Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

UNIVERSITÄT HAMBURG
MIN FAKULTÄT
FACHBEREIT INFORMATIK
BACHELOR'S DEGREE IN BIOINFORMATICS

BACHELOR'S THESIS

Automating PRISMA Systematic Reviews in Biomedicine applying Large Language Models

August 2024

Supervisor:
Prof. Dr. Jan Baumbach
jan.baumbach@uni-hamburg.de
Dr. Fernando M. Delgado-Chaves
fernando.miguel.delgado-chaves@uni-hamburg.de

Student:
Frederik Klein
frederik.klein@studium.uni-hamburg.de

Abstract

The exponential increase of scientific literature requires effective strategies for staying updated with expanding knowledge and identifying gaps in domain-specific areas. The traditional way to integrate these considerable quantities of knowledge is through systematic reviews, which are typically difficult since insights can spread across numerous publications and databases.

The main challenges encountered in current systematic review processes, even when using guidelines such as PRISMA, revolve around the significant human resource requirements, the use of incorrect or incomplete selection criteria, and the inherent biases introduced by manual selection processes. These difficulties severely limit the capacity to retrieve a complete and relevant article corpus that directly addresses the primary research question.

To address these problems, we provide AISAAC (Artificially Intelligent Screening Assistant for Academic Content), a Python package that uses Large Language Models (LLMs) to automate the screening steps of the systematic review process. AISAAC completes both title and abstract screening (Screening 1) and full-text review (Screening 2) in a matter of hours rather than months, allowing for the retrieval of a prefiltered group of articles. Furthermore, AISAAC helps to refine selection criteria either automatically or with an expert-centered approach, which improves decision-making processes.

Our findings show, that AISAAC still falls short in accurately conducting full-text screening compared to a human evaluation. The concordance between AISAAC's automated screening results and those obtained through conventional human-performed systematic reviews showed a maximum agreement percentage of 0.20% using a prevalence-adjusted and bias-adjusted kappa (PABAK) as metric. Further refining the selection criteria with AISAAC had notable impacts on the performance of some tested LLMs. Benchmarking across three different LLMs with three sets of selection criteria, two of which refined by AISAAC, on a systematic review conducted by the University of Maastricht revealed Mistral's Mixtral 8x7b model as the most effective.

If the current limitations of AISAAC are overcome, the anticipated impacts of a future iteration of AISAAC on the systematic review process include a significant reduction in time and human resource requirements, the provision of insights to improve selection criteria, and the establishment of a reproducible framework compatible with both proprietary and open-source LLMs. Consequently, AISAAC would present a novel strategy for improving the efficiency and accuracy of systematic literature reviews, thereby accelerating the identification of knowledge gaps and advancing research endeavors.

Keywords: PRISMA Systematic Reviews Automation; Large Language Models; Literature Retrieval Augmented Generation; Automatic Literature Screening;

Contents

1 Introduction

Researchers are currently facing an overwhelming amount of new knowledge, with over 1.8 million new articles published across 28100 journals each year [1]. The doubling rate of scientific output is projected at 9 years, leading to an explosion of knowledge in all scientific fields. In the medical field alone, the doubling time of knowledge decreased from 50 years in 1950 to 3.5 years in 2010 to just under 2 years in 2018 [2, 3]. This rapid expansion in knowledge hinders scientists to assimilate and apply it effectively, not only in research, but also concerning education and patient care [2].

In evidence-based medicine, systematic literature reviews (SRs) are typically conducted to clarify the existing state of research for a given topic. These SRs follow a strict methodology to identify, evaluate, and integrate relevant studies, enabling researchers to uncover research gaps and therefore areas that require further investigation [4]. However, executing SRs poses significant challenges, including the extensive time required, the labor intensity, the need for substantial human resources, and the potential for bias in manual selection [1, 4, 5]. Current solutions to address these issues, streamline the process of SRs, and ultimately standardize and enhance the quality of the review include standardized reporting formats, peer reviews, and bias assessment methods.

One such reporting format is the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA), which also includes a procedure to select criteria. It provides a standardized format for reporting crucial features such as literature search, selection of studies, data extraction, risk of bias assessment, and the synthesis of findings. Following PRISMA principles improves the comprehensiveness and transparency of systematic review reports [6].

PROSPERO (International Prospective Register of Systematic Reviews) is an online database for scientists, for registering SR protocols before starting the review, including information on the study question, search technique, and selection criteria. This facilitates peer review of the protocol, helps prevent unintentional repetition, and offers transparency into the intended procedures [7].

ROBINS-I (Risk Of Bias In Non-randomized Studies — of Interventions) is a method developed by the Cochrane Collaboration to assess the risk of bias in non-randomized intervention studies, such as cohort and case-control studies. It provides an approach for assessing potential biases in a variety of domains, including confounding, participant selection, intervention measurement, and missing data [8].

Conducting a high-quality SR requires screening thousands of records for inclusion or exclusion, initially based on titles and abstracts, and eventually on full texts, by two researchers, if not more. A SR therefore often takes more than 15 months to complete and publish [9].

Artificial intelligence (AI), particularly large language models (LLMs), presents a promising technology to create an array of new opportunities in medical research and treatment [10–15]. One possibility is to expedite and streamline the SR process while reducing the need for human labor. LLMs have shown the ability to understand and analyze large amounts of text within a short time, hence their helpfulness for screening titles, abstracts, and even full texts against predefined screening criteria [16, 17]. Recent studies have demonstrated impressive reasoning capabilities

for LLMs both open-source and proprietary, including the ability to perform complex cognitive tasks such as analogical reasoning, problem-solving, and inference generation [18–20]. These capabilities enable the use of RAG, a technique that combines information retrieval with LLMs, basing the model on outside knowledge sources therefore enriching the model's internal data representation [21, 22].

1.1 Related Works

Several recent studies have explored the use of AI, LLMs and machine learning (ML) to automate various steps in the systematic review process:

[23, 24] have explored the use of ML techniques to automate initial screening and citation screening. The findings suggest that ML can reduce the workload for human researchers by up to 40% in specific tasks and adequate circumstances with near-perfect recall.

[25] discuss the emerging applications of LLMs in the context of clinical trials. The findings — though they are merely indications for the potential of LLMs — are highly applicable to systematic reviews, as they address similar challenges related to handling extensive literature volumes, data extraction, and maintaining consistency in decision-making processes. [26] investigated the application of LLMs in the meta-analysis of diagnostic accuracy studies, validating their approach with real-world examples and indicating superiority over traditional meta-analysis techniques in specific scenarios. [27] created a framework for incorporating LLMs into systematic reviews, focusing on employing the models to assess risk of bias utilizing the ROBINS-I tool, showing promising results concerning consistency and credibility of the LLMs in a case study. [28] further highlights the potential increase in efficiency using LLMs in risk of bias evaluation.

[29, 30] investigated the efficacy of the closed-source models GPT-3.5 and GPT-4 in screening peer-reviewed and gray literature, discovering that the LLM can efficiently substitute human work in this phase if deployed carefully. Similarly, [31] presented a zero-shot generative LLM technique for automating systematic review screening and benchmarking eight different LLMs on five tests, also with encouraging results.

[32] propose a semi-automated approach for the first screening process in SRs by clustering schematically weighted contextual embeddings extracted by deep learning-based language models from document titles and abstracts. While the performed experiment lacks sufficient datasets for definitive conclusions, the results indicate the potential value of further investigation into the utilization of embedding models in the automation of SRs.

These studies demonstrate the potential for LLMs to streamline and enhance various aspects of the systematic review workflow, including article screening. Existing attempts to tackle the task of literature screening through natural language processing, LLMs, and ML often do not offer a human-in-the-loop system, using closed-source models or do not meet PRISMAs methodological standards.

This thesis therefore presents AISAAC (Artificially Intelligent Screening Assistant for Academic Content), a new tool that (1) uses the reasoning capabilities of LLMs to be monitored by humans, (2) allows the use of open-source models, (3) adheres to PRISMA standards and (4) utilizes contextual embeddings as proposed by [32]. It is specialized on full-text analysis but has the potential to be used for title and abstract analysis. AISAAC further provides functionality for evaluating

its performance based on manually classified documents and for refining the selection criteria automatically using either a human-centered or human-monitored process.

The code supporting this study is available in the AISAAC-repository, hosted on GitHub [33].

2 Methods

2.1 AISAAC Overview

AISAAC is a Python package using several LLMs to automatically perform the screening process of SR, determining whether a piece of literature is relevant for the researcher based on a set of criteria. Additionally, it has the functionality to evaluate the screening performance in comparison to a set of manually evaluated literature, referred to as the gold standard results by human researchers. AISAAC also allows changing and potentially optimizing the screening criteria based on automatic or semi-automatic approaches including LLMs, so-called human-on-the-loop and human-in-the-loop approaches.

2.1.1 Data Flow

As seen in Figure 1, the necessary inputs to perform an automatic screening as discussed in Section 2.2 are a set of documents to be evaluated and a set of criteria on which the screening will be based, referred to as screening criteria. The output is a list of binary relevancy evaluations per screening criterion and document with corresponding reasoning, as well as an overall binary evaluation per document. This document evaluation is “relevant” if and only if all the criteria for this document are assessed to be “relevant”. The researcher can therefore immediately see if and why a document is important to the literature review.

In order to evaluate the performance of a screening, as discussed in Section 2.4, AISAAC needs access to the automatic screening results and to a set of manually assessed screening results. A larger quantity of manually and automatically assessed documents leads to a more exact evaluation. The evaluation calculates several performance metrics, as discussed in Section 2.4, and the influence of each criterion on the result based on the feature importance, which is calculated with the help of a random forest classifier [34].

The necessary inputs for the criteria refinement process depend on the specific method used. The potential inputs include the optimizable criterion itself, the criterion’s feature importance, a set of contexts that it retrieved from the documents, and a set of annotations from an expert who may subsequently manually choose between different generated criteria. The possible configurations are further discussed in Section 2.3.

2.1.2 LLM Configuration

The tool design is based on the RAG architecture for LLMs using an embedding model that provides the necessary context and a large language model (LLM), that is able to understand and generate human-like text [12, 35, 36]. The LLM is used to decide on the relevancy of a document based on a retrieved context, or to generate alternative criteria. An optional third LLM specialized in re-ranking may be used to further narrow the similarity search results [37, 38]. The model configuration can be deployed locally using Ollama [39] or using cloud services. The open-source library LangChain [40] provides the capability to choose between server connection and local deployment, and to switch models before each run.

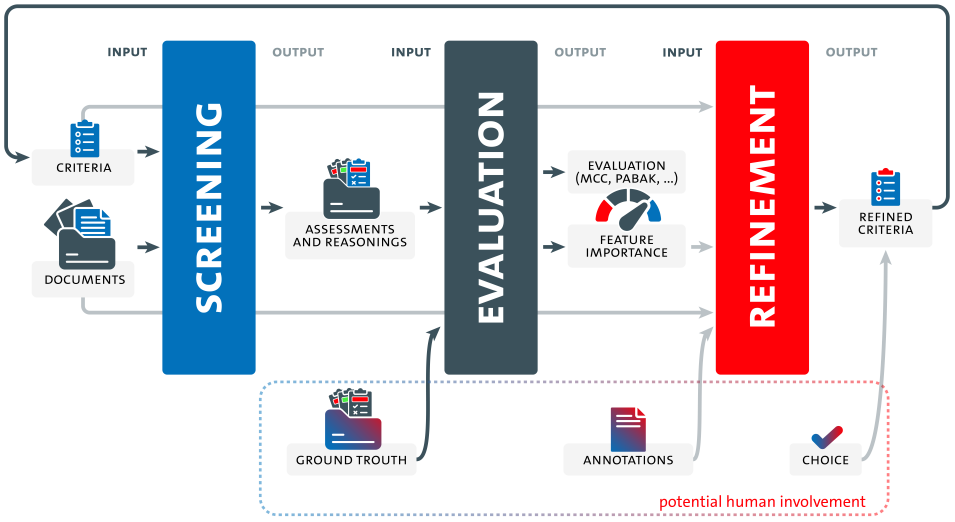

Figure 1: The three main functions of AISAAC; literature screening, result evaluation and selection criteria refinement. The screening function takes a set of criteria and a set of documents to assess the relevancy per document per criterion with a corresponding reasoning. If and only if all the criteria are assessed "relevant" for a document, the document itself gets evaluated as "relevant". The evaluation function compares AISAACs results with a manually assessed gold standard and calculates various performance metrics and the feature importance of each criterion — meaning the influence of the criterion on the performance. The criteria refinement method can take the criteria with their corresponding feature importance scores, a set of documents and human annotations to create a new set of criteria, that can be approved or narrowed down by a human.

2.2 Screening

The screening process needs a set of documents and a set of selection criteria, as discussed in Section 2.1.1. As seen in Figure 2, AISAAC imports, reads, and chunks these documents before embedding them into a vector database (vector DB) each [41]. Having one vector DB per document enables AISAAC to conduct a vector similarity search for every selection criterion for every document individually [42, 43]. The search results per criterion are then, together with the criteria themselves, passed into a single prompt. Said prompt invokes the LLM to create an output that can be parsed into an assessment of relevancy per criterion plus an overall document assessment. Should the parser fail to interpret the output, the same prompt is invoked on the LLM again for a fixed number of iterations.

2.2.1 Data Preprocessing

First, the system retrieves data from the provided documents in PDF or Markdown format using format-specific loaders provided by LangChain [40]. PDF data is additionally preprocessed to eliminate repetitive elements like headers and footers and merge all the pages into one. This helps to ensure that documents are properly converted into a consistent, noise-reduced structure that can be further processed.

In the next step, the chunking strategy divides the document's text into smaller blocks of text. Each of these blocks is called a chunk. This process reduces noise when embedding the document into a vector DB. The chunking strategy differs based on three main variables: chunk-size, chunk-overlap and chunking method. The chunk-size is the approximate amount of characters allowed per chunk. The chunk-overlap is the amount of characters that overlap between two adjacent

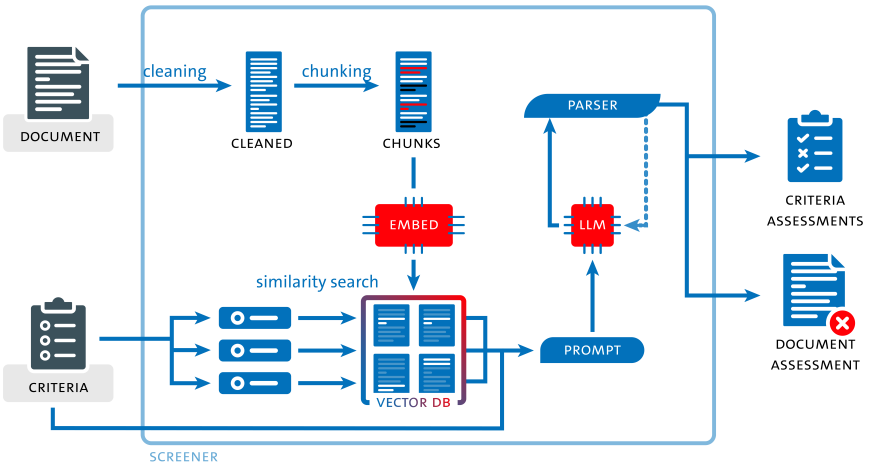

Figure 2: Overview of the processes within the screening function.

chunks. Allowing chunk-overlap helps with keeping semantic context. Lastly, the chunking method is the algorithm that determines, how exactly the text gets divided. The most naive approach is to split the text after an exact amount of characters. More nuanced approaches, like NLKT chunking [44], which is used in AISAAC, consider the structure of text and varies the chunk-size accordingly. These variables are adjustable in order to play to the strengths of the respective LLMs. The default values used for testing can be seen in Table 5 with the test results in Section 3.

The aforementioned embedding model transforms each document's text chunks into a database of vector representations with the support of Chroma DB [45], an open-source vector store solution. This creates a searchable vector database which allows retrieving relevant contextual chunks based on vector similarity rather than just keyword matching [41, 46]. The process results in one vector DB per document, stored locally on the machine. Having one vector database per document is an advantage over having a single database for all documents, since each document can be searched independently without the need of further meta-data.

2.2.2 Retrieval and Generation

To retrieve relevant context that can be passed into the LLM, a similarity search is conducted against the embedded corpus for each criterion, returning a set of significant chunks with corresponding similarity scores between 0 and 1 per criterion [42]. To ensure high relevance, chunks with scores below an adjustable threshold are removed. All default values can be seen in Table 5. If the optional re-ranking model is activated, the similarity search results get further filtered and ordered within this model, which potentially increases the relevancy of the retrieved chunks [38]. The retrieved chunks and the criteria themselves are then automatically transformed into a prompt based on a template and inputted to the LLM. All templates used throughout AISAAC can be seen in Table 6.

Based on the prompt, the RAG model generates an assessment per criterion that determines whether it applies on the retrieved contexts and therefore on the document. The model offers reasoning per criterion in human-like text to support the assessments, enabling human evaluators to follow the decision process and therefore improving its transparency and interpretability. Based

on the respective assessments, the document gets categorized as relevant if all the criteria are met.

The format of the assessment might vary due to the diverse outputs of LLMs [47]. A parser therefore tries to structure the model's output to correspond with a predefined parsing schema including the title, relevancy, and reasoning. This is important for having usable outputs. In case the parser fails at this task, the system re-queries the LLM automatically until the output can be correctly parsed. This requerying approach reduces the possibility of crashes or unprocessable outputs, while retaining the screening process's integrity through the predefined parsing schema. However, this strategy potentially requires additional computational resources, especially when using models that are unreliable in returning results in the desired format, since re-issuing queries to the LLM requires the same time and computational power as the first query again. Therefore, the re-querying is interrupted after a predefined number of failed parsing attempts, per default 5 as seen in Table 5.

2.3 Criteria Refinement

Screening criteria typically evaluate various aspects of research studies. Ambiguous, incomplete or imprecise criteria might lead to significantly different results between SRs deducted on the same corpus of literature, manual or automatic. The success of an SR is therefore directly dependent on the quality of the screening criteria. AISAAC provides a set of methods that are intended to help researches formulate or refine their criteria.

Using a similar approach for the refinement of criteria as for the screening process is effortless: The RAG architecture allows AISAAC to retrieve relevant context within a corpus of text, and the LLM is capable of generating and understanding human-like text which is important for generating criteria that are understandable to humans.

As seen in Figure 3, the whole process of criteria refinement consists of 5 stages: The inputs that form the prompt for the LLM, the generation of one or more criteria by the LLM, the choice of one of the criteria, the approval of that criterion, and finally the acceptance of that criterion. The different potential inputs allow the LLM to generate its criteria considering existing criteria, a human's annotations regarding the SR or a document, and the context of manually or automatically evaluated literature as a reference depending on what is available. None of these inputs have to be provided, but the richer the input, the more advanced the prompt for the LLM. Said prompt can therefore consist of an existing criterion, a set of contexts — which is generated the same way it is in the screening process, see Section 2.2 — and any written annotations by a human to help the LLM generate new criteria. This prompt is then passed to the LLM, which generates a set of potentially improved criteria based on this prompt. Either a human expert or the LLM can then select their preferred criterion from this set of options, or the LLM can generate a final new one based on the set of potential criteria. In the end, an expert is able to approve or disapprove a single criterion or potentially change the annotations and invoke a regeneration of criteria with the same method.

Combining these possibilities, AISAAC offers five different prebuilt criteria refinement methods that allow the researchers to choose their preferred method:

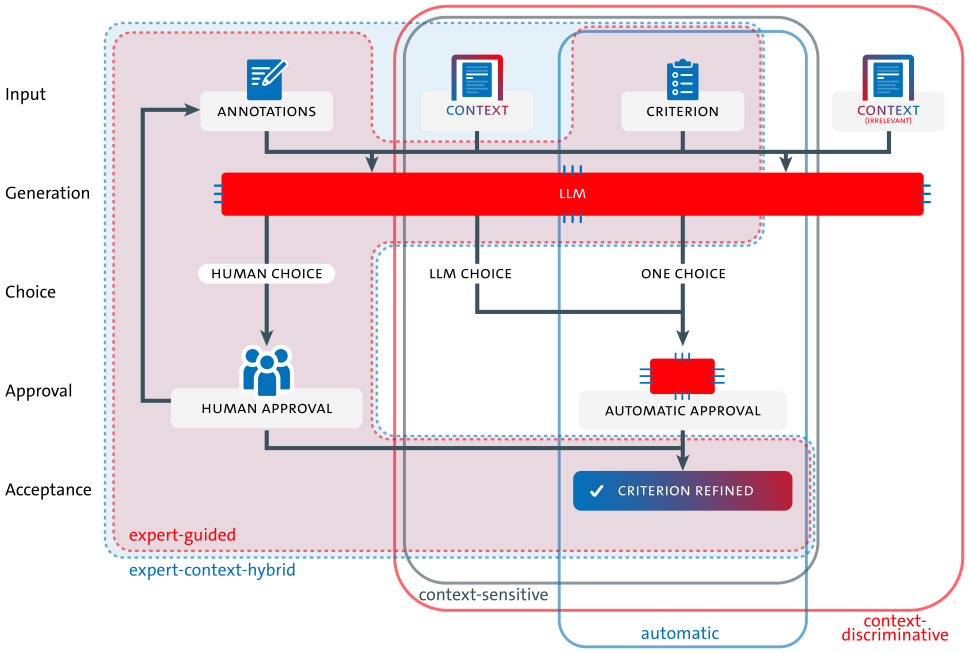

Figure 3: Overview of different criteria refinement methods. The input, choice, and approval process differs based on the used method. The automatic approach (blue line) takes the criterion and just generates one new possibility, that gets chosen, approved and accepted automatically. The context-sensitive approach (gray line) also takes into consideration relevant context and generates multiple possible criterion candidates. An LLM chooses one of these possibilities, which gets automatically approved and accepted. The context-discriminative (red line) differs from the context-sensitive approach by also considering irrelevant contexts. The expert-guided approach (red dotted line) does not consider any context, but allows for further annotations and lets a human choose and approve a generated criterion. If the human does not approve the criterion, it gets regenerated with the possibility to provide other annotations. The expert-context-hybrid approach (blue dotted line) uses context for the generation and is else identical to the expert-guided approach.

Expert-on-the-loop approaches These approaches don't require explicit intervention from a human but can be monitored to prevent mistakes.

Automatic Refinement: The LLM generates a single criterion without necessary additional input or human validation.

Context-Sensitive Refinement: Uses context from a predetermined number of randomly selected documents to generate a new criterion per document, which is then synthesized to a single criterion without further human validation.

Context-Discriminative Refinement: Leverages variance between a predetermined number of randomly selected relevant and irrelevant documents to create more precise criteria.

Expert-in-the-loop approaches These approaches require human interaction as the center of the decision process.

Expert-Guided Refinement: Allows expert input and selection from multiple LLM-generated options, with the option to regenerate infinitely with further input.

Hybrid Refinement: Combines the context-sensitive approach with expert guidance for extended criteria refinement

These different refinement methods allow the researcher to choose a situation adapted approach. The choice will mainly depend on the researcher's resources and expertise to formulate criteria, manually conduct parts of the SR and to monitor the LLMs. A potential decision tree can be seen

in Figure 4. Default values for the variables used in the refinement process are listed in Table 5.

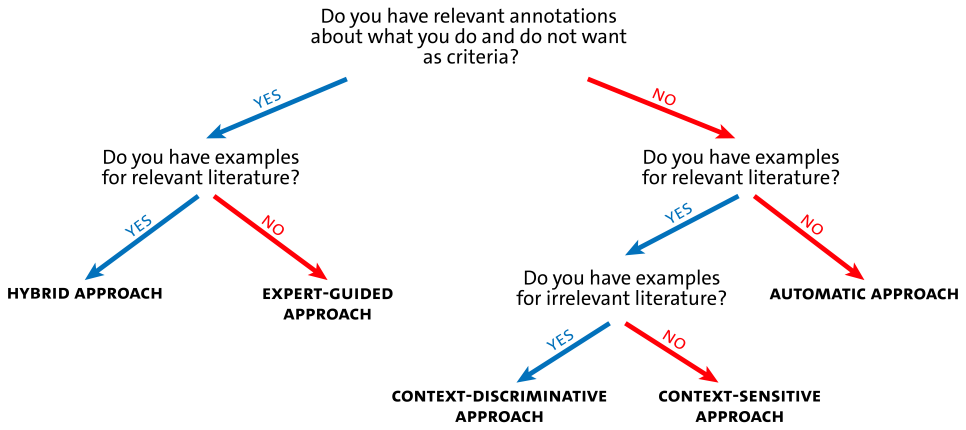

Figure 4: Potential decision tree for the choice of refinement method. Some relevant questions for the researchers are: Do you have relevant annotations about what you do and do not want as criteria that should be passed to the LLM? Do you have examples for relevant literature? Do you have examples for irrelevant literature? Based on the answers, the researcher can select the optimal refinement method.

2.4 Evaluation

In order to have a comparable evaluation, AISAAC needs a set of reproducible and quantitative metrics of performance. The very center of evaluation involves comparing AISAAC's screening results to a manually conducted reference SR's results in order to identify true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). This comparison is necessary for calculating key statistical measures indicating the performance of the system.

Precision measures the proportion of positive identifications (classifications as "relevant") that were actually correct. *Recall* measures the portion of actual positives that were correctly identified by the model. The score ranges from 0 to 1, with a score close to 1 being desirable. The *F-score*, or *F1 score*, combines precision and recall through a harmonic mean, which highlights the balance between recall and precision. The score ranges from 0 to 1, where 1 means perfect recall and perfect precision. A graphical representation known as *confusion matrix* visualizes the TP, TN, FP, and FN. A perfect confusion matrix would have only TP and TN values (top left and bottom right). *Matthews Correlation Coefficient (MCC)* measures the agreement of the prediction represented in the confusion matrix across all four categories, considering class disparities. MCC ranges from -1 to +1, where 0 indicates a random prediction, +1 a perfect match and -1 a perfect inverse match. *Cohen's Kappa* is calculated to account for the probability that agreement between AISAAC's screening results and the reference SR results by human researchers may occur by chance. This is especially important when the class distribution is imbalanced like it is in our reference SR results by human researchers, since just calculating the MCC can be misleading. The score range is the same as for MCC. A Kappa value above 0.6 is often considered good, with values above 0.8 indicating very good agreement, as proposed by Landis and Koch in 1977 [48]. *Prevalence-Adjusted Bias-Adjusted Kappa (PABAK)* is similar to Cohen's Kappa in range and interpretation, but it adjusts for prevalence and bias in the reference SR results by human researchers or model's results and therefore overcomes the challenges that Cohen's Kappa has for imbalanced, biased or prevalenced datasets [49]. MCC, Cohen's Kappa, and PABAK give a good indication for the

agreement with the reference SR results by human researchers, although every one of them has their limitations [50].

Further scores are the *rate of completion (RoC)*, which offers insights into the underlying reliability of the model in consistently producing accurate outputs, and the *feature importance* per criterion, which is based on a random forest classifier [34]. The feature importance metric can help identify irrelevant criteria that could potentially be replaced by more relevant ones, and determine which criteria should be refined by AISAAC's criteria refinement methods, as discussed in Section 2.3.

2.5 Benchmarking on a Systematic Review

To put AISAAC through its paces, we recreated a portion of a thyroid cancer systematic review from the University of Maastricht. The benchmarking included three LLM configurations running on the original criteria, as well as two refined sets of criteria.

2.5.1 Reference Dataset

The original review included an initial screening of 5000 papers based on titles and abstracts, with 180 selected for a more extensive full-text review. AISAAC was benchmarked on the task of full-text screening, evaluating its accuracy in applying the defined criteria to the selected documents from the University of Maastricht's thyroid cancer study. Due to constraints concerning accessibility and formatting, AISAAC was finally tested on a subset of 159 documents.

The original criteria from the University of Maastricht as seen in Table 1 included the type of thyroid cancer being studied, the study population (specifically, whether human subjects were involved), the nature of the study (excluding conference abstracts, reviews, studies without results, or studies solely focused on specific pathways or genetic alterations), the report of DNA alterations in thyroid cancer, the methodologies employed (e.g., Whole exome sequencing, Next generation sequencing), and the inclusion of studies analyzing TCGA data or reporting only RNA/protein sequencing. These original criteria were used for the benchmarking of AISAAC.

Criterion	Description
Thyroid Cancer Types	If the study involves any type of thyroid cancer such as Papillary TC, Follicular TC, Medullary TC, Poorly Differentiated TC, Anaplastic TC, Hurtle cell carcinoma, or Non-invasive follicular thyroid neoplasm with papillary-like nuclear features (NIFTP), then return True. However, If the study involves non-malignant entities or conditions other than the specified types of thyroid cancer, then return False.
Study Population	If the study is conducted on human subjects, then return True. Otherwise, if the study is conducted on an organism different than human, animals or uses cell lines, return False.
Study Type	If the article is a conference abstract, review, study without results (like a protocol), or model-based study, or if it investigates specific pathways or genetic alterations only, then return False. Otherwise return True.
DNA alterations	If the study is an original paper reporting on DNA alterations in thyroid cancer, then return True. Otherwise return False.
Methodology	If the study uses Whole exome sequencing (WES), Whole genome sequencing (WGS), Next generation sequencing (NGS), Sanger sequencing, Custom panel, or Microarray analysis, then return True. Otherwise, return False.
TCGA or RNA/protein	If the study involves computational analysis of TCGA data or any previously reported studies, or if it reports RNA/protein sequencing only, then return False. Otherwise, return True.

Table 1: Criteria for recreating the full text screening from the University of Maastricht determining the relevance of thyroid cancer papers.

For the purpose of AISAAC's evaluation, the dataset's ground truth was established based on the University of Maastricht's screening excluding the ones that couldn't be imported properly,

resulting originally in 47 papers being eliminated and 112 being included. It is worth noting that the University's ground truth data did not go into detail about why each document was included or excluded, but only providing the final decision.

2.5.2 Benchmarking Framework

The tested models were deployed according to their complexity and processing demands; models with up to 7 billion parameters were executed locally on a macOS system with an M1 chipset and 16 GB of RAM. The larger model, due to its considerable computing requirements, was hosted on a server capable of handling its additional processing demands. The computing times are therefore not directly comparable and should only be considered as an impression of the time that AISAAC could save when used for screening instead of human researchers.

The benchmarking software settings remained at their default values, as described in suppl. B, with one notable exception: the importance threshold for optimizing a criterion was set to a negative value to ensure that the criteria refinement process would change all the criteria for a more notable impact. Additionally, to address instances of model non-conformance, any generation attempts that failed five consecutive times were discarded and classified as "not assessed", guaranteeing clarity and integrity in the final analysis of the benchmarks.

To minimize the bias and inaccuracy of human interaction, only human-on-the-loop approaches were tested. The tests evaluated the Automatic and Context-Discriminative methods as described in Section 2.3 with no human interventions.

2.5.3 Benchmarked Models

AISAAC's benchmarking process covered three open-source models of different parameter sizes and different model families. Larger parameter sizes generally lead to longer computation times, with the possible benefit of higher informativeness, which is beneficial for this task [51]. All models used 4-bit quantization [52]. Local model configurations as discussed in Section 2.1.2 used nomic-embed-text as embedding model while the cloud-based configurations had to use gte-large to ensure it ran correctly on the cloud [53, 54]. Re-ranking was not performed to not further influence the benchmarking.

The tested models included:

LLM Name	Model Family	Parameters	Quantization	Provider	Embedding model
Gemma 2b	Gemma	2.51B	4.0	Google	nomic-embed-text
Llama 2 7b	Llama	6.74B	4.0	Meta	nomic-embed-text
Mixtral 8x7b	Mistral	45B	4.0	Mistral AI	gte-large

Table 2: Summary of LLMs and embedding models used upon analysis.

2.5.4 Benchmarking Procedures

The benchmarking process involved conducting three full-text screenings per tested model and evaluating each one. The first screening was conducted with the standard set of criteria as seen in Table 1, the second screening with a set of criteria that was refined by the automatic criteria refinement approach and the third with a set of context-discriminatively refined criteria as discussed in Section 2.3.

2.5.5 Evaluation Metrics

The performance metrics for the evaluated LLMs were consistent with those detailed in Section 2.4. Additionally, the computation time was recorded. Although the speeds are not unambiguously comparable due to varying computational resources, they provide a general indication of each model's computation speed. Human researches need a lot of time for the screening process, typically months, as described in Section 1. That value thus provides an indication of how much human labor would have been saved.

3 Results

3.1 Gold Standard Agreement

The most important results are the accuracy of the evaluations made by AISAAC in reflecting the human evaluation. High similarity score between AISAAC and human researchers mean high agreement, and would therefore allow AISAAC to substitute for a human in the screening step of SRs. As discussed in Section 2.4, MCC, Cohen's Kappa and PABAK are the scores that indicate the agreement best.

The scores show that AISAAC is not yet able to accurately replicate the provided human gold standard. As seen in Figure 5, the performances of all models were below a considerably good score, regardless of the criteria refinement method applied. However, the tested refinement methods showed notable impacts, with some improving the model's performance by up to 1135% while others led to declines of up to 228%, depending on the model and the metric. The time it took to conduct these screenings ranged from 40 minutes to around 4 hours. All the metrics can be seen in Suppl. A, Table 2.

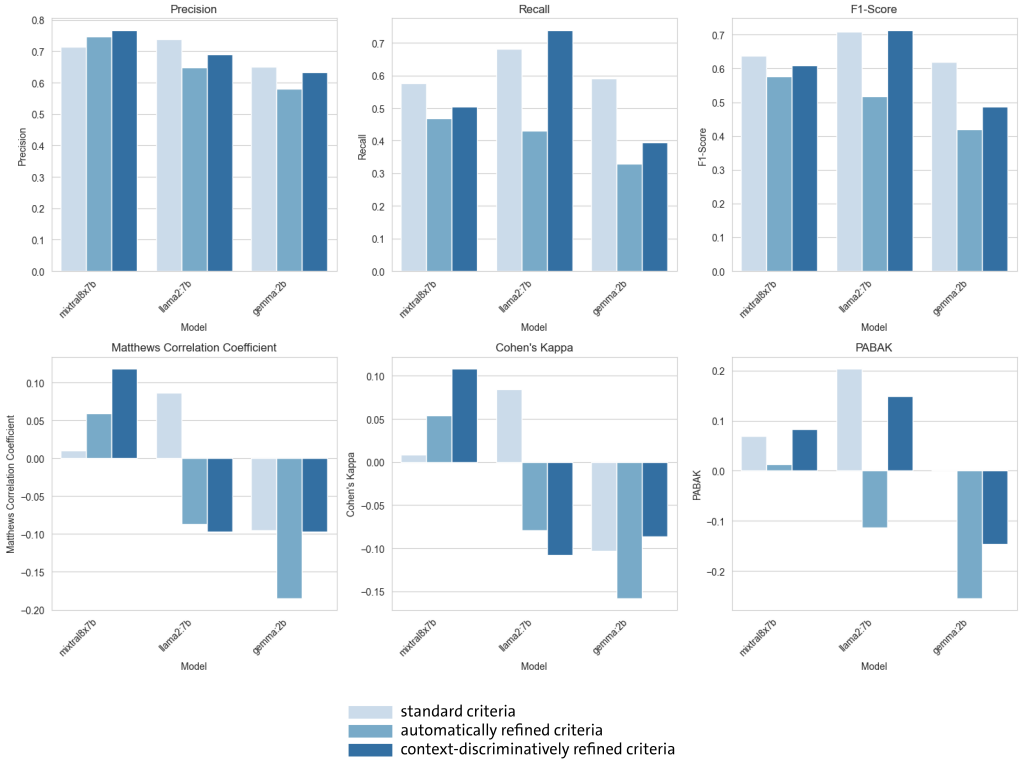

Figure 5: Performance scores of all models.

For the standard criteria, Llama 2 performs best with an MCC and Cohen's Kappa of around 0.085 and a notable PABAK score of 0.2 — which is the highest PABAK any model scored with any method. The precision scores range from 0.65 (Gemma) to 0.74 (Llama) for the standard criteria, with F1-scores from 0.62 (Gemma) to 0.71 (Llama).

The automatic refinement approach has a detrimental influence on Llama and Gemma, as only the RoC improved when compared to the standard criteria scores. All other metrics were af-

fected by declines of -27% to -200% for both Gemma and Llama 2 as seen in Table 3, resulting in negative scores, such as the PABAK of -0.25 with Gemma. The results for the automatic improvement approach with Mixtral are more divergent, with the MCC improving from 0.01 to 0.06 and Cohen's Kappa from 0.01 to 0.05, while the PABAK decreased from 0.07 to 0.01. Nonetheless, Mixtral's results were the highest across the board for the most important agreement indicators with automatically improved criteria.

The context-discriminative approach results in more polarizing findings, with a negative impact on Llama, neutral impact on Gemma and positive impact on Mixtral. The negative impact on Llama 2's MCC and Cohen's Kappa is significantly worse compared to the automatic approaches as seen in Table 3, with declines of over 210%, leading to negative scores again. However, PABAK only drops to 0.15, with minor changes in F-1 and a strong improvement in the RoC. Mixtral's scores increased significantly, with MCC and Cohen's Kappa increasing by more than tenfold to 0.12 and 0.11, respectively. PABAK improved slightly to 0.08, while F-Score and RoC were consistent. The automatic approach did not strongly affect Gemma — the scores changed by less than 25%. Cohen's Kappa improved, the F-1 score decreased.

Metric	**Mixtral 8x7b**		**Llama 2 7b**		**Gemma 2b**	
	Automatic	Context-Discriminative	Automatic	Context-Discriminative	Automatic	Context-Discriminative
MCC	467.62%	1030.48%	-200.00%	-212.46%	-93.86%	-1.68%
Cohen's Kappa	512.50%	1135.23%	-193.66%	-227.78%	-52.91%	16.44%
PABAK	-81.64%	19.65%	-155.26%	-27.38%	undefined	undefined
F1-Score	-9.61%	-4.48%	-26.99%	0.61%	-32.11%	-21.47%
RoC	-0.63%	-1.26%	33.34%	66.61%	14.58%	0.00%

Table 3: Percentual change for all models from the standard criteria to the automatically refined criteria and from the standard criteria to the context-discriminatively refined criteria.

Of the tested models, Llama 2 achieved the best agreement scores on the standard criteria with a PABAK of 0.2, but with a low RoC of 58%. The refinement methods had the best result for Mixtral with improvement rates of up to 513% and 1135% respectively, with the context-discriminative approach being the better of the two tested. The impact of the refinement methods depends highly on the model used and were partly negative, as for Llama 2.

3.2 Inter-Method Agreement

The heatmaps in Figure 6 display the agreement between the results of different models and different criteria using MCC and PABAK. The graphs reveal three key observations:
First, Mixtral 8x7b's different criteria show higher agreement between each other (both MCC and PABAK) than with the gold standard, with the best score being 0.37 (both MCC and PABAK) as seen in Tables 3 and 4, in comparison to 0.12 (MCC) and 0.08 (PABAK), as seen in Figure 5.
Second, the MCC heatmap shows that the highest agreement for any screening conducted by Mixtral 8x7b is with another screening conducted by Mixtral 8x7b on a different set of criteria (0.37). Gemma shows the same pattern (0.24).
Third, MCC and PABAK scores are mostly very close to each other. These findings are an indication that there was not a lot of bias and prevalence within the models.

The clustermap in Figure 7 shows the classification results of the tested models with the three sets of criteria. It allows checking the behaviors of the different models with different criteria and

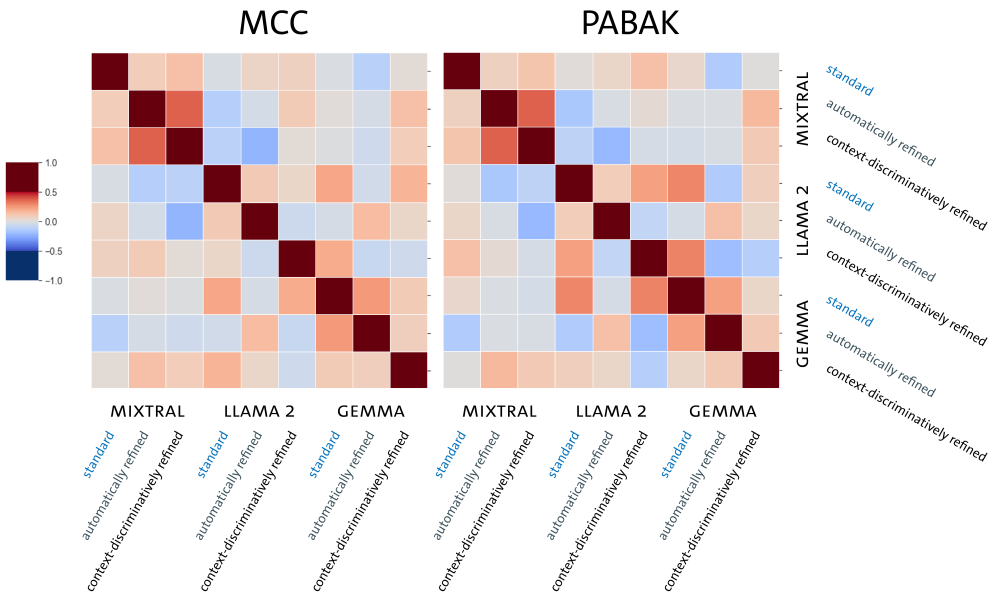

Figure 6: MCC and PABAK scores of the output from different models with different criteria.

to identify emerging patterns. The categorization predictions are binary, with red representing "True/Relevant", blue representing "False/Irrelevant", and gray representing "Null/Not assessed". The dendrogram above the heatmap indicates the hierarchical links and similarities in prediction patterns across models and criteria.

Apart from confirming the observations based on the heatmaps, there are no unexpected patterns visible in the clustermap.

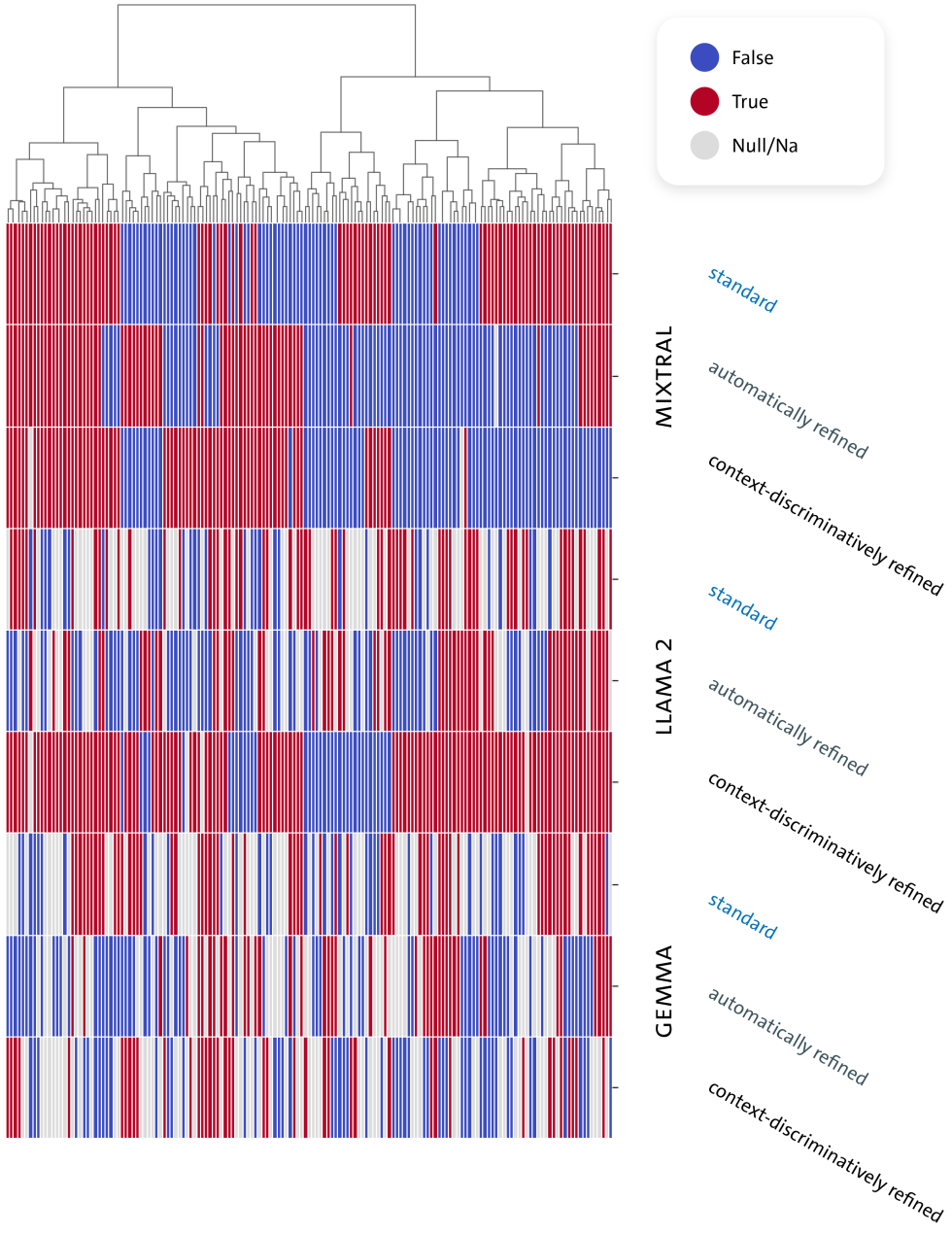

Figure 7: Clustermap of all results without the gold standard, grouped by similarity.

4 Discussion

Our findings show, that AISAAC, though the criteria refinement methods produced promising improvement rates of up to 1135%, falls short of human performance on this SR with a maximum PABAK score of 0.2. The time it took to conduct a screening went from months for human researchers to a maximum of 4 hours for AISAAC, which is a significant reduction in time.

The best tested model on the standard criteria is Llama 2 7b, according to MCC, Cohen's Kappa, and PABAK-scores of 0.09, 0.08, and 0.2 respectively. The refinement methods had substantial impacts ranging from -228% to +1135% changes on the measured performances, though the actual result depended highly on the model that was used to perform the criteria refinement. MCC and Cohen's Kappa improved on average stronger than PABAK which indicates that, even if the criteria refinement lead to better results, it also introduced hallucination or bias. Still, using refinement methods to further improve selection criteria seems to make sense with the right model and method. Mixtral 8x7b performed best for both improvement methods according to the relative and the absolute changes of MCC, Cohen's Kappa and PABAK, with the context-discriminative approach being the more performant. The best performance overall was achieved by Mixtral 8x7b with the context-discriminative approach to refine the criteria, scoring an MCC of 0.12, Cohen's Kappa of 0.11 and PABAK of 0.08.

There are a few reasons why the scores vary from model to model and from criteria to criteria. All models have originally been trained on different data, have different reasoning capabilities, and different sizes, although there are not enough data points to draw any definitive conclusions. Our findings suggest that the quality of the result does not improve with the size of the model per se, although that must be further examined. This aligns with current studies [51].

4.1 Limitations

Several limitations were observed during the process that might have had an influence on the overall performance of AISAAC.

Extracting text from PDF documents presented a challenge that inevitably impacted the quality of input data. Formatting issues and missing content were occasionally encountered, likely due to the complexity of PDF structures, variations in formatting, or limitations in text extraction tools like PyPDF. These issues compromised the input data quality, influencing the overall performance of the LLMs. A PDF conversion that removes all numbers from a document, for example, cannot lead to a good result — neither for a human nor an artificial screening researcher.

Another limitation of the deployed models is the lack of multi-modality. Graphs, images and tables in the PDF files are ignored, since the models cannot understand them. An LLM able to properly assess a pie chart has a big advantage over others that cannot.

The current chunking strategy may not be sufficient to capture the whole context of large documents. Different chunk sizes, chunk overlaps and character splitting strategies (such as naïve, recursive, tokenized, etc) were not systematically evaluated, which could have a substantial impact on the model's capacity to comprehend and appropriately identify documents.

The RAG architecture itself has its own limitations [55]. For example, the quality of the algorithm converting chunks into vector embeddings heavily influences the quality of the similarity search.

The embedding must accurately conceptualize the deeper meaning of the words and texts, which, in itself, is a hard task. This is typically done by training models on the input data, but could also be based on a knowledge graph [56]. On top of that choice, there are different kinds of embedding models, for example word-embeddings like Word2Vec and sentence-embeddings like Universal Sentence Encoder (USE) [57–62].
Next up, the similarity search algorithm itself may not be capable enough to determine the most relevant chunks per criterion. Conducting a high-dimensional nearest neighbor search like the vector similarity search is computationally expensive and therefore usually approximated, which introduces inaccuracies [63]. Apart from the trade-off between accuracy and computational power, the distance metric to identify the nearest neighbor is not trivial and can be based on different metrics, such as the Euclidian Distance or Cosine Distance [64, 65].

Combining all selection criteria and their corresponding retrieved chunks into a single LLM query, as AISAAC does, may result in computing time and resource savings compared to crafting the response in distinctive processes per criterion. When aggregating criteria, it is essential to take into account the trade-off between efficiency advantages and the possible loss of context-specific relevance or specificity, which might lead to worse results with LLMs that don't support long context windows.

Another possible limitation is the models themselves. Current LLMs still have problems with accuracy, hallucination, irrelevance, toxicity, and bias to name a few challenges [12, 66]. Strong and accurate reasoning capabilities without hallucination are essential for the LLM that is deployed in the RAG pipeline, since it has to justify the relevancy of a paper based on a few retrieved contexts. Further improvements in the field of LLMs, especially concerning reasoning capabilities, will most likely lead to better results within AISAAC.

The potential for inaccuracies within the gold standard itself cannot be overlooked. The fact that the highest MCC was scored by comparing results between two methods rather than against the gold standard raises questions regarding the reliability of human-generated classifications. However, MCC and PABAK scores stayed below a considerably good score of 0.6 across all models and methods, which suggests that inaccuracies within the gold standard are neither the primary nor the only issue. A better benchmarking framework would consider a statistically relevant amount of human researches and models to conduct the same SR and compare the average result.

4.2 Future Work

The current limitations of AISAAC and the benchmarking framework leave a lot of improvement for future iterations throughout the whole pipeline.

First and foremost, developing a more robust benchmarking framework is vital. This framework should involve a statistically relevant number of human researchers and models to ensure the reliability of human-generated classifications and improve the gold standard. Without a way of accurately and reliably measuring AISAAC's performance, it is not possible to evaluate the impact of the changes.

As for the pipeline, the preprocessing steps as discussed in Section 2.2.1 can be optimized. Testing different input formats (such as Markdown or raw text) or alternative extraction tools to PyPDF

might help address the input-related issues. Incorporating multi-modality allows the model to utilize a broader range of information and context, which would most likely improve the results. Future iterations of AISAAC should also investigate appropriate chunking strategies, including variations in chunk sizes, overlaps, and chunking method, in order to potentially increase performance.

Additionally, there is a need to enhance the RAG pipeline. Different embedding and similarity search techniques should be investigated. Exploring advanced techniques such as re-ranking, multistep reasoning, and self-consistency checks could increase relevant retrieval and accurate reasoning capabilities, and reduce hallucination, which in turn would increase overall performance [37, 67, 68]. Future work could also compare RAG with alternative architectures, like fine-tuning, for systematic review screening [69].

5 Conclusion and Outlook

In this paper, we introduce AISAAC, a Python package that aims to help researchers conduct a SR in less time and with less human expense by automating the screening process of the SR with the help of LLMs in a RAG architecture. AISAAC also helps with refining the screening criteria. We benchmarked AISAAC using three different LLM configurations and two criteria refinement methods by recreating a SR about thyroid cancer conducted by the University of Maastricht.

We successfully reduced the time it takes to conduct a screening from several months to hours at most, at the cost of performance, with the highest agreement between AISAAC and the human researchers being 0.2 using PABAK for Llama 2 7b on standard criteria, and 0.12 using MCC for Mixtral 8x7b on context-discriminatively refined criteria. The refinement methods had notable impacts on the results, both negative and positive, ranging from -228% to +1135% depending on the model and metric. Mixtral 8x7b performed best on refining the criteria, with the downside of introduced hallucination or bias.

These findings demonstrate the potential of LLMs in facilitating SRs, but underscore the complexity of the task and the need for further research and development in several areas. Future works should focus on improving input methods, chunking strategies, multi-modal article extraction, the RAG pipeline, benchmarking various models, and developing an unambiguous and definitive gold standard that can be tested against.

While AISAAC and similar tools show promise, they currently serve best as assistive technologies to augment human expertise rather than as complete replacements for human reviewers in systematic reviews. The ultimate goal remains to develop tools that can significantly reduce the time and human resources required for SRs from months to hours with no human involvement, while maintaining or improving the quality and comprehensiveness of the review process. The contributions of AISAAC, including its ability to accelerate the screening process and refine screening criteria with notable impact, represent a significant step towards achieving this goal.

References

[1] Regina Lenart-Gansiniec. The dilemmas of systematic literature review: the context of crowdsourcing in science. *International Journal of Contemporary Management*, 58(1):11–21, 2022.

[2] P. Densen. Challenges and opportunities facing medical education. *Transactions of the American Clinical and Climatological Association*, 122:48–58, 2011.

[3] Seagate Technology LLC and International Data Corporation (IDC). The digitization of the world from edge to core. `https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf`, 2022. Accessed: April 6, 2024.

[4] Christine B. Feak and John M. Swales. *Telling a Research Story: Writing a Literature Review*, volume 2 of *English in Today's Research World*. University of Michigan Press, Ann Arbor, 2009.

[5] Richard Mallett, Jessica Hagen-Zanker, Rachel Slater, and Maren Duvendack. The benefits and challenges of using systematic reviews in international development research. *Journal of development effectiveness*, 4(3):445–455, 2012.

[6] Prisma. `http://www.prisma-statement.org/?AspxAutoDetectCookieSupport=1`, 2024. Accessed: April 6, 2024.

[7] Julie H. Schiavo. Prospero: An international register of systematic review protocols. *Medical Reference Services Quarterly*, 38(2):171–180, 2019.

[8] Jonathan AC Sterne, Miguel A Hernán, Barnaby C Reeves, Jelena Savović, Nancy D Berkman, Meera Viswanathan, David Henry, Douglas G Altman, Mohammed T Ansari, Isabelle Boutron, et al. Robins-i: a tool for assessing risk of bias in non-randomised studies of interventions. *bmj*, 355, 2016.

[9] R. Borah, A.W. Brown, P.L. Capers, and K.A. Kaiser. Analysis of the time and workers needed to conduct systematic reviews of medical interventions using data from the prospero registry. *BMJ Open*, 7(2):e012545, 2017.

[10] Y. Zhang, S. Liang, Y. Feng, et al. Automation of literature screening using machine learning in medical evidence synthesis: a diagnostic test accuracy systematic review protocol. *Systematic Reviews*, 11:11, 2022.

[11] C. Hamel, M. Hersi, S. E. Kelly, et al. Guidance for using artificial intelligence for title and abstract screening while conducting knowledge syntheses. *BMC Medical Research Methodology*, 21:285, 2021.

[12] Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. Large language models: a comprehensive survey of its applications, challenges, limitations, and future prospects. *Authorea Preprints*, 2023.

[13] P. N. Ramkumar, K. N. Kunze, H. S. Haeberle, et al. Clinical and research medical applications of artificial intelligence. *Arthroscopy*, 37:1694–1697, 2021.

[14] N. Noorbakhsh-Sabet, R. Zand, Y. Zhang, et al. Artificial intelligence transforms the future of health care. *American Journal of Medicine*, 132:795–801, 2019.

[15] J. Dahmen, M. E. Kayaalp, M. Ollivier, et al. Artificial intelligence bot chatgpt in medical research: the potential game changer as a double-edged sword. *Knee Surgery, Sports Traumatology, Arthroscopy*, 31:1187–1189, 2023.

[16] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics (TACL)*, 2023.

[17] Anze Xie Ying Sheng Lianmin Zheng Joseph E. Gonzalez Ion Stoica Xuezhe Ma Dacheng Li*, Rulin Shao* and Hao Zhang. How long can open-source llms truly promise on context length?, June 2023.

[18] Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey, 2023.

[19] Siwei Yang, Bingchen Zhao, and Cihang Xie. Aqa-bench: An interactive benchmark for evaluating llms' sequential reasoning ability, 2024.

[20] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models, 2022.

[21] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

[22] IBM Research. What is retrieval-augmented generation? `https://research.ibm.com/blog/retrieval-augmented-generation-RAG`. Accessed: April 6, 2024.

[23] Tanja Bekhuis and Dina Demner-Fushman. Towards automating the initial screening phase of a systematic review. In *MEDINFO 2010*, volume 160 of *Studies in Health Technology and Informatics*, pages 146–150. IOS Press, 2010.

[24] Byron C. Wallace, Thomas A. Trikalinos, Joseph Lau, Carla Brodley, and Christopher H. Schmid. Semi-automated screening of biomedical citations for systematic reviews. *BMC Bioinformatics*, 11(1):55, 2010.

[25] J. L. Ghim and S. Ahn. Transforming clinical trials: the emerging roles of large language models. *Translational and Clinical Pharmacology*, 31(3):131, 2023.

[26] C. Frömke, M. Kirstein, and A. Zapf. A semiparametric approach for meta-analysis of diagnostic accuracy studies with multiple cut-offs. *Research Synthesis Methods*, 13(5):612–621, 2022.

[27] B. Hasan, S. Saadi, N. S. Rajjoub, M. Hegazi, M. Al-Kordi, F. Fleti, et al. Integrating large language models in systematic reviews: a framework and case study using robins-i for risk of bias assessment. *BMJ Evidence-Based Medicine*, 2024.

[28] A. J. Nashwan and J. H. Jaradat. Streamlining systematic reviews: harnessing large language models for quality assessment and risk-of-bias evaluation. *Cureus*, 15(8), 2023.

[29] Eugene Syriani, Istvan David, and Gauransh Kumar. Assessing the ability of chatgpt to screen articles for systematic reviews, 2023.

[30] Q. Khraisha, S. Put, J. Kappenberg, A. Warraitch, and K. Hadfield. Can large language models replace humans in systematic reviews? evaluating gpt-4's efficacy in screening and extracting data from peer-reviewed and grey literature in multiple languages. *Res Syn Meth*, pages 1–11, 2024.

[31] S. Wang, H. Scells, S. Zhuang, M. Potthast, B. Koopman, and G. Zuccon. Zero-shot generative large language models for systematic review screening automation. *ECIR2024. arXiv preprint arXiv:2401.06320*, 2024.

[32] R. Alchokr, M. Borkar, S. Thotadarya, G. Saake, and T. Leich. Supporting systematic literature reviews using deep-learning-based language models. In *Proceedings of the 1st International Workshop on Natural Language-based Software Engineering*, pages 67–74, 2022.

[33] Frederik Klein. Aisaac. `https://github.com/FrederikKlein/AISAAC`, 2024. Accessed: August 8, 2024.

[34] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[35] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.11401*, 2020.

[36] Ivano Lauriola, Alberto Lavelli, and Fabio Aiolli. An introduction to deep learning in natural language processing: Models, techniques, and tools. *Neurocomputing*, 470:443–456, 2022.

[37] Changhua Pei, Yi Zhang, Yongfeng Zhang, Fei Sun, Xiao Lin, Hanxiao Sun, Jian Wu, Peng Jiang, Junfeng Ge, and Wenwu Ou. Personalized re-ranking for recommendation. In *Proceedings of the Thirteenth ACM Conference on Recommender Systems*, pages 3–11, Copenhagen, Denmark, 2019. ACM, ACM.

[38] Mr.Sagar Tambe and A.N.Nawathe. Increasing the performance of text base image search engine using attribute assisted reranking model. *International Journal of Advance Research and Innovative Ideas in Education*, 3:350–352, 2017.

[39] Ollama Team. Ollama: The ultimate toolkit for multimodal machine learning. https://github.com/ollama/ollama, 2024. Accessed: June 29, 2024.

[40] LangChain AI Team. Langchain: Building applications with llms through composability. https://github.com/langchain-ai/langchain, 2024. Accessed: June 29, 2024.

[41] Toni Taipalus. Vector database management systems: Fundamental concepts, use-cases, and current challenges. *Cognitive Systems Research*, 85:101216, 2024.

[42] Dustin Lange. *Effective and Efficient Similarity Search in Databases*. PhD thesis, University of Potsdam, Potsdam, Germany, 2013. Dissertation zur Erlangung des akademischen Grades Doktor-Ingenieur (Dr.-Ing.) in der Wissenschaftsdisziplin Informationssysteme.

[43] Christian Böhm, Stefan Berchtold, and Daniel A. Keim. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Comput. Surv.*, 33(3):322–373, sep 2001.

[44] Deepa Yogish, TN Manjunath, and Ravindra S Hegadi. Review on natural language processing trends and techniques using nltk. In *Recent Trends in Image Processing and Pattern Recognition: Second International Conference, RTIP2R 2018, Solapur, India, December 21–22, 2018, Revised Selected Papers, Part III 2*, pages 589–606. Springer, 2019.

[45] Chroma Core Team. Chroma: Core embedding database. https://github.com/chroma-core/chroma, 2024. Accessed: June 29, 2024.

[46] Xingrui Xie, Han Liu, Wenzhe Hou, and Hongbin Huang. A brief survey of vector databases. *2023 9th International Conference on Big Data and Information Analytics (BigDIA)*, pages 364–371, 2023.

[47] Michael Xieyang Liu, Frederick Liu, Alexander J. Fiannaca, Terry Koo, Lucas Dixon, Michael Terry, and Carrie J. Cai. "we need structured output": Towards user-centered constraints on large language model output. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, CHI '24. ACM, May 2024.

[48] J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977. Accessed: July 23, 2024.

[49] Guanmin Chen, Peter Faris, Brenda Hemmelgarn, Robin L. Walker, and Hude Quan. Measuring agreement of administrative data with chart data using prevalence unadjusted and adjusted kappa. *BMC Medical Research Methodology*, 9(1):5, 2009.

[50] Shaun O'Leary, Marte Lund, Tore Johan Ytre-Hauge, Sigrid Reiersen Holm, Kaja Naess, Lars Nagelstad Dalland, and Steven M. McPhail. Pitfalls in the use of kappa when interpreting agreement between multiple raters in reliability studies. *Physiotherapy*, 100(1):27–35, 2014.

[51] Joy Mahapatra and Utpal Garain. Impact of model size on fine-tuned llm performance in data-to-text generation: A state-of-the-art investigation. *arXiv preprint arXiv:2407.14088*, 2024. License: CC BY 4.0.

[52] Zhuocheng Gong, Jiahao Liu, Jingang Wang, Xunliang Cai, Dongyan Zhao, and Rui Yan. What makes quantization for large language models hard? an empirical study from the lens of perturbation. *arXiv preprint arXiv:2403.06408*, 2024. License: CC BY 4.0.

[53] Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. Nomic embed: Training a reproducible long context text embedder, 2024.

[54] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning, 2023.

[55] Jerry Liu. Building production-ready rag applications. https://www.youtube.com/watch?v=TRjq7t2Ms5I, 2024. Accessed: 2024-07-23.

[56] Guoming Lu, Lizong Zhang, Minjie Jin, Pancheng Li, and Xi Huang. Entity alignment via knowledge embedding and type matching constraints for knowledge graph inference. *Journal of Ambient Intelligence and Humanized Computing*, 13:5199 – 5209, 2021.

[57] Xin Rong. word2vec parameter learning explained, 2016.

[58] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani,

and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[59] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[60] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information, 2017.

[61] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[62] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

[63] Allan Borodin, Rafail Ostrovsky, and Yuval Rabani. Lower bounds for high dimensional nearest neighbor search and related problems. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, STOC '99, page 312–321, New York, NY, USA, 1999. Association for Computing Machinery.

[64] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional space. In Jan Van den Bussche and Victor Vianu, editors, *Database Theory — ICDT 2001*, pages 420–434, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[65] Archana Singh, Avantika Yadav, and Ajay Rana. K-means with three different distance metrics. *International Journal of Computer Applications*, 67(10), 2013.

[66] Vamsi Aribandi et al. Ext5: Towards extreme multi-task scaling for transfer learning. *arXiv preprint arXiv:2205.00445*, 2022.

[67] Nisarg Patel, Mohith Kulkarni, Mihir Parmar, Aashna Budhiraja, Mutsumi Nakamura, Neeraj Varshney, and Chitta Baral. Multi-logieval: Towards evaluating multi-step logical reasoning ability of large language models, 2024.

[68] Furui Cheng, Vilém Zouhar, Simran Arora, Mrinmaya Sachan, Hendrik Strobelt, and Mennatallah El-Assady. Relic: Investigating large language model responses using self-consistency. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA, 2024. Association for Computing Machinery.

[69] Angels Balaguer, Vinamra Benara, Renato Luiz de Freitas Cunha, Roberto de M. Estevão Filho, Todd Hendry, Daniel Holstein, Jennifer Marsman, Nick Mecklenburg, Sara Malvar, Leonardo O. Nunes, Rafael Padilha, Morris Sharp, Bruno Silva, Swati Sharma, Vijay Aski, and Ranveer Chandra. Rag vs fine-tuning: Pipelines, tradeoffs, and a case study on agriculture, 2024.

A Supplementary Material: Benchmarking Results

Model	MCC	Cohen's Kappa	PABAK	F1-Score	Rate of Completion
mixtral 8x7b no improv	0.0105	0.0088	0.0692	0.6373	**1.0000**
mixtral 8x7b auto improv	0.0596	0.0539	0.0127	0.5761	0.9937
mixtral 8x7b con improv	**0.1187**	**0.1087**	0.0828	0.6087	0.9874
llama2 7b no improv	0.0866	0.0846	**0.2043**	0.7087	0.5849
llama2 7b auto improv	-0.0866	-0.0792	-0.1129	0.5175	0.7799
llama2 7b con improv	-0.0974	-0.1081	0.1484	**0.7130**	0.9748
gemma 2b no improv	-0.0954	-0.1034	0.0000	0.6190	0.6038
gemma 2b auto improv	-0.1849	-0.1581	-0.2545	0.4202	0.6918
gemma 2b con improv	-0.0970	-0.0864	-0.1458	0.4860	0.6038

Table 1: Overview over Most Relevant Performance Metrics for Various Models. The best score per category is highlighted

Model	TP	TN	FP	FN	Ratio ofCompletion	Precision	Recall	F1-score	MCC	CohenïsKappa	PABAK	SuccessfullyAnalyzed Articles	Articles with NoPredictions	computing times(in sec)
mixtral8x7b_no_improv	65	20	26	48	1.0	0.7143	0.5752	0.6373	0.0105	0.0088	0.0692	159	0	6360
mixtral8x7b_auto_improv	53	27	18	60	0.9937	0.7465	0.4690	0.5761	0.0596	0.0539	0.0127	158	1	6670
mixtral8x7b_con_improv	56	29	17	55	0.9874	0.7671	0.5045	0.6087	0.1187	0.1087	0.0828	157	2	6907
llama2:7b_no_improv	45	11	16	21	0.5849	0.7377	0.6818	0.7087	0.0866	0.0846	0.2043	93	66	13219
llama2:7b_auto_improv	37	18	20	49	0.7799	0.6491	0.4302	0.5175	-0.0866	-0.0792	-0.1129	124	35	14804
llama2:7b_con_improv	82	7	37	29	0.9748	0.6891	0.7387	0.7130	-0.0974	-0.1081	0.1484	155	4	14317
gemma:2b_no_improv	39	9	21	27	0.6038	0.6500	0.5909	0.6190	-0.0954	-0.1034	0.0000	96	63	2442
gemma:2b_auto_improv	25	16	18	51	0.6918	0.5814	0.3290	0.4202	-0.1849	-0.1581	-0.2545	110	49	2609
gemma:2b_con_improv	26	15	15	40	0.6038	0.6341	0.3939	0.4860	-0.0970	-0.0864	-0.1458	96	63	2518

Table 2: Performance Metrics for Different Models

		Mixtral 8x7b			Llama 2 7b			Gemma 2b		
		Standard	Automatic	Context-Discriminative	Standard	Automatic	Context-Discriminative	Standard	Automatic	Context-Discriminative
Mixtral 8x7b	Standard	-	0.0914	0.1330	-0.0205	0.0566	0.0752	-0.0110	-0.1216	0.0217
	Automatic	0.0914	-	0.3679	-0.1319	-0.0325	0.0966	0.0092	-0.0273	0.1369
	Context-Discriminative	0.1330	0.3679	-	-0.1150	-0.2291	0.0184	-0.0069	-0.0503	0.0916
Llama 2 7b	Standard	-0.0205	-0.1319	-0.1150	-	0.1075	0.0505	0.2147	-0.0445	0.1795
	Automatic	0.0566	-0.0325	-0.2291	0.1075	-	-0.0613	-0.0286	0.1539	0.0507
	Context-Discriminative	0.0752	0.0966	0.0184	0.0505	-0.0613	-	0.1979	-0.0747	-0.0490
Gemma 2b	Standard	-0.0110	0.0092	-0.0069	0.2147	-0.0286	0.1979	-	0.2433	0.1003
	Automatic	-0.1216	-0.0273	-0.0503	-0.0445	0.1539	-0.0747	0.2433	-	0.0807
	Context-Discriminative	0.0217	0.1369	0.0916	0.1795	0.0507	-0.0490	0.1003	0.0807	-

Table 3: MCC between different models and with standard, automatically refined and context-discriminatively refined criteria.

		Mixtral 8x7b			Llama 2 7b			Gemma 2b		
		Standard	Automatic	Context-Discriminative	Standard	Automatic	Context-Discriminative	Standard	Automatic	Context-Discriminative
Mixtral 8x7b	Standard	-	0.0759	0.1210	0.0108	0.0484	0.1355	0.0417	-0.1455	0.0000
	Automatic	0.0759	-	0.3718	-0.1613	-0.0161	0.0260	-0.0105	-0.0092	0.1579
	Context-Discriminative	0.1210	0.3718	-	-0.1087	-0.2131	-0.0260	-0.0316	-0.0185	0.1064
Llama 2 7b	Standard	0.0108	-0.1613	-0.1087	-	0.0909	0.2308	0.2941	-0.1475	0.0794
	Automatic	0.0484	-0.0161	-0.2131	0.0909	-	-0.0909	-0.0270	0.1364	0.0526
	Context-Discriminative	0.1355	0.0260	-0.0260	0.2308	-0.0909	-	0.2979	-0.1963	-0.1277
Gemma 2b	Standard	0.0417	-0.0105	-0.0316	0.2941	-0.0270	0.2979	-	0.2308	0.0508
	Automatic	-0.1455	-0.0092	-0.0185	-0.1475	0.1364	-0.1963	0.2308	-	0.1045
	Context-Discriminative	0.0000	0.1579	0.1064	0.0794	0.0526	-0.1277	0.0508	0.1045	-

Table 4: PABAK between different models and with standard, automatically refined and context-discriminatively refined criteria.

B Supplementary Material: Default Values

Parameter	Value
CHROMA_PATH	chroma
DATA_PATHS	{”Data/Excluded”, ”Data/Included”}
BIN_PATH	bin
RESULT_PATH	results
RESULT_FILE	results.csv
ORIGINAL_RESULT_FILE	original_results.csv
ORIGINAL_RESULT_PATH	gold_standard_data
MODEL_CLIENT_URL	http://localhost:11434
EMBEDDING_MODEL	nomic-embed-text:latest
RAG_MODEL	mixtral:latest
LOCAL_MODELS	True
DATA_FORMAT	*.pdf
RANDOM_SUBSET	False
SUBSET_SIZE	5
RELEVANCE_THRESHOLD_CUTOFF	0.7
APPLY_RELEVANCE_THRESHOLD	True
APPLY_RERANKING	False
SIMILARITY_SEARCH_K	4
CHUNK_SIZE	1000
CHUNK_OVERLAP	100
VERBOSE_CODE	True
LOGGING_LEVEL	DEBUG
PROGRESS_BAR	True
RESET_RESULTS	True
BASE_DIR	aisaac
PROMPT_TEMPLATE	None
QUESTION	None
CSV_HEADER	{”title”, ”converted”, ”embedded”, ”relevant”, ”checkpoints”, ”reasoning”}
FEATURE_IMPORTANCE_THRESHOLD	0.1
MAX_FEATURE_IMPROVEMENT_DOCUMENTS	20
NUMBER_EXPERT_CHOICES	3
IMPORTANCE_GREATER_THAN_THRESHOLD	True

Table 5: Default configuration of the tool.

Scenario	Prompt
Screening prompt	[INST] Answer the question based only on the following context: {context} With the following checkpoints: {checkpoints} --- Answer the question based on the above context: {question} {format_instructions} [/INST]
Generating improved checkpoints with context	[INST] Answer the question based only on the following context: {context} With the following criterion: {checkpoint} And the following expert annotations: {annotations} --- Answer the question based on the above context: What is a better criterion? {format_instructions} [/INST]
Generating improved checkpoints without context	[INST] A criterion is a string describing indicators for including or excluding a piece of data in a dataset. Here is a suboptimal criterion: {checkpoint} Please provide a better one considering the following expert annotations if there are any: {annotations} --- {format_instructions} [/INST]
Average a criterion	[INST] A criterion is a string describing indicators for including or excluding a piece of data in a dataset. Here is a list of criteria that might be a better for the problem at hand. Create the best criterion based on the following criteria: {checkpoints} {format_instructions} [/INST]
Generate a context-discriminative criterion	[INST] A criterion is a plain text string used to determine whether to include or exclude a piece of data in a dataset. Your task is to create a new criterion based on the provided sets of contexts. You are provided with examples where the current criterion has correctly evaluated (True) and incorrectly evaluated (False). Generate a refined criterion that improves the accuracy of these evaluations. Current Criterion: {checkpoint} Contexts Evaluated as True (Should be True): {positive_contexts} Contexts Evaluated as False (Should be False): {negative_contexts} Objective: Develop a more accurate criterion that enhances the differentiation between the True and False contexts. Directly provide the new criterion; do not describe how to create it or the reasoning behind it. {format_instructions} [/INST]

Table 6: Different prompt templates used throughout AISAAC.