DTU Engineering Technology
Department of Engineering Technology

Artificial Intelligence for Scientific Research and Scholarly Communication

Methods, Systems, and Evaluation

Master Thesis



Artificial Intelligence for Scientific Research and Scholarly Communication
Methods, Systems, and Evaluation

Master Thesis
November 2026

By
Xin Huang

Copyright: Reproduction of this publication in whole or in part must include the customary bibliographic citation, including author attribution, report title, etc.
Cover photo: Vibeke Hempler, 2012
Published by: DTU, Department of Engineering Technology, Lautrupvang 15, 2750 Ballerup, Denmark
https://engtech.dtu.dk/

Approval

This thesis has been prepared over five months at the Department of Engineering Technology, Technical University of Denmark (DTU), in fulfilment of the degree of Master of Science in Computer Science and Engineering.

Xin Huang - s243442

.....

Signature

.....

Date

Abstract

Research output keeps growing, but the resources for quality assurance, and for peer review in particular, have not kept pace. Large language models (LLMs) can produce plausible but fabricated references and content errors in large numbers. At present they are also among the few tools that are cheap and fast enough to support checking at this scale. This thesis studies how LLMs and LLM-based agents can support scholarly quality control. It addresses three questions. Which tasks are suitable for this kind of support, and how should the tools be designed? How well do the tools perform, and how can they be evaluated when no established ground truth is available? Under what conditions is the assistance reliable, where are its limits, and how should it be used?

The thesis arranges verification tasks on a three-level delegation ladder, which is defined by two criteria. The first, decidability, asks whether a deterministic external adjudication procedure exists. The second, ground-truth availability, asks where the reference answers needed for evaluation come from. The three levels are full delegation (L1), checked delegation (L2) and advisory delegation (L3). For each level the thesis presents one system, and all three are deployed on the scholarly platform CSPaper. The framework is also formulated as a simple decision model, in which seven propositions state the assumptions behind the thresholds, scores and evaluation metrics used in the systems.

Reference Check is at L1. The LLM only parses the reference list into entries. The decision itself is made by six scholarly databases and deterministic matching rules, and entries that cannot be found are marked as Unverified and handed back to a human. According to the platform, the system detected 93% of 100 problematic references taken from real papers. In a separate run on the 42 real references of one paper, 41 were verified.

Correctness Check is at L2. It looks for objective errors in a paper's derivations, numbers and arguments. Its output is a list of flagged passages, each of which has to be verified by a human. In a preliminary experiment, the prototype pipeline was run on 20 papers from the third-party benchmark FLAWS, with one run per model. The four models reached an accuracy@10 of between 25% and 40%. For the two models for which the FLAWS authors report a single-prompt baseline, the results were above that baseline (40.0% against 39.1% for gpt-5, and 35.0% against 19.8% for gemini-2.5-pro). The journal TMLR organised an evaluation to select an AI reviewer. In it, the CSPaper review system, which is built around Correctness Check, agreed best with the editors' judgement. TMLR then chose it to provide an advisory AI review alongside the human reviews. For now, tools of this kind can alert reviewers to suspicious passages but cannot replace the reviewer's judgement.

Reference Finder is at L3. Its task is reference completion. There is no unique correct answer to which paper a sentence should cite, so the system can only make suggestions. All candidates are taken from the results returned by up to nine scholarly search sources, and the LLM is used only to identify statements and to rerank the candidates. Existing benchmarks do not fit reference completion for a whole paper, so the thesis designs a strip-and-recover protocol: the reference list and the citation markers of a paper are removed, and the system is measured by how many of the originally cited works it recovers. In four test runs on the original text of three papers, the system identified 46 statements that need a citation and, for 40 of them, found a paper with a relevance score of at least 7. All 205 entries in the suggested bibliographies carry a DOI, an arXiv identifier or a link, so each one can be looked up and checked.

One approach runs through the thesis: LLMs as the language interface, deterministic sources as the adjudicator, and humans handling the residual.

Acknowledgements

I would like to express my sincere gratitude to the three people who supervised this thesis. Lele Cao, my company supervisor at CSPaper, and Lei You, Assistant Professor in the Department of Engineering Technology at DTU, guided it from the choice of topic through the methods to the writing. Birger Andersen, Professor in the same department at DTU, gave me valuable insight and support throughout the project. I am grateful to all three for the time they gave to my work, and for their advice and encouragement whenever I ran into difficulties. Without them I could not have completed this thesis, and I have grown a great deal under their supervision.

My warm thanks also go to Kai Xie, Weiping Ding, and Yong Du, and to the whole research and development team of CSPaper. The three systems in this thesis could not have been developed and deployed on the platform without the team's support in engineering and product work. I also thank Scholar7 AB for the chance to carry out this project on a live platform. Working with this team has taught me what it takes to turn an idea into a working product.

Finally, I am deeply grateful to my family and friends. Their understanding and support have been with me throughout my master's studies, and they mattered most in the months when I was working and writing this thesis at the same time. These five months were demanding, and I would not have managed them alone. As my master's studies come to an end, I thank everyone who has helped me along the way.

Statement

Generative AI tools were used during the preparation of this thesis to assist with drafting and revising the text, translating it into English, polishing the language and refining code. I reviewed and revised all AI-generated suggestions, and I take responsibility for the final text, the implementation and analysis, and the conclusions presented in this thesis.

Contents

Approval . . . ii
Abstract . . . iii
Acknowledgements . . . iv
Statement . . . v

1 Introduction 1
1.1 Background . . . 1
1.2 The two sides of large language models . . . 1
1.3 Objectives and choice of topics . . . 2
1.4 Research questions . . . 3
1.5 Approach: the delegation ladder . . . 3
1.6 Methods and evaluation . . . 4
1.7 Contributions . . . 5
1.8 Structure of the thesis . . . 5

2 Background and Related Work 6
2.1 LLM hallucination and citation integrity . . . 6
2.2 Automated reviewing and error detection . . . 7
2.3 Agents for the research workflow and citation recommendation . . . 7
2.4 Human–AI collaboration: automation bias and LLM-as-judge . . . 9
2.5 Positioning and gaps . . . 10

3 The Delegation Framework 11
3.1 Two criteria for delegation . . . 11
3.2 The three levels of delegation . . . 12

4 A Mathematical Formulation of Delegation 14
4.1 Basic setting . . . 14
4.2 Full delegation: matching thresholds and abstention . . . 15
4.3 Checked delegation: scoring, truncation and evaluation . . . 15
4.4 Advisory delegation: ranking and evaluation metrics . . . 17
4.5 What the three ground-truth mechanisms allow one to conclude . . . 18
4.6 Assumptions and limitations . . . 18

5 L2 Checked Delegation: Correctness Check 20
5.1 Task definition . . . 20
5.2 The six-stage pipeline . . . 20
5.3 Experimental setup . . . 21
5.4 Results of the preliminary experiment . . . 22
5.5 Results by injecting family . . . 23
5.6 Report volume and over-reporting . . . 23
5.7 External evaluation: TMLR's selection of an AI reviewer . . . 23
5.8 Summary . . . 25

6 L3 Advisory Delegation: Reference Completion with Reference Finder 26
6.1 Task definition . . . 26
6.2 System design . . . 26

6.3 Evaluation protocol 28
6.4 Test results 28
6.5 Summary 29

7 L1 Full Delegation: Reference Check 30
7.1 Task definition 30
7.2 System design 30
7.3 Evaluation 31
7.4 Summary 31

8 Cross-Case Analysis 32
8.1 Comparison of the three systems 32
8.2 Sources of ground truth and the strength of conclusions 33
8.3 Recommendations for use 33
8.4 Societal perspective and sustainability 34
8.5 Limitations 35

9 Conclusion and Future Work 36
9.1 Answers to the research questions 36
9.2 Contributions 36
9.3 Future work 37
9.4 Closing remarks 37

References 38

1 Introduction

1.1 Background

Research output keeps growing, but the resources for quality assurance have not grown with it. Peer review shows this most clearly, above all at the main machine learning conferences. ICLR received 1,013 submissions in 2018 and 19,814 in 2026. For ICLR 2027, the count after the submission deadline stood at 42,790, more than double the previous year's figure [1].

Reviewing capacity has not kept up with this growth. When the ICLR 2027 Program Chairs explained their new submission rules, they wrote that in a fast-growing field there may simply not be enough qualified reviewers to read everything people want to publish. For about 20% of the ICLR 2026 submissions, none of the authors was qualified to review [2]. ICLR 2027 therefore limits each author to 20 submissions and new authors to one. AISTATS, ACL and the journal TMLR have introduced similar limits, and submissions to TMLR tripled within a year [3].

At the same time, reviewers have only limited attention to give to each paper. Xi et al. [4] point out that the exponential growth of research output, combined with a limited number of human experts, makes it increasingly hard for reviewers to find errors reliably. Bianchi et al. [5] show that even papers published at top conferences contain a non-negligible number of objective errors, and that the average number per paper is rising each year. At NeurIPS, for example, the average rose from 3.8 in 2021 to 5.9 in 2025, an increase of 55.3%. A paper usually combines complex theoretical derivations, experimental design and a network of citations, and an objective error can sit in any of these. A reviewer under time pressure can hardly check every part thoroughly. Shah [6] likewise reports that expert reviewers may miss major errors in a paper and still recommend acceptance.

1.2 The two sides of large language models

Large language models (LLMs) have made papers, citations and reviews much cheaper to produce, while reviewers have no more time than before. You et al. [7] describe this imbalance as verification pressure:

$$\Lambda = \frac{R \cdot C_{\text{eff}}}{B_{\text{eff}}}$$

Here R is the number of claims that need to be evaluated per unit of time, C_{eff} is the cost of obtaining decisive evidence for one claim, and B_{eff} is the amount of evidence the community can process per unit of time. B_{eff} depends on reviewers' time, their expertise and the external checking tools available to them. As long as Λ does not exceed 1, most claims can in principle be verified. Once Λ exceeds 1, verification becomes selective, evaluation falls back on cheap proxy signals, and the outcome of an evaluation says less about whether the paper is correct. The submission caps described in Section 1.1 push R down, but LLMs push it up again. According to the ICLR chairs, the barrier to producing something that looks like a paper is now much lower than it used to be [2].

The change is most visible in citations. References are the evidence in a paper that is simplest to check, yet fabricated entries still pass review. Sakai et al. [8] found nearly 300 papers with fabricated references among the papers published at ACL, NAACL and

EMNLP in 2024 and 2025, more than a hundred of them from EMNLP 2025. Among the papers accepted at NeurIPS 2025, 100 fabricated citations were found in 53 papers, about 1% of all accepted papers [9]. Each case is a small problem on its own, but together they show that even this evidence is not being checked in full. A second problem with citations is omission. Robinson and Goodman [10] examined 1,523 randomised controlled trials and found that a trial report cited a median of only 21% of the earlier trials on the same question.

Reviewing itself is affected as well. About 21% of the reviews for ICLR 2026 were classified by a detection tool as fully AI-generated [11]. ICML 2026 found that 398 reviewers had violated the LLM policy they had agreed to follow and desk-rejected their 497 submissions, about 2% of all submissions [12]. Using LLMs to imitate reviews increases the output of evaluation without increasing the capacity to verify [7].

On the other hand, LLMs are currently among the few tools whose cost and speed can support checking at this scale, and they are already used in real review processes. AAAI-26 generated one clearly labelled AI review for each of the 22,977 submissions that entered full review [13]. In July 2026 TMLR announced a pilot of AI reviews for its Action Editors (AEs) to consult, and the reviewer used in the pilot is built around the system described in Chapter 5 of this thesis (Section 5.7). Under the definition of verification pressure, AI is genuinely useful where it lowers C_{eff} and raises B_{eff}. This means producing verification results that can be checked individually rather than imitating reviews, an orientation that You et al. [7] call verification-first. Current models cannot yet draw conclusions on their own, though. On the FLAWS benchmark, the top 10 suspicious excerpts listed by the best model hit the injected error in only 39.1% of the papers [4]. The question is therefore no longer whether to use AI but which work can be handed to it, how far, and who is responsible for the part it cannot handle.

1.3 Objectives and choice of topics

This project studies how artificial intelligence, in particular LLMs and LLM-based agents, can support scientific research and scholarly communication. I design and implement systems in this area and evaluate them empirically. I also examine how reliable such assistance is, where its limits are, and how it can be used responsibly in actual research and review processes. The concrete aim is to develop and evaluate a small set of LLM-based tools for research and scholarly quality control.

The project plan envisaged that two or three directions would be pursued in depth and that the systems that passed testing would be deployed on CSPaper, the platform of the partner company Scholar7 AB. After the literature review and the feasibility experiments of the first phase, I settled on three directions, each with one system.

1. Verifying a paper's references, including whether the cited works exist and whether their bibliographic details are correct. The system is Reference Check (product name: Reference Check Agent). It checks whether each reference can be matched to a record in a scholarly database (Chapter 7).

2. Assessing the paper itself, for example by checking important claims and looking for errors. The system is Correctness Check (product name: Correctness Check Agent). It checks a paper's core claims and looks for objective errors in the full text (Chapter 5).

3. Helping with other parts of the research process, such as finding relevant literature or spotting citations that may be missing. The system is Reference Finder. It finds

the statements in a paper that need a citation and recommends papers for them from real search results (Chapter 6).

All three systems are now deployed on CSPaper.

1.4 Research questions

The project plan poses three research questions.

- RQ1: Which tasks in scholarly quality control and the wider research process are suitable for LLM-based support, and how should tools for the selected tasks be designed?
- RQ2: How well do the resulting tools perform, and how can they be evaluated when no established ground truth or benchmark is available?
- RQ3: Under what conditions is this kind of assistance reliable, where are its limits, and how should it be used in research and peer review?

Table 1.1 shows how the thesis makes these questions concrete.

Table 1.1: How the research questions are made concrete in this thesis

Research question	Concrete form in this thesis	Where it is answered
RQ1	Which properties of a verification task determine how far it can be handed to AI; how each of the three systems is designed	Chapters 3 and 4; Sections 5.1, 5.2, 6.1, 6.2, 7.1 and 7.2
RQ2	How Correctness Check performs on the public benchmark FLAWS and in an external evaluation; how to build a credible evaluation for reference completion, which has no established ground truth; the test results available for Reference Check; how strong a conclusion each source of ground truth supports	Sections 5.3 to 5.7; 6.3 and 6.4; 7.3; 4.5 and 8.2
RQ3	Under what conditions the three systems are reliable and how they fail; in what form their output should be handed to people in reviewing and writing	Sections 5.8, 6.5 and 7.4; Chapter 8

1.5 Approach: the delegation ladder

Most research on AI reviewing asks whether AI can find the problems in a paper. The answer is almost always “some of them”, which does not help much when deciding whether to deploy a system. An error detector with a 40% detection rate is clearly not good enough if its output is treated as a conclusion. If it is only used to tell reviewers where to look, it may still bring a net benefit. This thesis therefore answers the three research questions within a single framework, that of delegation. The framework asks whether a system’s output should be treated as a verdict, a flag to be verified or a suggestion, at which step a human comes in, and who is responsible when something goes wrong.

I use two criteria to judge how far a verification task can be delegated (Chapter 3). Decidability asks whether there is an external procedure that does not rely on a generative model and can decide deterministically whether an answer is right or wrong. Ground-truth availability asks where the reference answers needed for evaluation come from: existing

records (exact), deliberately injected errors (injected), or a substitute recovered from the evaluated object itself (recovered). Together the two criteria place verification tasks on a three-level delegation ladder, shown in Figure 1.1.

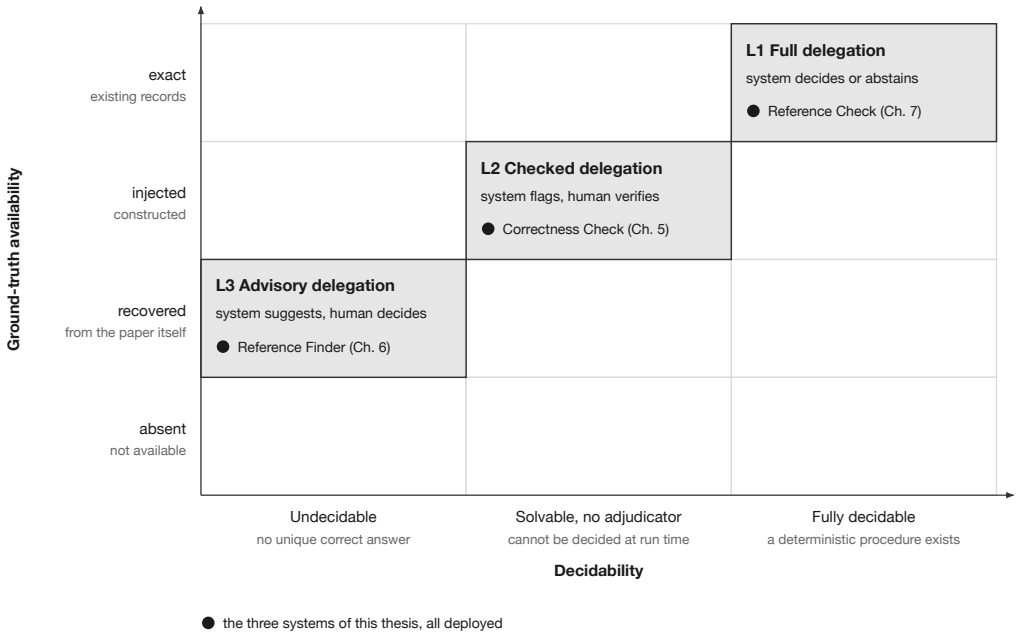

Figure 1.1: The delegation ladder. The horizontal axis is decidability and the vertical axis is ground-truth availability. The three grey cells are the three levels of delegation, and the dots are the three systems of this thesis.

- L1, full delegation. The task is decidable and ground truth already exists. The system can decide directly, and humans only handle the cases where it abstains. Reference Check is at this level.
- L2, checked delegation. The task cannot be decided at run time, but ground truth can be constructed. The system can only act as a prompter, and everything it points out has to be verified by a human. Correctness Check is at this level.
- L3, advisory delegation. The task has no unique correct answer, and only a substitute for ground truth is available. The system's output is only a suggestion, and the decision stays with the human. Reference Finder is at this level.

The level of delegation follows from the properties of the task. Adding more pipeline stages to an open task does not produce the deterministic verdict that L1 requires. Chapter 4 formulates the framework as a simple decision model and states the assumptions behind the thresholds, scores and evaluation metrics used in the three systems.

1.6 Methods and evaluation

The method has three parts.

Literature review. Chapter 2 reviews four areas of work: the problems that AI creates for scholarly publishing, automated quality control and peer review, agents for the research workflow, and human–AI collaboration.

System development. All three systems have a modular design and were developed iteratively. The code is under version control and comes with tests and documentation.

Evaluation. I use public data where suitable data exist and build test data where they do not. Correctness Check is evaluated on the third-party public benchmark FLAWS, and the thesis also draws on an external evaluation organised by TMLR. No existing benchmark fits reference completion for a whole paper, so I designed a strip-and-recover protocol for it and tested the system on real papers. For Reference Check, the thesis cites a test that the platform ran at launch on fabricated references that occurred in real papers.

1.7 Contributions

- C1, framework. A three-level delegation ladder defined by decidability and ground-truth availability (Chapter 3). Chapter 4 formulates the framework as a decision model and maps each part of it onto the implementation of the three systems.
- C2, systems. Correctness Check runs in production on CSPaper. After TMLR's evaluation, the review agent built around it was selected as the system for the journal's AI review pilot (Chapter 5). Reference Finder and Reference Check have been developed and deployed on the partner platform (Chapters 6 and 7).
- C3, evaluation. A preliminary evaluation of the Correctness Check prototype on FLAWS, with an analysis of how the judge and the source of the injected errors affect the results (Sections 5.4 and 5.5). A strip-and-recover protocol for reference completion, which has no established ground truth (Section 6.3), and tests of the system on real papers (Section 6.4).

1.8 Structure of the thesis

In the title of the thesis, Methods refers to the delegation ladder and the evaluation protocols, Systems to the three systems, and Evaluation to their empirical evaluation and comparative analysis.

Chapter 2 covers related work. Chapter 3 presents the framework, and Chapter 4 gives its mathematical formulation. Chapters 5, 6 and 7 each present one system and its evaluation. Chapter 8 looks at the three systems together and discusses the conditions for reliability, the limitations and the recommended use. Chapter 9 answers the three research questions and outlines future work.

2 Background and Related Work

This chapter reviews related work in the four areas listed in the project plan: the problems that AI creates for scholarly publishing (Section 2.1), automated quality control and peer review (Section 2.2), agents for the research workflow and citation recommendation (Section 2.3), and human–AI collaboration (Section 2.4). Section 2.5 summarises the gaps this thesis addresses.

2.1 LLM hallucination and citation integrity

It was documented early on that LLMs generate references to works that do not exist. Walters and Wilder [14] asked ChatGPT to write a short literature review with references on each of 42 topics and then checked all 636 references. Of the citations generated by GPT-3.5, 55% were fabricated, and with GPT-4 the share was still 18%. Even among the entries that did exist, 43% and 24% respectively contained substantive bibliographic errors. The existence of a reference and the correctness of its details therefore have to be checked separately.

Ji et al. [15] give a general survey of hallucination, and Agrawal et al. [16] go on to study whether a model can recognise the references it has made up itself. Among the different kinds of hallucination, fabricated references are a special case, because they can be checked against a database. This thesis therefore places reference verification at the lowest level of the delegation ladder (Section 3.2), and Reference Finder only produces recommendations from real search results.

Fabricated references have made their way into published papers. Sakai et al. [8] examined the papers published at ACL, NAACL and EMNLP in 2024 and 2025 and found nearly 300 papers with fabricated references, more than a hundred of which were accepted at the EMNLP 2025 main conference or Findings. A further 100 fabricated citations were found among the papers accepted at NeurIPS 2025. Ansari [9] sorts these into five failure modes, of which complete fabrication accounts for 66%. The platform test cited in Chapter 7 uses the corpus of Sakai et al.

Hallucination detection is closely related to a wider body of research and tools on citation integrity. The citation index scite [17] uses deep learning to classify citations by their context as supporting, contrasting or mentioning, and so goes beyond counting how often a work is cited. Cabanac et al. [18] detect tortured phrases, which is an example of screening the literature in bulk after publication.

Checking existence relies on open bibliographic databases. OpenAlex [19] provides fully open scholarly metadata, and the Semantic Scholar open data platform [20] offers a graph of around two hundred million papers together with an API. Both serve as data sources in this thesis.

Two things are missing in this line of work. Most existing measurements are retrospective spot checks, on a scale that cannot keep up with the volume of generated text. Existing tools also deal almost exclusively with references that are already there. They hardly cover works that should have been cited and were not, which is the case Reference Finder handles.

2.2 Automated reviewing and error detection

Shah [6] gives a systematic account of the problems of peer review in terms of scale, noise, bias and incentives, and of the computational methods tried so far. Since LLMs appeared, a good deal of work has asked whether a model can write useful review comments directly. One example is the large-scale empirical study by Liang et al. [21] of GPT-4-generated feedback on papers. The study examines how far GPT-4's feedback overlaps with the comments of human reviewers and whether authors find it useful, and it shows that LLM feedback is usable on some dimensions. Systems of this kind have also been tried at the scale of a conference. In a survey after the AAAI-26 pilot, authors and programme committee members preferred the AI reviews on dimensions such as technical accuracy and research suggestions [13]. Writing a whole review, however, is an open task that is hard to adjudicate. Instead of generating whole reviews, this thesis singles out the subtasks of reviewing that can be verified.

FLAWS [4] is the error localisation benchmark used in Chapter 5. It addresses the lack of ground truth in evaluation. The authors took papers accepted at ICML 2025 and had an LLM insert errors that undermine a key claim, which gave 713 paper–error pairs with ground truth. Of these, 265 were injected by GPT-5 and 448 by Gemini 2.5 Pro. The source papers all first appeared on arXiv after January 2025, and the data are released under a CC BY 4.0 licence. The type of error is not restricted at injection time, and the injecting model afterwards assigns each error to one of six categories. A detection counts under either of two criteria: a word-level normalised similarity of at least 0.5, or a positive verdict from an LLM judge, which is the model that injected the error. Among the raw LLM baselines reported in the paper, GPT-5 has the highest accuracy@10 at 39.1%, followed by DeepSeek Reasoner v3.1 at 35.2%, Grok 4 at 23.4%, Claude Sonnet 4.5 at 21.5% and Gemini 2.5 Pro at 19.8%. FLAWS is a public benchmark built by a third party, and I took no part in building it. Two of its authors later used the same method to construct a new set of papers with injected errors, which served to select an AI reviewer for the journal TMLR [22, see Section 5.7]. Bianchi et al. [5] use a GPT-5-based checker to count objective errors in published papers. Human review of the checker's annotations found a precision of 83.2%. This work shows that an LLM can be used for such a count, and also that the tool used for counting itself needs to be calibrated. Benchmarks aimed at verification have begun to appear as well. SciCoQA [23], for example, collects 635 cases in which a paper and its code disagree and is used to test whether models can detect such discrepancies.

Ground truth by injection has two limitations. The injected errors are generated by an LLM, so their distribution need not match the errors human authors make naturally, and detection rates cannot be carried over directly to real reviewing. The benchmark is also public, so later models may have seen it during training. A separate concern is the judge: having the injecting model act as judge may bring the preferences of its model family into the verdicts (Section 5.3).

2.3 Agents for the research workflow and citation recommendation

Citation recommendation as a task in its own right goes back to the context-aware citation recommendation of He et al. [24]. The survey by Färber and Jatowt [25] covers the methods and datasets since then. Current systems commonly use a two-stage structure of coarse retrieval followed by reranking [26, 27]. Reference Finder follows this structure, but neither stage needs training: retrieval goes through online scholarly APIs, and reranking is left to an LLM.

Approaches built around representations of scientific text do very well on offline benchmarks, but they need a training corpus and a vector index that has to be maintained. Representative work includes SciBERT [28], SPECTER [29], which trains document embeddings with the citation graph as the supervision signal, SciNCL [30], which improves on it by contrastive learning over neighbourhoods in the citation graph, and the multi-task SPECTER2 [31]. This thesis does not take that route, for reasons of engineering and deployment. Reference Finder needs no model training and keeps no local index. It calls online scholarly APIs, which are updated continuously. The intermediate results of every step can be inspected by a person, including the identified statements, the queries, the candidate pool and the reasons for the ranking. This also fits the requirement that every suggestion can be traced back to its source.

CiteWorth by Wright and Augenstein [32] treats the question of which sentences need a citation as a detection task of its own. It also points out a methodological problem that matters for this thesis. Removing citation markers from a corpus leaves dangling punctuation and sentence fragments, and a detector may learn nothing more than these traces. To avoid this, they keep only citations in standard format at the end of a sentence, clean up the dangling punctuation and check the result again with regular expressions. Their cleaning method was rated successful in 98.1% of cases in a human evaluation, against 92.78% for the naive approach. The strip-and-recover setting of Chapter 6 has the same problem.

Another related line of work has LLMs generate text with citations and then tests the quality of the citations. ALCE [33] is an evaluation benchmark for generated text and the quality of its citations. Bohnet et al. [34] turn the requirement that an output be attributable to a source into a systematic evaluation setting for question answering. Like this thesis, these works require the output to be grounded in search results, but their goal is different. They generate text. This thesis only recommends papers and links them to statements, and it generates no text that would appear in the paper. This is a deliberate restriction under advisory delegation (Section 6.1).

Agents for scientific literature have mostly worked on question answering and literature synthesis in recent years, for example PaperQA2 [35] and OpenScholar [36]. Closer to Reference Finder is work on finding the source for a given piece of text. CiteME [37] takes excerpts that cite exactly one paper and tests whether a model can identify that paper. Frontier models answering directly reach an accuracy of only 4.2% to 18.5%, the retrieval-equipped CiteAgent reaches 35.3%, and humans reach 69.7%. CiteGuard [38] raises the accuracy on this benchmark to 68.1%. ATTRICITE [39] explicitly treats the original authors' citations as an observable attribution signal, which matches the idea of recovered ground truth in Chapter 6. In all of this work, the input is an excerpt that contains a single citation whose position is already marked, and the answer is a single paper. MasterSet [40] is a paper-level benchmark for must-cite recommendation that takes only the title and abstract as input. It retrieves from more than 150,000 papers at 15 venues and is evaluated by Recall@K. The systems closest in form are CiteLLM [41] and the concurrent ReCite [42]. CiteLLM requires the authors themselves to select the sentences that need a citation, while ReCite works paragraph by paragraph and depends on a specially trained model. CiteLLM builds citation retrieval into a LaTeX editor and uses the LLM only to generate queries, to rank and to explain how a paper supports the text, which agrees with the retrieval-grounded design of Reference Finder. ReCite first decides which positions need a citation, then generates queries from the inferred citation intent, and searches again when the candidates do not support the claim.

In my literature search (rechecked in November 2026), I found no system that takes a complete paper without its references as input, outputs a deduplicated bibliography of missing citations for the whole paper, and has a paper-level evaluation protocol. Individual design choices, such as grounding in retrieval and first identifying the positions that need a citation, have all appeared in concurrent work. Reference Finder differs mainly in the granularity of its input and output and in the way it is evaluated.

Evaluation in this field has mostly consisted of reranking existing citations offline. The real citations are treated as positives, and the question is where they rank in a fixed corpus. Several problems that an end-to-end system has to face do not arise in this setting. Identifying the statements that need a citation is itself error-prone, the candidates are limited by the coverage of the search sources, and a paper that the authors did not cite may still deserve a citation. The CiteME line of work brings in online search, and with it the problem of candidate coverage. The position of the citation is given, however, so identification errors and the aggregation of several citations remain outside its scope. The protocol in Chapter 6 counts the errors of identification and retrieval in its paper-level metrics. For the third problem, it restricts how the result is read: precision measures the ability to recover the authors' original citations and is only a lower bound on the quality of the suggestions (Section 6.3).

2.4 Human–AI collaboration: automation bias and LLM-as-judge

Human factors research has long studied how much of a task should be handed to a machine. Parasuraman and Riley [43] analyse the relation between people and automation under four headings: use, misuse, disuse and abuse. Parasuraman et al. [44] treat the level of automation as a continuum and set it separately for four stages: information acquisition, information analysis, decision selection and action implementation. The delegation ladder proposed here is close in spirit but uses a different basis for its levels. For verification tasks, how far a task can be delegated depends on whether the system's output can be adjudicated independently.

This field has repeatedly found that people trust automated output too much. The experiments of Skitka et al. [45] show that with a decision aid people make errors of omission and errors of commission, a pattern known as automation bias. Buçinca et al. [46] find that cognitive forcing interventions can reduce overreliance on AI suggestions, although users rate the experience less favourably. The value of checked delegation rests on the premise that a human really does verify each finding. The discussion of mandatory verification in Section 8.3 builds on this.

Much of the empirical work on delegation and automation bias comes from aviation, driving and clinical decision support. The relative costs of misses and false alarms in those fields are not the same as in scholarly verification. For an author, a false accusation is especially costly, for example when a genuine reference is labelled as fabricated or a correct derivation as an error. Hardly any work applies the conclusions of human factors research to scholarly verification. The criteria in Chapter 3 and the recommendations in Section 8.3 are a small step in that direction.

Zheng et al. [47] give a systematic study of the use of LLMs to judge LLM output. Known biases include position bias, a preference for verbose answers, and self-enhancement bias, where the judge favours output from its own source. The evaluation in Chapter 5 uses an LLM judge as one of two matching channels, so this work bears on it directly.

2.5 Positioning and gaps

Table 2.1 relates the four areas to the chapters of this thesis. Taken together, they leave three gaps that this thesis addresses.

Table 2.1: Positioning of related work

Area	Deciding existence	Content correctness	Completeness of citations	Delegation framework
Hallucinated references and citation integrity (Section 2.1)	Covered [8, 14]	Not covered	Not covered (only existing references are checked)	None
Automated reviewing and error detection (Section 2.2)	Not covered	Covered [4, 5]	Not covered	Only a position paper [7]
Citation recommendation (Section 2.3)	Not addressed	Not addressed	Partly covered: offline reranking, recovery of single citations and paragraph-level systems [24, 37, 42]	None
Human–AI collaboration (Section 2.4)	Not addressed	Not addressed	Not addressed	A notion of levels, set by stage of information processing [44]
This thesis	Chapter 7	Chapter 5	Chapter 6	Chapters 3 and 4

First, there is no delegation framework organised by task properties. Existing work reports capability measures such as detection rates but does not say what kind of deployment those measures support. Human factors research has the notion of levels, but it sets them by stage of information processing and cannot distinguish verification tasks by how far they can be adjudicated. Chapter 3 answers this with two criteria, decidability and ground-truth availability.

Second, almost all existing evaluations are offline experiments. The three systems in this thesis are deployed on a live platform, and their design requirements come from that deployment. Findings must be verified by a human, a false accusation is disproportionately costly, and the wording of suggestions is restricted (Chapters 5 to 7).

Third, there is no system that completes the references of a whole paper, and no evaluation protocol for one. Reference Finder and the strip-and-recover protocol address exactly this (Chapter 6).

3 The Delegation Framework

Chapter 2 concluded that the literature does not say which kinds of task can be delegated, or how far, which is the first thing RQ1 asks. This chapter gives the answer proposed in this thesis. Section 3.1 defines two criteria and Section 3.2 describes the three levels of delegation. Together they account for most of what “Methods” refers to in the title of the thesis.

3.1 Two criteria for delegation

A verification task can be written as $T = (X, Y, C)$. X is the input, for example a paper or a single reference. Y is the set of possible outputs, for example a verdict, a set of findings or a list of recommendations. C is the standard of correctness: it specifies which outputs count as correct but does not provide a way to test them. The two criteria below answer two questions. Can the system, while it runs, tell from C whether an output is right or wrong? And can reliable reference answers be obtained when the system is evaluated?

The first criterion is decidability, which concerns run time. The task T is said to be decidable by A if there is an adjudication procedure

$$A : X \times Y \to \{\text{accept}, \text{reject}, \text{abstain}\}$$

that satisfies three conditions. First, the verdict of A does not depend on a generative model or on case-by-case human judgement. The typical implementation is a query to an authoritative database plus deterministic matching rules. Second, when A outputs “accept”, the output is sufficiently likely to satisfy C, and the cases in which A may be wrong can be enumerated and communicated to users. Third, inputs that A cannot decide go to “abstain” and are not forced into right or wrong. The abstentions are what is left to people at run time, and the system draws no conclusion about them. Decidability has three grades. Fully decidable: the correct answer is unique and there is a deterministic adjudication procedure with broad coverage, for example whether a reference exists. Solvable but without an adjudicator: there is an objectively right answer, but no deterministic procedure can decide it at run time, for example whether a derivation is correct. Undecidable: there is not even a unique correct answer, for example which paper a sentence should cite.

The second criterion is ground-truth availability, which concerns evaluation. It has four grades, which differ in where the correct outputs come from. Exact ground truth comes from existing authoritative records. Injected ground truth is constructed by the evaluator: known errors are inserted into real papers, and the list of insertions is the ground truth. Recovered ground truth is taken from the evaluated object itself: if the references of a paper are removed, the authors’ original citations become the ground truth, although this ground truth is biased. Absent means there is no reliable way to obtain ground truth. The first three grades can be used for evaluation, and this thesis refers to them together as the three ground-truth mechanisms.

The two criteria answer different questions. Decidability is about whether a deployed system can itself give a verdict for a class of inputs. Ground-truth availability is about whether the evaluator has answers to compare against. The middle level of the ladder exists for the case where the two disagree. Faced with a new paper, no database can tell at run time whether one of the system’s findings holds, yet at evaluation time ground truth

can be constructed by injection. Such a system can be evaluated rigorously but cannot be authorised to draw conclusions on its own. The level of delegation has to satisfy both constraints, and when they disagree the more conservative level applies. Table 3.1 shows how the two criteria correspond to the three levels.

Table 3.1: The two criteria and the three levels of delegation

	L1 Full delegation	**L2 Checked delegation**	**L3 Advisory delegation**
Decidability	Fully decidable	Solvable but without an adjudicator	Undecidable
Ground-truth availability	exact	injected	recovered
How the output is phrased	As a statement (“this entry has been verified to exist”)	As a flag to be checked (“this may be an error, please check”)	As a suggestion (“this sentence could cite …”)
Position of the human	Handles only what the system abstains on	Verifies each item before acting	Keeps the full decision
Cost of errors	A wrong “verified” is worse than a missed verification, so thresholds must be conservative	False alarms cost verification time; misses are invisible	Poor suggestions waste attention; no risk of a wrong verdict
In this thesis	Reference Check (Chapter 7)	Correctness Check (Chapter 5)	Reference Finder (Chapter 6)

3.2 The three levels of delegation

L1 is full delegation. The system gives a verdict directly, and its output is phrased as a statement. This authorisation has one precondition: the verdict comes from a deterministic source. An LLM may take part in the process, but not in the key step from evidence to conclusion. When the system abstains, this must mean “could not be verified” and must not be an accusation, because a false accusation is costly. When L1 fails, the cause lies in the adjudication procedure itself. If the database does not hold a record, a real paper ends up as unverified. If the matching rules are too loose, a fabricated entry may slip through. Gaps of the first kind can be listed one by one and communicated to users in advance. L1-style decisions also occur inside the other two systems, where an LLM proposes candidates and a deterministic source decides. This thesis calls such a substep a local L1 structure. In Reference Finder, candidates must come from real search results, and already-cited entries are identified by deterministic matching. Both are structures of this kind. The novelty check in Correctness Check is only an incomplete local L1 structure: the search is deterministic, but whether the novelty is in doubt is still judged by an LLM.

L2 is checked delegation. The system acts as a prompter. It reads the whole paper, lists the suspicious passages, each with its location and a reason, and hands them to a human for verification. Its output cannot trigger any action directly, and every finding has to be confirmed by a human first. Whether this level of delegation is worthwhile depends on two measures: how many more problems the system finds than a cheaper baseline, and how much verification work its false alarms cause. The two must be reported together, because either one alone is easy to inflate. If every sentence is marked as suspicious, recall is 1 and the output is useless. If only the few most certain suspicions are reported,

precision can be very high while few problems are found. L2 can fail in two ways. Random failures can be reduced by running the system several times or by using several models. Systematic failures mean that a model cannot detect a certain kind of error at all, and adding more models of the same kind does not help (Section 5.5).

L3 is advisory delegation. The system draws no conclusion about correctness. Its output is no more than an open list of candidates, and the interface has to word them as suggestions. An evaluation at this level has to allow for the fact that the metrics are computed against biased ground truth. A paper the authors did not cite is not necessarily irrelevant, so precision computed against the authors' citations is only an approximate lower bound on true precision (Section 4.4). The risk at L3 lies mainly in the quality of the recommendations: the search sources may not cover a field, a statement may be identified wrongly, and reranking may put a relevant paper far down the list. These failures waste the user's attention but do not produce wrong verdicts. A system for a task without established ground truth can therefore still be deployed, provided the wording in the interface stays within the bounds of a suggestion.

The three levels are ordered by how much is authorised, not by capability. Further up the ladder a system is not necessarily simpler, but it may assert less and do less automatically.

The approach that runs through the thesis can be put in one sentence: LLMs as the language interface, deterministic sources as the adjudicator, and humans handling the residual. The levels are degrees of authorisation under this division of labour, and each system sits at one of them. Reference Check is at L1, since a database answers whether a reference exists and the LLM's part is limited to parsing. Correctness Check is at L2. With no adjudication procedure at run time, findings can only be proposed by an LLM. The system uses a pipeline to constrain the form of the output, for example by requiring the location of the evidence, filtering by severity and merging findings about the same problem. The confirmation of each finding is left to a human. Reference Finder is at L3, where correctness is not defined and only the most basic requirement of the approach remains, namely that every recommended paper must exist. Candidates can only come from real search results, an entry generated from nothing cannot enter the output, and whether a paper is relevant is left to the human in the form of a suggestion. The ladder can also be used to classify the subtasks inside a system (Section 8.1). Chapter 4 turns the two criteria and the three levels into a simple model.

4 A Mathematical Formulation of Delegation

Chapter 3 defined the delegation ladder in words. This chapter builds a mathematical model of the three systems on top of existing theory. It states the assumptions under which their design choices are reasonable and points out where the model and the implementation differ.

4.1 Basic setting

In terms of what a system is allowed to do, the three levels of delegation correspond to three kinds of action. An L1 system may decide directly, or abstain and hand the inputs it is unsure about to a human. An L2 system can only raise a set of flags, and a flag has consequences only after a human has verified it. An L3 system gives a ranked list of suggestions, and the human decides whether to take them. The analysis uses four cost terms: the cost c_{fa} of a wrong verdict (including a false accusation), the cost c_{rev} of a human verifying one flag, the cost c_{abs} of abstaining and handing over to a human, and the value v of finding one real error. In scholarly verification, c_{fa} is much larger than c_{rev}. These quantities cannot be measured precisely, and the conclusions below only use their relative size. Table 4.1 lists how the quantities in the model map onto the three systems. All values are taken from the code.

Table 4.1: Quantities in the model and their counterparts in the implementation

Quantity in the model	**Counterpart in the systems**	**Actual value**
Matching confidence and threshold (Section 4.2)	Title similarity from the fuzzy matcher of Reference Check, which the other two systems reuse	token set > 90 and token sort > 80 when the authors agree; otherwise > 95 and > 90
Scoring of flags (Section 4.3)	Severity score in stage 5 of the L2 pipeline	Integer from 0 to 3, and 0 is filtered out; excerpts from claim verification get 1 extra point
Verification budget k (Section 4.3)	The k highest-scoring excerpts used in evaluation	$k = 10$
Hit criterion (Section 4.3)	Word-level edit-distance similarity, or an LLM judge	Similarity threshold 0.5
Threshold for statement identification (Section 4.4)	Confidence filter and per-section budget in step S1 of the L3 pipeline	Confidence at least 0.6; at most 8 statements per section, 16 for citation-dense sections
Candidate pool and relevance (Section 4.4)	Candidate pool of step S2 and listwise scoring of step S3 in the L3 pipeline	At most 30 papers per statement; integer from 0 to 10, top 5 kept, top 10 in evaluation
Matching against ground truth (Section 4.4)	Deciding in evaluation whether a recommendation is one of the authors' citations	Same DOI or arXiv identifier; otherwise title token set at least 95, years at most 1 apart, and the same first-author surname

4.2 Full delegation: matching thresholds and abstention

The core decision at L1 is whether two bibliographic records refer to the same work. Reference Check relies on this test to decide whether an entry found in a database is the reference being checked (Chapter 7). Reference Finder applies it when merging duplicate candidates and when deciding whether a suggestion is already cited by the authors, and Correctness Check uses it to exclude the paper under review itself. All of these use the same fuzzy matcher. It first normalises the titles and then computes two similarity scores between 0 and 100. The token set ratio ignores word order and repeated words, and the token sort ratio ignores only word order. When the authors of the entry agree with those of the candidate, the title only needs token set > 90 and token sort > 80. Agreement means that the first author appears among the first three authors of the candidate, or that at least 60% of the authors can be found in the candidate. When the authors do not agree or no author information is available, the title needs token set > 95 and token sort > 90, and a further rule handles the case where only the main title matches. If none of these conditions holds, the two records are treated as different works.

Let η be the system's confidence that a given decision is correct.

Proposition 4.1 (Abstention threshold)**.** The rule with the lowest expected cost is to decide only when η reaches a threshold θ^* and to abstain otherwise, where

$$\theta^* = 1 - \frac{c_{\text{abs}}}{c_{\text{fa}}} \qquad (\text{requiring } c_{\text{abs}} < c_{\text{fa}}).$$

Proof sketch. The expected cost of deciding is $(1-\eta)\cdot c_{\text{fa}}$, and the cost of abstaining is c_{abs}. The former does not exceed the latter if and only if $\eta \geq 1 - c_{\text{abs}}/c_{\text{fa}}$. This is the reject rule of Chow [48]; see also El-Yaniv and Wiener [49]. □

The more expensive a wrong verdict, the closer the threshold is to 1, and the matcher's two sets of thresholds are consistent with this. Agreement of the authors is one more piece of evidence, so the title threshold can be lower. These thresholds are engineering constants, however, and the similarity scores are not calibrated probabilities. The proposition does not yield values such as 90 or 95. It also assumes that someone really does take over the inputs on which the system abstains (Assumption A5). One implication is that a conservative threshold is only reasonable when abstaining is cheap. If the interface presents "could not be verified" as "suspected fabrication", the cost of abstaining rises, the optimal threshold falls, and the system is forced to decide when it is not confident enough. In one place the implementation departs from the model. The novelty check of Correctness Check records a pass when the search returns no papers at all. By Proposition 4.1, abstaining would be the more appropriate output here.

4.3 Checked delegation: scoring, truncation and evaluation

In stage 5 of Correctness Check, an LLM gives each candidate error a severity score from 0 to 3, and candidates with a score of 0 are filtered out. In the evaluation, the excerpts are sorted by score in descending order, and the top $k = 10$ are compared with the ground truth. The filter threshold and k determine how much the system reports. Suppose a paper has $\bar{m}$ real errors on average, a share Prec of the flags are real errors, a share Recall of the real errors are flagged, and a human verifies each flag and judges it essentially correctly (Assumption A1). The net benefit of deploying the prompter is then

$$U = \bar{m} \cdot \text{Recall} \cdot \left(v - \frac{c_{\text{rev}}}{\text{Prec}} \right).$$

Proposition 4.2 (Viability criterion for checked delegation)**.** As long as Recall > 0, the net benefit is positive if and only if Prec $> c_{\text{rev}}/v$. For a given Prec, the net benefit is proportional to Recall.

Precision decides whether a prompter is worth using, and recall decides how useful it is, so the two metrics should be reported together. In the implementation, Prec cannot be measured under injected ground truth, because a flag on an error that was already in the paper is counted as wrong. The accuracy@10 used in Chapter 5 is the recall of the top 10 flags on the injected errors. It carries no information about precision, and on its own it cannot tell whether the net benefit is positive or negative.

The current pipeline uses only one model per run. The analysis below considers an extension in which two models each check the paper and their flags are merged. Group the detection models and the models that inject the errors by family, and write μ for the miss rate of a detection model on errors injected by a given family.

Definition 4.1 (Random and systematic failure)**.** If the miss rate of a detection model does not depend on the family of the injecting model, its failures are called random. If the miss rate varies systematically with that family, they are called systematic. Same-family elevation means that each detection model has a miss rate μ_{hi} on errors injected by its own family and μ_{lo} on errors injected by the other family, with $\mu_{\text{hi}} > \mu_{\text{lo}}$, symmetrically for the two families.

Proposition 4.3 (Cross-family pairing beats same-family redundancy)**.** Suppose the flags of two detection models are merged by union, and on errors injected by the same family the two models miss independently of each other (Assumption A2). Under symmetric same-family elevation, the worst-case joint miss rate is $\mu_{\text{hi}} \cdot \mu_{\text{lo}}$ for a cross-family pair and μ_{hi}^2 for a same-family pair. When each family injects half of the errors, the joint miss rate is $\mu_{\text{hi}} \cdot \mu_{\text{lo}}$ for a cross-family pair and $(\mu_{\text{hi}}^2 + \mu_{\text{lo}}^2)/2$ for a same-family pair, which is not smaller.

Proof sketch. Under independence the joint miss rate is the product of the two miss rates. The comparison then follows from the inequality of arithmetic and geometric means. □

The proposition says that the more the failures follow the injecting family, the more is gained by checking with a model from another family. Neither of its two premises was supported in the preliminary experiment: the two families are not symmetric, and there is no evidence of same-family elevation (Section 5.5). Whether cross-family pairing really is better can only be tested by running two models jointly.

The evaluation has to decide whether a flag hits the injected error. The evaluation script uses two channels. The string channel computes the word-level edit-distance similarity between the flagged excerpt and the ground-truth excerpt, and a value of at least 0.5 counts as a hit. The judge channel gives the top 10 flags and the ground truth to an LLM, which judges them one by one. If either channel reports a hit, the paper counts as a hit.

Proposition 4.4 (Union matching is biased upwards)**.** When a hit in either channel counts as a hit, the combined hit rate is at least as high as that of each single channel, and so is the share of misses wrongly counted as hits.

The union is taken to tolerate paraphrase, and the price is that any bias in the results is upwards. In the preliminary experiment, every paper counted as a hit by the string

channel was also counted as a hit by the judge channel. The numbers in Chapter 5 therefore depend in practice on the judge.

4.4 Advisory delegation: ranking and evaluation metrics

For each statement that needs a citation, Reference Finder first searches, merges and deduplicates the results, and keeps at most 30 papers as the candidate pool. It then passes the whole pool to a single LLM call, which gives each paper a relevance score r from 0 to 10, and keeps the top 5 by r. When the results are aggregated across statements, the global score of a paper d is

$$s(d) = \max_i r_i(d) \cdot \ln\big(1 + |S(d)|\big),$$

where $S(d)$ is the set of statements that d supports.

Proposition 4.5 (Probability ranking principle)**.** If the relevance of the candidates is mutually independent, the estimates of the probability of relevance are accurate, and the gain accumulates additively with weights that decrease with rank (Assumption A6), then ranking by probability of relevance in descending order and taking the top k maximises the expected gain [50].

Ranking by r and truncating applies this principle with r standing in for the probability of relevance. The first gap is that the score r is an integer grade given by an LLM, not a calibrated probability. The second is that the pool limit is applied before reranking, in a fixed order of the search sources, so papers returned by a source later in the order may not make it into the pool even if they are more relevant.

In the evaluation, the references of a paper are removed, and the question is how many of the originally cited works the system can find again. Let P be the global bibliography output by the system, G the set of works the authors actually cited, and R the set of relevant works. The two main metrics are

$$\text{Precision} = \frac{|P \cap G|}{|P|}, \qquad \text{Recall} = \frac{|P \cap G|}{|G|}.$$

R cannot be observed, so G has to stand in for it.

Assumption A3 (Cited works are relevant). Every work the authors cite is relevant to the paper, that is, $G \subseteq R$.

Proposition 4.6 (Measured precision is a lower bound)**.** Under A3, $|P \cap G|/|P| \leq |P \cap R|/|P|$. If every relevant work is cited by the authors with the same probability ρ, and whether it is cited is independent of what the system recommends (Assumption A7), then the measured precision equals, on average, ρ times the true precision.

When the system recommends a work that is relevant but that the authors did not cite, it is counted as an error. Measured precision is therefore an approximate lower bound on the quality of the suggestions. A7 does not hold automatically either. If the model saw the paper during training, it may reproduce the authors' citations from memory.

Proposition 4.7 (Decomposition of recall)**.** Recall decomposes into three factors:

$$\text{Recall} = \Pr(\text{retrievable}) \cdot \Pr(\text{in pool} \mid \text{retrievable}) \cdot \Pr(\text{selected} \mid \text{in pool}).$$

The three factors depend on the coverage of the search sources, on statement identification and the pool limit, and on reranking, respectively.

4.5 What the three ground-truth mechanisms allow one to conclude

What an evaluation number can show depends on where the ground truth comes from. Exact ground truth comes from authoritative records. Within the coverage of the database, whether each output is right or wrong can be looked up directly. Injected ground truth holds only for the injected errors.

Lemma 4.1 (Transfer bound)**.** Suppose the loss on each instance lies between 0 and M (Assumption A4). Then the difference between the system's risk on natural errors and its risk on injected errors is at most M times the total variation distance between the two error distributions [51].

How far the two distributions differ cannot be measured, so a number obtained under injected ground truth can only be stated as "the detection rate on this kind of injected error". Injected data are also stratified by injecting family. The detection rate can be computed for each stratum separately, but the weight each stratum should have in a real setting is unknown, so results should be reported by stratum. Under recovered ground truth, precision is only a lower bound (Proposition 4.6).

Table 4.2: Ground-truth mechanisms and the conclusions they allow

Mechanism	What can be measured	What cannot be measured	Conclusion that can be drawn
exact	Detection rate within coverage	Inputs outside coverage	"The detection rate within coverage is X%", together with the coverage
injected	Detection rate on injected errors, which can be stratified	The gap to natural errors; the weights of the strata	"The detection rate on this kind of injected error is X%", reported by stratum
recovered	Recall relative to the authors' citations; a lower bound on precision	The authors' citation probability ρ; whether an uncited recommendation is relevant	"The recovery rate is X%, and an approximate lower bound on the quality of the suggestions is Y%"

The evaluation metrics in the following chapters are chosen according to this table: a metric should be a quantity that can be measured under the ground-truth mechanism at hand.

4.6 Assumptions and limitations

- **Assumption A1 (Accurate verification).** A human's judgement when verifying a flag is essentially correct. Used in Proposition 4.2.
- **Assumption A2 (Independent misses).** On errors injected by the same family, two detection models miss independently of each other. Used in Proposition 4.3.

- **Assumption A3 (Cited works are relevant).** Used in Proposition 4.6.
- **Assumption A4 (Bounded loss).** Used in Lemma 4.1.
- **Assumption A5 (Reliable confidence, and abstentions are taken over).** Used in Proposition 4.1.
- **Assumption A6 (Independent candidates, additive gain).** Used in Proposition 4.5.
- **Assumption A7 (Citation independent of recommendation).** Used in the second half of Proposition 4.6. It is a common assumption in positive-unlabelled learning [52].

There are two main gaps between the propositions and the implementation. First, scores are not probabilities. Title similarity, severity and relevance are engineering scores or integer grades given by an LLM, so the propositions can explain the direction of a design choice but cannot fix a specific threshold. Second, the propositions only use the relative size of the costs, and the actual values of c_{fa}, c_{rev} and v vary with the setting. In addition, Proposition 4.3 analyses an extension, and the same-family elevation it assumes was not supported in the preliminary experiment (Section 5.5).

All conclusions in this chapter come from existing theory. Proposition 4.1 comes from classification with a reject option [48] and selective prediction [49]. Learning to defer [53, 54] and algorithmic triage [55] study when a system should hand an input to a human. They treat the handover as a binary choice, and this chapter extends it to three levels with decreasing authorisation. Section 4.4 uses the probability ranking principle from information retrieval, together with methods for evaluation when relevance judgements are incomplete [50, 56]. Under recovered ground truth, precision can be bounded only from below, which is a problem of partial identification [57]. The transfer bound comes from the theory of domain adaptation [51]. The levels of automation introduced in Section 2.4 [44] are set by stage of information processing, whereas the levels in this chapter are set by what a system is allowed to do. The two can be used side by side.

5 L2 Checked Delegation: Correctness Check

This chapter presents the first system of the thesis, Correctness Check, which corresponds to direction 2 of the project plan. It sits at the middle level of the ladder: there is a definite answer to whether the derivations and numbers in a paper are correct, but no program can decide it automatically at run time. The system can therefore only flag possible errors, which a human then has to verify. Sections 5.1 and 5.2 describe the task and the design of the pipeline (RQ1), Sections 5.3 to 5.7 cover the experiments and an external evaluation (RQ2), and Section 5.8 sums up (RQ3).

5.1 Task definition

The input is the PDF of a paper, and the output is a report in two parts. The first part, Claim Validity and Evidence, lists the core claims of the paper and the result of checking each one: whether the evidence is sufficient, whether the novelty is in doubt, and where the gaps are. A claim with sufficient evidence and no doubt about its novelty gets a single line with the result of the check. A claim with insufficient evidence is written up in three parts: the result of the check, the gap in the evidence and a recommended fix. For a claim with sufficient evidence whose novelty is in doubt, possible prior work is listed as well. The second part, Findings Beyond the Claims, lists objective errors found in the full text outside the claims. Each group of findings carries a confidence label.

Under the criterion of Section 3.1, this task is “solvable but without an adjudicator”. A derivation either holds or does not, and a number either agrees with the source it cites or does not. No database holds the answer, however, so the judgement has to rest on understanding and reasoning. The system’s output is therefore no more than a set of leads to be verified. The question for this chapter is how much better the system does as a prompter than the cheaper alternative, a single LLM prompt (the raw LLM baseline), and how it fails.

5.2 The six-stage pipeline

The pipeline has six stages (Figure 5.1). The result of each stage is written to disk, so an interrupted run can continue from the last completed stage.

Stage 1 extracts one to five core claims from the full text, following the extraction method of Claimify [58]. Only sentences with a clear meaning are extracted, and only the context needed to understand a claim is added. Claims are extracted before checking so that verification centres on what the paper itself asserts. Stage 2 verifies the claims one by one and judges three things at once: whether the claim or its argument contains an objective error, whether the evidence in the paper is enough to support it, and whether it asserts a new contribution. Stage 3 only handles claims that assert a new contribution. An LLM generates search queries, the system runs an external search and takes at most 15 results, and an LLM then judges against these results whether the claimed “first” conflicts with existing literature. When the search returns nothing, the system records the claim as passed. When the check of a claim fails midway, the claim is skipped and the report does not mention it. The search was meant to be cut off at the paper’s publication date, but no date is currently passed to it, so this filter is not in effect.

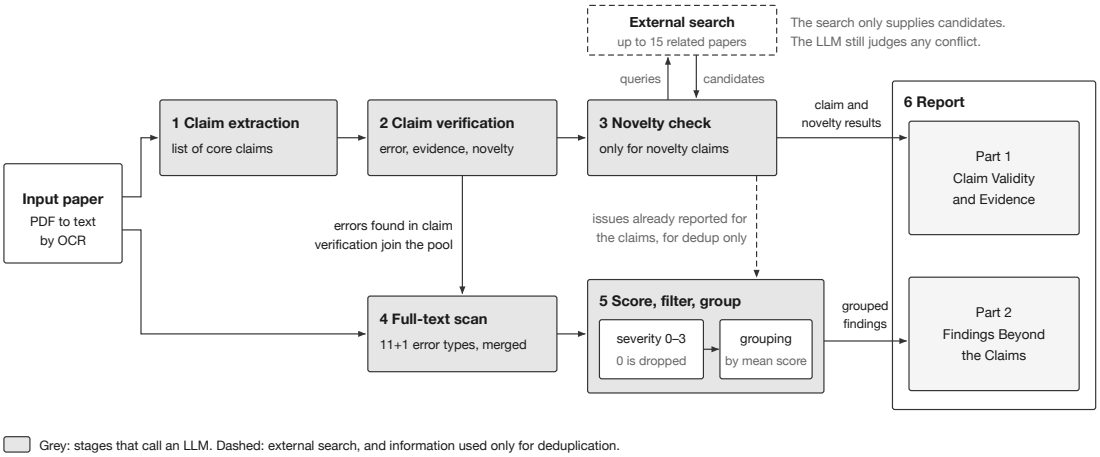

Figure 5.1: The six-stage pipeline of Correctness Check. The top row checks the claims, and the bottom row scans the full text. Stages with a grey background call an LLM.

Stage 4 scans the full text independently, with its prompt organised around a taxonomy of 11+1 types of objective error. The results are merged with the errors found in stage 2 to form the candidates. The scan starts from the full text and stage 2 starts from the claims, so the two sources complement each other. The taxonomy is the system's own, and subjective matters such as writing quality are explicitly excluded. It is not the taxonomy of the evaluation benchmark: FLAWS does not restrict the type when injecting an error, and the injecting model assigns it to one of six categories afterwards. Stage 5 first gives each candidate a severity score from 0 to 3. A score of 0 means a false alarm or a duplicate of the first part, and such candidates are filtered out. The stage then merges excerpts that describe the same problem into groups, ranks the groups by mean score, and converts the score into one of three labels: possible or minor, major, and critical. Stage 6 calls no LLM and only assembles the earlier results into the report.

The pipeline is split into stages to separate proposing candidates from filtering them. The filtering step is there to curb the overconfidence of a single LLM pass.

5.3 Experimental setup

The data come from FLAWS (Section 2.2). The benchmark contains papers with errors injected by an OpenAI model and papers with errors injected by a Gemini model.

For each paper the system outputs a set of error excerpts with severity scores. The evaluation script takes the 10 highest-scoring excerpts and compares them with the injected ground truth of that paper. There are two matching channels. The string channel requires a word-level edit-distance similarity of at least 0.5 between the predicted excerpt and the ground-truth text. In the judge channel, an LLM decides for each excerpt whether the two refer to the same place in the original text. The main metric is accuracy@10: a paper counts as a hit if at least one of its top 10 excerpts is matched by either channel. The original FLAWS evaluation works the same way. Because the two channels are combined by union, the result can only be biased upwards (Proposition 4.4).

The judge has two configurations. The original runs of the preliminary experiment follow FLAWS and let the model that injected the error act as judge. This thesis calls it the dynamic judge, and the main numbers in Table 5.1 come from this configuration. The other configuration uses gpt-5.2 as judge throughout, so that the results on the two groups

of papers can be compared directly. Results under this configuration are available for all four models as well. Each configuration has a weakness. The dynamic judge may be stricter for one family of papers than for the other, and the unified judge itself belongs to the OpenAI family.

5.4 Results of the preliminary experiment

Table 5.1 shows the preliminary results: 20 papers, the dynamic judge, and one run per model.

Table 5.1: Results of the preliminary experiment (20-paper subset)

Detection model	OpenAI-injected (n=8)	Gemini-injected (n=12)	Combined (n=20)	95% interval (combined)	Mean number of predicted excerpts
gpt-5	12.5% (1/8)	**58.3% (7/12)**	**40.0% (8/20)**	[20.0%, 60.0%]	7.85
gpt-5.2	0.0% (0/8)	41.7% (5/12)	25.0% (5/20)	[10.0%, 45.0%]	8.55
gemini-2.5-pro	**50.0% (4/8)**	25.0% (3/12)	35.0% (7/20)	[15.0%, 55.0%]	7.35
gemini-3.1-pro-preview	25.0% (2/8)	33.3% (4/12)	30.0% (6/20)	[10.0%, 50.0%]	6.55

Note: The metric is accuracy@10. The intervals come from a bootstrap that resamples papers (10,000 resamples). Under the unified gpt-5.2 judge, the combined results of the four models are 35.0%, 30.0%, 35.0% and 25.0%, in the same order.

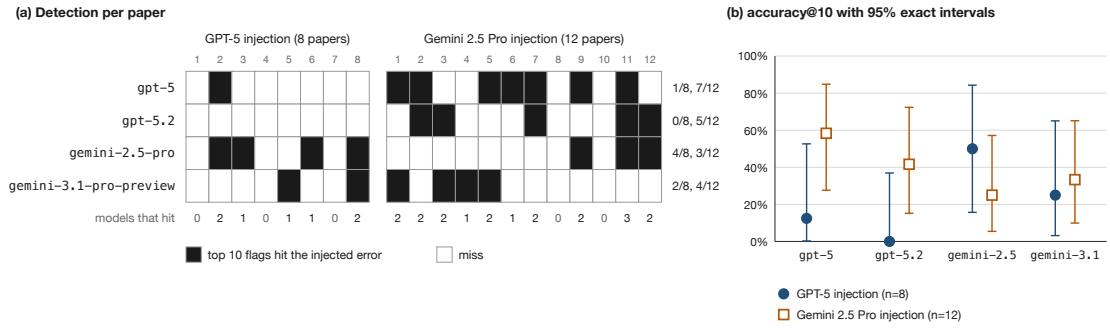

Figure 5.2: Per-paper results of the preliminary experiment (dynamic judge). (a) Whether each detection model hit the injected error on each of the 20 papers. A black cell is a hit, and the bottom row gives the number of models that hit the paper. (b) accuracy@10 by injecting family, with 95% Clopper–Pearson intervals.

The combined accuracy@10 of the four models lies between 25% and 40%. The FLAWS authors report raw single-prompt baselines on all 713 pairs: 39.1% for GPT-5 and 19.8% for Gemini 2.5 Pro [4]. For these two models, which can be compared directly, the pipeline scores above the published baseline. With gpt-5 it reaches 40.0% against the published 39.1%. With gemini-2.5-pro it reaches 35.0%, which is 15.2 percentage points above the published 19.8%. The intervals on this sample are wide (Table 5.1), and the advantage still needs to be confirmed in a larger experiment. Following this result, I optimised the system, which then went live on CSPaper. Researchers at Carnegie Mellon University

(CMU) evaluated it independently (Section 5.7).

5.5 Results by injecting family

When Table 5.1 is split by injecting family, the two groups differ clearly. With gpt-5, the detection rate is 58.3% on errors injected by Gemini but only 12.5% on errors injected by its own family, OpenAI. For gpt-5.2 the figures are 41.7% and 0.0%. The model gemini-2.5-pro is also better on errors injected by the other family, with 50.0% against 25.0%. The exception is gemini-3.1-pro-preview. These are differences in point estimates only, however, and each group has just 8 or 12 papers. A Fisher exact test on the two detection rates of each model gives p-values of 0.070, 0.055, 0.356 and 1.000, none of them below 0.05.

My first explanation for this pattern was a same-family effect: when the detection model and the injecting model belong to the same family, the injected errors are less likely to raise suspicion. The data in the FLAWS paper point to a simpler explanation. The paper reports accuracy@10 for five raw models separately by injecting model. On the 265 pairs injected by GPT-5 the values range from 13.6% to 30.9%, and on the 448 pairs injected by Gemini 2.5 Pro from 23.4% to 44.0%. Every model is 6.7 to 13.1 percentage points lower on the first group, whatever family the detection model belongs to. In other words, errors injected by GPT-5 are harder for all models. The low detection rates of the two OpenAI detection models on the OpenAI-injected group can be explained by the difficulty of the injected errors alone. The two Gemini detection models are not lower on the OpenAI-injected group (6/16 against 7/24). This runs against the direction seen in the FLAWS paper and looks more like small-sample fluctuation.

The asymmetry observed in the preliminary experiment therefore cannot be taken as evidence of a same-family effect. What the data do show is that the detection rate depends on which model injected the errors, a difference that the combined numbers hide. Separating a same-family effect from the difficulty of the injected errors would need a detection model from a third family as a control.

If a same-family effect exists, the remedy is to check with a model from another family. I call this cross-family verification, and Proposition 4.3 gives the conditions under which it helps. From the existing runs, I computed retrospectively how many errors a pair of models would miss together. On average there is almost no difference between same-family and cross-family pairs, with mean joint miss rates of 47.9% and 47.4%, and which pair is best changes with the judge configuration. These data neither support nor refute Proposition 4.3. Cross-family verification remains an idea to be tested.

5.6 Report volume and over-reporting

In the preliminary experiment, the four models reported between 6.55 and 8.55 error excerpts per paper on average. Reports beyond the injected error are not necessarily false alarms, because the paper may contain errors of its own. Bianchi et al. [5] found an average of 5.2 to 5.9 objective errors per paper in top-conference papers from 2025. One report produced by the system does contain a false alarm. The system claimed that the year and the DOI of a Nature citation in the paper did not agree, although the citation was correct. Under checked delegation, over-reporting does not lead to wrong verdicts, but it directly increases the verification workload.

5.7 External evaluation: TMLR's selection of an AI reviewer

In 2026 the journal TMLR planned a pilot of AI reviewing, in which every submission receives one AI-generated review in addition to the human reviews. To choose a system,

Rao and Shah [22] compared four candidates: the AI reviewer of CSPaper, ReviewerToo, and two general-purpose models, Claude Opus 4.7 and GPT-5.5. Shah is one of TMLR's Editors-in-Chief, and both authors are also authors of FLAWS. The evaluation only considers TMLR's first acceptance criterion: whether the claims of the paper are supported by accurate, convincing and clear evidence. TMLR's other acceptance criterion, whether the paper has an audience at the journal, is outside the scope of the evaluation.

The evaluators called the system provided by CSPaper through an API. This system is a review agent that CSPaper built to TMLR's review criteria and format, with Correctness Check (Section 5.2) at its core. A conflict of interest should be stated here: I am the main designer and developer of Correctness Check. The design, the runs and the annotation of the evaluation were all done by the evaluators. This section only relays the numbers in their report and does not compare them with the accuracy@10 measured in this chapter.

The evaluation uses two kinds of data. The first consists of real TMLR submissions, divided into three groups by the final judgement of the Action Editor (AE): accepted papers, rejected papers that failed only on claims and evidence, and rejected papers that failed both criteria. The ground truth is the AE's judgement on this criterion. The second consists of newly constructed papers with injected errors, built with the FLAWS method from NeurIPS 2025 papers. Their ground truth is always "no".

Table 5.2: Share of "yes" answers on claims and evidence in the TMLR selection evaluation [22]

System	Accepted (should be yes)	Failed only claims and evidence (should be no)	Failed both criteria (should be no)	Gemini-injected (should be no)	GPT-5-injected (should be no)
CSPaper	92.7% (n=41)	9.1% (n=44)	0.0% (n=46)	82.1% (n=67)	70.0% (n=50)
GPT-5.5	82.9% (n=41)	65.9% (n=44)	32.6% (n=46)	76.1% (n=67)	80.0% (n=50)
Claude Opus 4.7	82.9% (n=41)	34.1% (n=44)	23.9% (n=46)	86.6% (n=67)	94.0% (n=50)

Note: Higher is better in the first column, and lower is better in the others. ReviewerToo was only partly evaluated because its runs were unstable, and it is not listed.

On real submissions, CSPaper agrees best with the AEs. On the papers with injected errors, none of the three systems does well, and CSPaper still answers "yes" for 82.1% and 70.0% of the two groups. The evaluators annotated the Gemini-injected group by hand and read the results under two criteria. The strict criterion requires the system to answer "no" and to describe the injected error correctly. CSPaper does so for only 3 of the 67 papers, or 4.5%. GPT-5.5 reaches 10.4% and Opus 4.7 reaches 1.5%. The lenient criterion requires only that the review draws the reader's attention to the location of the error. Among the sampled and annotated reviews that answered "yes", 70.0% of those from CSPaper mentioned the injected error, against 40.0% for GPT-5.5 and 25.0% for Opus 4.7.

TMLR then announced that it would adopt CSPaper's AI reviewer [59]. According to the announcement, the AI review only assesses soundness and gives no recommendation to accept or reject. It is provided to the AE together with the human reviews. The AE

makes the final decision, and the AI review is not binding. Under the strict criterion, the detection rates of all three systems are very low, and no system can draw conclusions on its own. This agrees with Chapter 3, which places tasks of this kind at the level of checked delegation. The value of the system is seen mainly under the lenient criterion, that is, in its role as a prompter. TMLR's use of the system is itself a case of checked delegation, in which the AI gives an opinion and the decision rests with the AE.

5.8 Summary

For now, Correctness Check can only be used as a prompter. A system at this level should first be judged by its gain in recall over the raw baseline (Proposition 4.2). For this the chapter has only indirect evidence, namely that the pipeline scored above the published raw baseline in the preliminary experiment.

6 L3 Advisory Delegation: Reference Completion with Reference Finder

Reference Finder, the second system of the thesis, addresses direction 3 of the project plan: completing the citations a paper is missing. Its place is the third level of the ladder, because there is no unique correct answer to which paper a sentence should cite. The system can only make suggestions, and the author decides whether to take them. The task and the system design come first (Sections 6.1 and 6.2, RQ1), then the evaluation protocol and the test results (Sections 6.3 and 6.4, RQ2), and finally a summary (Section 6.5, RQ3).

6.1 Task definition

The system takes a paper as input, finds the statements in the body that need a citation, searches scholarly databases, and recommends existing papers for each statement. Papers the authors already cite are listed separately and are not recommended again. The system complements Reference Check (Chapter 7). Reference Check asks whether the existing references are real, whereas Reference Finder looks for what should have been cited and was not [60].

The system only recommends papers and links them to statements, and its search relies on online APIs. Training or fine-tuning a model, classifying citation intent, writing citation sentences and building a local index of the whole literature are all outside its scope.

There is no objective standard for whether a sentence needs a citation, and there is more than one paper it could cite, so this task can only be handled by advisory delegation. The system is retrieval-grounded. The LLM only scores and ranks the papers returned by the search and never writes a citation itself, so every recommendation is a real paper with a link that can be opened. This rules out only one kind of error, namely recommending a paper that does not exist. It does not guarantee that a recommended paper is relevant to the statement. The relevance score is only a judgement about the abstract, and the author still has to read the paper before citing it.

6.2 System design

The pipeline has four steps (Figure 6.1), and only S1 and S3 call an LLM.

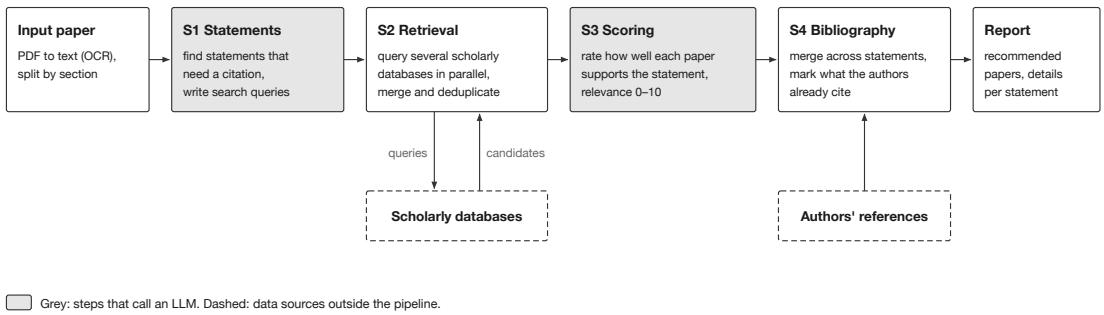

Figure 6.1: The pipeline of Reference Finder. Steps with a grey background call an LLM, and dashed boxes are data sources outside the pipeline.

S1 finds the statements that need a citation, such as a result the paper compares against, a method or dataset it names, or a fact about the state of the field. It also generates search queries for each statement. By default it only handles statements that have no citation yet. S2 sends the queries in parallel to nine scholarly databases and search services, then merges and deduplicates the results into a candidate pool. In S3 an LLM reads the title and abstract of each candidate next to the statement and gives a relevance score from 0 to 10. S4 merges the recommendations for all statements into one suggested bibliography and compares it with the authors' existing references. Papers that are already cited are listed separately.

S1 calls the LLM once for each section. It finds the statements that need a citation and gives, for each of them, a reason, a citation type, a confidence value and two or three search queries. There are five citation types: background, method, comparison, data or resource, and support for a claim. The prompt requires each statement to be a verbatim excerpt from the body, and the system checks whether it really occurs there. To keep the model from marking too much, the prompt lists cases that should not be marked, such as common knowledge, the paper's own contributions, purely mathematical derivations and transitional sentences. It also asks the model to decide the type first and then to say why external support is needed. Statements below the confidence threshold are dropped, and each section has an upper limit on the number of statements (Table 4.1).

S2 searches the sources concurrently for each statement. Lexical engines that only accept keywords receive the keyword queries, a semantic index receives the statement itself, and sources that can handle both receive both. The interval between requests is set separately for each source, in most cases following its official terms or documentation. When a single source fails, the system skips it and carries on. The candidates returned by the sources are first deduplicated by exact match of the DOI or arXiv identifier, and then by fuzzy matching with the matcher that is reused from Reference Check (Section 4.2). At most 30 papers are kept as the candidate pool.

S3 calls the LLM once for each statement and scores all candidates in the pool together. This is known as listwise reranking, and it makes the scores directly comparable. Each candidate is presented with its title, the first 150 words of its abstract, its year, its venue and its citation count. The model gives each candidate a relevance score from 0 to 10, judges whether citing it is essential, and writes one sentence of justification.

S4 aggregates across statements into a global bibliography. A paper that is recommended for several statements is merged into one entry and given a global score (Section 4.4). The step then takes the authors' existing reference list and matches it against the suggested entries by deterministic fuzzy matching. Entries that match are marked as already cited and placed after all new suggestions. On the engineering side, the output format of each structured LLM call is constrained by a data model, and a call is retried automatically when validation fails. The result of every step is written to disk, so an interrupted run can continue.

Reference Finder went live on CSPaper in November 2026. On the platform it runs after a paper has been reviewed and uses the same paper text as the review, with the PDF converted to text by OCR. Papers scoring below 7 are not recommended. The results come in three groups: recommended (a score of at least 8 for at least one statement), possibly related (a score of 7), and already in the references. Users can also view the results by statement: why each statement needs a citation, which papers support it, and which search queries were used. Recommended papers can be exported as BibTeX. The platform's review reports contain a section listing related work that may be missing,

and Reference Finder checks each item in it. An item that is found in the databases and scores at least 7 for some statement joins the recommendations. One that is found but does not directly support any statement is marked as Verified. One with no matching record is marked as Couldn't verify and is left for the author to look up. According to the launch page, a run usually takes 4 to 7 minutes [60].

The tests in Section 6.4 use the research implementation of this pipeline, whose parameters are not exactly those of the platform version. The research implementation connects to eight search sources, keeps the 5 most relevant papers per statement, and does not filter at the threshold of 7 (Table 4.1).

6.3 Evaluation protocol

Reference completion has no established ground truth. This thesis uses a strip-and-recover approach: take papers published at top conferences, delete all citation markers and the reference list, give the remaining text to the system, and see how many of its recommendations are works the authors originally cited. The main metrics are precision and recall over the output for the whole paper (Section 4.4): the share of recommendations that are among the authors' citations, and the share of the authors' citations that are recovered. A work the authors did not cite may still deserve to be cited, so precision measured this way is only a lower bound on the quality of the suggestions (Proposition 4.6).

6.4 Test results

The research implementation completed four full runs on three machine learning papers, one of which was run twice (Table 6.1). The three papers are one arXiv preprint and two anonymous submissions, and the input was always the original text of the paper.

Table 6.1: Runs of the Reference Finder research implementation on three papers

Run	Paper	Statements needing a citation	Suggested bibliography	Of these, relevance at least 7	Of these, already in the authors' references	Time
1	A	10	46	18	1	9 min
2	B	20	78	37	9	21 min
3	C	5	26	15	7	4 min
4	C	11	55	31	5	5 min

Note: All four runs used the same general-purpose LLM. The suggested bibliography is the number of papers left after the recommendations for all statements are merged and deduplicated, and the two columns starting with "Of these" are parts of it. Relevance is scored by the LLM in S3. Time runs from uploading the paper to the finished report.

Together the four runs identified 46 statements that need a citation, and for 40 of them at least one paper with a relevance score of 7 or higher was found. The suggested bibliographies contain 205 entries in total. Every entry has a DOI, an arXiv identifier or a link, so it can be looked up and checked. Of these entries, 22 match the authors' existing references and are listed separately. S1 is done by an LLM, and the number of statements it identifies also depends on the confidence threshold and the per-section budget, which is why the two runs on paper C gave different results.

For the platform version, the launch page reports one run [60]. On an EMNLP 2025 paper, the system found 19 statements that need a citation, recommended 25 papers, and

marked one as already in the authors' references. Of the six papers that the review report suggested adding, two were confirmed to exist but did not directly support any statement in the paper, and were marked as Verified. For the other four, the author information did not match the database record. They were marked as Couldn't verify and left for the author to check.

6.5 Summary

All of Reference Finder's candidates come from actual search results. The system only makes suggestions, and the author decides whether to take them, so it can be deployed even without established ground truth. In the tests on three papers, the system found a paper with a relevance score of at least 7 for 40 of the 46 statements, and every recommended paper can be looked up and checked (Section 6.4). Statements the system does not identify and papers the search does not find remain invisible to the author, who is still responsible for that part.

Two kinds of problems can be stated in advance: the databases cover non-English literature and books less well, and external search services may be temporarily unavailable. In both cases fewer papers are found, but no citation is fabricated.

In an evaluation, precision and recall have to be reported together, because either one alone can be raised by adjusting how much the system outputs.

7 L1 Full Delegation: Reference Check

The third system, Reference Check (product name: Reference Check Agent), corresponds to direction 1 of the project plan and sits at the lowest level of the ladder. The existence of a reference can be settled by a lookup in scholarly databases, so the system can give the verdict itself and hand back to a human what it cannot decide. Sections 7.1 and 7.2 relate to RQ1, Section 7.3 to RQ2, and Section 7.4 to RQ3.

7.1 Task definition

The system checks the validity of references: whether each reference can be matched to an external scholarly record, with bibliographic details that agree well enough to confirm the match. It does not check whether the cited work supports what the text says about it, which is a harder problem. Verified therefore means that the work was found and the details agree, which says nothing about whether the citation is used appropriately [61].

Under the criterion of Section 3.1, this task is decidable. The decision is made by database lookup and deterministic matching rules and does not rely on a generative model. Every entry that cannot be confirmed is marked as Unverified, which means “could not be verified”. The cases in which the decision can go wrong can be listed in advance, and the main one is that the databases do not index a work.

7.2 System design

The user pastes in the text of the references. It can be copied straight from a PDF or a LaTeX .bib file, in any format. The system works in three steps. In the first step an LLM parses the text into individual references and extracts the title and authors. This is the only place where an LLM is called. The second step queries several scholarly databases in turn. The third step compares titles and authors with deterministic rules (Section 4.2), and entries that match are marked as Verified.

Reference entries are a typical case of irregular input. There are dozens of citation styles, journal names are often abbreviated, text copied from a PDF contains line breaks, and author names are written in different ways. Parsing is therefore left to an LLM. For each entry, only the title and the surnames of at most the first three authors are kept. The more fields the parser extracts, the more room there is for an LLM error to affect the later queries, and the title and the surnames are enough to support a match. The LLM only parses and takes no part in the verdict. The consequences of a parsing error are therefore limited: if a title or an author is parsed wrongly, no matching record is found afterwards, and the entry ends up as Unverified. If the verdict were left to an LLM, its errors would reach the user directly as firm conclusions.

The lookup queries six databases in a fixed order, each of them by title. The general bibliographic databases with the broadest coverage come first, followed by a computer science bibliography, a preprint server and a book catalogue, and the last database is there to catch what the earlier ones missed. As soon as one database returns a match with an accessible link, the entry is marked as Verified and the remaining databases are not queried. This keeps the number of queries per reference as low as possible. The price is that a wrong match in an early database cannot be corrected by a later one, so the matching thresholds are set conservatively (Section 4.2). When a request times out

or is rate-limited, the system treats it as if the database had returned nothing and moves on to the next one.

The report gives a result for each reference. A Verified entry is shown with a confidence score for the match and a link to the external record, so the user can check it. Unverified entries are left for the user to look up. The launch page stresses that Unverified does not mean fabricated. Common reasons include incomplete bibliographic details, differences in format, obscure venues, works that the databases do not index, and older publications [61].

The advice the launch page gives to three kinds of users shows how this level of delegation is meant to be used. Authors run it once before submission and complete their bibliographic details. Reviewers use it to direct their attention: if a key claim depends on an Unverified reference, it is worth a few minutes of manual checking. Editors use it as a first screen before sending a paper out for review.

7.3 Evaluation

The platform reported one test at launch. A sample of 100 problematic references was taken from the papers with fabricated references identified by Sakai et al. [8], and the system detected 93% of them [61]. The sample consists of fabricated references that occurred in real papers. It contains both entries that cannot be found and entries whose bibliographic details have been altered [61].

A second run used genuine references. The system checked the 42 references of one paper and marked 41 as Verified, taking 1.4 seconds per reference on average. The single Unverified entry is a recently published conference paper. It exists, but it could not be found in the databases at the time. Of the 41 Verified entries, 38 were found in the first database, two in the second and one in the third. The run shows two things: most entries can be verified in the first database, and a recent publication may not have been indexed yet.

7.4 Summary

Of the three systems, this is the only one whose verdicts come from a deterministic source. The LLM's part is limited to parsing, the decision is made by databases and matching rules, and every verdict is backed by a record that the user can open. The Unverified entries are left to the user. The label is worded as "could not be verified", and the user looks each one up.

An Unverified entry may be a work that the databases do not index, a case in which parsing or matching failed, or a reference that really is fabricated. The system does not distinguish between these, and the user confirms each entry. The three cases also call for different remedies. Coverage gaps are narrowed by adding data sources, parsing and matching failures are reduced by engineering work, and entries that do not exist are left for a person to confirm. Because Unverified is a mixture of these three cases, it can only be an abstention and never an accusation.

The main cause of failure is database coverage: books, non-English literature and recent publications are more likely to be missed. Gaps of this kind can be listed in advance and communicated to users. The consequence is some additional Unverified entries, but no genuine reference is called fabricated.

An evaluation at this level should report the detection rate on fabricated entries together with the pass rate on real entries.

8 Cross-Case Analysis

This chapter looks at the three systems together and answers RQ3: under what conditions this kind of assistance is reliable, where its limits are, and how it should be used.

8.1 Comparison of the three systems

Table 8.1 summarises the three systems.

Table 8.1: Comparison of the three systems

	Reference Check (L1)	**Correctness Check (L2)**	**Reference Finder (L3)**
Task	Whether a reference exists	Whether the content of a paper has objective errors	Which further papers should be cited
Who decides	Databases and matching rules	The human; the system only flags	The human; the system only suggests
What is left to the human	Entries that could not be verified	Verifying each finding; missed errors are invisible	Deciding on each suggestion; papers not found are invisible
Ground truth for evaluation	Confirmed fabricated references (exact)	Injected errors (injected)	The authors' original citations (recovered)
Evidence so far	The platform reports detecting 93% of fabricated references (100 entries); 41 of the 42 real references of one paper passed	accuracy@10 of 25% to 40% in the preliminary experiment (20 papers); best agreement with the Action Editors in the external evaluation, and 4.5% detected under the strict criterion	In four runs on three papers, a paper with a relevance score of at least 7 was found for 40 of 46 statements

The differences between the three systems can be summed up by the level of delegation. Further up the ladder the task is more open, the system can conclude less on its own, and more is left to the human. The level is determined by the properties of the task, not by the amount of engineering. However much the pipeline of Correctness Check is improved, its conclusions still need human verification as long as there is no deterministic source to check them against.

Internally, the three systems divide the work in the same way. The LLM handles understanding and generating language, for example parsing references, proposing suspected errors, and identifying statements that need a citation. Every step that can be checked goes to databases and deterministic rules, for example whether a work exists and whether the authors already cite it. The final judgement stays with the human.

Whether a system can use AI should not be judged for the system as a whole. One has to separate the steps that only need language processing from the steps that have to reach a conclusion. Steps that a deterministic source can decide should be given to it as far as possible, and different sources should complement one another, because a system that depends on a single data source inherits its gaps. Reference Check queries six databases in turn, and Reference Finder combines several search sources.

8.2 Sources of ground truth and the strength of conclusions

As set out in Section 4.5, the meaning of an evaluation figure depends on where its ground truth comes from (Table 4.2). Exact ground truth consists of existing records and does not depend on the system under evaluation, so conclusions drawn from it come closest to right and wrong in the ordinary sense. Reference Check falls under this mechanism, and the platform's test used references that people had already confirmed to be fabricated. Its weak point is coverage: an entry that the databases do not index may still exist.

Injected ground truth is put into the evaluated paper by the evaluator, as in the FLAWS experiment of Chapter 5. It gives an open task answers that can be checked exactly. The price is that only the ability to find injected errors is measured. How far this ability carries over to natural errors depends on how different the two error distributions are, and that difference cannot be measured (Lemma 4.1). Injected errors also bear the mark of the injecting model: the errors injected by GPT-5 are harder for every model (Section 5.5).

Recovered ground truth is taken from the evaluated paper itself. The strip-and-recover protocol of Chapter 6 removes the authors' original citations and then scores against them. This ground truth is biased, because a paper that the authors did not cite may still deserve a citation. Precision is therefore only an approximate lower bound on the quality of the suggestions (Proposition 4.6).

The ground-truth mechanism limits how strong a conclusion can be, and this limit has nothing to do with how good the system is. The external evaluation in Section 5.7 is an example. With the Action Editors' judgement as ground truth, the CSPaper system is the best. With injected errors as ground truth and under the strict criterion, GPT-5.5 is the best. The two rankings do not contradict each other, because they answer different questions. A reported percentage should therefore always state which kind of ground truth it rests on.

When the results are split by the source of the injected errors, the detection rates differ widely. In the FLAWS paper, the 39.1% of raw GPT-5 splits into 30.9% and 44.0%. In the preliminary experiment, the 40.0% of gpt-5 splits into 12.5% and 58.3%, although each group has only 8 or 12 papers. The combined figure hides this difference, so results should be reported by stratum whenever systematic failure is suspected.

The two kinds of failure call for different responses. When misses are unrelated across runs and across models, random failures can be reduced by redundancy. Systematic failures cannot be removed by running the system again. If a kind of error is generally harder for one family of models, adding models of the same family only repeats the same weakness, and a different source is needed to make up for it.

8.3 Recommendations for use

For platforms, journals and conferences that use these tools, this thesis makes four recommendations.

1. Verification should be a required step. Research on automation bias shows that merely giving people the option to verify does not ensure that they do (Section 2.4). A finding of Correctness Check should not be the basis of any formal conclusion until a human has verified it. Nor should the suggestions of Reference Finder come with a button that accepts them all at once.

2. Every output should carry evidence that can be checked: a link to the database record, the location of an error and the reason for it, or the source of a recommended

paper.

3. The wording should not go beyond the level of delegation. “Could not be verified” is not “fabricated”, “possible error” is not “error”, and “suggested addition” is not “missing”. Wording that is stronger than the level of delegation raises that level without authorisation, and it leads users to overestimate how reliable a conclusion is. An author who has been wrongly flagged faces a high cost in defending the paper, so those who use the tools should offer a channel for appeal and correction. They should also rule out using the raw output of a tool directly as grounds for a rejection or an accusation.

4. Known gaps should be stated in advance, for example the limited coverage of non-English literature and books in the databases.

TMLR’s AI review pilot is a practical example. The AI review only assesses whether the claims of a paper are supported by evidence and gives no recommendation on acceptance. The authors can respond, and the Action Editor makes the final decision [59, Section 5.7]. In terms of the framework, the output of the system is placed in the position of checked delegation: the Action Editor is the one who verifies, and the authors’ response acts as the channel for appeal.

8.4 Societal perspective and sustainability

All three systems address work in reviewing and writing that is mechanical but necessary. Ideally, once these checks are handed to a machine, reviewers can spend their time on the judgements that only people can make, such as whether a contribution is valuable and whether the choice of method is sound. Given the gap between submission numbers and reviewing capacity described in Section 1.1, this is one of the few workable ways of easing the pressure. This redistribution of work also has risks. Reviewers may concentrate on the problems listed in a report and stop checking what the report does not mention. Responsibility may be diluted, because users assume that the tool has already checked. Mechanical checking is also one of the ways in which new reviewers gain experience. All three risks can only be confirmed by user studies, and those who design a review process should take them into account.

Tools of this kind also have a cost. Reference Check takes 1.4 seconds per reference on average (Section 7.3), and the four runs of Reference Finder took between 4 and 21 minutes each (Table 6.1). The more closed a step is, the less it needs an expensive general-purpose model. A small model may be enough, or no model at all. It should also be said that these systems trade extra computation for human attention, and the human verification work is not reduced.

The coverage of the search sources and databases is uneven. Non-English literature, books, regional journals and older work are less likely to be found. Authors who depend on such literature are more likely to have their references marked as “could not be verified”, and they receive fewer suggestions for missing citations. Failures of this kind show up as omissions and do not turn into accusations against the author, but different authors do not receive the same help. If the reports of Correctness Check were used as a signal of quality, differences in false alarms between groups could become an extra burden for some of them. For authors who are not native speakers, the effect of such tools goes in two directions. Objective checks shift a reviewer’s attention from language to content, which may reduce the unfairness caused by a language disadvantage. If these authors’ citation habits lead to more omissions or false alarms, however, their burden grows again.

Those who deploy the tools should state the known gaps in advance (Section 8.3) and monitor, for each group of authors, how useful the tools are and how often they misjudge.

8.5 Limitations

1. The empirical evidence is limited in scale. Correctness Check was tried on 20 papers with one run per model in the preliminary experiment, and the Reference Finder tests cover three papers and show how the system runs. For Reference Check, the detection rate comes from a single test reported by the platform. The conclusions of the thesis are therefore indicative at best.
2. The results for Correctness Check all come from injected errors and cannot be carried over directly to errors that occur naturally. Whether an error counts as a hit is decided by an LLM judge, so the results depend on the judge.
3. The preliminary experiment ran the prototype before optimisation, and the external evaluation assessed the optimised and deployed system. The evaluation methods differ as well, so the numbers cannot be compared directly. The tests of Reference Finder use the research implementation, whose parameters are not exactly those of the platform version.
4. The papers used in the evaluations are mainly English-language papers in computer science and machine learning.
5. The recommendations in Section 8.3 are design principles. How reviewers and authors will actually use these tools has to be answered by user studies.

9 Conclusion and Future Work

9.1 Answers to the research questions

RQ1: Which tasks are suitable for LLM-based support, and how should the tools be designed? How far a verification task can be handed to AI depends on whether its conclusions can be checked against a deterministic source, and on whether reliable ground truth is available for evaluation. Tasks that can be checked, such as establishing whether a reference exists, can be delegated fully. For tasks that have a correct answer but cannot be checked automatically, the system can only flag possible errors for a human to verify, as in Correctness Check. For tasks without a unique correct answer, the system can only make suggestions, as in reference completion. I built one system for each of these three kinds of task. Their designs follow the same division of labour: the LLM handles language, the steps that can be checked go to databases and rules, and the final judgement stays with the human.

RQ2: How well do the tools perform, and how can they be evaluated when no established ground truth is available? On 20 FLAWS papers, the prototype of Correctness Check reached an accuracy@10 of 25% to 40%. For the two models for which the FLAWS authors report a single-prompt baseline, the results were above that baseline (40.0% against 39.1%, and 35.0% against 19.8%). The sample has only 20 papers, however, and this advantage remains to be confirmed. In the external evaluation organised by TMLR, the optimised system agreed best with the Action Editors, but under the strict criterion it identified only 4.5% of the injected errors [22]. Reference Check detected 93% of fabricated references in the test reported by the platform [61]. In another run, 41 of the 42 real references of one paper were verified. Reference Finder completed four test runs on three papers and, for 40 of the 46 statements, found a paper with a relevance score of at least 7. Every recommended paper can be looked up and checked. For tasks without established ground truth, I designed the strip-and-recover protocol, which uses the authors' original citations as substitute ground truth.

RQ3: Under what conditions is this kind of assistance reliable, where are its limits, and how should it be used? The steps whose conclusions can be checked against a database are reliable. The rest of the output is reliable only after human verification. The current systems cannot yet replace a reviewer's judgement, and the available evidence is also very limited (Section 8.5). In use, verification should be a required step, every output should come with evidence, and the wording should match what the system can actually guarantee (Section 8.3).

9.2 Contributions

The thesis makes three contributions. First, it proposes a three-level delegation ladder defined by decidability and ground-truth availability, together with a mathematical formulation (Chapters 3 and 4). Second, it implements three systems and deploys them on CSPaper. After TMLR's evaluation, the review agent built around Correctness Check was selected as the system for the journal's AI review pilot (Chapters 5 to 7). Third, it carries out a preliminary evaluation of Correctness Check on FLAWS, designs the strip-and-recover protocol for reference completion, and tests Reference Finder on real papers (Sections 5.4, 6.3 and 6.4).

9.3 Future work

I see two directions for future work. One is to run a user study of how these tools affect the behaviour of reviewers and authors in actual reviewing and writing. The other is to apply the delegation ladder to other verification tasks, such as checking whether a paper agrees with its code, and to test whether the framework holds beyond the three systems studied here.

9.4 Closing remarks

LLMs make it easier for fabricated citations and content errors to appear in large numbers, and the systems that can check them at a similar scale are, for now, mostly built around LLMs as well. The approach of this thesis can be stated in one line: LLMs as the language interface, deterministic sources as the adjudicator, and humans handling the residual. The three systems show how far this division of labour goes on three kinds of task: a system can decide by itself whether a reference exists, can only flag possible errors in the content, and can only suggest citations that may be missing.

References

[1] Paper Copilot. *ICLR statistics*. 2026. URL: https://papercopilot.com/statistics/iclr-statistics/ (visited on Nov. 12, 2026).

[2] J. Andreas et al. “Submission policies for ICLR 2027”. In: *ICLR Blog* (Sept. 2, 2026). URL: https://blog.iclr.cc/2026/09/02/submission-policies-for-iclr-2027/.

[3] D. Singh Chawla. “Inundated AI conferences impose submissions caps”. In: *Nature Index* (Oct. 13, 2026). DOI: 10.1038/d41586-026-03153-5.

[4] S. Xi et al. *FLAWS: A benchmark for error identification and localization in scientific papers*. 2025. arXiv: 2511.21843.

[5] F. Bianchi et al. *To err is human: Systematic quantification of errors in published AI papers via LLM analysis*. 2025. arXiv: 2512.05925.

[6] N. B. Shah. “Challenges, experiments, and computational solutions in peer review”. In: *Communications of the ACM* 65.6 (2022), pp. 76–87. DOI: 10.1145/3528086.

[7] L. You, L. Cao, and I. Gurevych. *Preventing the collapse of peer review requires verification-first AI*. 2026. arXiv: 2601.16909.

[8] Y. Sakai, H. Kamigaito, and T. Watanabe. *HalluCitation matters: Revealing the impact of hallucinated references with 300 hallucinated papers in ACL conferences*. 2026. arXiv: 2601.18724.

[9] S. Ansari. *Compound deception in elite peer review: A failure mode taxonomy of 100 fabricated citations at NeurIPS 2025*. 2026. arXiv: 2602.05930.

[10] K. A. Robinson and S. N. Goodman. “A systematic examination of the citation of prior research in reports of randomized, controlled trials”. In: *Annals of Internal Medicine* 154.1 (2011), pp. 50–55. DOI: 10.7326/0003-4819-154-1-201101040-00007.

[11] M. Naddaf. “Major AI conference flooded with peer reviews written fully by AI”. In: *Nature* 648.8093 (2025), pp. 256–257. DOI: 10.1038/d41586-025-03506-6.

[12] A. Agarwal et al. “On violations of LLM review policies”. In: *ICML Blog* (Mar. 18, 2026). URL: https://blog.icml.cc/2026/03/18/on-violations-of-llm-review-policies/.

[13] J. Biswas et al. *AI-assisted peer review at scale: The AAAI-26 AI review pilot*. 2026. arXiv: 2604.13940.

[14] W. H. Walters and E. I. Wilder. “Fabrication and errors in the bibliographic citations generated by ChatGPT”. In: *Scientific Reports* 13, 14045 (2023). DOI: 10.1038/s41598-023-41032-5.

[15] Z. Ji et al. “Survey of hallucination in natural language generation”. In: *ACM Computing Surveys* 55.12, 248 (2023). DOI: 10.1145/3571730.

[16] A. Agrawal et al. “Do language models know when they’re hallucinating references?” In: *Findings of the Association for Computational Linguistics: EACL 2024*. Association for Computational Linguistics, 2024, pp. 912–928. DOI: 10.18653/v1/2024.findings-eacl.62.

[17] J. M. Nicholson et al. “scite: A smart citation index that displays the context of citations and classifies their intent using deep learning”. In: *Quantitative Science Studies* 2.3 (2021), pp. 882–898. DOI: 10.1162/qss_a_00146.

[18] G. Cabanac, C. Labbé, and A. Magazinov. *Tortured phrases: A dubious writing style emerging in science. Evidence of critical issues affecting established journals*. 2021. arXiv: 2107.06751.

[19] J. Priem, H. Piwowar, and R. Orr. *OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts*. 2022. arXiv: 2205.01833.

[20] R. Kinney et al. *The Semantic Scholar open data platform*. 2023. arXiv: 2301.10140.

[21] W. Liang et al. “Can large language models provide useful feedback on research papers? A large-scale empirical analysis”. In: *NEJM AI* 1.8, AIoa2400196 (2024). DOI: 10.1056/AIoa2400196.

[22] V. Rao and N. B. Shah. *Evaluating AI reviewers' ability to assess soundness for deployment in the TMLR journal*. Preprint. 2026. URL: https://www.cs.cmu.edu/~nihars/preprints/TMLR_AI_reviewer_evals.pdf.

[23] T. Baumgärtner and I. Gurevych. *SciCoQA: Quality assurance for scientific paper–code alignment*. 2026. arXiv: 2601.12910.

[24] Q. He et al. “Context-aware citation recommendation”. In: *Proceedings of the 19th International Conference on World Wide Web*. Association for Computing Machinery, 2010, pp. 421–430. DOI: 10.1145/1772690.1772734.

[25] M. Färber and A. Jatowt. “Citation recommendation: Approaches and datasets”. In: *International Journal on Digital Libraries* 21.4 (2020), pp. 375–405. DOI: 10.1007/s00799-020-00288-2.

[26] C. Bhagavatula et al. “Content-based citation recommendation”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, 2018, pp. 238–251. DOI: 10.18653/v1/N18-1022.

[27] N. Gu, Y. Gao, and R. H. R. Hahnloser. “Local citation recommendation with hierarchical-attention text encoder and SciBERT-based reranking”. In: *Advances in Information Retrieval (ECIR 2022)*. Springer, 2022, pp. 274–288. DOI: 10.1007/978-3-030-99736-6_19.

[28] I. Beltagy, K. Lo, and A. Cohan. “SciBERT: A pretrained language model for scientific text”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 2019, pp. 3615–3620. DOI: 10.18653/v1/D19-1371.

[29] A. Cohan et al. “SPECTER: Document-level representation learning using citation-informed transformers”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020, pp. 2270–2282. DOI: 10.18653/v1/2020.acl-main.207.

[30] M. Ostendorff et al. “Neighborhood contrastive learning for scientific document representations with citation embeddings”. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2022, pp. 11670–11688. DOI: 10.18653/v1/2022.emnlp-main.802.

[31] A. Singh et al. “SciRepEval: A multi-format benchmark for scientific document representations”. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2023, pp. 5548–5566. DOI: 10.18653/v1/2023.emnlp-main.338.

[32] D. Wright and I. Augenstein. “CiteWorth: Cite-worthiness detection for improved scientific document understanding”. In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, 2021, pp. 1796–1807. DOI: 10.18653/v1/2021.findings-acl.157.

[33] T. Gao et al. “Enabling large language models to generate text with citations”. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2023, pp. 6465–6488. DOI: 10.18653/v1/2023.emnlp-main.398.

[34] B. Bohnet et al. *Attributed question answering: Evaluation and modeling for attributed large language models*. 2022. arXiv: 2212.08037.

[35] M. D. Skarlinski et al. *Language agents achieve superhuman synthesis of scientific knowledge*. 2024. arXiv: 2409.13740.
[36] A. Asai et al. *OpenScholar: Synthesizing scientific literature with retrieval-augmented LMs*. 2024. arXiv: 2411.14199.
[37] O. Press et al. "CiteME: Can language models accurately cite scientific claims?" In: *Advances in Neural Information Processing Systems* 37 (2024). URL: https://arxiv.org/abs/2407.12861.
[38] Y. M. Choi et al. "CiteGuard: Faithful citation attribution for LLMs via retrieval-augmented validation". In: *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2026, pp. 6241–6257. DOI: 10.18653/v1/2026.acl-long.282.
[39] Y. M. Choi et al. *ATTRICITE: Training an open 4B model for citation recovery toward faithful attribution*. 2026. arXiv: 2609.14248.
[40] M. T. R. Ratul et al. "MasterSet: A large-scale benchmark for must-cite citation recommendation in the AI/ML literature". In: *Proceedings of the 2026 SIAM International Conference on Data Mining (SDM)*. 2026. URL: https://arxiv.org/abs/2604.17680.
[41] M. Hong et al. *CiteLLM: An agentic platform for trustworthy scientific reference discovery*. 2026. arXiv: 2602.23075.
[42] Y. Huang et al. *ReCite: Agentic reasoning for faithful citation*. 2026. arXiv: 2609.09156.
[43] R. Parasuraman and V. Riley. "Humans and automation: Use, misuse, disuse, abuse". In: *Human Factors* 39.2 (1997), pp. 230–253. DOI: 10.1518/001872097778543886.
[44] R. Parasuraman, T. B. Sheridan, and C. D. Wickens. "A model for types and levels of human interaction with automation". In: *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans* 30.3 (2000), pp. 286–297. DOI: 10.1109/3468.844354.
[45] L. J. Skitka, K. L. Mosier, and M. Burdick. "Does automation bias decision-making?" In: *International Journal of Human-Computer Studies* 51.5 (1999), pp. 991–1006. DOI: 10.1006/ijhc.1999.0252.
[46] Z. Buçinca, M. B. Malaya, and K. Z. Gajos. "To trust or to think: Cognitive forcing functions can reduce overreliance on AI in AI-assisted decision-making". In: *Proceedings of the ACM on Human-Computer Interaction* 5.CSCW1, 188 (2021). DOI: 10.1145/3449287.
[47] L. Zheng et al. "Judging LLM-as-a-judge with MT-Bench and Chatbot Arena". In: *Advances in Neural Information Processing Systems* 36 (2023). URL: https://arxiv.org/abs/2306.05685.
[48] C. K. Chow. "On optimum recognition error and reject tradeoff". In: *IEEE Transactions on Information Theory* 16.1 (1970), pp. 41–46. DOI: 10.1109/TIT.1970.1054406.
[49] R. El-Yaniv and Y. Wiener. "On the foundations of noise-free selective classification". In: *Journal of Machine Learning Research* 11 (2010), pp. 1605–1641.
[50] S. E. Robertson. "The probability ranking principle in IR". In: *Journal of Documentation* 33.4 (1977), pp. 294–304. DOI: 10.1108/eb026647.
[51] S. Ben-David et al. "A theory of learning from different domains". In: *Machine Learning* 79.1–2 (2010), pp. 151–175. DOI: 10.1007/s10994-009-5152-4.
[52] C. Elkan and K. Noto. "Learning classifiers from only positive and unlabeled data". In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge*

Discovery and Data Mining. Association for Computing Machinery, 2008, pp. 213–220. DOI: 10.1145/1401890.1401920.

[53] D. Madras, T. Pitassi, and R. Zemel. “Predict responsibly: Improving fairness and accuracy by learning to defer”. In: *Advances in Neural Information Processing Systems* 31 (2018), pp. 6150–6160.

[54] H. Mozannar and D. Sontag. “Consistent estimators for learning to defer to an expert”. In: *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 2020, pp. 7076–7087. URL: https://proceedings.mlr.press/v119/mozannar20b.html.

[55] M. Raghu et al. *The algorithmic automation problem: Prediction, triage, and human effort*. 2019. arXiv: 1903.12220.

[56] C. Buckley and E. M. Voorhees. “Retrieval evaluation with incomplete information”. In: *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 2004, pp. 25–32. DOI: 10.1145/1008992.1009000.

[57] C. F. Manski. *Partial identification of probability distributions*. Springer, 2003. DOI: 10.1007/b97478.

[58] D. Metropolitansky and J. Larson. “Towards effective extraction and evaluation of factual claims”. In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2025, pp. 6996–7045. DOI: 10.18653/v1/2025.acl-long.348.

[59] TMLR. *AI reviews at TMLR for assessing soundness*. Medium. July 4, 2026. URL: https://medium.com/@TmlrOrg/ai-reviews-at-tmlr-for-assessing-soundness-4405d7813cc8.

[60] CSPaper. *Reference Finder: Find the citations your paper is missing*. Scholar7 AB. Nov. 2026. URL: https://cspaper.org/articles/reference-finder.

[61] CSPaper. *Reference validity check: A step towards verification-first AI for paper peer review*. Scholar7 AB. Mar. 2026. URL: https://cspaper.org/articles/reference-validity-check.

Technical
University of
Denmark

Lautrupvang 15
2750 Ballerup
Tlf. 4525 1700

https://engtech.dtu.dk/