

USING CHAO'S ESTIMATOR AS A STOPPING CRITERION FOR TECHNOLOGY-ASSISTED REVIEW

M.P. Bron 

Utrecht University
Utrecht

The Netherlands National Police
The Hague
m.p.bron@uu.nl

P.G.M. van der Heijden 


Utrecht University
Utrecht

University of Southampton
Southampton
p.g.m.vanderheijden@uu.nl

A.J. Feelders 

Utrecht University
Utrecht

a.j.feelders@uu.nl

A.P.J.M. Siebes 

Utrecht University
Utrecht

a.p.j.m.siebes@uu.nl

2024-03-30

ABSTRACT

Technology-Assisted Review (TAR) aims to reduce the human effort required for screening processes such as abstract screening for systematic literature reviews. Human reviewers label documents as relevant or irrelevant during this process, while the system incrementally updates a prediction model based on the reviewers' previous decisions. After each model update, the system proposes new documents it deems relevant, to prioritize relevant documents over irrelevant ones. A stopping criterion is necessary to guide users in stopping the review process to minimize the number of missed relevant documents and the number of read irrelevant documents. In this paper, we propose and evaluate a new ensemble-based Active Learning strategy and a stopping criterion based on Chao's Population Size Estimator that estimates the prevalence of relevant documents in the dataset. Our simulation study demonstrates that this criterion performs well on several datasets and is compared to other methods presented in the literature.

1 Introduction

In extensive studies such as legal proceedings, criminal investigations, and systematic reviews in academia, researchers and investigators gather evidence or information by screening information found in large text databases or corpora. The task is to find all pieces of information relevant to the subject of the investigation. Often, the investigator starts by using (Boolean) search queries to pre-select documents from the database. Formulating these queries is not a trivial task, as it is the objective to capture (nearly) all relevant documents. Often, the resulting set of candidate documents that the researchers have to process is enormous, while the prevalence of relevant documents within these sets can be very low.

More formally, we have a dataset \mathcal{D} containing candidate documents. During the review process, these documents are read by domain experts and labeled as either *relevant* or *irrelevant*. Read documents are referred to as *labeled*, and we maintain two sets \mathcal{L}^+ and \mathcal{L}^- , for the relevant and irrelevant documents, respectively. The objective is to find all the remaining *unlabeled* relevant documents belonging to the set \mathcal{U}^+ .

The prevalence for systematic review corpora ranges from below 1 to 35 % [8], so most candidate documents are not relevant. Traditionally, the investigator reviewed each document in \mathcal{D} , resulting in a large amount of work. Recently, systems were proposed and built that potentially reduce the human effort needed by limiting the number of irrelevant documents shown to the reviewer [9, 15, 37, 43]. These systems use machine learning to recommend documents based on prior review decisions. We refer to these systems as *Technology-Assisted Review* (TAR) systems as coined in [15]. Many recent TAR systems use *Active Learning* (AL) to update the classifier after each or several review decisions iteratively. AL is a machine learning technique that is used to train a classifier with fewer labeled data points while retaining good performance. In this setting, the model can interactively query an oracle (i.e., the domain expert) to label data points with the desired output of the Machine Learning model (i.e., in the case of a classification task, the class of the data point). In our case, the model should predict the relevancy of each document. Because the model is frequently

retrained, it could reduce the number of instances that should be labeled by querying the most informative documents. Many TAR systems that use AL show the user the top- k unseen documents according to the classifier’s predictions. As the classifier is retrained frequently, the ranking is refined as well, reducing the number of documents that have to be screened.

In the case of abstract screening for systematic reviews, state-of-the-art systems can find all relevant documents after screening only 5 – 40 % of the corpus by using this general methodology [9, 37]. A caveat is that these systems lack reliable stopping criteria. Simulation studies show that we can reduce work if we know the prevalence *a priori*. However, the number of relevant documents is not known beforehand in a real-world situation. Because of this, an investigator may stop too early, resulting in the omission of important information. Conversely, stopping too late causes unnecessary effort.

In this work, we describe a method to determine the prevalence of relevant documents using *Population Size Estimation* (PSE) methods. These methods are used in official statistics and public health to estimate population size when only part of the population is observed. PSE methods are related to *Capture Mark Recapture models*, originating from ecology, where these models are used to estimate the population size of wildlife. In our case, we want to estimate the size of the set of relevant documents, i.e., the number of relevant documents. During systematic reviews, only a subset of the relevant documents is observed, that is, only the set of documents that the investigators read. A review can only be stopped if the reviewers believe no relevant documents have been missed or their recall target is met.

In this work, we investigate if PSE is a suitable technique for deciding when to terminate the TAR procedure. As our main contribution, we show how two versions of Chao’s Estimator [10], a PSE method, can be integrated into a TAR system and used within a stopping criterion for the review process. Furthermore, we present the results of an extensive simulation study in which we compared this stopping criterion to various other methods presented in the literature.

2 Related Work

The task of Technology Assisted Review (TAR) systems is to retrieve a significant number, if not all, of the relevant documents within a dataset \mathcal{D} . To achieve this, Active Learning is often continuously applied to the dataset, a process commonly known as Continuous Active Learning (CAL) in the literature. Continuous Active Learning aims to minimize the number of irrelevant documents while maximizing the number of retrieved relevant documents. Over the years, Cormack and Grossman have developed a variety of CAL methods, with the most prominent method being AutoTAR [14]. We describe the CAL procedure in Algorithm 1 .

Many CAL procedures require a set of seed documents provided by the reviewer. This set needs to contain at least one relevant document, but it does not need to be a document from \mathcal{D} ; it may also contain a topic description. Additionally, one example of an irrelevant document is needed. These are then used as inputs \mathcal{L}^+ and \mathcal{L}^- for Algorithm 1 . In each iteration, a classifier is fitted to the currently labeled information. Then, a batch containing the top- b documents is selected according to the ranking based on the classifier’s predictions. After the user has labeled each document in the batch, the process is repeated until \mathcal{U} is empty or a stopping criterion has been triggered. The procedure aims to optimize the retrieval of relevant documents by updating the classifier each iteration. Given a good stopping criterion, this procedure enables the user to minimize the workload while finding nearly all relevant documents.

AutoTAR is an adaptation of the CAL procedure [14]. Instead of just training on the labeled documents \mathcal{L}^+ , \mathcal{L}^- , it samples a set of documents from the unlabeled set \mathcal{U} , which are temporarily assumed to be irrelevant. This is a fair assumption, given the low prevalence of relevant documents in most datasets. Moreover, AutoTAR increases the batch size of each iteration by 10 %. ASReview [37] has a fixed batch size of 1 and uses dynamic resampling to deal with imbalanced training data to improve the classifier’s performance. In recent years, several other algorithms that also adhere to the CAL paradigm have been proposed, with each their own adjustments, have been proposed (inter alia [9, 43]).

2.1 Stopping Criteria

As described above, the CAL procedure leaves the question open of how to stop the review process (i.e., the STOPPINGCRITERION procedure, line 15 in Algorithm 1 , is not given). Researchers have recently developed various approaches to solve the stopping problem. In [30], the authors provide a taxonomy to classify the diverse range of stopping criteria. The authors classify the criteria according to two axes, namely *applicability to TAR methods* and the *guarantees* these methods offer. For applicability, each method can fall into one of the following three categories.

Interventional methods. This category of rules intervenes in the selection strategy of documents. Some interventional methods depend on a specific sampling methodology; others even deviate from the general CAL paradigm. These alterations enable the usage of specific statistical methods or tests. Some sampling strategies allow the use of an estimator to estimate the number of relevant documents within the corpus.

Algorithm 1 The Continuous Active Learning algorithm. The algorithm requires as parameters a dataset \mathcal{D} , an unlabeled set of documents \mathcal{U} , labeled documents \mathcal{L}^+ , \mathcal{L}^- , a classifier C , a batch size b . The Active Learning procedure selects new documents according to the relevance predictions of the classifier C , which are updated after each batch of labeling decisions.

```

1: procedure CAL( $\mathcal{D}, \mathcal{U}, \mathcal{L}^+, \mathcal{L}^-, C, b$ )
2:    $S \leftarrow \mathbf{false}$  ▷ Variable indicating whether CAL can be stopped
3:   while  $|\mathcal{U}| > 0$  and not  $S$  do
4:      $C.FIT(\mathcal{L}^+, \mathcal{L}^-)$ 
5:      $\mathcal{B} \leftarrow \text{SELECT}(\mathcal{U}, C, b)$ 
6:     for  $d \in \mathcal{B}$  do
7:        $y \leftarrow \text{REVIEW}(d)$  ▷ Performed by the human reviewer
8:       if  $y = \text{Relevant}$  then
9:          $\mathcal{L}^+ \leftarrow \mathcal{L}^+ \cup \{d\}$ 
10:      else
11:         $\mathcal{L}^- \leftarrow \mathcal{L}^- \cup \{d\}$ 
12:      end if
13:       $\mathcal{U} \leftarrow \mathcal{U} \setminus \{d\}$ 
14:    end for
15:     $S \leftarrow \text{STOPPINGCRITERION}(\mathcal{D}, \mathcal{U}, \mathcal{L}^+, \mathcal{L}^-, C, b)$ 
16:  end while
17:  return  $\mathcal{L}^+, \mathcal{L}^-$ 
18: end procedure
19: procedure SELECT( $\mathcal{U}, C, b$ )
20:   $\mathbf{P} \leftarrow C.PREDICT(\mathcal{U})$  ▷ Returns the relevance score for all  $d$  in  $\mathcal{U}$ 
21:   $\mathbf{R} \leftarrow \text{RANK}(\mathcal{U}, \mathbf{P})$ 
22:   $\mathcal{B} \leftarrow \text{HEAD}(\mathbf{R}, \mathcal{U}, b)$  ▷ Gets the top- $b$  documents
23:  return  $\mathcal{B}$ 
24: end procedure

```

Standoff methods. Methods that fall in this category can be used in combination with any TAR system, as these methods do not depend on any sampling strategy.

Hybrid methods. Some methods interleave or divide the process into phases, alternating the original method with periods in which another sampling strategy is used.

For the second axis, *guarantees*, each method falls into one of the following two categories.

Heuristic. Heuristic rules make a stopping decision based on general patterns observed in, for example, the recall statistics of the review. However, as these methods do not have a formal statistical grounding, they do not offer strong guarantees besides the results of the criterion on known datasets.

Certification rules. Certification rules provide a formal statistical guarantee that the stopping point has certain properties and/or that the rule provides a formal statistical estimate of effectiveness at the stopping point.

In the following sections, we list several criteria and list their classification according to this taxonomy.

2.1.1 Pragmatic Criteria (Standoff & Heuristics)

Pragmatic criteria are often based on the recall statistics of the process. A commonly used heuristic is to stop the TAR after k consecutive irrelevant document suggestions. Examples of values of k found in the literature are 50 and 200 [8]. However, studies have shown that this method often results in low recall or little work savings [8]. Moreover, this method frequently fails to meet the widely used recall target of 95%.

Another trivial method is to stop screening after reviewing half of the documents in \mathcal{D} [39]. A variant based on this is the criterion Rule2399 [16], which stops the procedure if the size of the set of read documents satisfies $|\mathcal{L}| \geq 1.2 \cdot |\mathcal{L}^+| + 2399$.

2.1.2 Baseline Inclusion Rate (Hybrid & Heuristic)

An example of a hybrid method is the Baseline Inclusion Rate [35]. In this approach, a random sample S of the dataset \mathcal{D} is taken initially, before the TAR procedure. All the documents in S are then reviewed. If S is large enough, the ratio $\frac{|S^+|}{|S|}$ should approximate the ratio $\frac{|\mathcal{D}^+|}{|\mathcal{D}|}$, where S^+ and \mathcal{D}^+ denote the subsets of relevant documents in S and \mathcal{D} ,

respectively. After labeling \mathcal{S} , the Active Learning phase is started. The process is then stopped when $|\mathcal{L}^+| \geq \frac{|\mathcal{S}^+|}{|\mathcal{S}|}|\mathcal{D}|$. However, due to sampling uncertainty, a large sample may be needed to obtain a good estimate of the prevalence. This process may consume a lot of time, and since the sample is random, there is no guarantee of saving any work during this period. Furthermore, this estimation is static, which may lead to a review with no work savings if the estimate is even slightly too high [8].

2.1.3 Target method (Hybrid & Heuristic)

In [15], the Target method is proposed, which aims for high recall and guarantees a recall of at least 70 %. This method divides the TAR process into two phases. First, the method randomly samples documents from the dataset until k relevant documents are found. The size of k depends on the user and the dataset, but the recommended value for $k = 10$, according to [15]. When these documents have been found, the system proceeds to the second phase by starting a standard TAR procedure, for example, AutoTAR. However, the documents’ judgments from the previous phase are not given to the TAR system of the second phase, so, from the machine learning perspective, the TAR procedure restarts from scratch. The stopping criterion is triggered when all k relevant documents from the first phase are rediscovered during the second phase.

2.1.4 Knee method (Standoff & Heuristic)

An already established heuristic is the *knee method* [15]. Most recall curves from TAR systems have an inflection point (which looks like a knee, hence the name). This method compares the slopes before and after the knee. When the ratio ρ between the two slopes becomes larger than a specific threshold or bound, the review process should be stopped. The slope ratio can be calculated as follows [15, 41]:

$$\rho(\mathcal{L}_t) = \frac{|\mathcal{L}_i^+|}{|\mathcal{L}_t|} \frac{|\mathcal{L}_t| - |\mathcal{L}_i|}{|\mathcal{L}_t^+| - |\mathcal{L}_i^+| + 1},$$

where t is the current iteration and i is the iteration that maximizes the perpendicular distance between the point $(|\mathcal{L}_i|, |\mathcal{L}_i^+|)$ and the line that goes through the origin $(0, 0)$ and the point $(t, |\mathcal{L}_t^+|)$. The bound is dynamic; it decreases as the number of relevant documents increases. In [15], the bound for iteration t is defined as $\text{bound}_t = 156 - \min(|\mathcal{L}_t^+|, 150)$. The Knee criterion is triggered when $\rho(\mathcal{L}_t) \geq \text{bound}_t$ and $|\mathcal{L}_t| \geq 1000$. The Knee method is designed with the batch size scheme of AutoTAR in mind. However, this method can easily be adjusted to work with any batching scheme [30], so this method is a standoff method.

2.1.5 Budget method (Standoff & Heuristic)

The Budget method [15] combines aspects of both the Knee method and the Target method, as well as observations on the recall statistics of TAR systems. This method can be stopped when either of the following two criteria are met:

1. The first criterion is based on the observation that after reading 75% of the dataset \mathcal{D} using random sampling, we can assume that we have found approximately 75% of the relevant documents. Additionally, we can assume that most TAR methods will improve upon random sampling. Therefore, the Budget method specifies that we can stop when the size of the read documents $|\mathcal{L}| \geq 0.75|\mathcal{D}|$.
2. The second criterion is based on the Knee method and Target method. We can observe that during phase one of the Target method, with target set size k , the number of randomly sampled documents would be $|\mathcal{L}_{\text{target}}| = k \frac{|\mathcal{D}|}{|\mathcal{D}^+|}$ for a dataset \mathcal{D} and its positive component \mathcal{D}^+ . At each iteration, we record the set of relevant documents as \mathcal{L}^+ . Since $\mathcal{L}^+ \subseteq \mathcal{D}^+$, $k \frac{|\mathcal{D}|}{|\mathcal{L}^+|}$ should be at least as large as the random sample size in the Target method. Combined with the slope ratio of the Knee method, the Budget method is triggered when $\rho(\mathcal{L}) \geq 6$ and $|\mathcal{L}| \geq k \frac{|\mathcal{D}|}{|\mathcal{L}^+|}$.

Just like the Knee method, the Budget method is designed with the batch size scheme of AutoTAR in mind, but can be adapted easily to work with any batching scheme [30].

2.1.6 AutoStop (Interventional & Certification)

Whereas the previously discussed methods are heuristics, the AutoStop method [31] is a topic-wise interventional certification method and is thus tightly coupled to its sampling strategy. The method aims to estimate the number of relevant documents \mathcal{D}^+ and use that estimate to decide when to stop by calculating the expected recall. AutoStop consists of four modules.

1. *Ranking module*. The procedure is similar to AutoTAR from the machine learning perspective. However, the main difference is on the inference side: the trained model is used to process all documents in \mathcal{D} instead of only the unlabeled set \mathcal{U} . The resulting posterior probabilities are then used to produce a ranking.

2. *Sampling module.* The sampling strategy is unique in that it makes major adjustments to the sampling procedure of the CAL procedure (see Algorithm 1). The normal CAL procedure (as in Algorithm 1) selects a batch with the top- k documents in \mathcal{U} . Instead, AutoStop makes a ranking on \mathcal{D} . Then, this ranking is used to sample, with replacement, from \mathcal{D} where each document d is weighted according to its rank so higher-ranked documents have a higher sampling probability.
3. *Estimation module.* This sampling strategy enables us to produce an unbiased estimate of the total number of relevant documents from the sampling history. An estimate can be calculated using either the Horvitz-Thompson estimator or Hansen-Hurwitz estimator. Besides a point estimate, we can also calculate its variance and, subsequently, a confidence interval of the estimate.
4. *Stopping module.* The stopping module offers two strategies: an *optimistic* stopping criterion calculates the expected recall according to the point estimate. The stopping criterion is triggered when the recall target that the user has set has been achieved. The other is a *conservative* criterion, which instead bases its decision on the upper bound of the estimate’s CI.

The estimation module estimates the size of \mathcal{D}^+ and the variance of this estimate. This feature allows the creation of stopping criteria with several recall targets and confidence levels. A downside of this method is its memory usage. The estimation module consumes approximately 20 GB for a set of 15000 documents, and memory consumption grows quadratically in terms of the dataset size [31]. Larger datasets must be divided into smaller manageable parts, and then the AutoStop procedure is performed for each part separately to overcome memory limitations. Unfortunately, this creates some additional overhead as knowledge is not shared between parts.

2.1.7 Quant (CI) Rule (Standoff & Certification)

The Quant rule [41] bases its estimate on the relevance probabilities of labeled documents and the unlabeled documents. Assuming that the model is well calibrated, that is, supposing we take a large sample of documents with a given probability p , then the prevalence of relevant documents is approximately p . Suppose at iteration t we have fitted a model with parameters θ_t , then we can estimate the number of relevant documents in the set of labeled documents as follows.

$$|\widehat{\mathcal{L}}_t^+| = \sum_{j \in \mathcal{L}_t} p(y = 1 | d_j; \theta_t) \quad .$$

For the unlabeled documents, a similar procedure is performed:

$$|\widehat{\mathcal{U}}_t^+| = \sum_{j \in \mathcal{U}_t} p(y = 1 | d_j; \theta_t) \quad .$$

Then, the recall can be estimated as follows:

$$\hat{R}_t = \frac{|\widehat{\mathcal{L}}_t^+|}{|\widehat{\mathcal{L}}_t^+| + |\widehat{\mathcal{U}}_t^+|} \quad .$$

When $\hat{R}_t \geq R_{\text{tar}}$, where R_{tar} denotes the target recall, then the stopping criterion is triggered. Besides this point estimate, [41] provide a method to calculate the variance of this estimator, which in turn can be used to calculate a 95 % confidence interval (± 2 standard deviations). Given the size of \mathcal{L}^+ , the recall estimate \hat{R} can be used to produce an estimate of the size of $|\widehat{\mathcal{D}}^+|$. These estimates can then be used for a conservative and optimistic stopping criterion in a similar fashion as in AutoStop.

2.1.8 Hypergeometric method (Hybrid, Standoff & Certification)

In [8], a statistical stopping criterion for abstract screening for systematic reviews based on statistical testing is introduced. In this work, both a hybrid and standoff method are proposed. Their method is centered around the hypergeometric distribution. The authors assert that the number of missing relevant papers contained in a random sample follows the hypergeometric distribution.

The standoff version of their method works as follows. After each iteration, the labeled set is divided into two parts around a pivot iteration i . The method then calculates the probability that the current recall equals or exceeds the target recall. The recall target is specified as τ_{tar} . Then, we iterate over each pivot i and calculate the probability

$$p_i = P(X \leq \mathcal{L}^+ - \mathcal{L}_i^+) \quad ,$$

where

$$X \sim \text{Hypergeometric}(\mathcal{D} - \mathcal{L}_i^+, K_{\text{tar}}, \mathcal{L} - \mathcal{L}_i) \quad ,$$

and

$$K_{\text{tar}} = \left\lfloor \frac{\mathcal{L}^+}{\tau_{\text{tar}}} - \mathcal{L}_i^+ + 1 \right\rfloor .$$

After iterating over all pivots, the TAR process is stopped if there is an iteration i where $\min(p_i) < \alpha$ for some confidence level α . In [8] $\alpha = 0.05$ is chosen. We will refer to this method as *CMH-Standoff*.

Their hybrid (Ranked Quasi Sampling strategy in [8]) method consists of two phases. In the first phase, the TAR procedure is followed as normal until the standoff criterion, as described above, is triggered (here $\alpha = 0.5$). Random sampling will then be used for the remaining part of the screening. Like the standoff version, the test is repeated each iteration; however, with one alteration, the pivot iteration i is fixed on the iteration in which the standoff rule was triggered. This version only misses the target recall of 95 % or above in a few scenarios ([8] reports 0.59 % of the runs on several datasets with different seed documents). However, this robustness comes at a cost, as their method relies on random sampling. The result is that the average work reduction over random sampling (WSS, see Equation 8) achieved with their stopping criterion is only 17 %. We will refer to this method as *CMH-Hybrid*.

3 Methodology

In our work, we propose a novel stopping criterion for Technology-Assisted Review (TAR) that uses a Population Size Estimator to estimate the size of the set of relevant documents \mathcal{D}^+ . To be more precise, we adopt Chao’s moment estimator [10] and a Poisson Regression version of this estimator [32]. The sampling procedure and estimator are intertwined, that is, without this specific sampling procedure, these estimators cannot be used. First, we describe Population Size estimators in the context of systematic search tasks and TAR. Then, we describe our sampling procedure, followed by an overview of the aforementioned estimators. This is followed by an overview of the stopping criteria that use the estimates. This section concludes with a more detailed description of the classification algorithms and the Active Learning procedure.

3.1 Population Size Estimation for Technology-Assisted Review

Population Size Estimation (PSE) techniques are commonly used to estimate the total size of only partially observed populations, such as animal and human populations [2]. Besides calculating the size of human and animal populations, PSE methods were also applied to estimate the number of other partially observed sets of objects or phenomena, such as the number of hidden faults within a software package [13]. PSE may involve linking multiple lists recording observations of individuals or the number of times an individual is observed. These records can then be used to determine the capture probabilities of individuals, which in turn can be used to estimate the size of the entire population. In the case of TAR, the estimand is the number of relevant documents within a dataset, that is, the size of \mathcal{D}^+ . The set \mathcal{D}^+ consists of two parts, the set that has been found by the user (\mathcal{L}^+) and the set of documents that have yet eluded the search process (\mathcal{U}^+). More formally, $\mathcal{D}^+ = \mathcal{L}^+ \cup \mathcal{U}^+$. The user can stop the process once the estimate $|\widehat{\mathcal{D}^+}|$ approaches $|\mathcal{L}^+|$ and consequently the $|\widehat{\mathcal{U}^+}|$ becomes low enough.

3.1.1 PSE for Search Tasks

The use of PSE techniques for search tasks has been explored previously in the literature [27, 33, 36, 40]. For example, [40] presented a method employing a PSE to estimate the number of omissions from a systematic literature review. The basic outline of this approach is as follows: multiple reviewers conduct independent searches for documents relevant to a specific topic and decide for each document they review if it is relevant to this topic. For simplicity, we assume that the reviewers are unanimous in deciding the relevancy of each encountered document i . The sets of relevant documents \mathcal{L}_j^+ for each reviewer j may differ, as the search skills of individual reviewers vary. In the end, upon completing their tasks, the reviewers link their sets \mathcal{L}_j^+ , to identify for each document i the reviewers by which it was discovered.

The linking procedure is performed as follows (we are using the notation from [10]). Suppose we have a review committee \mathcal{C} consisting of $C = |\mathcal{C}|$ reviewers. We represent the result of the search process as a $N \times C$ matrix $\mathbf{X} = (X_{ij})$ where N is the size of the set of *all* relevant documents (that is, both the documents that were found and the documents that eluded the reviewers) and $C = |\mathcal{C}|$ is the size of the committee. Then, we specify the elements of \mathbf{X} as

$$X_{ij} = I [\text{document } i \text{ is present in } \mathcal{L}^+ \text{ of reviewer } \mathcal{C}_j] \quad ,$$

where $I[A]$ is an indicator function: $I[A] = 1$ if the event A occurs and 0 otherwise. This results in a matrix in which each row represents a document and each column a reviewer. The cells then contain a 1 if the document i has been

found by reviewer \mathcal{C}_j and 0 otherwise. Then,

$$n = \sum_{i=1}^N I \left[\sum_{j=1}^C X_{ij} \geq 1 \right],$$

denotes the number of distinct relevant documents that have been found by at least one reviewer, in other words, the number of relevant documents that has been found by the committee as a whole. Furthermore, we specify the frequency statistic

$$f_k = \sum_{i=1}^N I \left[\sum_{j=1}^C X_{ij} = k \right], \quad k = 0, 1, \dots, C,$$

which denotes the number of documents that have been found by exactly k reviewers. Of course, matrix \mathbf{X} is not fully observed: only the n rows for the documents that were found at least once are observed and as we do not know how many documents have been missed by all reviewers, we do not know the size of the N dimension of matrix \mathbf{X} . Consequently, we do not know how many documents have frequency statistic f_0 . Then, the estimand can be defined as

$$\hat{N} = n + \hat{f}_0,$$

where \hat{N} denotes the estimate for the total number of relevant documents, of which \hat{f}_0 are unobserved. In Section 3.2, we will further discuss the models.

3.1.2 PSE without multiple reviewers

A limitation of the approach sketched above is the need for multiple human reviewers in the review procedure to enable the estimation of the number of omitted relevant documents. In this work, we propose to adapt the sampling strategy to allow us to estimate using PSE without relying on multiple reviewers. We employ an ensemble of Active Learning methods that individually rank and propose documents. For each method, we maintain a registration list containing the identifiers of documents identified by each method. Our approach draws on the Query-By-Committee and Query-By-Bagging paradigms used in Active Learning, as introduced by [34]. In Query-By-Committee, an ensemble $\mathcal{C} = \mathcal{C}_1, \dots, \mathcal{C}_n$ is constructed, comprising multiple classifiers of different classification algorithms, such as Multinomial Naïve Bayes, Logistic Regression, and Random Forest. Since each classifier has a distinct decision function, each will likely produce a unique ranking. In Query-By-Bagging, each classifier is presented with a unique subset of the labeled corpus. Our approach combines both methods; we incorporate a diverse range of classification algorithms and independent training sets for each classifier.

The canonical Query-By-Committee method typically pools classifier decisions using a query strategy such as *Vote Entropy*. This method involves each classifier in the ensemble voting for its prediction on an unlabeled instance, and the instance with the most disagreement among committee members is selected for human review. However, in our approach, each committee member has its own query strategy and functions as an independent TAR system. We do not use an overarching query strategy that combines the results of these methods; instead, each proposed document is selected by choosing one of the members in a round-robin or random fashion. The selected committee member then presents an instance for review according to its individual query strategy.

In this approach, it is possible for a member \mathcal{C}_i to propose an instance d_k for review in iteration t , which was already proposed by \mathcal{C}_j in a previous iteration t' . To handle this situation, our method ensures that the label for d_k is given to the member \mathcal{C}_i . This is illustrated in Figure 1, where document 22 is first proposed by \mathcal{C}_2 and then in the next iteration by \mathcal{C}_1 . The labeling decision is transferred to \mathcal{C}_2 , and from the user’s perspective, there is no difference. The process continues by selecting one of the committee members again until a document is proposed that has not been proposed by any of the other methods.

After each label decision, the estimation module generates a matrix \mathbf{X} from the system as follows.

$$X_{ij} = I [\text{document } i \text{ is present in } \mathcal{L}^+ \text{ from committee member } \mathcal{C}_j].$$

Then, a PSE model uses \mathbf{X} to estimate the number of omissions. The stopping module compares the estimate and its confidence interval to the current recall statistics and decides if the TAR procedure can be terminated. If not, this procedure is repeated by retraining the models and sampling new documents from the updated rankings.

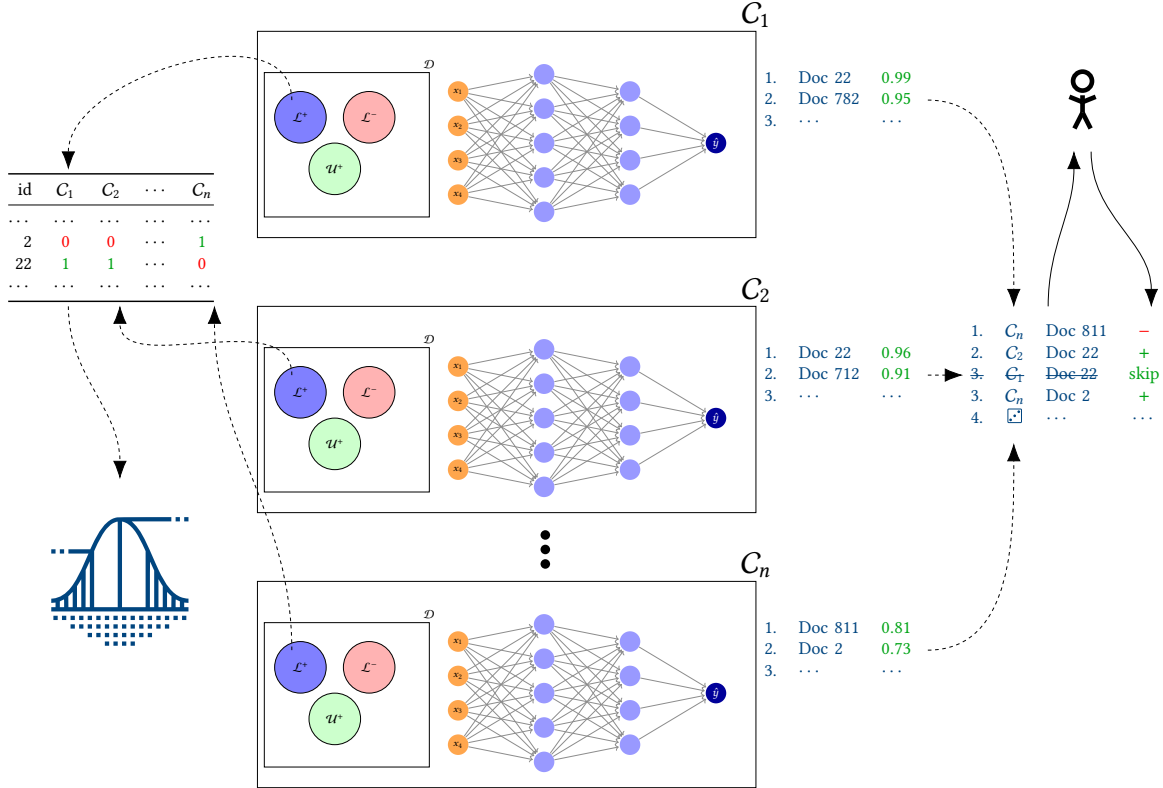


Figure 1: This figure shows an architectural overview of our method. The Active Learning module consists of several committee members $\{C_1, \dots, C_n\}$, with each its own labeled and unlabeled state. Each of the members can have a Machine Learning Model (for illustrative purposes represented as an Artificial Neural Network). The rankings of each of the members are combined by going through each member in a round-robin or random fashion and selecting the top of the stack. The estimation module can query the labeled states of each of the member to construct a contingency table and fit a PSE model.

Table 1: Frequency statistics for the example run in Figure 2.

f_0	f_1	f_2	f_3	f_4	f_5	n
?	40	33	17	2	0	92

3.2 Chao's Moment Estimator

In our work, we use Chao's moment estimator [10] and a Poisson Regression adaptation by Rivest and Baillargeon [32]. While several PSE methods use the full matrix \mathbf{X} to model \hat{N} , both models only use the frequency statistics f_k . In the following sections, we will introduce Chao's estimator and the Poisson regression adaptation through an example.

We execute the procedure described in Section 3.1.2 on the dataset collected for a systematic review [29]. This dataset is part of the test collection of [37]. This dataset contains 2481 documents, of which 120 are relevant, so the ground truth for $N = 120$. We simulate 500 iterations (i.e., 500 review decisions) using our methodology. At iteration $t = 500$, we have the following frequency statistics, displayed in Table 1. We omit matrix \mathbf{X} for practical reasons.

At iteration $t = 500$, the number of retrieved relevant documents $n = 92$. Using the frequency statistics, we can use Chao's moment estimator to obtain a point estimate of the total number of omitted relevant documents \hat{f}_0 and the resulting total number of relevant documents \hat{N} . Chao's estimator is formulated as follows,

$$\hat{N} = n + \hat{f}_0, \quad \hat{f}_0 = \begin{cases} \frac{f_1^2}{2f_2} & \text{if } f_2 > 0 \\ \frac{f_1(f_1-1)}{2(f_2+1)} & \text{if } f_2 = 0 \end{cases} . \quad (1)$$

For the derivation of Equation 1 for the case $f_2 > 0$, we refer to [10 - 786]. The case for $f_2 = 0$ is needed as the upper formula cannot be calculated when $f_2 = 0$ due to a division by zero. In [11], an adjusted formula is given for when

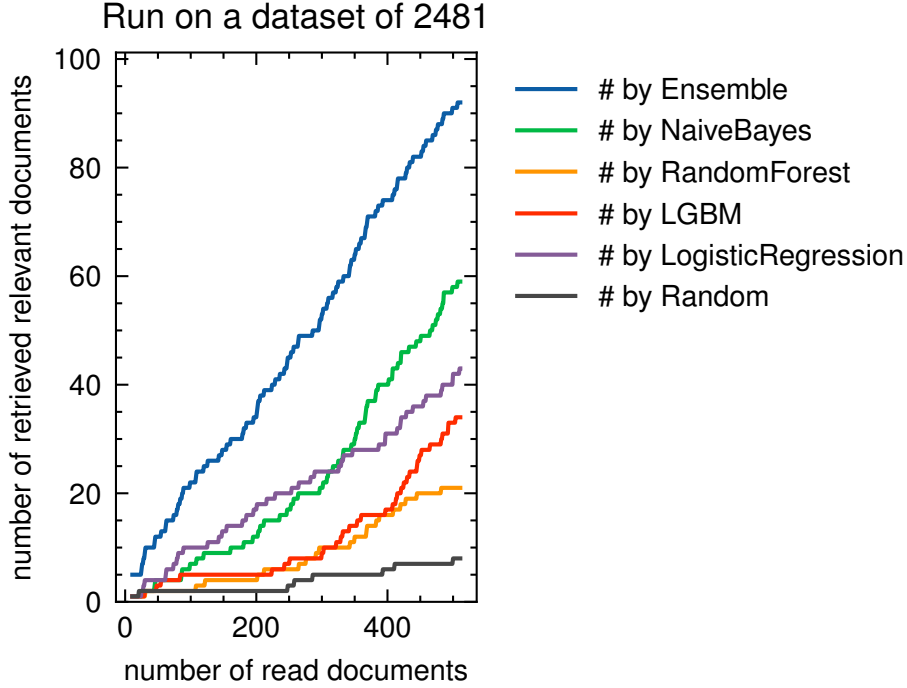


Figure 2: An example run for 500 iterations on a dataset. The *Ensemble* curve shows the number of documents that have been found by the overall system. The other curves display the number of relevant documents that have been found by the individual members within \mathcal{C} . The reader may notice that curves start slightly after 0 documents and end slightly after 500 documents. This is caused by the fact that our method requires five relevant and five irrelevant documents at the start of the process (see Section 3.5.4), which results in this shift.

$f_2 = 0$. For the example data, the estimate is

$$\hat{N} = 92 + \frac{40^2}{2 \cdot 33} = 116.24,$$

which differs 3.76 from the true value of $N = 120$.

3.2.1 Confidence Interval

In order to increase the reliability of the stopping criterion, it would be beneficial to have a measure of confidence for our point estimates. Stopping at a moment when the variance is high is not ideal. As in [31], we introduce a *conservative* stopping criterion, which uses the upper bound of a 95 % confidence interval as the stopping criterion (we describe the criteria in more detail in Section 3.4). Chao [10, 11] provides the following variance estimator.

$$\hat{\sigma}_{\hat{N}}^2 = \begin{cases} f_2 \left(\frac{1}{4} \left(\frac{f_1}{f_2} \right)^4 + \left(\frac{f_1}{f_2} \right)^3 + \frac{1}{2} \left(\frac{f_1}{f_2} \right)^2 \right) & \text{if } f_2 > 0 \\ \frac{f_1(f_1-1)}{2} + \frac{f_1(2f_1-1)^2}{4} - \frac{f_1^4}{4\hat{N}} & \text{if } f_2 = 0 \end{cases}. \quad (2)$$

Then the confidence interval can be estimated as,

$$\left[n + \frac{\hat{N} - n}{Q}, n + (\hat{N} - n) Q \right], \quad (3)$$

where

$$Q = e^{1.96 \sqrt{\ln \left(1 + \frac{\hat{\sigma}_{\hat{N}}^2}{(\hat{N} - n)^2} \right)}}, \quad (4)$$

in which 1.96 is the critical value of the normal distribution. For the case of $f_2 > 0$ in Equation 2, we refer to [10], for the case of $f_2 = 0$, we refer to [11]¹.

¹In [11], the equation for variance for both cases contains a term k . Conform [11, Equation (6a)] we rewrite the formula for both cases to a version without this term k so that it matches the equation for variance in [10].

For the example data, the variance is 100.82. Then, according to Equation 3 and Equation 4, the 95 % CI for this data is [103.11, 144.88]. In section Section 3.4, we further detail how the point estimates and intervals are used to stop the review process.

3.2.2 Model assumptions

Chao’s estimator is based on the *heterogeneity model* M_h introduced in [6, 7]. The model M_h assumes that the capture probability only varies among the individuals (so, the relevant documents in our case). Some documents have a higher probability of being selected by the committee members than others. Chao [10] assumes that for $(p_{ij} = p_i)$, where $i = 1, 2, \dots, N$ and $j = 1, 2, \dots, C$, then, p_1, p_2, \dots, p_N are a random sample from an unknown probability distribution function. Chao’s estimator assumes that the number of observations of an individual (in our case, the number of committee members that have found a relevant document i) is a realization from a zero-truncated Poisson distribution. Therefore, it is crucial to verify that the number of observations of a relevant document (which can be one of $\{1, 2, 3, 4, 5\}$) can be assumed to be a realization of a (truncated) Poisson distribution.

Poisson originally formulated his distribution as a limit of the binomial distribution [23] with a success probability p and N realizations, with N approaching infinity, p tending to zero, and Np remaining finite and equivalent to the Poisson parameter λ . However, even when N is small, the Poisson distribution can reasonably approximate the binomial distribution, given that p is small enough [38]. In our case, N is small, and the chance of encountering a document is also small.

The Poisson parameter can vary for each document i , allowing for heterogeneity in capture probabilities. This is convenient as some documents may be harder to find than others. Moreover, some methods may be better suited to finding a particular document than others. For example, the predictions of the Logistic Regression classifier may differ from the predictions of a Random Forest, even when the same set of documents are given as training data. In this case, for a document i the number of observations is stated as being $\lambda_i = \lambda_{i,LR} + \dots + \lambda_{i,RF}$. λ_i is a Poisson parameter $\lambda_{i,LR}$ and $\lambda_{i,RF}$ are also Poisson parameters. This follows from a property of the Poisson distribution found by [12]; a Poisson is infinitely divisible. If you have two independent Poisson random variables, X_1 with parameter λ_1 and X_2 with parameter λ_2 , then the sum of these two random variables, $X_1 + X_2$, will also follow a Poisson distribution with parameter $\lambda_1 + \lambda_2$. In our case, each relevant document i has Poisson parameter λ_{ij} where j is one of the members of the committee which uses a specific classification algorithm. The sum of, in our case, the five Poisson parameters leads to a Poisson parameter for each document i : $\lambda_i = \sum_{j \in C} \lambda_{ij}$.

An issue that sometimes arises in Population Size Estimation and Capture-recapture studies is *contagion*, which happens when the capture of an individual changes the probability of capturing it a second time (for example, an animal may change its behavior after capture, or the researcher may become better at finding that specific animal after observing it). Contagion violates the Poisson assumption (which follows from the independence between the trials in the binomial distribution). However, as our committee members search independently, we know that the contagion problem is not present: the Poisson parameters λ_{ij} for other members C_j do not change after its capture by any other member.

Given the properties of our problem and the Poisson distribution, we can state that this distribution is suitable and fulfills the assumptions of Chao’s estimator. Accounting for the heterogeneity, Chao’s estimator provides a lower bound on the number of relevant documents. However, a simulation study in [10] showed that, in many cases, it is a good estimator for N in general. Notice that Chao’s estimator only uses the frequencies of the documents discovered once and twice. The intuition behind the Chao estimator is that (for the ecology use-case) if you have seen many animals once (relative to the number of animals seen twice), then probably there are a lot more that you have missed completely; it would be surprising if you would have seen all unique animals exactly once. The more animals you have seen twice (relative to those seen once), the larger the probability you have seen most of them. Chao’s estimator formalizes this intuition and provides a lower bound by only considering the number of individuals seen once and twice.

3.3 Poisson Regression version

As mentioned earlier, we also use a Poisson regression version of Chao’s Moment estimator as presented by Rivest et al. [32]. This model also takes the frequencies $f_{3,4,5}$ into account, in addition to the frequencies $f_{1,2}$. Moreover, this model enables us to obtain a 95 % confidence interval using the profile likelihood instead of the asymptotic approach from [10]. The profile likelihood method has been advocated by many statisticians [1, 17, 19, 21]. We describe this method below. We will refer to this method as *Chao (Rivest)*, while we will refer to Chao’s Moment Estimator as *Chao (1987)*.

Using the data in Table 1, we can specify the design matrix for the *Chao (Rivest)* model in Table 2. The model has $C - 2$ parameters, called η parameters, for modeling heterogeneity in capture probabilities within the set of relevant documents. In our case, as $C = 5$, we have 3 η parameters. The Y variable contains the frequency statistics (from top to bottom, f_5 to f_1).

Table 2: The data (Y), which contains f_5 to f_1 (above to below). The rest of the columns belong to the design matrix for the *Chao (Rivest)* model.

Y	Intercept (γ)	beta (β)	eta3 (η_3)	eta4 (η_4)	eta5 (η_5)
0	1	5	3	2	1
2	1	4	2	1	0
17	1	3	1	0	0
33	1	2	0	0	0
40	1	1	0	0	0

Table 3: The fitted coefficients from the data presented in Table 2 before and after removal of negative η parameters.

	Before removal			After removal		
	Est.	S.E.	p	Est.	S.E.	p
Intercept (γ)	3.19	0.36	<0.001	3.50	0.22	<0.001
beta (β)	0.50	0.24	0.033	0.29	0.11	0.010
eta3 (η_3)	-0.07	0.45	0.885			
eta4 (η_4)	-1.19	0.87	0.174			
eta5 (η_5)	-20.63	42 247.17	1.000			

We use the package RCapture [32] to fit the model, which uses a standard Generalized Linear Model fitting algorithm. Given the data in Table 1, this algorithm fits the model with the following parameters as presented in the first half (“Before removal”) of Table 3. In this algorithm, all η parameters fitted with a negative coefficient are set to zero, as these parameters should theoretically be greater than or equal to zero [32] (when set to zero, these parameters are effectively removed from the design matrix). Note that after setting an η parameter to zero and fitting a new model, the other η parameters could be fitted with a negative coefficient, so this process is repeated until all η parameters are positive or removed. For this data, the algorithm does indeed remove all η parameters. The parameters of the final model are presented in the second half (“After removal”) of Table 3.

Using the parameters in Table 3, we can calculate the estimate for the number of relevant documents as follows;

$$\hat{N} = n + e^{\hat{\gamma}} = 92 + e^{3.5} = 125.18 .$$

This value differs by 5.18 from the ground truth $N = 120$ and is higher than the estimate by *Chao (1987)* (116.24).

3.3.1 Confidence Interval

The confidence interval for *Chao (Rivest)* is calculated by using the deviance or log-likelihood ratio of the models from complete tables, introduced in [17]. This procedure is set up as follows. Suppose we have an incomplete table with only the observed counts (for example, Table 2). We can extend and complete this table by adding a row for the unobserved count u . Then, we need to find the Poisson model for the extended table PE with the lowest deviance. We can find this by a search for u in the interval $u \in \left[0, \frac{3}{2}\hat{f}_0\right]$, equivalently $\hat{N}_C = n + u$, for a conditional estimate for N based on the complete table. We record for each model the log likelihood for

$$L(\hat{N}_C, \hat{\theta}_{N_C}; n) = D_{PE} - 2ct,$$

where D_{PE} is the deviance for model PE and a correction term [17, 32].

$$ct = \begin{cases} u - \hat{N}_C - \frac{\log \frac{u}{\hat{N}_C}}{2}, & \text{if } \hat{N}_C > 100 \wedge u \geq 2 \\ -\hat{N}_C + \frac{\log 2\pi \hat{N}_C}{2}, & \text{if } \hat{N}_C > 100 \wedge u \in [0, 1] \\ \log \frac{u^u \cdot \hat{N}_C!}{\hat{N}_C^{N_C} \cdot u!}, & \text{otherwise} \end{cases} .$$

Then, we find the value \hat{u}^* that maximizes this log-likelihood (or minimizes the deviance). By using the asymptotic χ_1^2 distribution, we can find the values u that increase this value by an amount k_α , where k_α is a critical value calculated using the quantile function from this χ_1^2 distribution [32]. In this case, for a 95 % CI, the critical value $k_{\alpha=0.05} = 3.84$.

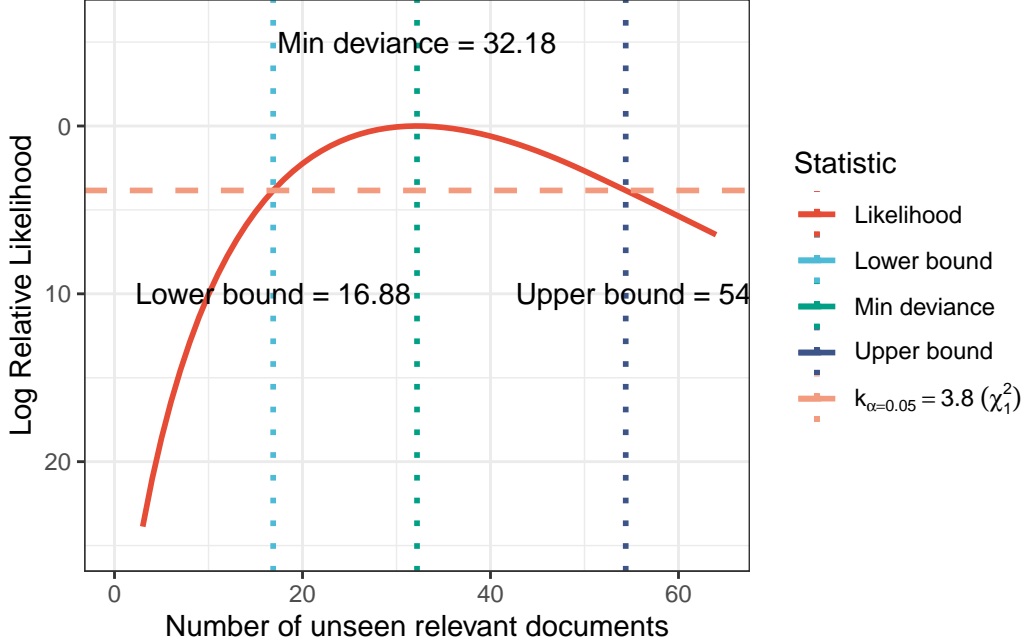


Figure 3: Calculating the 95 % confidence interval using the profile likelihood method for the frequency statistics in Table 1. In this figure, the log-likelihood for u^* is subtracted from the likelihood in aid of the visualization. Additionally, we inverted the y -axis for this purpose. By finding the values u that intersect with the line $y = k_{\alpha=0.05} = 3.84$, we can find the lower bound and upper bound of the interval.

This procedure is visualized in Figure 3 for the data in Table 1. Combining the interval found for \hat{f}_0 with $n = 92$, the 95 % CI obtained using this method is $[108.88, 146.39]$, which is similar to the interval obtained using *Chao (1987)* ($[103.11, 144.88]$).

3.4 Stopping Criterion

Using the estimate \hat{N} and the corresponding 95 % CI for N , we can determine if we can terminate the TAR procedure. The user can specify a recall target, such as 95 % recall (note that the 95% recall target is not be confused with the 95 % of the CI). The system tracks the estimate and CI to determine if the stopping criterion has been met. However, there are multiple ways to decide on the recall statistics and estimates.

In our implementation, similar to [31], we use the estimates as described in the previous sections in the following two ways:

Conservative. We use the *upper bound* of the CI \hat{N}_{sup} to determine the current recall estimate. The current recall estimate of iteration t is defined as $\hat{R}_t = \frac{|\mathcal{L}_t^+|}{\hat{N}_{\text{sup}}}$.

Optimistic. Here, we use the *point estimate* \hat{N} of the estimator as to determine the current recall estimate. For this criterion, the current recall of iteration t estimate is defined as $\hat{R}_t = \frac{|\mathcal{L}_t^+|}{\hat{N}}$.

Both methods are triggered when $\hat{R}_t \geq R_{\text{target}}$. The estimated recall percentage is rounded to nearest integer value to allow some numerical imprecision. Moreover, the criteria can also only be triggered after $|\mathcal{L}_t| > 100$, as the estimates may fluctuate heavily in the first phase of the procedure.

Combined with the two estimators, we provide four stopping criteria, which can be used with a user specified recall target.

- *Chao (1987)* - Conservative
- *Chao (1987)* - Optimistic
- *Chao (Rivest)* - Conservative
- *Chao (Rivest)* - Optimistic

3.5 Active Learning procedure

In the following sections, we describe the Machine Learning and Active Learning aspects of our method. First, we briefly describe the feature extraction method that we employ, followed by the classifiers that are used within the ensemble. As the data is often imbalanced, dynamic resampling is used as a balancing procedure. Finally, we describe the query strategy and batching scheme.

3.5.1 Feature Extraction

All documents are represented as TF-IDF vectors for all classification algorithms. We only include terms with a minimum document frequency of 2 and limit the term matrix to 3000 terms by selecting based on the term frequency across the dataset in question. Moreover, English stop words are excluded from the term matrix. If desired, our framework allows substituting the TF-IDF vectorizer with another algorithm; for example, Doc2Vec or SentenceBERT on the committee member level, allowing the use of multiple vector representations concurrently. However, in our experiments, we will not use this feature.

3.5.2 Classifiers

We create an ensemble of various learners, each of which uses a unique classification algorithm. The decision boundaries of each algorithm will differ, resulting in a different order of document selection. The algorithms we use are:

Multinomial Naive Bayes. Multinomial Naive Bayes is a probabilistic classification algorithm frequently used for text classification tasks. It is also used in TAR systems, e.g., it is the default classification algorithm in [37]. In most cases, this algorithm is used with documents in Bag-of-Words representation, such as TF-IDF vectors.

Logistic Regression. Logistic Regression is a common classification algorithm used in TAR, for instance, in AutoTAR [15] and derivatives. The method is also used or available as an option in [31, 37].

Random Forest. Random Forest [22] is an ensemble learning method that uses multiple decision trees to make predictions. Each tree in the forest is trained on a randomly selected bootstrap sample of the training data, and at each node, the best split is chosen among a randomly selected subset of the features. The final prediction is made by a majority vote among the trees in the forest. Ranking is possible by using the mean predicted class probabilities of the trees in the forest. The class probability of a single tree is defined as the fraction of training samples in the leaf that have the same class as the leaf. The Random Forest classifier is available as an option in [37].

Light Gradient Boosting Machine (LGBM) Light GBM [28] is a gradient boosting framework that uses a tree-based learning algorithm. It is designed to be highly efficient and scalable, with faster training speeds and lower memory usage compared to other popular gradient boosting frameworks. To our knowledge, it is not used in any existing TAR systems, but it performs similarly to the Random Forest method for some datasets.

Random Sampling We also include one member that does not use machine learning nor active learning. The idea behind this is that we may capture instances in unexplored areas of the search space that are not covered by the greedy searching machine learning-based committee members.

Support Vector Machines (SVM) is another viable option to consider instead of one of the previously mentioned machine learning methods. However, during our initial experiments, we discovered that using SVM significantly impacted the training time for each iteration and, thus, the total runtime of each experiment. In practical applications, this may not be a concern, as manual review time typically exceeds the training time of the models. We anticipate substituting one of the classifier algorithms with SVM will not substantially impact the results.

3.5.3 Balancing

Many classification algorithms encounter difficulties when fitting models with limited data, especially in the case of imbalanced datasets. This limitation may be because the prevalence of the relevant class is generally low in most TAR datasets. To address this challenge, one potential solution is to balance the training data. One method called dynamic resampling [20, 37], rebalances the training data by oversampling documents from the positive class \mathcal{L}^+ and undersampling from \mathcal{L}^- . The amount of oversampling and undersampling is dynamic and depends on the sizes of the sets \mathcal{L}^- , \mathcal{L}^+ and \mathcal{L} . The method ensures that the size of the training data remains the same in terms of $|\mathcal{L}|$. A more detailed description of this method is given in [20].

Our early experiments have demonstrated that using this method significantly improves the performance of our models, particularly in terms of WSS@95 (Work Saved over Sampling at 95 % recall), leading us to employ this procedure to balance the training data for all the classifier models in our system.

3.5.4 Training, Ranking, and Sampling

Each of the members’ models is trained using their own labeled sets \mathcal{L}_i , and the training data is balanced using the method described previously. Then, we predict for each document within the member’s \mathcal{U}_i its relevancy probability. These scores are used to rank each document. Each member i prepares a batch of b_i documents, which are greedily sampled from a batch which consists of the top- b_i documents from the ranking produced by the model. Then, when the user queries a document from the committee, one of the members is randomly selected to propose a document, which will be the first document in the batch. When all documents from a member’s batch have been labeled, its model is retrained. After retraining, we update the batch size b_i in the same manner as in AutoTAR [15]: $b_i \leftarrow b_i + \lceil \frac{b_i}{10} \rceil$. The initial batch size $b_i = 1$.

This process is repeated until the stopping criterion is met. Each committee member needs one document from each class as initial training data to start the process. As there are five members in our committee, we need five relevant documents in total (and five non-relevant ones).

4 Experimental Setup

Below, we describe our experimental setup and research questions. We also briefly describe the datasets that are used for benchmarking. Furthermore, we list the existing methods that we take into account in our comparison.

4.1 Research questions

In TAR, the goal is to retrieve as many relevant documents as possible (achieving a high recall) while minimizing the workload in terms of review work. A good stopping criterion should achieve its recall target with minimal cost. In order to evaluate how our method performs (as well as compared to other methods), we study the following research questions.

1. How does our Active Learning strategy perform in terms of WSS@95 and WSS@100 compared to other methods?
2. Can our stopping criteria help the user achieve its recall target in a timely fashion?
3. How reliable are our stopping criteria?
4. How do our stopping criteria compare to other methods that estimate the number of relevant documents?
5. How do our stopping criteria compare to other methods that do not provide such an estimate?

4.2 Study design

To ensure that our findings are generalizable to new and unseen datasets, we run each Active Learning method and corresponding stopping criteria on a large collection of datasets of various domains. Moreover, we will repeat the experiment multiple times for each datasets. The datasets are described in more detail in Section 4.3.

4.2.1 Active Learning initialization

Many TAR procedures require a seed set of relevant and irrelevant documents to start the Active Learning loop. Earlier work has shown that the documents in the seed set can influence the results [8]. Our method, consisting of several committee members (each representing an active learning strategy), requires a seed set of five relevant and five irrelevant documents. To ensure that our results do not depend on a single seed set, we repeat our experiments with varying sets of seed documents. We use 30 distinct seed sets for each dataset and method. To ensure a fair comparison, the methods that can work with a smaller seed set also get a seed set of the same size as our method. Moreover, the set of sampled documents depends on the seed value that is given to the Pseudo Random Generator; this means that an experiment for a method A with seed s and an experiment with method B with the same seed value s use the same documents to initialize the Active Learning procedure.

4.2.2 Feature Extraction

All methods in our study use TF-IDF feature vectors. To ensure a fair comparison, we keep the configuration the same for each method, so as per Section 3.5.1, limited to 3000 terms after filtering English stop words from the vocabulary.

4.2.3 Evaluation metrics

We let each algorithm run until all documents are screened. During the experiment, each criterion can signal when it is triggered. Moreover, if the method produces an estimate for the size of \mathcal{D}^+ , then this estimate is also registered. When a method triggers a stopping criterion, the following metrics are recorded.

Effort. The percentage of documents that have been screened after triggering the stopping criterion.

$$E = \frac{|\mathcal{L}|}{|\mathcal{D}|} \tag{5}$$

Table 4: Main statistics of the corpora included in our experiments. Here, N is the number of datasets/topics within each corpus. We report the median and the interquartile range of the dataset statistics within each corpus.

	SYNERGY, N = 20	clef2017, N = 31	clef2018, N = 24	clef2019, N = 19
# Relevant	67 (32, 106)	92 (49, 126)	67 (39, 277)	64 (33, 78)
# Irrelevant	3,554 (1,690, 6,864)	3,211 (1,498, 6,950)	5,064 (1,700, 8,553)	3,158 (1,770, 5,412)
Size	3,577 (1,725, 7,329)	3,241 (1,596, 7,261)	5,123 (1,898, 8,592)	3,169 (1,840, 5,506)
Prevalence (%)	1.5 (0.8, 5.0)	2.4 (1.0, 5.6)	2.8 (1.0, 6.6)	1.8 (0.9, 5.3)

Recall. The percentage of relevant documents that have been found based on the *a priori* knowledge from the ground truth dataset. We will record the recall when the stopping criterion has been satisfied.

$$R = \frac{|\mathcal{L}^+|}{|\mathcal{L}^+ \cup \mathcal{U}^+|} \quad (6)$$

Recall Error. For the methods that can specify a recall target; the error is the absolute difference between the achieved recall and the target recall when the stopping criterion is triggered, divided by the recall target.

$$RE = \frac{|R_{\text{stop}} - R_{\text{target}}|}{R_{\text{target}}} \quad (7)$$

Work Saved over Sampling. This metric expresses the work reduction over random sampling. We calculate this as follows:

$$WSS = \frac{|\mathcal{U}|}{|\mathcal{D}|} - \left(1 - \frac{|\mathcal{L}^+|}{|\mathcal{L}^+ \cup \mathcal{U}^+|}\right) \quad (8)$$

loss_{er}. This metric introduced in [15] aims to assess both review costs and recall. It is defined as:

$$\text{loss}_{\text{er}} = (1 - R)^2 + \left(\frac{100}{|\mathcal{D}|}\right)^2 \cdot \left(\frac{|\mathcal{L}|}{|\mathcal{L}^+| + 100}\right)^2, \quad (9)$$

where R is the recall as defined in Equation 6. This metric consists of two term. The first is the loss due to missing relevant documents, which becomes higher when the recall is low. The second term is the loss in terms of effort. The two scalar values 100 in the metric are considered a correction for reasonable extra work for achieving a high recall.

Target met. In this metric, we test for each individual run if the recall target was met. We report the percentage of runs over all datasets in which this is the case.

Triggered. For each individual run, we record if the stopping criterion was triggered before all documents were exhausted. As with the *Target met* metric, we report the percentage of runs over all datasets in which this is indeed the case.

Running the experiment till exhaustion enables us to measure what would have happened if the method did not stop. This enables us to assess how many documents would have remained for a method to achieve its target recall in case it stopped the process too early.

4.3 Datasets

We use benchmark datasets of several corpora to ensure the results of our methods are generalizable to unseen datasets. One of the corpora used in our experiments consists of systematic literature reviews from [18]. This corpus contains datasets with inclusion and exclusion records from several real-world published systematic reviews from various domains: psychology, the medical field, and information sciences, among others. We also include in our experiments three corpora from the Conference and Labs of the Evaluation Forum (CLEF) Technology-Assisted Reviews in Empirical Medicine datasets from the years 2017, 2018, and 2019 [24–26]. This corpus is in TREC format. The CLEF Task was aimed to evaluate search methods aimed to identify all relevant works for a systematic literature review in empirical medicine.

Table 4 shows some of the dataset characteristics of each of the corpora. We describe the individual datasets in more detail in Appendix A in Table 12, Table 13, Table 14, and Table 15.

4.3.1 Selection

As said earlier, our method needs at least five initial relevant documents to train a machine learning model for each committee member. Moreover, methods such as the Target rely on the fact that there are at least ten documents within the dataset. Therefore, we opt to only include datasets in our experiments that contain at least ten relevant documents because the inclusion of runs in which more than half of all relevant documents as prior knowledge may distort the results. The CLEF corpora contain datasets with an extremely small size. We decided to exclude datasets with less than 500 documents for two reasons: first, many criteria do not work well on these datasets (e.g., Stop200, Stop400, Budget, Knee, Conservative, and Optimistic). The second reason is that the advantage of using TAR on these datasets is minimal compared to the work required to process the whole dataset. Furthermore, we exclude two datasets from the CLEF corpora due to their large size; we cannot assess the standard version of AUTOSTOP on these datasets due to memory limitations (see Section 2.1.6).

4.4 Comparison to other methods

To assess our method, we will compare the metrics to methods presented in earlier work. We will include several methods discussed in Section 2.1. An overview of all the methods included in our experiments is given in Table 5.

Table 5: An overview of all methods included in the experiments. The *I*, *S*, and *H* in the *Applicability* column stand for *Interventional*, *Standoff*, and *Hybrid*, whereas the *C* and *H* in the *Certification* column stand for *Certification* and *Heuristic*. The *AL* column gives the *Active Learning* method that is used with the criteria in the experiments. The CMH methods are the Hypergeometric methods from [8] referring to the first letters of the authors’ surnames. Our implementation of some methods is based on the implementation in TARexp [42], if this is the case it is listed.

Method	Applicability	Certification	AL	Source
AutoStop	I	C	AutoStop	[31]
Budget	S	H	AutoTAR	[15, 42]
Chao (ours)	I	C	Ensemble	
CMH-Standoff	S	C	AutoTAR	[8, 42]
CMH-Hybrid	H	C	AutoTAR	[8]
Half	S	H	AutoTAR	[42]
Knee	S	H	AutoTAR	[15, 42]
Quant (CI)	S	C	AutoTAR	[41, 42]
Rule2399	S	H	AutoTAR	[42]
Stop after k	S	H	AutoTAR	
Target	H	H	AutoTAR	[15]

For the standoff methods, we choose to apply them to AutoTAR, as this method is considered state of the art and does not perform any additional work to decide when to stop. For AutoStop, we specifically implemented the Horvitz-Thompson variant, as suggested by the authors [31].

4.5 Implementation

We provide a Python library, `python-allib` (see [5]), which implements our methods and all the baselines, some of which are adapted from TARexp [42]. The TARexp package only allows the comparison of stop criteria that fall in the standoff category (see Section 2.1), but not methods of interventional nature. Our framework allows various forms of ranking and arranging the reviewer workload so interventional methods can be implemented. The library is based on the Python package `instancelib` [4], enabling integration within annotation software. Furthermore, we provide a repository on Github² and ZENODO (see[3]) that contains the scripts which the reader can use to reproduce our results.

5 Results

5.1 Comparing Sampling strategies

In Table 6, we show the performance of each sampling strategy. The AutoTAR method globally outperforms the others, which is to be expected as this method does not perform any additional work to enable the estimation of the current recall. This is also a result reported in earlier work (e.g.,[31]). Overall, AutoTAR has a mean WSS@95 of 74.3 % vs. ours (Ensemble) of 63 % and 45.5 % for AUTOSTOP. Our method does not outperform AutoTAR in terms of WSS@95. This result was to be expected for our Ensemble method, because our method uses Random Sampling for

²The repository can be found on <https://github.com/mpbron/allib-chao-experiments>. The repository of `python-allib` can be found on <https://github.com/mpbron/allib>.

Table 6: Comparison of Work Savings between Active Learning strategies when using a Perfect Stopping criterion that directly stops after the recall target has been achieved.

Target		Sampling Strategy		
		AUTOSTOP	AutoTAR	Ensemble
95 %	Effort (%)	51 ± 12	22 ± 16	33 ± 19
	WSS (%)	45 ± 13	74 ± 16	63 ± 19
	loss-er	0.09 ± 0.07	0.02 ± 0.05	0.05 ± 0.07
100 %	Effort (%)	66 ± 18	42 ± 29	52 ± 26
	WSS (%)	34 ± 18	58 ± 29	48 ± 26
	loss-er	0.14 ± 0.10	0.07 ± 0.10	0.09 ± 0.10

selecting approximately 20 % of the instances. Note that the reported standard deviation in Table 6 indicates that some datasets are more difficult than others. For example, AutoTAR only achieves a WSS@95 of 10.3 % on the Moran dataset. Considering only the WSS@95 and WSS@100, our method stays closer to the performance of AutoTAR than AUTOSTOP.

5.2 Recall and Estimator curves

In our experiments, the stopping criteria are called every ten review decisions. If the stopping criterion uses an estimator, it will also record the current point estimate and confidence interval. In a real-world application, the system can plot these estimates together with the recall statistics of the process. This plot can be an informative aid to users in deciding whether or not to stop the review process.

In Figure 4, we show and compare the estimates and stopping points of our method, AUTOSTOP, Quant, and CMH for runs of two datasets. These methods allow the specification of recall targets, however the CMH method does not provide an estimate on the current level of recall or the number of relevant documents. Note that for our method (Chao), we show the results of the individual committee members from within the ensemble. These members are displayed in light gray.

In Figure 4a, our estimator fluctuates drastically during the start of the process. When more documents are found by multiple committee members, the estimates become more stable although new documents keep being discovered. A similar behavior is visible for AUTOSTOP in Figure 4c. The Quant rule [41] overestimates the number of relevant documents for an extended period. Because of the large CI, Quant’s Conservative recall criteria are never triggered for the high recall targets. The *Chao (Rivest) - Conservative* 100 % target is not triggered for our method on the Van Dis dataset. AUTOSTOP’s criterion is only triggered a few documents before all documents are exhausted (holds for both datasets). Note that for the runs displayed in Figure 4, the seed sets for a dataset are kept the same among all methods, allowing a fair comparison. For these runs, our methods need less reader effort than AUTOSTOP. Furthermore, the AUTOSTOP requires less effort than the CMH and Quant methods, even though AutoTAR is more efficient in retrieving all relevant documents than AUTOSTOP.

5.3 Criteria with recall targets

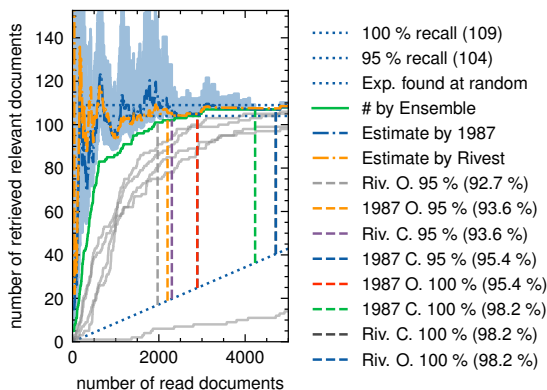
In this section, we compare the criteria with recall targets over all datasets. We will consider the result metrics for each time the stop criteria were triggered. For each of the 94 datasets in our collection, we have 30 runs with for each run a different seed sets. This results in 2820 runs per method. In Table 7 and Table 8, the results for all seven metrics are given for all methods and, if applicable, for various recall targets. We report the mean and standard deviation (if applicable) for each score, enabling the reader to assess the dispersion of each score. We discuss the results of the Conservative and Optimistic criteria separately. Moreover, we report the results of the CMH method [8], which does not provide estimates but allows the specification of a recall target. We list the standoff version as in Table 9. We will discuss several recall targets, but we will mainly focus on the high 95 % and 100 % recall targets.

5.3.1 Conservative methods

In Figure 5 and Figure 6, the results of all the runs with a 95 % and 100 % recall target are displayed. The analysis and figures are based on work performed in [8], in which the authors performed and presented a similar analysis.

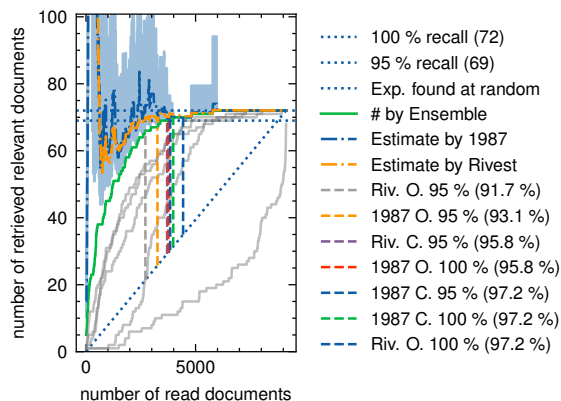
When comparing the Conservative methods for 95 %, we can see that while the tail of the distribution of the *Chao (1987)* method of the recall is 5.91 percent points lower than *AUTOSTOP*’s recall, our WSS is 32.46 points higher. Our method’s recall vs. work savings trade-off is slightly more leaned towards the latter. The results of the two Chao versions are similar but slightly in favor of Chao’s original estimator. Note that for *Chao (Rivest)*, there are many runs for which the work savings are below 5 %. This is still the case for *Chao (1987)*; however, it is less pronounced.

Run on a dataset with 109 inclusions out of 12700



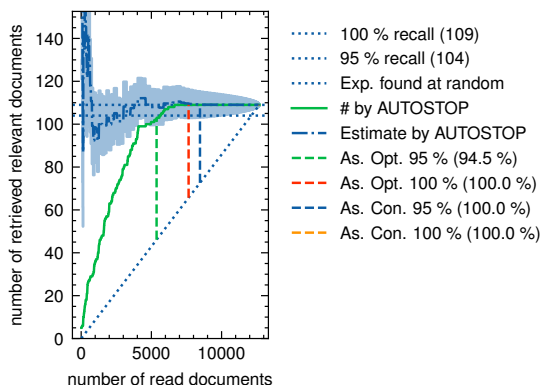
(a) Chao – CLEF2017-CD011548 dataset (ours)

Run on a dataset with 72 inclusions out of 9128



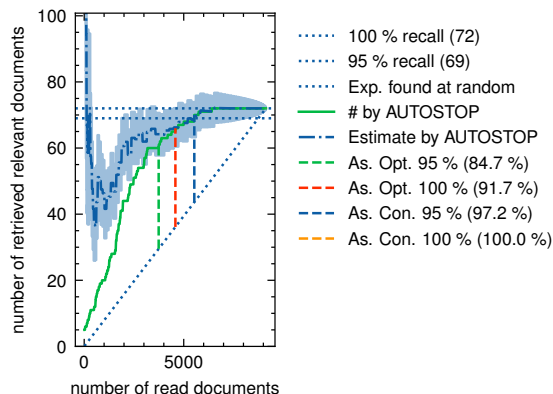
(b) Chao – Van Dis dataset (ours)

Run on a dataset with 109 inclusions out of 12700



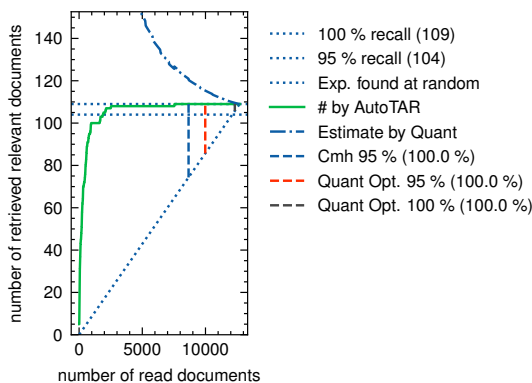
(c) AUTOSTOP – CLEF2017-CD011548 dataset

Run on a dataset with 72 inclusions out of 9128



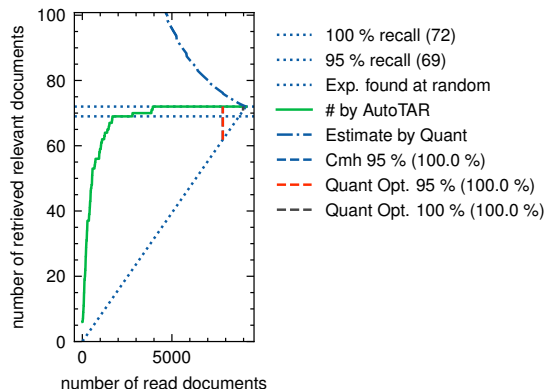
(d) AUTOSTOP – Van Dis dataset

Run on a dataset with 109 inclusions out of 12700



(e) Quant & CMH – CLEF2017-CD011548 dataset

Run on a dataset with 72 inclusions out of 9128



(f) Quant & CMH – Van Dis dataset

Figure 4: Recall curves for two datasets. The dashed blue diagonal line shows how many documents would have been found at random. The horizontal lines show the 95 and 100 % recall targets. The vertical dashed lines show when the stopping criteria have been triggered. The ribbons around the estimates show their confidence intervals.

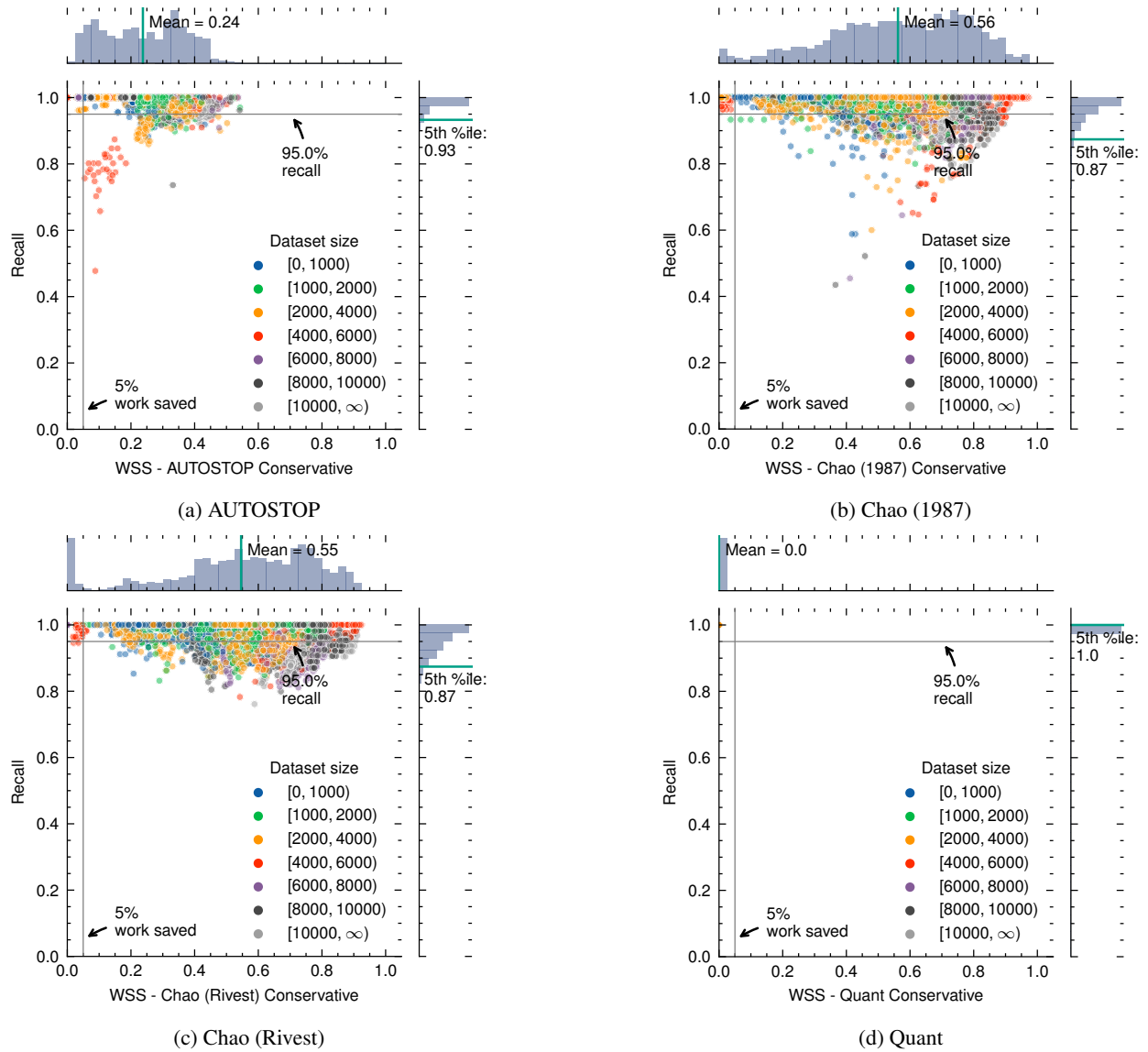


Figure 5: These figures display the results of all runs on all datasets in terms of Work Saved over Sampling and Recall for the Conservative Stopping criteria with a 95 % recall target. The colors are indicative of the dataset size. The outer graphs show the overall distribution of WSS and Recall. For each method, the mean WSS and the 5th percentile of the recall are shown.

Table 7: This table shows all metrics recorded for each Conservative criterion. Note that for the Effort, Recall, WSS, loss_{er} and Error, the mean and standard deviation are reported.

Target	Conservative Methods				
	AUTOSTOP	Chao (1987)	Chao (Rivest)	Quant	
70 %	Effort (%)	28 ± 6	17 ± 13	17 ± 12	83 ± 23
	Recall (%)	74 ± 9	74 ± 14	75 ± 14	100 ± 1
	WSS (%)	46 ± 12	57 ± 16	58 ± 16	16 ± 22
	loss-er	0.11 ± 0.06	0.10 ± 0.09	0.10 ± 0.08	0.33 ± 0.25
	Error (%)	11 ± 10	18 ± 12	17 ± 12	42 ± 2
	Target met (%)	66	64	65	100
	Triggered (%)	100	100	100	44
80 %	Effort (%)	37 ± 6	22 ± 16	21 ± 14	93 ± 16
	Recall (%)	85 ± 8	83 ± 11	82 ± 11	100 ± 0
	WSS (%)	47 ± 10	61 ± 16	61 ± 16	7 ± 15
	loss-er	0.09 ± 0.05	0.07 ± 0.06	0.07 ± 0.06	0.36 ± 0.23
	Error (%)	9 ± 7	12 ± 8	11 ± 8	25 ± 0
	Target met (%)	76	65	62	100
	Triggered (%)	100	100	100	20
90 %	Effort (%)	56 ± 11	31 ± 20	29 ± 18	99 ± 3
	Recall (%)	94.8 ± 5.3	92.1 ± 6.7	90.7 ± 7.1	100.0 ± 0.0
	WSS (%)	39 ± 11	61 ± 19	62 ± 17	1 ± 3
	loss-er	0.15 ± 0.14	0.06 ± 0.08	0.06 ± 0.07	0.36 ± 0.22
	Error (%)	6.6 ± 4.4	6.4 ± 4.6	6.4 ± 4.8	11.1 ± 0.0
	Target met (%)	86	68	59	100
	Triggered (%)	100	100	100	5.3
95 %	Effort (%)	74 ± 14	40 ± 23	41 ± 25	100 ± 0
	Recall (%)	98.2 ± 3.3	95.9 ± 5.0	95.3 ± 4.2	100.0 ± 0.0
	WSS (%)	24 ± 12	56 ± 22	55 ± 24	0 ± 0
	loss-er	0.25 ± 0.21	0.08 ± 0.10	0.11 ± 0.19	0.37 ± 0.22
	Error (%)	4.20 ± 2.39	3.88 ± 3.63	3.62 ± 2.51	5.26 ± 0.00
	Target met (%)	89	70	59	100
	Triggered (%)	100	100	94	0
100 %	Effort (%)	100 ± 1	52 ± 27	84 ± 26	100 ± 0
	Recall (%)	100.00 ± 0.07	98.02 ± 4.35	99.47 ± 1.06	100.00 ± 0.00
	WSS (%)	0 ± 0	46 ± 26	15 ± 25	0 ± 0
	loss-er	0.36 ± 0.22	0.10 ± 0.10	0.34 ± 0.25	0.37 ± 0.22
	Error (%)	0.00 ± 0.07	1.98 ± 4.35	0.53 ± 1.06	0.00 ± 0.00
	Target met (%)	100	53	70	100
	Triggered (%)	56	99	33	0

For the 100 % target recall, the differences between the criteria are more pronounced. For instance, the AUTOSTOP criterion is not triggered in many runs, and the work savings are minimal for the few runs it is triggered. This is even more the case for the Quant Rule, as it is never triggered. There is a large difference between the Work Savings between *Chao (Rivest)* and *Chao (1987)* for the 100 % criterion. This is because the *Chao (Rivest)* can also provide a CI for when f_1 and f_2 both become zero, whereas *Chao (1987)* cannot. Also, when $\hat{N} = n$, the upper bound of the CI is by definition n (see Equation 3).

For all conservative criteria, it holds that when the upper bound of the CI $\hat{N}_{\text{sup}} > n$, the stopping criteria for 100 % recall cannot be triggered (unless \hat{R} is rounded up to 100 %). For several datasets, the *Chao (1987)* - Conservative estimates are not triggered during the run due to the fact that the CI is still not small enough (e.g., visible in Figure 4b). However, for the runs that it is triggered, this still results in a mean recall of 46.05 %.

Considering the lower recall targets (e.g., 70 % and 80%), our method’s mean recall (as presented in Table 7) is slightly higher than the target recall. However, the standard deviation and error rates of all estimator methods decrease as the recall target increases. This result was also reported in [31]. The Quant Conservative method is not often triggered; this results in a very high recall, even for the lower recall targets. The percentage of times this criterion is triggered tends to zero as the recall target increases.

Regarding reliability for the high recall targets, our method does not achieve its recall target as often as AUTOSTOP. However, our method is close, as reported by the low error rate. Given the higher work savings and the mean recall compared to the other estimator methods, especially the *Chao (1987)* method provides a good alternative to the AUTOSTOP Criterion. This is especially true for the 100 % criteria. As the *Chao (1987)* criterion is triggered 99.37 % of the time vs. 55.93 % for AUTOSTOP.

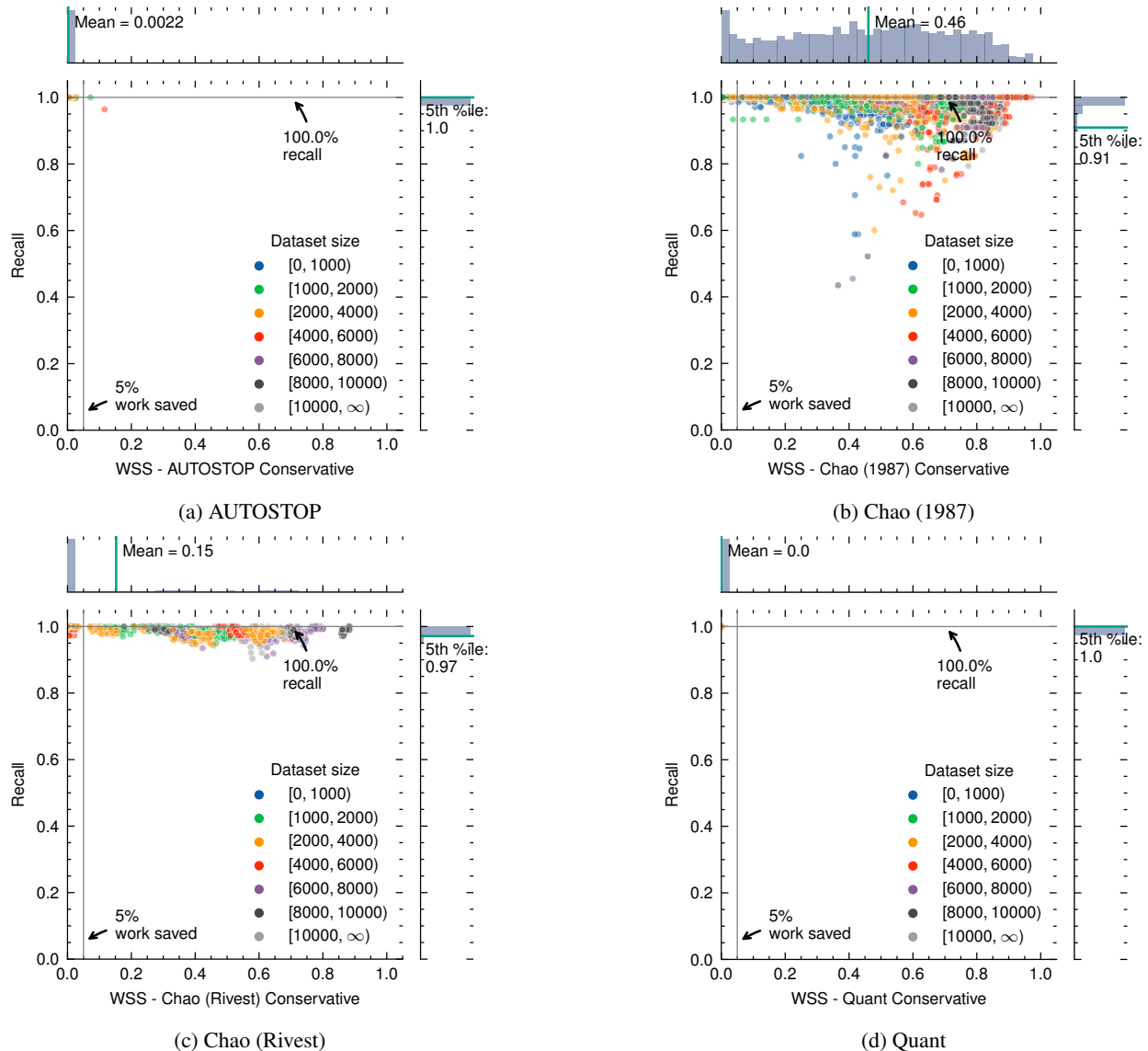


Figure 6: These figures display the results of all runs on all datasets in terms of Work Saved over Sampling and Recall for the Conservative Stopping criteria with a 100 % recall target. The colors are indicative of the dataset size. The outer graphs show the overall distribution of WSS and Recall. For each method, the mean WSS and the 5th percentile of the recall are shown.

5.3.2 Optimistic methods

For the optimistic methods, the results are presented in Table 8. In Figure 7, the results of all the estimator methods are displayed. The achieved recall by our stopping criteria is lower than AUTOSTOP reaches; this is especially visible through the 5th percentile of the recall scores, which is 3.03 percent points lower for *Chao (Rivest)*. There is also a significant difference between our criteria, as the 5th percentile is 11.59 points higher for Rivest’s Poisson Regression version. Our results show that *Chao (Rivest)* outperforms *Chao (1987)* for all recall targets in terms of recall. For the 100 % recall target, both our estimators improve over AUTOSTOP. For *Chao (1987)*, the recall distributions are similar, while the work savings are improved. For *Chao (Rivest)*, the 5th percentile of the recall scores lies at 94.5 %, which

Table 8: This table shows all metrics recorded for each Optimistic criterion, as well as the CMH method. Note that for the Effort, Recall, WSS, loss_{er} and Error, the mean and standard deviation are reported.

Target		Optimistic Methods			
		AUTOSTOP	Chao (1987)	Chao (Rivest)	Quant
70 %	Effort (%)	23 ± 7	11 ± 9	12 ± 10	45 ± 10
	Recall (%)	66 ± 9	57 ± 17	63 ± 16	98 ± 3
	WSS (%)	42 ± 11	47 ± 16	51 ± 16	53 ± 10
	loss_{er}	0.15 ± 0.07	0.22 ± 0.14	0.17 ± 0.12	0.10 ± 0.08
	Error (%)	11 ± 10	24 ± 17	20 ± 15	40 ± 5
	Target met (%)	27	24	33	100
	Triggered (%)	100	100	100	100
80 %	Effort (%)	30 ± 8	13 ± 11	15 ± 11	57 ± 10
	Recall (%)	76 ± 9	65 ± 16	71 ± 14	99 ± 2
	WSS (%)	46 ± 11	52 ± 16	56 ± 15	43 ± 11
	loss_{er}	0.10 ± 0.06	0.16 ± 0.12	0.12 ± 0.08	0.14 ± 0.12
	Error (%)	9 ± 8	22 ± 16	16 ± 12	24 ± 2
	Target met (%)	28	17	27	100
	Triggered (%)	100	100	100	100
90 %	Effort (%)	38 ± 8	18 ± 14	20 ± 14	72 ± 10
	Recall (%)	86 ± 7	77 ± 13	82 ± 10	100 ± 1
	WSS (%)	48 ± 11	59 ± 15	62 ± 15	28 ± 10
	loss_{er}	0.08 ± 0.05	0.09 ± 0.08	0.06 ± 0.05	0.22 ± 0.16
	Error (%)	7 ± 7	16 ± 13	11 ± 9	11 ± 1
	Target met (%)	24	12	21	100
	Triggered (%)	100	100	100	100
95 %	Effort (%)	43 ± 9	23 ± 17	25 ± 16	82 ± 8
	Recall (%)	90 ± 7	84 ± 11	88 ± 7	100 ± 1
	WSS (%)	47 ± 10	62 ± 16	64 ± 16	18 ± 8
	loss_{er}	0.08 ± 0.05	0.06 ± 0.06	0.05 ± 0.04	0.27 ± 0.19
	Error (%)	6 ± 6	12 ± 11	8 ± 6	5 ± 1
	Target met (%)	16	10	19	100
	Triggered (%)	100	100	100	100
100 %	Effort (%)	55 ± 11	39 ± 24	46 ± 21	98 ± 2
	Recall (%)	95.9 ± 5.4	95.9 ± 5.5	98.3 ± 2.1	100.0 ± 0.1
	WSS (%)	41 ± 10	56 ± 22	52 ± 21	2 ± 2
	loss_{er}	0.10 ± 0.06	0.06 ± 0.07	0.09 ± 0.10	0.35 ± 0.22
	Error (%)	4.1 ± 5.4	4.1 ± 5.5	1.7 ± 2.1	0.0 ± 0.1
	Target met (%)	20	23	39	98
	Triggered (%)	100	100	100	99

is a large improvement over AUTOSTOP (86.21 %). However, AUTOSTOP provides a slightly higher recall than *Chao (Rivest)* for the lower recall targets. A trait that is visible for all Optimistic criteria is that all methods tend to underestimate the number of relevant documents. However, the Recall Error decreases as the recall target becomes higher.

The Quant method overestimates the number of relevant documents for all recall targets. While the Quant method meets the recall target in nearly all cases, the error in recall prediction is very high. This is visible in the results for the 70 % recall target: for nearly all runs, the method stops at a point where (almost) all relevant documents from the dataset are retrieved.

5.3.3 CMH method (Standoff version)

The results for the standoff version of the CMH method are displayed in Table 9. In Figure 9, the WSS and Recall scores for the CMH methods are displayed. In Figure 4e and Figure 4f, the results of the 95 % target are shown for that particular run. The results are similar to the results of the Quant Optimistic method, although the percentage of runs that this criterion is triggered is lower. This holds especially for the 100 % and 95 % targets: the 100 % criterion is never triggered, and the 95 % recall target is only triggered 77.02 % of the time. As the average recall for the 70 % recall target is already 60.1 %, it is evident that this criterion underestimates the recall for a long time.

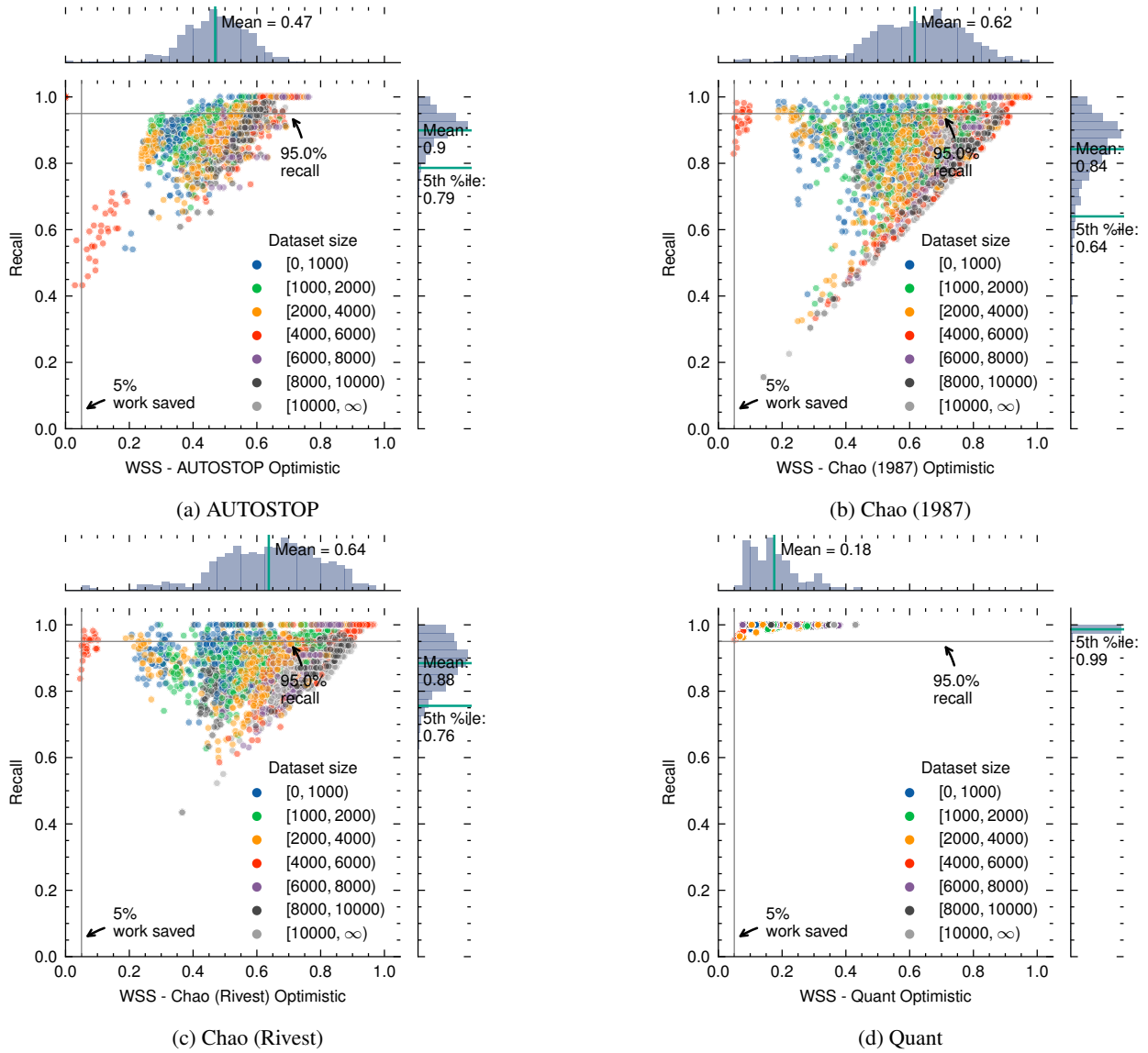


Figure 7: These figures display the results of all runs on all datasets in terms of Work Saved over Sampling and Recall for the Optimistic Stopping criteria with a 95 % recall target. The colors are indicative of the dataset size. The outer graphs show the overall distribution of WSS and Recall. For each method, the mean WSS and the 5th percentile of the recall are shown.

Table 9: This table shows all metrics recorded for the CMH standoff method, for each of the studied recall targets. Note that for the Effort, Recall, WSS, $loss_{er}$ and Error, the mean and standard deviation are reported.

CMH Standoff Method					
	70 %	80 %	90 %	95 %	100 %
Effort (%)	38 ± 13	48 ± 15	65 ± 18	82 ± 16	100 ± 0
Recall (%)	98.54 ± 2.23	99.18 ± 1.43	99.64 ± 0.74	99.87 ± 0.37	100.00 ± 0.00
WSS (%)	60 ± 14	51 ± 15	35 ± 18	18 ± 16	0 ± 0
loss-er	0.07 ± 0.07	0.11 ± 0.12	0.20 ± 0.21	0.30 ± 0.24	0.37 ± 0.22
Error (%)	41 ± 3	24 ± 2	11 ± 1	5 ± 0	0 ± 0
Target met (%)	100	100	100	100	100
Triggered (%)	100	100	97	77	0

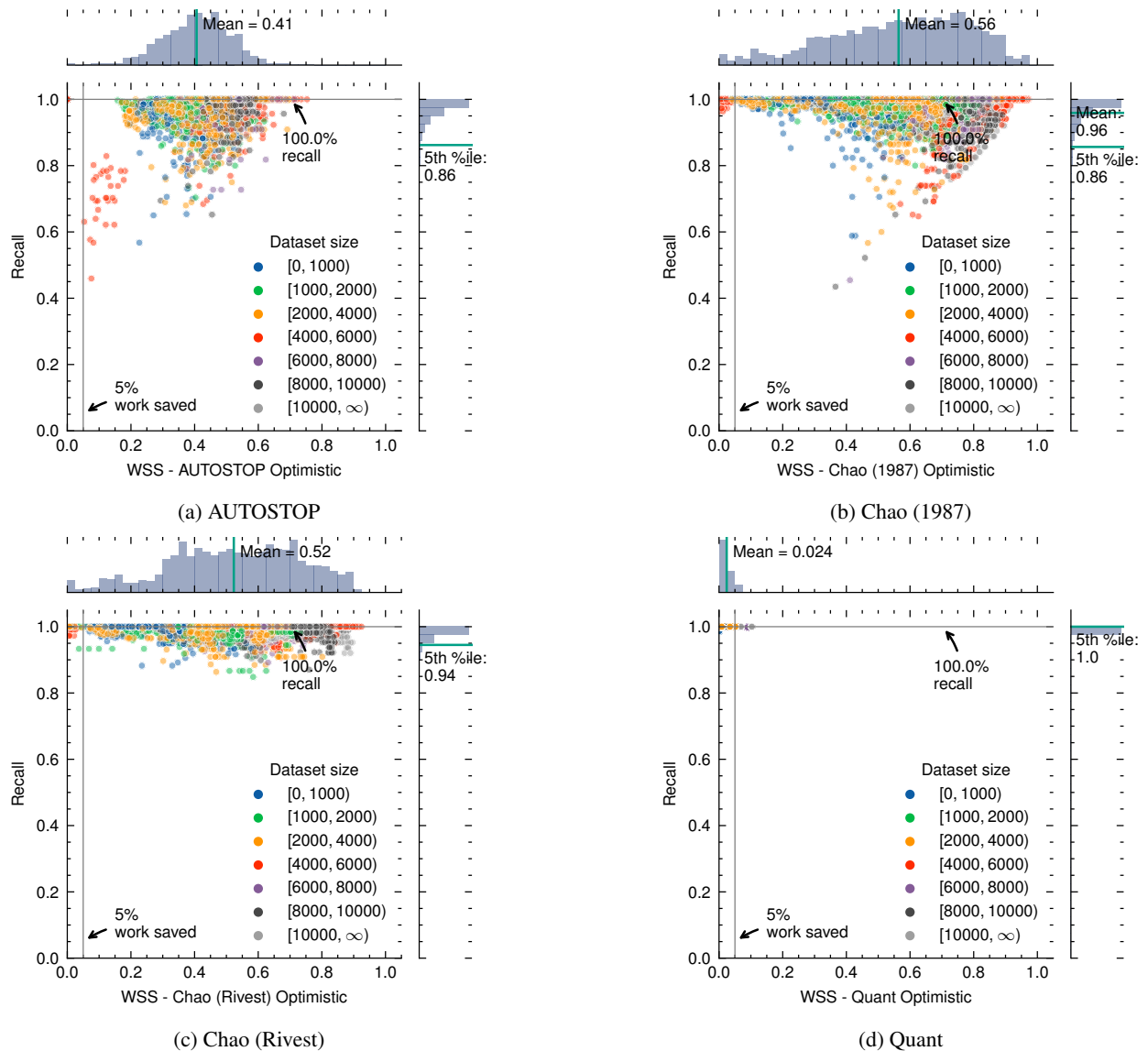


Figure 8: These figures display the results of all runs on all datasets in terms of Work Saved over Sampling and Recall for the Optimistic Stopping criteria with a 100 % recall target. The colors are indicative of the dataset size. The outer graphs show the overall distribution of WSS and Recall. For each method, the mean WSS and the 5th percentile of the recall are shown.

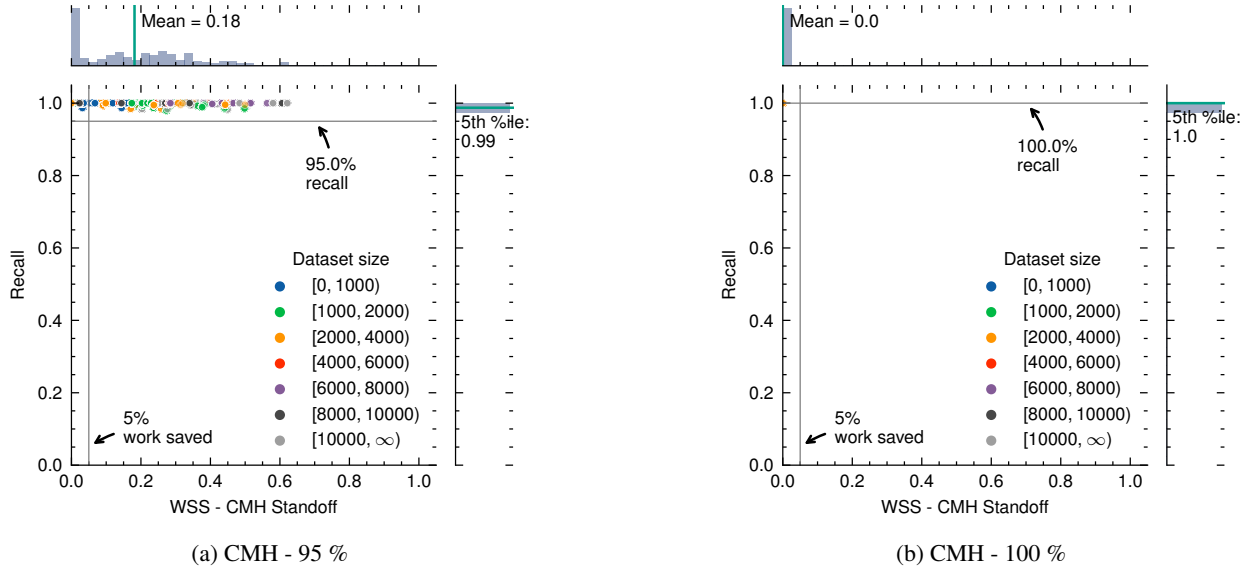


Figure 9: These figures display the results of all runs on all datasets in terms of Work Saved over Sampling and Recall for the CMH criteria.

Table 10: This table shows all metrics recorded for all hybrid methods. The target recall of the CMH method is 95 %, the Target method does not allow the specification of a recall target. Note that for the Effort, Recall, WSS, and loss_{er} , the mean and standard deviation are reported. Furthermore, we report the rate various recall targets are achieved.

	Hybrid Methods	
	CMH	TARGET
Effort (%)	82 ± 16	38 ± 20
Recall (%)	100 ± 0	92 ± 9
WSS (%)	18 ± 15	54 ± 18
loss-er	0.30 ± 0.24	0.10 ± 0.13
Target@70 (%)	100	97
Target@80 (%)	100	90
Target@90 (%)	100	69
Target@95 (%)	100	48
Target@100 (%)	80	19
Triggered (%)	92	100

5.4 Hybrid methods

In this section, we study the results of two hybrid methods: the CMH-Hybrid method [8] and the Target method [15]. While the CMH method discussed in the previous section is a standoff method; this CMH method is not a standoff method, as it consists of two phases: the first phase consists of sampling using AutoTAR until the null hypothesis of the hypergeometric test, as described in Section 2.1.8 is rejected with $\alpha = 0.5$ with a target recall of 95 %. Then, the method proceeds with screening through random sampling. The procedure is stopped until the null hypothesis of the hypergeometric test is rejected, but now with an $\alpha = 0.05$. We studied the scenario where the target recall is 95 % for both phases (given the fact that the 100 % method is never triggered). The results of this test are reported in Table 10. The results are similar to the results reported in [8], which indicated a recall above 95 % accompanied by a WSS of 17 % (in our experiments 18.2%), on a different, but partially overlapping collection of test datasets. The Target method reports a mean recall of 91.85 %, with a WSS of 54.17 %, which just differs 2.03 percent points from ours (our recall is 95.92 % for *Chao (1987) - Conservative*).

For the hybrid version of the CMH method, we can make the same remarks as for the standoff version. The burden of rejecting the null hypothesis of the Hypergeometric test is high, especially when $\alpha = 0.05$. This results in low work savings, although the achieved recall levels is high.

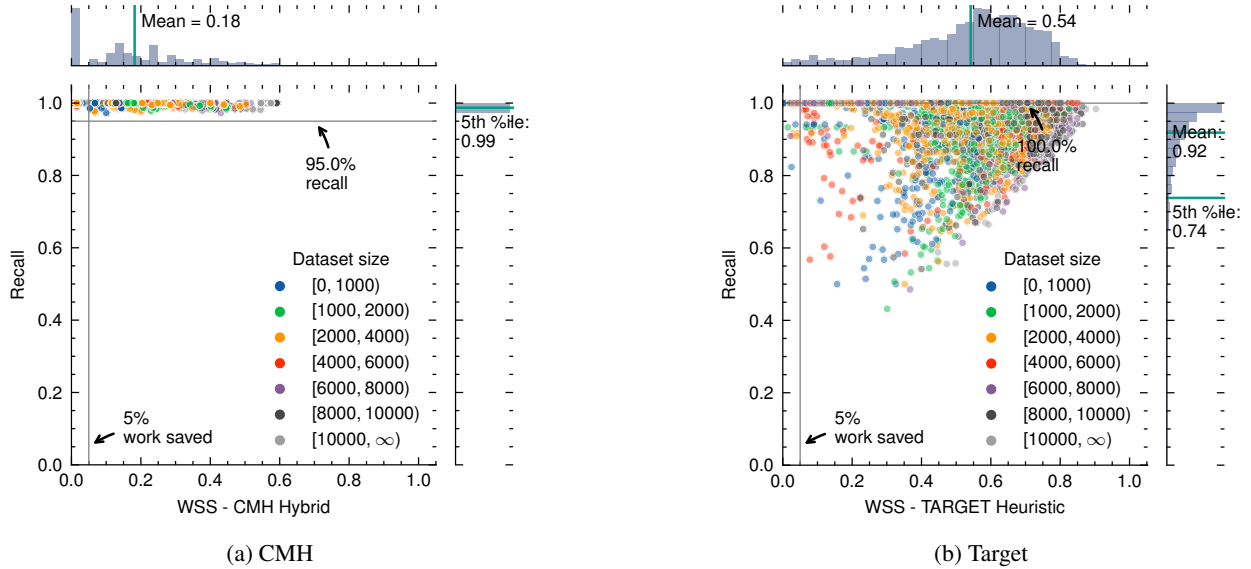


Figure 10: These figures display the results of all runs on all datasets in terms of Work Saved over Sampling and Recall for the hybrid stopping criteria.

Table 11: This table shows all metrics recorded for all heuristic standoff methods. Note that for the Effort, Recall, WSS, loss_{er} , the mean and standard deviation are reported. Furthermore, we report the rate at which various recall targets are achieved, even though these are not specified by the heuristics.

	Standoff Methods					
	Budget	Half	Knee	Rule2399	Stop200	Stop400
Effort (%)	35 ± 20	53 ± 2	82 ± 29	68 ± 32	41 ± 29	52 ± 31
Recall (%)	97.04 ± 3.77	98.96 ± 3.15	99.25 ± 1.96	98.78 ± 4.74	98.13 ± 4.54	99.22 ± 2.23
WSS (%)	62 ± 19	46 ± 4	18 ± 28	31 ± 31	57 ± 28	47 ± 31
loss-er	0.08 ± 0.12	0.10 ± 0.06	0.34 ± 0.25	0.22 ± 0.22	0.07 ± 0.10	0.11 ± 0.14
Target@70 (%)	100	100	100	98	100	100
Target@80 (%)	99	99	100	98	99	100
Target@90 (%)	95	98	100	98	97	99
Target@95 (%)	76	95	93	95	89	96
Target@100 (%)	41	67	77	80	51	72
Triggered (%)	100	100	34	60	94	85

5.5 Standoff Heuristics

The standoff criteria presented in Table 11 do not allow specifying recall targets. All heuristics seem to achieve a high recall; however, some methods are not always triggered. Especially the Knee heuristic, which is only triggered 33.72% of the time. We suspect that this is due to the fact that there are many datasets for which the number of relevant documents is below 100, in which case the Knee has problems [15]. The Budget method, which is the best-performing adaptive heuristic, is more adjusted to this. The (mostly) static heuristics *Half* and *Rule2399* heuristic also offer good results in terms of recall but are not efficient on which AutoTAR can provide a good ranking. Moreover, for datasets containing less than 2399 documents, *Rule2399* does not trigger. In Table 11, the success rates of achieving various recall targets are presented. Our experiments show that the Budget method achieves the 95% target in 76.25% of the experiments (for *Chao (1987) - Conservative*, this is 69.86%). For the 95% recall level, our method does not outperform the Budget method in terms of recall and work savings. However, for the 100% recall level, it does (Budget 40.63% success rate vs. *Chao (1987) - Conservative* 53.09%).

6 Discussion

In the previous section, we described the results of our simulation experiment. In this section, we discuss the results and answer the research questions as posed in Section 4.1.

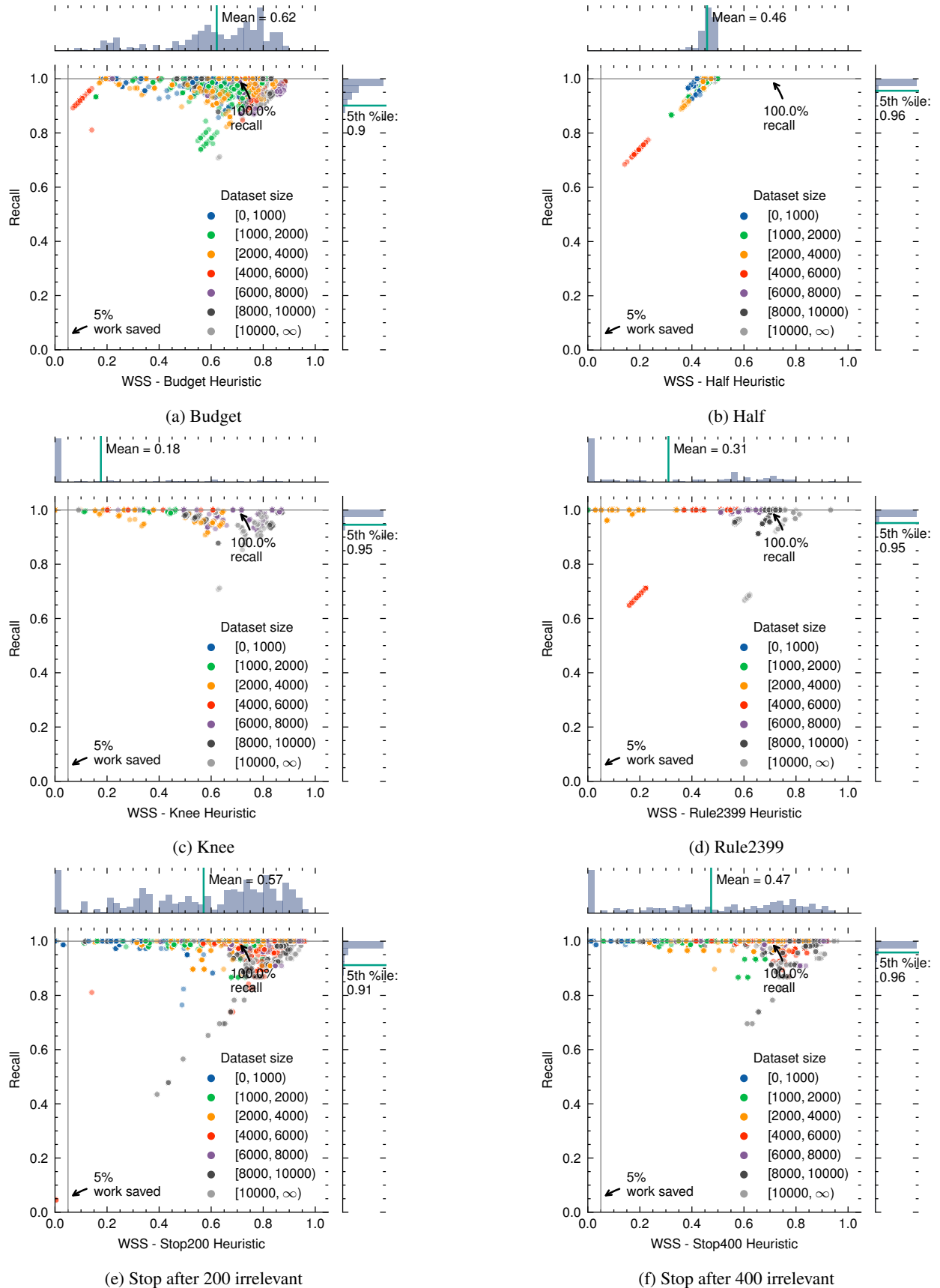


Figure 11: These figures display the results of all runs on all datasets in terms of Work Saved over Sampling and Recall for the standoff heuristic stopping criteria.

6.1 Research Questions

6.1.1 How does our Active Learning strategy perform in terms of the WSS@95 and WSS@100 metrics compared to other methods?

Here, we compare the retrieval capabilities of the three sampling strategies used by the stopping criteria we included in our experiments: AutoTAR, AUTOSTOP, and our Ensemble method. Our method outperforms AUTOSTOP’s sampling strategy but does not outperform AutoTAR in terms of WSS@95 and WSS@100 (see Table 6). One can expect this result as our method performs extra work to enable the use of Population Size Estimation methods to determine the number of relevant documents within the dataset. Moreover, our method selects approximately 20 % of the documents through random sampling, which is not an efficient retrieval strategy. Despite this, our method stays closer to AutoTAR’s performance, which has a mean WSS@95 score of 74.3 % vs. ours (Ensemble) of 63 % and 45.5 % for AUTOSTOP. The overhead introduced by our method is less than AUTOSTOP’s, enabling a reduction of the reviewers’ effort given a good stopping criterion.

6.1.2 Can our stopping criteria help the user achieve its recall target in a timely fashion?

The *Chao (1987) - Conservative* method with a recall target of 95 % has a mean recall of 95.92 % with a Work Saved over Sampling of 56.21 %. For the 100 % target, the *Chao (1987) - Conservative* criterion has a mean recall of 98.02 %, which is slightly below the target recall. The *Chao (Rivest) - Optimistic - 100 %* criterion achieves an average recall of 98.29 % with a WSS of 52.33 %. Although these results do not always deliver a perfect recall of 100 %, the additional burden of using our stopping criterion is not large. When we compare that *effort*, the percentage of the dataset the reviewer read, that was performed up to the point the criterion was triggered (45.96 %) with the effort required for a perfect stopping criterion (apriori knowledge; same selection strategy) 51.97 %, we see that on average the difference is small; this means that when the criterion would have been perfect, a similar effort would have been required. Compared to the most efficient sampling method, AutoTAR, (42.42 %), our method would require 3.53 percentage points more work than the most efficient stopping point.

6.1.3 How reliable are our stopping criteria?

Here, we consider the 95 % and 100 % criteria only. While the *Chao (1987) - Conservative - 95 %* method has a very high mean recall, which is slightly above its target (95.92 %), the amount of times the target has been met is 69.86 %. However, the mean error in predicting the recall is small (3.88 points), so when the target is not achieved, the result is often close to its target. The 100 % recall target is achieved in 53.09 % of the runs for *Chao (1987) - Conservative - 100 %*. The Optimistic methods are less reliable than the Conservative methods due to lacking a CI. Yet, the *Chao (Rivest) - Optimistic - 100 %* method provides excellent results, with a mean recall of 98.29. For all targets, the *Chao (Rivest)* version of this criterion provides better results than the *Chao (1987)* variant. Considering point estimates only, the Rivest method is the obvious choice.

6.1.4 How do our stopping criteria compare to other methods that estimate the size of relevant documents?

Considering the Conservative criteria, our methods outperform AUTOSTOP in terms of WSS and Effort and $loss_{er}$, while retaining a similar recall, although slightly lower. However, on average, the 95 % criterion, reaches a recall above its target. As we can see in Table 7, the number of times the 95 % target is met is higher for AutoSTOP. However, when considering the Optimistic criteria (Table 8), the *Chao (Rivest)* method outperforms AutoSTOP for both the 95 % and the 100 % targets in terms of Work Savings/Effort, Recall, and Reliability. Compared to the *Chao (1987) - Optimistic*, *Chao (1987) - Conservative* methods with a recall target of 100 % also outperform their AUTOSTOP counterparts in terms of WSS and Effort while providing a similar recall. Moreover, for the *Chao (1987) - Conservative* method, our method is more functional compared to AUTOSTOP’s counterpart, as our method is triggered 99.37 % of the time, vs. 55.93 %.

The Quant method [41] overestimates the number of relevant documents with a large number, which results in the fact that this method is not often triggered on time (for the 70 % recall target, the mean recall is already 98.27 %). Considering the Conservative Criterion, the situation worsens due to the overestimation, resulting in this method never being triggered with a high recall target.

6.1.5 How do our stopping criteria compare to other methods that do not provide an estimate?

The main other criteria in our experiments are the Knee Method [15], Budget Method [15], Target Method [15] and the CMH Method [8]. From these methods, the Budget method is very close in terms of recall, outperforming our method in terms of WSS for the 100 % recall target. Considering reliability for the 100 % target, the *Chao (1987) - Conservative* method is a better choice, as the success rate is higher. The Budget method also lacks an estimate for the current level of recall, which does not give the user any information. The CMH method allows the specification of a recall target, yet it suffers a similar fate as the Quant method, as it overestimates the current level of recall.

6.2 Limitations

There are some limitations to the generalizability of our results resulting from how we designed our experiments. We will describe these below.

6.2.1 Dataset selection

We selected datasets for which the number of relevant documents was at least ten. The main reason for this selection is that our method needs at least five relevant documents as seed data. Moreover, the Target and Budget methods expect at least ten relevant documents in the dataset. As all methods are initialized with seed sets that contain five relevant documents, we deemed that including datasets that contain less than ten documents would provide unrealistic results. The consequence is that we cannot extend our findings to all datasets in general; however, the vast majority of the datasets/topics within the corpora included in our study contained more than ten relevant documents. Moreover, we excluded dataset sets with less than 500 documents; however, the merits using TAR in datasets that small is low.

6.2.2 Size and contents of the seed set

As mentioned, our methods need a seed set that contains five relevant documents. The choice of seed documents may influence the results of our method. In our experiments, we aimed to control for the effect of the seed set by repeating the experiment 30 times. In each run, a different seed set was used by using 30 random samples. In a real-world application, the set of seed documents the user provides may contain relevant documents that were selected *not* at random (for example, documents with very similar content). The 30 seed sets are likely not exhaustive enough to capture all these scenarios. Further investigation is needed to control for these scenarios, for example, by increasing the number of experiments or testing for specific seed sets.

6.2.3 Strictness of the Conservative Criterion

In the Conservative Criterion, the recall is calculated using the upper bound of the confidence interval provided by the Estimator. For very high recall targets (e.g., 95 % and especially 100 %), the confidence interval must be negligible to trigger the criterion. In some scenarios, for example, in Figure 4b, the confidence interval of the *Chao (Rivest)* method is only one unit above the point estimates at iteration $t = 8000$, which already coincides with the recall curve. The dataset in question, the *Van Dis* dataset, contains 72 relevant documents so the upper bound of the recall estimate is currently 98.6 % (given an upper bound of 73), thus smaller than the 100 % required by the criterion. This strictness is not only present within our method but also for AUTOSTOP and Quant. This effect is less pronounced in datasets with much more relevant documents, as a single document has less influence on the recall estimate. In a real-world scenario, the user may deviate from the strictness of this criterion and be more lenient; for example, if the point estimates or CI are stable for a long time, the user may accept a slightly smaller predicted recall. It is not trivial to make a stopping criterion that considers this leniency. We opted to be very strict in our criterion, as making the stopping criterion more complex also makes interpreting the results more challenging. In a real-world setting, the results of our methods and AUTOSTOP may be better (or worse) if the user freely decides to stop based on the estimates produced by the estimators.

6.3 Future work

While the results of our experiments are promising, some areas can be further investigated. As mentioned before, the results of the optimistic criteria and, in turn, the point estimates of the Chao Estimators are not as reliable as desired. This result was also observed for AUTOSTOP and the methods the authors tested in [31]. Moreover, the error of the estimators is higher for the lower recall targets, also when we consider the confidence interval. Besides Chao’s Estimator, there are various other Population Size Estimators available (for instance, models that take dependencies between committee members into account), as well as extensions to Chao’s Estimator (e.g., extending the model with covariates found in the dataset). In future research, these methods can be explored and compared to the results presented here.

A second line of work is to perform a user study, which studies how and when users stop the review given the decision by a stopping criterion or an estimate. This study could give insights into how users respond to the predictions by PSE methods, which could be further used to adapt the stopping criterion.

As mentioned above, the seed set’s size and contents may influence the performance of our method. Moreover, in a real-world setting, the user must have five relevant documents available to initialize the AL procedure, which is not always possible. In the scenario where little prior known relevant documents are available, a system can be designed that first finds relevant documents with (for example) AutoTAR and switches to our method when sufficient documents are found. Another option is to generate several distinct synthetic documents using Large Language Models to start the procedure in place of real relevant documents. Research and development of these extensions may improve the applicability of our method.

7 Conclusion

In this work, we used Chao's Population Size Estimator to determine the number of relevant documents during the TAR process. This estimate can indicate if the number of documents that have yet eluded the reviewers. The reviewers can then, given this information, decide to stop the review when a recall target has been satisfied. Population Size Estimators are not directly applicable to the general CAL paradigm. In this work, we presented a novel sampling strategy that makes these methods possible while minimizing the number of irrelevant documents the system proposes. We employed two versions of Chao's estimator, *Chao (1987)*, based on Chao's Moment estimator as presented in [10]. The other, *Chao (Rivest)* is based on Rivest's Poisson Regression version of the former as presented in [32]. The estimates from these methods are then used within a stopping criterion for the review process. For each estimator, we built two criteria: an Optimistic method, which uses the point estimates, while the Conservative uses the 95 % confidence interval. An extensive simulation study showed us that the proposed estimators and criteria work well. The *Chao (Rivest) - Optimistic* method clearly outperforms other estimator-based methods regarding recall and work savings. The *Chao (1987) - Conservative* method challenges methods presented in previous work, as it achieves a similar level recall while improving work savings. We expect that further research into PSE and extensions of our method will improve the reliability and applicability of this method.

References

- [1] Aitkin, M.A. et al. 1989. *Statistical Modelling in GLIM*. Oxford University Press.
- [2] Böhning, D. et al. eds. 2017. *Capture-Recapture Methods for the Social and Medical Sciences*. Chapman and Hall/CRC.
- [3] Bron, M.P. 2024. [Code Repository for Using Chao's Estimator as a Stopping Criterion for Technology-Assisted Review](#). Zenodo.
- [4] Bron, M.P. 2023. [Python Package instancelib](#). Zenodo.
- [5] Bron, M.P. 2024. [Python Package python-allib](#). Zenodo.
- [6] Burnham, K.P. and Overton, W.S. 1978. Estimation of the size of a closed population when capture probabilities vary among animals. *Biometrika*. 65, 3 (Dec. 1978), 625–633. DOI:<https://doi.org/10.1093/biomet/65.3.625>.
- [7] Burnham, K.P. and Overton, W.S. 1979. Robust Estimation of Population Size When Capture Probabilities Vary Among Animals. *Ecology*. 60, 5 (1979), 927–936. DOI:<https://doi.org/10.2307/1936861>.
- [8] Callaghan, M.W. and Müller-Hansen, F. 2020. Statistical stopping criteria for automated screening in systematic reviews. *Systematic Reviews*. 9, 1 (Dec. 2020), 1–14. DOI:<https://doi.org/10.1186/s13643-020-01521-4>.
- [9] Chai, K.E.K. et al. 2021. Research Screener: A machine learning tool to semi-automate abstract screening for systematic reviews. *Systematic Reviews*. 10, 1 (Apr. 2021), 93. DOI:<https://doi.org/10.1186/s13643-021-01635-3>.
- [10] Chao, A. 1987. Estimating the Population Size for Capture-Recapture Data with Unequal Catchability. *Biometrics*. 43, 4 (Dec. 1987), 783. DOI:<https://doi.org/10.2307/2531532>.
- [11] Chao, A. 2005. [Species Estimation and Applications](#). *Encyclopedia of Statistical Sciences*. S. Kotz et al., eds. Wiley.
- [12] Charlier, C.V.L. 1905. Die zweite Form des Fehlergesetzes. *Meddelanden fran Lunds Astronomiska Observatorium Serie I*. 26, (Aug. 1905), 1–8.
- [13] Chun, Y.H. 2006. Estimating the number of undetected software errors via the correlated capture-recapture model. *European Journal of Operational Research*. 175, 2 (Dec. 2006), 1180–1192. DOI:<https://doi.org/10.1016/j.ejor.2005.06.023>.
- [14] Cormack, G.V. and Grossman, M.R. 2015. Autonomy and Reliability of Continuous Active Learning for Technology-Assisted Review.
- [15] Cormack, G.V. and Grossman, M.R. 2016. [Engineering Quality and Reliability in Technology-Assisted Review](#). *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval - SIGIR '16* (New York, New York, USA, 2016), 75–84.
- [16] Cormack, G.V. and Grossman, M.R. 2017. Technology-Assisted Review in Empirical Medicine: Waterloo Participation in CLEF eHealth 2017. *Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14, 2017* (2017).
- [17] Cormack, R.M. 1992. Interval Estimation for Mark-Recapture Studies of Closed Populations. *Biometrics*. 48, 2 (1992), 567. DOI:<https://doi.org/10.2307/2532310>.
- [18] de Bruin, J. et al. 2023. [SYNERGY - Open machine learning dataset on study selection in systematic reviews](#). DataverseNL.
- [19] Edwards, A.W.F. 1972. *Likelihood: An account of the statistical concept of likelihood and its application to scientific inference*. University Press.

- [20] Ferdinands, G. et al. 2020. *Active learning for screening prioritization in systematic reviews - A simulation study*. Open Science Framework.
- [21] Harding, E.F. 1986. Modelling: The Classical Approach. *Journal of the Royal Statistical Society: Series D (The Statistician)*. 35, 2 (1986), 115–134. DOI:<https://doi.org/10.2307/2987516>.
- [22] Ho, T.K. 1995. *Random decision forests*. *Third International Conference on Document Analysis and Recognition* (Montreal, Canada, 1995), 278–282.
- [23] Johnson, N.L. et al. 2005. *Univariate Discrete Distributions*. John Wiley & Sons.
- [24] Kanoulas, E. et al. 2017. CLEF 2017 technologically assisted reviews in empirical medicine overview. *CEUR Workshop Proceedings*. 1866, (Sep. 2017), 1–29.
- [25] Kanoulas, E. et al. 2018. CLEF 2018 technologically assisted reviews in empirical medicine overview: 19th Working Notes of CLEF Conference and Labs of the Evaluation Forum, CLEF 2018. *CEUR Workshop Proceedings*. 2125, (Jul. 2018).
- [26] Kanoulas, E. et al. 2019. CLEF 2019 technology assisted reviews in empirical medicine overview. *CEUR Workshop Proceedings*. 2380, (2019), 9–12.
- [27] Kastner, M. et al. 2009. The capture–mark–recapture technique can be used as a stopping rule when searching in systematic reviews. *Journal of Clinical Epidemiology*. 62, 2 (Feb. 2009), 149–157. DOI:<https://doi.org/10.1016/j.jclinepi.2008.06.001>.
- [28] Ke, G. et al. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems* (2017).
- [29] Kwok, K.T.T. et al. 2020. Virus metagenomics in farm animals: A systematic review. *Viruses*. 12, 1 (2020). DOI:<https://doi.org/10.3390/v12010107>.
- [30] Lewis, D.D. et al. 2021. *Certifying One-Phase Technology-Assisted Reviews*. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (New York, NY, USA, Oct. 2021), 893–902.
- [31] Li, D. and Kanoulas, E. 2020. When to Stop Reviewing in Technology-Assisted Reviews: Sampling from an Adaptive Distribution to Estimate Residual Relevant Documents. *ACM Transactions on Information Systems*. 38, 4 (Oct. 2020), 1–36. DOI:<https://doi.org/10.1145/3411755>.
- [32] Rivest, L.-P. and Baillargeon, S. 2007. Applications and Extensions of Chao’s Moment Estimator for the Size of a Closed Population. *Biometrics*. 63, 4 (2007), 999–1006. DOI:<https://doi.org/10.1111/j.1541-0420.2007.00779.x>.
- [33] Rücker, G. et al. 2011. Boosting qualifies capture–recapture methods for estimating the comprehensiveness of literature searches for systematic reviews. *Journal of Clinical Epidemiology*. 64, 12 (Dec. 2011), 1364–1372. DOI:<https://doi.org/10.1016/j.jclinepi.2011.03.008>.
- [34] Seung, H.S. et al. 1992. *Query by committee*. *Proceedings of the fifth annual workshop on Computational learning theory* (New York, NY, USA, Jul. 1992), 287–294.
- [35] Shemilt, I. et al. 2014. Pinpointing needles in giant haystacks: Use of text mining to reduce impractical screening workload in extremely large scoping reviews. *Research Synthesis Methods*. 5, 1 (2014), 31–49. DOI:<https://doi.org/10.1002/jrsm.1093>.
- [36] Stelfox, H.T. et al. 2013. Capture-mark-recapture to estimate the number of missed articles for systematic reviews in surgery. *The American Journal of Surgery*. 206, 3 (Sep. 2013), 439–440. DOI:<https://doi.org/10.1016/j.amjsurg.2012.11.017>.
- [37] van de Schoot, R. et al. 2021. An open source machine learning framework for efficient and transparent systematic reviews. *Nature Machine Intelligence*. 3, 2 (Feb. 2021), 125–133. DOI:<https://doi.org/10.1038/s42256-020-00287-7>.
- [38] van der Heijden, P.G.M. et al. 2003. Estimating the Size of a Criminal Population from Police Records Using the Truncated Poisson Regression Model. *Statistica Neerlandica*. 57, 3 (2003), 289–304. DOI:<https://doi.org/10.1111/1467-9574.00232>.
- [39] Wallace, B.C. et al. 2013. Active Literature Discovery for Scoping Evidence Reviews. *Proceedings of the KDD Workshop on Data Mining for Healthcare (KDD-DMH’13)* (2013), 14–19.
- [40] Webster, A.J. and Kemp, R. 2013. Estimating Omissions From Searches. *American Statistician*. 67, 2 (May 2013), 82–89. DOI:<https://doi.org/10.1080/00031305.2013.783881>.
- [41] Yang, E. et al. 2021. *Heuristic stopping rules for technology-assisted review*. *DocEng 2021 - Proceedings of the 2021 ACM Symposium on Document Engineering* (Limerick, Ireland, Aug. 2021), 31:1–31:10.
- [42] Yang, E. and Lewis, D.D. 2022. *TARexp: A Python Framework for Technology-Assisted Review Experiments*. *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, Jul. 2022), 3256–3261.

- [43] Yu, Z. and Menzies, T. 2019. FAST2: An intelligent assistant for finding relevant papers. *Expert Systems with Applications*. 120, (2019), 57–71. DOI:<https://doi.org/10.1016/j.eswa.2018.11.021>.

A Dataset statistics

Table 12: Dataset statistics for the SYNERGY corpus [18]

Dataset	# Relevant	# Irrelevant	Size	Prevalence (%)
Appenzeller-Herzog	26	2847	2873	0.9
Brouwer	62	38052	38114	0.2
Chou	15	1893	1908	0.8
Hall	104	8689	8793	1.2
Jeyaraman	96	1079	1175	8.2
Leenaars	583	6633	7216	8.1
Meijboom	37	845	882	4.2
Menon	74	901	975	7.6
Moran	111	5103	5214	2.1
Muthu	336	2383	2719	12.4
Oud	20	932	952	2.1
Radjenovic	48	5887	5935	0.8
Smid	27	2600	2627	1.0
Walker	762	47613	48375	1.6
Wassenaar	111	7557	7668	1.4
Wolters	19	4261	4280	0.4
van Dis	72	9056	9128	0.8
van de Schoot	38	4506	4544	0.8
van der Valk	89	636	725	12.3
van der Waal	33	1937	1970	1.7

Table 13: Dataset statistics for the CLEF 2017 corpus [24]

Dataset	# Relevant	# Irrelevant	Size	Prevalence (%)
CD009135	77	714	791	9.7
CD008081	26	944	970	2.7
CD010023	52	929	981	5.3
CD009944	98	1064	1162	8.4
CD008691	67	1243	1310	5.1
CD007427	59	1398	1457	4.0
CD010632	27	1472	1499	1.8
CD009020	154	1422	1576	9.8
CD009185	92	1523	1615	5.7
CD009551	46	1865	1911	2.4
CD011134	200	1738	1938	10.3
CD009372	25	2223	2248	1.1
CD007394	92	2450	2542	3.6
CD009647	56	2729	2785	2.0
CD008054	206	2940	3146	6.5
CD010438	30	3211	3241	0.9
CD009323	98	3757	3855	2.5
CD008803	99	5121	5220	1.9
CD010173	23	5472	5495	0.4
CD010276	54	5441	5495	1.0
CD009519	104	5867	5971	1.7
CD009579	138	6317	6455	2.1
CD009925	460	6071	6531	7.0
CD009591	143	7847	7990	1.8
CD010653	45	7957	8002	0.6

(continued)

Dataset	# Relevant	# Irrelevant	Size	Prevalence (%)
CD011984	442	7738	8180	5.4
CD011975	604	7582	8186	7.4
CD008782	45	10462	10507	0.4
CD011548	109	12591	12700	0.9
CD010339	114	12689	12803	0.9
CD009593	63	14844	14907	0.4

Table 14: Dataset statistics for the CLEF 2018 corpus [25]

Dataset	# Relevant	# Irrelevant	Size	Prevalence (%)
CD012009	37	499	536	6.9
CD008759	60	872	932	6.4
CD011431	297	885	1182	25.1
CD011912	36	1370	1406	2.6
CD008892	69	1430	1499	4.6
CD010657	139	1720	1859	7.5
CD008122	272	1639	1911	14.2
CD011053	12	2223	2235	0.5
CD010864	44	2461	2505	1.8
CD010502	229	2756	2985	7.7
CD011926	40	4010	4050	1.0
CD010296	53	4549	4602	1.2
CD009175	65	5579	5644	1.2
CD011126	13	5987	6000	0.2
CD012010	290	6540	6830	4.2
CD011515	127	7117	7244	1.8
CD012599	575	7473	8048	7.1
CD010680	26	8379	8405	0.3
CD008587	79	9073	9152	0.9
CD011686	64	9665	9729	0.7
CD012179	304	9528	9832	3.1
CD012281	23	9853	9876	0.2
CD012165	308	9914	10222	3.0
CD010213	599	14599	15198	3.9

Table 15: Dataset statistics for the CLEF 2019 corpus [26]

Dataset	# Relevant	# Irrelevant	Size	Prevalence (%)
CD001261	72	499	571	12.6
CD012551	68	523	591	11.5
CD007867	17	926	943	1.8
CD012669	71	1189	1260	5.6
CD009069	78	1679	1757	4.4
CD009642	62	1860	1922	3.2
CD008874	118	2264	2382	5.0
CD010753	29	2510	2539	1.1
CD010558	37	2778	2815	1.3
CD009044	11	3158	3169	0.3
CD012661	192	3175	3367	5.7

(continued)

Dataset	# Relevant	# Irrelevant	Size	Prevalence (%)
CD012069	320	3159	3479	9.2
CD006468	52	3821	3873	1.3
CD011787	111	4258	4369	2.5
CD012080	77	6566	6643	1.2
CD012567	11	6724	6735	0.2
CD010038	23	8844	8867	0.3
CD011768	54	9104	9158	0.6
CD011686	64	9665	9729	0.7