International Journal of Data Science and Analytics
https://doi.org/10.1007/s41060-025-00777-0

REGULAR PAPER

Large-scale simulation study of active learning models for systematic reviews

Jelle Jasper Teijema[1] · Jonathan de Bruin[2] · Ayoub Bagheri[1] · Rens van de Schoot[1]

Received: 20 September 2024 / Accepted: 25 March 2025

Abstract
Despite progress in active learning, evaluation remains limited by constraints in simulation size, infrastructure, and dataset availability. This study advocates for large-scale simulations as the gold standard for evaluating active learning models in systematic review screening. Two large-scale simulations, totaling over 29 thousand runs, assessed active learning solutions. The first study evaluated 13 combinations of classification models and feature extraction techniques using high-quality datasets from the SYNERGY dataset. The second expanded this to 92 model combinations with additional classifiers and feature extractors. In every scenario tested, active learning outperformed random screening. The performance gained varied across datasets, models, and screening progression, ranging from considerable to near-flawless results. The findings demonstrate that active learning consistently outperforms random screening in systematic review tasks, offering significant efficiency gains. While the extent of improvement varies depending on the dataset, model choice, and screening stage, the overall advantage is clear. Since model performance differs, active learning systems should remain adaptable to accommodate new classifiers and feature extraction techniques. The publicly available results underscore the importance of open benchmarking to ensure reproducibility and the development of robust, generalizable active learning strategies.

Keywords Active learning · Systematic review · Screening phase · Large-scale simulation

1 Introduction

Methodologies used to reduce the screening labor for systematic reviews are continually being introduced [3, 30, 35–37, 57, 58, 60]. Particularly, the use of active learning for prioritization in systematic review screening [11, 23, 48, 63] has seen significant progress and innovation. This application of active learning has been integrated into several screening software tools [1, 12, 24–26, 31, 41, 42, 44, 46, 59, 61, 62, 64, 66], employing a variety of machine learning models to improve prioritization efficiency. However, while model development has been rapid, evaluation remains inconsistent. Additionally, many tools lack the flexibility to incorporate different machine learning models, limiting their applicability in research settings.

A simulation study emulates user labeling decisions using prelabeled data, which enables the recreation of a systematic review's precise performance. By adjusting parameters such as prior knowledge, feature extractors, and, classifiers and then re-running the simulations with the same prelabeled data, performance evaluations of machine learning models can be conducted (where in the current paper a model refers to a configuration comprising a feature extraction and classifier). A single simulation can provide insights, but the true value emerges with replicability and scalability.

Empirical support for active learning in screening prioritization largely relies on simulations. While these simulations are generally implemented adequately, many studies would benefit from larger, broader, and more reproducible simulations to strengthen their conclusions and practical relevance.

✉ Jelle Jasper Teijema
j.j.teijema@uu.nl

Jonathan de Bruin
j.debruin1@uu.nl

Ayoub Bagheri
a.bagheri@uu.nl

Rens van de Schoot
a.g.j.vandeschoot@uu.nl

[1] Department of Methodology and Statistics, Utrecht University, Padualaan 14, 3584 CH Utrecht, The Netherlands

[2] Department of Research and Data Management Services, Utrecht University, Padualaan 14, 3584 CH Utrecht, The Netherlands

Published online: 02 May 2025

Improving simulation quality helps maximize time savings, as the choice of model can translate into hours or even days of work saved. However, performance simulation studies in this field face several limitations, including minimal use of data, a lack of diversity in studied domains, limited model comparisons, and the use of non-standardized metrics, as shown in the systematic review by [55]. Addressing these challenges would improve the reliability and generalizability of active learning in systematic review screening.

First, the median number of datasets used in simulation studies evaluating the performance of multiple active learning models was under four datasets. The limited median number of datasets used in these studies may constrain the generalizability of the findings, as it is likely that the performance for a single or couple of datasets is not interpretable as general performance.

Some studies have incorporated multiple datasets [9, 32, 34, 68]. However, studies that use more than four datasets still predominantly focus on medical reviews, as shown in Fig. 1. Even the high-quality Cohen dataset [11], often considered the gold standard in the field, is limited to drug class reviews. Expanding the range of datasets beyond medical topics would further enhance the generalizability of active learning models across different domains and disciplines.

Third, it was found that most simulation studies that compared active learning models typically involved no more than three distinct models. From the evaluated studies, we identified 13 combinations of well-performing classifiers and feature extractors that are frequently utilized. Examples of most often used classifiers include Logistic Regression, Random Forest, and Support Vector Machines, paired with feature extraction techniques like TF-IDF and word embeddings. Additionally, a review of the active learning software tools[1] reveals a predominant use of Support Vector Machine in software.

Fourth, although many metrics exist [39], most major studies focus on a single one. The most commonly used is the *Work Saved over Sampling @ 95%* (WSS@95%) metric [11]. While useful for evaluating performance at a fixed recall level, WSS@95% provides no insight into how a model performs in scenarios such as quickly retrieving relevant records after a cold start or identifying the last-to-find records [7, 19, 22]. Evaluating scenarios like retrieval efficiency in the first 100 papers or the ability to find the final relevant paper would help tailor models to specific screening challenges.

These limitations set the stage for our study on the diversity of data and models in simulation studies.

The SYNERGY dataset [14], used in our simulations, is the most diverse collection of systematic review datasets currently available. It spans multiple disciplines, including medicine, psychology, computational sciences, and biology, making it suitable for testing active learning methods across fields. In addition to covering various research fields, the dataset includes a wide range of dataset sizes and relevance densities, allowing models to be tested under different screening conditions, from rare relevant records to high-prevalence scenarios.

Moreover, new models are continuously being developed, such as deep learning architectures and ensemble methods, which show promise in various applications [5, 10, 17, 27]. However, these newer models have yet to be widely adopted in simulation studies, which indicates a gap between model development and their application in active learning simulations.

The gap between the frequent use of certain models in active learning software tools and their evaluation with simulation studies can largely be attributed to the complexities involved in setting up and running simulations, especially on a large scale. Establishing a robust simulation infrastructure is a significant undertaking. While some software allows for limited simulation capabilities [21], most of the code used in the 48 simulation studies reviewed was custom-made.

To facilitate large-scale simulations, software such as ASReview [59] is essential, enabling seamless integration of various models. Additionally, workflow generators like ASReview's Makita [56] play a crucial role in setting up repeatable and reproducible simulations. These tools make it possible to leverage larger datasets, such as SYNERGY [14], for more extensive evaluations. However, the scale of such datasets results in a substantial number of simulation runs, necessitating adequate infrastructure to ensure efficient execution [45]. This study demonstrates the full simulation pipeline, providing a framework for future research in active learning for systematic review screening.

In the current study, we conduct two large-scale simulations to evaluate the performance of the active learning-based pipeline across a broad range of systematic review datasets. The **first simulation study** focuses on the 13 combinations of classifiers and feature extraction techniques identified in the systematic review, exploring both inter- and intra-dataset variability. These simulations use only two records to start the active learning cycle. One relevant and one irrelevant document, which together, are known as the *Prior Knowledge*.

In the **second simulation study**, we expand our analysis to 92 feature extractor (Table 2) and classifier (Table 3) combinations, selecting models that have performed well in natural language processing but are rarely used in active learning for systematic reviews. This includes both pretrained and newly trained models, evaluated for their potential to improve systematic review performance and precision. The amount of prior knowledge is increased based on findings from the first study and insights from previous research [7].

[1] github.com/Rensvandeschoot/software-overview-machine-learning-for-screening-text#overview-of-available-models.

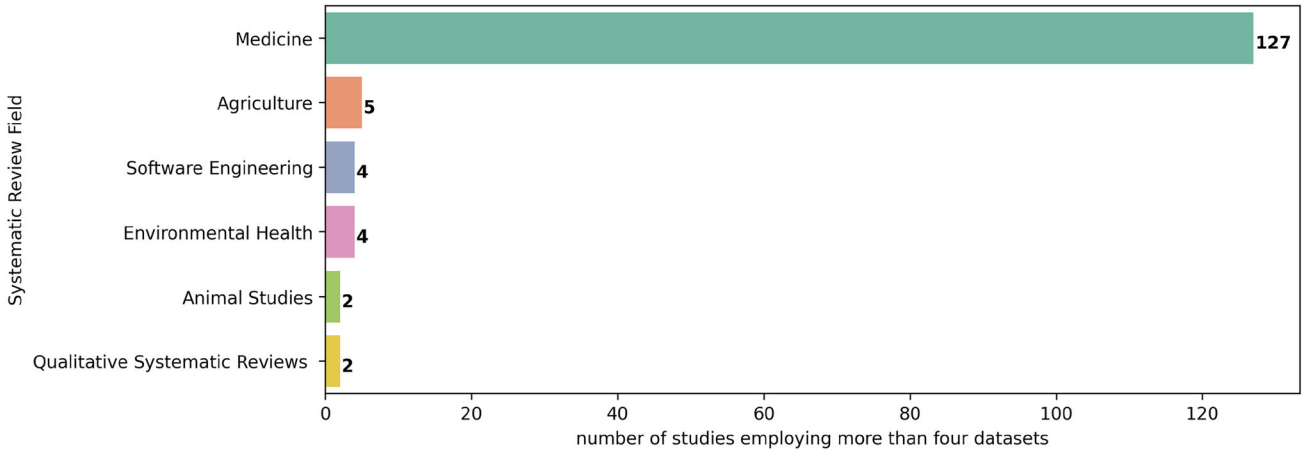


Fig. 1 Distribution of fields in simulation studies employing more than four datasets, as found in [53]

Our study has two key objectives:

1. To analyze variability in simulation studies, both across datasets and within individual datasets, as well as differences between models.
2. To evaluate model performance across different phases of the simulation: early screening, final screening, and overall effectiveness.

1.1 Background AI aided screening

Screening prioritization is explained in [11, 59], and in-depth in Box 1 of [29]. These offer a detailed exploration of active learning, current simulation research, and challenges in assessing active learning models in systematic reviews. This background provides the context for understanding the methods and applications of active learning.

Systematic reviews are a method for synthesizing evidence to answer specific research questions [13]. This process typically involves several phases: formulating a research question, designing a search strategy, screening records for relevance, and synthesizing the findings. Among these, the screening phase, where researchers evaluate large sets of titles and abstracts, is especially time-intensive and is the focus of this work. Screening prioritization is part of the PRISMA checklist, as mentioned as *Priority screening* in box 3 of [40].

Active learning is a form of machine learning that does not require a fully labeled dataset. Instead, its performance is refined in iterative cycles through interactions with human reviewers. At each iteration, the model requests labels from the human reviewer, learns from that new information, and improves its predictions. The enhanced model then selects records more accurately, enabling the human reviewer to label only the most informative items. This positive feedback loop is especially suitable for systematic review screening, which generally begins with little labeled data and generates these labels as part of the review process.

Although an active learning cycle can start with no data, prior knowledge accelerates initial training and avoids a *cold start*. Without prior knowledge, the model must rely on random screening. Even a small number of labeled documents improves early performance. In practice, reviewers often already know of some relevant documents, even if those items do not perfectly match the research question.

In a systematic review pipeline, the model improves with each labeling cycle, reaching sufficient accuracy to identify relevant documents earlier than random screening. This accelerated discovery leads to an increasingly sparse distribution of relevant documents, while many irrelevant documents remain. Once no new relevant items appear within a defined interval, screening can stop, reducing the manual workload.

Throughout this work, the expression "the machine learning model", or simply, "model", refers to the combined process of feature extraction and classification.

2 Methodology

2.1 Data

For both simulation studies, we use the SYNERGY dataset [14], detailed in Table 1. SYNERGY is a dataset the highest quality dataset available in this new field tailored for study selection in systematic reviews. It entails multiple datasets of scientific literature retrieved from bibliographic databases.

While this data collection is the most diverse dataset in terms of research categories currently available, the chosen data collection is still 50% "medicine, NOS,". It does include 3 datasets from computational sciences, 7 from psychology (with and without medicine), and 3 from biology (with medicine). In terms of number of records, datasets

vary from 238 to 48,375 (7 very small: < 1000; 7 small: $1000-3000$; 10 medium: $4000-10,000$; and 2 very large: $> 30,000$). Density of relevant records also varies from "needle-in-a-haystack" ($< 0.25\%$) to more than "one-in-five" ($> 20\%$), with fairly even representation across data sets (4 data sets - very rare: $< 0.5\%$; 6 - rare: $0.5-0.99\%$; 7 - average: $1.0-2.2\%$; 8 - frequent: $4-15\%$; 1 - abundant: $> 20\%$).

Each dataset corresponds to a single published systematic review and consists of rows of data, where each row represents a scientific record (e.g., journal article, preprint, or report). Each record contains the title, abstract, and a binary inclusion/exclusion label (0 or 1), indicating whether the researcher included the record in their systematic review. These datasets provide the basis for our simulations and reflect the results of manual screening performed during the systematic review process.

In our context, each record corresponds to a scientific publication (e.g., journal article, report, or preprint) retrieved from recognized bibliographic databases. Each row in the data set contains the publication's title, abstract, and label (inclusion/exclusion) indicating its relevance to the systematic review. This structure allows us to apply screening and classification methods consistently across different sources and topic areas.

Given the large size of the Walker_2018 dataset (48,375 records), the number of simulations is limited to five per model for the first study. This approach optimizes resource allocation. For the second study, the dataset size is reduced using a stratified sampling technique while preserving the original class distribution. The dataset is down-sampled to 4837 records, maintaining the label ratios (before: label 0—0.984248, label 1—0.015752; after: label 0—0.984288, label 1—0.015712). This reduction ensures more efficient use of computational resources while retaining the representativeness of the data (Table 1).

2.2 Overview simulation design

We perform simulations on 26 prelabeled datasets derived from existing systematic reviews. In the first simulation study, we run all permutations of the relevant record, classifier, feature extractor, and SYNERGY dataset. In the second simulation study, we increase the classifier and feature extractor pool to 92 combinations but remove the relevant records from the permutations.

$$N_{\text{sim, S1}} = 13_{\text{models}} \cdot n_{\text{relevant}}$$

$$N_{\text{sim, S2}} = 92_{\text{models}}$$

$$N_{\text{sim, total}} = \sum_{i=1}^{n_{\text{datasets}}} \left(N_{\text{sim, S1},i} + N_{\text{sim, S2},i} \right)$$

The first simulation study is used to gauge the reliability and stability of the results in preparation for the second study, where we replace iterating over all relevant records with running a single simulation per combination per dataset. To combat the extra instability in this new format, we increase the amount of prior knowledge, as the increased prior knowledge will reduce the amount of performance fluctuation in the first cycles of the simulation. Based on the findings of [7], we set the prior knowledge to a level that ensures stability, using a number of records that will minimize early-cycle variability while maintaining a realistic screening scenario.

Our study aims to achieve a high degree of reproducibility by adopting an open-source approach and making all data and code openly accessible [29]. The simulations are run on the open-source cloud platform Exoscale[2] We developed a custom Docker image [50], which is available to the public. An in-depth explanation of the Docker image's functionality can be found on its GitHub[3] The processing tasks within the Docker image are managed using a Kubernetes cluster[4]

2.3 Models

The classifiers in this study are trained during runtime, both for users using the software and for simulation studies. At each iteration of the active learning process, a new classifier is trained on the current set of labeled data, enabling adaptation to the dataset as labeling progresses. This approach is particularly suited to systematic reviews. In this context, which often involves frontier research, researchers are typically the first to construct the dataset as part of the review process. Therefore, no preexisting labeled data are available for training classifiers beforehand. The strength of the active learning approach lies in its ability to dynamically adapt to the data as new labels become available during the screening process.

The methodology for feature extractor training varies. Simpler feature extractors, such as TF-IDF and Doc2Vec, can generate embeddings without pretraining the weights. In contrast, transformer-based extractors, like MiniLM and Sentence-BERT, are pretrained and used without fine-tuning on a specific dataset, leveraging their general-purpose embeddings instead.

Hyperparameter optimization for the machine learning models was not performed in this study. Instead, optimized parameters were adopted directly from the original ASRe-

[2] exoscale.com.

[3] Software available at github.com/jteijema/asreview-simulation-project.

[4] kubernetes.io.

Table 1 Summary of datasets used in simulations: the table details all datasets, their topics, the total number of records, and the number of relevant records they contain

Dataset	Topic	Total number of records	Number of relevant records	% of relevant records	Simulations S1	Simulations S2
Appenzeller-Herzog_2019	Medicine	2.873	26	0.9%	338	92
Bos_2018	Medicine	4.878	10	0.2%	130	92
Brouwer_2019	Medicine, Psychology	38.114	62	0.2%	806	92
Chou_2003	Medicine	1.908	15	0.8%	195	92
Chou_2004	Medicine	1.630	9	0.6%	117	92
Donners_2021	Medicine	258	15	5.8%	195	92
Hall_2012	Computer science	8.793	104	1.2%	1.352	92
Jeyaraman_2020	Medicine	1.175	96	8.2%	1.248	92
Leenaars_2019	Chemistry, Medicine, Psychology	5.812	17	0.3%	221	92
Leenaars_2020	Medicine	7.216	583	8.1%	7.579	92
Meijboom_2021	Medicine	882	37	4.2%	481	92
Menon_2022	Medicine	975	74	7.6%	962	92
Moran_2021	Biology, Medicine	5.214	111	2.1%	1.443	92
Muthu_2021	Medicine	2.719	336	12.4%	4.368	92
Nelson_2002	Medicine	366	80	21.9%	1.040	92
Oud_2018	Medicine, Psychology	952	20	2.1%	260	92
Radjenovic_2013	Computer science	5.935	48	0.8%	624	92
Sep_2021	Psychology	271	40	14.8%	520	92
Smid_2020	Computer science, Mathematics	2.627	27	1.0%	351	92
Valk_2021	Medicine, Psychology	725	89	0.8%	1.157	92
van_de_Schoot_2018	Medicine, Psychology	4.544	38	12.3%	494	92
van_der_Waal_2022	Medicine	1.970	33	1.7%	429	92
van_Dis_2020	Medicine, Psychology	9.128	72	0.8%	936	92
Walker_2018	Biology, Medicine	48.375	762	1.6%	65*	92
Wassenaar_2017	Biology, Chemistry, Medicine	7.668	111	1.4%	1.443	92
Wolters_2018	Medicine	4.280	19	0.4%	247	92
Total		169.288	2.834		27.001	2.392

The table also includes the number of simulations run during study one and study two. *The Walker_2018 dataset has a reduced number of simulations

view software package where available. For models not included in the ASReview software package, developer-recommended settings were used.

The feature extractors utilized in the first simulation study are TF-IDF, Doc2Vec, MiniLM, and Sentence-BERT (all-mpnet-base-v2). The classifiers selected are Logistic Regression, Naive Bayes, Random Forest, and Support Vector Machine. The reasoning behind the selection of these models for the first simulations is based on frequently used models in other simulation studies. [55].

Note that Naive Bayes cannot use feature matrices containing negative vectors. As such, when feature extractors Doc2Vec, MiniLM, or all-mpnet-base-v2 are used, which produce negative vectors, Naive Bayes cannot be employed as a classifier. Therefore, the only viable pairing involving Naive Bayes is with TF-IDF, which limits the total number of models evaluated to 13.

1. TF-IDF + Logistic Regression, TF-IDF + Naive Bayes, TF-IDF + Random Forest, TF-IDF + Support Vector Machine, Doc2Vec + Logistic Regression, Doc2Vec + Random Forest, Doc2Vec + Support Vector Machine, MiniLM + Logistic Regression, MiniLM + Random Forest, MiniLM + Support Vector Machine, Sentence-BERT + Logistic Regression, Sentence-BERT + Random Forest, and Sentence-BERT + Support Vector Machine.

For the second study, the scope is expanded to include additional models. MiniLM is replaced with larger models that share a similar architecture. In total, the second study evaluates 13 feature extractors and 8 classifiers. However, certain limitations arise due to the nature of the embeddings and classifiers:

- Neural networks cannot process sparse embeddings because the input layer would need to be excessively wide (e.g., matching the vocabulary size, which exceeds 40,000 dimensions).
- Naive Bayes cannot handle negative embeddings.

Taking these constraints into account, the total number of combinations evaluated is calculated as follows:

- 6 classifiers compatible with all embeddings $\times$ 13 feature extractors = 78 combinations.
- 3 Naive Bayes classifiers compatible with positive embeddings only = 3 combinations.
- 11 neural network simulations compatible with specific embeddings = 11 combinations.

In total, this results in 92 unique combinations evaluated in the second study. A complete list of evaluated feature extractors is found in Table 2 and classifiers in Table 3.

Tables 2 and 3 categorize the processing speed of classifiers and feature extractors during their active learning cycles. Table 2 categorizes text embedding speeds as 'Fast' (seconds), 'Medium' (minutes), or 'Slow' (hours) and details the features each method extracts.

Basic methods focus on word occurrence and frequency, while word co-occurrence and context capture relationships between words. Semantic meaning emerges with sufficient context, reflecting deeper word significance. Syntax analyzes sentence structure, and attention dynamically weights word relevance, enabling focus on meaningful text parts. Language agnosticism is noted for methods applicable across multiple languages. Together, these features provide tools for interpreting text across various dimensions.

Table 3 similarly assigns 'Fast' to processing times of less than one second per cycle, 'Medium' to less than three seconds, and 'Slow' to more than three seconds. These values were calculated from the results of this study. This analysis is based on Table 1 from previous work [54], which presents similar statistics. Comparisons of identical classifiers and feature extractors between the two studies yield consistent results, reinforcing the reliability of these findings.

2.4 Prior knowledge

In our first simulation study, we conduct one simulation for each relevant record, together with 10 fixed irrelevant records, using the ARFI (All Relevant - Fixed Irrelevant) template from ASReview-Makita [56] to minimize inter-dataset variation. This standardized method ensures consistency across datasets and increases reproducibility for large-scale simulation studies. The total number of simulations conducted in study one is detailed in Table 1.

In the second study, we use a prior knowledge set consisting of five relevant records and ten irrelevant ones, selected at random but kept constant across models. Increasing prior knowledge helps reduce variability caused by differences in how informative individual prior records are.

Using the MultiModel template from Makita, we establish a consistent simulation template that encompasses every permutation of classifiers, feature extractors, and datasets, while ensuring that the prior knowledge remains unchanged for each dataset.

2.5 Evaluation

The study yields two sets of results. The direct performance results of the simulations and the meta-analyses focus on the variability present within these findings.

Table 2 Feature extractors in the second phase of the simulations, in approximate order of speed of the embedding calculations

Name	Description	Pretrained	Speed	Extracted features	Study
OneHot	Encodes text in a binary fashion, representing each word as a 1 (present) or 0 (absent) in the vector, ignoring word order and context	No	Fast	Word presence	2
TF-IDF	Enhances the OneHot approach by weighting words based on their frequency across records, helping to identify more informative words	No	Fast	Weighted word frequency	1 & 2
Doc2Vec (Vector Size 40)	Generates record embeddings in a 40-dimensional space, capturing the semantic meaning of words in context	No	Medium	Word Co-occurrence, Document-Level Context, Semantic Meaning	1 & 2
Doc2Vec (Vector Size 120)	Similar to its 40-dimensional counterpart but provides a more detailed representation by using 120 dimensions	No	Medium	Word Co-occurrence, Document-Level Context, Semantic Meaning	2
Doc2Vec Non-Negative	A 40 vector wide Doc2Vec embedding scaled between 0 and 1, allowing for use with Naive Bayes	No	Medium	Word Co-occurrence, Document-Level Context, Semantic Meaning	2
MiniLM [65]	A small, fast transformer model that retains much of the larger models' ability to capture nuanced language features	Yes	Medium	Syntax, Semantic Meaning, Context, Attention	1
distiluse-base-multilingual-cased-v2 [43]	A multilingual version of a sentence embedding model, designed to understand and represent a wide range of languages	Yes	Medium	Syntax, Language Independent Semantic Meaning, Context, Attention	2
FastText (Wiki-news-300d-1 M-subword) [5]	Employs subword information, allowing it to generate embeddings for words not seen during training by breaking them down into smaller units	Yes	Medium	Subword Information, Semantic Meaning	2
SpaCy[a]	Uses an NLP framework to provide embeddings that are particularly well-suited for syntactic analysis	Yes	Medium	Syntax, Word level semantic meaning	2
Word2Vec (Google-News-300)[b]	Generates word embeddings based on the context in which words appear, using a large corpus of Google News articles for training	Yes	Medium	Context, Semantic Meaning	2
mxbai-embed-large-v1 [27]	A pretrained sentence transformer made by mixedbread.ai. Especially targeted at creating embeddings for document retrieval	Yes	Slow	Syntax, Semantic Meaning, Context, Attention	2
all-mpnet-base-v2 (head-only)[c]	A transformer-based model optimized for sentence embeddings. Head-only [49]: any tokens outside of the token length are disregarded	Yes	Slow	Syntax, Semantic Meaning, Context, Attention	1 & 2
all-mpnet-base-v2 (hier. mean)	Hierarchical mean [49]: any tokens outside of the token length are embedded separately, and then averaged together into one embedding	Yes	Slow	Syntax, Semantic Meaning, Context, Attention	2
LaBSE [17]	A multilingual version of a sentence embedding model by Google	Yes	Slow	Syntax, Language Independent Semantic Meaning, Context, Attention	2

[a] https://huggingface.co/spacy/en_core_web_lg
[b] https://code.google.com/archive/p/word2vec/
[c] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

Table 3 Classifiers used in the second phase of the simulations, along with classifier description and approximate speed

Name	Description	Speed (current study)	Study
Naive Bayes	Based on Bayes' theorem with independence assumptions between features	Fast (0.02 s)	1 & 2
Logistic Regression	Predicts probabilities for binary outcomes based on linear combinations of features	Fast (0.19 s)	1 & 2
k-nearest neighbors	Classifies based on the majority label among the k-nearest samples	Fast (0.62 s)	2
Random Forest	Ensemble of decision trees, improving prediction accuracy through averaging	Medium (1.04 s)	1 & 2
AdaBoost	Boosts the performance of decision trees through a focus on incorrectly classified instances	Medium (2.45 s)	2
Neural Network	2 layered, fully connected. Learns complex patterns using two layers of interconnected nodes	Slow (5.07 s)	2
Support Vector Machine	Finds the hyperplane that best separates different classes in the feature space. Speed depends on dataset size	Slow (6.86 s)	1 & 2
XGBoost	Scalable optimized gradient boosting model	Slow (16.34 s)	2

2.5.1 Simulation Study 1 (reliability)

Simulation study one first assesses inter-dataset variability and then intra-dataset variability, via the performance results between datasets, models, and prior knowledge settings. To evaluate this variability, we use the Loss across datasets and models. Simulation study one also introduces APD heatmaps.

Recall curves Recall curves are a common method for visualizing simulation results, depicting the fraction of relevant records found versus the fraction of screened records. By stacking recall curves from multiple simulations, we can better assess active learning performance. In this study, $13 \times 25 = 325$ stacked recall curves from the first study are available on the persistent results website.

This paper highlights a selection of these stacked recall curves to illustrate examples of good, average, and poor performance, as well as the influence of prior knowledge. The figures include the *perfect performance* curve, representing the optimal scenario where all relevant records are identified before encountering any irrelevant ones. For datasets with a higher proportion of relevant records, this curve is naturally less steep.

Normalized Recall Regret The Normalized Recall Regret metric quantifies the overall performance of an active learning model by measuring how the recall curve is distributed between the optimal and the worst possible screening performance. Regret is commonly used to measure the difference between the actual performance and an ideal benchmark. Our contribution normalizes this value resulting in a value between 0 and 1.

Unlike point-based metrics like WSS or Recall, the Normalized Recall Regret provides a holistic assessment by evaluating the area under the recall curve (AUC) and can therefore be treated as a loss function. It is computed as the difference between the optimal AUC and the actual AUC, divided by the difference between the optimal AUC and the worst AUC.

- Optimal AUC: This is the area under a *perfect recall curve*, where relevant records are identified as early as possible. Mathematically, it is computed as

$$Nx \times Ny - \frac{Ny \times (Ny - 1)}{2}$$

  where Nx is the total number of records and Ny is the number of relevant records.
- Worst AUC: This represents the area under a worst-case recall curve, where all relevant records appear at the end of the screening process. This is calculated as

$$\frac{Ny \times (Ny + 1)}{2}$$

- Actual AUC: This is the area under the recall curve produced by the model during the screening process. It can be obtained by summing up the cumulative recall values for the labeled records.

$$\text{Normalized Recall Regret} = \frac{Ny \times \left(Nx - \frac{Ny-1}{2}\right) - \sum \text{Cumulative Recall}}{Ny \times (Nx - Ny)} \tag{1}$$

For simplicity and ease of interpretation, we refer to Normalized Recall Regret as *Loss* throughout the paper, as it quantifies the loss of recall. A loss value of 0 represents perfect performance, while a loss of 1 corresponds to the worst possible performance. A loss of 0.5 indicates that the model's performance is midway between these outer values. However, it does not necessarily imply random screening. While lower loss values generally indicate better performance, the interpretation of a specific loss score depends on how the recall is distributed throughout the screening process.

Variability We visualize the performance per dataset using the inter-dataset boxplot. Each box in this boxplot is a combined performance that includes all classifiers, feature extractors, and prior knowledge settings for a single dataset. This will provide a representation of the performance per dataset, giving insight into inter-dataset performance variation.

To visualize intra-dataset variability, the inter-dataset boxplot is split into 25 separate boxplots, one for each dataset (except Walker_2018, which has too few simulations for a fair comparison). Unlike the inter-dataset boxplot, where model performances were combined, these dataset-specific boxplots separate the 13 models into individual boxes. Within each boxplot, the performance range is determined solely by the selection of prior knowledge, allowing for an assessment of its impact on performance and a direct comparison between models.

Average Pair Distance Heatmap Recall curves do not provide insights into the specific discovery order of individual records. This limits their utility for cluster identification. Some stacked recall curves exhibit a distinct split shape due to unique discovery time groups, as illustrated in Fig. 2. Here, the only variable leading to these different curves is the specific record used as prior knowledge. We hypothesize that these distinctive paths could be attributed to clusters of records that are highly interrelated, yet show little correlation to the rest of the dataset. In cases where the prior knowledge exists in one of these clusters, the recall graph is likely to trace the unique, separate curve.

Using Time to Discovery, we generate visualizations in the form of heat maps. To create the heatmaps, we organize the data from the simulations into a three-dimensional array,

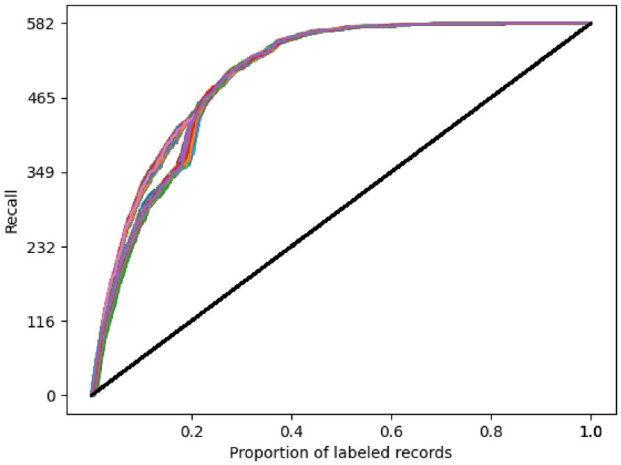


Fig. 2 Hypothetical example recall curve with a clear indication for clustering records

indexed by record ID, Time to Discovery TD, and simulation number n. We measure the distance between discovery for each pair of records across all simulations, with the distance being the number of records between the discovery of a first and second record, defined as

$$d_{i,j} = |\text{TD}_i - \text{TD}_j| \tag{2}$$

where TD_i and TD_j are the Time to Discovery of the first and second records, respectively, and $d_{i,j}$ represents the absolute difference between TD_i and TD_j.

We compute the log-transformed average across simulations to create the Average Pair Distance (APD) array using

$$\text{APD}_{i,j} = \ln\left(\frac{1}{N}\sum_{n=1}^{N} d_{i,j,n}\right)$$

In this array, both axes correspond to record IDs i and j for all IDs, and each cell value denotes the average distance in the discovery sequence between two specific records. $\text{APD}_{i,j}$ denotes the log-transformed average pair distance between records i and j, N is the total number of simulations, and $d_{i,j,n}$ is the distance between records i and j for simulation n.

The data are then visualized as a color-coded heatmap. The APD heatmap can be sorted by the average discovery sequence to present a clear view of potential 'hotspots'—clusters of records that appear closely related in their discovery sequence, but relatively independent from the rest of the dataset.

2.5.2 Simulation Study 2 (performance)

The second study evaluates the performance of 92 models across three key stages of the screening process. The analy-

sis begins with ranking models based on Loss. Next, overall performance is assessed by examining effectiveness at different points along the WSS curve. Starting performance is measured to identify models that excel at retrieving relevant records early in the process (first 100 records), which is particularly important for mitigating the cold start problem in active learning. Finally, Last-to-Find performance is analyzed to determine how well models identify the most challenging records in the final stages.

Overall Performance Overall performance is evaluated by plotting Work Saved over Sampling values in increments from $WSS@10\%$ to $WSS@100\%$, capturing each model's trajectory throughout the screening process. While previous analyses focused on Loss, this evaluation measures WSS at predefined thresholds for a more complete assessment.

A heatmap of the top 20 performing models is presented, followed by a graph showing WSS across all models. If even more granular data is desired, we present an interactive persistent website. [5] A sample of this interactive website is given.

Starting Performance To assess which models perform best in the early stages of screening, the number of relevant records identified within the first 100 screened records (including prior knowledge) is measured. This benchmark is based on expert analysis indicating that approximately 100 records can be screened within an hour. This approach helps us pinpoint the models that are most suitable for active learning sessions where time is limited. Effectively, we count the number of relevant records with a Time to Discovery (TD) [19] below 100 for each simulation.

A key challenge in active learning is the cold start problem, where models initially lack sufficient training data to make accurate relevance predictions. The starting performance metric helps assess how well different models overcome this limitation by effectively utilizing prior knowledge and quickly retrieving relevant records.

The theoretical lower bound for this metric is 5 relevant records, as each simulation begins with 5 relevant records provided as prior knowledge. The theoretical upper bound is given by:

$$\frac{1}{26}\sum_{d=1}^{26}(\min(r_d, (100-10))) = 50.84$$

where d represents the datasets and r_d represents the total number of relevant records for each dataset. Since 10 irrelevant records are included as prior knowledge, they are subtracted from the 100-record maximum. This calculation results in a theoretical upper limit of 51 records found.

Starting performance is particularly relevant for end-users under time constraints, such as those conducting rapid reviews or exploratory screening. While prior knowledge can help mitigate the cold start problem in active learning, its availability varies. When prior knowledge is limited, models with strong starting performance become important, as they are better able to retrieve relevant studies early in the process.

In the same time-sensitive scenarios, feature extractor computational speed also becomes a factor. Faster embeddings enable a quicker start, allowing for more screening within a limited time frame. Models that combine high starting performance with low computational cost are therefore the most suitable for time-constrained tasks.

Last-to-Find Performance The last-to-find section looks at the WSS@100% metric, which evaluates the work saved when *all* relevant records are found. This will identify models that are most effective at finding the final relevant records in the screening process. A bar chart ranks models based on WSS@100% performance values, with additional analysis on embedding calculation speed to illustrate the trade-offs between model complexity and computational cost.

For end-users of active learning in systematic review support, last-to-find performance is particularly relevant. This metric is critical because active learning aims to optimize the search process, stopping the screening when the stopping rule is triggered. Minimizing the number of missed records at this point is essential to avoid missing relevant records. The number of missed records depends, in effect, on the stopping rule and the model's performance in identifying the final relevant records [6]. If a model excels at finding easily identifiable papers but struggles with more difficult ones, it may lead to a gap in discovery, leading to an overestimation of dataset sparsity. This can occur if the model is overly focused on easily classified records and lacks robustness against noise, potentially causing relevant studies to remain undetected.

Beyond ranking models by their ability to find difficult records, the analysis also shows the trade-off between model performance and computational requirements. More complex models may improve retrieval but also require greater computational resources, which affects practical implementation.

2.6 Infrastructure

The simulations are performed using the simulation functionality of ASReview (simulation 1: v1.2, simulation 2: v1.5) [15], and facilitated through a Kubernetes cluster powered by 4 CPU-optimized processing nodes, amassing a total of 128 processing cores. The simulations are executed in a cloud infrastructure, and the results are stored in a persistent S3 bucket.

Our procedure is primarily a juggling act between managing the element sizes on the cluster and controlling simulation overhead. On one hand, packing all simulations into a sin-

[5] Live: https://jteijema.github.io/synergy-simulations-website/models.html, Persistent: http://doi.org/10.5281/zenodo.13169790.

gle pod[6] is inadvisable due to the inherent strength of Kubernetes being its ability to distribute tasks across pods. Conversely, segmenting all simulations into individual pods leads to unmanageable processing overhead by necessitating a distinct simulation environment for each job. Although workload queue systems presented a viable option, this introduces significant complexity and is, based on literature, opted against [45]. We establish a system wherein a single pod is set up for a single template run. In the first simulation study, the template provided to the pod is the ARFI template, and for the second study the MultiModel template.

Step-by-step We provide a detailed step-by-step guide of the simulation execution process:

- Job files[7] are created for each combination of variables under investigation.
- These job files are automatically dispatched to the cluster for processing.
- Each job is allocated a minimum of four CPU cores. If the cluster has sufficient memory resources available, the job is provided with the necessary processing power. If not, it is kept waiting in a jobs queue.
- The Docker image generates a Makita workflow specific to the dataset and models, as designated in the job.
- Following this, the cluster proceeds to run all simulations detailed in the Makita workflow using the ASReview simulation back-end and subsequently extracts the simulation metrics.
- These metrics are sent to an S3 storage bucket, providing a repository from which further analysis can be performed.

2.7 Availability of results and replicability

The visualization results of this simulation study are made available as a GitHub Page [52]. The webpage features recall curves for each simulation conducted during the study, covering all datasets. It also covers the per-model performance for each classifier and feature extractor used in this study. This totals 325 stacked recall graphs for simulation study one, representing the 27001 recall curves collectively, and 21 stacked model performance graphs for study two. These visuals allow any researcher to dissect and analyze our study results in depth. The persistent repository includes the necessary instructions to run the website locally using the Python built-in web server functionality, allowing further users to easily re-host the website and its functionality in the event the website is no longer available.

To promote research persistence and replicability, all raw results from this study are made available on DataverseNL [51]. By providing these resources, we aim to encourage further exploration and utilization of our findings in the active learning for systematic reviews community.

3 Results

3.1 Simulation Study 1

Figure 3 reflects the performance results using the Loss for every individual dataset. The difference in mean performance is high between datasets, ranging from just marginally better than random sampling of records to near-flawless results. The range of performance within a single dataset changes from one dataset to another and indicates how differently various machine learning models perform. For some datasets, the large range means that some models perform much better or worse than others, while a small range suggests that all models perform similarly.

Following this, we examine the performance of each model and dataset in the first simulation study. The intra-dataset variability presented in Fig. 4 showcases performance for all simulations in study 1. Here, a large box and whiskers indicate that the selected prior knowledge significantly impacts simulation performance, while a small range suggests a limited impact.

While the top-performing datasets (e.g., *Jall_2012, Leenaars_2019, others*) show very similar results across most models, there is a noticeable difference in performance between models for other datasets (e.g., *Appenzeller-Herzog_2019, Bos_2018, others*). This variability is more significant for some datasets than others. In most cases, the performance range for a single dataset and model is relatively narrow in the intra-dataset evaluation.

Figure 5 presents four examples from the 325 stacked recall curves generated in simulation study 1, along with two recall curves from study 2. The recall curves are compared to the ideal performance represented by the perfect line. The first, *Van de Schoot 2018 - Naive Bayes with TF-IDF*, demonstrates good performance. *Jeyaraman 2020 - Logistic Regression with TF-IDF* performs moderately, with a steep initial section but declining performance in later stages, suggesting a different model might be needed to optimize performance. *Moran 2021—Logistic Regression with TF-IDF* is an example of a dataset that is challenging to classify, likely due to factors beyond our experimental setup. Finally,

[6] A "pod" in Kubernetes refers to a single instance of a running process or application. It is the smallest and simplest unit of deployment in Kubernetes, encapsulating one or more containers and associated resources. Pods enable the grouping and management of containers within a Kubernetes cluster.

[7] A "job" in Kubernetes refers to a resource that manages the execution of a specific task or job within a cluster. It represents a one-time task that runs to completion, rather than continuously running like other Kubernetes resources. A job ensures that a pod completes the assigned task before considering the job as finished.

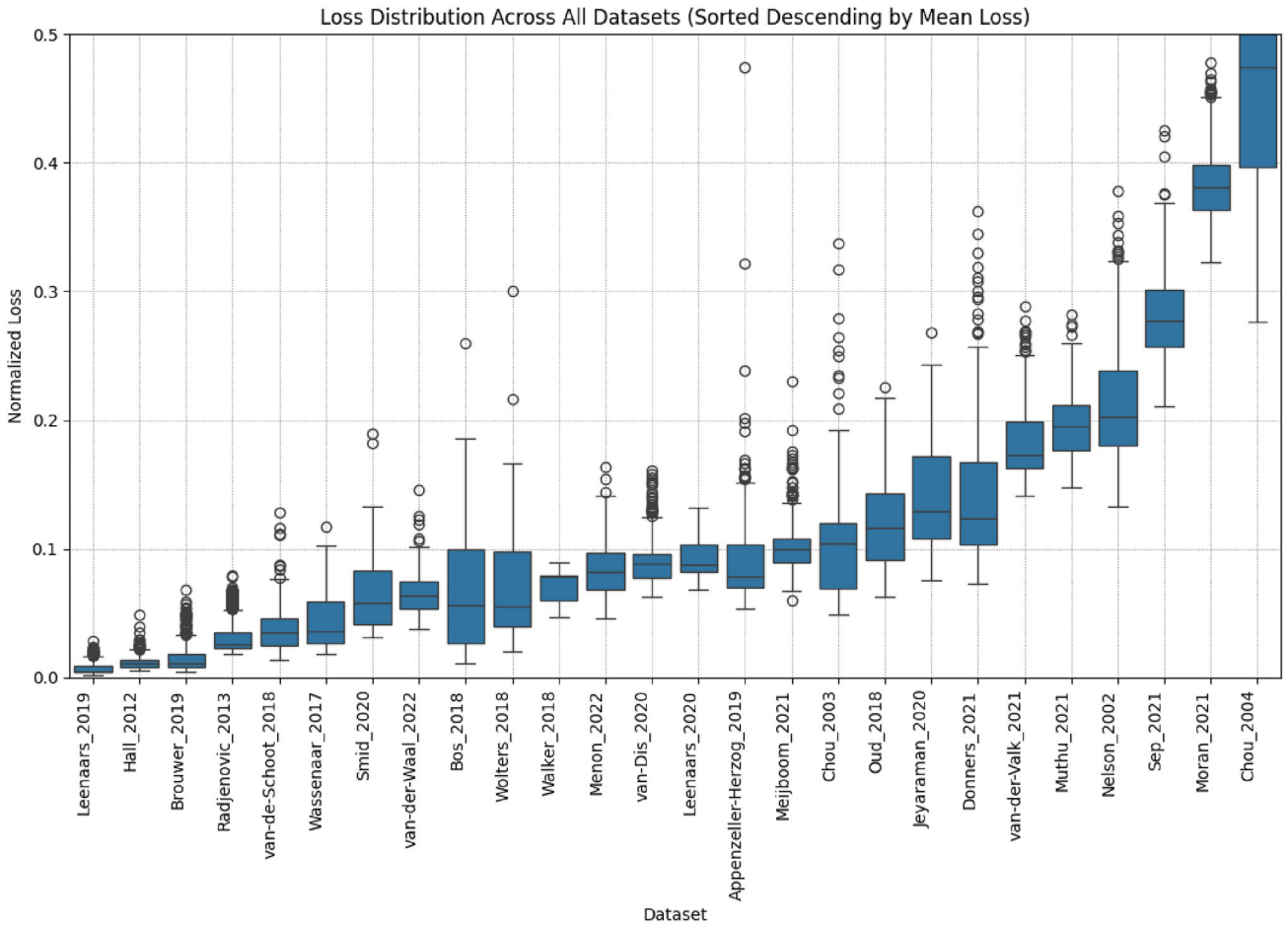


Fig. 3 Illustration of the variability in Loss inter-datasets. Datasets are ordered ascending based on their mean Loss values, from best to worst. The plot aims to highlight the dispersion in model performance when applied to different datasets

Oud 2018—Random Forest with all-mpnet-base-v2 shows a recall curve where the performance is more influenced by the selection of prior knowledge than on average. The two recall curves from the second simulation study include every simulation run on a dataset in this study, combining the performance across all models.

Average Pair Distance Heatmap We observe patterns in the recall curve of the "Jeyarama_2020_-m_logistic_-e_all-mpnet-base-v2" dataset that suggest the presence of potential clusters. We select five records from the recall curve that lie closely together but are distinct from the other curves, identifying them as a potential cluster. The recall curve (1), cluster subset (2), and corresponding APD heatmap with the main cluster of records (3) and subcluster (4) are shown in Fig. 6. When examining the documents from the subset, we find a significant overlap between the subcluster of identified records from the recall curve and the observed subcluster in the APD heatmap. These consistent observations across different datasets provide evidence for the existence and influence of clustering in record discovery.

3.2 Simulation Study 2

Ranking Model Performance Figure 7 presents the mean loss for each classifier–feature extractor combination, with the standard error represented by the black error bars. Lower loss values indicate better model performance. From this plot, it becomes clear some models perform better than others, but no decisive best model can be selected.

While some models perform better on average, the best-performing model varies significantly across datasets. As shown in Table 4, out of the 26 available datasets, 14 different classifier–feature extractor combinations achieved the lowest loss at least once. This suggests that no single model consistently outperforms others across all datasets. The most frequent top performer, "mxbai-embed-large-v1 transformer with Random Forest," was the best model in only 7 of 26 cases, while several other combinations appeared just once or twice.

Overall Performance Figure 8 illustrates the performance of all active learning models using Work Saved over Sampling. The plot shows that the WSS values generally increase

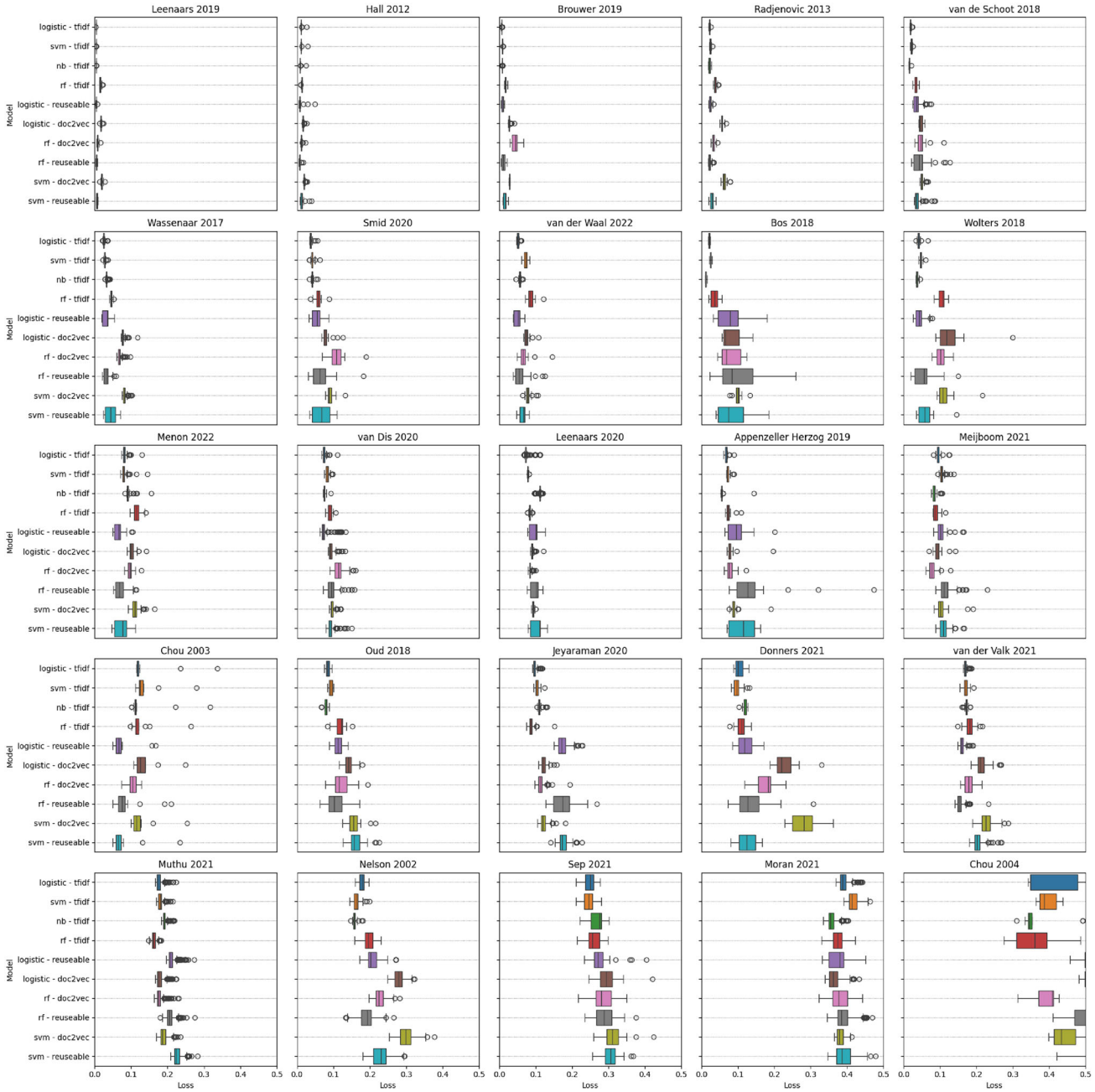
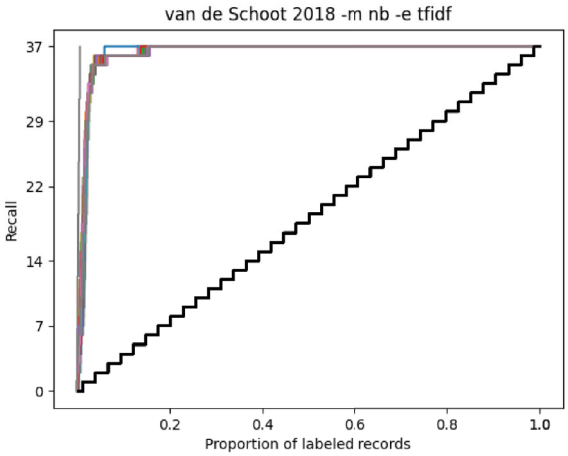


Fig. 4 The intra-dataset variability for each model and dataset combination, represented by Loss. Ordered based on the order of Fig. 3 and mean model loss

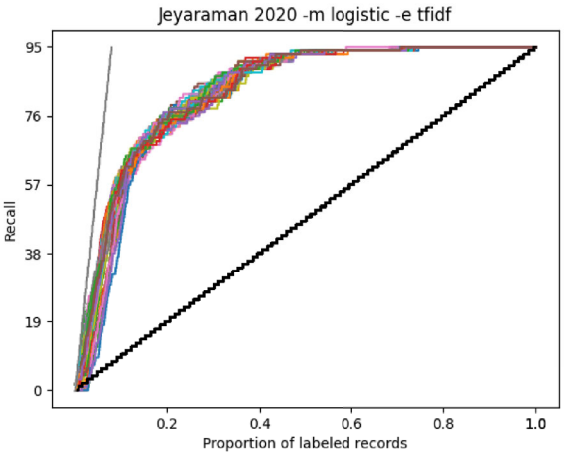
as the simulation progresses, demonstrating the effectiveness of active learning in reducing the number of records that need to be manually screened. However, the WSS values consistently decrease toward the end as the models search for the last-to-find relevant records. The results shown in this figure represent the average performance across all datasets included in the simulation. As this figure obscures specific model performances on individual datasets, this plot is not highly informative for detailed analysis. For a more granular view, separate figures that display the performance of individual models are available on the interactive persistent website. Two of such figures, namely the model performances images for mxbai and logistic regression,[8] are included in the image.

Starting Performance Figure 9 shows the average number of relevant records found after screening 100 records in the simulation. The hue indicates the type of feature extractor

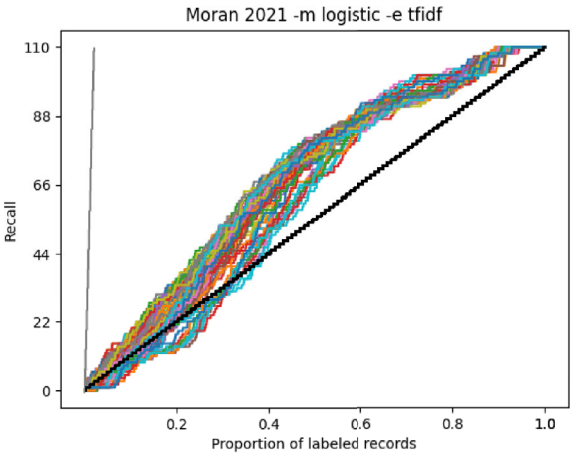
[8] https://jteijema.github.io/synergy-simulations-website/models.html#mxbai, https://jteijema.github.io/synergy-simulations-website/models.html#logistic.



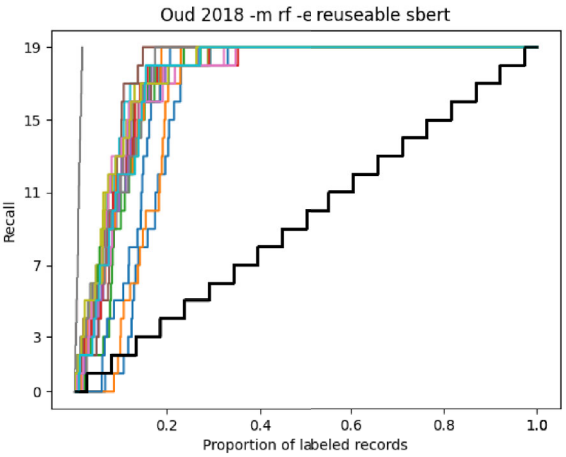
(a) Van de Schoot 2018 - Naive Bayes with TF-IDF



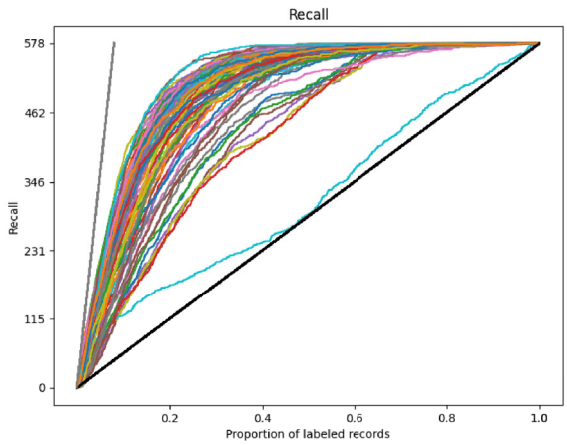
(b) Jeyaraman 2020 - Logistic Regression with TF-IDF



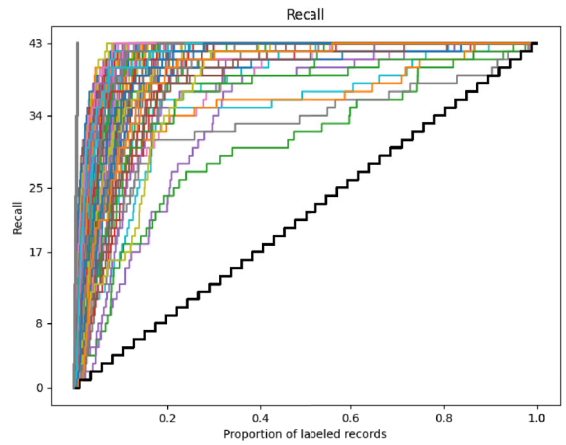
(c) Moran 2021 - Logistic Regression with TF-IDF



(d) Oud 2018 - Random Forest with all-mpnet-base-v2



(e) Leenaars 2020 - Every Model



(f) Radjenovic 2013 - Every Model

Fig. 5 A selection of recall curves from the over 29 thousand available. In panels a through d, each line represents a single simulation using a different record as prior knowledge. In panels e and f, each line represents a single simulation with a unique feature extractor–classifier combination

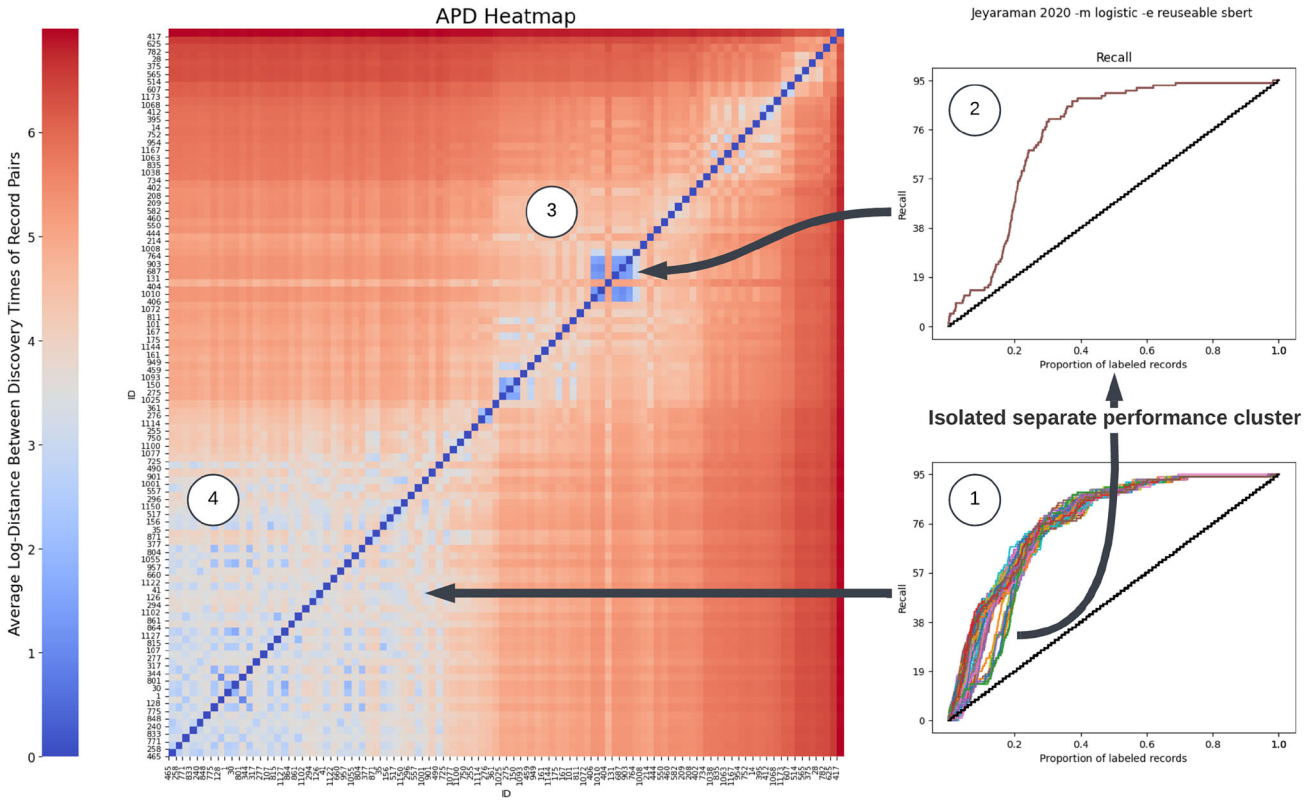


Fig. 6 Combined representation of the Average Pair Distance (APD) heatmap (3, 4) and recall curves for the dataset Jeyaraman 2020 with the Logistic Regression model and all-mpnet-base-v2 feature extraction. The heatmap uses color coding to indicate pair distances, where the pair distance is defined as the log difference in discovery time between two records. Red signifies larger distances (greater differences in discovery time), while blue represents smaller distances. Adjacent to the heatmap, the lower right corner displays all recall curves for the dataset (1), while the upper right corner shows a subset of recall curves corresponding to a specific cluster (2)

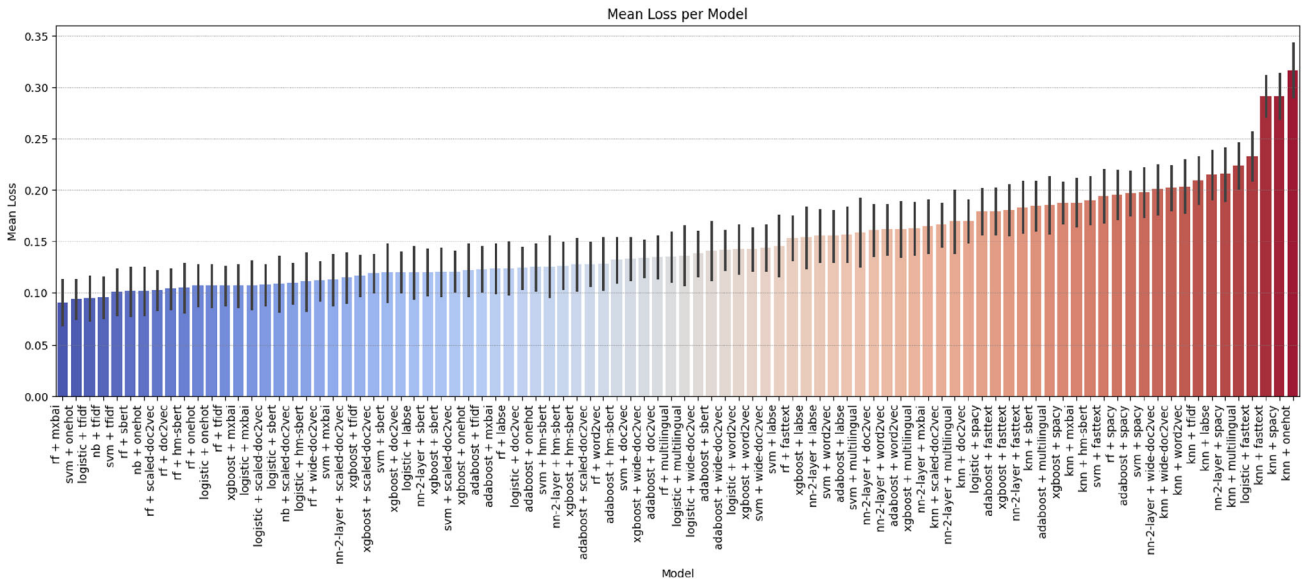


Fig. 7 Mean loss per model with standard error

Table 4 Summary of how often each model was the top performer for a dataset

Model combination	Top performer count
Random Forest with mxbai-embed-large-v1	7
Random Forest with all-mpnet-base-v2 (hierarchical mean)	3
Naive Bayes with TF-IDF	2
Naive Bayes with scaled Doc2Vec	2
XGBoost with OneHot	2
Neural Network (2-layer) with all-mpnet-base-v2 (hierarchical mean)	2
Neural Network (2-layer) with mxbai-embed-large-v1	1
XGBoost with mxbai-embed-large-v1	1
XGBoost with TF-IDF	1
Naive Bayes with OneHot	1
Random Forest with scaled Doc2Vec	1
Random Forest with OneHot	1
Logistic Regression with all-mpnet-base-v2 (hierarchical mean)	1
Logistic Regression with all-mpnet-base-v2	1

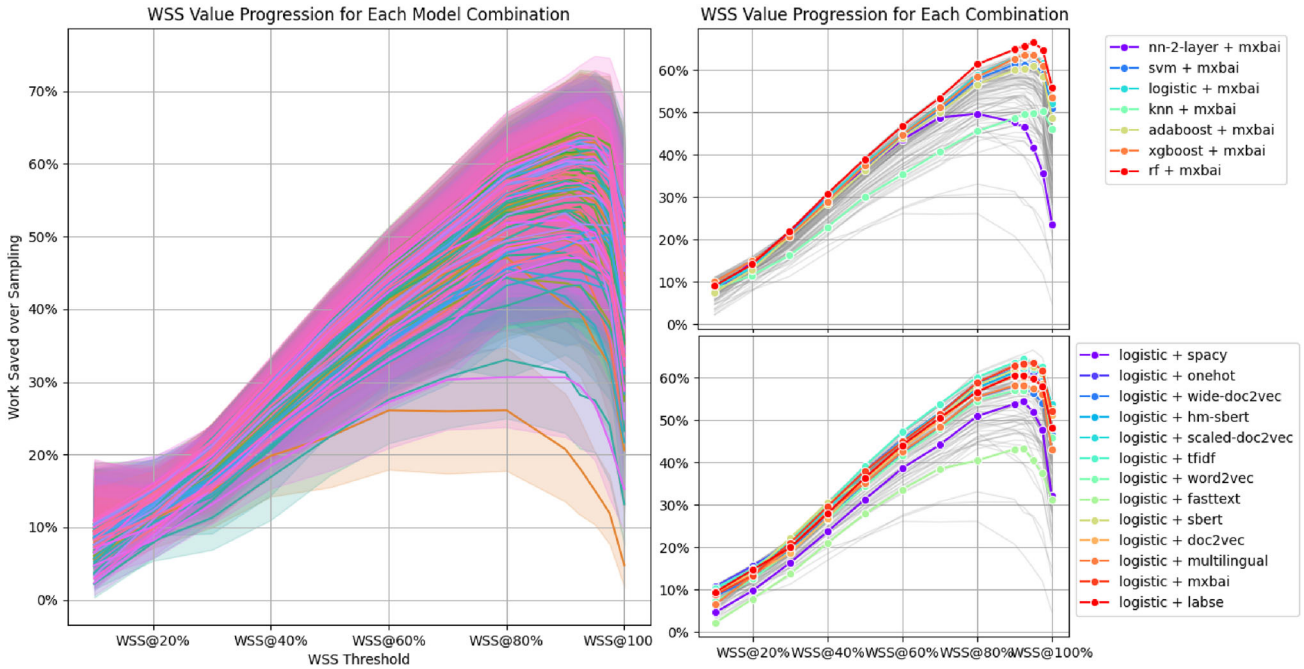


Fig. 8 Plot showing the performance of all active learning models, in Work Saved over Sampling. Also showing the derivative plots only using mxbai or logistic regression

used, in terms of calculation speed of the embedding computation. Along with the graph, Table 5 shows how often each model performs best in terms of finding the most relevant records within 100 screened. As the amount of relevant records found is less granular than the WSS score previously used, multiple models can tie for the top performance. Out of 92 models evaluated, 44 models reached the top-performing spot at least once. The table highlights models that achieved this distinction more than once.

Last-to-Find Performance In Fig. 10, the performance of each model is shown using work saved over sampling after all relevant records have been found (WSS@100%). The (flat) *random sampling* bar represents the performance without using active learning, therefore its work saved is 0. The speed of the used feature extractor is again given by the hue of the bar.

4 Discussion

This study aimed to analyze variability in simulation studies, both across datasets and within individual datasets, as well

Table 5 Summary of how often each model was the top performer for the first 100 records. Only those that performed best more than once are shown

Model combination	100-record top performer count
Random Forest with all-mpnet-base-v2 (hierarchical mean)	6
Random Forest with mxbai-embed-large-v1	5
Naive Bayes with OneHot	4
Support Vector Machine with OneHot	3
Naive Bayes with scaled Doc2Vec	2
Naive Bayes with TF-IDF	2
Random Forest with Doc2Vec	2
Random Forest with all-mpnet-base-v2 (head-only)	2
Logistic Regression with OneHot	2

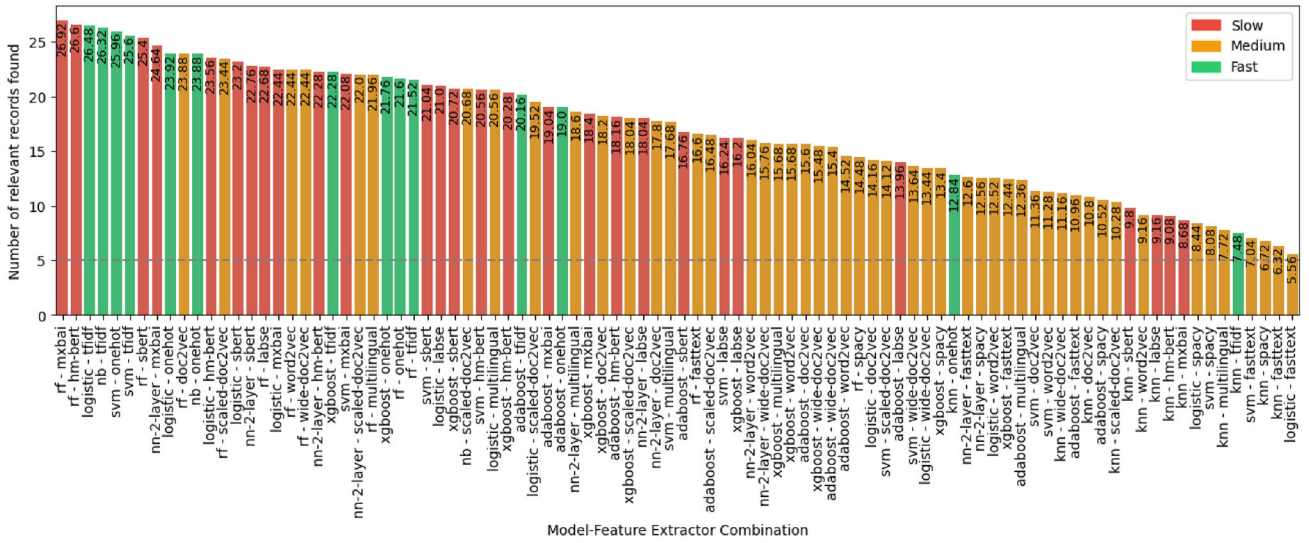


Fig. 9 The average number of relevant records found per model in the first 100 screened records. The lower limit is 5 records (prior knowledge). The color indicates the feature extractor embedding speed

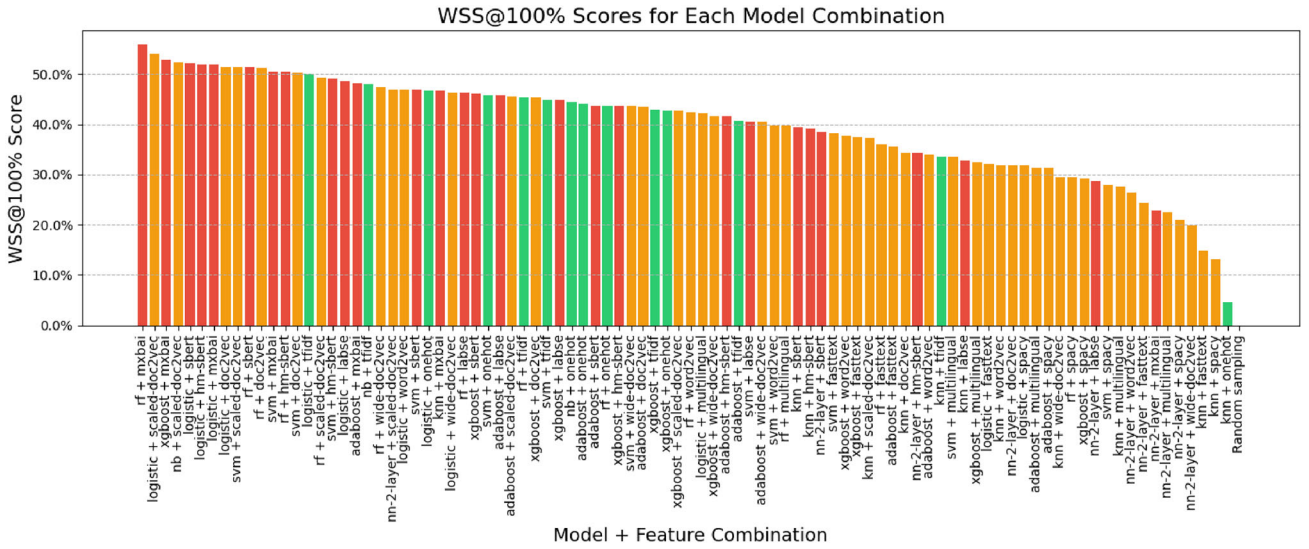


Fig. 10 The mean WSS@100% values for each model, measured after every record has been found, ordered by work saved over the sampling. The color indicates the computation embedding speed of the feature extractor

as to evaluate model performance across different screening phases. The simulation results confirm that active learning-based systematic review screening consistently outperforms random sample screening across all tested scenarios. Our work builds on the groundwork of many others, such as [59, 60], by conducting large-scale simulations that systematically assess 13 commonly used classifier–feature extraction combinations, along with 92 additional models. By incorporating a broader set of datasets and models than prior studies, our results provide a more comprehensive assessment of active learning in systematic review screening.

The performance differences between datasets are informative [18, 59]. Some datasets consistently yield low Loss scores [20, 28], suggesting easier classification, while others produce higher Loss scores with low variation between models, indicating a more challenging task [33, 47]. The most revealing datasets show significant differences in performance between models [16, 38]. While these datasets may provide insights into what model-specific advantages lead to better performance, those same datasets can also lead to bias when used exclusively for the comparison between models. Single or low-count dataset experiments risk overly optimistic or pessimistic outcomes due to dataset-specific biases; for robust conclusions, models should be validated across diverse datasets rather than limited samples.

Some datasets, such as Chou_2004 or Moran_2021, show negligible performance improvements compared to random screening, reflecting realistic situations where active learning may offer little benefit. However, the current study shows that even the weakest models generally detect enough of a pattern to slightly outperform random sampling. If no classifier, regardless of its complexity, manages to identify such a pattern in a given dataset, it strongly suggests that the dataset itself lacks any exploitable signal.

The influence of specific dataset characteristics, such as size or topic, remains unclear. No significant correlation was found between identifiable dataset features and model preference. With a larger number of datasets, stronger statistical power might reveal such correlations if they exist, but 26 is insufficient to draw conclusions. This result also suggests that commonly used dataset descriptors may not capture the factors influencing model effectiveness.

The study investigated whether an optimal model consistently outperforms others across systematic review datasets. Results indicate that some models significantly outperform others, including:

- Random Forest with mxbai-embed-large-v1
- Random Forest with all-mpnet-base-v2 (hierarchical mean)
- Naïve Bayes with TF-IDF
- Naïve Bayes with scaled Doc2Vec
- XGBoost with OneHot encoding
- Neural Network (2-layer) with all-mpnet-base-v2 (hierarchical mean)

No model achieved superior performance across all datasets. The model with the lowest average loss was still outperformed in 19 out of 26 datasets. These findings confirm that no classifier–feature combination is universally optimal. Instead, the optimal model varies depending on the specific dataset, highlighting the need for a flexible, adaptable approach when creating active learning software.

For general performance, both simple and complex models rank among the top performers, indicating that either type can excel depending on the dataset. For starting performance, the best models identified on average 27 out of a theoretical maximum of 51 relevant records, within the first 100 records screened. Compared to general performance, simpler models tend to rank higher in starting performance. More complex models, which rely on a larger number of parameters and features, typically require more data to generate effective screening orders [54]. Since starting performance is evaluated with limited data, simpler models tend to perform relatively better. However, some complex models also perform well, particularly those using pretrained feature extractors. Models like *hm-bert* and *mxbai* are pretrained, while others, like *Word2Vec* and *Doc2Vec*, are trained during simulation. Because the pretrained models have already been exposed to large amounts of data, the general rule that complex models require more data remains valid; these models have simply already encountered more data.

For last-to-find records [7, 19, 22], advanced models significantly outperform simpler models due to their ability to detect contextual and semantic relationships. Additionally, the "last few difficult documents" phenomenon is consistently observed across datasets. While the trends in starting and last-to-find performance align with expectations, it is valuable that our findings confirm them empirically.

Despite the widespread use of Support Vector Machines (SVMs) in active learning [2, 8, 42, 67], our results suggest that SVM can underperform compared to several other classifiers. This raises the question of whether its common use is justified or whether alternative classifiers should be considered more frequently. However, since SVM has shown strong results in prior studies further research should investigate whether better hyperparameter tuning or architectural adjustments could improve its performance.

The Normalized Recall Regret metric provides a broader assessment of active learning performance than point-based metrics like Work Saved over Sampling ($WSS@X\%$, [11]). While $WSS@X\%$ is widely used, there is no unified metric for evaluating systematic review screening models [39]. Unlike $WSS@X\%$, which evaluates performance at a fixed recall level, our metric captures the overall effectiveness of a model across the entire screening process. This makes it

useful for both general model evaluation and optimization. By treating regret as a Loss function, model performance can be compared more directly across datasets, supporting better model selection and pipeline optimization.

Recall graphs illustrate active learning performance but do not capture clustering dynamics. The current study introduces a new visualization method that reveals how records are discovered over time, providing deeper insight into dataset structure. Certain recall curves display distinct shapes. Visualizing the average discovery sequences in APD heatmaps highlights underlying clusters. The observed recall curve shapes correlated with these identified clusters. This indicates both the existence of clusters and the validity of the assumption that these can be identified by the shape of the recall curve.

The current study has limitations. While many model settings were explored, not all possible configurations were covered. Other variables such as different samplers and balancers were left unexplored. This study considers a model to be a combination of a feature extractor and a classifier, rather than each component separately. The interplay between the feature extractor and classifier within a model can influence overall performance for better or worse, and some combinations were not compatible. This analysis focuses exclusively on the results for each complete model combination to avoid the complexities and data incompleteness of dissecting the contributions of feature extractors and classifiers independently.

Another important aspect not covered is hyperparameter optimization. Many classifiers have tunable parameters that can significantly impact performance, and future research should explore whether certain models could achieve even better results with fine-tuned parameters.

Future studies should focus on identifying and analyzing more dataset characteristics, as this might lead to a better understanding of the relationship with model performance. Advanced feature extraction techniques that capture more complex lexical categories, alongside topic expert-driven dataset analysis, could help uncover underlying patterns. This might improve performance predictions and lead to more informed model selection strategies.

There is a need to enhance the performance of underperforming datasets, as they offer the most room for improvement. An open question is whether or not these datasets are performing as well as possible, or if yet undiscovered patterns exist that could further improve classification outcomes. Regardless of whether improvement is possible, stabilizing their performance across multiple simulations and reducing variability is crucial to ensure consistent and reliable results. Researchers should also consider contributing new screening data to the SYNERGY dataset to make it even more relevant and broad-based.

A promising direction is to treat the time-to-discovery of records as time-to-event data. This would open up survival analysis, a well-established branch of statistics, for use in our framework. Applying preexisting tools like the Kaplan–Meier estimator and accounting for censored data (where some records remain undiscovered) would allow for deeper statistical analyses to compare discovery rates across models and datasets, uncovering factors that influence efficiency.

In the evolution of systematic review automation, Large Language Models (LLMs) present a promising candidate for enhancing classification tasks. Given their capabilities in natural language understanding, LLMs have the potential to (semi-)automate the classification process. A hybrid approach, combining active learning strategies with LLM-driven classification, could offer a balanced solution. In the current active learning pipeline, some dataset segments go unscreened when the dataset appears sufficiently sparse, and the stopping rule is reached [6]. LLMs could address this gap by automatically screening these overlooked portions, while human experts focus on a subset of ambiguous or high-importance cases. This would facilitate a more efficient and reliable review process, enabling researchers to better manage large volumes of data. The integration of LLMs into the classification pipeline could therefore contribute significantly to the stability and accuracy of classification, warranting its exploration in future studies.

4.1 Recommendations

Our recommendations for end-users assume the use of an "average" dataset. Evaluating the unique characteristics of the datasets so they may lead to more tailored model recommendations falls beyond the scope of this study. We provide general guidance to support researchers in applying our approach to systematic reviews across various domains.

When screening in a **limited time** or for a limited number of records (in our experiments, 100 records or one hour of screening time), we recommend using a combination of either Naive Bayes or Logistic Regression with TF-IDF. This recommendation is based on the results shown in Table 2, Table 5 and Fig. 9. Although not the best performer, these models rank a very close third and fourth out of 92 models. The reason for this recommendation over the number one and two models is that these models are computationally lightweight. When computational time is a limiting factor, the time saved by using a faster model allows for screening more documents, leading to more data which leads to a larger performance boost than using a slower transformer to embed the dataset, given the similar performance levels. Another point to consider is the explainability factor. Even when computational time is not a concern, these less complex models offer a significantly more interpretable process compared to transformer models.

For the **last-to-find records**, screening often involves reviewing a larger number of documents. In such cases, more complex models tend to be more time-efficient. Notably, *Random Forest* with the *mxbai-embed-large-v1* embedding consistently performs significantly in identifying these difficult records, making it the recommended option. This recommendation also applies to switching models. If a model change is planned during the review process, this model is the preferred choice.

Finally, we recommend that platforms facilitating active learning remain open to the implementation of new machine learning algorithms as open-source projects. This study demonstrates that new approaches can improve performance. Given the rapid pace of these developments, the open extensibility of software supporting active learning is the most obvious and sustainable option.

4.1.1 User considerations

Users should consider the following when selecting their approach in performing a systematic review supported by active learning:

1. **Available time**: If time is limited and the focus is on screening efficiently, lightweight models such as Naive Bayes or Logistic Regression with TF-IDF are ideal. These models save computational time, enabling more documents to be screened in less time, without a significant drop in performance.
2. **Scope of the search**: If time is not a constraint and the goal is to either increase the scope of the search (e.g., retrieve more data from the database) or ensure high recall (e.g., increase the emphasis on finding all potentially relevant records), then complex models such as Random Forest with *mxbai-embed-large-v1* are recommended. This is especially suitable for users willing to invest more time in achieving exhaustive results.
3. **Optimal workflow**: To balance efficiency and thoroughness, users should follow the SAFE procedure [4]. This work provides a workflow that ensures an evidence-based strategy for determining when to stop screening and switch between models if needed. The selected models for this procedure are those recommended in the previous section.

4.2 Conclusion

Empirical evidence is the foundation of any scientific discipline, especially in data science and machine learning. In a rapidly progressing field like active learning for systematic reviews, which is fundamentally empirical, it is crucial to base the adoption of new methodologies on robust, large-scale evidence.

This large-scale simulation study evaluated active learning strategies for systematic reviews, testing whether an optimal model consistently outperforms others across multiple datasets. No such model was found. Instead, different models performed best at different stages of the review process and across different datasets.

These results highlight the importance of large-scale empirical evidence in systematic review simulations and set a higher standard for future research in this field.

Acknowledgements We would like to acknowledge the invaluable contributions of the software engineers at VSHN AG (https://www.vshn.ch/en/) for their technical assistance. Their expertise assisted us in the establishment and maintenance of our computational infrastructure. Their commitment to maintaining high-quality standards was instrumental in the successful execution of this large-scale simulation study. We extend our thanks for their professionalism, expertise, and efforts. We are grateful to the open science community for providing the datasets and tools that enabled us to conduct this extensive study. Special thanks to the creators and maintainers of the SYNERGY dataset, ASReview, and the ASReview-Makita software, whose efforts are making significant contributions to the advancement of active learning for systematic review methodologies. We thank the EU Open Clouds for Research Environments Project (OCRE) project, our funding body, who provided the necessary resources and support that enabled us to carry out this study.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Usage of generative AI and AI-assisted technologies in the writing process During the preparation of this work, the authors used *Open Source* Generative AI to increase language readability. After the use of this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

1. Adam, G.P., Wallace, B.C., Trikalinos, T.A.: Semi-automated tools for systematic searches. In: Methods in Molecular Biology. Springer US, pp. 17–40 (2021). https://doi.org/10.1007/978-1-0716-1566-9_2
2. Ambert, K.H., Cohen, A.M., Burns, G.A.P.C., et al.: Virk: an active learning-based system for bootstrapping knowledge base develop-

ment in the neurosciences. Front. Neuroinform. **7**(DEC), 38 (2013). https://doi.org/10.3389/fninf.2013.00038
3. Beller, E., Clark, J., Tsafnat, G., et al.: Making progress with the automation of systematic reviews: principles of the international collaboration for the automation of systematic reviews (ICASR). Syst. Rev. **7**(1), 77 (2018). https://doi.org/10.1186/s13643-018-0740-7
4. Boetje, J., van de Schoot, R.: The safe procedure: a practical stopping heuristic for active learning-based screening in systematic reviews and meta-analyses. Syst. Rev. **13**(1), 81 (2024)
5. Bojanowski, P., Grave, E., Joulin, A., et al.: Enriching word vectors with subword information. (2016). arXiv preprint arXiv:1607.04606
6. Bron, M.P., van der Heijden, P.G., Feelders, A.J., et al.: Using chao's estimator as a stopping criterion for technology-assisted review (2024). arXiv preprint arXiv:2404.01176
7. Byrne, F., Hofstee, L., Teijema, J., et al.: Impact of active learning model and prior knowledge on discovery time of elusive relevant papers: a simulation study. Syst. Rev. **13**(1), 175 (2024)
8. Carey, N., Harte, M., McCullagh, L.: The use of a text-mining screening tool for systematic review of treatments for relapsed/refractory diffuse large b-cell lymphoma. Int. J. Technol. Assess. Health Care **37**(S1), 2 (2021)
9. Carvallo, A., Parra, D., Lobel, H., et al.: Automatic document screening of medical literature using word and text embeddings in an active learning setting. SCIENTOMETRICS **125**(3), 3047–3084 (2020). https://doi.org/10.1007/s11192-020-03648-6
10. Chen, T., Guestrin, C.: Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016)
11. Cohen, A.M., Hersh, W.R., Peterson, K., et al.: Reducing workload in systematic review preparation using automated citation classification. J. Am. Med. Inform. Assoc. **13**(2), 206–219 (2006)
12. Cowie, K., Rahmatullah, A., Hardy, N., et al.: Web-based software tools for systematic literature review in medicine: systematic search and feature analysis. JMIR Med. Inform. **10**(5), e33219 (2022). https://doi.org/10.2196/33219
13. Cumpston, M., Li, T., Page, M.J., et al.: Updated guidance for trusted systematic reviews: a new edition of the cochrane handbook for systematic reviews of interventions. Cochrane Database Syst. Rev. **2019**(10) (2019)
14. De Bruin, J., Ma, Y., Ferdinands, G., et al.: SYNERGY - Open machine learning dataset on study selection in systematic reviews (2023). https://doi.org/10.34894/HE6NAQ
15. developers, A.L.: Asreview lab v1.2—a tool for AI-assisted systematic reviews (2023). https://doi.org/10.5281/zenodo.7821585
16. Donners, A.A., Rademaker, C.M., Bevers, L.A., et al.: Pharmacokinetics and associated efficacy of emicizumab in humans: a systematic review. Clin. Pharmacokinet. **60**(11), 1395–1406 (2021)
17. Feng, F., Yang, Y., Cer, D., et al.: Language-agnostic BERT sentence embedding (2020). CoRR arxiv:2007.01852
18. Ferdinands, G.: Ai-assisted systematic reviewing: selecting studies to compare Bayesian versus frequentist SEM for small sample sizes. Multivar. Behav. Res. **56**(1), 153–154 (2021)
19. Ferdinands, G., Schram, R., de Bruin, J., et al.: Performance of active learning models for screening prioritization in systematic reviews: a simulation study into the average time to discover relevant records. Syst. Rev. **12**(1), 100 (2023)
20. Hall, T., Beecham, S., Bowes, D., et al.: A systematic literature review on fault prediction performance in software engineering. IEEE Trans. Softw. Eng. **38**(6), 1276–1304 (2011)
21. Hamel, C., Kelly, S.E., Thavorn, K., et al.: An evaluation of Distillersr's machine learning-based prioritization tool for title/abstract screening—impact on reviewer-relevant outcomes. BMC Med. Res. Methodol. (2020). https://doi.org/10.1186/s12874-020-01129-1
22. Harmsen, W., de Groot, J., Harkema, A., et al.: Machine learning to optimize literature screening in medical guideline development. Syst. Rev. **13**(1), 177 (2024)
23. Hashimoto, K., Kontonatsios, G., Miwa, M., et al.: Topic detection using paragraph vectors to support active learning in systematic reviews. J. Biomed. Inform. **62**, 59–65 (2016). https://doi.org/10.1016/j.jbi.2016.06.001
24. Howard, B.E., Phillips, J., Tandon, A., et al.: Swift-active screener: accelerated document screening through active learning and integrated recall estimation. Environ. Int. **138**, 105623 (2020)
25. Jimenez, R.C., Lee, T., Rosillo, N., et al.: Machine learning computational tools to assist the performance of systematic reviews: a mapping review. BMC Med. Res. Methodol. **22**(1), 322 (2022). https://doi.org/10.1186/s12874-022-01805-4
26. Khalil, H., Ameen, D., Zarnegar, A.: Tools to support the automation of systematic reviews: a scoping review. J. Clin. Epidemiol. **144**, 22–42 (2022). https://doi.org/10.1016/j.jclinepi.2021.12.005
27. Lee, S., Shakir, A., Koenig, D., et al.: Open source strikes bread—new fluffy embeddings model (2024). https://www.mixedbread.ai/blog/mxbai-embed-large-v1
28. Leenaars, C.H., Drinkenburg, W.P., Nolten, C., et al.: Sleep and microdialysis: an experiment and a systematic review of histamine and several amino acids. J. Circadian Rhythms **17**, 7 (2019)
29. Lombaers, P., de Bruin, J., van de Schoot, R.: Reproducibility and data storage for active learning-aided systematic reviews. Appl. Sci. **14**(9), 3842 (2024). https://doi.org/10.3390/app14093842
30. Marshall, I.J., Wallace, B.C.: Toward systematic review automation: a practical guide to using machine learning tools in research synthesis. Syst. Rev. **8**(1), 163 (2019). https://doi.org/10.1186/s13643-019-1074-9
31. Mauricio, D., Gonzalez, N.: Optimizacion de estrategias de busquedas cientificas medicas utilizando tecnicas de inteligencia artificial. Ph.D. thesis (2021). https://doi.org/10.11144/javeriana.10554.58492
32. Molinari, A., Kanoulas, E.: Transferring knowledge between topics in systematic reviews. Intell. Syst. Appl. **16**, 200150 (2022)
33. Muthu, S., Ramakrishnan, E.: Fragility analysis of statistically significant outcomes of randomized control trials in spine surgery: a systematic review. Spine **46**(3), 198–208 (2021)
34. Norman, C.R., Leeflang, M.M.G., Porcher, R., et al.: Measuring the impact of screening automation on meta-analyses of diagnostic test accuracy. Syst. Rev. **8**(1), 243 (2019). https://doi.org/10.1186/s13643-019-1162-x
35. O'Connor, A.M., Tsafnat, G., Gilbert, S.B., et al.: Still moving toward automation of the systematic review process: a summary of discussions at the third meeting of the international collaboration for automation of systematic reviews (ICASR). Syst. Rev. **8**(1), 57 (2019). https://doi.org/10.1186/s13643-019-0975-y
36. Olorisade, B.K., De Quincey, E., Andras, P., et al.: A critical analysis of studies that address the use of text mining for citation screening in systematic reviews (2016). https://doi.org/10.1145/2915970.2915982
37. Olorisade, B.K., Brereton, P., Andras, P.: Reproducibility of studies on text mining for citation screening in systematic reviews: evaluation and checklist. J. Biomed. Inform. **73**, 1–13 (2017). https://doi.org/10.1016/j.jbi.2017.07.010
38. Oud, M., Arntz, A., Hermens, M.L., et al.: Specialized psychotherapies for adults with borderline personality disorder: a systematic review and meta-analysis. Aust. N. Z. J. Psychiatry **52**(10), 949–961 (2018)
39. O'Mara-Eves, A., Thomas, J., McNaught, J., et al.: Using text mining for study identification in systematic reviews: a systematic review of current approaches. Syst. Rev. **4**(1), 1–22 (2015)
40. Page, M.J., Moher, D., Bossuyt, P.M., et al.: Prisma 2020 explanation and elaboration: updated guidance and exemplars for reporting systematic reviews. BMJ **372**, n160 (2021)

41. Pellegrini, M., Marsili, F.: Evaluating software tools to conduct systematic reviews: a feature analysis and user survey. Form@re Open Journal per la formazione in rete **21**(2), 124–140 (2021). https://doi.org/10.36253/form-11343
42. Przybyła, P., Brockmeier, A.J., Kontonatsios, G., et al.: Prioritising references for systematic reviews with robotanalyst: a user study. Res. Synthesis Methods **9**(3), 470–488 (2018)
43. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (2019). http://arxiv.org/abs/1908.10084
44. Robledo, S., Aguirre, A.M.G., Hughes, M., et al.: "Hasta la vista, baby" – will machine learning terminate human literature reviews in entrepreneurship? J. Small Bus. Manag. 1–30 (2021). https://doi.org/10.1080/00472778.2021.1955125
45. Romanov, S., Siqueira, A.S., de Bruin, J., et al.: Optimizing ASReview simulations: a generic multiprocessing solution for 'light-data' and 'heavy-data' users. Data Intell. 1–19 (2024). https://doi.org/10.1162/dint_a_00244
46. Scott, A.M., Forbes, C., Clark, J., et al.: Systematic review automation tools improve efficiency but lack of knowledge impedes their adoption: a survey. J. Clin. Epidemiol. **138**, 80–94 (2021). https://doi.org/10.1016/j.jclinepi.2021.06.030
47. Sep, M.S., Vellinga, M., Sarabdjitsingh, R.A., et al.: The rodent object-in-context task: a systematic review and meta-analysis of important variables. PLoS ONE **16**(7), e0249102 (2021)
48. Settles, B.: Active learning literature survey. Tech. rep (2009)
49. Sun, C., Qiu, X., Xu, Y., et al.: How to fine-tune bert for text classification? (2020). arXiv:1905.05583
50. Teijema, J.J.: jteijema/asreview-simulation-project: v1.1.4 (2023). https://doi.org/10.5281/zenodo.7993561
51. Teijema, J.J.: Simulation data for: large-scale simulation study of active learning models for systematic reviews (2023). https://doi.org/10.34894/NYFSJY
52. Teijema, J.J.: jteijema/synergy-simulations-website: release on Zenodo (2024). https://doi.org/10.5281/zenodo.13169790
53. Teijema, J.J., van den Brand SAGE, Bagheri, A., et al.: Simulation-based active learning for systematic reviews: a systematic review of the literature-repository (2023). https://doi.org/10.17605/OSF.IO/T9HGM
54. Teijema, J.J., Hofstee, L., Brouwer, M., et al.: Active learning-based systematic reviewing using switching classification models: the case of the onset, maintenance, and relapse of depressive disorders. Front. Res. Metrics Anal. **8**, 1178181 (2023)
55. Teijema, J.J., Seuren, S., Anadria, D., et al.: Simulation-based active learning for systematic reviews: a scoping review of the literature. Preprint (2023)
56. Teijema, J.J., van de Schoot, R., Ferdinands, G., et al.: Makita-a workflow generator for large-scale and reproducible simulation studies mimicking text labeling. Softw. Impacts **21**, 100663 (2024)
57. Thomas, J., McNaught, J., Ananiadou, S.: Applications of text mining within systematic reviews. Res. Synth. Methods **2**(1), 1–14 (2011). https://doi.org/10.1002/jrsm.27
58. Thomas, J., Noel-Storr, A., Marshall, I., et al.: Living systematic reviews: 2. Combining human and machine effort. J. Clin. Epidemiol. **91**, 31–37 (2017)
59. van de Schoot, R., de Bruin, J., Schram, R., et al.: An open source machine learning framework for efficient and transparent systematic reviews. Nat. Mach. Intell. **3**(2), 125–133 (2021). https://doi.org/10.1038/s42256-020-00287-7
60. Van Dinter, R., Tekinerdogan, B., Catal, C.: Automation of systematic literature reviews: a systematic literature review. Inf. Softw. Technol. **136**, 106589 (2021). https://doi.org/10.1016/j.infsof.2021.106589
61. Wagner, G., Lukyanenko, R., Paré, G.: Artificial intelligence and the conduct of literature reviews. J. Inf. Technol. **37**(2), 209–226 (2022)
62. Wallace, B.: Abstrackr: Software for semi-automatic citation screening. https://effectivehealthcare.ahrq.gov/products/abstractr/abstract (2012)
63. Wallace, B.C., Trikalinos, T.A., Lau, J., et al.: Semi-automated screening of biomedical citations for systematic reviews. BMC Bioinform. **11**(1), 55 (2010). https://doi.org/10.1186/1471-2105-11-55
64. Wang, L.L., Lo, K.: Text mining approaches for dealing with the rapidly expanding literature on covid-19. Brief. Bioinform. **22**(2), 781–799 (2021)
65. Wang, W., Wei, F., Dong, L., et al.: Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers (2020). arXiv:2002.10957
66. Xuan, Q., Jiali, L., Yuning, W., et al.: Application of natural language processing in systematic reviews. Chin. J. Evid.-Based Med. **21**(6), 715–720 (2021). https://doi.org/10.7507/1672-2531.202012150
67. Yu, Z., Carver, J.C., Rothermel, G., et al.: Assessing expert system-assisted literature reviews with a case study. Expert Syst. Appl. **200**, 116958 (2022)
68. Zou, J., Kanoulas, E.: Towards question-based high-recall information retrieval. ACM Trans. Inf. Syst. **38**(3), 1–35 (2020). https://doi.org/10.1145/3388640