

# AiReview: An Open Platform for Accelerating Systematic Reviews with LLMs

Xinyu Mao

The University of Queensland  
Brisbane, Australia  
xinyu.mao@uq.edu.au

Teerapong Leelanupab

The University of Queensland  
Brisbane, Australia  
t.leelanupab@uq.edu.au

Martin Potthast

University of Kassel and hessian.AI  
Kassel, Germany  
martin.potthast@uni-kassel.de

Harrisen Scells

University of Kassel and hessian.AI  
Kassel, Germany  
harry.scell@uni-kassel.de

Guido Zuccon

The University of Queensland  
Brisbane, Australia  
g.zuccon@uq.edu.au

## ABSTRACT

Systematic reviews are fundamental to evidence-based medicine. Creating one is time-consuming and labour-intensive, mainly due to the need to screen, or assess, many studies for inclusion in the review. Several tools have been developed to streamline this process, mostly relying on traditional machine learning methods. Large language models (LLMs) have shown potential in further accelerating the screening process. However, no tool currently allows end users to directly leverage LLMs for screening or facilitates systematic and transparent usage of LLM-assisted screening methods. This paper introduces (i) an extensible framework for applying LLMs to systematic review tasks, particularly title and abstract screening, and (ii) a web-based interface for LLM-assisted screening. Together, these elements form AiReview—a novel platform for LLM-assisted systematic review creation. AiReview is the first of its kind to bridge the gap between cutting-edge LLM-assisted screening methods and those that create medical systematic reviews. The tool is available at <https://aireview.ielab.io>. The source code is also open sourced at <https://github.com/ielab/ai-review>.

## KEYWORDS

Systematic Reviews, Large Language Models.

### ACM Reference Format:

Xinyu Mao, Teerapong Leelanupab, Martin Potthast, Harrisen Scells, and Guido Zuccon. 2025. AiReview: An Open Platform for Accelerating Systematic Reviews with LLMs. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

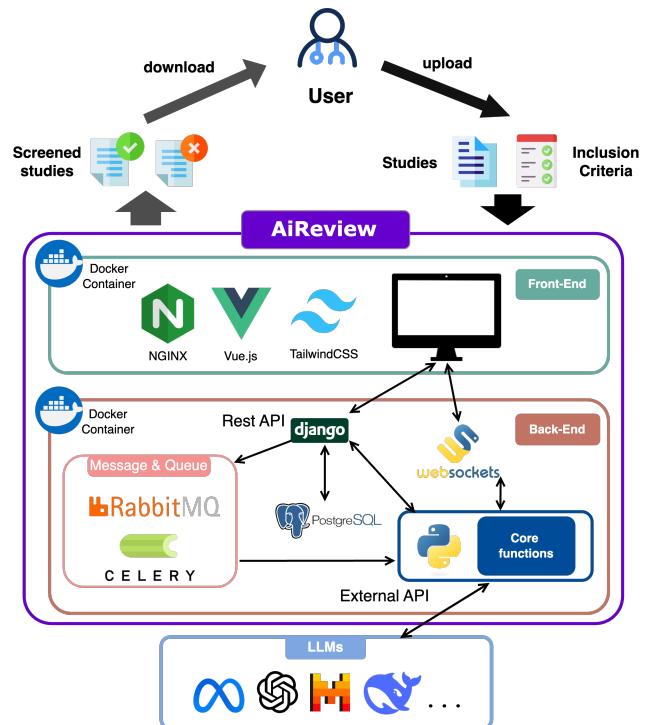
## 1 INTRODUCTION AND RELATED WORK

Systematic reviews (SRs) are comprehensive literature reviews that identify and appraise relevant studies to answer targeted research questions. The most labour-intensive part of conducting a SR is title and abstract screening, where tens of thousands of studies

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*Conference'17, July 2017, Washington, DC, USA*

© 2025 Association for Computing Machinery.  
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00  
<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

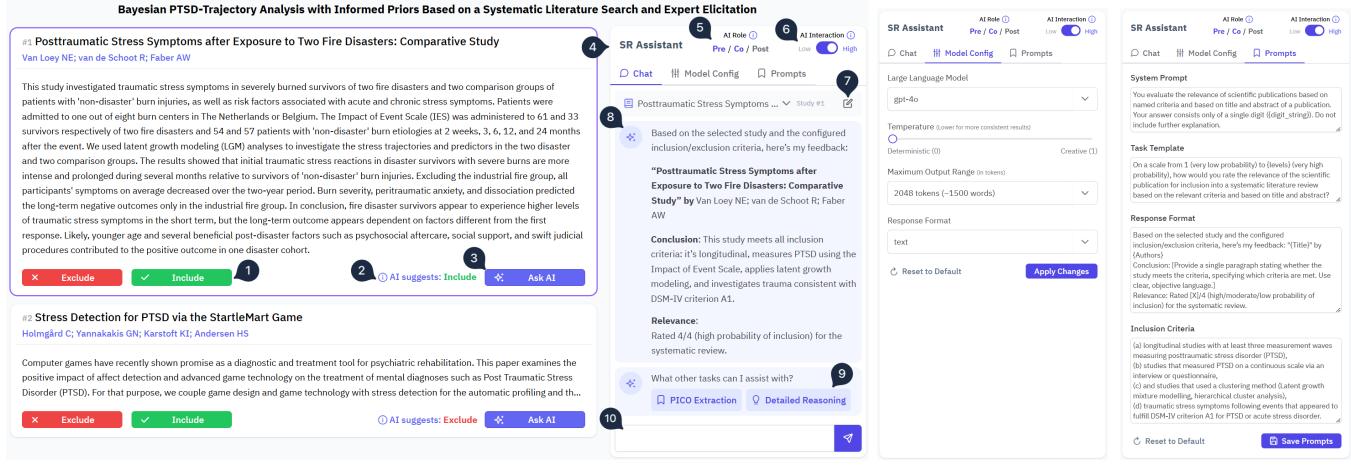


**Figure 1: Holistic architecture and workflow of AiReview.**

(e.g., retrieved from Boolean search engines like PubMed) need to be screened, or in other words, assessed by humans [14]. Screeners, typically medical researchers and librarians, judge each study based on predefined inclusion/exclusion criteria, often specified in terms of population, intervention, comparison, and outcome (PICO). Studies assessed as relevant are then re-assessed at the full-text level, so faster title and abstract screening can reduce the overall time of systematic review creation by running these two stages asynchronously. To accelerate title and abstract screening, open source tools such as ASReview [16]<sup>1</sup> and DenseReviewer [11]<sup>2</sup> have emerged; especially with recent advances in Large Language Models (LLMs), alongside the emergence of commercial tools such as

<sup>1</sup><https://asreview.nl>

<sup>2</sup><https://densereviewer.ielab.io/>



(a) AiReview screening interface with the SR Assistant Panel

(b) Model Config

(c) Prompts

**Figure 2: In the screening interface (a), users can select studies to expand for abstract screening, indicated with a purple edge. They have the option to include or exclude the selected study ① for further detailed review. LLM suggestions ② are immediately visible, as the settings are configured for Pre-reviewer ⑤ and high LLM interaction. When Co-reviewer ⑤ is enabled, users can engage the SR assistant by clicking the ‘Ask AI’ button ③, which reveals interactive features within the right panel ④. This panel includes three tabs: ‘Chat’, ‘Model Config’ (b) and ‘Prompts’ (c) for interacting with the LLM, adjusting model settings, and editing prompts, respectively. Users can start a new chat via ⑦. The response area ⑧ has LLM feedback for inclusion based on the interaction level. In ‘low’ mode, users are limited to interact with the LLM by predefined options, i.e., PICO Extraction and Detailed Reasoning ⑨, while in ‘high’ mode ⑥, users can directly prompt the LLM using ⑩. The ‘Model Config’ tab allows users to change the LLM model, temperature, and response settings. The ‘Prompts’ tab enables users to edit LLM prompts about the objective and persona, instructions in a task template, response format, and inclusion criteria.**

Elicit.<sup>3</sup> However, open source solutions—whether from developers or researchers—are usually distributed simply as raw scripts, posing a high entry barrier for screeners who conduct SRs but may lack expertise in the underlying techniques of these solutions. Commercial tools are often cautious in allowing users access to the underlying LLMs, and their implementation details are typically opaque, e.g., the prompts are not editable by end users. For example, among popular screening tools, only EPPI Reviewer currently specifies support for GPT-4-based data extraction and judgement, but it restricts this usage to post-screening evaluation scenarios.<sup>4</sup> Elicit claims to support SRs through text summarisation and intervention extraction but shows only the first few top-ranked studies.

We propose an analysis framework for categorising LLM usages by roles and interaction levels with human screeners to address the gap in the fact that no current SR software systematically integrates LLM-assisted screening. We present AiReview, a platform designed to (i) enable end users (e.g., medical researchers, librarians) with access to LLMs for SR tasks, e.g., title and abstract screening; and (ii) investigate the impact of LLMs on SR tasks, especially title and abstract screening. Our platform allows transparent control of LLMs, aligning with the guidelines for AI usage in SRs [2].

<sup>3</sup><https://elicit.com/solutions/systematic-reviews>. Note that Elicit does not perform title and abstract screening; instead, it uses LLMs for information extraction, supporting other downstream tasks.

<sup>4</sup><https://eppi.ioe.ac.uk/cms/Default.aspx?tabid=3921>

## 2 TITLE AND ABSTRACT SCREENING WITH AI REVIEW

An overview of AiReview’s architecture is shown in Figure 1. At a high-level, users upload studies retrieved from PubMed in nbib format, along with the corresponding inclusion criteria for the SR as the input for LLM-assisted screening. After the screening, users can download the screened studies. Specifically, AiReview is deployed using two individual Docker containers, which can run on a single cloud instance. The front-end container delivers a web-based interface built with Vue.js and Tailwind CSS, with Nginx serving static content. The back-end container manages the system’s core logic. It incorporates a PostgreSQL database for storing user-uploaded collections, screening results, and related data. RabbitMQ and Celery are employed for message queuing and asynchronous task management, enabling efficient handling of long-running processing requests and concurrent execution of multiple requests before http timeout. The core functionality of AiReview is implemented in Python, connecting to LLMs via external APIs to enable LLM-assisted screening. The front end communicates with the Python-based back end via REST APIs built with Django and WebSockets, enabling two-way communication between the user’s browser and the server for handling LLM streaming responses.

We present how users can leverage AiReview for their SR tasks. Figure 2a illustrates the screening interface. The interface is divided

**Table 1: Roles and interaction patterns of LLMs in systematic review screening.**

Role	Workflow	LLM Interaction Level	
		Low support	High support
<b>Pre-reviewer</b> Pre-screens studies with automated scoring and reasoning [3, 5, 7, 15]		👉 Show results upon requested	👁️ Reveal results along with studies
<b>Co-reviewer</b> Provides live assistance during human screening [1, 8, 9]		👉 Help options for predefined tasks	🗨️ In addition to options, enable chat
<b>Post-reviewer</b> Reviews human decisions and processes remaining studies [9, 13]		👉 Show LLM decisions for comparison	✖️ Check potential incorrect decisions, enable chat

into left and right panels.<sup>5</sup> The left panel lists all uploaded studies, allowing users to decide inclusion or exclusion via green and red buttons. The right panel provides the SR Assistant, which acts as a copilot during screening. During the initial setup, users can choose the LLM roles they want to use and the desired level of LLM interaction for assisting the upcoming screening task. Here we present a case where screeners enable an LLM to suggest inclusion and exclusion decisions before screening (indicated as ‘Pre’ in the AI Role) and activate an LLM to collaborate as a SR Assistant during screening (indicated as ‘Co’ in the AI Role). The screening list of studies can be sorted directly by an LLM or from other recent approaches [10, 12]. If LLM-assisted screening is enabled, users also need to provide inclusion/exclusion criteria. AiReview offers system prompts and task templates that users can edit and customise as needed. During screening, users can interact with LLMs in the side panel, according to their predefined AI interaction preferences. After screening, users can review their decisions or compare them with LLM decisions (indicated as ‘Post’ in the AI Role) and export the files (e.g., in nbib format) for downstream tasks.

Figure 2b illustrates the model configuration interface. Users can customise the SR Assistant by selecting their preferred LLM and adjusting output style at the model level. AiReview supports both commercial (e.g. OpenAI GPTs<sup>6</sup>) and open-source LLMs (e.g. Meta LLaMa series<sup>7</sup>, Mistral AI<sup>8</sup>, Deepseek<sup>9</sup>), accessible via APIs. The SR Assistant is globally controlled by the AI Interaction switch at the system level, while response characteristics—such as diversity, length, and structure—can be configured here.

Figure 2c shows the prompt interface. AiReview loads predefined prompts for LLM-assisted screening, allowing users to check and edit them as needed. The listed prompts range from general to specific: System Prompt sets the basic instructions for the LLM, Task Template defines the task for LLM, Response Format controls

the style of the response, and Inclusion Criteria is provided by users and controls the SR-Assistant screening.

### 3 FRAMEWORK FOR CATEGORISING LLM USE CASES

Table 1 presents a framework we developed and applied for AiReview that systematically categorises LLM use cases in the screening task by role, workflow, and interaction level. LLMs serve three roles in the screening workflow: pre-reviewer, co-reviewer, and post-reviewer; or *before-, with-, and after* human. We found that the LLM’s place in the workflow affects the level of bias introduced. Conceptually, having LLMs make initial judgements will substantially impact human decisions, whereas positioning LLMs after human screening will see less influence [4]. Finally, we discuss cases where screeners need different levels of support from the LLM to manage potential bias, which also guide our system design.

From this point, we introduce a ‘level of interaction’ dimension (also referred to as collaboration integration [6]), which subdivides each role based on the amount of support provided by the LLM (either high or low). Generally, the distinguishing factor between low and high interaction levels is whether the LLM’s response is displayed to screeners (e.g., visibly shown or triggered by screeners (e.g., a click ). Specifically, when the LLM is used as a pre-reviewer, recommendations and suggestions are already prepared before human screening. Thus, screeners desiring low LLM support must click the Ask AI button to display results for each study. If high support is desired, the results will be shown immediately upon entering the screening UI. When the LLM is used as a co-reviewer, screeners desiring lower LLM support have limited access to predefined LLM prompts. If high support is desired, screeners can freely chat with the LLM. When the LLM is used as a post-reviewer, the LLM can serve as a second reviewer to provide independent feedback. Similar to the pre-reviewer role, if low support is desired, the LLM will only show results for comparison. If high support is desired, the LLM will actively display comments based on screeners’ decisions, and screeners can also freely chat with the LLM.

<sup>5</sup>The SR and studies for screening are from van de Schoot et al. [17]. GPT-4o is used as the LLM, with prompts adapted from Dennstädt et al. [5].

<sup>6</sup><https://platform.openai.com/docs/models>

<sup>7</sup><https://www.llama.com/docs/model-cards-and-prompt-formats/>

<sup>8</sup>[https://docs.mistral.ai/getting-started/models/models\\_overview/](https://docs.mistral.ai/getting-started/models/models_overview/)

<sup>9</sup>[https://api-docs.deepseek.com/quick\\_start/pricing](https://api-docs.deepseek.com/quick_start/pricing)

**Table 2: LLM Pipeline Categorisation with Effort Saved.** Effort savings are conceptually deducted and represented using  $\blacksquare$  symbols, where the Full Pipeline, which provides the highest level of automation and assistance, is marked with 7  $\blacksquare$ s. Other pipelines are assigned proportionally fewer  $\blacksquare$ s based on their relative effort savings.

Category	Pipeline	Pre	Co	Post	Effort Saved
Decision-making	Pre-Only	✓	✗	✗	2 $\blacksquare$
Live Collaboration	Pre-Co Pipeline	✓	✓	✗	4 $\blacksquare$
	Co-Only	✗	✓	✗	3 $\blacksquare$
	Pre-Post Pipeline	✓	✗	✓	3 $\blacksquare$
Quality Control	Co-Post Pipeline	✗	✓	✓	3 $\blacksquare$
	Post-Only	✗	✗	✓	1 $\blacksquare$
Full Assistance	Full Pipeline	✓	✓	✓	7 $\blacksquare$

AiReview supports both single and composable pipelines through the use of the three LLM roles identified above: ‘pre-reviewer’ (P), ‘co-reviewer’ (C), and ‘post-reviewer’ (Q). We categorise the valid use cases—seven in total (P; C; Q; (P, C); (P, Q); (C, Q); (P, C, Q))—along with their associated conceptual effort savings in Table 2. Note that screeners are only able to use the post-reviewer role once all studies have been screened, and the LLM does not use any of the information from the screener to make assessments. Thus, the pre-reviewer role is designed to assist with decision-making, while the post-reviewer is designed to assist with quality control.

We estimate the effort saved by each LLM pipeline by reasoning about the amount of manual effort reduced by each role. First, we hypothesise that applying all three roles in conjunction with one another—pre-reviewer (P), co-reviewer (C), and post-reviewer (Q)—saves more effort than using any other combination of roles:  $(P, C, Q) > P \vee C \vee Q$  and  $(P, C, Q) > (P, C) \vee (C, Q) \vee (P, Q)$ . Pre-reviewer (P) reduces workload before human involvement, whereas co-reviewer (C) only assists without actively replacing manual effort, leading to  $P > C$ . In contrast, post-reviewing (Q) merely validates human decisions and does not impact initial manual effort, meaning it inherently assumes prior human involvement. Consequently, post-reviewer is the least effective in reducing conceptual effort, leading to  $P > Q$  and  $C > Q$ . Between the two combinations,  $(P, C)$  reduces effort both before and during screening, whereas  $(C, Q)$  only assists and validates post-screening, establishing  $(P, C) > (C, Q)$ . Additionally,  $(P, Q)$  saves effort before screening like  $(P, C)$ , but lacks live assistance, meaning  $(P, C) > (P, Q)$ . Since suggesting assessments before screening is more effective than during screening, we also establish  $(P, Q) > (C, Q)$ . Summarising these relationships:

$$(P, C, Q) > (P, C) > (P, Q) > (C, Q) > P > C > Q.$$

We present the ranking with  $\blacksquare$ s in Table 2. We also hypothesise that the effort savings correlate with the bias introduced by LLMs: the more human effort saved, the more bias is introduced.

**Table 3: Real-World Use Cases of LLM-Assisted Systematic Review Screening**

Scenario	Suggested Pipeline(s)	Illustrative Example
Students Learning to Screen New researchers receive real-time feedback while screening.	Co-Only Pipeline	<p>&gt;User: "I am unsure if this study meets the PICO criteria. Can you provide feedback?"</p> <p>LLM: "The study mentions the correct population but lacks details on the intervention. Check the Methods section."</p>
Resource-Limited Teams Teams with fewer screeners use the LLM as an additional reviewer for consistency.	Full Pipeline	<p>User: "We have only one reviewer. Can you act as a second screener and provide justifications?"</p> <p>LLM: "For this paper, I recommend inclusion based on the intervention. The next paper lacks a comparator and should be excluded."</p>
Quality Control After Screening LLM identifies inconsistencies post-screening, ensuring criteria adherence.	Co-Post Pipeline	<p>User: "Can you review included studies and highlight any inconsistencies?"</p> <p>LLM: "I noticed that two similar studies were handled differently. Do you want to revisit this decision?"</p>

Based on the pipelines in Table 2, Table 3 illustrates three scenarios where AiReview addresses specific needs in real-world settings. When students use AiReview to learn how to do T&A screening, the teaching team can activate the co-reviewer mode, allowing students to seek help from the SR Assistant. Students can critically evaluate the AI’s suggestions and learn from the process, even when AI hallucinates and provides incorrect information. Similarly, AiReview supports screening teams with varying sizes: as a second screener in resource-limited settings for timely progress or as an additional screener to ensure quality control.

This analysis framework is not only applicable to title and abstract screening, but can also generalise to other tasks where LLMs have potential to help. For example, in query formulation, LLMs can serve as a pre-, co-, or post-builder for Boolean queries [18], the main way studies are retrieved for systematic reviews.

## 4 CONCLUSION AND FUTURE WORK

In this paper, we introduce AiReview and our analysis framework for LLM use cases, showcasing T&A screening as an example of a LLM-assisted tool. We plan to conduct a user study using this platform to investigate how different roles and interaction levels of LLMs affect human’s screening decision and perceived utility, and if it can benefit screeners with LLM usage budgets. In the future, we will expand this platform by including LLM use cases for other SR tasks, such as Boolean query formulation and data extraction, to explore the possibility of building an end-to-end solution for systematic review creation. Additionally, we aim to scale beyond current individual screening to support collaborative screening with multiple team members, leveraging LLMs to assist in resolving disagreements. Finally, we will investigate whether LLM-assisted screening can benefit from previous non-LLM ranking methods, such as DenseReviewer [11].

## REFERENCES

- [1] Michiel P Bron, Berend Greijn, Bruno Messina Coimbra, Rens van de Schoot, and Ayoub Bagheri. 2024. Combining Large Language Model Classifications and Active Learning for Improved Technology-Assisted Review. *CEUR Workshop Proceedings* (2024).
- [2] Giovanni E Cacciamani, Timothy N Chu, Daniel I Sanford, Andre Abreu, Vinay Duddalwar, Assad Oberai, C-C Jay Kuo, Xiaoxuan Liu, Alastair K Denniston, Baptiste Vasey, et al. 2023. PRISMA AI reporting guidelines for systematic reviews and meta-analyses on AI in healthcare. *Nature medicine* 29, 1 (2023), 14–15.
- [3] Christian Cao, Jason Sang, Rohit Arora, Robert Kloosterman, Matthew Cecere, Jaswanth Gorla, Richard Saleh, David Chen, Ian Drennan, Bijan Teja, et al. 2024. Prompting is all you need: LLMs for systematic review screening. *medRxiv* (2024), 2024–06.
- [4] Alexander S Choi, Syeda Sabrina Akter, JP Singh, and Antonios Anastasopoulos. 2024. The LLM Effect: Are Humans Truly Using LLMs, or Are They Being Influenced By Them Instead? *arXiv preprint arXiv:2410.04699* (2024).
- [5] Fabio Dennstädt, Johannes Zink, Paul Martin Putora, Janna Hastings, and Nikola Cihoric. 2024. Title and abstract screening for literature reviews using large language models: an exploratory study in the biomedical domain. *Systematic Reviews* 13, 1 (2024), 158.
- [6] Guglielmo Faggioli, Laura Dietz, Charles LA Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, et al. 2023. Perspectives on large language models for relevance judgment. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*. 39–50.
- [7] Eddie Guo, Mehul Gupta, Jiawen Deng, Ye-Jean Park, Michael Paget, and Christopher Naugler. 2024. Automated paper screening for clinical reviews using large language models: Data analysis study. *Journal of Medical Internet Research* 26 (2024), e48996.
- [8] Paul Herbst and Henning Baars. 2023. Accelerating literature screening for systematic literature reviews with Large Language Models-development, application, and first evaluation of a solution.. In *LWDA*. 41–51.
- [9] Aleksi Huotala, Miikka Kuutila, Paul Ralph, and Mika Mäntylä. 2024. The Promise and Challenges of Using LLMs to Accelerate the Screening Process of Systematic Reviews. *arXiv preprint arXiv:2404.15667* (2024).
- [10] Xinyu Mao, Bevan Koopman, and Guido Zuccon. 2024. A Reproducibility Study of Goldilocks: Just-Right Tuning of BERT for TAR. In *European Conference on Information Retrieval*. Springer, 132–146.
- [11] Xinyu Mao, Teerapong Leelanupab, Harrisen Scells, and Guido Zuccon. 2025. DenseReviewer: A Screening Prioritisation Tool for Systematic Review based on Dense Retrieval. (2025). arXiv:2502.03400 [cs.IR] <https://arxiv.org/abs/2502.03400>
- [12] Xinyu Mao, Shengyao Zhuang, Bevan Koopman, and Guido Zuccon. 2024. Dense Retrieval with Continuous Explicit Feedback for Systematic Review Screening Prioritisation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2357–2362.
- [13] Takehiko Oami, Yohei Okada, and Taka-aki Nakada. 2024. Performance of a Large Language Model in screening citations. *JAMA Network Open* 7, 7 (2024), e2420496–e2420496.
- [14] Joshua R Polanin, Terri D Pigott, Dorothy L Espelage, and Jennifer K Grotjepeter. 2019. Best practice guidelines for abstract screening large-evidence systematic reviews and meta-analyses. *Research Synthesis Methods* 10, 3 (2019), 330–342.
- [15] Eugene Syriani, Istvan David, and Gauransh Kumar. 2024. Screening articles for systematic reviews with ChatGPT. *Journal of Computer Languages* (2024), 101287.
- [16] Rens Van De Schoot, Jonathan De Bruin, Raoul Schram, Parisa Zahedi, Jan De Boer, Felix Weijdema, Bianca Kramer, Martijn Huijts, Maarten Hoogerwerf, Gerbrich Ferdinand, et al. 2021. An open source machine learning framework for efficient and transparent systematic reviews. *Nature machine intelligence* 3, 2 (2021), 125–133.
- [17] Rens van de Schoot, Marit Sijbrandij, Sarah Depaoli, Sonja D Winter, Miranda Olff, and Nancy E Van Loey. 2018. Bayesian PTSD-trajectory analysis with informed priors based on a systematic literature search and expert elicitation. *Multivariate Behavioral Research* 53, 2 (2018), 267–291.
- [18] Shuai Wang, Harrisen Scells, Bevan Koopman, and Guido Zuccon. 2023. Can ChatGPT write a good boolean query for systematic review literature search?. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1426–1436.