From Studies to Supported Claims: A Survey of Automated Survey Generation

Abstract

Automated survey generation uses language models to turn scholarly literature into reviews, from related-work sections to systematic reviews and meta-analyses. Fluent drafts can omit eligible studies, separate extracted values from their populations and time points, and attach real citations to unsupported claims. We survey methods through the steps that connect studies to claims: selecting papers, recording findings with their conditions, organizing the evidence and writing comparisons grounded in sources. A running example, a review of exercise as a treatment for depression, shows how errors at each step affect the review. We compare methods across review types and distinguish systems that revise only their prose from those that can also change their outline or search for missing evidence. Evaluations show that strong presentation scores coexist with weak informational depth and that the accuracy of model judges varies by dimension. User studies show that assistance can save time while leaving researchers to verify the evidence and its interpretation. Progress requires keeping findings tied to their sources and conditions through revision, and identifying which earlier checks an update invalidates.

1 Introduction

A literature review gathers studies on a topic, organizes them and explains what they establish together and what remains open. Reviews let researchers place a new contribution, learn an unfamiliar field or judge the evidence for a decision. They range from a paper's related-work section to a topic survey, a systematic review that answers a focused question from all eligible studies, or a meta-analysis that pools results statistically. Each form requires researchers to select sources, interpret their findings and decide what can be concluded across them. This work is slow, and reviews in active fields soon fall out of date [1, 2]. Large language models (LLMs) can draft fluent summaries, but summarizing papers individually leaves the hardest part of a review undone: deciding which studies belong together and what their combined evidence supports.

To take on that work, systems have expanded beyond summarizing supplied papers. Related-work generators select and connect findings from a target paper's references [3, 4]. Starting from a topic adds discovery and organization: systems search for papers, plan the account and draft a whole survey [5, 6]. Protocol-guided workflows also screen and extract results for a specified question [1], drawing on systematic-review automation, a sibling field that has long assisted these stages without necessarily writing the review [7, 8]. We call this family *automated survey generation*: systems that select and organize findings from scholarly studies to write a review whose claims cite their supporting sources. It is the scholarly counterpart of broader deep-research agents, which search and write reports from sources of any kind [9]. Within scholarly literature, the request ranges from one scientific question [10] to a field-wide survey or a protocol-defined evidence synthesis.

Consider a review of exercise as a treatment for depression. Summarizing five supplied trials requires describing each one faithfully. Comparing them also requires knowing whether they studied similar populations, measured the same outcomes and used comparable controls. Two difficulties come from reviewing itself, and automation makes them harder to detect. (i) *Coverage*

is hard to establish. Relevant trials may use terms the request omits [11], and screening can discard eligible studies even after retrieval [12]. (ii) *Findings depend on their conditions.* A trial of exercise against a waiting list cannot establish that exercise works as well as antidepressants. Extractors can also pair a subgroup's outcome with the whole sample's size [1]; pooling then spreads the error into the review's conclusion [13]. A readable draft can hide both missing studies and invalid comparisons, so checking its prose alone will miss failures in the evidence beneath it.

Automation also makes two familiar reviewing errors harder to detect: plausible organization and fluent revisions can conceal changes that a researcher did not make or check. (iii) *Plausible outputs can hide missing evidence.* Sensible taxonomy labels can leave studies unassigned [2], just as real citations can accompany unsupported statements [14]. (iv) *Revision can undo a check.* Rewriting can change a verified claim [15], and a new outline can leave evidence under obsolete headings [16]. We therefore follow the connections from studies to claims (Figure 1): discovery and selection (§3), evidence representation (§4), organization (§5) and cited synthesis (§6). Protocol-guided synthesis makes the conditions for extraction and pooling explicit (§7). Whole systems differ in which earlier decisions they can revise (§8), while evaluation (§9) and researcher oversight (§10) must catch errors that survive into the draft. Comparing these connections explains why a strong writing stage cannot compensate for missing or misrecorded evidence.

Prior surveys examine these connections within particular review types. Accounts of related-work generation compare extractive and abstractive methods and representations of supplied references [3, 4]. Surveys of systematic-review automation follow review stages and researcher involvement, finding that screening receives more support than extraction and synthesis [7, 8], including in recent LLM workflows [17]; a review of automated meta-analysis also covers statistical pooling [18]. The broader deep-research literature organizes systems around planning, question development, web exploration and report generation [9]. These accounts leave methods for the same evidence problem separated by review type. Checking a citation against a passage and checking an extracted effect against a table both ask whether a claim preserves what its source establishes. Comparing them together exposes which checks can be shared and which depend on the review's purpose. Our contributions are:

- A formulation shared by related-work sections, topic surveys, systematic reviews and meta-analyses. It separates the studies a system selects, the records it reads from them, the structure it builds over them and the claims that cite them, so that each can be checked on its own.
- A comparison of revision designs by the information available to later steps and the earlier decisions they can change. It explains which failures paragraph rewriting, replanning and renewed evidence acquisition can each repair.
- A synthesis of evaluation evidence showing that strong presentation scores coexist with weak informational depth. User studies also show why the time saved by assistance must be assessed alongside the verification it leaves to researchers.

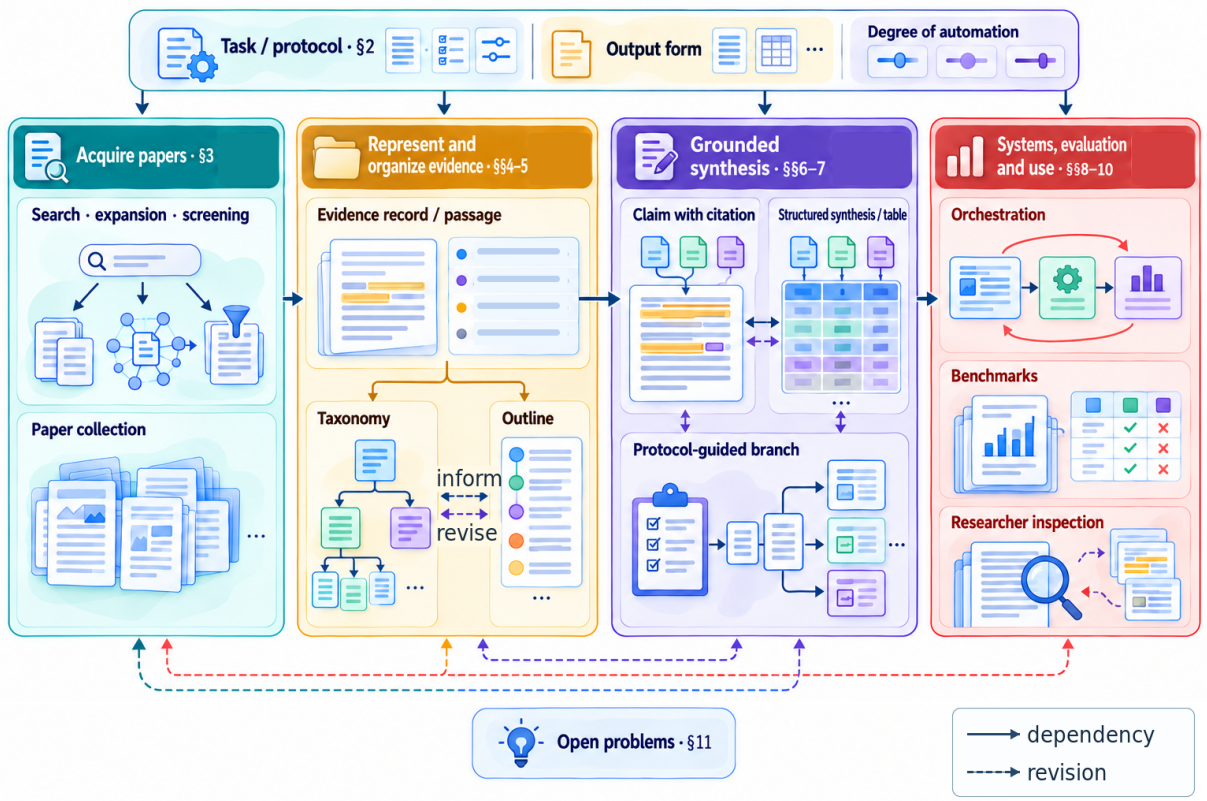


Figure 1: The task and protocol govern the path from study selection to cited claims (§§3–6), including extraction and pooling when required (§7). Whole systems decide which steps can be revisited (§8); evaluation checks their outputs (§9), and researchers inspect and correct their decisions (§10). Solid arrows show what each step depends on, and dashed arrows show revisions from later steps to earlier ones. The resulting dependencies define open problems for maintaining supported reviews (§11) [1, 10, 19–22].

2 Foundations

Scholarly reviewing turns studies into an account of what their evidence establishes. We distinguish three questions: what the review must establish (§2.1), how systems represent the evidence and its organization (§2.2), and how methods vary in their inputs and decisions (§2.3).

2.1 Review types

A reviewer begins with a question or topic, gathers studies that bear on it, reads their findings and organizes an account that cites its evidence. Review types differ in how the reviewer chooses the studies and combines their findings. We distinguish the evidence selected from the claims written so that each can be checked separately.

The input to an automated system is a request q and a collection of documents $\mathcal{D}$, such as a bibliographic database or a set of supplied papers. A system selects a *study set* $S \subseteq \mathcal{D}$, which should match the target set S^* of studies called for by the request. Reading a study yields an *evidence record*, which keeps its finding together with its conditions. We write it as $e = (d, v, \kappa)$: a finding or value v, the study $d \in S$ that reports it, and the conditions κ under which it holds, such as the population, the comparison, the time point or the dataset. The system organizes the studies by assigning them to the categories of a taxonomy or the sections of an outline (§2.2), and it writes a text made of claims. Each claim cites the selected studies that establish it. *Citation validity* means that a citation names a real study in S. *Claim support* means that the cited studies establish the claim under the conditions it states.

In the running example, q asks for a review of exercise as a treatment for depression, and $\mathcal{D}$ is a biomedical database. Suppose the request includes only trials in adults diagnosed with depression. The table below follows the review from search to writing and shows a failure at each step. These failures are independent: a system can find every eligible trial and still misrecord their results, or record them correctly and cite them for claims they do not make. The final row separates citation validity from claim support. The trial exists and belongs in S, but its waiting-list comparison cannot establish equivalence to antidepressants.

Object	In the running example	One way it fails
Study set S	Trials of exercise for depression found in a biomedical database	A trial of a 'physical activity program' is missed, and a yoga trial in students without a diagnosis is admitted
Record $e = (d, v, \kappa)$	A trial's reduction in symptom scores, with its arm, scale and time point	The reduction is kept, but not that it held only in the supervised arm at twelve weeks
Assignment	Trials placed under aerobic, resistance and mind–body exercise	The aerobic category is sensible, but some aerobic trials are placed elsewhere
Claim	'Exercise works as well as antidepressants', citing trial A	Trial A is real and in S, but it compared exercise with a waiting list

Related-work sections When the request supplies a target paper, the review exists to position that paper's contribution. A related-work section for a new trial of supervised walking in older adults with depression would stress the earlier trials it departs from, such as trials in younger adults or of unsupervised exercise, and pass quickly over the rest. The references are usually supplied, so S is given, and the system's work is to choose what each reference contributes and how it relates to the target paper [4]. The task therefore tests selection and positioning within sources that the authors chose, not discovery.

Topic surveys and maps When the request names only a topic, the system must also choose the questions that organize the review. A *map* of the literature reports which topics and kinds of study occur: how many exercise trials test aerobic training and how many test yoga, and in which populations. An *explanatory survey* also says how the approaches differ and what their results establish together, for instance whether supervised programs report larger improvements than unsupervised ones and, if so, whether the trials differ in other ways that could explain it. Maps and explanatory surveys both rest on a taxonomy that groups the studies (§5), but only the explanatory survey has to justify comparisons across its groups.

Systematic reviews A *systematic review* specifies which studies belong in the target set S^* before the search begins. Its *protocol* states the question, the eligibility criteria and the data to extract from each study [7]. For the running example, the question might be whether supervised exercise reduces depressive symptoms in adults diagnosed with major depression, compared with usual care. The criteria would then admit only randomized trials in diagnosed adults, and the extraction plan would list the symptom scale, the time point and the group sizes. Systems described as systematic-review automation implement these constraints to different degrees. One multi-agent prototype generates the question and search strings, filters and extracts, and compiles a report, but it states no exclusion criteria, so nothing fixes which studies it should leave out [23]. Explicit criteria give another team a basis for reproducing and checking the inclusion decisions.

Meta-analyses A *meta-analysis* derives its central claim by computation. It pools the effects reported by the included trials into one estimate and tests their *heterogeneity*, whether the trials disagree more than chance would allow [18]. A *forest plot* shows each trial's effect beside the

pooled estimate. Pooling needs more than the numbers. Each effect must be converted to a common scale, and two records can be pooled only when their conditions match the question: the same comparison, a compatible outcome and a comparable time point. A table listing each trial's reported improvement lets a reader compare the trials, but it is not yet input to a meta-analysis if one trial reports a change on a clinician-rated scale at eight weeks and another reports remission rates at six months.

2.2 Intermediate artifacts

Automated systems make the evidence and organization of Section 2.1 explicit in intermediate artifacts. They read studies into records, group the records or studies, and use those groups to plan and write the review. Each artifact exposes a different decision for checking: a researcher can inspect a misplaced trial in a taxonomy before it becomes part of a misleading comparison.

A *taxonomy* classifies a field through categories, relations between categories, and the assignment of studies to categories. Relations and assignments can be judged separately (§5.1). An *outline* orders sections, each with the questions it answers and the studies assigned to it. The same taxonomy supports different outlines. One outline gives each type of exercise its own section. Another asks in turn whether exercise helps, for whom and compared with what, and compares the types of exercise within each of those sections.

An *evidence schema* names the attributes to compare across studies, such as the type of exercise, the population, the comparator and the effect, and a *comparison table* fills those attributes for each study. Choosing useful columns and filling the cells faithfully are different tasks, and some systems split them: one proposes columns from titles and abstracts, then fills cells from full texts (ArxivDIGESTables [20]). Columns also need interpretation. A generated column called 'Weekly dose' may hold the same information as a reference column called 'Exercise volume', so a metric that compares column names exactly will miss the match [20]. A schema is defined by what its columns mean, not by what they are called.

A *citation graph*, built from the studies' reference lists, represents documents as nodes and citations as directed edges. It helps to find studies (§3.1) and to group them (§5.1), although its edges do not say how two studies relate (§4.2).

2.3 Development and design space

Automated survey generation has gradually taken over decisions that earlier systems left to supplied inputs or to researchers (Figure 2). Early work wrote related work from supplied references [4], generated chapters from cited abstracts [24] or screened abstracts against review criteria [25]. Alongside this single-stage work, starting from a topic made finding studies part of the task, with agents learning to search and follow references [19]. Comparing the papers required records and structure: extracted tables and taxonomies made their attributes and groupings explicit [2, 20]. Work on cited claims connected drafting to retrieval and source checks [10], while protocol-guided synthesis joined screening and extraction to standardized outcomes [1]. In parallel, whole systems combined these decisions, moving from fixed topic-to-draft pipelines [5, 26] toward feedback that can reopen the plan or the search [22, 27]. The five lines differ in which decision they automate and which intermediate product they check.

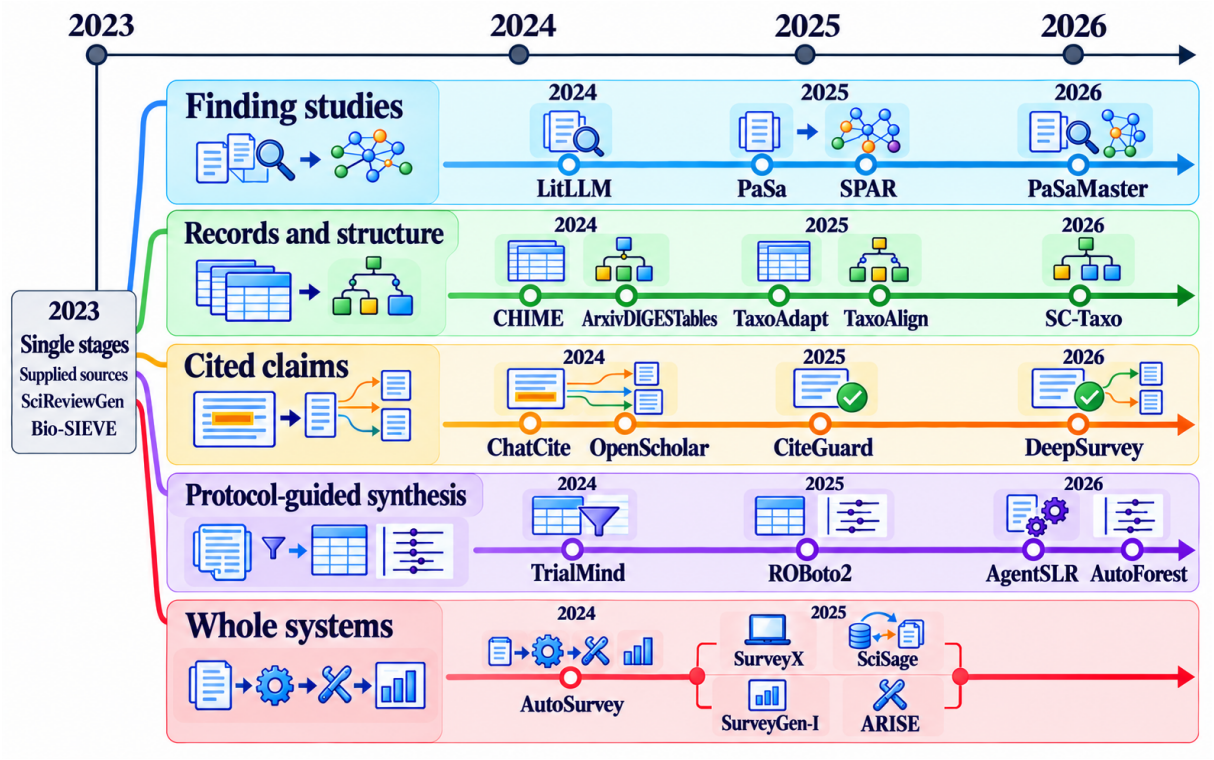


Figure 2: Automated survey generation expanded from single stages with supplied sources into five lines of work: finding studies (§3), building records and structure (§§4–5), grounding claims (§6), protocol-guided synthesis (§7) and whole systems (§8) [1, 2, 5, 6, 10, 19–22, 24, 25, 27–39].

We compare the resulting design space through five decisions: how to find and select studies (§3); what to retain from them and how to organize it (§§4–5); how to write claims grounded in their sources (§6); how to apply a protocol when extracting and pooling results (§7); and which earlier steps later feedback can reopen (§8). Three input assumptions cut across these decisions.

Supplied sources versus discovered sources Whether S is supplied or must be found changes what a good output shows. Given the trials, a system shows that it can describe them accurately. Given only the topic, it must also show that it found the right trials. Datasets for related-work and table generation supply the documents [20, 40], whereas review workflows search and filter before they synthesize [23]. A system can do one of these well and the other badly while producing an equally polished review.

Abstracts versus full texts Full texts contain the methods and results that abstracts leave out, including many of the conditions κ that a record needs, which is why the table generator of Section 2.2 fills its cells from full texts [20]. More text does not help by itself. In one related-work evaluation, giving a model full texts instead of abstracts lowers the share of its statements that the sources support, and more of its correct statements cite the wrong paper [40]. Access to full text shifts the problem to selecting what the writer should see (§4).

Scope and autonomy Supplied references, a fixed question and a fixed outline each remove a decision from the system. The task should identify who sets eligibility, chooses comparison columns, changes scope and accepts the text. A system can extract for a researcher's question or generate questions from a topic [23]. In an exercise topic survey, adding adolescents can expose missing work. In a systematic review, the same change rewrites eligibility and changes which studies support the conclusions.

3 Literature discovery and selection

Discovery builds the study set S: it should recover the eligible studies in S^* and exclude other papers. Search gathers candidates, including work that the request's wording does not reach, and selection decides which candidates belong. A broad search increases the screening burden, while strict selection risks dropping eligible studies. The selected papers also supply terms and references for further search. We compare how systems expand and refine searches (§3.1), then how they rank, screen and retain the candidates (§3.2).

3.1 Literature search

A search finds a study only by following some link to it. The link can be wording or meaning shared with the request, a citation from a paper already found, or a subtopic that earlier results suggest. Each kind of link reaches studies that the others miss, and each also brings in papers that do not belong. A study with no link to anything already found stays out of view, so the results alone cannot show when the search is complete.

Keyword and dense retrieval The first obstacle is vocabulary. A search for 'exercise' and 'depression' can miss a trial titled 'A walking program for older adults with major depressive disorder'. Benchmarks for scientific search test this case by keeping generated questions only when they share few words with the titles of the papers they ask for (LitSearch [11]). On such questions, *dense retrievers*, which compare learned representations of the question and the paper, place a relevant paper among their first five results far more often than keyword search does. Reranking their candidates with an LLM helps further, while commercial search engines do worse than even keyword search [11]. Keyword search can close part of the gap when an LLM first expands the request with synonyms and related terms. On one biomedical benchmark, expanded queries place about three in five source articles among the top 200 results, against one in two for a keyword query weighted by term frequency, and they add little when the questions already resemble the article titles [41]. A supplied target paper can guide both steps: its abstract supplies keywords and a basis for reranking the results [28]. Learned matching and query expansion both help bridge vocabulary differences, with expansion adding most when the original request poorly matches the article titles.

Citation expansion Citations reach papers missed by query matching. A trial of yoga for depression may never match a search about exercise, yet it appears in the reference list of a trial that compared yoga with aerobic training. *Co-citation*, which links papers that are often cited together, reaches papers cited alongside the ones already found, such as the early trials that later trials all cite even when their titles speak of 'physical training'. One survey system expands its retrieval along co-citation links and reranks the candidates by relevance, citation impact and diversity, and the references it selects overlap more with those of human-written surveys than those chosen by naive retrieval (SurveyGen [42]). Citation expansion therefore changes the selected bibliography, improving agreement with human surveys beyond query-based retrieval alone.

The search procedure can be trained on complete exploration sessions rather than specified only through prompts. For example, PaSa [19] divides the work between a crawler that searches and follows references and a selector that reads titles and abstracts to judge relevance. Candidates wait in a queue until the selector reaches them, so the crawler can explore widely without committing to every paper it finds. Trained with reinforcement learning over whole search sessions, a small model running this procedure finds more of the papers that real researchers' queries call for than Google search with queries rewritten by a large model, and more than a

much larger model prompted to follow the same procedure [19]. On these researcher queries, training the search policy matters more than model size alone.

The balance between finding more papers and admitting irrelevant ones is measured with precision and recall. *Precision* P is the fraction of retrieved papers that are relevant. *Recall* R is the fraction of all relevant papers retrieved. *F1* combines them as $F_1 = 2PR/(P + R)$. In a system that decomposes the request, follows one layer of references and judges relevance with separate agents, the reference-following step raises recall on two benchmarks but lowers precision even after relevance filtering, because it greatly enlarges the pool of candidates (SPAR [29]). The same tradeoff separates whole systems. On one benchmark, the systems favor different goals. (i) A deployed search assistant recovers the most relevant papers at very low precision. (ii) The trained crawler recovers nearly as many at low precision. (iii) The decomposing system recovers about half as many with the highest F1 [29]. The more a system expands, the more it depends on selection to keep S close to S^*.

Refining queries from results The papers found so far can also show what to search for next. Researchers can narrow a broad query by choosing among subtopics found in its initial results, either through a displayed grouping or in dialogue [43, 44]. Generating a Boolean query from a chosen subtopic improves precision among the first 20 results over the original query [43]. An agent can make the same move itself. One planner writes a checklist of search constraints and dispatches agents to search, follow citations and verify web results. It changes strategy when their evidence leaves checklist items unmet (PaSaMaster [30]). A draft can drive the search as well. Turning key statements of a draft answer into verification questions and searching for them finds references beyond those that supported the first draft, and raises recall against the reference lists of published reviews [41]. In the running example, a draft sentence about exercise in older adults would send the search after trials in that population, which the original query treated as a minor variation. Subtopics, unmet constraints and unsupported draft claims each give the next search a target that the original request did not specify.

Stopping Searches stop for different reasons. (i) A learned policy ends exploration, or its candidate queue runs out [19]. (ii) A fixed depth limit caps reference following [29]. (iii) A researcher expands from a chosen core, prunes each round and decides when to stop [45]. None of these signals detects an eligible line of work outside the explored network. A run that has followed every reference of the exercise trials it found can still miss all trials of tai chi, if the papers it reads never cite that work. When agents are asked to collect every paper that meets several constraints, the sets they return overlap little with the target set [46]. The risk is greatest for claims of absence: a review that says no trial has tested exercise in adolescents is only as reliable as the search that found none.

3.2 Study selection

Selection decides which candidates enter S, and its errors are not symmetric. A wrongly included study can still be removed at a later stage, but a wrongly excluded one is lost, so screening for systematic reviews favors recall [47]. Ordering candidates and learning from labels reduce the work of screening; retaining decisions avoids repeating that work during an update.

Ranking for screening Ranking decides which candidates a reviewer reads first; *screening* applies eligibility criteria to decide which to retain. A ranker can put likely eligible papers first by scoring abstracts against the review question and criteria, then breaking ties in a second ranking step (LGAR [48]). Without training labels for the review, this ranker places eligible studies higher than a previous method that screens by answering questions about each abstract,

and stating the research questions explicitly improves the ranking further [48]. Some criteria require full text, so screening datasets also pair complete papers with review criteria [49]. A ranking of abstracts is best used to decide what to read first, not what to exclude.

Adaptive screening A ranking made before screening can use the question and criteria, but each subsequent label also says something about eligibility. *Active learning* uses these labels to train a classifier and asks the reviewer for the next labels that would help it most. The choice of which papers to ask about can change too: uncertainty directs the reviewer to borderline candidates, while diversity reaches underrepresented parts of the pool. A *bandit algorithm* learns from past rewards which strategy to try next. In simulated screening, choosing strategies this way finds eligible studies with fewer labels than keeping one strategy fixed, even with few initial labels (AutoDiscover [50]). Learning which strategy to use reduces the labeling work within an already retrieved collection.

Refreshing the collection A review of exercise for depression starts to age as soon as a new trial appears. *Living reviews* repeat their searches on a schedule and screen only what is new. This requires storing decisions as well as papers: after searching several scholarly engines and removing duplicates, a review toolbox carries earlier labels forward into the next round (CRUISE-Screening [47]). New labels can then serve a second purpose. Recording a classifier's prediction before the reviewer decides lets its accuracy be measured on decisions it could not have seen. Keeping the sequence of predictions and decisions supports both review maintenance and evaluation during use.

4 Evidence representation

Evidence representation determines whether a finding v stays tied to its study d and conditions κ in the record $e = (d, v, \kappa)$. A one-sentence summary of an exercise trial keeps the improvement v but can drop the arm and the time point in κ. A field for the effect size keeps the number but can lose the sentence that qualifies it, and a link between two trials says that they are related without saying whether they agree. Representations built from single papers decide how much of a paper's text and conditions a record carries (§4.1), and representations that relate papers decide what a system can say about two studies together (§4.2).

4.1 Selective reading

A system cannot read every page of every paper, and more text does not always help, so it chooses what to read. Each choice can lose something that later steps need. A passage left unread may hold the condition that explains a result, and a compressed note may drop that condition after it was read. Systems reduce this loss by reading the passages a record needs and by storing each finding with its conditions and its source.

Selected passages and extraction Finding the right passage and extracting what it says are separate problems, and the second can remain hard after the first is solved. One pipeline starts from a finding stated in the abstract, locates the paragraphs that state the underlying hypothesis, and then retrieves the paragraphs that report its statistical evidence [51]. On social and behavioral science articles, retrieving targeted paragraphs extracts hypotheses better than prompting with the full text, and supplying the correct paragraph helps much more, so locating the passage accounts for most of the loss in hypothesis extraction. Statistical evidence is different. Its extraction stays only moderately accurate even from the correct paragraph, because the extractor has to keep a test statistic together with the variables and the relationship it describes

[51]. For an exercise trial, the record must keep the effect size together with its groups and time point. Targeted reading locates that evidence; extracting the conditions κ preserves what the value means.

Shared working memory Selective reading can decide not only which passage to extract but which paper to open next. A bounded working memory supplies that decision with a compressed account of what has already been read. For example, a related-work generator rewrites its memory after each section, then chooses the next paper and section from that memory, the abstracts and its reading history (Select, Read, and Write [52]). Restricting the next choice to citation-graph neighbors keeps related papers together. Under this restriction, an LLM judge rates the sections above those written from all references in one long context or from sentences chosen to match the reference section [52]. Compression therefore changes the reading path as well as the writer's input: a discarded condition may be unavailable both for interpreting an earlier finding and for choosing the next source.

Structured records and provenance Structured fields let a system compare findings by their conditions; links to the source let a researcher check the fields. An infectious-disease dashboard illustrates the value of keeping conditions beside a number. It stores reproduction-number estimates with their intervals, methods, locations and dates, and links each record to its publication [53]. Researchers can inspect the setting and uncertainty before comparing estimates.

The conditions of a finding are often spread across text, tables and figures: the methods section describes the intervention, a table gives the group sizes, and a plot shows the outcome over time. One curation harness gives an agent a tool for each format, such as section-level text, parsed tables, and gridlines for reading values off plots, and it records which text, table or figure each extracted value came from, its *provenance* (Beaver [54]). On a curation task built from published Alzheimer's disease studies, the harness agrees with expert-curated records far more often than general-purpose agents, and the largest gains come on attributes that require combining evidence across formats [54]. A record that keeps its conditions and its provenance lets a reader who doubts a value return to the cell or the plot it came from.

4.2 Cross-paper representations

A review must say how studies relate, not only that they do. Citations and shared terms are cheap to collect. They show which studies belong together, but not whether two studies agree. Representations that say how studies relate, such as written comparisons or shared attributes, need more structure. A system must extract that structure from the papers, or people must supply it.

Citation and semantic context A citation graph supplies context about a paper's place in the literature. Adding a learned representation of that graph to an extractor improves its identification of salient entities and relations among tasks, datasets, methods and metrics [55]. The graph says where a paper sits in the literature, but its edges do not say how two papers relate. A trial that cites another may confirm it, contradict it or mention it in passing. Semantic similarity has the same limit, since two abstracts can be close in wording and report opposite effects. To tell whether two trials agree, or whether their conditions are compatible, a system needs their records and an explicit relation between them.

Passages and entity graphs Shared entities help find related studies, while source passages preserve the qualifications around their findings. Normalizing 'HAM-D' and 'Hamilton Depression Rating Scale' to one node would connect trials that use different names for the same scale. A

direct comparison indexes the same polymer-literature corpus as both paragraphs and a graph of subject–relation–object tuples [56]. The graph retrieves needed context more often, but its answers retain less detail about the conditions of findings than passage-based answers do. The two representations serve complementary purposes: normalized entities connect the studies, and the passages behind them preserve what each finding means.

Explicit paper relationships and research chains Relations between papers can also be written down. One approach summarizes each paper's objectives, methods, findings and contributions, and then describes in natural language how each pair of papers relates and how each paper is used by those that cite it [57]. *Research chains* serve a narrower purpose: they extract pairs of motivation and method, together with experimental results, so that a reader can follow both the methods and the reported results in publication order (ResearchPulse [58]). In the running example, such a chain would show how exercise programs changed across trials and whether the reported effects changed with them. Separating a paper's findings from descriptions of its links to other papers lets a researcher check a claimed development against both studies.

Evidence maps Shared concept types let a collection answer the aggregate questions that a map of the literature asks (§2.1). One system extracts concepts such as models, datasets and data types from titles and abstracts, stores them with paper metadata, and builds graphs of which concepts occur together, so that researchers can ask, for example, which data types a field uses most [59]. An extractor can miss concepts, and two concepts in the same abstract may never have been tested together [59]. Linking counts back to study records gives a researcher a way to check what was counted. Another system extracts the methods each study used into a knowledge graph and counts how often each method appears among the empirical studies in its collection on ozone exposure and cardiovascular disease, with each count traceable to the studies behind it (HySemRAG [60]). Curated knowledge graphs take the opposite route. Domain experts define reusable templates and annotate contributions against them, so papers can be compared on shared properties, at the cost of extensive manual work [61]. The more explicit the comparison, the more structure has to be extracted or supplied.

Schemas that follow the review's question A comparison table needs a schema, and the right schema depends on what the comparison is for. Trials of exercise for depression could be compared by dose for a clinician choosing a program, or by how likely their designs are to have biased the results, for a methodologist judging the evidence. When generated columns are compared with those of a human-written table, human judges find that many columns missing from the reference are still useful [20], so the reference schema is one good answer among several. Stating the purpose of the table as an explicit question improves schema generation over inferring it from the table's caption and the sentences that cite it [62]. A schema is a choice about what the review will compare, so it should come from the review's question rather than from the papers alone.

5 Taxonomies and survey outlines

Taxonomies group studies, while outlines turn those groups into a sequence of questions (Figure 3). Both require checking their contents as well as their labels: a sensible category can omit relevant trials, and a well-ordered section can repeat an earlier one. Systems build taxonomies from citation communities, expert examples or the corpus itself, and check them at several levels (§5.1). They plan outlines from the topic alone, from published surveys or from a taxonomy, and some keep revising the outline while they read and write (§5.2).

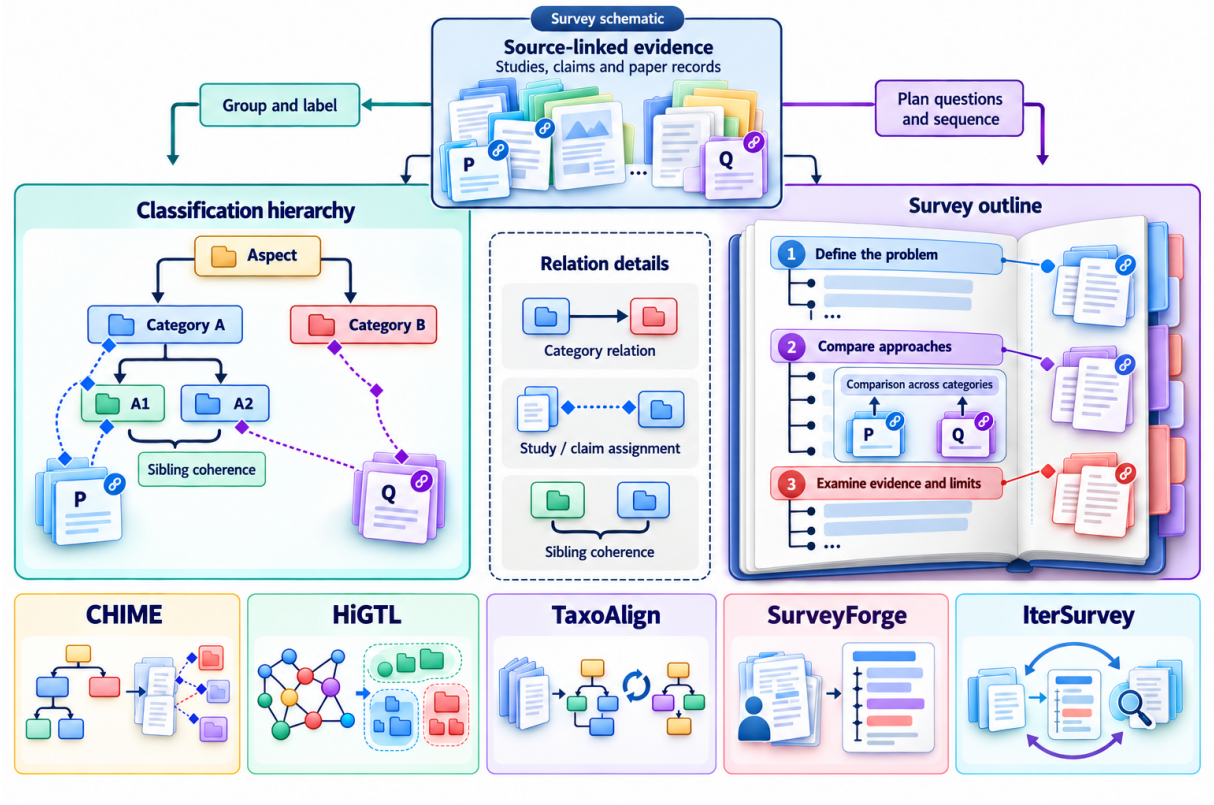


Figure 3: A taxonomy groups studies, while an outline orders the questions answered about them. The taxonomy (left) can be judged in three separate ways, shown in the center: (i) whether each category belongs under its parent, (ii) whether sibling categories form a coherent group and (iii) whether each claim or study is assigned where it belongs. The outline (right) can draw studies from different categories into one comparison [2, 16, 31, 63, 64].

5.1 Scholarly taxonomy construction

A field can be divided in several reasonable ways, and one study often fits more than one division. With no single right answer to match, methods take their evidence for a division from the field's citations, from published surveys or from the papers themselves. They must still check that each category sits at the right level and that each study appears wherever it belongs.

Where the structure comes from Citation communities and expert examples provide different starting points for a taxonomy. One method clusters a citation graph recursively, using each paper's content as its features, and has an LLM name each cluster, so the groups follow citation communities (HiGTL [63]). Citations can separate lines of work that share vocabulary: trials of yoga and trials of aerobic exercise both report 'depressive symptoms', but they can build on different earlier studies. On citation graphs drawn from computer-science reviews, the resulting taxonomies match the reviews' own more closely than those of text-based and prompted baselines [63]. Another method learns how published surveys divide a field. It writes a taxonomy from topic-relevant slices of the papers with an instruction-tuned model, and a stronger model then checks each parent–child link and its grounding in the slices (TaxoAlign [31]). Its training targets keep the topical structure of published surveys and drop sections such as introductions and conclusions, and its taxonomies align better with human-written ones than those from direct prompting [31]. Citations supply a grouping signal that the text alone lacks, while expert examples supply the shape that readers of a survey expect.

Adapting the structure to the corpus A fixed criterion and a fixed depth fit some parts of a literature better than others. Generating candidate aspects at each node lets the criterion change as the topic narrows: the top of the exercise taxonomy might divide trials by type of exercise, and the node for aerobic exercise might divide them by population [65]. Generated aspects recover reference headings and categories better than a fixed set, but match the reference table of contents less closely. Increasing the number of clusters also makes the groups purer while fragmenting the hierarchy [65]. The corpus can also show where a taxonomy should grow. One method classifies papers along separate dimensions, such as tasks, methods and datasets, and deepens a node when many papers map to it and widens it when many papers fit none of its children (TaxoAdapt [21]). In the exercise review, a crowded node for aerobic exercise would be split further, and trials of tai chi that fit neither the aerobic nor the resistance category would prompt a new sibling. LLM judges rate the resulting taxonomies as more coherent and better at keeping a consistent granularity than those of competing methods [21]. Adapting the criterion and the depth to the corpus puts detail where the evidence is dense, as long as the partitions stay coarse enough to form a usable hierarchy.

Checking a concept's level Cluster purity alone cannot place a concept in a hierarchy. Trials that all monitored heart rate form a coherent cluster, yet 'heart-rate monitoring' is a measurement, not a kind of 'aerobic exercise'. The parent–child checks used when learning from survey examples address this distinction [31]. A complementary approach separates the tree of document clusters from the tree of concepts and reconciles them level by level (SC-Taxo [32]). It generates a heading with its parent and children in view, then checks across siblings for overlap and mismatched granularity. Removing either check lowers reference agreement [32]. Both approaches judge a category's relation to other categories, beyond how well its label describes its documents.

Category relations and claim assignments Category labels can pass expert inspection while their contents remain incomplete. In a claim-based hierarchy, each abstract is condensed into claims and each claim may belong to several categories or none (CHIME [2]). Experts judge parent–child links, sibling coherence and claim assignments separately. They accept almost all parent–child links, find sibling groups less consistently coherent, and find many missing study assignments. Training a small model on their corrections improves assignment F1 by about 13 points [2]. The distinction changes what needs repair: correcting a hierarchy's labels leaves its unassigned evidence missing.

A trial of aerobic exercise in older adults belongs under both a type of exercise and a population, and the methods differ in what they do with it. Dimensional taxonomies keep it in both places, one per dimension [21], and claim-level hierarchies let a claim sit in several categories [2]. Overlapping clusters keep it in both only at the lowest level, until the upper levels force a single parent [63], and aspect search assigns it to one aspect and one cluster at each node [65]. Keeping both memberships lets either section find the trial; forcing a single parent makes one of the two relationships harder to recover.

5.2 Survey outline generation

An outline gives each section a question to answer and the studies to answer it with. Fixing the outline before drafting lets a system write sections separately, but the plan can reflect only what the system knew when it made it. A plan made from paper titles captures little of what the studies contain, and even one made from their content cannot foresee which sections will overlap. Revising the plan while reading and writing addresses both gaps.

Prompted and staged outlines One system generates a title, section headings and an abstract from the topic and the titles of the supplied references, then selects references for each section and writes it [26]. The resulting outline recovers most reference headings, but the authors report unreliable citations and analysis when the writer lacks the references' content [26]. Another system grounds its plan in abstracts. It drafts candidate outlines from batches of retrieved abstracts, merges them, and uses each section's description to retrieve evidence for writing (AutoSurvey [5]). Both assign a scope to each section before drafting; the abstract-based plan also uses paper content to decide what those scopes should be.

Outlines learned from published surveys Planning can borrow an expert's organization or derive one from the current papers. Retrieved human-written outlines provide examples of how to divide a topic into main questions and narrower subquestions (SurveyForge [64]). Turning each heading into a retrieval query then connects that division to its evidence. Relevant survey examples improve the generated outline more than random examples do, and even random examples improve on using none [64]. A taxonomy provides a different starting point when the papers themselves should determine the groups.

Taxonomy-informed composition A taxonomy makes the plan depend on explicit paper groups. Holding the writer fixed, combining this plan with citation-graph retrieval slightly improves assessed structure over planning from retrieved abstracts (HiReview [63]). A tree alone leaves the comparisons within each group unspecified. Adding comparison tables supplies those dimensions before drafting: candidate outlines can be generated from both tree and tables, then one chosen for writing (MVSS [66]). Survey examples guide the shape of the account; taxonomies and tables connect that shape to the current studies and their comparable attributes.

Recurrent reading and outlining Merging outlines written from separate batches of papers can miss connections across batches. Reading in rounds lets those connections change a common plan instead. In IterSurvey [16], each batch contributes *paper cards*, records of contributions, methods and findings, that update the outline and suggest the next search terms. A similarity check ends exploration when the outline stabilizes. Before writing, the system reorganizes the outline for presentation and reassigns papers from earlier headings to the final sections. This procedure produces more coherent structures than merging separately drafted outlines [16]. Reassignment keeps the evidence gathered during exploration available after its headings change.

Replanning during writing Writing can also show that the plan is wrong. In the exercise review, a finished section on aerobic exercise may already discuss most of the trials that a later section on older adults was meant to cover. One system plans subsections with dependencies between them, writes them in stages, and stores the content and terms that each finished section introduces. At each stage boundary, a replanner uses that memory to merge, delete or reorder the sections not yet written (SurveyGen-I [22]). Keeping the plan fixed instead lowers the structure and consistency scores of the result [22]. Recurrent outlining revises the plan from what the system reads, and replanning revises it from what the system has written. Replanning can therefore remove redundancy that becomes visible only after a section has been drafted.

6 Citation-grounded synthesis

Synthesis turns selected evidence into supported claims and comparisons (Figure 4). The cited studies must support both the findings taken from each source and the relationships asserted across sources. Attribution ties each claim to the passage that supports it, which means choosing

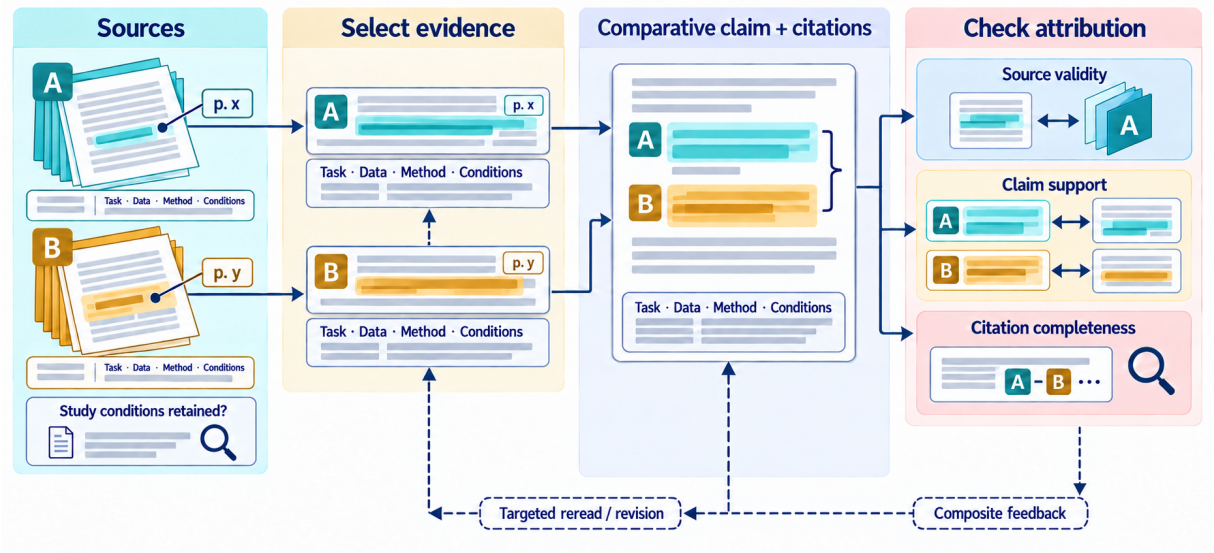


Figure 4: Comparative claims keep the conditions of their source findings and require three checks: (i) whether each cited source is a real study in the review (validity), (ii) whether it supports the claim (support) and (iii) whether every claim that needs a citation has one (citation completeness). A failed check sends the writer back to reread a source or revise the claim [10, 33–35, 67].

the right detail from the right source and then checking the result (§6.1). Comparison relates findings across studies, which means saying what the trials share and how they differ rather than listing them side by side, in prose or in tables (§6.2).

6.1 Source attribution

A citation can name a real paper and still fail to support its claim. A list of the selected sources is enough to check that a cited paper exists. Checking support needs the passage that states the detail the claim relies on. Attribution therefore depends on whether the writer, and any later checker, reads the right part of the source.

Selecting evidence for a citation The detail a claim needs determines which parts of a source the writer must read. In the walking trial's related work, a claim about an earlier trial's exercise dose needs its session frequency, which may appear only in the methods. Generating citation sentences from relevant source spans produces text closer to the authors' sentences than generation from abstracts does. Human judges also prefer the span-based sentences to those written from abstracts [67]. Summaries of a paper's central contribution are different: annotators prefer abstracts, choosing matching passages only when the citing sentence concerns another detail [68]. Abstracts therefore suit a broad account of the contribution, while more specific claims need the passages that establish their details.

Finding those passages also depends on the purpose of the citation. Predicting whether it supplies background, describes a method or compares findings improves reference ranking on four citation datasets (SciRGC [69]). The walking trial needs earlier doses for comparison, but only the choice of scale when justifying its outcome measure. Once a source is chosen, the preceding draft alone often fails to retrieve the relevant spans because it describes the work in different words; author-supplied keywords improve retrieval [67]. Citation intent helps choose the source, and a more explicit request for the required detail helps find the evidence within it.

Constraining sources during generation Assigning papers to sections gives the writer a bounded source set, and checking citation keys against that set catches invented references. In

DeepSurvey [35], each paragraph uses only its section's papers; detected citation errors enter a memory that guides regeneration. This writing module makes almost all citations resolvable, compared with about three in five without it, but raises claim support by only a few points. The two outcomes separate the checks: membership in the source set establishes that a paper exists, while support requires reading what it says.

Checking citations against sources A missing citation poses the reverse problem: given a passage, find the paper it came from. A recovery agent searches candidate papers' full texts and can also read more of the citing document (CiteGuard [34]). With the same model, it identifies the cited paper for about ten percentage points more passages than an earlier agent. Full-text search accounts for most of that gain, while additional citing context adds little once candidate full texts are available [34]. The comparison favors spending the extra reading on candidate sources before expanding the surrounding draft.

Support and citation completeness Support checking must compare the scope of a claim with the scope of the cited finding. A review that says exercise reduced symptoms in adults, citing a trial that enrolled only adults over sixty, makes a broader claim than its evidence. In long-form question answering, writing from retrieved passages still leaves gaps: on one dataset, even the best models give about half of their answers without full support from the passages they cite. Adding citations afterwards is worse: a model that answers without retrieval and then attaches citations is about as correct, but far fewer of its statements are supported by the passages it cites [70]. Retrieval before writing preserves more support than attaching citations afterwards; checking each claim against its passages addresses the gaps that remain.

A second property is *citation completeness*: every statement that needs a citation has one. Scientific question-answering systems address support and completeness together in a loop. One drafts from retrieved passages, generates feedback on what the draft lacks, retrieves more evidence and revises, and finally checks which statements need citations and inserts missing ones from the passages it already holds (OpenScholar [10]). Its answers are judged more correct than those of a general-purpose model and of PaperQA2, a concurrent agent that also writes from retrieved passages [10]. Applied to whole topic articles, PaperQA2 leaves about a quarter as many statements uncited as the corresponding Wikipedia articles, and fewer of its cited statements are unsupported [71]. A missing citation is cheap to repair when the evidence is already in hand, but an unsupported claim needs new evidence or a narrower statement.

6.2 Comparative synthesis

A comparison needs several studies in view at the same moment, each with its own conditions. A writer that summarizes one study at a time can group studies sensibly and still never relate them. Once a comparison exists, any later step that reshapes it, such as normalizing a table's cells or rewording a section, can separate a result from the condition it depends on.

From grouping to comparative statements An explanatory survey compares studies rather than only grouping them (§2.1). A paragraph about three exercise trials should explain how their doses, populations or comparators differ and what those differences mean for their results. One tool separates grouping the cited papers into themes from writing the paragraph, and lets the researcher edit both (ScholaCite [72]). In its related-work case study, fully generated sections cite papers together in the same sentence far less often than human-written sections, and writing with the tool brings this rate close to that of the human-written sections. The model groups papers reasonably well but rarely integrates them [72]. Joint citations identify places to inspect

for comparison. Researchers still need to check what the sentence says about the grouped studies, so grouping and comparative writing remain separate opportunities for intervention.

Incremental comparative writing Comparison requires a later paper to change the account of an earlier one. Two designs make that possible at different points. Incremental generation extracts key elements from each reference, adds references one at a time to a comparative summary, and votes among candidate continuations (ChatCite [33]). Human annotators prefer the complete design much more often than its variant without incremental generation; an LLM judge also ranks it first but attributes more of the gain to key-element extraction. A memory-based generator instead controls reading order: choosing citation-graph neighbors makes its sections discuss more references together than a retrieval baseline does (Select, Read, and Write [52]). The first design compares while updating a draft, and the second brings related sources together before drafting. Both give the writer more than an isolated summary of each reference.

How much text a section needs More retrieved text can broaden a section's coverage or lower its quality, depending on the retriever. Writing a survey section from several chunks instead of one uses more of the cited papers and covers more of a checklist derived from the reference section. With a different retriever, however, doubling the number of chunks lowers both scores and the judged quality. Sections written from papers found by a search engine, rather than from the supplied ones, score worst of all [73]. The same increase in input can help one retrieval setup and harm another, so chunk count alone does not establish that a section used more useful evidence.

Populating comparison tables A table compares studies cell by cell, so each cell needs its own evidence. One system turns each column into a question for each paper, asks for an answer with supporting excerpts from the full text, and then rewrites the values in each column so that they read consistently (ArxivDIGESTables [20]). With reference columns supplied, human judges accept only about a fifth of generated cells as complete matches. Captions and the text discussing the table increase partial matches, while complete matches change little [20]. Rewriting for consistency carries its own risk, since a cell that reads 'aerobic, 12 weeks' may drop the fact that only the supervised arm improved. Choosing the rows is a further decision. One benchmark tests it with candidate sets that include distractor papers and requests that describe the comparison without naming the columns (arXiv2Table [74]). Its method revises the paper selection and the schema together over batches of papers and replaces values that the sources do not support with 'N/A', which improves cell accuracy, paper selection and the consistency of comparisons between cells over a fixed two-stage pipeline [74]. Carrying a table into prose adds one more place for errors, which one system checks by aligning its concept tree, tables, outline and text and revising conflicts among them (MVSS [66]). Each transformation needs its own check. (i) Extraction needs source excerpts. (ii) Normalization must preserve qualifications. (iii) Row selection must retain relevant studies. (iv) Prose must agree with the table. Keeping source links through these transformations lets a later check recover the evidence behind a cell or sentence.

Coordinating and revising longer accounts Across a whole survey, a comparison that works locally can repeat an earlier one or change terms, calling the same trials 'supervised exercise' in one section and 'structured training' in another. Writing later sections with a memory of the content and terms already introduced keeps them consistent, and a final refinement pass matters most: removing it lowers the assessed quality more than removing any other component (SurveyGen-I [22]). Revision that changes the scope of a claim needs the sources again. When a review narrows 'exercise reduces symptoms' to 'supervised aerobic exercise reduces symptoms at twelve weeks', the writer must confirm that the cited trials report exactly that. Such revisions

can use notes on each paper's experiments, assumptions and limitations, stored together with the original passages (DeepSurvey [35]). Keeping the original passages available lets a writer recheck the revised scope against the trial rather than against an earlier paraphrase.

7 Protocol-guided evidence synthesis

A review protocol turns choices that a topic survey leaves open into explicit tests. The eligibility criteria fix S^*, the extraction plan fixes which conditions κ each record must keep, and an appraisal instrument judges how each study's results were produced (§7.1). A meta-analysis then computes its central claim from records whose conditions match (§7.2). Comparisons with researchers' decisions expose losses between these steps (Table 1). A study can be retrieved and still dropped, included and still misrecorded, or recorded correctly and pooled with an incompatible one. Statistical software already computes pooled effects and heterogeneity, but getting correct, comparable records into it still takes substantial manual work [18].

7.1 Review automation

A protocol turns screening, extraction and appraisal into tests that a system applies to each paper. Some tests can be answered from what a paper states directly, such as its topic or its citation details. Others turn on a single detail or on a judgment about how a study was run. Automated systems fail most often on the second kind. Their errors also carry forward, since a study wrongly dropped by an early test never reaches the later ones.

Inclusion decisions Screening applies the criteria to each candidate, and the hard cases match the topic but fail one condition. A trial that adds exercise to antidepressant treatment and compares the combination with antidepressants alone studies exercise for depression, yet it answers a different question from a trial that compares exercise with usual care. A model instruction-tuned on the objectives, criteria and abstracts of many Cochrane reviews classifies abstracts from unseen reviews more accurately than zero-shot ChatGPT, and as accurately as a classifier trained on most of the labels of each review (Bio-SIEVE [25]). The training labels need care. Many reviews record the final decision, made after reading the full text, so an 'exclude' label may rest on a detail that the abstract does not contain. On abstracts relabeled at the abstract stage, where such studies are kept for full-text review, zero-shot ChatGPT prompted to be lenient is as accurate as the trained models and recalls more of the eligible studies [25]. A screener for the abstract stage needs labels made at that stage, not final decisions.

Where the model's judgment enters the review matters as well. Consulting a model before a decision, discussing a candidate with it and asking it for an independent second judgment give researchers different ways to use advice (AiReview [78]). Letting the model exclude candidates instead makes its decision a gate on later reading. In an evaluation on epidemiological reviews, human screening of abstracts followed by AI screening of full texts recovers more of the included studies than AI at both stages, and AI that skips the abstract stage and screens every full text recovers nearly as many, at lower precision [37]. When abstract screening comes first, full-text screening sees only the studies it retains. Errors at the abstract stage therefore remove studies before the stronger evidence can be read.

Evidence retained after retrieval The included set must be checked after screening and writing, because retrieving an eligible study does not keep it in the review. Benchmarks for this loss pair review questions and criteria with known included studies. For example, MetaSyn [12] draws on published meta-analyses, matches their included studies to PubMed and builds candidate pools with near misses that fail one criterion, like the add-on trial above. Its retriever

Table 1: Protocol-guided review automation compared against human decisions and extractions.

Study	Task / record	Supplied evidence	Human reference / correction	Main finding	Domain
Bio-SIEVE [25]	Eligibility decision	Abstract; objectives; criteria	Evaluation: expert abstract labels	Matches per-review models on final labels	Biomedical
Protocol extraction [75]	Citation fields; finding excerpts	PDFs; protocol; extraction guidance	Evaluation: human extraction	Key findings and most planted errors missed	Health policy
ROBoto2 [36]	RoB2 signaling answers	Trial text; question guidance	Evaluation: expert labels; workflow: edits	Twice the experts' high-risk calls	Child health
MetaSyn [12]	Included-study list; synthesis report	PI/ECO; criteria; retrieved abstracts	Evaluation: checked review annotations	At least half of included studies lost after retrieval	Multiple fields
Domain fine-tuning [76]	Review-question answers	Selected PDFs; generated Q&A	Evaluation: expert grading	Better-supported answers; a wrong participant total	Learning analytics
AI co-scientist [77]	PICOS labels; medical answers	Titles/abstracts; full-text passages	Evaluation: expert labels and judgments	Retrieval helps questions with joint constraints	Dementia; NCDs
TrialMind [1]	Standardized trial outcomes	Study text; target fields/outcomes	Evaluation: checked labels; workflow: edits	More accurate numbers; subgroup pairing errors	Oncology
AutoForest [38]	Arm statistics; forest plots	Study PDFs; ICO	Workflow: expert/student edits	Edits fix residual extraction errors	Biomedical
Structural diagnosis [13]	Study-attributed association tuples	Full-text Markdown; fixed schema	Evaluation: expert reference records	Fields bound to the wrong role, method or paper	Five fields
AgentSLR [37]	Parameters; models; outbreaks	Abstracts; full-text Markdown; schemas	Evaluation: epidemiologist records	Field extraction is the bottleneck	Epidemiology

Evaluation denotes human reference labels or judgments; workflow denotes available correction or editing. RoB2: Risk of Bias 2; PI/ECO: population, intervention/exposure, comparator, outcome; ICO: intervention, comparator, outcome; PICOS: population, intervention, comparator, outcome, study design; NCD: noncommunicable disease.

places most included studies among the top candidates, yet no tested end-to-end system keeps more than about half in its report. Most losses occur after retrieval [12], so retrieval recall alone substantially overstates the evidence retained in the finished review.

Protocol-conditioned extraction An extraction plan says what each record must contain, and simple fields are much easier to fill than interpretive ones. Given the protocol and extraction guidance for a review of health policy and spatial planning, an LLM recovered every source's citation details but few of its key findings. These included the policies' implementation principles, strengths and weaknesses. The excerpts it returned were mostly relevant, but it missed most of those the human reviewers had extracted. Asked to check human-extracted data, it caught only 2 of 39 deliberately introduced errors [75]. Epidemiological reviews add a different problem, because one paper can report many parameters, models and outbreaks. Field-level F1 uses the calculation in Section 3.1 to compare extracted fields with expert records. Against epidemiologists' extractions, agents detect whether a paper contains a requested kind of data more reliably than

they extract its fields, and no tested model exceeds about two-thirds field-level F1 (AgentSLR [37]). Screening errors remove whole papers, whereas extraction errors lose observations inside the papers that remain.

Methodological appraisal *Appraisal* asks how far a study's results can be trusted. For randomized trials, the Risk of Bias 2 (RoB2) instrument poses *signaling questions* about how the trial was conducted, such as whether outcome assessors knew which participants exercised. Fixed rules turn the answers into risk-of-bias judgments about randomization, outcome measurement and other aspects of the trial. One system retrieves passages from the trial report for each signaling question, proposes an answer with a rationale, and applies the instrument's rules (ROBoto2 [36]). Run alone on child-health trials, it rated about twice as many trials at high risk of bias as the experts did, largely because the models often answered that the report gave no information. When experts used the system, they replaced its answer about two times in five and edited its rationale more than a quarter of the time [36]. A second tool answers the same signaling questions with an LLM and derives the judgments with the instrument's official rules, which is more accurate than prompting for the judgments directly (AutoForest [38]). Editable answers let experts correct the interpretation of the trial; explicit rules then propagate those corrections into consistent risk judgments.

7.2 Study-level synthesis

To combine included studies, systems either answer questions from their records or compute pooled effects. The first needs retrieval of relevant study attributes; the second also needs compatible outcomes and correctly paired group statistics.

Answering from the included studies Answering from included studies requires a way to reuse what the review has already read. Systems can train a model on those findings or retrieve from an explicit evidence store. Fine-tuning a small open model on question–answer pairs generated from the papers of one systematic review of learning-analytics dashboards makes its answers to reformulated questions better supported than those of the base model [76]. Expert reading still finds errors that aggregate scores hide, such as an answer that misstates the total number of participants across the reviewed studies, and every added or corrected study means training again [76]. Storing the study attributes explicitly avoids retraining. One system classifies each study's population, intervention, comparator, outcome and study design (PICOS), stores these attributes for retrieval and answers medical questions from them together with full-text passages (AI co-scientist [77]). A question such as 'which trials of supervised exercise in diagnosed adults compared it with usual care?' needs such attributes, whereas 'what is known about exercise and mood?' does not. In the system's own reviews, generation without retrieval remained competitive for broad summaries. Retrieval helped most with joint constraints and integration across studies [77], making explicit study attributes most useful when the question distinguishes among trials.

Updating a review that uses a fine-tuned model changes more than its study collection. Adding or correcting training examples changes how the model answers from that collection, so the training procedure becomes part of what another team needs to reproduce the review. A proposed extension of PRISMA, the reporting guideline for systematic reviews, asks authors using domain-fine-tuned models to document the fine-tuning data, training, evaluation, and ethical and legal aspects (PRISMA-DFLLM [79]). Repeating a literature search preserves the collection's currency (§3.2); reproducing answers from a fine-tuned model also requires a record of how that model was obtained.

Outcome standardization Pooling needs each value converted to a common scale without losing its outcome and population. One clinical pipeline takes the target fields, outcomes and cohorts, extracts the raw results, and writes a Python program that standardizes them (TrialMind [1]). On oncology reviews it extracts numerical results more accurately than a general-purpose LLM with human post-processing, and experts preferred its forest plots in most comparisons [1]. Its errors show the hazard that remains. A trial may report the number of patients treated early in the paper and the outcomes of one subgroup later, and the extractor then pairs the subgroup's outcomes with the whole cohort. In the running example, the same error would pool the improvement of a trial's older participants with the size of the full trial. Each number appears in the source, but the pairing is wrong.

Extraction linked to statistical computation The next step joins extraction to the computation itself. One tool extracts arm-level statistics, such as event counts and sample sizes or means and standard deviations, from study PDFs for a chosen intervention, comparator and outcome. It lets the user edit them and passes them to standard meta-analysis software, which computes the effects, intervals and heterogeneity and draws the forest plot (AutoForest [38]). More than 80% of automatically extracted fields are correct, and expert editing raises this to about 90%. The assisted workflow also takes far less time than manual preparation [38]. Showing the extracted numbers beside the plot they produce lets a reviewer check them at the point where an error would change the result.

Linking observations before aggregation Aggregation carries extraction errors into corpus-level claims. A diagnostic study of association studies in five fields asks models to extract tuples in which the variables, their roles, the method, the effect and the conditions all refer to the same observation in the same paper [13]. Models recognize the individual fields reasonably often but almost never bind them into correct tuples: they reverse independent and dependent variables, or attach a real pair of variables to the wrong method or effect. Counts computed across the corpus from these tuples are rarely right, even though the fields themselves are extracted with moderate accuracy [13]. These binding errors and the subgroup errors in outcome standardization show that statistical computation cannot repair an input that ties a value to the wrong study, population, outcome or comparison.

8 End-to-end survey-generation systems

Whole systems combine discovery, evidence representation, organization and writing (Table 2). We compare them by which earlier decisions a later step can change. Suppose a system writing the exercise review reaches the section on older adults and finds only two trials. A fixed pipeline writes a thin section. The possible repairs depend on the design. (i) A stored collection of paper attributes can be queried again. (ii) Replanning can merge the thin section into another. (iii) A critic with retrieval access can search for more trials. Architectures differ in what information they retain for later control (§8.1). Multi-agent coordination determines who can make those revisions, what evidence they share, what runs in parallel and when the work stops (§8.2).

8.1 System architectures

An intermediate artifact changes the workflow only if later decisions use it. We compare three sources of information for controlling revision (Figure 5): stored evidence for rewriting paragraphs, drafted sections for revising the remaining plan, and execution histories for choosing the next operation. Whether a revision may acquire new evidence is a separate choice (§8.2).

Table 2: Survey-generation systems compared by evidence storage, planning and revision.

System	Inputs	Evidence	Structure / control	Synthesis / revision
AutoSurvey [5]	Topic; paper database	Retrieved abstracts	Merged outlines; parallel drafting	Polishing with neighboring sections; best of several drafts
SurveyX [6]	Topic; keywords	Attribute trees summarizing each paper	Preparation then generation	Paragraphs rewritten with retrieved attributes
SurveyGen / QUAL-SG [42]	Topic; literature database	Abstracts; citation and venue data	Reranking by relevance, impact and diversity; outline then sections	Section by section
SurveyForge [64]	Topic; paper and outline databases	Papers stored for the survey and for each section	Outline learned from published surveys; parallel drafting	Parallel drafts merged and refined
IterSurvey [16]	Topic query; paper database	Paper cards (contributions, methods, findings)	Outline revised while reading	Review and refinement of the whole draft
SurveyGen-I [22]	Topic; description	Passages traced to cited sources; memory of written content	Staged writing; replanning of unwritten sections	Drafting and final refinement
LLM×Map-Reduce-V3 [80]	Topic; dialogue; optional documents	Grouped references; per-paper digests	Planner chooses the next tool	Writing from the refined outline
AutoSurvey2 [81]	Topic; paper database	Section analyses of abstracts	Fixed sequence of stages	Sections written from the analyses
MVSS [66]	Topic; paper database	Concept tree; comparison tables	Outline chosen from tree and tables	Tree, tables and text aligned
SciSage [27]	User query; research sources	Titles, abstracts and full texts	Outline from a chosen template; critics at three levels	Criticism can trigger retrieval and revision
ARISE [39]	Theme; optional references	Summaries stored under citation keys	Outline merges that keep each section's sources	Rewrites chosen by rubric scores

Inputs name the starting request and supplied resources; evidence names the material passed to planning or writing. SurveyGen denotes its complete retrieval-to-writing configuration (QUAL-SG).

Staged preparation and generation A staged system prepares its evidence before writing and reuses it while writing. One system acquires and filters papers and builds a forest of attribute trees that summarize each paper's content. It uses the forest to suggest the outline, and later turns each drafted paragraph into a query against the same forest to add detail or correct an attribution without searching again (SurveyX [6]). In the exercise review, rewriting a paragraph about supervised programs could pull each trial's dose and duration from the stored attributes. Another system writes the subsections of its merged outline (§5.2) in parallel, polishes each with its neighbors in view, and selects the best of several complete drafts (AutoSurvey [5]). The attribute store supports evidence-based correction during rewriting; neighboring drafts support consistency across sections.

A shared state can support reuse without changing the sequence of operations. One system stores the topic, outline, papers, analyses and generated sections together, while its preparation, analysis, synthesis and finalization stages run in a fixed order (AutoSurvey2 [81]). Later stages can reuse every stored product; the fixed outline still determines which sections they write.

Feedback before and during writing The outline methods of Section 5.2 also differ in when a revision takes effect. A system that settles its outline while exploring fixes its writing units before drafting begins, and then improves the draft with a reviewer–refiner loop over the

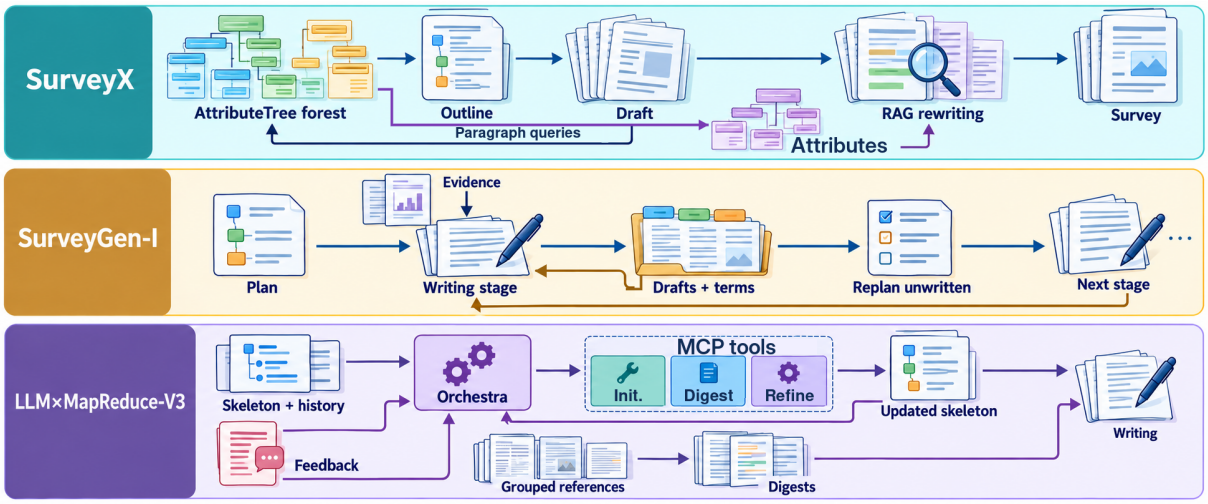


Figure 5: Stored evidence, drafted sections and execution histories enable different revisions (§8.1). A staged system (top) builds a store of paper attributes before writing and queries it again when rewriting paragraphs. A replanning system (middle) revises the unwritten sections after each writing stage, using the drafts and terms written so far. An adaptive system (bottom) has a planner (Orchestra) choose the next tool from the outline (the skeleton), execution history and user feedback. MCP, the Model Context Protocol, supplies the shared tool interface [6, 22, 80].

whole text (IterSurvey [16]). Replanning at stage boundaries instead passes each updated plan to retrieval for the remaining sections (SurveyGen-I [22]). Settling the outline first permits independent drafting; replanning instead lets later retrieval and writing respond to what earlier sections have already explained.

Adaptive tool orchestration A fixed workflow can share all of its intermediate products and still execute the same operations in the same order. Adaptive orchestration gives a planner control over that order. Tools for initializing an outline, digesting papers and refining structure expose a shared interface, and the planner chooses among them using intermediate outputs, earlier actions and tool descriptions (LLM×MapReduce-V3 [80]). Dialogue also lets a user clarify the research intent as the outline develops. Compared with fixed shared-state workflows and stage-boundary replanning, the unit of control is now the next tool action, which can respond to both the execution history and that clarification.

8.2 Multi-agent coordination

Splitting a survey among agents means deciding what each agent can see and what it can change. These two decisions set which errors the system can still repair. A reviser without search access cannot fill a gap in the evidence, and a writer that sees only its own section cannot keep its terms consistent with the others. Wider access also means more reading and more rounds of revision, so the same decisions set how long a survey takes.

Targeting revision Critics can repair a weak section either by finding missing evidence or by improving its use of existing sources. One system applies criticism at three levels. (i) Outline feedback reorganizes the plan. (ii) Section feedback can trigger retrieval and regeneration. (iii) Document-level reviewers with expert personas vote on the weakest sections before a final pass (SciSage [27]). Removing reflection lowers LLM ratings most for critical analysis and structure [27]. Because its critic can trigger retrieval, such a system can repair the thin section on older adults by finding more trials. Another system confines revision to evidence it already holds.

Several reviewers score the draft against a common rubric, their feedback becomes a revision plan, and each revised section may use only the summaries stored under its existing citation keys (ARISE [39]). When four experts rated five surveys before and after this process, their rubric totals rose from about 70 to about 84 out of 100 [39]. Rewriting within a fixed collection repairs how a section uses known studies; repairing an omission from that collection requires retrieval access.

Shared and section-specific memory Memory policies decide how much shared evidence each agent sees. In section-specific memory, a writer sees only its assigned evidence. Shared content memory also exposes earlier sections and their terminology (§6.2); an execution history exposes earlier tool actions (§8.1). The choice determines which kinds of earlier decisions another agent can use. A shared pool can also carry interpretation from one role to the next. In a system that writes humanities essays from classical primary sources, agents annotate, compare, check provenance and sample evidence in a shared pool before binding each claim to its passages (SPIRE [82]). It recovers more of the relevant primary-source evidence than text retrieval and the other baselines it was compared with [82]. The labels with which agents compare passages stay attached to those passages and their provenance as the evidence moves toward the essay, so each role's interpretation can be inspected by the next.

Concurrency and integration Writing sections in parallel moves coordination to what the writers receive before drafting and to the reconciliation afterwards. Parallel writers can share a common outline and be reconciled afterwards by polishing or by merging their drafts [5, 64]. A graph of prerequisites offers a middle course: the independent subsections of a stage are written in parallel, and a dependent stage waits until its prerequisites are finished [22]. Web-report systems face the same problem at a larger scale. One links an evolving outline to a bank of evidence, and its writer retrieves only the evidence for the current section and drops those sources from its context after drafting it. This section-by-section writer earns better citation accuracy and insight scores than a writer given the whole evidence bank at once (WebWeaver [83]). Another organizes research as a tree of subtasks that expand and run asynchronously and prunes redundant branches as findings arrive, reaching report quality comparable to its base system up to five times faster (ParallelResearch [84]). The two designs place coordination at different boundaries: the writer receives evidence for one section, while the research scheduler manages dependencies and redundant work. Parallel execution still needs an explicit rule for which evidence each finished section uses.

Stopping and revision budgets Stopping rules bound either evidence acquisition or revision. Acquisition can end after a minimum number of papers once the outline looks complete, with a maximum paper count as a fallback [16]. Revision instead stops when feedback is exhausted, rubric scores pass a threshold, or an attempt limit is reached [27, 39]. The first kind of limit determines how much evidence reaches writing; the second determines how much work a draft receives. Systems also differ widely in what a survey costs. Under one benchmark's configurations, one system takes roughly three quarters of an hour per survey while two others take 5 to 14 minutes [85]. Refinement can take a large share of that time: in the rubric-guided system, evaluation and rewriting take 30 to 40 percent of the total processing time, the price of the expert-rated gains reported above [39].

Comparisons between whole systems mix all of these choices. A system that reads more papers, passes more evidence to its writers, or judges and regenerates its draft more often can score higher for reasons other than its architecture, and a draft revised against one model's judgments may simply suit that judge (§9.3). Whole-system scores are therefore most useful alongside controlled component comparisons and reported source and revision budgets.

9 Evaluation and benchmarks

Evaluation checks whether a survey finds the required studies, organizes their findings and supports its claims. These properties can vary independently: a system can recover the target set S^* yet misassign its studies or cite them for claims they do not support. Benchmarks differ in what they give the system, from a fixed set of sources to a search engine (§9.1; Table 3). They also differ in what they compare the output against, from a published survey to a checklist of facts (§9.2), and in who scores it: a metric, an LLM judge or an expert (§9.3).

9.1 Benchmark design

What a benchmark gives the system decides which failures its scores can locate. When the sources are supplied, a low score points to how the system read and wrote. When the system must also search, a low score can come from either step. Supplying more makes a failure easier to locate, but leaves more of a real review's work untested.

Supplied-source generation These benchmarks give the system its sources and ask it to select, organize and write from them. They cover chapters from cited abstracts [24], sections from cited full texts [73], related work from a target paper and its references [40], and comparison tables from fixed papers [20]. For the exercise review, the trials are supplied; success means preserving their findings and choosing details that answer the requested question. Reference texts or tables specify one expert solution, while human or model judges assess qualities such as faithfulness and usefulness.

Text-overlap measures compare a generated section with a human-written reference section. *ROUGE* uses the precision, recall or F1 calculations of Section 3.1 on shared word sequences or longest common subsequences, depending on the variant. *BERTScore* instead matches each token to the most similar contextual token embedding in the other text. It averages those similarities in each direction as soft precision and recall, then combines them with F1. G-Eval instead asks an LLM to assign a rubric score. For tables, *schema recall* is the fraction of reference columns matched by meaning to generated columns. *Value accuracy* is the fraction of cell pairs accepted as matching by the chosen scorer; human evaluation distinguishes complete and partial matches. Supplying the reference columns isolates cell filling from schema-alignment errors [20].

Results distinguish evidence access from evidence use. Supplying full texts instead of abstracts can lower the fraction of supported statements and increase attribution errors [40]. Table values also match human tables only partly even when the correct columns are supplied [20]. Giving a system the right papers removes discovery errors, but leaves selection within those papers and faithful extraction as substantial problems.

Writing from a reference survey's sources A second family fixes both the bibliography and explicit demands on the review. The system receives the full texts cited by a reference survey, a topic request and structural constraints. Success means consolidating the supplied evidence into an account that satisfies those demands. DeepSynth-Eval [86] checks facts and constraints derived from the reference, penalizing incorrect mentions as well as omissions.[1] With the same summarized sources, planning, chapter-by-chapter writing and polishing cover more of the checklist than single-pass generation [86]. Staging therefore improves consolidation even when discovery is held fixed.

[1] Each checklist item receives $+1$ for a correct mention, 0 for omission and -1 for an incorrect mention. Each group's sum is divided by a specified saturation threshold and capped at one. Group scores are then combined by weighted averages for general content, structural constraints and the overall score [86].

Table 3: Survey-generation benchmarks grouped by supplied sources and discovery requirements.

Resource	Task / output	Given evidence / search	Reference / evaluation unit	Main measures
Supplied-source generation				
SciReviewGen [24]	Review chapter	Fixed cited abstracts; review and chapter titles	Human chapter	ROUGE; human quality judgments
SurveySum [73]	Survey section	Fixed cited full texts; online-search variant	Human section and cited-paper set	Reference F1; G-Eval; checklist coverage
OARelatedWork [40]	Related-work section	Fixed target paper and cited papers; abstract/full-text variants	Human section, citations and statements	ROUGE/BERTScore; reference F1; factuality labels
ArxivDIGES-Tables [20]	Comparison table	Fixed papers; abstracts for schema, full text for cells	Human columns and aligned cells	Schema recall; value accuracy
Writing from a reference survey's sources				
DeepSynth-Eval [86]	Constrained survey	Reference bibliography in full text; topic and structural constraints	Reference-derived fact and constraint checklists	Weighted checklist scores; error penalties
Discovery and generation				
SurGE [87]	Scientific survey	Topic; search supplied arXiv metadata/abstract corpus	Human survey, headings and references	Reference recall; citation support; soft heading recall
ReportBench [14]	Academic report	Prompt; online search with publication cutoff	Survey bibliography; cited/uncited statements	Reference precision/recall; citation match
SurveyLens [88]	Scientific survey	Topic; system-specific retrieval	Human survey entries; discipline rubrics	Rubric scores; semantic alignment
TaxoBench [89]	Paper taxonomy	Topic with search; or fixed expert paper set	Expert paper set, assignments and hierarchy	Retrieval F1; ARI/V-measure; tree/path alignment
DeepSurvey-Bench [85]	Scientific survey	Topic; each system's retrieval protocol	Annotated human surveys; seven academic-value criteria	Academic-value rubric ratings
SurveyBench [90]	Scientific survey	Topic; system-specific retrieval	Human surveys; topic-specific questions	Rubric ratings; quiz-answer scores
ResearchArena [91]	Paper discovery, ranking; organization	Topic; search supplied full-text corpus	Survey citations, influence labels; mind-maps	Precision/recall@K; nDCG; MRR
DeepScholar-Bench [92]	Related-work synthesis	Target abstract; arXiv API search with date cutoff	Human section, nuggets and cited sources	Nugget/reference coverage; citation precision
General web reports				
DeepResearch Bench [93]	General research report	Expert task; open web search	Model-generated report; statement–URL pairs	Adaptive relative quality (RACE); citation support (FACT)

The measures listed are defined in Sections 9.1 and 9.2; rubric calibration is discussed in Section 9.3. ARI: adjusted Rand index. The discovery groups distinguish controlled source access from each system's own retrieval; general web reports use a model-generated reference.

Discovery and generation These benchmarks require finding sources as well as writing or organizing them. Controlled-access versions provide arXiv metadata and abstracts (SurGE [87]), an offline full-text corpus (ResearchArena [91]), the arXiv search interface for related-work generation (DeepScholar-Bench [92]), or web search up to a publication cutoff (ReportBench [14]). They compare outputs with published surveys, bibliographies or extracted facts. Others retain each system's own retrieval and compare complete systems on common topics, source-access

differences included [85, 88]. TaxoBench [89] isolates organization by also supplying the expert paper set. These settings distinguish failures to find evidence from failures to use it.

Retrieval rankings apply precision and recall at a cutoff K to the first K papers (§3.1). When labels mark which papers are influential, *normalized discounted cumulative gain* (nDCG) divides discounted gains by those of the ideal ordering. At cutoff K, each positive label at rank i contributes $1/\log_2(i+1)$; summing those contributions and normalizing rewards influential papers placed early. *Mean reciprocal rank* (MRR) averages the reciprocal rank of the first positively labeled paper across requests [91]. Bibliography overlap, structural alignment and supported content then assess the final output (§9.2). Ranking a useful paper highly and retaining it in a supported final claim are separate successes.

General web-report benchmarks broaden the source setting beyond scholarly literature. DeepResearch Bench [93] supplies expert tasks, uses a commercial agent's report as the reference, and evaluates both relative report quality (RACE) and source support (FACT).[2] A model-generated reference measures agreement with that agent's account, whereas published-survey references measure agreement with a scholarly author's account. Source access, reference construction and scoring rules must accompany any comparison of benchmark scores.

9.2 Evaluation criteria

Every criterion compares the survey with something taken as correct. For some criteria, the reference is the evidence itself, such as the passage a citation names. For others, it records one author's choices, such as a published bibliography or outline. A survey that departs from such a reference may have made a different reasonable choice rather than an error, so these scores measure agreement more than correctness.

Reference coverage and relevance The study set is usually scored against a published survey's bibliography, which stands in for S^*. Reference metrics apply the calculations of Section 3.1 to cited papers. *Reference recall* is the fraction of the reference bibliography cited by the generated survey. *Reference precision* is the fraction of its cited papers found in that bibliography. F1 combines these two fractions. On a survey benchmark that supplies a searchable collection, the benchmark's own retriever finds about two thirds of the reference papers among its top thousand candidates, yet the surveys that systems write cite fewer than one in ten of them [87]. The best agent on the taxonomy benchmark retrieves about a fifth of the papers the experts cited [89], and two commercial deep-research agents each recover about 3% of the reference bibliography, although one cites three times as many papers as the other [14]. Across these settings, most reference papers never reach the output. Retrieval and post-retrieval selection both contribute to that loss.

A bibliography is an incomplete specification of relevance, however. When an LLM judge rates topical relevance, only about half of the citations in human-written reference lists count as relevant, whereas most of the papers chosen by the strongest retrieval systems do, because many human citations point to foundational methods, classic algorithms or software rather than to the topic itself [94]. A reference review of exercise trials might cite the statistics paper behind its pooling method: a generated review that omits it loses recall, while a relevant trial that the reference missed earns nothing. Comparison tables raise the same problem, since columns missing from a human-written table can still be useful (§4.2). Reference overlap measures agreement with one author's selection, and topical relevance measures whether each paper bears on the topic, so the two need to be reported separately.

[2] RACE weights criterion scores within dimensions and dimension scores within each report, then divides the candidate's weighted score by the sum of candidate and reference scores. FACT deduplicates statement–URL pairs and judges each against its web page. *Citation accuracy* is the supported fraction of these pairs; *effective citations* are their supported count, averaged across tasks [93].

Assignments and hierarchical organization Organization benchmarks receive generated taxonomies and expert references, either over supplied papers or over independently retrieved sets. Success requires both assigning papers to appropriate categories and placing those categories at suitable levels. The *adjusted Rand index* measures agreement on whether pairs of papers belong together, subtracts expected agreement by chance and normalizes by the remaining possible agreement. *V-measure* combines homogeneity, which penalizes mixing reference categories in one generated group, and completeness, which penalizes splitting a reference category across groups, by their harmonic mean [89]. These compare partitions, while hierarchy measures compare category trees and the paths leading to papers.[3]

Computing assignment scores on papers shared with the reference isolates organization from retrieval loss. With the experts' exact paper sets, generated taxonomies remain shallower than the references; forcing greater depth can fragment their categories [89]. Raw path scores sit close to an unorganized baseline, but are also sensitive to depth and category vocabulary. Together, shallow structures, fragmented categories and weak path alignment identify different departures from expert organization that a single hierarchy score would conceal.

Outline metrics compare generated and reference headings. In SurGE, *soft heading recall* measures semantic overlap relative to the reference headings, down-weighting near-duplicate headings [87].[4] One-to-one semantic alignment instead prevents repeated matching to the same reference entry, with a further penalty for redundancy (SurveyLens [88]). An outline organized by comparator rather than exercise type may still serve its reader while matching the reference poorly. Heading alignment is useful for checking reference coverage; judging an alternative organization also requires checking what its sections explain.

Citation support and completeness Citation evaluation receives generated statements, their citations and the cited source text. Success requires both supported statements and citations wherever support is needed. In long-form question answering, *citation recall* is the fraction of generated statements fully supported by their cited passages; uncited statements count as unsupported. *Citation precision* is the fraction of citations that contribute to that support [70]. These use statements and citation contributions rather than the paper overlap counted by reference metrics. An academic-report benchmark reports the *citation match rate*, the fraction of cited statements supported by their named sources. It verifies uncited statements separately by having several web-connected models vote on them (ReportBench [14]). About a fifth of the cited statements of one commercial deep-research agent are not supported by the sources they cite [14]. A bibliography can therefore overlap the reference while individual claims overstate what the cited papers show. Source checks detect this failure directly.

What the checker can see also changes what support means. One survey benchmark checks cited abstracts at the document, section and sentence level and gives any reference in the ground-truth bibliography full credit for relevance at the document level [87]. A related-work benchmark accepts support from nearby cited sentences and treats the target paper as implicitly cited when scoring how many reference claims the text covers [92]. These rules decide which statements count as attributed. For a claim about an effect limited to the supervised arm, the checker needs the passage reporting that qualification. Support scores should specify the source text and citation context the checker was allowed to read.

[3] TaxoBench's semantic tree edit distance minimizes the cost of inserting, deleting and renaming category nodes, with renaming costs derived from label similarity; its normalized form divides by the total category count of both trees. Semantic path similarity aligns ancestor labels in order for each matched paper, penalizes label differences and extra ancestors, maps each cost J to $1/(1+J)$, and averages across matched papers [89].

[4] Each heading contributes the reciprocal of its summed embedding similarities with headings in the same set. Summing these contributions gives the set's soft cardinality. Soft overlap is the sum of the two sets' cardinalities minus their union's cardinality; soft heading recall divides that overlap by the reference set's cardinality [87].

Synthesis and informational depth Content evaluations ask whether a reader can learn the topic's findings and comparisons from the generated survey. They use questions, fact checklists or rubric dimensions to specify success. Joint-citation graphs identify where studies are discussed together [72]; questions and checklists go further by testing which information that discussion preserves. For the exercise review, a question might ask which comparator most supervised-exercise trials used, a detail that a fluent overview can omit.

In SurveyBench [90], topic-specific quiz questions come from human surveys and are answered by retrieving from each candidate survey. The *quiz-answer score* is an LLM's quality rating from zero to ten for an answer drawn from the candidate survey; it is set to zero if the survey lacks supporting evidence. *Nugget coverage* is the fraction of short reference-derived facts recovered in the output [92]. *Checklist coverage* is the fraction of reference-derived facts or requirements satisfied in the output; error penalties distinguish incorrect mentions from omissions (§9.1). Rubric judgments instead rate dimensions such as in-depth comparison, critical analysis, research gaps and future directions (DeepSurvey-Bench [85]). Discipline-specific rubrics adapt the criteria and their weights to the field (SurveyLens [88]). These methods distinguish recoverable information from a judge's assessment of how well it is presented.

The distinction changes the verdict on generated surveys. On SurveyBench's computer-science topics, a commercial deep-research agent receives favorable content ratings, yet the best system scores about 3 on a quiz scale anchored at 10 for the human reference survey [90]. Academic-value ratings likewise place structure and presentation above comparison and critical analysis [85]. Discipline-specific rubrics find dedicated survey generators strongest on outlines and deep-research agents strongest on content [88]. Across these evaluations, favorable structure and presentation ratings coexist with missing information and weak comparisons. A high writing-quality score therefore leaves the survey's informational depth unestablished.

9.3 Evaluator validity

An evaluator's errors become errors in the benchmark's results. An evaluator that agrees with experts on one property may not agree on another: a judge validated on outlines may still misjudge content or support. A score can also reflect conditions other than the method being compared, such as whether the system saw the reference survey or whether a commercial service has changed since the run.

Rubrics and expert calibration Evaluator calibration compares automatic scores with expert judgments of the same survey property. Success means reproducing expert distinctions rather than merely producing stable scores. *Pairwise agreement* is the fraction of text pairs on which the judge and experts prefer the same output. Breaking related-work evaluation into local constraints on citation faithfulness, positioning and presentation, each illustrated with contrasting examples, yields a judge whose choices between two texts match experts' more often than general-purpose judge models do (GREP [95]). Agreement varies by dimension. Rubrics fitted for each discipline turn a slightly negative correlation between a holistic judge's outline rankings and experts' rankings into a strong positive one on computer-science surveys, but the correlation for content stays weak [88]. A model judge can still favor its own prose. On Wikipedia-style surveys of natural-language-processing concepts, an LLM rated its own generated texts above human-written reference articles, while its ratings correlated weakly with experts' on each dimension, factuality included [96]. A judge calibrated for outline quality cannot be assumed reliable for content or factuality; each dimension needs its own expert comparison.

Errors of support checkers Support checkers receive a claim and source text and decide whether the text establishes it. Their *accuracy* is the fraction of decisions agreeing with human

support labels, which requires comparable access to the source. A benchmark combining attribution datasets finds that models miss details and numbers; in some cases annotators had read more context than the snippet supplied to the model. Fine-tuning improves these decisions but leaves substantial errors [97]. Judges based on *natural language inference*, which decide whether one text entails another, also struggle with partial support [70]. An effect reported only for the supervised arm can leave a claim about exercise in general only partly supported. A binary verdict can conceal the missing qualification. Support classification needs its own validation, because calibration on writing quality says nothing about it.

Reference dependence and reproducibility Scoring against references makes the target specific at the cost of tying it to one author's choices. Reference-based judges make their scoring target explicit: one anchors an LLM judge to human-written reference surveys [98], and another rewards similarity to human outlines, content and bibliographies, with preference comparisons made by an LLM rather than by experts [99]. References can also leak into the output. A benchmark that audits whether systems retrieved the human survey they are compared with finds that nearly half of the deep-research agents' outputs, and most of one agent's, draw substantially on it, although excluding these outputs leaves the ranking of system types unchanged [88]. In the running example, copying the reference review's bibliography would earn overlap without independently finding the trials. Rebuilding tasks from recent papers reduces opportunities to reuse familiar reference material (DeepScholar-Bench [92]). Leakage audits and refreshed tasks test whether high overlap reflects independent synthesis.

Reproducing a comparison also requires the same sources, models, prompts and retrieval conditions, which commercial systems do not hold fixed. One benchmark collected its commercial agents' outputs through their web interfaces [14], and another evaluates commercial systems whose configurations change over time [88]. A score collected later may therefore reflect different sources or model behavior as well as a different method. Saving the retrieved evidence, outputs, prompts and model identifiers makes the comparison inspectable even when the original service can no longer reproduce the run.

10 Researcher oversight

Researcher oversight connects automated decisions to the purpose and evidence of a review. We distinguish control over studies and their interpretation, personalization for an intended reader, and verification after rewriting. These require different evaluations: usability ratings describe the experience of using a tool, while controlled task comparisons measure changes in time and accuracy.

Editable research decisions Control over intermediate artifacts lets researchers change the basis of a survey before they edit its prose. One system lets researchers change the references, the assignment of papers to categories, the outline and the generated text, and lets them state the criterion by which papers are grouped (InteractiveSurvey [100]). A researcher who wants the exercise trials organized by the population studied rather than by type of exercise can say so, and move trials between the resulting groups. Thirty-four researchers gave the system high usability ratings [100]. Another tool places control earlier, in searching, following citations and filtering (Arc [101]). Comparing successive versions of a search shows which papers a change to the query added or removed, and the tool's suggested relevance labels come with justifications. In its comparison with conventional workflows, following citations took under a minute with the tool against about eleven minutes by hand. Filtering 15 papers took about nine minutes with the tool against about eleven by hand, but seven of those nine minutes went to checking the suggestions [101]. The suggestions were more accurate than the participants' own unassisted

labels, yet every participant checked them thoroughly, and one found a correct label whose justification did not match expert reasoning [101]. Assistance saved time by turning screening into verification, not by removing the need for it.

Linking extracted evidence to its sources extends this control from organizing papers to interpreting them. One tool, which combines researcher actions with AI suggestions, lets researchers pull evidence snippets onto a canvas, arrange and connect them, and ask for suggested themes, while each snippet keeps its link to the source document (ScholarMate [102]). The evidence for this design is a two-student usability pilot on familiar papers [102]. Its distinctive control is at the passage level: a researcher can group the passages that report adherence to exercise programs into a theme, inspect the snippets behind a suggested theme, and follow each one back to its trial.

Personalization and authorial intent Control also concerns whom the report is for. One system infers a profile of the researcher from papers they supply, translates it into editable actions for each question, and highlights the passages of the report that it personalized (MyScholarQA [103]). The profile persists across questions, so a researcher can adjust the actions for one question without redefining their background. The system does well on an offline personalization benchmark, but in think-aloud sessions with researchers who regularly use deep-research tools, participants found profiles that overstated their expertise and actions that narrowed their questions too much. They still valued being able to see and edit the personalization [103]. Asked to predict which outputs would dissatisfy the users, LLM judges did no better than always predicting the majority outcome [103]. In the running example, a profile built from a researcher's clinical papers may cast them as a clinician choosing treatments when they are a trialist looking for gaps in study design, and a report that faithfully follows that profile still answers the wrong question.

Keeping a researcher's intent intact is also work during rewriting. In interviews, eight researchers who use generative AI for literature reviews described fragmented workflows that mix generation, domain tools and manual checking of sources [15]. They reported *semantic drift*: iterative rewriting weakened technical nuance, moved citations to passages they did not support, or changed the intended argument. Checking sources often meant leaving the drafting tool to consult the original papers, and splitting the work across chat sessions made it hard to trace which statements had been verified [15]. A rewrite that turns 'reduced symptoms in adults with mild to moderate depression' into 'reduces depression' keeps its citation but no longer says what the trial found. That change invalidates the earlier check even though the researcher has already approved the outline and the citation remains in place.

Verification across revisions Reusing a check after an edit requires a record of the claim and evidence that were checked. Existing controls expose parts of that history: linked snippets identify the material behind a theme [102], while search versions show changes to the collection [101]. Neither study tests whether those records let a researcher reuse verification after rewriting. Inclusion criteria can change as well: participants in the search-tool study described renegotiating them as unfamiliar studies appeared, a change the tool did not support [101]. The interview authors therefore call for persistent memory of verified statements and citations across sessions [15]. Such a record would need to identify both changed claims and changed inclusion decisions, so that a researcher repeats the checks affected by an edit rather than checking the whole review again.

Across these short sessions and bounded tasks, the clearest gains come from editable decisions and faster access to evidence. Assistance can reduce search and screening time, while researchers still spend substantial effort checking suggestions and revised claims. The supported role is therefore assistance with inspectable intermediate decisions; evaluating that role over a completed

review is the next step (§11).

11 Research directions

Improving a single review stage leaves errors that cross stage boundaries. We organize the open problems around those dependencies: search coverage, revised organization, comparisons under different conditions, attribution through edits, useful evaluation and controlled updating. The dependencies concern both maintaining support through changes and testing whether that support serves the intended reader.

Coverage and stopping under changing literature Search systems need to estimate how much of the target set S^* remains undiscovered. Learned search policies and refinement from judgments of relevance improve what is found [19, 30], and repeated searches keep a collection current [47], yet agents asked to collect every paper that meets detailed constraints return sets that overlap little with the target [46]. Outline stability can stop acquisition (§8.2), yet an outline can stabilize around the lines of work the search reached while omitting another entirely (§3.1). Coverage estimates should account for how the seed papers, the citation expansion and the relevance judgments depend on one another, and distinguish evidence that has gone out of date from evidence that was never found. Such estimates would let a review state how far its claims about what the literature lacks can be trusted.

Taxonomy and outline revision Systems now revise their organization from paper assignments, further reading or drafted prose [16, 21, 22], yet plausible categories still leave many studies unassigned [2]. A new trial can fit an existing category, expose a missing distinction, or challenge the organizing criterion itself. A trial that adds exercise to antidepressants might fit under comparators, call for a new category of combined treatment, or suggest that the whole review should be organized by comparator rather than by type of exercise. Each case calls for a different change to memberships, category definitions and the outline, and moving a trial can change what an earlier comparison appears to show. Systems should record which changes come from new evidence and which from a new way of explaining it, and keep the sources behind both the old and the new organization.

Comparative claims under differing conditions Tables with linked sources let readers inspect comparisons [20], but extractors still tie values to the wrong population, method or study [1, 13]. Two trials that seem to disagree may answer different questions, one measuring symptoms at twelve weeks in diagnosed adults and the other remission at six months in a mixed sample. A record that carries its study, population, outcome, comparison and time point, and marks conditions that are partly reported or named differently, would let synthesis tell a disagreement from a difference in setting. Multimodal sources make such records harder to build (§4.1). Curation that links each value back to its text, table or figure helps [54], and a new benchmark tests agents on questions that require integrating such content across many papers [104]. The next test is whether a system preserves those source links and conditions through the cross-paper comparisons that use them.

Attribution through long-document revision Constrained drafting, citation checking against source content and memory of what has been written reduce invalid citations and inconsistent prose [22, 34, 35], but none of them decides which earlier checks survive an edit. Narrowing a claim, merging two comparisons or changing a qualification can invalidate a verification while the citation stays attached [15]. A position paper argues for provenance at the

level of individual claims, linking each statement to the evidence behind it [105]. A practical test would measure whether those links remain reliable through heavy rewriting and how much work their maintenance requires. Editing systems need to track dependencies among conclusions as well as between conclusions and sources, because a later conclusion that paraphrases an earlier comparison without repeating its citations must change when that comparison does.

Assessment validity and research usefulness Benchmarks now separate discovery and organization from informational depth [89, 90], and evaluators prove reliable on some dimensions and not on others [88]. The missing connection is between benchmark gains and a survey's usefulness to a researcher with a particular purpose. Reference alignment rewards agreement with one established selection and organization, while a useful alternative might organize the same trials differently. Support scores leave open whether a reader understands a methodological disagreement or makes a better decision. Offline gains in personalization did not predict what real users found wrong [103]. The next step is to connect component measures to what readers learn and how much verification they do, which first requires saying whose needs an assessment serves.

Economical, reproducible updating with researcher control Persistent evidence stores, targeted rewriting and stored screening decisions make reuse possible [6, 39, 47], but a new trial can change a category, a pooled estimate and every conclusion built on them while most records stay valid. An update should find the decisions that need reconsidering without repeating every reading and check, because an update that silently invalidates earlier checks hands that work to the author. One untested proposal treats task specifications and model instructions as versioned artifacts that a program generates and optimizes [106]. This would make configuration changes explicit alongside changes to the evidence. Repeatability leaves a harder problem: researchers need to tell a newly supported conclusion from a changed model preference, and to carry the question they asked, the studies they approved, the extractions they checked and the conclusions they verified through each update. Updates that track these dependencies, and that preserve researchers' corrections as models, retrieval services and collections change, address both the cost of maintenance and the loss of control.

12 Conclusion

Automated survey generation now combines finding, reading, organizing and writing about the literature on a topic. Some systems also revise their plans as they read and write. Fluent drafts can still omit eligible studies, separate numbers from their populations and time points, leave studies unassigned, and attach real citations to unsupported claims. The checks that address these failures preserve different connections. Records carry the conditions of a finding, taxonomy checks inspect assignments as well as labels, citation checks compare claims with supporting passages, and revision returns to those sources when wording changes.

Each stage needs its own measurement, and current benchmarks show how far presentation runs ahead of supported content. The central challenge is to preserve support and researchers' corrections as both literature and text change. A new trial can alter a comparison and its conclusion; a rewritten sentence can undo an earlier check. Progress depends on estimating what searches miss, retaining findings with their conditions, tracking which checks an edit invalidates, and leaving researchers in control of the question and its interpretation.

References

[1] Zifeng Wang, Lang Cao, Benjamin Danek, Qiao Jin, Zhiyong Lu, and Jimeng Sun. Accelerating Clinical Evidence Synthesis with Large Language Models. arXiv preprint arXiv:2406.17755, 2024.

[2] Chao-Chun Hsu, Erin Bransom, Jenna Sparks, Bailey Kuehl, Chenhao Tan, David Wadden, Lucy Lu Wang, and Aakanksha Naik. CHIME: LLM-Assisted Hierarchical Organization of Scientific Studies for Literature Review Support. arXiv preprint arXiv:2407.16148, 2024.

[3] Xiangci Li and Jessica Ouyang. Automatic Related Work Generation: A Meta Study. arXiv preprint arXiv:2201.01880, 2022.

[4] Xiangci Li and Jessica Ouyang. Related Work and Citation Text Generation: A Survey. Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, 2024.

[5] Yidong Wang, Qi Guo, Wenjin Yao, et al. AutoSurvey: Large Language Models Can Automatically Write Surveys. arXiv preprint arXiv:2406.10252, 2024.

[6] Xun Liang, Jiawei Yang, Yezhaohui Wang, et al. SurveyX: Academic Survey Automation via Large Language Models. arXiv preprint arXiv:2502.14776, 2025.

[7] José de la Torre-López, Aurora Ramírez, and José Raúl Romero. Artificial intelligence to automate the systematic review of scientific literature. Computing, 2023.

[8] Francisco Bolaños, Angelo Salatino, Francesco Osborne, and Enrico Motta. Artificial Intelligence for Literature Reviews: Opportunities and Challenges. Artificial Intelligence Review, 2024.

[9] Wenlin Zhang, Xiaopeng Li, Yingyi Zhang, Pengyue Jia, Yichao Wang, Huifeng Guo, Yong Liu, and Xiangyu Zhao. Deep Research: A Survey of Autonomous Research Agents. arXiv preprint arXiv:2508.12752, 2025.

[10] Akari Asai, Jacqueline He, Rulin Shao, et al. OpenScholar: Synthesizing Scientific Literature with Retrieval-Augmented LMs. arXiv preprint arXiv:2411.14199, 2024.

[11] Anirudh Ajith, Mengzhou Xia, Alexis Chevalier, Tanya Goyal, Danqi Chen, and Tianyu Gao. LitSearch: A Retrieval Benchmark for Scientific Literature Search. arXiv preprint arXiv:2407.18940, 2024.

[12] Anzhe Xie, Weihang Su, Yujia Zhou, Yiqun Liu, and Qingyao Ai. Benchmarking LLM Agents on Meta-Analysis Articles from Nature Portfolio. arXiv preprint arXiv:2606.17041, 2026.

[13] Zhiyin Tan and Jennifer D'Souza. Diagnosing Structural Failures in LLM-Based Evidence Extraction for Meta-Analysis. arXiv preprint arXiv:2602.10881, 2026.

[14] Minghao Li, Ying Zeng, Zhihao Cheng, Cong Ma, and Kai Jia. Evaluating Deep Research Agents via Academic Survey Tasks. arXiv preprint arXiv:2508.15804, 2025.

[15] Brenda Nogueira, Werner Geyer, Andrew Anderson, Toby Jia-Jun Li, Dongwhi Kim, Nuno Moniz, and Nitesh V. Chawla. From Verification Burden to Trusted Collaboration: Design Goals for LLM-Assisted Literature Reviews. arXiv preprint arXiv:2512.11661, 2025.

[16] Hongbo Zhang, Han Cui, Yidong Wang, Yijian Tian, Qi Guo, Cunxiang Wang, Jian Wu, Chiyu Song, and Yue Zhang. Deep Literature Survey Automation with an Iterative Workflow. arXiv preprint arXiv:2510.21900, 2025.

[17] Dmitry Scherbakov, Nina Hubig, Vinita Jansari, Alexander Bakumenko, and Leslie A. Lenert. The emergence of Large Language Models (LLM) as a tool in literature reviews: an LLM automated systematic review. arXiv preprint arXiv:2409.04600, 2024.

[18] Lingbo Li, Anuradha Mathrani, and Teo Susnjak. Transforming Evidence Synthesis: A Systematic Review of the Evolution of Automated Meta-Analysis in the Age of AI. arXiv preprint arXiv:2504.20113, 2025.

[19] Yichen He, Guanhua Huang, Peiyuan Feng, Yuan Lin, Yuchen Zhang, Hang Li, and Weinan E. PaSa: An LLM Agent for Comprehensive Academic Paper Search. Annual Meeting of the Association for Computational Linguistics, 2025.

[20] Benjamin Newman, Yoonjoo Lee, Aakanksha Naik, Pao Siangliulue, Raymond Fok, Juho Kim, Daniel S. Weld, Joseph Chee Chang, and Kyle Lo. ArxivDIGESTables: Synthesizing Scientific Literature into Tables using Language Models. arXiv preprint arXiv:2410.22360, 2024.

[21] Priyanka Kargupta, Nan Zhang, Yunyi Zhang, Rui Zhang, Prasenjit Mitra, and Jiawei Han. TaxoAdapt: Aligning LLM-Based Multidimensional Taxonomy Construction to Evolving Research

Corpora. arXiv preprint arXiv:2506.10737, 2025.

[22] Jing Chen, Zhiheng Yang, Yixian Shen, Jie Liu, Adam Belloum, Chrysa Papagainni, and Paola Grosso. SurveyGen-I: Consistent Scientific Survey Generation with Evolving Plans and Memory-Guided Writing. arXiv preprint arXiv:2508.14317, 2025.

[23] Malik Abdul Sami, Zeeshan Rasheed, Kai-Kristian Kemell, Muhammad Waseem, Terhi Kilamo, Mika Saari, Anh Nguyen-Duc, Kari Systä, and Pekka Abrahamsson. System for systematic literature review using multiple AI agents: Concept and an empirical evaluation. arXiv preprint arXiv:2403.08399, 2024.

[24] Tetsu Kasanishi, Masaru Isonuma, Junichiro Mori, and Ichiro Sakata. SciReviewGen: A Large-scale Dataset for Automatic Literature Review Generation. arXiv preprint arXiv:2305.15186, 2023.

[25] Ambrose Robinson, William Thorne, Ben P. Wu, Abdullah Pandor, Munira Essat, Mark Stevenson, and Xingyi Song. Bio-SIEVE: Exploring Instruction Tuning Large Language Models for Systematic Review Automation. arXiv preprint arXiv:2308.06610, 2023.

[26] Yuxuan Lai, Yupeng Wu, Yidan Wang, Wenpeng Hu, and Chen Zheng. Instruct Large Language Models to Generate Scientific Literature Survey Step by Step. arXiv preprint arXiv:2408.07884, 2024.

[27] Xiaofeng Shi, Qian Kou, Yuduo Li, Ning Tang, Jinxin Xie, Longbin Yu, Songjing Wang, and Hua Zhou. SciSage: A Multi-Agent Framework for High-Quality Scientific Survey Generation. arXiv preprint arXiv:2506.12689, 2025.

[28] Shubham Agarwal, Gaurav Sahu, Abhay Puri, Issam H. Laradji, Krishnamurthy DJ Dvijotham, Jason Stanley, Laurent Charlin, and Christopher Pal. LitLLM: A Toolkit for Scientific Literature Review. arXiv preprint arXiv:2402.01788, 2024.

[29] Xiaofeng Shi, Yuduo Li, Qian Kou, Longbin Yu, Jinxin Xie, and Hua Zhou. SPAR: Scholar Paper Retrieval with LLM-based Agents for Enhanced Academic Search. arXiv preprint arXiv:2507.15245, 2025.

[30] Yuwen Du, Tian Jin, Jing Kang, et al. Towards Self-Evolving Agentic Literature Retrieval. arXiv preprint arXiv:2605.14306, 2026.

[31] Avishek Lahiri, Yufang Hou, and Debarshi Kumar Sanyal. TaxoAlign: Scholarly Taxonomy Generation Using Language Models. arXiv preprint arXiv:2510.17263, 2025.

[32] Shiqiang Cai, Nianhong Niu, Shizhu He, Kang Liu, and Jun Zhao. SC-Taxo: Hierarchical Taxonomy Generation under Semantic Consistency Constraints using Large Language Models. arXiv preprint arXiv:2605.00620, 2026.

[33] Yutong Li, Lu Chen, Aiwei Liu, Kai Yu, and Lijie Wen. ChatCite: LLM Agent with Human Workflow Guidance for Comparative Literature Summary. arXiv preprint arXiv:2403.02574, 2024.

[34] Yee Man Choi, Xuehang Guo, Yi R. (May) Fung, and Qingyun Wang. CiteGuard: Faithful Citation Attribution for LLMs via Retrieval-Augmented Validation. arXiv preprint arXiv:2510.17853, 2025.

[35] Ziyue Yang, Da Ma, Hanqi Li, et al. DeepSurvey: Enhancing Analytical Depth and Citation Reliability in Automated Survey Generation. arXiv preprint arXiv:2605.29522, 2026.

[36] Anthony Hevia, Sanjana Chintalapati, Veronica Ka Wai Lai, Thanh Tam Nguyen, Wai-Tat Wong, Terry Klassen, and Lucy Lu Wang. ROBoto2: An Interactive System and Dataset for LLM-assisted Clinical Trial Risk of Bias Assessment. arXiv preprint arXiv:2511.03048, 2025.

[37] Shreyansh Padarha, Ryan Othniel Kearns, Tristan Naidoo, et al. Evaluating AI-based Scientific Knowledge Synthesis with Epidemiological Systematic Reviews. arXiv preprint arXiv:2603.22327, 2026.

[38] Massimiliano Pronesti, Angelo Miculescu, Mohsin Kapdi, et al. AutoForest: Automatically Generating Forest Plots from Biomedical Studies with End-to-End Evidence Extraction and Synthesis. arXiv preprint arXiv:2606.02403, 2026.

[39] Zi Wang, Xingqiao Wang, Sangah Lee, and Xiaowei Xu. ARISE: Agentic Rubric-Guided Iterative Survey Engine for Automated Scholarly Paper Generation. arXiv preprint arXiv:2511.17689, 2025.

[40] Martin Docekal, Martin Fajcik, and Pavel Smrz. OARelatedWork: A Large-Scale Dataset of Related Work Sections with Full-texts from Open Access Sources. arXiv preprint arXiv:2405.01930, 2024.

[41] David Brett and Anniek Myatt. Patience is all you need! An agentic system for performing scientific literature review. arXiv preprint arXiv:2504.08752, 2025.

[42] Tong Bao, Mir Tafseer Nayeem, Davood Rafiei, and Chengzhi Zhang. SurveyGen: Quality-Aware Scientific Survey Generation with Large Language Models. arXiv preprint arXiv:2508.17647, 2025.

[43] Uri Katz, Mosh Levy, and Yoav Goldberg. Knowledge Navigator: LLM-guided Browsing Framework for Exploratory Search in Scientific Literature. arXiv preprint arXiv:2408.15836, 2024.

[44] Mingyu Huang, Shasha Zhou, Yuxuan Chen, and Ke Li. Conversational Exploration of Literature Landscape with LitChat. arXiv preprint arXiv:2505.23789, 2025.

[45] Nicholas Solovyev, Ryan Barron, Manish Bhattarai, Maksim E. Eren, Kim O. Rasmussen, and Boian S. Alexandrov. Interactive Distillation of Large Single-Topic Corpora of Scientific Papers. arXiv preprint arXiv:2309.10772, 2023.

[46] Lei Xiong, Kun Luo, Ziyi Xia, et al. AutoResearchBench: Benchmarking AI Agents on Complex Scientific Literature Discovery. arXiv preprint arXiv:2604.25256, 2026.

[47] Wojciech Kusa, Petr Knoth, and Allan Hanbury. CRUISE-Screening: Living Literature Reviews Toolbox. arXiv preprint arXiv:2309.01684, 2023.

[48] Christian Jaumann, Andreas Wiedholz, and Annemarie Friedrich. LGAR: Zero-Shot LLM-Guided Neural Ranking for Abstract Screening in Systematic Literature Reviews. arXiv preprint arXiv:2505.24757, 2025.

[49] Wojciech Kusa, Oscar E. Mendoza, Matthias Samwald, Petr Knoth, and Allan Hanbury. CSMeD: Bridging the Dataset Gap in Automated Citation Screening for Systematic Literature Reviews. arXiv preprint arXiv:2311.12474, 2023.

[50] Parsa Vares. Autodiscover: A reinforcement learning recommendation system for the cold-start imbalance challenge in active learning, powered by graph-aware thompson sampling. arXiv preprint arXiv:2602.05087, 2026.

[51] Sai Koneru, Jian Wu, and Sarah Rajtmajer. Context Selection for Hypothesis and Statistical Evidence Extraction from Full-Text Scientific Articles. arXiv preprint arXiv:2603.21193, 2026.

[52] Xiaochuan Liu, Ruihua Song, Xiting Wang, and Xu Chen. Select, Read, and Write: A Multi-Agent Framework of Full-Text-based Related Work Generation. arXiv preprint arXiv:2505.19647, 2025.

[53] Mahsa Shamsabadi and Jennifer D'Souza. From Keywords to Structured Summaries: Streamlining Scholarly Information Access. arXiv preprint arXiv:2402.14622, 2024.

[54] Sheng Zhang, Qin Liu, Renqian Luo, et al. Building Agent Harnesses for Scientific Curation from Multimodal Sources. arXiv preprint arXiv:2606.21005, 2026.

[55] Vijay Viswanathan, Graham Neubig, and Pengfei Liu. Leveraging the Citation Graph for Scientific Information Extraction. arXiv preprint arXiv:2106.01560, 2021.

[56] Sonakshi Gupta, Akhlak Mahmood, Wei Xiong, and Rampi Ramprasad. Retrieval Augmented Generation of Literature-derived Polymer Knowledge: The Example of a Biodegradable Polymer Expert System. arXiv preprint arXiv:2602.16650, 2026.

[57] Xiangci Li and Jessica Ouyang. Explaining Relationships Among Research Papers. arXiv preprint arXiv:2402.13426, 2024.

[58] Qi Chen, Jingxuan Wei, Zhuoya Yao, Haiguang Wang, Gaowei Wu, Bihui Yu, Siyuan Li, and Cheng Tan. ResearchPulse: Building Method-Experiment Chains through Multi-Document Scientific Inference. arXiv preprint arXiv:2509.03565, 2025.

[59] A Das, N Lourie, S Golkar, and M Pettee. What's In Your Field? Mapping Scientific Research with Knowledge Graphs and Large Language Models. arXiv preprint arXiv:2503.09894, 2025.

[60] Alejandro Godinez. HySemRAG: A Hybrid Semantic Retrieval-Augmented Generation Framework for Automated Literature Synthesis and Methodological Gap Analysis. arXiv preprint arXiv:2508.05666, 2025.

[61] Azanzi Jiomekong and Sanju Tiwari. An approach based on Open Research Knowledge Graph for Knowledge Acquisition from scientific papers. arXiv preprint arXiv:2308.12981, 2023.

[62] Vishakh Padmakumar, Joseph Chee Chang, Kyle Lo, Doug Downey, and Aakanksha Naik. Intent-Aware Schema Generation And Refinement For Literature Review Tables. arXiv preprint arXiv:2507.19521, 2025.

[63] Yuntong Hu, Zhuofeng Li, Zheng Zhang, Chen Ling, Raasikh Kanjiani, Boxin Zhao, and Liang Zhao. Taxonomy Tree Generation from Citation Graph. arXiv preprint arXiv:2410.03761v2, 2025.

[64] Xiangchao Yan, Shiyang Feng, Jiakang Yuan, Renqiu Xia, Bin Wang, Bo Zhang, and Lei Bai. Sur-

veyForge: On the Outline Heuristics, Memory-Driven Generation, and Multi-dimensional Evaluation for Automated Survey Writing. arXiv preprint arXiv:2503.04629, 2025.

[65] Kun Zhu, Lizi Liao, Yuxuan Gu, Lei Huang, Xiaocheng Feng, and Bing Qin. Context-Aware Hierarchical Taxonomy Generation for Scientific Papers via LLM-Guided Multi-Aspect Clustering. arXiv preprint arXiv:2509.19125, 2025.

[66] Yinqi Liu, Yueqi Zhu, Yongkang Zhang, Feiran Liu, Yutong Shen, Yufei Sun, Xin Wang, Renzhao Liang, Yidong Wang, and Cunxiang Wang. MVSS: A Unified Framework for Multi-View Structured Survey Generation. arXiv preprint arXiv:2601.09504, 2026.

[67] Xiangci Li, Yi-Hui Lee, and Jessica Ouyang. Cited Text Spans for Citation Text Generation. arXiv preprint arXiv:2309.06365, 2023.

[68] Shahbaz Syed, Ahmad Dawar Hakimi, Khalid Al-Khatib, and Martin Potthast. Citance-Contextualized Summarization of Scientific Papers. arXiv preprint arXiv:2311.02408, 2023.

[69] Xiangyu Li and Jingqiang Chen. SCIRGC: Multi-Granularity Citation Recommendation and Citation Sentence Preference Alignment. arXiv preprint arXiv:2505.20103, 2025.

[70] Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling LLMs to Generate Text with Citations (ALCE). arXiv preprint arXiv:2305.14627, 2023.

[71] Michael D. Skarlinski, Sam Cox, Jon M. Laurent, James D. Braza, Michaela Hinks, Michael J. Hammerling, Manvitha Ponnapati, Samuel G. Rodriques, and Andrew D. White. Language agents achieve superhuman synthesis of scientific knowledge. arXiv preprint arXiv:2409.13740, 2024.

[72] Anna Martin-Boyle, Aahan Tyagi, Marti A. Hearst, and Dongyeop Kang. Shallow Synthesis of Knowledge in GPT-Generated Texts: A Case Study in Automatic Related Work Composition. arXiv preprint arXiv:2402.12255, 2024.

[73] Leandro Carísio Fernandes, Gustavo Bartz Guedes, Thiago Soares Laitz, Thales Sales Almeida, Rodrigo Nogueira, Roberto Lotufo, and Jayr Pereira. SurveySum: A Dataset for Summarizing Multiple Scientific Articles into a Survey Section. arXiv preprint arXiv:2408.16444, 2024.

[74] Weiqi Wang, Jiefu Ou, Yangqiu Song, Benjamin Van Durme, and Daniel Khashabi. arXiv2Table: Toward Realistic Benchmarking and Evaluation for LLM-Based Literature-Review Table Generation. arXiv preprint arXiv:2504.10284, 2025.

[75] James Stewart-Evans, Emma Wilson, Tessa Langley, Andrew Prayle, Angela Hands, Karen Exley, and Jo Leonardi-Bee. Expediting data extraction using a large language model (LLM) and scoping review protocol: a methodological study within a complex scoping review. arXiv preprint arXiv:2507.06623, 2025.

[76] Teo Sušnjak, Peter Hwang, N. Reyes, A. Barczak, Timothy R. Mcintosh, and Surangika Ranathunga. Automating Research Synthesis with Domain-Specific Large Language Model Fine-Tuning. ACM Transactions on Knowledge Discovery from Data, 2024.

[77] Arya Rahgozar and Pouria Mortezaagha. AI Co-Scientist for Knowledge Synthesis in Medical Contexts: A Proof of Concept. arXiv preprint arXiv:2601.11825, 2026.

[78] Xinyu Mao, Teerapong Leelanupab, Martin Potthast, Harrisen Scells, and Guido Zuccon. AiReview: An Open Platform for Accelerating Systematic Reviews with LLMs. arXiv preprint arXiv:2504.04193, 2025.

[79] Teo Susnjak. PRISMA-DFLLM: An Extension of PRISMA for Systematic Literature Reviews using Domain-specific Finetuned Large Language Models. arXiv preprint arXiv:2306.14905, 2023.

[80] Yu Chao, Siyu Lin, Xiaorong Wang, Zhu Zhang, Zihan Zhou, Haoyu Wang, Shuo Wang, Jie Zhou, Zhiyuan Liu, and Maosong Sun. LLM×MapReduce-V3: Enabling Interactive In-Depth Survey Generation through a MCP-Driven Hierarchically Modular Agent System. arXiv preprint arXiv:2510.10890, 2025.

[81] Siyi Wu, Chiaxin Liang, Ziqian Bi, Leyi Zhao, Tianyang Wang, Junhao Song, Yichao Zhang, Keyu Chen, Benji Peng, and Xinyuan Song. AutoSurvey2: Empowering Researchers with Next Level Automated Literature Surveys. arXiv preprint arXiv:2510.26012, 2025.

[82] Yating Pan, Jiajun Zhang, Jun Wang, and Qi Su. Extending AI for Research to the Humanities: A Multi-Agent Framework for Evidence-Grounded Scholarship. arXiv preprint arXiv:2605.30947, 2026.

[83] Z Li, X Guan, B Zhang, S Huang, H Zhou, S Lai, M Yan, et al. WebWeaver: Structuring Web-Scale

Evidence with Dynamic Outlines for Open-Ended Deep Research. arXiv preprint arXiv:2509.13312, 2025.

[84] Lunyiu Nie, Nedim Lipka, Ryan A. Rossi, and Swarat Chaudhuri. Efficient Tree-Structured Deep Research with Adaptive Resource Allocation. arXiv preprint arXiv:2510.05145, 2025.

[85] Guo-Biao Zhang, Ding-Yuan Liu, Da-Yi Wu, Tian Lan, Heyan Huang, Zhijing Wu, and Xian-Ling Mao. DeepSurvey-Bench: Evaluating Academic Value of Automatically Generated Scientific Survey. arXiv preprint arXiv:2601.15307, 2026.

[86] Hongzhi Zhang, Yuanze Hu, Tinghai Zhang, et al. DeepSynth-Eval: Objectively Evaluating Information Consolidation in Deep Survey Writing. arXiv preprint arXiv:2601.03540, 2026.

[87] Weihang Su, Anzhe Xie, Qingyao Ai, Jianming Long, Xuanyi Chen, Jiaxin Mao, Ziyi Ye, and Yiqun Liu. SurGE: A Benchmark and Evaluation Framework for Scientific Survey Generation. Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2026.

[88] Beichen Guo, Zhiyuan Wen, Jia Gu, Haochen Shi, Jian Wang, Senzhang Wang, Haoyang Li, Ruosong Yang, and Shuaiqi Liu. SurveyLens: A Discipline-Aware Benchmark for Automatic Survey Generation. arXiv preprint arXiv:2602.11238, 2026.

[89] Ming Zhang, Jiabao Zhuang, Wenqing Jing, et al. Can Deep Research Agents Retrieve and Organize? Evaluating the Synthesis Gap with Expert Taxonomies. arXiv preprint arXiv:2601.12369, 2026.

[90] Zhaojun Sun, Xuzhou Zhu, Xuanhe Zhou, Xin Tong, Shuo Wang, Jie Fu, Guoliang Li, Zhiyuan Liu, and Fan Wu. Can LLM(-Agents) Write Academic Surveys that Align with Reader Needs? arXiv preprint arXiv:2510.03120, 2025.

[91] Hao Kang and Chenyan Xiong. ResearchArena: Benchmarking Large Language Models' Ability to Collect and Organize Information as Research Agents. arXiv preprint arXiv:2406.10291, 2024.

[92] Liana Patel, Negar Arabzadeh, Harshit Gupta, Ankita Sundar, Ion Stoica, Matei Zaharia, and Carlos Guestrin. DeepScholar-Bench: A Live Benchmark and Automated Evaluation for Generative Research Synthesis. arXiv preprint arXiv:2508.20033, 2025.

[93] Mingxuan Du, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. DeepResearch Bench: A Comprehensive Benchmark for Deep Research Agents. arXiv preprint arXiv:2506.11763, 2025.

[94] Gaurav Sahu, Laurent Charlin, and Christopher Pal. Rethinking Literature Search Evaluation: Deep Research Helps, and Human Citation Lists Are Not a Ground Truth. arXiv preprint arXiv:2605.29234, 2026.

[95] Furkan Şahinuç, Subhabrata Dutta, and Iryna Gurevych. Expert Preference-based Evaluation of Automated Related Work Generation. arXiv preprint arXiv:2508.07955, 2025.

[96] Fan Gao, Hang Jiang, Rui Yang, Qingcheng Zeng, Jinghui Lu, Moritz Blum, Dairui Liu, Tianwei She, Yuang Jiang, and Irene Li. Large Language Models on Wikipedia-Style Survey Generation: an Evaluation in NLP Concepts. arXiv preprint arXiv:2308.10410, 2023.

[97] Yifei Li, Xiang Yue, Zeyi Liao, and Huan Sun. AttributionBench: How Hard is Automatic Attribution Evaluation? arXiv preprint arXiv:2402.15089, 2024.

[98] Jiahao Zhao, Shuaixing Zhang, Nan Xu, and Lei Wang. SurveyEval: Towards Comprehensive Evaluation of LLM-Generated Academic Surveys. arXiv preprint arXiv:2512.02763, 2025.

[99] Beichen Guo, Zhiyuan Wen, Yu Yang, Peng Gao, Ruosong Yang, and Jiaxing Shen. Comprehensive Multifaceted and Similarity-Enhanced Benchmark for Automatic Survey Generation. arXiv preprint arXiv:2508.11310, 2025.

[100] Zhiyuan Wen, Jiannong Cao, Zian Wang, Beichen Guo, Ruosong Yang, and Shuaiqi Liu. InteractiveSurvey: An LLM-based Personalized and Interactive Survey Paper Generation System. arXiv preprint arXiv:2504.08762, 2025.

[101] Runlong Ye, Naaz Sibia, Angela Zavaleta Bernuy, Tingting Zhu, Carolina Nobre, Viktoria Pammer-Schindler, and Michael Liut. From Toil to Thought: Designing for Strategic Exploration and Responsible AI in Systematic Literature Reviews. Proceedings of the 31st International Conference on Intelligent User Interfaces, 2026.

[102] Runlong Ye, Patrick Yung Kang Lee, Matthew Varona, Oliver Huang, and Carolina Nobre. ScholarMate: A Mixed-Initiative Tool for Qualitative Knowledge Work and Information Sensemaking. arXiv preprint arXiv:2504.14406, 2025.

[103] Nishant Balepur, Malachi Hamada, Varsha Kishore, Sergey Feldman, Amanpreet Singh, Pao Siangliulue, Joseph Chee Chang, Eunsol Choi, Jordan Lee Boyd-Graber, and Aakanksha Naik. Language Models Don't Know What You Want: Evaluating Personalization in Deep Research Needs Real Users. arXiv preprint arXiv:2603.16120, 2026.

[104] Lei Xiong, Huaying Yuan, Zheng Liu, Zhao Cao, and Zhicheng Dou. PaperScope: A Multi-Modal Multi-Document Benchmark for Agentic Deep Research Across Massive Scientific Papers. arXiv preprint arXiv:2604.11307, 2026.

[105] Razeen A Rasheed, Somnath Banerjee, Animesh Mukherjee, and Rima Hazra. From Fluent to Verifiable: Claim-Level Auditability for Deep Research Agents. arXiv preprint arXiv:2602.13855, 2026.

[106] Teo Susnjak. Compiling Prompts, Not Crafting Them: A Reproducible Workflow for AI-Assisted Evidence Synthesis. arXiv preprint arXiv:2509.00038, 2025.