Article

Not peer-reviewed version

Integrating Large Language Models into Systematic Review Screening

Carlo Galli [*] , Anna Viktorovna Gavrilova , Elena Calciolari

Posted Date: 14 March 2025

doi: 10.20944/preprints202503.0981.v1

Keywords: Systematic Review; Large Language Models; Text Screening; Prompt Engineering; AI-assisted Screening

Protocol

Integrating Large Language Models into Systematic Review Screening

Carlo Galli [1,*], Anna V. Gavrilova [2] and Elena Calciolari [3,4]

[1] Histology and Embryology Laboratory, Department of Medicine and Surgery, University of Parma, Via Volturno 39, 43126 Parma, Italy
[2] Department of Biosciences, University of Milan, 20122 Milan, Italy
[3] Department of Medicine and Surgery, Dental School, University of Parma, 43126 Parma, Italy
[4] Centre for Oral Clinical Research, Institute of Dentistry, Faculty of Medicine and Dentistry, Queen Mary University of London, London E1 2AD, UK
* Correspondence: carlo.galli@unipr.it

Abstract: Large Language Models (LLMs) have recently emerged as a powerful option for partially automating the labor-intensive process of screening articles in systematic reviews. Unlike traditional semi-automated platforms that rely on iterative human feedback, LLM-based pipelines can operate in a zero-shot or few-shot manner, classifying abstracts according to predefined criteria. This paper offers a step-by-step methodology for researchers, librarians, and students seeking to incorporate LLMs—such as GPT-4—into systematic reviews. It discusses required software and data preprocessing, presents various prompt strategies, and emphasizes the importance of human oversight to maintain rigorous quality control. The proposed framework aims to provide best practices and offer guidance on managing costs, reproducibility, and prompt refinement. By following these guidelines, review teams can substantially reduce screening workloads without compromising the comprehensive nature of evidence-based research.

Keywords: systematic review; large language models; text screening; prompt engineering; AI-assisted screening

1. Introduction

Systematic reviews have become essential for evidence-based decision-making in several fields and particularly in medicine, because they synthesize all relevant studies on a particular topic in a transparent and methodical way [1]. By adhering to predefined protocols and rigorous inclusion criteria, systematic reviews aim at minimizing bias and generating high-level evidence to inform policy, clinical practice, and research priorities [2]. Standard guidance on these methods emphasizes formulating a clear research question (often using the PICO framework [3]), developing a detailed review protocol, conducting a comprehensive search of multiple databases, screening studies for eligibility, extracting data, assessing quality and risk of bias, and synthesizing findings for reporting [4].

One of the most labor-intensive and error-prone stages in a systematic review is the initial screening of titles and abstracts to identify pertinent studies that can be used as a base for the systematic review [5]. Double screening by independent reviewers has long been considered the gold standard [6], yet it is equally recognized that this level of rigor demands substantial time and human resources [7]. Large academic databases often generate thousands of search results, requiring researchers to manually sift through extensive lists of potentially relevant citations to pinpoint the small number of articles that meet inclusion criteria—often fewer than a dozen for a typical review. This process makes it both logistically challenging and costly for research teams to maintain high accuracy while also managing time constraints [8]. Although searching itself is highly structured

[9]—often guided by established protocols such as the Cochrane Handbook or NICE guidelines [10,11]—many reviews still rely predominantly on manual screening methods that are vulnerable to inconsistencies across reviewers, especially when they lack extensive experience or when the volume of studies is extremely large [12]. Because the accuracy of screening directly influences the reliability of the final synthesis, poorly executed screening risks omitting critical evidence, ultimately undermining the entire review process.

In response to these challenges, semi-automated tools such as Rayyan, Abstractr, or Research Screener have emerged to assist with citation management, de-duplication, and study selection [13]. Rayyan employs a web- and mobile-based AI-assisted environment that learns from inclusion and exclusion decisions and suggests likely matches [14]. Research Screener uses deep learning and text embeddings to re-rank articles for each new judgment made by the reviewer [15]. While these semi-automated methods decrease the number of abstracts that require manual review, they typically rely on iterative human feedback to train their predictive models [16]. This approach often helps maintain high recall—the proportion of truly relevant articles identified—but still requires a prolonged "learning phase" before users realize the most significant time savings.

Alongside semi-automated platforms, a variety of additional automation efforts have sought to refine each stage of a systematic review. Some tools focus exclusively on searching, such as LitSuggest, which recommends relevant articles from PubMed [17], whereas others support more advanced tasks like data extraction (RobotReviewer, ExaCT) [18,19]. Despite their potential, full automation remains elusive, particularly in later phases of a review where human expertise is needed to interpret nuanced results [12]. Moreover, most software currently operates in isolation, forcing researchers to stitch together different tools that are not always interoperable [16].

Recent advances in natural language processing (NLP) have begun to shift the focus from traditional machine learning pipelines to modern Large Language Models (LLMs), such as GPT-4 and other state-of-the-art architectures [20]. Unlike conventional semi-automated screening tools, LLMs can classify abstracts in a zero-shot or few-shot mode simply by relying on well-structured prompts that detail inclusion and exclusion criteria [21]. Multiple studies have evaluated LLMs against human screening in diverse domains and reported encouraging results, albeit with notable variability across different models and datasets [22]. Recent studies highlight both the promise and variability of LLMs in medical literature screening. For instance, Delgado-Chaves et al. (2025) evaluated 18 LLMs across three clinical domains, observing classification accuracy ranging from 40% to 92%. Their work emphasized the critical role of human oversight, showing how iterative refinements to inclusion/exclusion criteria could substantially enhance model performance. Similarly, it has been shown that systematic prompt optimization enabled GPT-4o and Claude-3.5 to achieve sensitivities and specificities approaching 98% for thoracic surgery meta-analyses, suggesting that targeted adjustments during screening rounds yield measurable improvements [23]. Meanwhile, investigations into open-source models revealed similar dependencies on design choices: testing of four LLMs on biomedical datasets documented dramatic fluctuations in sensitivity and specificity based on model selection and prompt phrasing [24]. And even high-performing models such as GPT-4 have been reported to falter when confronted with dataset imbalances or low-prevalence agreement scenarios—a potent reminder of the persistent gap between laboratory validation and real-world application [25]. Such findings attest to the growing promise of LLMs for accelerating the screening phase of systematic reviews, but also highlight the need for human oversight in verifying edge cases and ensuring high recall.

Modern LLMs can be rapidly adapted through prompt engineering, often making them more flexible for screening tasks in varied domains [26]. Nevertheless, clear protocols and refined inclusion/exclusion criteria remain vital because even the most advanced LLM can propagate errors if initial instructions or domain-specific nuances are overlooked [23,27]. Amid these opportunities and caveats, the question is no longer whether LLMs can assist in systematic review screening, but rather how best to implement them so that they enhance speed and consistency without compromising the rigorous standards necessary for high-quality evidence synthesis [28].

The aim of this paper is therefore to provide a practical, step-by-step guide for integrating LLMs into the literature screening stage of systematic reviews, maintaining the balance between computational efficiency and the methodological rigor essential for evidence-based conclusions.

2. Methodological Proposal

2.1. Key Definitions

Systematic reviews follow predefined protocols to collect and synthesize evidence in a transparent manner [1], while LLMs are sophisticated generative models capable of interpreting language [29].

Prompt engineering is the practice of carefully crafting the instructions or queries (prompts) presented to an LLM, with the goal of eliciting the most accurate or context-appropriate response [30]. In zero-shot classification, the model applies instructions to novel tasks without prior specialized training. In few-shot classification, it can absorb context from a handful of examples provided in the prompt [31].

Recall in screening refers to the proportion of truly relevant articles the model correctly identifies, whereas precision is the proportion of articles labeled relevant that genuinely meet the inclusion criteria [32]. Throughout this paper, both recall and precision serve as indicators of screening quality.

2.2. Conceptual Rationale

The proposed methodology builds on systematic review best practices and combines them with LLM-based screening. The primary rationale is that an LLM can operate in a zero-shot or few-shot capacity, enabling efficient classification of abstracts without a lengthy training phase, and saving time for researchers to focus on other aspects of the review process. By structuring prompts around well-defined inclusion and exclusion criteria, it becomes possible to exploit an LLM's language understanding to categorize studies and identify articles that are pertinent and relevant for the systematic review. This process still requires careful human intervention, especially for checking gray areas and refining prompt wording [28].

2.3. Scope and Requirements

Researchers and students who already have a grasp of systematic review processes and basic computational techniques will find this framework particularly accessible, although the level of technical proficiency required may vary according to the chosen implementation. At a minimum, users need reliable access to a modern LLM such as GPT-4, GPT-3.5, or Deepseek r1, which can be accessed through a cloud-based API [33]; some LLMs can be installed locally if suitable hardware and software are available [34]. Establishing API-based access involves setting up credentials and ensuring a stable internet connection, whereas running an LLM locally requires significant computational resources, including a dedicated GPU with sufficient memory (usually 8–16 GB of VRAM for smaller open-source models and substantially more for larger architectures). Cloud services such as Google Colab can be a useful resource to run models remotely on platforms equipped with the necessary resources [35]. These hardware demands can influence the scale of the review and the practicalities of high-volume screening, particularly when screening thousands of abstracts.

A critical element of this setup is a robust environment for data handling and preprocessing. Python, along with commonly used libraries like pandas [36], is an efficient choice for organizing references, removing duplicates, and converting downloaded records into a uniform tabular format (e.g., pandas' DataFrame) and possibly a uniform file format for data storage (e.g., CSV). Although coding expertise does not have to be extensive, a working knowledge of Python syntax, basic scripting, and command-line tools significantly streamlines the process of merging database outputs, cleaning messy metadata, and customizing LLM prompts [37]. Familiarity with virtual environments or package managers (such as conda or pip) can be particularly helpful for maintaining consistency

doi:10.20944/preprints202503.0981.v1

and reproducibility, since the rapid pace of AI development often results in frequent updates and version changes to software packages [38].

Teams should also consider the potential costs associated with API-based LLM services, especially if the review involves screening large numbers of abstracts [39]. Balancing the benefits of higher accuracy from more advanced models with the financial impact of repeated queries is vital for long-term feasibility. If budgets are constrained, smaller open-source models may provide an adequate starting point, even though they occasionally require more extensive prompt tuning or additional error checking to reach acceptable levels of recall and precision [40–42]. For users planning to work on private or sensitive datasets, local deployment of open-source or self-hosted models can address data security concerns, but this option does increase the burden of setup, hardware maintenance, and ongoing troubleshooting [43–46].

2.4. Prerequisites

Before integrating LLMs into the screening phase of a systematic review for clinical medicine, several fundamental methodological elements must be in place. The first consideration is a clear research question supported by fully defined inclusion and exclusion criteria, often summarized through frameworks such as PICO (Population, Intervention, Comparison, Outcome) or one of its close variations [3,47–49]. For instance, a review investigating "the effectiveness of antihypertensive Drug A versus placebo in reducing systolic blood pressure among adults with hypertension" would specify:

- Population: Adults aged 18–75 with primary hypertension,
- Intervention: Daily oral administration of Drug A,
- Comparison: Placebo,
- Outcome: Mean change in systolic blood pressure at 12 weeks.

PICO is a useful heuristic tool to breakdown a relevant clinical question into its constitutive components, so that an effective search strategy can be drafted, but also, as we will show, an effective LLM prompt.

Dai et al. demonstrated that the precision of an LLM's output depends substantially on how accurately these criteria are translated into prompts or instructions [23]. Articulating the review question in detail ensures that the model can target specific populations, interventions, and outcomes without veering into irrelevant territory [50].

3. Step-by-Step Methodology

3.1. Conduct a Broad Database Search

The screening workflow begins (Figure 1) with a comprehensive search for potential studies in relevant databases [51,52]. Database access forms a cornerstone of any systematic review [53,54]. Researchers should have access to the comprehensive or domain-specific databases—such as Medline, Embase, Scopus, or specialized repositories—to capture a broad array of publications [51,53–55]. Medline is undoubtedly the most renowned literature database in biomedicine and can be freely accessed both through its PubMed web portal but also directly through command line via API, using specific python libraries, such as Biopython [56], which is advantageous because retrieved articles can be stored in data structures that can be passed directly to LLMs.

In most systematic reviews, querying the databases involves crafting detailed strategies that reflect the components outlined by PICO. Queries are often expanded to include synonyms, related keywords, and Medical Subject Headings (MeSH), if applicable, to avoid overlooking relevant articles [57]. These elements can be combined in a database-specific syntax to query the database [58]. For example, a review evaluating *"the efficacy of cognitive behavioral therapy (CBT) versus antidepressants for reducing depressive symptoms in adolescents"* might generate a PubMed query structured as:

```
("Adolescent"[MeSH] OR "teen*"[tiab] OR "youth"[tiab])
AND ("Cognitive Behavioral Therapy"[MeSH] OR "CBT"[tiab])
```

AND (“Antidepressive Agents”[MeSH] OR “SSRI”[tiab] OR “SNRI”[tiab])
AND (“Depression”[MeSH] OR “depressive symptoms”[tiab])
AND (“Treatment Outcome”[MeSH] OR “remission”[tiab])

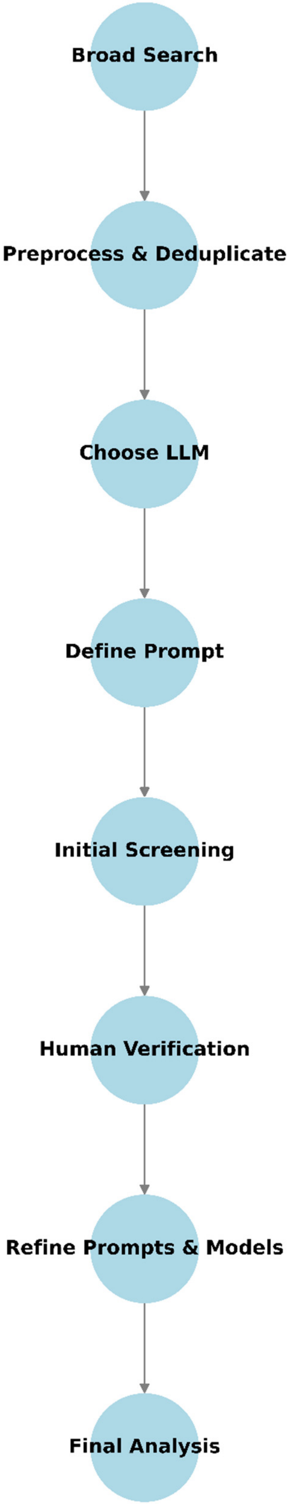

Figure 1. Flowchart of the LLM-based screening process. The diagram illustrates the sequential steps involved in utilizing a large language model (LLM) for systematic literature screening.

doi:10.20944/preprints202503.0981.v1

Once the searches are executed, the resulting citations are exported in a consistent format—such as CSV, RIS, or XML—that can be imported and processed by data-analysis libraries in Python (or other environments). If the query is conducted by accessing PubMed via API, the results can be stored as a python data structure (e,g, a pandas' DataFrame), ready for subsequent processing.

3.2. Preprocess and Deduplicate Records

After collating results from multiple databases, the records must be cleaned and standardized. This often involves removing exact duplicates—an issue that arises frequently when the same article appears across different platforms—and converting files to a uniform character encoding (e.g., UTF-8) to avoid text corruption. Researchers would typically unify column headers for Title, Abstract, Publication Date, and other relevant metadata so that subsequent methods, including LLM-based screening, can operate without confusion. Attention to detail at this stage pays off in later steps, as it prevents misclassification and missed articles caused by inconsistent data fields [59]. While preprocessing sounds mundane, data cleaning can dramatically improve downstream performance by reducing irrelevant noise that misleads both conventional machine learning pipelines and modern LLMs [27].

3.3. Choose the Right LLM(s) and Prompt

At the heart of any systematic review pipeline augmented by LLMs lies a twofold decision: which model to deploy and how to craft its prompts to optimize recall and precision. Smaller-scale, open-source architectures like GPT-2 or Flan T5 might suffice for pilot studies or when hardware and budget constraints prohibit more advanced solutions, yet these simpler models can struggle with contextual complexity, potentially requiring additional prompt engineering to maintain accuracy [60]. More recent and larger architectures, such as GPT-3.5, GPT-4, or other open-source models like the Deepseek family, have demonstrated superior performance in a variety of tasks, including domain-specific literature screening, but come at the cost of increased computational overhead and possible API usage fees.

Comparative evaluations consistently show that larger models can excel in contextual awareness, preserving coherence when confronted with intricate abstracts, while smaller models often produce output more quickly but risk overlooking nuanced details. Generation speed and context management can vary widely across models depending on both the underlying architecture and the target hardware environment [61].

The choice of model also intersects closely with the available infrastructure. Deployed solutions that rely on cloud-based APIs (e.g., GPT-3.5 or GPT-4 via OpenAI) offer scalability but can become expensive at high volumes and raise concerns over data security if abstracts contain sensitive information [43]. Even among API-based solutions, performance can differ if concurrency limits or prompt length restrictions apply, as larger context windows deliver more accurate results but also increase both processing time and token-related costs [62]. These efficiency considerations become especially relevant in systematic reviews, which can easily involve thousands of abstracts to classify. Overly long prompts, although potentially more instructive, may result in slower inference speeds, and a near-linear increase has been observed in total processing time in models receiving large prompt sizes [61]. Researchers must weigh the complexity of their prompts—particularly if they include multiple inclusion/exclusion criteria or domain-specific nuances—against the desire for rapid classification.

Developing effective prompts remains the other crucial pillar in model selection. Even advanced models with large context windows can produce erratic outputs if the instructions conflict or are excessively vague. Slight rewording of a prompt can either inflate false positives (when instructions are too permissive) or inadvertently exclude relevant studies (when instructions are too strict) [23,27].

3.4. Understanding Prompt Fundamentals and Challenges

A prompt refers to the instructions, context, or background information given to a LLM so that the model can respond in a manner consistent with the user's objectives [63]. Unlike traditional machine learning classifiers that rely on iterative retraining, modern LLMs use these prompts as immediate instructions, which guide the model's behavior. The structure of a well-crafted prompt typically includes a concise statement of the task (for example, "You are assisting in a systematic review"), any relevant context (such as inclusion/exclusion criteria or a description of the population and interventions), the textual data to be analyzed (i.e., the title or abstract), and explicit output instructions (indicating whether to "ACCEPT" or "REJECT"). In the context of systematic reviews, prompts often encode key methodological requirements—whether defined via PICO or other frameworks—so that an LLM can scan each abstract for relevant details like patient characteristics, study design, or reported outcomes [64].

Below is a template for a possible prompt that uses PICO criteria for a literature search of RCTs:

System: You are an AI assistant helping with a systematic review on [TOPIC OR CONDITION].

User Prompt: You will decide if each article should be ACCEPTED or REJECTED based on the following criteria:

Population (P): Adult patients (≥18 years) with [SPECIFIC POPULATION OR CONDITION]. If the abstract does not mention age, or does not clearly describe non-adult populations, do not penalize.

Intervention (I): Must involve [INTERVENTION 1] combined with [INTERVENTION 2]. If either is implied or partially mentioned, do not penalize.

Comparison (C): Ideally a group that uses [CONTROL OR COMPARISON], or some control lacking [KEY INTERVENTION]. If not stated but not contradicted, do not penalize.

Outcomes (O): Must measure [PRIMARY OUTCOME] or at least mention [SECONDARY OUTCOMES OR RELEVANT PARAMETERS]. If the abstract does not state outcomes explicitly but mentions [RELEVANT OUTCOME KEYWORDS], do not penalize.

Study design: Must be an RCT or strongly imply random allocation. If uncertain, do not penalize.

Follow-up: Minimum [X] months. If not stated or unclear, do not penalize unless it says <[X] months.

Decision Rule: If no criterion is explicitly violated, respond only with "ACCEPT." If any criterion is clearly contradicted (e.g., non-randomized design, pediatric population, <[X] months follow-up), respond with "REJECT." Provide no additional explanation.

Title: {title}

Abstract: {abstract}

Researchers can enhance prompt clarity by stripping away extraneous details, ensuring that essential instructions are easily distinguishable from background information. In systematic review applications, this often means specifying the precise triggers for "ACCEPT," e.g., randomized study designs and adult populations, versus the explicit triggers for "REJECT," e.g., purely animal research or pediatric cohorts.

The complexity of biomedical abstracts, which may discuss multiple interventions, outcomes, or populations, can pose a challenge if prompts are too broad, too vague, or contain contradictory statements. For instance, telling the LLM to accept studies if they mention any adult participants but then also requesting rejection if the study includes children under 18 could lead to confusion if the abstract features a mixed population. Careful wording of the criteria or prompt instructions can thus mitigate incorrect interpretations, a phenomenon made more likely when dealing with large corpora.

Another key challenge is ensuring that the model does not "hallucinate" details not actually present in the abstract [65]. Because LLMs are probabilistic text generators trained on diverse textual corpora, they can sometimes invent content—such as extra interventions, specific follow-up durations, or outcome measures—simply because the prompt or question implies these details are relevant, and countermeasures to mitigate this phenomenon are a fertile area of investigation [66–68]. A simple approach could be just encouraging the model to cite the exact words or phrases in the

doi:10.20944/preprints202503.0981.v1

abstract that justify its decision, although verifying the accuracy of these cited quotes still requires careful human oversight.

Prompt refinement typically evolves through iterative testing. Many researchers begin with a "soft" or inclusive instruction set that aims to maximize recall, then review a subset of "Accepted" outputs to identify obvious false positives that indicate the need for more stringent language. Likewise, a "strict" approach can guard against irrelevant articles but risks excluding borderline studies whose abstracts do not explicitly list every inclusion criterion. In such instances, a prompt that directs the model to label a study as "INSUFFICIENT INFORMATION" may help flag ambiguous cases for further manual review. Domain-specific jargon or abbreviations also introduce complexity, since the LLM might misinterpret specialized terms or incorrectly infer the presence of required conditions [69]. For instance, a study might use "RCT" in the text but never explicitly mention "randomized controlled trial," leading certain prompts to accept or reject the article prematurely if they only look for the spelled-out term. Researchers should therefore tailor prompts to the language patterns common in the target domain, possibly by leveraging known synonyms or by describing relevant terms in the instructions ("Consider an 'RCT' the same as a 'randomized controlled trial'"). Even in a best-case scenario, LLMs might misclassify abstracts that mention unclear or conflicting details. While advanced LLMs have grown remarkably adept at context-sensitive classification, no prompt can capture every edge case in biomedical literature, particularly in specialized reviews that examine niche interventions or unique study designs. Documenting prompt versions, analyzing errors, and iterating toward more precise instructions remain central to balancing recall, precision, and cost efficiency for large-scale screening efforts.

3.5. Perform Initial Screening

Once the prompt strategies are established, the LLM can be applied to classify each abstract as either "Accepted" or "Rejected." This step typically involves passing the abstract text and the relevant prompt to the LLM and collecting the output in a structured data frame. Metadata such as timestamps, model version, or confidence levels (if provided by the API or tool) can also be recorded for subsequent auditing and reproducibility. Consistent recordkeeping at this juncture lays the groundwork for quality assurance and the potential to replicate the screening approach in the future [12].

3.6. Human Verification and Error Analysis

Human expertise remains indispensable in systematic reviews, even when leveraging advanced language models [70]. Researchers typically begin by scrutinizing a subset of "Rejected" articles to identify false negatives—studies that were incorrectly excluded. If borderline cases appear in this category, adjustments to prompt wording or acceptance thresholds may be necessary. Conversely, a quick review of "Accepted" abstracts helps detect obvious false positives. This iterative feedback loop, reminiscent of semi-automated screening tools [14,15], can be accomplished more rapidly and flexibly through zero-shot or few-shot prompting in LLMs. Refinements continue until the screening team is satisfied that the model reliably captures relevant studies without becoming overly permissive. Once optimized, these prompt settings can be incorporated into ongoing or future screening efforts, with systematic refinement shown to improve both recall and precision while reducing reviewer workload [23]. Figure 2 illustrates a possible workflow for prompt refining.

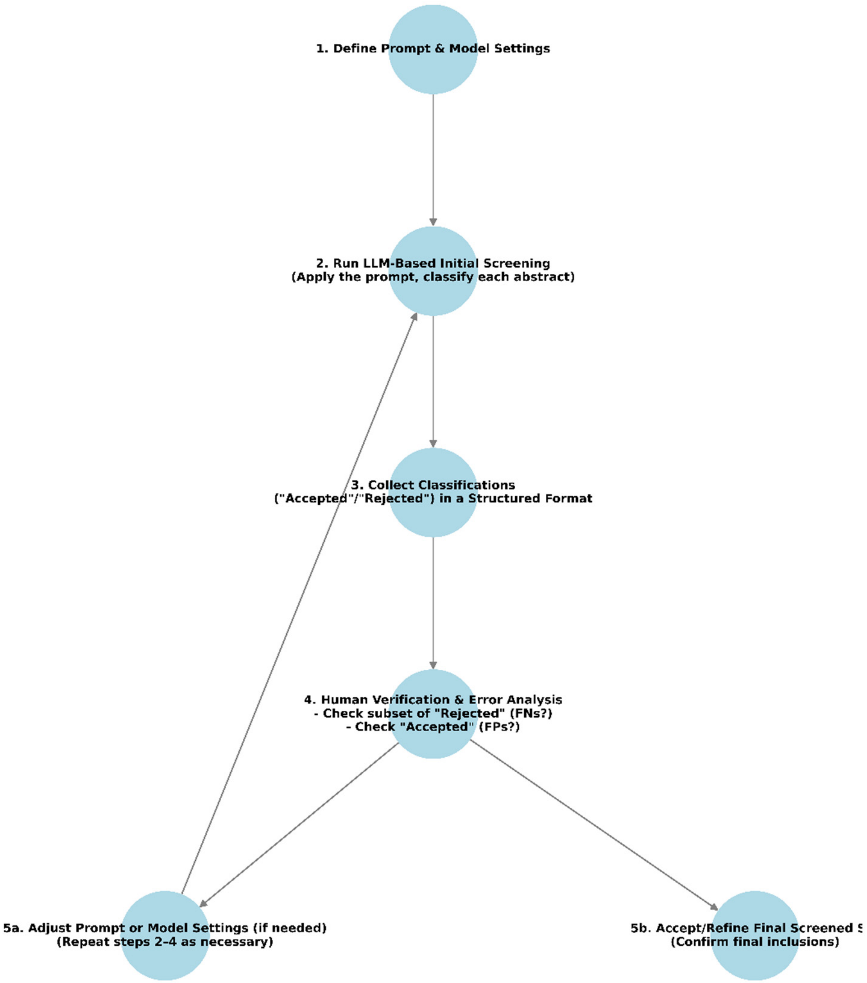
1. Define Prompt & Model Settings
2. Run LLM-Based Initial Screening
(Apply the prompt, classify each abstract)
3. Collect Classifications
("Accepted"/"Rejected") in a Structured Format
4. Human Verification & Error Analysis
- Check subset of "Rejected" (FNs?)
- Check "Accepted" (FPs?)
5a. Adjust Prompt or Model Settings (if needed)
(Repeat steps 2–4 as necessary)
5b. Accept/Refine Final Screened S
(Confirm final inclusions)

Figure 2. Flowchart of the prompt refining process. The diagram illustrates the sequential and iterative steps involved in refining the prompt for LL-based literature screening.

3.7. Best Practices and Recommendations

Maintaining high recall is critical, as omitting relevant articles risks weakening the entire review. To mitigate this, a consensus favors maximizing inclusivity during database searches and the early screening phases—accepting borderline cases to avoid excluding key studies prematurely [71–74].

Clear inclusion/exclusion criteria are essential to reduce unpredictable misclassifications, and all changes to prompts or model settings should be carefully documented, including the rationale and observed impacts on accuracy or recall. Pilot tests screening small sets of known abstracts may be very valuable for surfacing issues early, preventing downstream errors that might otherwise emerge only after processing thousands of articles. While these checks may seem laborious, they safeguard reproducibility and avert more significant oversights.

Cost is another practical consideration, particularly when using proprietary models with API-based pricing [75]. Although ongoing refinement and error analysis may eventually lower expenses by reducing unnecessary queries, research teams must weigh the financial overhead of repeated API calls against potential performance gains. A flexible, layered approach often balances efficiency and rigor effectively. Early screening rounds benefit from broad prompts and inclusive language to preserve potentially relevant studies, while later phases can adopt stricter criteria or incorporate prior labels to filter clearly irrelevant articles, thereby improving precision and reducing the full-text workload. Throughout this process, error analysis—especially the detection of false negatives—remains central to safeguarding evidence integrity. Strategically limited manual checks, such as

doi:10.20944/preprints202503.0981.v1

random sampling of "Rejected" articles or verification of ambiguous abstracts, confirm model reliability without requiring exhaustive rechecks [27], preserving the time-saving advantages of automation.

Ultimately, LLMs should be viewed as high-efficiency filters that augment—rather than replace—expert judgment. Whether employing a single inclusive prompt strategy or a multi-stage filtering model, iterative refinement and selective validation allow the method to adapt to the review's scope, resource constraints, and citation volume.

4. Discussion

The adaptability of LLMs offers a clear advantage over more rigid machine learning models [76]. In zero-shot or few-shot modes, the model's performance depends heavily on how well the prompt captures the essence of the inclusion and exclusion criteria. It has been shown that refining those criteria substantially boosts accuracy and can approach human-level recall [23,27]. These gains do not negate the importance of human expertise. Rather, oversight remains pivotal to interpret borderline abstracts, continually adjust prompts, and preserve the rigor of evidence synthesis [77].

Beyond these technical considerations, recent literature emphasizes the need for explicit guidelines to optimize LLM usage in research contexts [78–83], highlighting the importance of transparency in disclosing AI involvement and the ethical requirement of human accountability. As the importance of AI is growing exponentially in science as much as in everyday life, education on the strengths and limitations of language models is critical to users.

Scholars caution that LLMs should not replace expert judgment; rather, they should enhance it by rapidly filtering large volumes of text, provided their outputs are continually verified and documented for reproducibility. Transparency about any AI-assisted workflow is essential to maintain scientific integrity [78], and risks like hallucinations and bias must be mitigated through ongoing validation. [83] Likewise, Raj et al. suggest that structured methods—whether fine-tuning or retrieval-augmented techniques—can boost performance, but only if accompanied by guidelines that ensure data curation and prompt engineering are implemented consistently and responsibly [79,84]. Adopting such measures is of particular importance when applying LLM-based screening to medical fields, given the high stakes of omitting relevant studies or introducing biased results into the evidence base [80].

The workflow outlined in this paper addresses many of the time and resource challenges associated with traditional screening, but it also introduces new considerations, such as how to manage prompt complexity, maintain cost-effectiveness when making numerous API calls, and log each classification for reproducibility. These authors are convinced that combining LLM technologies with robust oversight, adherence to ethical standards, and comprehensive user training, will ensure that these tools bolster rather than compromise the credibility of systematic reviews.

5. Conclusions

Integrating LLMs into systematic review screening offers substantial time savings during the initial evaluation of abstracts, particularly when well-defined inclusion criteria are applied. The use of LLMs not only potentially expand the scope of literature that can feasibly be screened but also alleviate the burden on research teams. By selecting appropriate LLMs, tailoring prompts to their capabilities, conducting manual validation, and documenting iterative refinements, it is possible to achieve robust recall and precision while maintaining the integrity of evidence-based conclusions. Standardized guidelines are however needed to integrate these powerful instruments into the routine of literature screening. As LLMs and AI-based solutions continue to evolve, the step-by-step framework presented here provides a starting point for leveraging these tools responsibly and effectively in systematic review processes.

Author Contributions: Conceptualization, C.G., and E.C..; methodology, C.G.; software, A.V.G.; writing—original draft preparation, C.G.; writing—review and editing, A.V.G. and E.C.; All the authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No data were generated.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Mulrow CD (1994) Systematic Reviews: Rationale for systematic reviews. BMJ 309:597–599. https://doi.org/10.1136/bmj.309.6954.597
2. Parums D V (2021) Review articles, systematic reviews, meta-analysis, and the updated preferred reporting items for systematic reviews and meta-analyses (PRISMA) 2020 guidelines. Med Sci Monit 27:e934475-1
3. Methley AM, Campbell S, Chew-Graham C, et al. (2014) PICO, PICOS and SPIDER: a comparison study of specificity and sensitivity in three search tools for qualitative systematic reviews. BMC Health Serv Res 14:579. https://doi.org/10.1186/s12913-014-0579-0
4. Linares-Espinós E, Hernández V, Domínguez-Escrig J, et al. (2018) Methodology of a systematic review PALABRAS CLAVE
5. Dickersin K, Scherer R, Lefebvre C (1994) Systematic reviews: identifying relevant studies for systematic reviews. Bmj 309:1286–1291
6. Greenhalgh T, Thorne S, Malterud K (2018) Time to challenge the spurious hierarchy of systematic over narrative reviews? Eur J Clin Invest 48:e12931. https://doi.org/10.1111/eci.12931
7. Waffenschmidt S, Knelangen M, Sieben W, et al. (2019) Single screening versus conventional double screening for study selection in systematic reviews: a methodological systematic review. BMC Med Res Methodol 19:132. https://doi.org/10.1186/s12874-019-0782-0
8. Cooper C, Booth A, Varley-Campbell J, et al. (2018) Defining the process to literature searching in systematic reviews: a literature review of guidance and supporting studies. BMC Med Res Methodol 18:1–14
9. Furlan JC, Singh J, Hsieh J, Fehlings MG Reviews Methodology of Systematic Reviews and Recommendations
10. Cumpston M, Li T, Page MJ, et al. (2019) Updated guidance for trusted systematic reviews: a new edition of the Cochrane Handbook for Systematic Reviews of Interventions. Cochrane Database Syst Rev 2019:ED000142
11. Dunning J, Lecky F (2004) The NICE guidelines in the real world: a practical perspective. Emerg Med J 21:404
12. Van Dinter R, Tekinerdogan B, Catal C (2021) Automation of systematic literature reviews: A systematic literature review. Inf Softw Technol 136:106589
13. Wang Z, Nayfeh T, Tetzlaff J, et al. (2020) Error rates of human reviewers during abstract screening in systematic reviews. PLoS One 15:e0227742
14. Ouzzani M, Hammady H, Fedorowicz Z, Elmagarmid A (2016) Rayyan—a web and mobile app for systematic reviews. Syst Rev 5:1–10
15. Chai KEK, Lines RLJ, Gucciardi DF, Ng L (2021) Research Screener: a machine learning tool to semi-automate abstract screening for systematic reviews. Syst Rev 10:93. https://doi.org/10.1186/s13643-021-01635-3
16. Khalil H, Ameen D, Zarnegar A (2022) Tools to support the automation of systematic reviews: a scoping review. J Clin Epidemiol 144:22–42. https://doi.org/10.1016/j.jclinepi.2021.12.005
17. Allot A, Lee K, Chen Q, et al. (2021) LitSuggest: a web-based system for literature recommendation and curation using machine learning. Nucleic Acids Res 49:W352–W358

18. Marshall IJ, Kuiper J, Wallace BC (2016) RobotReviewer: evaluation of a system for automatically assessing bias in clinical trials. Journal of the American Medical Informatics Association 23:193–201
19. Kiritchenko S, De Bruijn B, Carini S, et al. (2010) ExaCT: automatic extraction of clinical trial characteristics from journal publications. BMC Med Inform Decis Mak 10:1–17
20. Sindhu B, Prathamesh RP, Sameera MB, KumaraSwamy S (2024) The evolution of large language model: Models, applications and challenges. In: 2024 International Conference on Current Trends in Advanced Computing (ICCTAC). IEEE, pp 1–8
21. Cao C, Sang J, Arora R, et al. (2024) Prompting is all you need: LLMs for systematic review screening. medRxiv 2024–2026
22. Scherbakov D, Hubig N, Jansari V, et al. (2024) The emergence of Large Language Models (LLM) as a tool in literature reviews: an LLM automated systematic review. arXiv preprint arXiv:240904600
23. Dai Z-Y, Shen C, Ji Y-L, et al. (2024) Accuracy of Large Language Models for Literature Screening in Systematic Reviews and Meta-Analyses
24. Dennstädt F, Zink J, Putora PM, et al. (2024) Title and abstract screening for literature reviews using large language models: an exploratory study in the biomedical domain. Syst Rev 13:158. https://doi.org/10.1186/s13643-024-02575-4
25. Khraisha Q, Put S, Kappenberg J, et al. (2024) Can large language models replace humans in systematic reviews? Evaluating <scp>GPT</scp> -4's efficacy in screening and extracting data from peer-reviewed and grey literature in multiple languages. Res Synth Methods 15:616–626. https://doi.org/10.1002/jrsm.1715
26. Blevins T, Gonen H, Zettlemoyer L (2022) Prompting Language Models for Linguistic Structure
27. Delgado-Chaves FM, Jennings MJ, Atalaia A, et al. (2025) Transforming literature screening: The emerging role of large language models in systematic reviews. Proceedings of the National Academy of Sciences 122:. https://doi.org/10.1073/pnas.2411962122
28. Lieberum J-L, Töws M, Metzendorf M-I, et al. (2025) Large language models for conducting systematic reviews: on the rise, but not yet ready for use—a scoping review. J Clin Epidemiol 111746. https://doi.org/10.1016/j.jclinepi.2025.111746
29. Zhao WX, Zhou K, Li J, et al. (2023) A survey of large language models. arXiv preprint arXiv:230318223
30. Gao A (2023) Prompt engineering for large language models. Available at SSRN 4504303
31. Dang H, Mecke L, Lehmann F, et al. (2022) How to prompt? Opportunities and challenges of zero-and few-shot learning for human-AI interaction in creative applications of generative models. arXiv preprint arXiv:220901390
32. Cottam JA, Heller NC, Ebsch CL, et al. (2020) Evaluation of Alignment: Precision, Recall, Weighting and Limitations. In: 2020 IEEE International Conference on Big Data (Big Data). IEEE, pp 2513–2519
33. Wang Y, Yu J, Yao Z, et al. (2024) A solution-based LLM API-using methodology for academic information seeking. arXiv preprint arXiv:240515165
34. Kumar BVP, Ahmed MDS (2024) Beyond Clouds: Locally Runnable LLMs as a Secure Solution for AI Applications. Digital Society 3:49
35. Bisong E (2019) Google Colaboratory. In: Bisong E (ed) Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners. Apress, Berkeley, CA, pp 59–64
36. Mckinney W (2010) Data Structures for Statistical Computing in Python. In: van der Walt S, Millman J (eds) Proceedings of the 9th Python in Science Conference. pp 51–56
37. Grigorov D (2024) Harnessing Python 3.11 and Python Libraries for LLM Development. In: Introduction to Python and Large Language Models: A Guide to Language Models. Springer, pp 303–368
38. Maji AK, Gorenstein L, Lentner G (2020) Demystifying Python Package Installation with conda-env-mod. In: 2020 IEEE/ACM International Workshop on HPC User Support Tools (HUST) and Workshop on Programming and Performance Visualization Tools (ProTools). IEEE, pp 27–37
39. Shekhar S, Dubey T, Mukherjee K, et al. (2024) Towards optimizing the costs of llm usage. arXiv preprint arXiv:240201742

40. Irugalbandara C, Mahendra A, Daynauth R, et al. (2024) Scaling down to scale up: A cost-benefit analysis of replacing OpenAI's LLM with open source SLMs in production. In: 2024 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). IEEE, pp 280–291
41. Ding D, Mallick A, Wang C, et al. (2024) Hybrid llm: Cost-efficient and quality-aware query routing. arXiv preprint arXiv:240414618
42. Chen L, Zaharia M, Zou J (2023) Frugalgpt: How to use large language models while reducing cost and improving performance. arXiv preprint arXiv:230505176
43. Yan B, Li K, Xu M, et al. (2024) On protecting the data privacy of large language models (llms): A survey. arXiv preprint arXiv:240305156
44. Yao Y, Duan J, Xu K, et al. (2024) A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. High-Confidence Computing 100211
45. Huang B, Yu S, Li J, et al. (2023) Firewallm: A portable data protection and recovery framework for llm services. In: International Conference on Data Mining and Big Data. Springer, pp 16–30
46. Feretzakis G, Verykios VS (2024) Trustworthy AI: Securing sensitive data in large language models. AI 5:2773–2800
47. Cooke A, Smith D, Booth A (2012) Beyond PICO. Qual Health Res 22:1435–1443. https://doi.org/10.1177/1049732312452938
48. Frandsen TF, Bruun Nielsen MF, Lindhardt CL, Eriksen MB (2020) Using the full PICO model as a search tool for systematic reviews resulted in lower recall for some PICO elements. J Clin Epidemiol 127:69–75. https://doi.org/10.1016/j.jclinepi.2020.07.005
49. Brown D (2020) A Review of the PubMed PICO Tool: Using Evidence-Based Practice in Health Education. Health Promot Pract 21:496–498. https://doi.org/10.1177/1524839919893361
50. Scells H, Zuccon G, Koopman B, et al. (2017) Integrating the Framing of Clinical Questions via PICO into the Retrieval of Medical Literature for Systematic Reviews. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. ACM, New York, NY, USA, pp 2291–2294
51. Chigbu UE, Atiku SO, Du Plessis CC (2023) The Science of Literature Reviews: Searching, Identifying, Selecting, and Synthesising. Publications 11:2. https://doi.org/10.3390/publications11010002
52. Patrick LJ, Munro S (2004) The literature review: demystifying the literature search. Diabetes Educ 30:30–38
53. Heintz M, Hval G, Tornes RA, et al. (2023) Optimizing the literature search: coverage of included references in systematic reviews in Medline and Embase. Journal of the Medical Library Association 111:599–605. https://doi.org/10.5195/jmla.2023.1482
54. Lu Z (2011) PubMed and beyond: a survey of web tools for searching biomedical literature. Database 2011:baq036–baq036. https://doi.org/10.1093/database/baq036
55. Page D Systematic Literature Searching and the Bibliographic Database Haystack
56. Cock PJA, Antao T, Chang JT, et al. (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics 25:1422–1423
57. Lu Z, Kim W, Wilbur WJ (2009) Evaluation of query expansion using MeSH in PubMed. Inf Retr Boston 12:69–80
58. Stuart D (2023) Database search translation tools: MEDLINE transpose, ovid search translator, and SR-accelerator polyglot search translator. Journal of Electronic Resources in Medical Libraries 20:152–159
59. Yang M, Adomavicius G, Burtch G, Ren Y (2018) Mind the gap: Accounting for measurement error and misclassification in variables generated via data mining. Information Systems Research 29:4–24
60. Galli C, Colangelo MT, Guizzardi S, et al. (2025) A Zero-Shot Comparison of Large Language Models for Efficient Screening in Periodontal Regeneration Research. Preprints (Basel). https://doi.org/10.20944/preprints202501.2029.v1
61. Agarwal L, Nasim A (2024) Comparison and Analysis of Large Language Models (LLMs)
62. Wu Y, Gu Y, Feng X, et al. (2024) Extending context window of large language models from a distributional perspective. arXiv preprint arXiv:241001490
63. Beurer-Kellner L, Fischer M, Vechev M (2023) Prompting is programming: A query language for large language models. Proceedings of the ACM on Programming Languages 7:1946–1969

64. Colangelo MT, Guizzardi S, Meleti M, et al. (2025) How to Write Effective Prompts for Screening Biomedical Literature Using Large Language Models. Preprints (Basel). https://doi.org/10.20944/preprints202502.0396.v1
65. Huang L, Yu W, Ma W, et al. (2025) A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. ACM Trans Inf Syst 43:1–55. https://doi.org/10.1145/3703155
66. Bhattacharya R (2024) Strategies to mitigate hallucinations in large language models. Applied Marketing Analytics 10:62–67
67. Gosmar D, Dahl DA (2025) Hallucination Mitigation using Agentic AI Natural Language-Based Frameworks. arXiv preprint arXiv:250113946
68. Hassan M (2025) Measuring the Impact of Hallucinations on Human Reliance in LLM Applications. Journal of Robotic Process Automation, AI Integration, and Workflow Optimization 10:10–20
69. Mai HT, Chu CX, Paulheim H (2024) Do LLMs really adapt to domains? An ontology learning perspective. In: International Semantic Web Conference. Springer, pp 126–143
70. Duenas T, Ruiz D (2024) The risks of human overreliance on large language models for critical thinking. Research Gate, 2024e URL http://dx doi org/1013140/RG 2:
71. Page MJ, Higgins JPT, Sterne JAC (2019) Assessing risk of bias due to missing results in a synthesis. Cochrane handbook for systematic reviews of interventions 349–374
72. Goossen K, Tenckhoff S, Probst P, et al. (2018) Optimal literature search for systematic reviews in surgery. Langenbecks Arch Surg 403:119–129
73. Ewald H, Klerings I, Wagner G, et al. (2022) Searching two or more databases decreased the risk of missing relevant studies: a metaresearch study. J Clin Epidemiol 149:154–164
74. Cooper C, Varley-Campbell J, Carter P (2019) Established search filters may miss studies when identifying randomized controlled trials. J Clin Epidemiol 112:12–19
75. Wong E (2024) Comparative Analysis of Open Source and Proprietary Large Language Models: Performance and Accessibility. Advances in Computer Sciences 7:1–7
76. Ray S (2019) A Quick Review of Machine Learning Algorithms. In: 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon). IEEE, pp 35–39
77. Tang X, Jin Q, Zhu K, et al. (2024) Prioritizing safeguarding over autonomy: Risks of llm agents for science. arXiv preprint arXiv:240204247
78. Kim JK, Chua M, Rickard M, Lorenzo A (2023) ChatGPT and large language model (LLM) chatbots: The current state of acceptability and a proposal for guidelines on utilization in academic medicine. J Pediatr Urol 19:598–604. https://doi.org/10.1016/j.jpurol.2023.05.018
79. Ranjan R, Gupta S, Singh SN (2024) A comprehensive survey of bias in llms: Current landscape and future directions. arXiv preprint arXiv:240916430
80. Ullah E, Parwani A, Baig MM, Singh R (2024) Challenges and barriers of using large language models (LLM) such as ChatGPT for diagnostic medicine with a focus on digital pathology–a recent scoping review. Diagn Pathol 19:43
81. Barman KG, Wood N, Pawlowski P (2024) Beyond transparency and explainability: on the need for adequate and contextualized user guidelines for LLM use. Ethics Inf Technol 26:47
82. Barman KG, Caron S, Claassen T, De Regt H (2024) Towards a benchmark for scientific understanding in humans and machines. Minds Mach (Dordr) 34:6
83. Jiao J, Afroogh S, Xu Y, Phillips C (2024) Navigating llm ethics: Advancements, challenges, and future directions. arXiv preprint arXiv:240618841
84. Patil R, Gudivada V (2024) A review of current trends, techniques, and challenges in large language models (llms). Applied Sciences 14:2074

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.