Combining Large Language Model Classifications and Active Learning for Improved Technology-Assisted Review

Michiel P. Bron[1,2,*], Berend Greijn[1,3], Bruno Messina Coimbra[3], Rens van de Schoot[3] and Ayoub Bagheri[3]

[1]Utrecht University, Department of Information and Computing Sciences, Faculty of Science, Utrecht, The Netherlands
[2]The Netherlands' National Police, The Hague, The Netherlands
[3]Utrecht University, Department of Methods and Statistics, Faculty of Social Sciences, Utrecht, The Netherlands

Abstract

Technology-assisted review (TAR) is software that aids in high-recall information retrieval tasks, such as abstract screening for systematic literature reviews. Often, TAR systems use a form of Active Learning (AL); during this process, human reviewers label documents as relevant or irrelevant according to a screening protocol, while the system incrementally updates a classifier based on the reviewers' previous decisions. After each model update, the system uses the classifier to rerank the remaining workload by prioritizing predicted relevant documents over irrelevant ones, enabling a reduced workload. Recently, studies have been performed that study the ability of solely using Large Language Models (LLMs) to perform this task by supplying the LLM prompts that contain the task, screening protocol, and a document from the corpus. The LLM then provides a classification of the document in question. While the results of these studies are promising, the LLM's predictions are not error-free, resulting in a recall or precision that is lower than desired. In this work, we propose a new Active Learning method for TAR that integrates the results of the LLM in the review process that may correct some of the shortcomings of the LLM results, leveraging a reduced workload with respect to current TAR systems.

Keywords

technology-assisted review, active learning, large language model, information retrieval, weak supervision

1. Introduction

Technology-assisted review (TAR) is software that aids in high-recall (information) retrieval (HRR) tasks. An example of such a task is performing a Systematic Literature Review (for example, in medicine [1]), but there are also applications in the legal domain (e.g., e-Discovery [2], but also the processing of Freedom of Information Act Requests, criminal investigation, etc.). For all these search tasks, it is important that nearly all relevant information is found, so these have a recall target of 75 – 100 % [3].

In these extensive studies, the researchers, attorneys, or investigators gather evidence or information by screening documents stored in large databases or corpora. The task is to find nearly all information relevant to the subject of the investigation. In the case of Systematic Literature Reviews, the researcher starts by using specialized search queries to select documents from databases. Formulating these queries is not a trivial task, as it is the objective to capture (nearly) all relevant documents. These queries should not be too restrictive to minimize the chance that a relevant document is missed; researchers often use disjunctions rather than conjunctions. Consequently, the resulting set of candidate documents the researchers process is often enormous, while the prevalence of relevant documents within these sets can be very low.

More formally, we can specify this task as follows: we have a dataset $\mathcal{D}$ containing all the candidate documents found after the initial keyword search. During the review process, these documents are read by the domain experts and labeled as either *relevant* or *irrelevant*. Read documents are referred to as

IAL@ECML-PKDD'24: 8th Intl. Worksh. & Tutorial on Interactive Adaptive Learning, Sep. 9th, 2024, Vilnius, Lithuania
*Corresponding author
m.p.bron@uu.nl (M. P. Bron); b.greijn@uu.nl (B. Greijn); b.messinacoimbra@uu.nl (B. Messina Coimbra); a.g.j.vandeschoot@uu.nl (R. van de Schoot); a.bagheri@uu.nl (A. Bagheri)
0000-0002-4823-6085 (M. P. Bron); 0000-0001-5092-9625 (B. Messina Coimbra); 0000-0001-7736-2091 (R. van de Schoot); 0000-0001-6366-2173 (A. Bagheri)

Table 1
Typical process statistics for Systematic Literature Reviews. Users query multiple databases using keyword search, which yields a candidate set of records $\mathcal{D}$ which are screened. In this work, we aim to optimize this screening phase. After the title-abstract screening phase, the reviewers will read the full-text of the remaining set of documents ($\mathcal{D}^+$), which will determine the definitive eligibility for inclusion in the review or meta-analysis.

Databases	Keyword Search ($\mathcal{D}$)	Title Abstract ($\mathcal{D}^+$)	Full-text
$> 10^8$	10000	180	50

labeled. During the process, we maintain two sets $\mathcal{L}^+$ and $\mathcal{L}^-$ for the labeled relevant and irrelevant documents. The remaining unlabeled documents belong to the set $\mathcal{U}$. Traditionally, researchers screened all documents in $\mathcal{D}$. Technology-Assisted Review are then systems or algorithms that aid the reviewers in reducing the reviewing workload [4], while still aiming to find all relevant documents $\mathcal{D}^+$.

Early TAR methods consisted of first creating a randomly sampled subset of $\mathcal{D}$ and training a classifier on the labeled dataset $\mathcal{L}$. Then, that classifier is used to classify the remaining documents in $\mathcal{U}$ [5]. Many recent TAR systems use a form of *Active Learning* to update the classifier after each or several review decisions iteratively [6, 7, 8, 9, 10, 11]. 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 each document's relevancy or inclusion status. In canonical Active Learning, the selection strategy aims to select the "most informative" examples from the perspective of the classifier. An example of such a strategy is Uncertainty Sampling [12]. The goal of canonical AL is to create a good inductive classifier, that can be used to classify previously unseen documents not found in the pool of potential training examples.

Within TAR, the model is used in a transductive setting only, i.e., the model is only used to retrieve the relevant data within the pool. The model is not used after the retrieval task has been completed [13]. Many TAR systems (e.g., [14, 7, 9, 8]) use *relevance sampling* [15], a greedy batch sampling method that selects a batch $\mathcal{B}$ with the top-k documents with the highest probability of belonging to the class of relevant documents according to the trained model. After the annotation of each document in $\mathcal{B}$, the model is retrained, and a new ranking for the documents in $\mathcal{U}$ is produced. The objective is then to find all the remaining *unlabeled* relevant documents belonging to the set $\mathcal{U}^+$, while minimizing reading documents that belong to the set $\mathcal{U}^-$.

For abstract screening, $\mathcal{D}$ consists of title-abstract pairs, which the reviewers for eligibility for the researcher's systematic review or meta-analysis. The researchers follow a protocol that consists of inclusion and exclusion criteria to determine the eligibility of a record (in Section 4 - Figure 2, an example of such a protocol is displayed). This protocol should be followed strictly to ensure fairness and mitigate bias. Typical statistics of this process are given in Table 1.

Eligibility cannot always be determined from the title-abstract pair only due to the limited amount of information stored there, so reading the full-text of the paper is necessary to decide on definitive eligibility. Reading the full-text is associated with a high cost. Title-abstract screening greatly reduces the number of papers that have to be screened fully. TAR systems then aid in reducing the number of irrelevant title-abstract pairs so that not all records have to be screened.

Recently, methods have been proposed that use generative Large Language Models (LLMs) systems to perform title-abstract screening (inter alia [16, 17, 11, 18]). The main approach is to prepare a prompt that delineates the task and specifies the criteria, followed by the title and abstract. After supplying the prompt to the LLM, it will provide an answer and a decision on the inclusion status of that record. Obtaining results can be automated by making a program or script that automatically processes a dataset through the models' API. In [16], the authors report a mean accuracy of $\pm$ 90 % with a recall of 76 %. However, the performance varied per dataset, with recall scores ranging from 59 % to 100 %. In another study, the reported precision is low for some datasets [11], which may result in a higher

screening workload than current AL-based systems offer.

LLMs are prone to hallucination, where the LLMs generate responses that seem plausible but are factually incorrect [19]. Moreover, LLMs are very eager to provide an answer even though there is no information provided in the LLM's training data or within the prompt to give a good answer [20]. With the current limitations, using the LLMs to determine the inclusion status of the title and abstract pairs may not be reliable enough.

In [21], the authors propose a system that combines (canonical) AL with Weak Supervision (e.g., noisy labels provided by a black-box model). To our knowledge, a TAR method that combines AL and noisy labels (e.g., from an LLM or another model) has not been presented yet. In this work, we propose a system that combines LLM classifications and Active Learning to improve the efficacy of the TAR procedure. Our main contributions can be summarized as follows:

1. A system that provides more detailed LLM classifications for all the criteria in the screening protocol instead of a single binary label for inclusion.
2. A system that makes LLM classifications more transparent by making the LLM provide a detailed explanation for each classification.
3. An Active Learning method that incorporates the LLM results to reduce the workload of the review.
4. A preliminary experimental evaluation of our method and several suggestions for future work.

In the following section, we will briefly overview previous work on TAR, LLM classification and techniques for combining weak supervision and AL. After that, we will explain our method, which consists of an LLM classifier and an Active Learning method that incorporates its predictions. As the LLM classifier assigns labels to each specific criterion, we introduce a case study in which we study a novel dataset that contains labels for each record at the criterion level, enabling us to assess the performance of our method. Finally, we will present our initial experiments and results, followed by a discussion and suggestions for future work.

2. Related Work

Most TAR approaches are based on the Continuous Active Learning (CAL) algorithm (see Algorithm 1) [22]. In this process, a model is trained on the documents that have already been reviewed. The model is then used to rerank the remaining documents in $\mathcal{U}$. Several CAL procedures [8, 23, 9, 7] 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 description of the research topic as a *pseudo-document*. Additionally, one example of an irrelevant document is needed.

AutoTAR [14] extends the CAL procedure, which is still considered state-of-the-art and has been included in many studies as a baseline, for example, when studying ideal performance vs. the performance of a stopping criterion [10, 24, 25]. 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; a fair assumption, given the low prevalence of relevant documents in most datasets. ASReview [9], open-source TAR software specialized for abstract screening, resamples the data to improve the performance in the presence of imbalanced training data. FASTREAD2 [7] modifies the CAL procedure with the goal of detecting human errors during the review procedure, as noisy human labels may occur [26].

CAL, as described in Algorithm 1, leaves the question of a Stopping Criterion open (i.e., the StoppingCriterion procedure, line 15 in Algorithm 1, is not given). Formulating a good stopping criterion is an area of active research. Some practitioners use pragmatic criteria based on time constraints or stop when the returns diminish (e.g., when TAR proposes k irrelevant documents in a row; however, specifying k is target and topic dependent)[27]. Several heuristics [14, 7, 28, 27] (for example, characteristics of the recall curve) have been proposed, as well as methods that change the CAL procedure to allow the use of statistical methods that predict when a recall target has been achieved (inter alia [10, 23, 24]).

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 k. 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.

procedure CAL($\mathcal{D}, \mathcal{U}, \mathcal{L}^+, \mathcal{L}^-, C, k$)
  $S \leftarrow$ **false** ▷ Variable indicating whether CAL can be stopped
  while $|\mathcal{U}| > 0$ **and not** S **do**
    C.Fit($\mathcal{L}^+, \mathcal{L}^-$)
    $\mathcal{B} \leftarrow$ Select($\mathcal{U}, C, k$)
    for $d \in \mathcal{B}$ **do**
      $y \leftarrow$ Review(d) ▷ Performed by the human reviewer
      if $y =$ Relevant **then**
        $\mathcal{L}^+ \leftarrow \mathcal{L}^+ \cup \{d\}$
      else
        $\mathcal{L}^- \leftarrow \mathcal{L}^- \cup \{d\}$
      end if
      $\mathcal{U} \leftarrow \mathcal{U} \setminus \{d\}$
    end for
    $S \leftarrow$ StoppingCriterion($\mathcal{D}, \mathcal{U}, \mathcal{L}^+, \mathcal{L}^-, C, k$)
  end while
  return $\mathcal{L}^+, \mathcal{L}^-$
end procedure
procedure Select($\mathcal{U}, C, k$)
  $\mathbf{P} \leftarrow C$.Predict($\mathcal{U}$) ▷ Returns the relevance score for all d in $\mathcal{U}$
  $\mathbf{R} \leftarrow$Rank($\mathcal{U}, \mathbf{P}$)
  $\mathcal{B} \leftarrow$ Head($\mathbf{R}, \mathcal{U}, k$) ▷ Gets the top-$k$ documents
  return $\mathcal{B}$
end procedure

The classifiers that are used in these systems are often based on classical Machine Learning algorithms like Multinomial Naïve Bayes, Logistic Regression (AutoTAR), and Support Vector Machines combined with TF-IDF features. However, some recent studies explore using neural networks and deep learning (e.g., [3, 29]).

This work focuses on applying TAR to aid abstract screening for systematic reviews. In this field, state-of-the-art systems can find (nearly all) after screening 5 – 40 % of the corpus by using this general methodology [8, 9], but performance is dataset and query dependent. A frequently used metric to assess the efficacy of TAR systems metric is *Work Saved over Sampling* (WSS) which indicates the work savings over the use of random sampling (i.e., traditional screening) [5]. This metric can be calculated after the procedure was terminated after a stopping criterion was triggered or when a recall target has been achieved according to the ground truth; WSS@95, which indicates the the work savings over random sampling at the moment when 95 % recall is achieved, is a frequently used metric for TAR systems targeting Systematic Literature Reviews (inter alia [23, 8, 9]).

In contrast to the AL-based methods, after the popularization of generative Large Language Models like ChatGPT-3.5 and GPT-4 [30], systems have been proposed that use these models to perform screening tasks. The main approach is to prepare a prompt that delineates the task and specifies the criteria, followed by the title and abstract [16, 17, 31, 18]. Many approaches use ChatGPT-3.5 or GPT-4 [16], several [11, 18] use open-source LLMs such as Llama 2 [32]. In [18], a large simulation study is performed to assess the performance of several LLMs on popular TAR datasets (CLEF2017, CLEF2018, CLEF2019) [33, 34, 35]; however, in this study, the LLM predicts the inclusion status only on the title of the systematic review, not its screening protocol (the CLEF datasets do not offer a lot of information on

the screening protocol, although the keyword searches are available and a topic description is available). Contrary to the other methods, [18] compares the next token probabilities of *yes* and *no* (which are used to indicate the inclusion decision), which can be used as a measure of confidence.

There have been several works that combine or compare LLMs and Active Learning. For example, in [36], the authors compare the performance of LLMs and models that have been trained with Active Learning. One of the findings is that with a limited number of labeled documents, the AL-trained models outperform the LLMs that perform zero-shot classification despite being significantly smaller in terms of training parameters. In [37], a method is proposed that integrates an LLM as an annotator for the creation of *Named Entity Recognition* (NER) models in underrepresented languages (e.g., African languages). Another work presents a method that generates synthetic data with LLMs, which are used to select the most interesting examples from the pool of unlabeled documents[38].

In [21], the authors present a method that combines AL with Weak Supervision and Transfer Learning. They present their results on training a classifier for classifying financial transactions (text data) in the presence of a black-box model (BBM) (a rule-based system). In this study, an annotator model is trained on agreement labels between the black-box model and the oracle's labels for each iteration along the typical classifier. The annotator model is used to determine per selected instance if the BBM's label can be trusted and accepted or if the human oracle should label it instead. With this method, the authors show that they could significantly lower annotation costs while retaining an accuracy close to the traditional AL setting.

3. Methodology

In this section, we describe the general architecture of our method. Our TAR procedure consists of two main components: a method to obtain classifications from the LLM and an Active Learning procedure that is used to rank the records during the review phase. Our AL procedure, *LLM+CAL*, uses the results of the LLM to reduce the review workload further.

3.1. Obtaining LLM classifications

In [31, 16], a prompt contained the task and the full screening protocol. The task for the LLM was then to answer only with a final inclusion decision (e.g., choose between *INCLUDE* or *EXCLUDE*). This setup can be regarded as a black-box system, as it is impossible to determine any of its reasoning for making the decision. Also, the LLM does not provide any information about the confidence in its prediction besides a probability of predicting the token that represents the word *INCLUDE* or *EXCLUDE* over the space of all possible output tokens.

Chain-of-thought prompting is a method to improve the accuracy of LLMs when performing complex reasoning. With this method, it is specifically requested in the prompt to *think step-by-step* in addition to a few examples of appropriate answers. The aim is to let the LLM reason about its "thought process" verbosely, which results in a higher probability that the final answer is correct [39]. By adjusting the prompt to let the LLM respond with chain-of-thought steps in a structured way, we aim to make the process more transparent for the reviewer. In addition, we ask the LLM to provide *rationales* (i.e., select fragments cited directly from the record in question), which enables tracing the decision to the source document. In Figure 1, we display the prompt template that we use in our experiments, which contains - besides the instruction - a few examples of appropriate answers. We wrote a parser that parses the LLM answer into a structured datatype. In a real-world application, the rationales can be used to highlight fragments in the abstracts used in the LLM's decision-making, enabling easy verification and correction for the end-user in an annotation interface. Another significant difference between the studies in earlier work and ours is that we consider each criterion in the protocol separately. We noticed many classification errors in initial experiments when the whole screening protocol was considered. We list some major error categories below:

Hallucination. The model makes up factually incorrect but seemingly plausible answers.

Missing knowledge or context. The model does not know enough information about a topic that a human reviewer might know (e.g., technical jargon)

Incorrect reasoning. The information extraction works correctly, but the inclusion rules are not followed, causing a misclassification.

Ignoring instructions. Only a part of the screening protocol was used according to the LLM's chain-of-thought response. Some LLMs have problems following all instructions in the prompt, especially when the instructions are long and complex. Larger models like GPT-4 are less prone to this but have a higher computational and financial cost.

Often, the LLM followed the protocol partially: consider a dataset with four criteria, the LLM considered three criteria correctly but mistakenly ignored one of them, causing a misclassification of

```
ASSIGNMENT: You are a helpful assistant who helps screen abstracts and titles of scientific papers. You answer
questions by citing evidence in the given text followed by a YES or NO or UNKNOWN decision. When there is no
evidence in the title and abstract, decide with UNKNOWN. Only answer with NO if there is absolute evidence given
that the answer is NO. In the absence of evidence or when nothing is mentioned, always answer UNKNOWN. Use the
following format:

REASONING: (Think step by step to answer the question; use the information in the title and abstract and
work your way to an answer. Your full reasoning and answer should be given in this field)
EVIDENCE: (List sentences or phrases from the title and abstract used to answer the question in the previous field.
Answer in bullets (e.g., - "quoted sentence"). Each quoted sentence should have its own line. If there is no evidence,
write down []). In this field, only directly cite from the TITLE and ABSTRACT fields. DO NOT USE YOUR OWN
WORDS, AND ADHERE TO THE LIST FORMAT!
ANSWER: (Summarize your answer from the REASONING field with YES or NO or UNKNOWN. DO NOT WRITE
ANYTHING AFTERWARDS IN THIS FIELD.)
Write nothing else afterward.

EXAMPLE RESPONSE 1:
REASONING: To answer the question, we need to find information about [...]. The title and the abstract mention that
[...]. Furthermore, the study aims to [...], suggesting that this is indeed the case. So, the answer to this question is
YES.
EVIDENCE:
- "Sentence evidence 1"
- "Sentence evidence 2"
ANSWER: YES

EXAMPLE RESPONSE 2:
REASONING: To answer the question, we need to find information about [...]. The title and abstract say something
about [...] but do not mention anything about [...]. As there is no definitive evidence, the answer should be
UNKNOWN.
EVIDENCE: []
ANSWER: UNKNOWN

EXAMPLE RESPONSE 3:
REASONING: To answer the question, we need to find information about [...]. The title and abstract say something
about [...]. This statement rules out that [...]. As there is evidence to the contrary, the answer should be NO.
EVIDENCE:
- "Sentence evidence 1"
ANSWER: NO

TITLE: {title}
ABSTRACT: {abstract}
QUESTION: {question}
```

Figure 1: The prompt template that was used during the experiments. The first part delineates the task. The second paragraph contains instructions on formatting responses, with detailed instructions per field. Next, three example responses are given. Finally, the title, abstract, and one of the criteria are supplied.

the whole instance due to a mistake. This setup makes it challenging to detect failures due to a specific criterion. Mistakes become only apparent by combing through the (semi-structured) LLM answers containing information on all criteria.

We aim to mitigate this by considering each criterion separately, making the set of instructions shorter and less complex, which results in a higher accuracy. The system can then infer the inclusion status of a record by applying a simple logical formula to the model's decision on the criteria (for example, Figure 2).

Despite the reduced complexity, it is still possible that the LLMs make classification errors, for example, due to hallucination, possibly because of missing knowledge. We hypothesize that these errors will not always happen at random, especially for the latter cause. Suppose the LLM makes an incorrect classification for a specific criterion due to missing knowledge. In that case, the LLM will likely make a similar mistake for instances similar to the one in question. Collecting the rationales and chain-of-thought fragments of misclassifications and training models on them might aid in predicting when the LLM makes a mistake or a correct decision.

We used LangChain [40] to build our LLM classification pipeline. This package enables us to target multiple Large Language Models. In our experiments, we only worked with ChatGPT-3.5 (specifically, version `0301`); however, the method can be applied to GPT-4 or models of other vendors, such as open-source models published on repositories like HuggingFace [41].

3.2. Active Learning method

As in canonical TAR, we represent each document as a high-dimensional vector. A typical feature extraction method is a bag-of-words method like TF-IDF that TAR systems frequently use. Combining sparse feature matrices and classical machine learning methods offers fast retraining and reranking of the documents in $\mathcal{U}$. The AutoTAR baseline uses TF-IDF combined with a Logistic Regression classifier. In our approach, we will also use TF-IDF and Logistic Regression to ensure that changes in performance are not due to changes in the document representation.

During the process, the labeling task is specified as follows: we have a feature space $\mathcal{X}_{\text{tiab}}$, which contains the feature vectors of the title-abstract (tiab) records. Each document presented to the oracle gets, for each of the criteria (see Figure 2), a label in the space $\mathcal{Y}_{\text{crit}_i} = \{+, ?, \neg\}$ corresponding to *True (Yes)*, *Unknown*, *False (No)*. The option *Unknown* is vital in this phase, as it is not always the case that the information needed to determine eligibility for a criterion is present in the title and abstract.

Our method, LLM+CAL, consists of two phases: the first phase is called LLMPreferred, which is - in essence - a version of the method AutoTAR, but in this version constrained to select from the unlabeled documents that are included by the LLM ($\mathcal{U} \cap \mathcal{L}_{\text{LLM}}^{\{+,?\}}$). As initial training data, the whole screening protocol is given in addition to a random sample of 100 LLM-excluded documents ($\mathcal{L}_{\text{LLM}}^{-}$). This phase is applied until 25 consecutive irrelevant documents are proposed, which might indicate that the set of relevant documents may be exhausted.

Because the possibility exists that there are relevant documents that the LLM does not find, we will switch to the CriteriaWSA method, which can query all documents within $\mathcal{U}$. First, all labeled data $\mathcal{L}$ from the first phase is transferred to this method. Then, several machine learning models are trained:

Inclusion Judgment Classifier. A Binary Classifier trained on the labeled data after transforming the data to $\mathcal{Y}_{\text{binary}} = \{+, \neg\}$, trained on the data in $\mathcal{L}$, in a similar fashion as AutoTAR. The criterion judgments are transformed using the formula specified in Figure 2, which will result in a label in the space $\mathcal{Y}_{\text{ternary}} = \{+, ?, \neg\}$. We can then transform $\mathcal{Y}_{\text{ternary}}$ to $\mathcal{Y}_{\text{binary}}$ by changing each $?$ into a $+$.

Acceptance Classifier. A Binary Classifier that determines Acceptance for each inclusion criterion. This is similar to a method presented in [21]. Here, for each criterion i, we obtain binary agreement labels $z \in \mathcal{Z}$, where $\mathcal{Z} = \{0, 1\}$. This is determined by comparing the LLM predictions and the labeled data in $\mathcal{L}^i$: each instance receives a label *Accept* (1) if the LLM prediction agrees with the human-annotated label. Otherwise, the label *Reject* (0) is given. However, contrary to the

other models in our system and the method in [21], the model is not trained on the Title-Abstract records ($\mathcal{X}_{\text{tiab}}$), but on the LLM's reasoning fragments $\mathcal{X}_{\text{ans}_i}$ (see Figure 4 for example data) of criterion i.

Given a TAR task that has four inclusion criteria ($\{a, b, c, d\}$), we obtain the following pairs for each labeled for each labeled record:

- $\mathcal{X}_{\text{tiab}} \times \mathcal{Y}_{\text{crit}_a} \times \mathcal{Y}_{\text{crit}_b} \times \mathcal{Y}_{\text{crit}_c} \times \mathcal{Y}_{\text{crit}_d}$
- $\mathcal{X}_{\text{tiab}} \times \mathcal{Y}_{\text{binary}}$
- $\mathcal{X}_{\text{tiab}} \times \mathcal{Y}_{\text{ternary}}$
- $\mathcal{X}_{\text{ans}_a} \times \mathcal{Z}_a$
- $\mathcal{X}_{\text{ans}_b} \times \mathcal{Z}_b$
- $\mathcal{X}_{\text{ans}_c} \times \mathcal{Z}_c$
- $\mathcal{X}_{\text{ans}_d} \times \mathcal{Z}_d$

During each annotation round, a batch of ten documents is given to the oracle using relevance sampling based on the ranking produced by the inclusion judgment classifier. The batch size of ten is an initial default value for this parameter. Smaller, larger, and dynamic batch sizes can be explored in future work. Another ten documents are sampled based on a ranking that is based on the predictions of the LLM and the Acceptance Classifier using the following equation:

$$\text{score}_i(\hat{y}_i^{\text{LLM}}, p_i^{\text{acc}}) = \begin{cases} 0.75 + 0.25 p_i^{\text{acc}} & \text{if } \hat{y}_i^{\text{LLM}} = + \\ 0.5 + 0.25 p_i^{\text{acc}} & \text{if } \hat{y}_i^{\text{LLM}} = ? \\ 0.5(1 - p_i^{\text{acc}}) & \text{if } \hat{y}_i^{\text{LLM}} = \neg \end{cases} . \tag{1}$$

Equation 1 is calculated for each study criterion i, where $\hat{y}_i^{\text{LLM}}$ is the LLM's prediction for criterion i and p_i^{acc} is the corresponding acceptance probability. The mean of those scores is calculated for each of the unlabeled documents. Then, this score is used to rank the remaining documents in $\mathcal{U}$. The rationale behind Equation 1 is that instances with a higher probability to be relevant (instances with criteria that have more *True* labels) are put before documents that have *Unknown* labels, followed by documents that have *False* labels. Labels that have *False* labels and a low acceptance probability will have a higher probability of being selected than documents with *False* labels that are certain. For the *True* and *Unknown* labels, the inverse holds if there is a higher acceptance probability, they are preferred over instances with lower acceptance probability. This is still an initial formulation that may not always work optimally; other options can be explored in future research.

After this batch of twenty documents has been prepared, they are given to the oracle for labeling unless the LLM has found exclusionary evidence for a specific criterion and its acceptance probability is above 80 % (unless that criterion is a reason for exclusion for all remaining documents in $\mathcal{U}$); these examples are skipped but may be proposed again in another round if the acceptance probability drops below 80 %.

This process is repeated until a stopping criterion is triggered, the oracle decides to stop the review, or $\mathcal{U}$ is exhausted. In our experiments, we will stop querying after reviewing $|\mathcal{L}_{\text{LLM}}^{\{+,?\}}|$ documents.

4. Case Study

In this work, we compare the performance of various TAR methods on a dataset that is collected for a systematic review (at the time of writing in preparation) that aims to identify common latent groups or classes of PTSS/PTSD (Post-traumatic Stress Symptoms / Post-traumatic Stress Disorder) trajectories, as well as their prevalence and predictors, which may give a better understanding how and under what circumstances PTSS/PTSD presentations may develop [42]. For this purpose, researchers reviewed a large corpus of records after querying several databases. During the review, the records were labeled on various levels, which we list below.

Inclusion Criteria:

a : Is the study a longitudinal/prospective study with at least three time point assessments?

b : Does the study assess PTSD symptoms as a continuous variable? [Followed by a list of eligible scales]

c : Does this study mention that individuals are exposed to traumatic events?

d : Did the study conduct a PTSD trajectory analysis? [Followed by a list of eligible methods]

A study s can be included in the review when all criteria are satisfied (so, $\forall s \in \mathcal{D}^{+}, a(s) \wedge b(s) \wedge c(s) \wedge d(s)$).

Figure 2: An excerpt of the screening protocol that was used within this case study.

Title. Some documents can be excluded by considering the title only. For example, animal studies are never eligible, and the fact that a study is an animal study can become clear from reading the title. We only study the records that have not been excluded by title screening.

Criterion. The eligibility of a study for inclusion depends on four inclusion criteria (see Figure 2). For each criterion $i \in \{a, b, c, d\}$, a label $\mathcal{Y}_{\text{crit}_i} = \{+, ?, \neg\}$, corresponding to *True*, *Unknown*, *False* can be given. In Figure 3, some statistics per criterion are displayed.

Title Abstract. Using the logical formula in Figure 2, an inclusion judgment can be made for each criterion, so this level can be derived from the criterion level without additional human effort. This will result in a label in the space $\mathcal{Y}_{\text{ternary}} = \{+, ?, \neg\}$. Because an instance can have an *Unknown* label for one or more criteria, the final eligibility of such a study must be determined by reading the entire paper without exclusionary evidence in the record.

Full-text level: Final eligibility depends on reading the full-text of the study. This level is not considered in this work because this label needs more information than is available in this dataset (i.e., the full-text of every record).

This dataset is unique compared to other frequently used datasets used for benchmarking TAR systems (e.g., [33, 34, 35]) have only binary inclusion information, sometimes only on the full-text level. Moreover, while these datasets are based on real-world search tasks, there is little to no information about the inclusion/exclusion criteria available. The SYNERGY [43] corpus consists of several systematic reviews (including an earlier version of the PTSS dataset [44]) with links to the publications from which the screening protocols can be obtained. Unfortunately, only inclusion labels on the full-text level are included, so we cannot study retrieval efficacy fairly (we can only consider recall of the set of papers that are included based on the full-text, which is a subset of the Title-Abstract included papers; therefore, we cannot distinguish title-abstract inclusions from the false positives). To our knowledge, the dataset used in this case study (for the systematic review in [42]) is the only systematic review with labels on the criterion level.

We will consider only the records of one reviewer after title screening here, which results in a set of 4836 records after some data cleaning. Our dataset then contains $|\mathcal{D}^{\{+,?\}}| = 183$ records that are included on the title-abstract level, resulting in a prevalence of 3.78 %. One observation that can be drawn from Figure 3 is that criterion d determines the title-abstract inclusion label (displayed as *judgment*) the most.

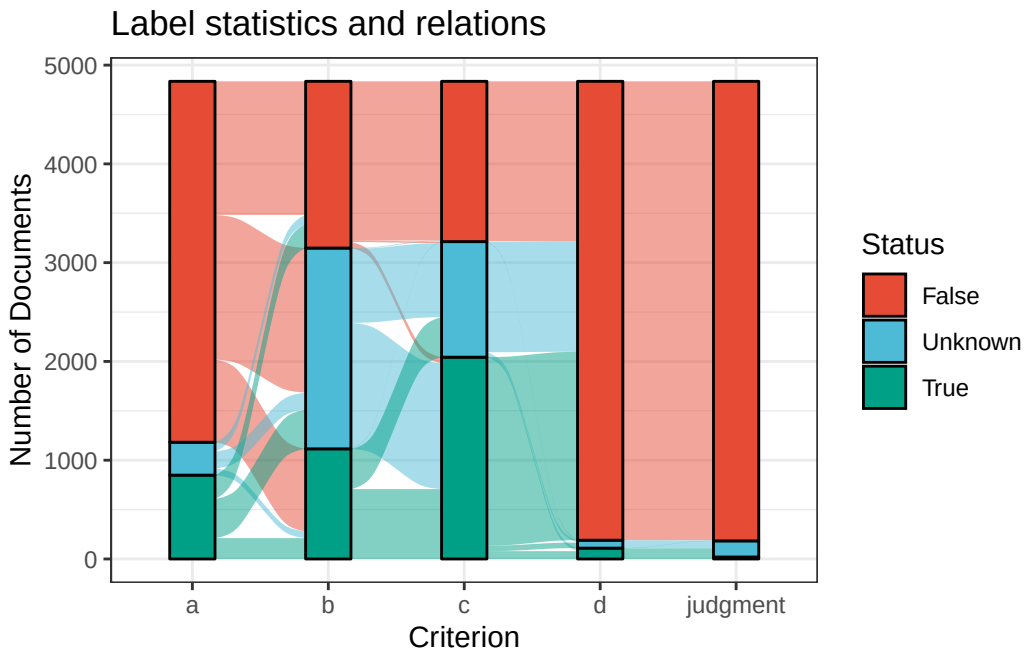


Figure 3: Label statistics displayed in an alluvial diagram, which shows some of the relations between the labels, for example, that only a tiny subset of the documents for which b is *False*, criterion c is *True*. Also, it becomes clear that criterion d excludes the most documents of all the criteria.

5. Experimental evaluation

We compare several methods in a small simulation study on the dataset described in the previous section.

- AutoTAR, a state-of-the-art TAR method,
- The LLM Classifier, as described in Section 3.1,
- LLM+CAL, our AL method that integrates the predictions of the LLM Classifier, as described in Section 3.2).

In this study, we only compare retrieval efficacy as we leave the question of a good stopping criterion open. Therefore, we constrain the run to the number of documents that are predicted by the LLM to be still eligible for inclusion (i.e., the number of documents with the label for which the inclusion judgment prediction is *True* or *Unknown*, $|\mathcal{L}_{\text{LLM}}^{\{+,?\}}|$). We let each algorithm run until this number is reached. Then, we can compare the performance of the LLM classifier and the AL-based methods with the same review effort. During the experiment, we will record when various recall levels are triggered. We will record the following metrics (calculated in the space $\mathcal{Y}_{\text{binary}}$).

Recall. The percentage of relevant documents found based on the *a priori* knowledge from the ground truth dataset.

$$R = \frac{|\mathcal{L}^+|}{|\mathcal{D}^+|} \tag{2}$$

Work Saved over Sampling. This metric expresses the work reduction over random sampling [5]. We calculate this as follows. We will record this value for several recall targets:

$$WSS = \frac{|\mathcal{U}|}{|\mathcal{D}|} - \left(1 - \frac{|\mathcal{L}^+|}{|\mathcal{D}^+|}\right) \tag{3}$$

Equation 3 is used in the AL setting. In the context of a classifier, we equate $\mathcal{U}$ to the set of documents predicted to be irrelevant (the reviewers do not read those documents). For the LLM

Classifier, we can adapt the equation as follows.

$$WSS = \frac{|\mathcal{L}^{-}_{\text{LLM}}|}{|\mathcal{D}|} - \left(1 - \frac{|\mathcal{L}^{+}_{\text{LLM}}|}{|\mathcal{D}^{+}|}\right) \tag{4}$$

The rest of the section is structured as follows: first, we describe the results of the LLM classification, followed by the results of a simulation study in which we compare the aforementioned AL-based TAR methods.

5.1. LLM Classification results

In Figure 4, we display an example of an annotated record. After parsing the response, we can highlight the fragments the LLM used in its decision-making. This overview is available for every instance in the dataset. When used in an annotation interface, the LLM explanations might aid users in their decision-making process, possibly reducing the screening time per document.

In Table 2, confusion matrices per criterion are displayed. A clear observation from Table 2 is that the LLM is more cautious in excluding papers than the human reviewer: the confusion matrices show high numbers of studies with ground truth *False* and predictions *Unknown* for all criteria. One of the causes is that when there is no written evidence to make a decision about a criterion, for example, whether or not a *PTSD trajectory analysis* (criterion d) was performed, the LLM would predict *Unknown*. This might seem like the correct decision in this situation. However, experienced human reviewers might exclude a paper based on their knowledge of the field by inferring that from other characteristics (for example, when the abstract describes a methodology that makes it impossible to use one of the eligible methods).

The LLM's definition of specific terms or the meaning of concepts might diverge from the reviewers'. For example, for criterion c, in some cases, the LLM eagerly infers from the descriptions of the studied populations that these might be exposed to trauma, which might not explicitly be mentioned in the record. Fortunately, the number of falsely excluded documents per criterion is low.

When combining the LLMs prediction, we can infer the title-abstract level predictions using the logical formula specified in Figure 2. In Table 2, the confusion matrix for this level is displayed, both on the ternary and binary levels. On this level, we obtain an accuracy of 78.52 % (ternary level), with a recall of 91.26 % on the binary level. In absolute numbers, this results in the fact that only 16 studies were missed out of the 183. The precision on the binary level is 12.9 %, resulting in a Work Saved over Sampling of 64.48 % (with Equation 4).

Document M8746
Result: $\neg a, ?b, c, \neg d$ *vs.* **Ground Truth** $\neg a, ?b, c, \neg d$
Title: Gender-based violence and its association with mental health among Somali women in a Kenyan refugee camp: a latent class analysis

Abstract: BACKGROUND: In conflict-affected settings[c], women and girls are vulnerable to gender-based violence[c] (GBV). GBV is associated with poor long-term[c] mental health such as anxiety, depression and post-traumatic[c] stress disorder (PTSD). Understanding the interaction between current violence and past conflict-related violence with ongoing mental health is essential for improving mental health service provision in refugee camps. METHODS: Using data collected from[d] 209 women attending GBV case management centres[d] in the Dadaab refugee camps, Kenya[d], we grouped women by recent experience of GBV using latent class analysis[d] and modelled the relationship between the groups and symptomatic scores[d] for anxiety, depression and PTSD[d] using linear regression. RESULTS: Women with past-year experience of intimate partner violence alone may have a higher risk of depression than women with past-year experience of non-partner violence alone (Coef. 1.68, 95% CI 0.25 to 3.11). Conflict-related violence[c,d] was an important risk factor for poor mental health among[c,d] women who accessed GBV services[c,d], despite time since occurrence (average time in camp was 11.5 years) and even[c,d] for those with a past-year experience of GBV[c,d] (Anxiety: 3.48, 1.85-5.10; Depression: 2.26, 0.51-4.02; PTSD: 6.83, 4.21-9.44)[c,d]. CONCLUSION: Refugee women who experienced past-year intimate partner violence or conflict-related violence may be at increased risk of depression, anxiety or PTSD. Service providers should be aware that compared to the general refugee population, women who have experienced violence may require additional psychological support and recognise the enduring impact of violence that occurred before, during and after periods of conflict and tailor outreach and treatment services accordingly.
Reasoning:

- $\neg a$ The title and abstract do not mention anything about a longitudinal/prospective study with multiple time-point assessments. As there is no evidence to suggest that the study is longitudinal/prospective with at least three-time point assessments, the answer is NO.
- $?b$ The question asks if the study assesses PTSD symptoms as a continuous variable and provides a list of eligible and ineligible PTSD scales. The abstract mentions that the study modelled the relationship between groups of women and symptomatic scores for anxiety, depression, and PTSD using linear regression. However, the abstract does not mention any specific PTSD scales used in the analysis. Therefore, it is not clear whether the study used any of the eligible or ineligible PTSD scales provided in the question, and the answer should be UNKNOWN.
- c The study examines the association between gender-based violence (GBV) and mental health, including anxiety, depression, and post-traumatic stress disorder (PTSD), among women in a Kenyan refugee camp. The abstract mentions that conflict-affected settings put women at risk for GBV, which is associated with poor long-term mental health such as anxiety, depression, and PTSD. The study also found that conflict-related violence was an important risk factor for poor mental health among women who accessed GBV services, despite time since occurrence. Therefore, it is likely that the individuals in the study were exposed to traumatic events.
- $\neg d$ The study aimed to investigate the relationship between GBV and mental health among Somali women in a Kenyan refugee camp. However, the methods section does not mention conducting a PTSD trajectory analysis. Therefore, the answer is NO.

Figure 4: Here, we display an annotated record (title and abstract from [45]) from the case study. This figure was generated by processing the responses generated by ChatGPT-3.5 to the prompts that were created for this record. The LLM cited the highlighted fragments as rationales for making the decision. Fragments containing evidence to include the record based on the criteria in Figure 2 are highlighted in green. Fragments highlighted in red are used to exclude a record. Yellow highlights indicate contradictory evidence, meaning the information is used as evidence for inclusion for one criterion and exclusionary evidence for another. Below the abstract, the reasoning of the LLM is listed per criterion. (Note that the breaks between highlights are automatically added to prevent overflowing lines during typesetting.)

Table 2
Confusion matrices for the LLM classifier (rows: ground truth, columns: predictions) These results were obtained by classifying each document in the dataset using ChatGPT-3.5.

Criterion a	True	Unknown	False
True	744	80	25
Unknown	116	163	54
False	254	1654	1746

Criterion b	True	Unknown	False
True	399	670	46
Unknown	124	1833	76
False	14	1532	142

Criterion c	True	Unknown	False
True	1660	376	7
Unknown	627	519	24
False	575	963	85

Criterion d	True	Unknown	False
True	103	6	1
Unknown	27	43	10
False	66	1759	2821

Table 3
Confusion matrix for inclusion status (rows: ground truth, columns: predictions). These results were obtained by classifying each document in the dataset using ChatGPT-3.5.

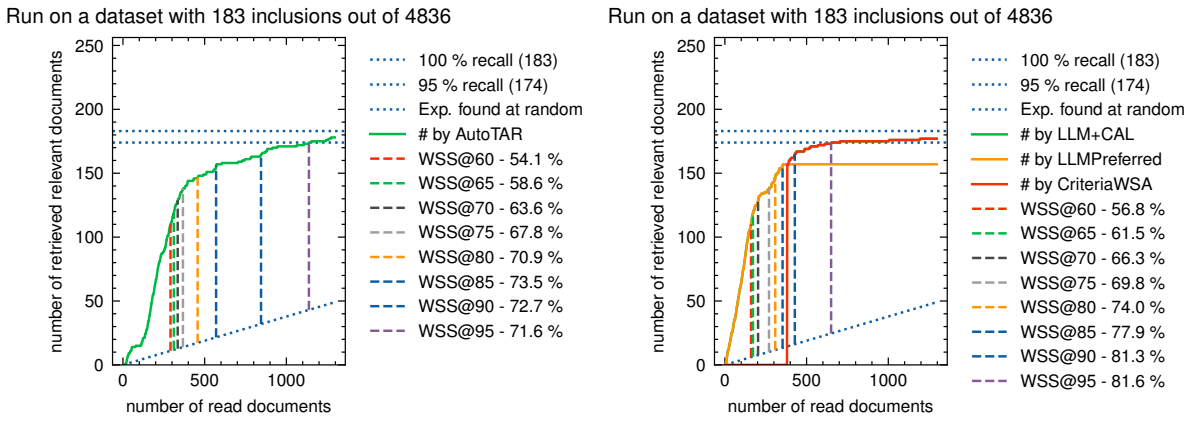
Inclusion	True	Unknown	False
True	11	8	1
Unknown	12	136	15
False	4	1124	3525

Inclusion (Binary)	True	False
True	167	16
False	1128	3525

5.2. Active Learning methods

After obtaining the LLM's results, we conducted several simulation runs of the AutoTAR baseline and our LLM+CAL method. Because both methods contain components in which random sampling takes place, we performed 30 runs per method to account for this. We stopped each simulation run after supplying the oracle 1295 papers, which is the number of documents the LLM predicted to be included ($|\mathcal{L}_{\text{LLM}}^{\{+,-\}}|$). Stopping at this moment allows a comparison of the LLM's recall to those of these methods given the same human reviewing effort. The recall curves of the methods are displayed in Figure 5. The mean recall (after stopping the simulation) of the AutoTAR method is 96.52 %, which is above the recall obtained with the LLM given the same human review effort. With the combined method, a similar recall is obtained (96.68 %), finding 177 out of 183 documents, reducing the number of missed studies from 16 to 6.

The mean recall after stopping the simulation is roughly the same for both AL methods. However, when considering other recall targets, it is evident that our combined method outperforms the baseline. For example, at 95 % recall, our method has a mean WSS@95 of 80.53 % versus 71.41 % of AutoTAR. This indicates that using the LLM predictions gives an additional advantage in retrieving relevant documents faster. In Figure 6, we give an overview of the performance for several other targets, of which all indicate that the LLM+CAL method outperforms the AutoTAR baseline.



(a) AutoTAR (baseline) (b) LLM+CAL (ours)

Figure 5: Recall curves that show retrieval statistics for both methods on the dataset of the case study. 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 several recall targets have been achieved.

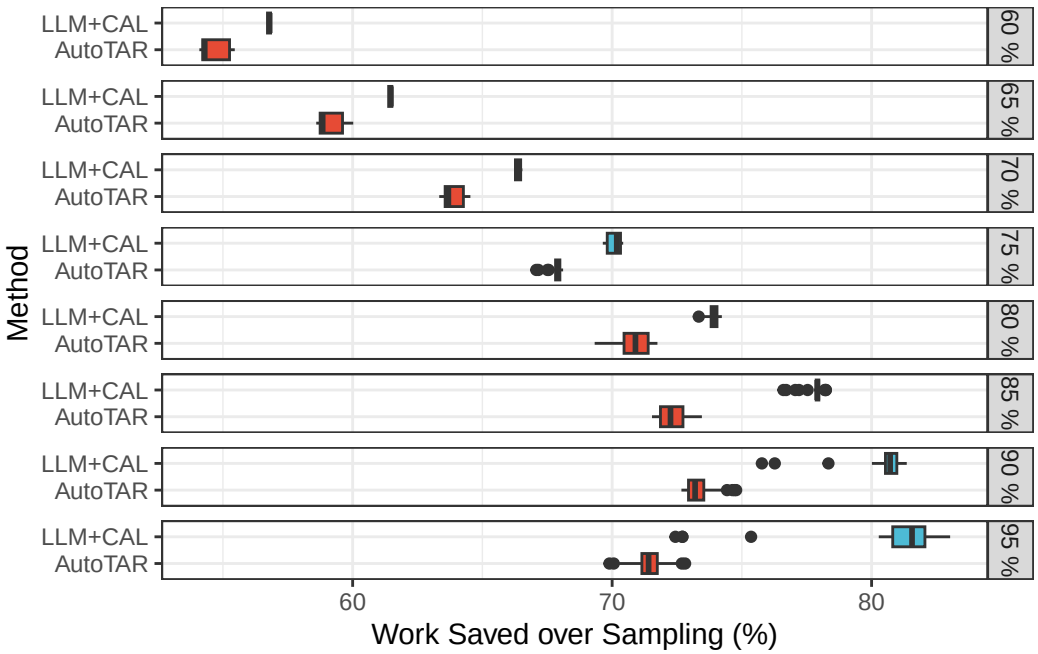


Figure 6: Here, we display, per recall target, the Work Saved over Sampling scores of the runs. We conducted multiple runs ($n = 30$) per method. It is clearly visible that our combined method (LLM+CAL) outperforms the AutoTAR baseline for every recall target.

6. Discussion

We have shown some preliminary results on our method, which indicate that adding LLM predictions is beneficial to obtaining relevant documents at a lower cost than with the state-of-the-art method AutoTAR as our LLM+CAL method yields higher work savings at several recall targets. Moreover, a reviewer could achieve a better recall and WSS than obtained using only the LLM classifier. We have presented a system that builds upon earlier LLM methods for Systematic Literature Reviews by making the predictions more fine-grained by addressing each inclusion criterion separately. Moreover, our approach aims to make the predictions more accurate and explainable by leveraging chain-of-thought reasoning and asking the LLM to cite from the title-abstract record directly. Our method takes some ideas from [21] in combining AL and the noisy labels from, in our case, an LLM annotator.

We evaluated our method on a single dataset, which may impact the generalizability of our results. Unfortunately, testing on more datasets is not feasible at the time of writing, as our method requires that the dataset has criterion-level labels. It may be interesting if we can adapt the method to work with feedback on the binary inclusion level, which might enable us to consider more datasets that do not have labels this fine-grained. Another interesting avenue is comparing the performance of our method on different LLM results than presented here. The LLM predictions may slightly differ when another model is used or when alternative formulations of inclusion criteria and general instructions are used. Further investigation is needed to determine what impact non-optimal instructions have on the LLM's accuracy and the ability of our method to correct lower-quality weak labels.

The method we presented here is still relatively simple; several extensions can be made that might further improve the efficacy. For example, incorporating Transfer Learning (as in [21]). Another area that can be explored further is the sampling strategy. Currently, our sampling strategy is based on a binary Logistic Regression classifier and TF-IDF features (as in AutoTAR). Considering other classifiers like Neural Networks and text embeddings like SentenceBERT [46] might yield additional performance gains over traditional methods.

We currently do not use the criterion-level labels during model training and subsequently rank documents in $\mathcal{U}$ with those models. Designing a good method that combines the results of the four classifiers in a ranking is not trivial. Equation 1 is a starting point (now applied to LLM only) but not optimal. Relations between criteria have also not been taken into account yet. For example, assume a scenario where, within nearly all labeled records in $\mathcal{L}$, the proposition $a \wedge b \wedge d \rightarrow c$ holds. When, for a new instance, the LLM predicts the following labels $\{a, b, \neg c, d\}$, this record may be an interesting example to review for the oracle because it is an exception to what has been seen so far.

So far, the LLM rationales have not been used to train the classifier. In [47], (human annotated) rationales were used as additional training data besides $\mathcal{X}_{\text{tiab}}$ for TAR for Systematic Literature Reviews, suggesting it might be beneficial to consider the LLM rationales during training as well.

As mentioned before, we have left the question of a stopping criterion open. One avenue could be to combine the method with an existing stopping criterion or to use the LLM predictions to determine an optimal stopping point.

During a review, regardless of whether it is performed in the traditional setting or with TAR, labeling mistakes occur due to human error [7, 26]. As in [21], our method assumes that the oracle always makes the correct decision; however, this may not always be the case. Presenting the LLM rationales and chain-of-thought fragments (like in Figure 4) may help the oracle to make better decisions and prevent some mistakes, but the extent of this has to be further investigated. Also, the Active Learning part of our method could be adapted to consider the possibility of human errors.

We believe several ideas presented here might also benefit research areas other than TAR. For example, the LLM framework presented here can be applied to text classification tasks in general. However, adapting our method to a canonical AL setting is more appropriate in this setting. The framework we presented here enables obtaining weak labels at a low cost, with little engineering effort besides writing a good labeling protocol, and chain-of-thought prompting may aid in spotting errors within them, enabling more efficient creation of text classification models.

Acknowledgements

We thank the anonymous reviewers for their insightful comments, which helped improve this article's quality. This work was sponsored by a grant from the Dutch Research Council (Domain Social Sciences and Humanities [SSH]), with file no. 406.22.GO.048. Moreover, this work was sponsored by a grant from the Human-Centered Artificial Intelligence focus area at Utrecht University.

References

[1] B. C. Wallace, T. A. Trikalinos, J. Lau, C. Brodley, C. H. Schmid, Semi-automated screening of biomedical citations for systematic reviews, BMC Bioinformatics 2010 11:1 11 (2010) 1–11. doi:10.1186/1471-2105-11-55.

[2] J. R. Baron, R. C. Losey, M. D. Berman, American Bar Association (Eds.), Perspectives on Predictive Coding: And Other Advanced Search Methods for the Legal Practitioner, American Bar Association, Chicago, Illinois, 2016.

[3] E. Yang, S. MacAvaney, D. D. Lewis, O. Frieder, Goldilocks: Just-Right Tuning of BERT for Technology-Assisted Review, in: M. Hagen, S. Verberne, C. Macdonald, C. Seifert, K. Balog, K. Nørvåg, V. Setty (Eds.), Advances in Information Retrieval, volume 13185, Springer International Publishing, Cham, 2022, pp. 502–517. doi:10.1007/978-3-030-99736-6_34.

[4] D. W. Oard, W. Webber, Information Retrieval for E-Discovery, Foundations and Trends® in Information Retrieval 7 (2013) 99–237. doi:10.1561/1500000025.

[5] A. M. Cohen, W. R. Hersh, K. Peterson, P. Y. Yen, Reducing workload in systematic review preparation using automated citation classification, Journal of the American Medical Informatics Association 13 (2006) 206–219. doi:10.1197/jamia.M1929.

[6] G. V. Cormack, M. R. Grossman, Engineering Quality and Reliability in Technology-Assisted Review, in: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '16, ACM Press, New York, New York, USA, 2016, pp. 75–84. doi:10.1145/2911451.2911510.

[7] Z. Yu, T. Menzies, FAST2: An intelligent assistant for finding relevant papers, Expert Systems with Applications 120 (2019) 57–71. doi:10.1016/j.eswa.2018.11.021.

[8] K. E. K. Chai, R. L. J. Lines, D. F. Gucciardi, L. Ng, Research Screener: A machine learning tool to semi-automate abstract screening for systematic reviews, Systematic Reviews 10 (2021) 93. doi:10.1186/s13643-021-01635-3.

[9] R. van de Schoot, J. de Bruin, R. Schram, P. Zahedi, J. de Boer, F. Weijdema, B. Kramer, M. Huijts, M. Hoogerwerf, G. Ferdinands, A. Harkema, J. Willemsen, Y. Ma, Q. Fang, S. Hindriks, L. Tummers, D. L. Oberski, An open source machine learning framework for efficient and transparent systematic reviews, Nature Machine Intelligence 3 (2021) 125–133. doi:10.1038/s42256-020-00287-7.

[10] D. Li, E. Kanoulas, When to Stop Reviewing in Technology-Assisted Reviews: Sampling from an Adaptive Distribution to Estimate Residual Relevant Documents, ACM Transactions on Information Systems 38 (2020) 1–36. doi:10.1145/3411755.

[11] F. Dennstädt, J. Zink, P. M. Putora, J. Hastings, N. Cihoric, Title and abstract screening for literature reviews using large language models: An exploratory study in the biomedical domain, Systematic Reviews 13 (2024) 158. doi:10.1186/s13643-024-02575-4.

[12] D. D. Lewis, W. A. Gale, A Sequential Algorithm for Training Text Classifiers, in: W. B. Croft, C. J. van Rijsbergen (Eds.), Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland, 3-6 July 1994 (Special Issue of the SIGIR Forum), ACM/Springer, 1994, pp. 3–12. doi:10.1007/978-1-4471-2099-5_1.

[13] P. Lombaers, J. de Bruin, R. van de Schoot, Reproducibility and Data Storage for Active Learning-Aided Systematic Reviews, Applied Sciences 14 (2024) 3842. doi:10.3390/app14093842.

[14] G. V. Cormack, M. R. Grossman, Autonomy and Reliability of Continuous Active Learning for Technology-Assisted Review, 2015. arXiv:1504.06868.

[15] G. Salton, C. Buckley, Improving retrieval performance by relevance feedback, J. Am. Soc. Inf. Sci. 41 (1990) 288–297.

[16] E. Guo, M. Gupta, J. Deng, Y.-J. Park, M. Paget, C. Naugler, Automated Paper Screening for Clinical Reviews Using Large Language Models: Data Analysis Study, Journal of Medical Internet Research 26 (2024) e48996. doi:10.2196/48996.

[17] D. Wilkins, Automated title and abstract screening for scoping reviews using the GPT-4 Large Language Model, 2023. doi:10.48550/arXiv.2311.07918. arXiv:2311.07918.

[18] S. Wang, H. Scells, S. Zhuang, M. Potthast, B. Koopman, G. Zuccon, Zero-shot Generative Large Language Models for Systematic Review Screening Automation, 2024. doi:10.48550/arXiv.2401.06320. arXiv:2401.06320.

[19] N. M. Guerreiro, D. M. Alves, J. Waldendorf, B. Haddow, A. Birch, P. Colombo, A. F. T. Martins, Hallucinations in Large Multilingual Translation Models, Transactions of the Association for Computational Linguistics 11 (2023) 1500–1517. doi:10.1162/tacl_a_00615.

[20] S. Feng, W. Shi, Y. Wang, W. Ding, V. Balachandran, Y. Tsvetkov, Don't Hallucinate, Abstain: Identifying LLM Knowledge Gaps via Multi-LLM Collaboration, 2024. doi:10.48550/arXiv.2402.00367. arXiv:2402.00367.

[21] L. Rauch, D. Huseljic, B. Sick, Enhancing Active Learning with Weak Supervision and Transfer Learning by Leveraging Information and Knowledge Sources, in: D. Kottke, G. Krempl, A. Holzinger, B. Hammer (Eds.), Proceedings of the Workshop on Interactive Adaptive Learning Co-Located with European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2022), Grenoble, France, September 23, 2022, volume 3259 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 27–42.

[22] G. V. Cormack, M. R. Grossman, Evaluation of machine-learning protocols for technology-assisted review in electronic discovery, in: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '14, Association for Computing Machinery, New York, NY, USA, 2014, pp. 153–162. doi:10.1145/2600428.2609601.

[23] M. W. Callaghan, F. Müller-Hansen, Statistical stopping criteria for automated screening in systematic reviews, Systematic Reviews 9 (2020) 1–14. doi:10.1186/s13643-020-01521-4.

[24] M. P. Bron, P. G. M. van der Heijden, A. J. Feelders, A. P. J. M. Siebes, Using Chao's Estimator as a Stopping Criterion for Technology-Assisted Review, 2024. doi:10.48550/arXiv.2404.01176. arXiv:2404.01176.

[25] M. Stevenson, R. Bin-Hezam, Stopping Methods for Technology-assisted Reviews Based on Point Processes, ACM Transactions on Information Systems 42 (2023) 73:1–73:37. doi:10.1145/3631990.

[26] W. Harmsen, J. de Groot, A. Harkema, I. van Dusseldorp, J. de Bruin, S. van den Brand, R. van de Schoot, Machine learning to optimize literature screening in medical guideline development, Systematic Reviews 13 (2024) 177. doi:10.1186/s13643-024-02590-5.

[27] J. Boetje, R. van de Schoot, The SAFE procedure: A practical stopping heuristic for active learning-based screening in systematic reviews and meta-analyses, Systematic Reviews 13 (2024) 81. doi:10.1186/s13643-024-02502-7.

[28] E. Yang, D. D. Lewis, O. Frieder, Heuristic stopping rules for technology-assisted review, in: DocEng 2021 - Proceedings of the 2021 ACM Symposium on Document Engineering, ACM, Limerick, Ireland, 2021, pp. 31:1–31:10. doi:10.1145/3469096.3469873.

[29] J. J. Teijema, L. Hofstee, M. Brouwer, J. de Bruin, G. Ferdinands, J. de Boer, P. Vizan, S. van den Brand, C. Bockting, R. van de Schoot, A. Bagheri, Active learning-based systematic reviewing using switching classification models: The case of the onset, maintenance, and relapse of depressive disorders, Frontiers in Research Metrics and Analytics 8 (2023). doi:10.3389/frma.2023.1178181.

[30] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell,

A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H. W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S. P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S. S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin, S. Jomoto, B. Jonn, H. Jun, T. Kaftan, Ł. Kaiser, A. Kamali, I. Kanitscheider, N. S. Keskar, T. Khan, L. Kilpatrick, J. W. Kim, C. Kim, Y. Kim, J. H. Kirchner, J. Kiros, M. Knight, D. Kokotajlo, Ł. Kondraciuk, A. Kondrich, A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan, T. Lee, J. Leike, J. Leung, D. Levy, C. M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanju, K. Malfacini, S. Manning, T. Markov, Y. Markovski, B. Martin, K. Mayer, A. Mayne, B. McGrew, S. M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko, P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Neelakantan, R. Ngo, H. Noh, L. Ouyang, C. O'Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pantuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F. d. A. B. Peres, M. Petrov, H. P. d. O. Pinto, Michael, Pokorny, M. Pokrass, V. H. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae, A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotsted, H. Roussez, N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens, J. Sitkin, K. Slama, I. Sohl, B. Sokolowsky, Y. Song, N. Staudacher, F. P. Such, N. Summers, I. Sutskever, J. Tang, N. Tezak, M. B. Thompson, P. Tillet, A. Tootoonchian, E. Tseng, P. Tuggle, N. Turley, J. Tworek, J. F. C. Uribe, A. Vallone, A. Vijayvergiya, C. Voss, C. Wainwright, J. J. Wang, A. Wang, B. Wang, J. Ward, J. Wei, C. J. Weinmann, A. Welihinda, P. Welinder, J. Weng, L. Weng, M. Wiethoff, D. Willner, C. Winter, S. Wolrich, H. Wong, L. Workman, S. Wu, J. Wu, M. Wu, K. Xiao, T. Xu, S. Yoo, K. Yu, Q. Yuan, W. Zaremba, R. Zellers, C. Zhang, M. Zhang, S. Zhao, T. Zheng, J. Zhuang, W. Zhuk, B. Zoph, GPT-4 Technical Report, https://arxiv.org/abs/2303.08774v6, 2023.

[31] E. Syriani, I. David, G. Kumar, Assessing the Ability of ChatGPT to Screen Articles for Systematic Reviews, CoRR abs/2307.06464 (2023). doi:10.48550/ARXIV.2307.06464. arXiv:2307.06464.

[32] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. Canton-Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, T. Scialom, Llama 2: Open Foundation and Fine-Tuned Chat Models, CoRR abs/2307.09288 (2023). doi:10.48550/ARXIV.2307.09288. arXiv:2307.09288.

[33] E. Kanoulas, D. Li, L. Azzopardi, R. Spijker, CLEF 2017 technologically assisted reviews in empirical medicine overview, CEUR Workshop Proceedings 1866 (2017) 1–29.

[34] E. Kanoulas, D. Li, L. Azzopardi, R. Spijker, 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 (2018).

[35] E. Kanoulas, D. Li, L. Azzopardi, R. Spijker, CLEF 2019 technology assisted reviews in empirical medicine overview, CEUR Workshop Proceedings 2380 (2019) 9–12.

[36] Y. Lu, B. Yao, S. Zhang, Y. Wang, P. Zhang, T. Lu, T. J.-J. Li, D. Wang, Human Still Wins over LLM: An Empirical Study of Active Learning on Domain-Specific Annotation Tasks, 2023. doi:10.48550/arXiv.2311.09825. arXiv:2311.09825.

[37] N. Kholodna, S. Julka, M. Khodadadi, M. N. Gumus, M. Granitzer, LLMs in the Loop: Leveraging

Large Language Model Annotations for Active Learning in Low-Resource Languages, 2024. doi:10.48550/arXiv.2404.02261. arXiv:2404.02261.

[38] S. S. Wagner, M. Behrendt, M. Ziegele, S. Harmeling, SQBC: Active Learning using LLM-Generated Synthetic Data for Stance Detection in Online Political Discussions, 2024. doi:10.48550/arXiv.2404.08078. arXiv:2404.08078.

[39] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. V. Le, D. Zhou, Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, Advances in Neural Information Processing Systems 35 (2022) 24824–24837.

[40] H. Chase, LangChain, 2022.

[41] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, HuggingFace's Transformers: State-of-the-art Natural Language Processing, 2020. doi:10.48550/arXiv.1910.03771. arXiv:1910.03771.

[42] R. van de Schoot, B. Coimbra, T. Evenhuis, P. Lombaers, M. van Zuiden, B. Grandfield, J. de Bruin, J. Teijema, L. de Bruin, R. Neeleman, E. Jalsovec, Trajectories of PTSD following traumatic events: A systematic and multi-database review, PROSPERO, 2023.

[43] J. de Bruin, Y. Ma, G. Ferdinands, J. Teijema, R. van de Schoot, SYNERGY - Open machine learning dataset on study selection in systematic reviews, 2023. doi:10.34894/HE6NAQ.

[44] R. van de Schoot, M. Sijbrandij, S. Depaoli, S. D. Winter, M. Olff, N. E. van Loey, Bayesian PTSD-Trajectory Analysis with Informed Priors Based on a Systematic Literature Search and Expert Elicitation, Multivariate Behavioral Research 53 (2018) 267–291. doi:10.1080/00273171.2017.1412293.

[45] M. Hossain, R. J. Pearson, A. McAlpine, L. J. Bacchus, J. Spangaro, S. Muthuri, S. Muuo, G. Franchi, T. Hess, M. Bangha, C. Izugbara, Gender-based violence and its association with mental health among Somali women in a Kenyan refugee camp: A latent class analysis, Journal of Epidemiology and Community Health 75 (2021) 327–334. doi:10.1136/jech-2020-214086.

[46] N. Reimers, I. Gurevych, Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992. doi:10.18653/v1/D19-1410.

[47] C. Shama Sastry, E. E. Milios, Active neural learners for text with dual supervision, Neural Computing and Applications 32 (2020) 13343–13362. doi:10.1007/s00521-019-04681-0.