over sampling (WSS), which is the percentage reduction in the number of records needed to screen achieved by using active learning instead of screening records at random; WSS is measured at a given level of recall of relevant records, for example 95%, indicating the work reduction in screening effort at the cost of failing to detect 5% of the relevant records. For some researchers it is essential that all relevant literature on the topic is retrieved; this entails that the recall should be 100% (that is, WSS@100%). We also propose the amount of relevant references found after having screened the first 10% of the records (RRF10%). This is a useful metric for getting a quick overview of the relevant literature.

Results. For every dataset, 15 runs were performed with one random inclusion and one random exclusion (see Fig. 2). The classical

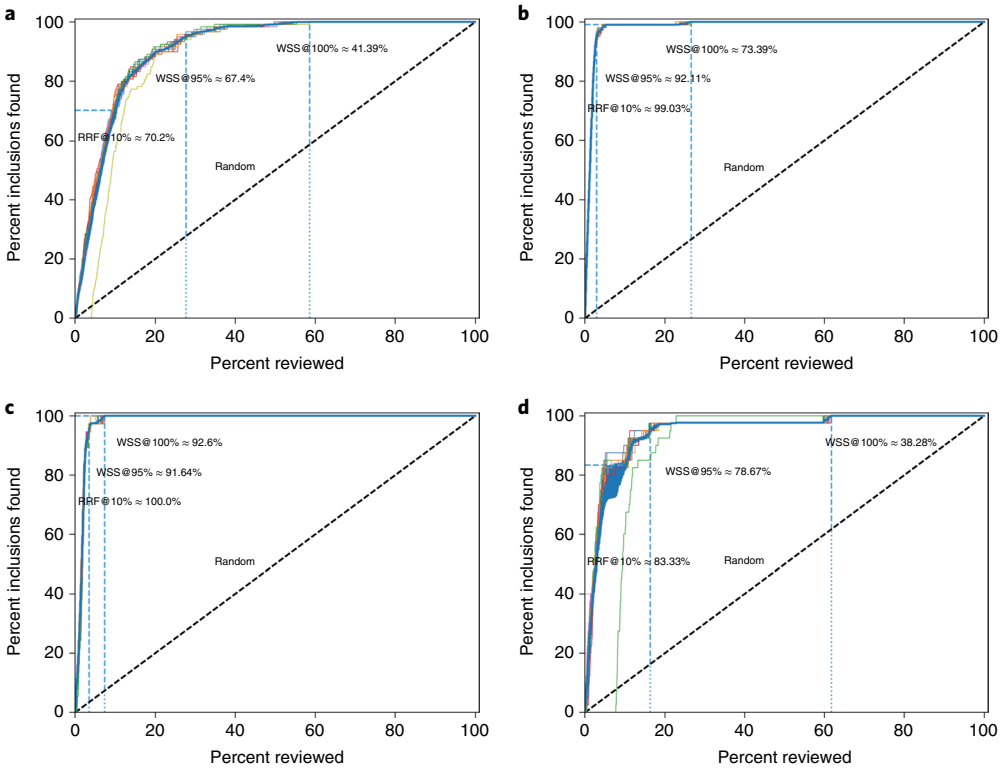

Fig. 2 | Simulation results. a–d, Results of the simulation study for the results for a study systematically review studies that performed viral metagenomic next-generation sequencing in common livestock (**a**), results for a systematic review of studies on fault prediction in software engineering (**b**), results for longitudinal studies that applied unsupervised machine learning techniques on longitudinal data of self-reported symptoms of posttraumatic stress assessed after trauma exposure (**c**), and results for a systematic review on the efficacy of angiotensin-converting enzyme inhibitors (**d**). Fiteen runs (shown with separate lines) were performed for every dataset, with only one random inclusion and one random exclusion. The classical review performances with randomly found inclusions are shown by the dashed lines.

review performance with randomly found inclusions is shown by the dashed line. The average work saved over sampling at 95% recall for ASReview is 83% and ranges from 67% to 92%. Hence, 95% of the eligible studies will be found after screening between only 8% to 33% of the studies. Furthermore, the number of relevant abstracts found after reading 10% of the abstracts ranges from 70% to 100%. In short, our software would have saved many hours of work.

Usability testing (user experience testing)

We conducted a series of user experience tests to learn from end users how they experience the software and implement it in their workflow. The study was approved by the Ethics Committee of the Faculty of Social and Behavioral Sciences of Utrecht University (ID 20-104).

Unstructured interviews. The first user experience (UX) test—carried out in December 2019—was conducted with an academic research team in a substantive research field (public administration and organizational science) that has conducted various systematic reviews and meta-analyses. It was composed of three university professors (ranging from assistant to full) and three PhD candidates. In one 3.5 h session, the participants used the software and provided feedback via unstructured interviews and group discussions. The goal was to provide feedback on installing the software and testing the performance on their own data. After these sessions we prioritized the feedback in a meeting with the ASReview team, which resulted in the release of v.0.4 and v.0.6. An overview of all releases can be found on GitHub[27].

A second UX test was conducted with four experienced researchers developing medical guidelines based on classical systematic reviews, and two experienced reviewers working at a pharmaceutical non-profit organization who work on updating reviews with new data. In four sessions, held in February to March 2020, these users tested the software following our testing protocol. After each session we implemented the feedback provided by the experts and asked them to review the software again. The main feedback was about how to upload datasets and select prior papers. Their feedback resulted in the release of v.0.7 and v.0.9.

Systematic UX test. In May 2020 we conducted a systematic UX test. Two groups of users were distinguished: an unexperienced group and an experienced user who already used ASReview. Due to the COVID-19 lockdown the usability tests were conducted via video calling where one person gave instructions to the participant and one person observed, called human-moderated remote testing[49]. During the tests, one person (SH) asked the questions and helped the participant with the tasks, the other person observed and made notes, a user experience professional at the IT department of Utrecht University (MH).

To analyse the notes, thematic analysis was used, which is a method to analyse data by dividing the information in subjects that all have a different meaning[50] using the Nvivo 12 software[51]. When something went wrong the text was coded as showstopper, when something did not go smoothly the text was coded as doubtful, and when something went well the subject was coded as superb. The features the participants requested for future versions of the ASReview tool were discussed with the lead engineer of the ASReview team and were submitted to GitHub as issues or feature requests.

The answers to the quantitative questions can be found at the Open Science Framework[52]. The participants ($N=11$) rated

the tool with a grade of 7.9 (s.d. = 0.9) on a scale from one to ten (Table 2). The unexperienced users on average rated the tool with an 8.0 (s.d. = 1.1, $N = 6$). The experienced user on average rated the tool with a 7.8 (s.d. = 0.9, $N = 5$). The participants described the usability test with words such as helpful, accessible, fun, clear and obvious.

The UX tests resulted in the new release v0.10, v0.10.1 and the major release v0.11, which is a major revision of the graphical user interface. The documentation has been upgraded to make installing and launching ASReview more straightforward. We made setting up the project, selecting a dataset and finding past knowledge is more intuitive and flexible. We also added a project dashboard with information on your progress and advanced settings.

Continuous input via the open source community. Finally, the ASReview development team receives continuous feedback from the open science community about, among other things, the user experience. In every new release we implement features listed by our users. Recurring UX tests are performed to keep up with the needs of users and improve the value of the tool.

Conclusion

We designed a system to accelerate the step of screening titles and abstracts to help researchers conduct a systematic review or meta-analysis as efficiently and transparently as possible. Our system uses active learning to train a machine learning model that predicts relevance from texts using a limited number of labelled examples. The classifier, feature extraction technique, balance strategy and active learning query strategy are flexible. We provide an open source software implementation, ASReview with state-of-the-art systems across a wide range of real-world systematic reviewing applications. Based on our experiments, ASReview provides defaults on its parameters, which exhibited good performance on average across the applications we examined. However, we stress that in practical applications, these defaults should be carefully examined; for this purpose, the software provides a simulation mode to users. We encourage users and developers to perform further evaluation of the proposed approach in their application, and to take advantage of the open source nature of the project by contributing further developments.

Drawbacks of machine learning-based screening systems, including our own, remain. First, although the active learning step greatly reduces the number of manuscripts that must be screened, it also prevents a straightforward evaluation of the system's error rates without further onerous labelling. Providing users with an accurate estimate of the system's error rate in the application at hand is therefore a pressing open problem. Second, although, as argued above, the use of such systems is not limited in principle to reviewing, no empirical benchmarks of actual performance in these other situations yet exist to our knowledge. Third, machine learning-based screening systems automate the screening step only; although the screening step is time-consuming and a good target for automation, it is just one part of a much larger process, including the initial search, data extraction, coding for risk of bias, summarizing results and so on. Although some other works, similar to our own, have looked at (semi-)automating some of these steps in isolation[53,54], to our knowledge the field is still far removed from an integrated system that would truly automate the review process while guaranteeing the quality of the produced evidence synthesis. Integrating the various tools that are currently under development to aid the systematic reviewing pipeline is therefore a worthwhile topic for future development.

Possible future research could also focus on the performance of identifying full text articles with different document length and domain-specific terminologies or even other types of text, such as newspaper articles and court cases. When the selection of past knowledge is not possible based on expert knowledge, alternative methods could be explored. For example, unsupervised learning or pseudolabelling algorithms could be used to improve training[55,56]. In addition, as the NLP community pushes forward the state of the art in feature extraction methods, these are easily added to our system as well. In all cases, performance benefits should be carefully evaluated using benchmarks for the task at hand. To this end, common benchmark challenges should be constructed that allow for an even comparison of the various tools now available. To facilitate such a benchmark, we have constructed a repository of publicly available systematic reviewing datasets[57].

The future of systematic reviewing will be an interaction with machine learning algorithms to deal with the enormous increase of available text. We invite the community to contribute to open source projects such as our own, as well as to common benchmark challenges, so that we can provide measurable and reproducible improvement over current practice.

Data availability

The results described in this paper are available at the Open Science Framework (https://doi.org/10.17605/OSF.IO/2JKD6)[43]. The answers to the quantitative questions of the UX test can be found at the Open Science Framework (OSF.IO/7PQNM)[52].

Code availability

All code to reproduce the results described in this paper can be found on Zenodo (https://doi.org/10.5281/zenodo.4024122)[42]. All code for the software ASReview is available under an Apache 2.0 license (https://doi.org/10.5281/zenodo.3345592)[27], is maintained on GitHub[63] and includes documentation (https://doi.org/10.5281/zenodo.4287120)[28].

Received: 4 June 2020; Accepted: 17 December 2020;
Published online: 1 February 2021

References

1. Bornmann, L. & Mutz, R. Growth rates of modern science: a bibliometric analysis based on the number of publications and cited references. *J. Assoc. Inf. Sci. Technol.* **66**, 2215–2222 (2015).
2. Gough, D., Oliver, S. & Thomas, J. *An Introduction to Systematic Reviews* (Sage, 2017).
3. Cooper, H. *Research Synthesis and Meta-analysis: A Step-by-Step Approach* (SAGE Publications, 2015).
4. Liberati, A. et al. The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate health care interventions: explanation and elaboration. *J. Clin. Epidemiol.* **62**, e1–e34 (2009).
5. Boaz, A. et al. *Systematic Reviews: What have They Got to Offer Evidence Based Policy and Practice?* (ESRC UK Centre for Evidence Based Policy and Practice London, 2002).
6. Oliver, S., Dickson, K. & Bangpan, M. *Systematic Reviews: Making Them Policy Relevant. A Briefing for Policy Makers and Systematic Reviewers* (UCL Institute of Education, 2015).
7. Petticrew, M. Systematic reviews from astronomy to zoology: myths and misconceptions. *Brit. Med. J.* **322**, 98–101 (2001).
8. Lefebvre, C., Manheimer, E. & Glanville, J. in *Cochrane Handbook for Systematic Reviews of Interventions* (eds. Higgins, J. P. & Green, S.) 95–150 (John Wiley & Sons, 2008); https://doi.org/10.1002/9780470712184.ch6.
9. Sampson, M., Tetzlaff, J. & Urquhart, C. Precision of healthcare systematic review searches in a cross-sectional sample. *Res. Synth. Methods* **2**, 119–125 (2011).
10. Wang, Z., Nayfeh, T., Tetzlaff, J., O'Blenis, P. & Murad, M. H. Error rates of human reviewers during abstract screening in systematic reviews. *PLoS ONE* **15**, e0227742 (2020).
11. Marshall, I. J. & Wallace, B. C. Toward systematic review automation: a practical guide to using machine learning tools in research synthesis. *Syst. Rev.* **8**, 163 (2019).
12. Harrison, H., Griffin, S. J., Kuhn, I. & Usher-Smith, J. A. Software tools to support title and abstract screening for systematic reviews in healthcare: an evaluation. *BMC Med. Res. Methodol.* **20**, 7 (2020).
13. 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.* **4**, 5 (2015).

14. Wallace, B. C., Trikalinos, T. A., Lau, J., Brodley, C. & Schmid, C. H. Semi-automated screening of biomedical citations for systematic reviews. *BMC Bioinf.* **11**, 55 (2010).
15. Cohen, A. M., Hersh, W. R., Peterson, K. & Yen, P.-Y. Reducing workload in systematic review preparation using automated citation classification. *J. Am. Med. Inform. Assoc.* **13**, 206–219 (2006).
16. Kremer, J., Steenstrup Pedersen, K. & Igel, C. Active learning with support vector machines. *WIREs Data Min. Knowl. Discov.* **4**, 313–326 (2014).
17. Miwa, M., Thomas, J., O'Mara-Eves, A. & Ananiadou, S. Reducing systematic review workload through certainty-based screening. *J. Biomed. Inform.* **51**, 242–253 (2014).
18. Settles, B. *Active Learning Literature Survey* (Minds@UW, 2009); https://minds.wisconsin.edu/handle/1793/60660
19. Holzinger, A. Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Inform.* **3**, 119–131 (2016).
20. Van de Schoot, R. & De Bruin, J. *Researcher-in-the-loop for Systematic Reviewing of Text Databases* (Zenodo, 2020); https://doi.org/10.5281/zenodo.4013207
21. Kim, D., Seo, D., Cho, S. & Kang, P. Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec. *Inf. Sci.* **477**, 15–29 (2019).
22. Nosek, B. A. et al. Promoting an open research culture. *Science* **348**, 1422–1425 (2015).
23. Kilicoglu, H., Demner-Fushman, D., Rindflesch, T. C., Wilczynski, N. L. & Haynes, R. B. Towards automatic recognition of scientifically rigorous clinical research evidence. *J. Am. Med. Inform. Assoc.* **16**, 25–31 (2009).
24. Gusenbauer, M. & Haddaway, N. R. Which academic search systems are suitable for systematic reviews or meta-analyses? Evaluating retrieval qualities of Google Scholar, PubMed, and 26 other resources. *Res. Synth. Methods* **11**, 181–217 (2020).
25. 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**, e012545 (2017).
26. de Vries, H., Bekkers, V. & Tummers, L. Innovation in the Public Sector: a systematic review and future research agenda. *Public Adm.* **94**, 146–166 (2016).
27. Van de Schoot, R. et al. *ASReview: Active Learning for Systematic Reviews* (Zenodo, 2020); https://doi.org/10.5281/zenodo.3345592
28. De Bruin, J. et al. *ASReview Software Documentation 0.14* (Zenodo, 2020); https://doi.org/10.5281/zenodo.4287120
29. *ASReview PyPI Package* (ASReview Core Development Team, 2020); https://pypi.org/project/asreview/
30. *Docker container for ASReview* (ASReview Core Development Team, 2020); https://hub.docker.com/r/asreview/asreview
31. Ferdinands, G. et al. *Active Learning for Screening Prioritization in Systematic Reviews—A Simulation Study* (OSF Preprints, 2020); https://doi.org/10.31219/osf.io/w6qbg
32. Fu, J. H. & Lee, S. L. Certainty-enhanced active learning for improving imbalanced data classification. In *2011 IEEE 11th International Conference on Data Mining Workshops* 405–412 (IEEE, 2011).
33. Le, Q. V. & Mikolov, T. Distributed representations of sentences and documents. Preprint at https://arxiv.org/abs/1405.4053 (2014).
34. Ramos, J. Using TF–IDF to determine word relevance in document queries. In *Proc. 1st Instructional Conference on Machine Learning* Vol. 242, 133–142 (ICML, 2003).
35. Pedregosa, F. et al. Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
36. Reimers, N. & Gurevych, I. Sentence-BERT: sentence embeddings using siamese BERT-networks Preprint at https://arxiv.org/abs/1908.10084 (2019).
37. Smith, V., Devane, D., Begley, C. M. & Clarke, M. Methodology in conducting a systematic review of systematic reviews of healthcare interventions. *BMC Med. Res. Methodol.* **11**, 15 (2011).
38. Wynants, L. et al. Prediction models for diagnosis and prognosis of COVID-19: systematic review and critical appraisal. *Brit. Med. J.* **369**, 1328 (2020).
39. Van de Schoot, R. et al. *Extension for COVID-19 Related Datasets in ASReview* (Zenodo, 2020). https://doi.org/10.5281/zenodo.3891420.
40. Lu Wang, L. et al. CORD-19: The COVID-19 open research dataset. Preprint at https://arxiv.org/abs/2004.10706 (2020).
41. Fraser, N. & Kramer, B. *Covid19_preprints* (FigShare, 2020); https://doi.org/10.6084/m9.figshare.12033672.v18
42. Ferdinands, G., Schram, R., Van de Schoot, R. & De Bruin, J. *Scripts for 'ASReview: Open Source Software for Efficient and Transparent Active Learning for Systematic Reviews'* (Zenodo, 2020); https://doi.org/10.5281/zenodo.4024122
43. Ferdinands, G., Schram, R., van de Schoot, R. & de Bruin, J. *Results for 'ASReview: Open Source Software for Efficient and Transparent Active Learning for Systematic Reviews'* (OSF, 2020); https://doi.org/10.17605/OSF.IO/2JKD6
44. Kwok, K. T. T., Nieuwenhuijse, D. F., Phan, M. V. T. & Koopmans, M. P. G. Virus metagenomics in farm animals: a systematic review. *Viruses* **12**, 107 (2020).
45. Hall, T., Beecham, S., Bowes, D., Gray, D. & Counsell, S. A systematic literature review on fault prediction performance in software engineering. *IEEE Trans. Softw. Eng.* **38**, 1276–1304 (2012).
46. van de Schoot, R., Sijbrandij, M., Winter, S. D., Depaoli, S. & Vermunt, J. K. The GRoLTS-Checklist: guidelines for reporting on latent trajectory studies. *Struct. Equ. Model. Multidiscip. J.* **24**, 451–467 (2017).
47. van de Schoot, R. et al. Bayesian PTSD-trajectory analysis with informed priors based on a systematic literature search and expert elicitation. *Multivar. Behav. Res.* **53**, 267–291 (2018).
48. Cohen, A. M., Bhupatiraju, R. T. & Hersh, W. R. Feature generation, feature selection, classifiers, and conceptual drift for biomedical document triage. In *Proc. 13th Text Retrieval Conference* (TREC, 2004).
49. Vasalou, A., Ng, B. D., Wiemer-Hastings, P. & Oshlyansky, L. Human-moderated remote user testing: orotocols and applications. In *8th ERCIM Workshop, User Interfaces for All* Vol. 19 (ERCIM, 2004).
50. Joffe, H. in *Qualitative Research Methods in Mental Health and Psychotherapy: A Guide for Students and Practitioners* (eds Harper, D. & Thompson, A. R.) Ch. 15 (Wiley, 2012).
51. *NVivo v.12* (QSR International Pty, 2019).
52. Hindriks, S., Huijts, M. & van de Schoot, R. Data for UX-test ASReview - June 2020. *OSF* https://doi.org/10.17605/OSF.IO/7PQNM (2020).
53. Marshall, I. J., Kuiper, J. & Wallace, B. C. RobotReviewer: evaluation of a system for automatically assessing bias in clinical trials. *J. Am. Med. Inform. Assoc.* **23**, 193–201 (2016).
54. Nallapati, R., Zhou, B., dos Santos, C. N., Gulcehre, Ç. & Xiang, B. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proc. 20th SIGNLL Conference on Computational Natural Language Learning* 280–290 (Association for Computational Linguistics, 2016).
55. Xie, Q., Dai, Z., Hovy, E., Luong, M.-T. & Le, Q. V. Unsupervised data augmentation for consistency training. Preprint at https://arxiv.org/abs/1904.12848 (2019).
56. Ratner, A. et al. Snorkel: rapid training data creation with weak supervision. *VLDB J.* **29**, 709–730 (2020).
57. *Systematic Review Datasets* (ASReview Core Development Team, 2020); https://github.com/asreview/systematic-review-datasets
58. Wallace, B. C., Small, K., Brodley, C. E., Lau, J. & Trikalinos, T. A. Deploying an interactive machine learning system in an evidence-based practice center: Abstrackr. In *Proc. 2nd ACM SIGHIT International Health Informatics Symposium* 819–824 (Association for Computing Machinery, 2012).
59. Cheng, S. H. et al. Using machine learning to advance synthesis and use of conservation and environmental evidence. *Conserv. Biol.* **32**, 762–764 (2018).
60. Yu, Z., Kraft, N. & Menzies, T. Finding better active learners for faster literature reviews. *Empir. Softw. Eng.* **23**, 3161–3186 (2018).
61. Ouzzani, M., Hammady, H., Fedorowicz, Z. & Elmagarmid, A. Rayyan—a web and mobile app for systematic reviews. *Syst. Rev.* **5**, 210 (2016).
62. Przybyła, P. et al. Prioritising references for systematic reviews with RobotAnalyst: a user study. *Res. Synth. Methods* **9**, 470–488 (2018).
63. *ASReview: Active learning for Systematic Reviews* (ASReview Core Development Team, 2020); https://github.com/asreview/asreview

Acknowledgements

We would like to thank the Utrecht University Library, focus area Applied Data Science, and departments of Information and Technology Services, Test and Quality Services, and Methodology and Statistics, for their support. We also want to thank all researchers who shared data, participated in our user experience tests or who gave us feedback on ASReview in other ways. Furthermore, we would like to thank the editors and reviewers for providing constructive feedback. This project was funded by the Innovation Fund for IT in Research Projects, Utrecht University, the Netherlands.

Author contributions

R.v.d.S. and D.O. originally designed the project, with later input from L.T. J.d.Br. is the lead engineer, software architect and supervises the code base on GitHub. R.S. coded the algorithms and simulation studies. P.Z. coded the very first version of the software. J.d.Bo., F.W. and B.K. developed the systematic review pipeline. M.Huijts is leading the UX tests and was supported by S.H. M.Hoogerwerf developed the architecture of the produced (meta)data. G.F. conducted the simulation study together with R.S. A.H. performed the literature search comparing the different tools together with G.F. J.W. designed all the artwork and helped with formatting the manuscript. Y.M. and Q.F. are responsible for the preprocessing of the metadata under the supervision of J.d.Br. R.v.d.S, D.O. and L.T. wrote the paper with input from all authors. Each co-author has written parts of the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s42256-020-00287-7.

Correspondence and requests for materials should be addressed to R.v.d.S.

Peer review information *Nature Machine Intelligence* thanks Jian Wu and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.