Can the Current Generation of LLMS Produce an Adequate Index?

Nov 2025

Prepared by the AI Committee of the American Society for Indexing

Elizabeth Bartmess

Michele R. Combs

Abstract

This paper supplements our previously-published white paper by investigating the capacity of the current generation of large language models (LLMs) to index a book chapter. The white paper found that an earlier generation of LLMs performed poorly on the three criteria of completeness, navigability, and accuracy. We find that while the current generation does not underindex as severely, it has issues with excessive subheadings and navigability, and continues to fail a standard indexing test for accuracy. LLMs still cannot produce an adequate index.

Introduction

In our white paper[1], we reported on the performance of several large language model (LLM) AI chatbots with respect to indexing. They performed poorly across the board. We found that they underindexed, failed to provide readers with effective access to subtopics and navigation to related topics, and failed a standard test for indexing accuracy.

In this update, we assess, using a slightly modified approach, whether the newest models of two LLMs, as of September/October 2025, can produce adequate indexes.

[1] Elizabeth Bartmess and Michele R. Combs, *LLM-Generated Book Indexes: Can They Replace Professionally Created Indexes?* Prepared by the AI Committee of the American Society for Indexing (2025), https://asindexing.org/ai-news/white-paper-ai-index/

Methods

Materials, LLMs, and prompts

Materials to be indexed

We initially planned to repeat our tests from the white paper with the same books and the same LLMs, but ran into new technological constraints.

First, ChatGPT 5 declined to index a full book, saying it would be too large a project; it would only index a single chapter. Second, the newer models of Claude (Opus 4 and 4.1) were unable to index the previous material due to length limits. Even after cutting 10 pages from the book to get it under the length limit, Claude's Opus 4 model was still unable to provide a complete index before hitting the length limit.

Due to those technological issues, for the current paper we elected to do shorter tests using a single chapter. This approach comes with an important caveat: the primary use case for a book index is a book. That is, most indexes are created from books, not individual chapters, and this influences index content and structure in important ways. Additionally, LLM performance degrades with the length of input[2,3,4], meaning that chapter-based tests may overestimate LLMs' performance. If an LLM can index a chapter effectively, that does not mean it will be able to index a book effectively. However, if an LLM *cannot* index a chapter effectively, it is unlikely to be able to index a book effectively either, given that indexing books typically involves more complexity than indexing chapters.

The first author had previously created a professional index for chapter 2 of an out-of-print book on computers[5] for use in reviewing AI-based indexing software[6], and we use that here as a comparison index. That book is a relatively straightforward work pitched at a general audience;

---

[2] Mosh Levy, Alon Jacoby, and Yoav Goldberg, "Same Task, More Tokens: the Impact of Input Length on the Reasoning Performance of Large Language Models," arXiv preprint, arXiv:2402.14848v1 (February 2024), https://ar5iv.labs.arxiv.org/html/2402.14848v1
[3] Kelly Hong, Anton Troynikov, and Jeff Huber, *Context Rot: How Increasing Input Tokens Impacts LLM Performance* (Chroma Research, July 2025), https://research.trychroma.com/context-rot
[4] Onn Yun Hui, "Evaluating Long Context Lengths in LLMs: Challenges and Benchmarks," *Medium*, March 4, 2025, https://onnyunhui.medium.com/evaluating-long-context-lengths-in-llms-challenges-and-benchmarks-ef77a220d34d
[5] Daniel S. Halacy, *Computers—The Machines We Think With* (New York: Harper & Row, 1962).
[6] Our reviews of AI-based indexing software can be found on our website.

it is similar to one book used for the white paper (*How to Stop Worrying and Start Living*) but less similar to the two others (*Behavior of Organisms* and *The Psychology of Day-Dreams*, both of which are scholarly books aimed at more specialized audiences). While we will compare our current findings to those of the white paper, we encourage the reader not to take comparisons too far; differences in performance may be due both to changes in the LLMs and to differences in the material being indexed.

However, we expect our findings to still be able to speak to whether the current generations of LLMs can make an adequate index.

Supporting materials

In our previously-published white paper, for two of the LLMs, we included a PDF copy of the "Criteria for the ASI Excellence in Indexing Award" to provide the LLMs with indexing guidance.

For this paper, we provided LLMs with both the "Criteria" and with a copy of our white paper. Our rationale for including the white paper was that human feedback is a common and standard technique for improving AI performance, and the white paper is, to our knowledge, the most complete evaluation of AI indexing performance available. Using it to attempt to improve LLM performance is an obvious next step.

LLMs

For the white paper, we used ChatGPT 4.5 and Claude 3 Opus. Here we use ChatGPT 5.0 and Claude Opus 4.1. Since the date of the white paper tests, Claude has begun using context across conversations (i.e., it remembers and uses information from past sessions). Consequently, we used Claude’s Incognito Mode to prevent bleed-over from other conversations and past indexing attempts.

We also used Perplexity for the white paper. For those experiments, Perplexity generated an index without page numbers, which limited the analyses we could do on its provided index and would also limit our ability to make comparisons between its current version and the white paper version. Consequently, we elected to omit Perplexity from the current analyses.

Prompt

We used the following prompt:

> Please create a professional-quality index for the attached chapter, using the same techniques that a professional indexer would use (e.g., cross-references, subheadings, etc). Please ensure that the completed index adheres to the guidelines in the attached Criteria for the ASI Excellence in Indexing Award. Please also use the attached AI white paper to improve your output.

Results

Here we report the performance of ChatGPT 5 and Claude 4.1 Opus on the chapter with the performance of ChatGPT 4.5 and Claude 3 Opus on books as reported in our previously-published white paper.[7]

Completeness

We use three primary measures of completeness: main entries (which we will cover here along with discussion of subheadings), names, and titles of works.

Main entries and subheadings

Main entries are the alphabetized entries a reader will use to look up names, terms, and concepts, for example "behavior problems" in a work on dogs. For indented indexes, subheadings are the indented terms appearing below some main headings, and are used when needed to break down information further.

The index created by a professional indexer contained 122 main headings and 24 subheadings.

ChatGPT

For our test chapter, ChatGPT 5 provided an index with 59 main headings, 48% as many as the professionally created index. While this is an improvement over ChatGPT 4.5's indexes, which

[7] Elizabeth Bartmess and Michele R. Combs, *LLM-Generated Book Indexes: Can They Replace Professionally Created Indexes?* Prepared by the AI Committee of the American Society for Indexing (2025), https://asindexing.org/ai-news/white-paper-ai-index/

had only 20% as many main headings as one book's original index and 15% as many as another's, ChatGPT is still severely underindexing.

A complementary problem is that ChatGPT's index also included an inordinate number of subheadings: 131, over five times as many as the professionally created index. In this case, ChatGPT produced various entries such as:

abacus, 4–7
ancient origins of, 4–5
base systems used in, 5–6
design variations, 5–6
educational uses of, 6
etymology of term, 5
modern uses of, 6
portable counters as precursors to, 4–5
soroban (Japanese abacus), 5–6
symbolism and cultural implications, 6
teaching and therapeutic value, 6

As you can see, all of the subheadings refer to the same three pages; this violates indexing best practices and makes for an unnecessarily cumbersome index that requires the reader to wade through considerably excess verbiage in search of what they're after. It also bloats the index, making it longer and costlier to print.

In good indexing practice, subheadings are used only to break down excessively long page ranges (like "abacus, 1–50") or when a topic is discussed in more than five or six locations in the text (that is, when the entry would be followed by more than five or six page numbers or ranges). The appropriate index entry here would simply be "abacus, 4–7", without subheadings.

Claude

Claude 4.1 Opus provided an index with 112 main headings, 92% as many as the professionally created index's 122 main headings. This is an improvement over both ChatGPT's current index and Claude 3 Opus's index, which had 35% as many main headings as the book's original index.

However, Claude's current index showed the same issue as ChatGPT's index with excessive subheadings. It contained six times as many subheadings as the original index, with entries such as:

abacus, 4-6

- decimal system variations, 5
- etymology (*abax*), 5
- Japanese *soroban*, 5
- Korean *son pan*, 5-6
- Loy's decimal point marker, 6
- miniaturization of, 6
- modern uses, 5-6
- origins, 4
- political implications, 5-6

As mentioned above, excessive and redundant subheadings violate indexing best practice, frustrate the reader, and increase the cost of publication. We will discuss subheadings again in the summary to this section.

Names

Indexes are expected to capture the names of individual people who are significantly discussed in the work. The professionally created index contains 39 names of people, including one pseudonym.

ChatGPT

ChatGPT 5 included 15 names of people in its index (39% as many as the professionally created index). Although this is a noticeable improvement over ChatGPT 4.5's indexes, which had 4.8% and 4.1% as many names as the original indexes, it is still severely underindexing names.

There were also a couple of oddities in its name indexing. In addition to an entry for "Pascal, Blaise" it included an entry for "Blaise Pascal", and a cross reference from "Ramón Lull" to "Lull, Ramón". This is not indexing best practice. While indexers do sometimes double-post names or use cross-references to point readers to the correct version of a name, we limit this to instances where readers are likely to look in the wrong place, such as with (for example) some Japanese names in books for American audiences where a subset of readers might not identify the family vs personal name correctly, or cases where an author has a very well-known pseudonym (e.g. Clemens, Samuel. *See* Twain, Mark.)

Claude

Claude 4.1 Opus provided an index with 38 names of people including one pseudonym, 97% as many names as in the professionally created index. However, three of these names were passing mentions that should not have been included in the index, meaning that Claude correctly picked up 35 names of people including one pseudonym, 90% of the names in the professionally created index. This is a moderate improvement over Claude 3 Opus's index, which included 84% as many names as the original index.

An index that omits 10% of indexable names is still not an adequate index, although it raises the possibility that, if future models of Claude continue to improve in this area—something that remains to be seen—AI might be of use as an assistant to a professional indexer in creating a name index.

Titles of works

Titles of works include books, short stories, songs, movies, television shows, plays, etc., that are significantly discussed. The professionally created index contained 14 titles of works, which were posted both as main headings and under the author's names.

ChatGPT

ChatGPT 5 included only one title of a work as a main heading in its index (7% as many as in the professionally created index). It also posted three titles of works as subheadings under the author's names. While this is an improvement over ChatGPT 4.5's indexes, for which it failed to index *any* titles of works, ChatGPT is still severely underindexing titles of works and posting them inconsistently (they should be posted both as main headings and, depending on publisher preferences, under the author's name as well).

Claude

Claude 4.1 Opus's index included three titles of works (21% of the professionally created index's titles). Two of these also appeared as subheadings under the author's names, as did a third that was not given its own main heading. All four were also included as subheadings under a "literature, computers in" entry (which could and likely should have been condensed into just a main heading). This is an improvement over Claude 3 Opus's index, which included no titles of works, and better than ChatGPT, but still very poor performance overall.

Missing pages

The professionally created index covered all of the chapter's 31 pages.

ChatGPT 5 failed to include any references to pages 21 through 31, omitting a third of the chapter entirely. (ChatGPT 4.5 failed to index 93% of one book's pages and 21% of the second book's pages.)

Claude 4.1 Opus's index covered all 31 pages. (Claude 3 Opus failed to index 2% of the pages it was given.)

Average numbers of entries per page

Typical indexes range from two to ten entries per page on average, with scholarly works toward the higher end and "trade" books towards the lower end. The professionally created index averaged 6.7 entries per page.

Despite having missed a third of the chapter, ChatGPT 5's index averaged 10.97 entries per page; when only the pages it did index were counted, the index averaged 17.0 entries per page. This extensive overindexing stands in contrast to ChatGPT 4.5's severe underindexing (for which it averaged 0.19 entries per page and 0.29 entries per page) and is likely due to the current version's excessive use of redundant subheadings. overindexing violates indexing best practice, bloats the index size, and frustrates the reader by burying genuinely significant material in an avalanche of insignificant chaff.

Claude 4.1 Opus's index averaged 11.7 entries per page, compared with 1.8 entries per page for its index for the white paper. In contrast with Claude 3 Opus's severe underindexing, it is overindexing the chapter, although not as severely as ChatGPT.

Completeness: Summary

ChatGPT 5 continues to underindex main headings generally as well as names and titles of works. At the same time, it overanalyzes the work, creating excessive, needless subheadings.

While Claude 4.1 Opus is only mildly underindexing main headings and names, it continues to underindex titles of works, and—like ChatGPT 5—produces copious unneeded subheadings.

Regarding underindexing, one might ask whether, if only some topics are missing, a professional indexer could not just take an LLM-generated index and add in the missing topics.

Theoretically, yes, but the time and effort required would be cost-prohibitive. Determining which topics or names are *not* included in an existing index would require the slow and laborious process of checking the index every time any topic or name is encountered in the text to see if it's there. In addition, adding terms also potentially creates the need for cross-references (*See* or *See also*), so the indexer would need to review the index as a whole to determine whether any are warranted.

A similar issue comes up with excessive subheadings—could a professional indexer not just review all entries with subheadings and collapse them? For a chapter index, perhaps; subheadings are unlikely to be necessary for more than a few entries and in most cases could simply be deleted, with their page references collapsed into the main heading if not already present. However, for a book-length work—the typical index use case—significantly more entries are likely to *need* subheadings, so for each entry the indexer would need to manually determine which subheading phrasings would appropriately capture the work in question, by reading the corresponding page references along with any information necessary to put those pages in context, which could include the rest of the chapter or even the rest of the book. In addition, deleting terms in an index also requires verifying that those terms are not the target of cross-references elsewhere in the index; if they are, those cross-references would also have to be adjusted.

Navigability

For the white paper, we investigated how well the LLM-generated indexes handled the book's primary topic (metatopic), i.e. whether they provided an entry that served as a snapshot of the overall structure of the book and that pointed the reader out to subtopics. We also looked at whether they used cross-references effectively. Here we repeat that analysis with some slight adjustments.

Snapshot of the chapter

Individual chapters do not necessarily need to point the reader out to subtopics (in contrast to the book's metatopic, which does need to point the reader out to subtopics). The chapter we used for this paper has a relatively simple chronological structure that did not require pointing out to subtopics. The professional indexer represented that chapter metatopic as follows:

computers' precursors and history
abacus, 4–6
Babylonians and Hindus, 6–7
clocks, 7–8
early mechanical devices, 19–23
the Greeks, 8–10
Hollerith, Marquand, and Shannon, 26–28
Leibnitz, Voltaire, Pascal, and Napier, 13–14
in literature, 10, 14–19, 27–28
Lull, 11–14
modern computers, 30–31
prehistory, 1–4
symbolic logic, 23–25
World War II, 29–30

We would typically expect a professional index to include a similar entry that covered the entire chapter topic, and to which a hypothetical book metatopic entry could point. The chapter topic entry need not necessarily begin with “computers”; in a book entirely about computers, “history and precursors” would also work fine, and could easily be referenced from the hypothetical book metatopic entry (“computers: history and precursors, *see* history and precursors”).

ChatGPT

ChatGPT 5 did not include any entries beginning either with “computers” or “history”, and did not appear to contain any entry that covered the chapter metatopic. ChatGPT 4.5’s indexes did include several entries beginning with a key word from the book’s metatopic, but those covered only small chunks of text. It does not appear to be attempting to provide an entry that is a snapshot of the chapter as a whole, a starting point from which the reader can navigate to any page in the work.

Claude

Claude 4.1 Opus’s index includes an entry for “computing machines, early mechanical” whose subheadings cover 9 pages of the 31-page chapter. It included additional entries for other types of computing and computers, but only as separate main headings, not grouped together to form a snapshot. It did not include any entries beginning with “history”.

This contrasts with Claude 3 Opus’s index, which did include an entry on the book’s metatopic, although—as with here—it appeared to be less an attempt to portray a snapshot of the metatopic with access to subtopics than simply a list of where it had encountered that particular topic within the book's pages.

Cross-reference use

Indexers use cross-references (*see* and *see also*) for multiple reasons: to point the reader to subtopics (dogs: breeds, *see* breeds); to point the reader to related topics (ghosts: *see also* poltergeists); and for vocabulary control, for example when the reader may look something up under a different term than the author's preferred term (cats: *see* felines).

When an entry that is a candidate for a *see* reference has only a small number of page numbers (typically less than 5–6), indexers generally don't employ a cross-reference; instead, they **double-post**, creating an entry with the same page numbers at each location:

> cats, 15, 66, 192–198
>
> felines, 15, 66, 192–198

A *see also* reference is used when there is *additional* information about or related to the original topic at the entry to which the reference points (the target entry). If all the page numbers at the target entry are already included under the first entry, a *see also* reference is not appropriate, as there is no additional information at the target entry.

The professional index included three cross-references, each used for a different reason. The cross-reference "counters: *see also* abacus" was used because an abacus is technically a counter and a reader might look it up under that term, but the word "counter" only occurs at the very start of the abacus page range. The cross-reference "man: *see* humans" was used because the book was written when "man" was a common synonym for "human", but the indexer was anticipating a more contemporary audience who is more likely to look up the non-gendered term. Finally, the cross-reference "mathematics: *see* numbers" was used because the content about mathematics heavily overlapped with the content about numbers. The "numbers" entry included multiple kinds of content about numbers, and the indexer felt it was better to leave the subheadings present in order to clarify the meaning of the main heading (as well as to leave room for hypothetical other chapters which might also include material on numbers).

ChatGPT

The previous version of ChatGPT, 4.5, had a total of two cross-references in its two previous book indexes, both of which should have been double-posts. In contrast, ChatGPT 5 included

29 cross-references for a single chapter, almost ten times as many as the professionally created index.

Eleven of these cross-references were to targets that did not exist in the created index, a clear error. Two of those ten were also *from* entries that did not exist in the material being indexed and were actually topics of books covered in the white paper ("Affect and emotion" and "Drive (motivation)". Six were to entries whose page ranges were already covered by the original entry and were therefore redundant. Another ten were to entries whose page references should simply have been added to the original entry, rather than sending the reader to another location. One was unnecessary due to proximity (abacuses: *see* abacus), and one was unnecessary because it was from an uninverted version of the name (Ramon Lull: *see* Lull, Ramón). As mentioned above, while indexers do sometimes use cross-references to refer users from an uninverted to an inverted name (or vice versa), this is typically done only when the readers might look under the wrong one; it is not typically done for Western names in books aimed at Western audiences because it is not necessary.

In short, all of ChatGPT 5's 29 cross-references contravened indexing best practice.

Claude

The earlier version of Claude, Claude 3 Opus, had four cross-references in its index: two reasonable vocabulary-control entries used to direct a reader from a term they might look up to the terms used in the text, and two *see also* cross-references that pointed a reader from a small/specific topic to a very large topic (which is not generally done and certainly not appropriate in these instances; cross-references are typically used the other way around, to point readers from a general topic to a more specific one covered in more depth elsewhere).

In contrast, Claude 4.1 Opus's index had 16 cross-references for the chapter. Of those 16, 12 were redundant, as there were no additional page references at the target entry that weren't already in the original entry. Another was from "computing machines, early mechanical" to "specific machines"; any such machines should have been posted as subheadings under the original entry, which already covered a fair bit of ground. One was from an entry on a topic that in fact belonged to the white paper, not the chapter, which pointed to an entry that also belonged to the white paper (the deeply-ironic-in-this-context "cross-references in indexes: *See* navigability in indexing").

Another cross-reference, from Computing-Tabulating-Recording Company to the company it later became (IBM), exemplifies a common use of cross-references (to point the reader from an outdated name to a current one), although the professional indexer when faced with the same material chose not to create a similar cross-reference on the grounds that no reader would look for material about IBM under Computing-Tabulating-Recording Company.

An additional cross-reference, “calculating machines: *See* computing machines” was from one synonym to another (and to a target that didn’t exist, although there was a “computing machines, early mechanical” entry).

In summary, 14 of 16 of Claude 4.1 Opus’s cross-references contravened indexing best practices, one was questionable (the Computing-Tabulating-Recording Company entry, for reasons mentioned above), and one pointed to a nonexistent target.

Navigability: summary

Neither LLM-generated index included an entry serving as a snapshot of the chapter metatopic. In addition, all of ChatGPT’s 29 cross-references and 14 of Claude’s 16 cross-references contravened indexing best practices. This suggests that ChatGPT and Claude are unlikely to provide effective access to subtopics and related topics in a book-length work.

One might ask, could an indexer not just take an LLM-generated index and add in the necessary structure? The answer is that it would be difficult, if not impossible, to "retrofit" a poorly structured index with a good one. The reason that professionally-created indexes provide navigability is because they are based on the deep understanding of the book’s structure that indexers derive from actually working their way through the book. Indexing is an iterative process, in which the indexer is constantly adjusting and enhancing their mental map of the book's content, which is then expressed in the index's structure. Although the chapter we used here has a relatively simple and encapsulated structure, many books—especially complex scholarly works—require index structures that do not break down along simple chapter or section lines. Indexers can (and often do) make initial guesses at index structure based on (for example) reading the table of contents and the opening and closing chapters, but the process of indexing often results in the need to alter or refine that initial guess. An accurate and elegant index structure is a byproduct of the deep understanding that comes from having indexed a book, not an add-on that can be applied or overlaid later.

Accuracy

To investigate accuracy, we used the copyeditor's rule of thumb, a standard test for indexing accuracy:

> "A copyeditor's rule of thumb for checking accuracy in an index is to randomly look up 10 percent of the entries in the referenced pages. If only one or two inaccurate entries are discovered, then the editor should check another 10 percent of the entries. If no inaccuracies are found in the second group, one can hope that the inaccuracies of the first group are anomalous. But if more inaccuracies are discovered, the entire index must be checked.....If an index is full of inaccurate reference locators, it is not usable."[8]

When performing this analysis for our white paper, we uncovered numerous other types of problems that would still require remediation but do not violate the copyeditor's rule of thumb. Consequently, we use the same coding scheme here to categorize the errors we found:

- **Hallucinations:** the concept in the index did not exist at the page referenced, whether by the term given in the index or by another name. This category includes hallucinated main entries, hallucinated subheadings (even if the main entry was correct), and hallucinated page range ends (i.e., where the discussion in the text either ends well before, or extends well beyond, the end of the page range given in the index).

- **Impositions:** instances where the concept is present on the page but the LLM has imposed its own terminology over that of the author's.

- **Otherwise bad entries:** this category covers a variety of other indexing mistakes. For example, passing mentions, where the term is on the page but there is no substantive discussion; excessively long page ranges; and instances where the nature of the discussion was mischaracterized. Entries that were hallucinations or impositions, but also bad entries for some other reason, were only counted as hallucinations or impositions.

- **Adequate entries:** the concept in the entry was present on the referenced pages, the words used in the index reflected the author's use of terminology, and the entry did not have other major issues. This category included some entries that didn't have optimal

---

[8] Nancy C. Mulvany, *Indexing Books*, 2nd ed. (Chicago: University of Chicago Press, 2005), Kindle edition.

phrasing but weren't technically wrong. In other words, we evaluated LLMs' performance not against ideal indexes, but against acceptable ones.

Because all of the above (except for adequate entries) would require remediation before passing the index to a client, we report all of them; however, only hallucinations count as violations of the copyeditor’s rule of thumb.

	ChatGPT 5	Claude 4.1 Opus
Hallucinations	5	4
Impositions	7	3
Otherwise bad entries	22	30
Adequate entries	6	15
Page references checked (total page references in index)	40 (195)	52 (264)
Error rate	34/40 (85%)	36/52 (69%)

Table 1. Copyeditor’s rule of thumb results.

Both LLMs failed the copyeditor’s rule of thumb test. ChatGPT had 2 hallucinations in the first 10% of entries checked, necessitating checking another 10%, which uncovered another 3 hallucinations. Claude had 1 hallucination in the first 10% of entries, necessitating checking another 10%, which also uncovered 3 additional hallucinations.

Compared to ChatGPT 4.5’s indexes, ChatGPT 5's index had relatively fewer hallucinations, relatively more otherwise bad entries, and about the same proportion of impositions. Claude 4.1 Opus's index had relatively fewer hallucinations, relatively fewer impositions, and relatively more bad entries, compared to Claude 3 Opus’s index. When we compare error rates—the percentage of entries needing remediation, including hallucinations, impositions, and otherwise bad entries—ChatGPT 5’s error rate was 85%, versus ChatGPT 4.5’s error rates of 87% and 77%, and Claude 4.1 Opus’s error rate was 69%, versus Claude 3 Opus’s error rate of 75%.

All of these are errors that would need to be found and manually remediated before the index could be sent to a client. However, manually checking every entry in an index is a slow and time-consuming process that would take most indexers far longer than indexing the book. (This is also the reason that indexers often charge the same amount for a later edition of a book as they do for an initial edition: it is typically faster and more efficient to re-index a book than it is to manually check and update an existing index.) In research with future LLM models, we intend to empirically investigate the amount of time needed for an indexer to manually check and remediate an LLM-generated index versus to index the work in question.

Conclusion

The current generation of LLMs does not produce adequate, let alone excellent, indexes. The indexes produced by ChatGPT 5 and Claude 4.1 Opus would require substantial error-checking and remediation before being passed to a client.

As mentioned in the above sections, error-checking and remediation are time-consuming and it is often more efficient to simply index the book from scratch. Determining what is missing from an index is an arduous task that involves checking the index every time a topic or name is encountered in the text to see if it's there. Determining which subheadings are appropriate and necessary for a given entry involves reading and understanding the passages under discussion, ideally in the context of the rest of the book.

Identifying and fixing problematic entries in an existing untrustworthy index, whether they be hallucinations, impositions, or other issues, requires manually checking every entry; it is faster and easier to read and index the book correctly than it is to jump from each index entry to the corresponding spot in the book and read and understand that section without the benefit of context.

Finally, navigability in particular requires understanding each piece of information in the context of the rest of the book, and in the context of the rest of the index, and the best way to do this involves the deep understanding that comes from indexing the book. Imposing a good structure onto a poorly-structured index after the fact is extremely difficult, since a sound structure is a natural outgrowth of the mental map an indexer builds while working though a text.

For these reasons we do not think LLM-generated indexes are not adequate on their own and cannot serve as the basis or rough draft of a professionally generated index, nor do we think they are likely to be able to do so in the future.

Name indexing, however, may be an exception to this. Indexers typically capture names as part of indexing a book, but in some cases it is possible to do name indexing separately from topic indexing. The current version of Claude only minimally underindexed names, raising the question of whether—if future models of Claude continue to improve—Claude could serve as an assistant to a professional indexer by providing a name index, a possibility we will investigate with next year's model.