# CHIME: LLM-Assisted Hierarchical Organization of Scientific Studies for Literature Review Support

Anonymous ACL submission

## Abstract

Literature review requires researchers to synthesize a large amount of information and is increasingly challenging as the scientific literature expands. In this work, we investigate the potential of LLMs for producing hierarchical organizations of scientific studies to assist researchers with literature review. We define hierarchical organizations as tree structures where nodes refer to topical categories and every node is linked to the studies assigned to that category. Our naive LLM-based pipeline for hierarchy generation from a set of studies produces promising yet imperfect hierarchies, motivating us to collect CHIME, an expert-curated dataset for this task focused on biomedicine. Given the challenging and time-consuming nature of building hierarchies from scratch, we use a human-in-the-loop process in which experts *correct errors* (both links between categories and study assignment) in LLM-generated hierarchies. CHIME contains 2,174 LLM-generated hierarchies covering 472 topics, and expert-corrected hierarchies for a subset of 100 topics. Expert corrections allow us to quantify LLM performance, and we find that while they are quite good at generating and organizing categories, their assignment of studies to categories could be improved. We attempt to train a corrector model with human feedback which improves study assignment by 12.6 F1 points. We release our dataset and models to encourage research on developing better assistive tools for literature review.

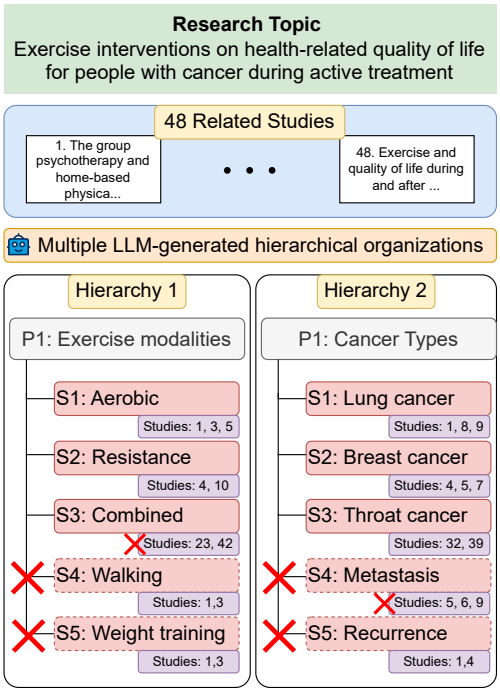


Figure 1: Given a set of related studies on a topic, we use LLMs to identify top-level categories focusing on different views of the data (such as P1 and P2), generate multiple hierarchical organizations, and assign studies to different categories. However, these categories and study assignments can contain errors. As illustrated in the figure, the categories `Walking` and `Weight training` are not coherent with their siblings ($S1 - S3$) in hierarchy 1 since they are more specific, and the categories `Metastasis` and `Recurrence` are incorrectly assigned to the parent category in hierarchy 2 since they are not types of cancer.

## 1 Introduction

Literature review, the process by which researchers synthesize many related scientific studies into a higher-level organization, is valuable but extremely time-consuming. For instance, in medicine, completing a review from registration to publication takes 67 weeks on average (Borah et al., 2017) and given the rapid pace of scholarly publication, reviews tend to go out-of-date quickly (Shojania et al., 2007). This has prompted development of tools for efficient literature review (Altmami and Menai, 2022). Most tools have focused on automating review generation, treating it as a multi-document summarization task (Mohammad et al., 2009; Jha et al., 2015; Wallace et al., 2020; DeYoung et al., 2021; Liu et al., 2023b), sometimes using intermediate structures such as hierarchies/outlines to better scaffold generation (Zhu et al., 2023), with limited success. However, recent work on assessing the utility of NLP tools like

LLMs for systematic review reveals that domain experts prefer literature review tools to be assistive instead of automatic (Yun et al., 2023).

Motivated by this finding, we take a different approach and focus on the task of generating hierarchical organizations of scientific studies to assist literature review. As shown in Figure 1, a hierarchical organization is a tree structure in which nodes represent topical categories and every node is linked to a list of studies assigned to that category. Inspired by the adoption of LLMs for information organization uses such as clustering (Viswanathan et al., 2023) and topic modeling (Pham et al., 2023a), we investigate the potential of generating hierarchies with a naive LLM-based approach, and observe that models produce promising yet imperfect hierarchies out-of-the-box.

To further assess and improve LLM performance, we collect CHIME (**C**onstructing **HI**erarchies of bio**M**edical **E**vidence), an expert-curated dataset for hierarchy generation. Since building such hierarchies from scratch is very challenging and time-consuming, we develop a human-in-the-loop protocol in which experts *correct errors* in preliminary LLM-generated hierarchies. During a three-step error correction process, experts assess the correctness of both links between categories as well as assignment of studies to categories, as demonstrated in Figure 1. Our final dataset consists of two subsets: (i) a set of 472 research topics with up to five LLM-generated hierarchies per topic (2,174 total hierarchies), and (ii) a subset of 100 research topics sampled from the previous set, with 320 expert-corrected hierarchies.

Expert-corrected hierarchies allow us to better quantify LLM performance on hierarchy generation. We observe that LLMs are already quite good at generating and organizing categories, achieving near-perfect performance on parent-child category linking and a precision of 77.3% on producing coherent groups of sibling categories. However, their performance on assigning studies to relevant categories (61.5% F1) leaves room for improvement. We study the potential of using CHIME to train "corrector" agents which can provide feedback to our LLM-based pipeline to improve hierarchy quality. Our results show that finetuning a FLAN-T5-based corrector and applying it to LLM-generated hierarchies improves study assignment by 12.6 F1 points. We release our dataset containing both LLM-generated and expert-corrected hierarchies, as well as our LLM-based hierarchy generation and correction pipelines, to encourage further research on better assistive tools for literature review.

In summary, our key contributions include:

- We develop an LLM-based pipeline to organize a collection of papers on a research topic into a labeled, human-navigable concept hierarchy.
- We release CHIME, a dataset of 2174 hierarchies constructed using our pipeline, including a "gold" subset of 320 hierarchies checked and corrected by human experts.
- We train corrector models using CHIME to automatically fix errors in LLM-generated hierarchies, improving accuracy of study categorization by 12.6 F1 points.

## 2 Generating Preliminary Hierarchies using LLMs

The first phase of our dataset creation process focuses on using LLMs to generate preliminary hierarchies from a set of related studies, which can then be corrected by experts. We describe our process for collecting sets of related studies and our LLM-based hierarchy generation pipeline.

### 2.1 Sourcing Related Studies

We leverage the Cochrane Database of Systematic Reviews[1] to obtain sets of related studies, since the systematic review writing process requires experts to extensively search for and curate studies relevant to review topics. We obtain all systematic reviews and the corresponding studies included in each review from the Cochrane website (Wallace et al., 2020). We then filter this set of systematic reviews to only retain those which: (i) are open-access (to facilitate dataset sharing), and (ii) include at least 15 and no more than 50 corresponding studies. We discard reviews with very few studies since a hierarchical organization is unlikely to provide much utility, while reviews with more than 50 studies are discarded due to the inability of LLMs to effectively handle such long inputs (Liu et al., 2023a). Our filtering criteria leave us with 472 systematic reviews (or sets), each including an average of 24.7 studies, which serve as input to our hierarchy generation pipeline.

### 2.2 Hierarchy Generation Pipeline

Prior work on using LLMs for complex tasks has shown that decomposing the task into a series of

[1] https://www.cochranelibrary.com/cdsr/reviews

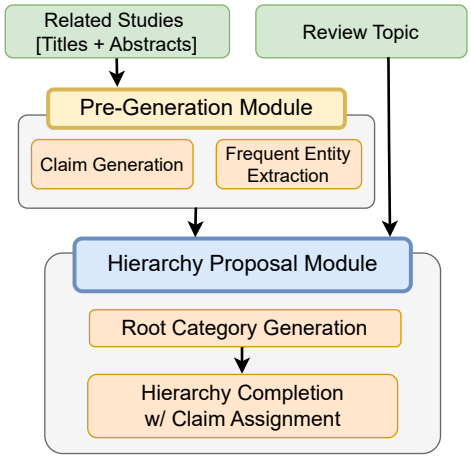


Figure 2: LLM-based pipeline for preliminary hierarchy generation given a set of related studies on a topic

steps or sub-tasks often elicits more accurate responses (Kojima et al., 2022; Wei et al., 2022b; D'Arcy et al., 2024). Motivated by this, we decompose hierarchy generation from a set of related scientific studies into three sub-tasks: (i) compressing study findings into concise claims, (ii) initiating hierarchy generation by generating *root* categories, and (iii) completing hierarchy generation by producing remaining categories and organizing claims under them. Our hierarchy generation pipeline consists of a *pre-generation module* that tackles task (i) and a *hierarchy proposal module* that handles tasks (ii) and (iii) (see Figure 2)). Additionally, our pipeline can generate multiple (up to five) potential hierarchies per topic. We describe our pipeline module in further detail below and provide complete prompt details in Appendix A.

#### 2.2.1 Pre-Generation Module

This module extracts relevant content from a set of studies to use as input for hierarchy proposal.

**Claim generation.** We generate concise claim statements from a given scientific study to reduce the amount of information provided as input to subsequent LLM modules. Providing a study abstract as input, we prompt a LLM to generate claims describing all findings discussed. We qualitatively examine the claim generation capabilities of two state-of-the-art LLMs: (i) GPT-3.5 (June 2023 version) and (ii) CLAUDE-2. Our assessment indicates that GPT-3.5 performs better in terms of clarity and conciseness; therefore we extract claim statements for all studies in our dataset using this model. Additionally, to assess whether generated claims contain hallucinated information, we run a fine-tuned DEBERTA-V3 NLI model (Laurer et al., 2024) on abstract-claim pairs. We observe that 98.1% of the generated claims are entailed by their corresponding study abstracts, indicating that claims are generally faithful to source abstracts. These sets of generated claims are provided as input to the hierarchy proposal module.

**Frequent entity extraction.** Based on preliminary exploration, we observe that simply prompting LLMs to generate hierarchies given a set of claims often produces hierarchy categories with low coverage over the claim set. Therefore, we extract frequently-occurring entities to provide as additional cues to bias category generation. We use SCISPACY (Neumann et al., 2019) to extract entities from all study abstracts, then aggregate and sort them by frequency. The 20 most frequent entities are used as additional keywords to bias generated categories towards having high coverage.

#### 2.2.2 Hierarchy Proposal Module

The aim of this module is to generate final hierarchies in two steps within a single prompt: (i) generate possible categories that can form the root node of a hierarchy (i.e., categories that divide claims into various clusters), and then (ii) generate the complete hierarchy with claim organization. For instance, considering the example in Figure 1, step (i) would produce root categories "exercise modalities" and "cancer types" and step (ii) would produce all sub-categories ($S1 - S5$) and organize studies under them (e.g., assigning studies 1, 3, 5 under S1).

**Root category generation.** With outputs from the pre-generation module and a research topic (systematic review title), we prompt the LLM to generate up to five top-level aspects as possible root categories for hierarchies.

**Hierarchy completion.** With generated root categories, this step aims to produce a complete hierarchy. We prompt the LLM to produce one hierarchy per root category, with every non-root category also containing numeric references to claims categorized under it. Note that in our setting, a claim may be assigned to multiple categories or remain uncategorized. A manual comparison of GPT-3.5 and CLAUDE-2 outputs shows that CLAUDE-2 generates root categories more relevant to the research topic and is also more accurate at correctly assigning claims, so we use this model for the hierarchy proposal module.

Using this pipeline, we generate 2,174 prelimi-

nary hierarchies (∼4.6 hierarchies per review) for our curated set of 472 systematic reviews (or sets of related studies).

## 3 Correcting Hierarchies via Human Feedback

The second phase of our dataset creation process involves correction of preliminary LLM-generated hierarchies via human feedback. Correcting these hierarchies is challenging because of two issues. First, the volume of information present in generated hierarchies (links between categories, claim-category links, etc.) makes correction very time-consuming, especially in a single pass. Second, since categories and claims in a hierarchy are interlinked, corrections can have cascading effects (e.g., changing a category name can affect which claims should be categorized under it). These issues motivate us to decompose hierarchy correction into three sub-tasks, making the feedback process less tedious and time-consuming. Furthermore, each sub-task focuses on the correction of only one category of links to mitigate cascading effects. These three sub-tasks are: (i) assessing correctness of parent-child category links, (ii) assessing coherence of sibling category groups, and (iii) assessing claim categorization.

### 3.1 Assessing Parent-Child Category Links

In this sub-task, given all parent-child category links from a hierarchy (e.g., $P1 \rightarrow S[1-5]$ in Figure 1), for each link, humans are prompted to determine whether the child is a valid sub-category of the parent. Annotators can label parent-child category links using one of the following labels: (i) parent and child categories have a hypernym-hyponym relationship (e.g., exercise modalities → aerobic exercise), (ii) parent and child categories are not related by hypernymy but the child category provides a useful breakdown of the parent(e.g., aerobic exercise → positive effects), and (iii) parent and child categories are unrelated (e.g., aerobic exercise → anaerobic exercise). Categories (i) and (ii) are positive labels indicating valid links, while category (iii) is a negative label capturing incorrect links in the existing hierarchy.

### 3.2 Assessing Coherence of Sibling Categories

For a hierarchical organization to be useful, in addition to validity of parent-child category links, all sibling categories (i.e., categories under the same parent, like $S1 - S5$ in Figure 1) should also be coherent. Therefore, in our second sub-task, given a parent and all its child categories, we ask annotators to determine whether these child categories form a coherent sibling group. Annotators can assign a positive or negative coherence label to each child category in the group. For example, given the parent category "type of cancer" and the set of child categories "liver cancer", "prostate cancer", "lung cancer", and "recurrence", the first three categories are assigned positive labels, while "recurrence" is assigned a negative label since it is not a type of cancer. All categories assigned a negative label capture incorrect groups in the existing hierarchy.

### 3.3 Assessing Claim Categorization

Unlike the previous sub-tasks which focus on assessing links between categories at all levels of the hierarchy, the final sub-task focuses on assessing the assignment of claims to various categories. Given a claim and *all* categories present in the hierarchy, for each claim-category pair, humans are prompted to assess whether the claim contains any information relevant to that category. The claim-category pair is assigned a positive label if relevant information is present, and negative otherwise. For every category, we include the path from the root to provide additional context which might be needed to interpret it accurately (e.g., "positive findings" has a broader interpretation than "chemotherapy → positive findings"). Instead of only assessing relevance of categories under which a claim has currently been categorized, this sub-task evaluates all claim-category pairs in order to catch recall errors, i.e., cases in which a claim could be assigned to an category but is not categorized there currently.

### 3.4 Feedback Process

**Data Sampling:** Due to the time-intensiveness of the correction task, we collect annotations for 100 / 472 randomly-sampled topics, and further filter out hierarchies which cover less than 30% of the claims associated with that topic. This leaves us with 320 hierarchies to collect corrections for. For the parent-child link assessment sub-task, this produces 1,635 links to be assessed. For sibling coherence, after removing all parent categories with only one child, we obtain 574 sibling groups to be assessed. Lastly, for claim categorization, the most intensive task, we end up with 50,723 claim-category pairs to label.

| | Precision | Recall | F1 |
|---|---|---|---|
| Task 1 | 0.999 | - | - |
| Task 2 | 0.773 | - | - |
| Task 3 | 0.716 | 0.539 | 0.615 |

Table 1: Performance assessment of our LLM-based pipeline on expert-curated hierarchies for 100 topics. Recall and F1 for tasks 1 and 2 cannot be measured since we only get positive predictions from the pipeline.

**Annotator Background:** We recruit a team of five experts with backgrounds in biology or medicine to conduct annotations. Two of these experts are authors on this paper, and the remaining three were recruited via Upwork.[2] Every annotator is required to first complete a qualification test, which includes sample data from all three sub-tasks, and must achieve reasonable performance before they are asked to annotate data.

**Annotation Pilots:** Given the complexity and ambiguity of our tasks, we conduct several rounds of pilot annotation with iterative feedback before commencing full-scale annotation. This ensures that all annotators develop a deep understanding of the task and can achieve high agreement. After each pilot, we measure inter-annotator agreement on each sub-task. Due to the presence of unbalanced labels in tasks 1 and 2, we compute agreement using match rate; for task 3, we report Fleiss' kappa. At the end of all pilot rounds, we achieve high agreement on all sub-tasks, with match rates of 100% and 78% on tasks 1 and 2 respectively and Fleiss' kappa of 0.66 on task 3.

### 3.5 Assessment of Preliminary Hierarchies

An additional benefit of collecting corrections for preliminary hierarchies (as described above) is that this data allows us to quantify the quality of our LLM-generated hierarchies and measure the performance of our hierarchy generation pipeline.

**Parent-child link accuracy.** Interestingly, we observe almost perfect performance on this sub-task, with only one out of 1635 parent-child links being labeled as incorrect where the pipeline put "Coffee consumption" under "Tea consumption and cancer risk". Of the remaining correct links, 75% are labeled as hypernym-hyponym links, and 25% as useful breakdowns of the parent category. This result demonstrates that LLMs are highly accurate at generating good sub-categories given a parent category, even when dealing with long inputs.

**Sibling coherence performance.** Next, we look into LLM performance on sibling coherence and observe that this is also fairly high, with 77% of sibling groups being labeled as coherent where "coherent" denotes a sibling group in which expert labels for all sibling categories are positive; otherwise, "zero." Among sibling groups labeled incoherent, we observe two common types of errors: (1) categories at different levels of granularity being grouped as siblings, and (2) one or more categories having subtly different focuses. For example, Fig. 1 demonstrates a type 1 error, where the sub-category "walking" is more specific and should be classified under "aerobic" but is instead listed as a sibling. An example of a type 2 error is the parent category "dietary interventions" with child categories "low calorie diets", "high/low carbohydrate diets", and "prepared meal plans". Here, though all child categories are dietary interventions, the first two have an explicit additional focus on nutritional value which "prepared meal plans" lacks, making them incoherent as a sibling group.

**Claim categorization performance.** The design of our claim categorization sub-task prompts annotators to evaluate the relationship between a given claim and every category in the hierarchy. Hence, when assessing whether annotators agree with the LLM's categorization of a claim under a category, we need to aggregate over the labels assigned to all claim-category pairs from the root to the target category under consideration. Formally, for a claim-category pair $(cl_i, ct_j)$, instead of only using label $l_{ij} = h((cl_i, ct_j))$ from human feedback $h$, we must aggregate over labels assigned to all ancestors of $ct_j$, i.e., $L = [h((cl_i, ct_1)), ..., h((cl_i, ct_j))]$, where $ct_1$ is the root category and $ct_j$ is the target category. We do this aggregation using an AND operation $l_{agg} = l_1 \wedge l_2 \wedge ... \wedge l_j$. After computing these aggregate labels, we observe that our LLM-based pipeline has reasonable precision (0.71), but much lower recall (0.53) on claim categorization. A low recall rate on this sub-task is problematic because, while it is easy for human annotators to correct precision errors (remove claims wrongly assigned to various categories), it is much harder to correct recall errors (identify which claims were missed under a given category), which necessitates a thorough examination of all studies.

[2] https://www.upwork.com/

## 4 Characterizing Hierarchy Complexity

Our dataset creation process produces 2,174 hierarchies on 472 research topics, with 320 hierarchies (for 100 topics) corrected by domain experts. We briefly characterize the complexity of all generated hierarchies, focusing on two aspects: (i) structural complexity, and (ii) semantic complexity.

### 4.1 Structural Complexity

**Hierarchy depth:** All generated hierarchies are multi-level, with a mean hierarchy depth of 2.5, and maximum depth of 5.

**Node arity:** On average, every parent has a node arity of 2.4 (i.e., has 2.4 child categories). However, node arity can grow as large as 10 for certain parent categories.

**Claim coverage:** Another crucial property of generated hierarchies is their coverage of claims since hierarchies containing fewer claims are easier to generate but less useful. We observe that given a set of claims, a typical hierarchy incorporates 12.3 claims on average. Additionally, very few claims from a set remain uncategorized, i.e., not covered by any generated hierarchy (2.6 on average).

These characteristics indicate that our LLM-generated hierarchies have interesting structural properties.

### 4.2 Semantic Complexity

**Category diversity:** Our dataset contains 4.6 hierarchies per research topic. We manually inspect a small sample of hierarchies for 10 research topics, and find that none of the hierarchies generated for a single topic contain any repeating categories. This signals that the multiple hierarchies we generate per topic represent semantically diverse ways of grouping/slicing the same set of claims.

**Adherence to PICO framework:** Systematic reviews in biomedicine typically use the PICO (population, intervention, comparator, outcome) framework (Richardson et al., 1995) to categorize studies. To understand how much our generated hierarchies adhere to this framework, we again inspect hierarchies for 10 research topics and label whether the root category focuses on a PICO element. We observe that 34 out of 46 hierarchies have a PICO-focused root category, making them directly useful for systematic review. Interestingly, the remaining hierarchies still focus on useful categories such as continuing patient education, study limitations, cost analyses etc. Thus, besides surfacing categorizations expected by the systematic review process, using LLMs can help discover additional interesting categorizations.

| | Task 1 | Task 2 | Task 3 |
|---|---|---|---|
| Train | 838 | 298 | 23,692 |
| Validation | 285 | 99 | 8,241 |
| Test ID | 327 | 115 | 13,595 |
| Test OOD | 185 | 62 | 5,195 |
| Total | 1,635 | 574 | 5,0723 |

Table 2: Dataset statistics for each correction sub-task

## 5 Automating Hierarchy Correction

As mentioned in §4, we hire five domain experts to correct hierarchies for 100 research topics. However, the correction process, despite our best efforts at task simplification and decomposition, is still time-consuming and requires domain expertise. Therefore, we investigate whether we can use our corrected hierarchy data to automate some correction sub-tasks. In particular, we focus on automating sibling coherence and claim categorization correction since Table 1 indicates that LLMs already achieve near-perfect performance on producing relevant child categories for a parent.

### 5.1 Experimental Setup

We briefly discuss the experimental setup we use to evaluate whether model performance on sibling coherence and claim categorization correction can be improved using our collected feedback data.

#### 5.1.1 Dataset Split

To better assess generalizability, we carefully construct two test sets, an in-domain (ID) and an out-of-domain (OOD) subset instead of randomly splitting our final dataset of 100 research topics. To develop our OOD test set, we first embed all 100 research topics by running `SPECTER2` (Singh et al., 2022) on the title and abstract of the Cochrane systematic review associated with each topic. Then, we run hierarchical clustering on the embeddings and choose one isolated cluster ($n = 12$ reviews) to be our OOD test set. Our manual inspection reveals that all studies in this cluster are about *fertility* and *pregnancy*. After creating our OOD test set, we then randomly sub-sample our ID test set ($n = 18$ reviews) from remaining research topics. This leaves us with 70 topics, which we split into training and validation sets. Detailed statistics for

our dataset splits, including number of instances for each correction sub-task, are provided in Table 2.

#### 5.1.2 Models

We evaluate two classes of methods for correction:

- **Finetuned LMs:** To assess whether correction abilities of smaller LMs can be improved by finetuning on our collected feedback data, we experiment with Flan-T5 (Chung et al., 2022), which has proven to be effective on many benchmarks.
- **Zero-Shot CoT:** To explore whether using chain-of-thought (CoT) prompting (Wei et al., 2022a) improves the ability of LLMs to do correction zero-shot without using our feedback data, we test OpenAI GPT-3.5 Turbo (`gpt-3.5-turbo-0613`) and GPT-4 Turbo (`gpt-4-1106-preview`).

Additional modeling details including CoT prompts are provided in Appendix B.

### 5.2 Correcting Sibling Coherence

Table 3 presents the performance of all models on the task of identifying sibling groups that are incoherent. Finetuning models on this task is challenging due to the small size of the training set ($n = 298$) and imbalanced labels. Despite upsampling and model selection based on precision, finetuned Flan-T5 models do not perform well on this task (best F1-score of 33.3%). Additionally LLMs also do not perform well despite the use of chain-of-thought prompting to handle the complex reasoning required for this task. At 51.5% F1, LLMs outperform finetuned models; however, their precision (46.7% for `GPT-4-Turbo`) is still not good enough to detect incoherent sibling groups confidently. These results indicate that this correction sub-task is extremely difficult to automate and will likely continue to require expert intervention.

### 5.3 Correcting Claim Categorization

Table 4 shows the performance of all models on the task of correcting assignment of claims to categories in the hierarchy. Following the strategy described in §3.5, given a claim, we first use our models to generate predictions for every claim-category pair (all category nodes) and then obtain the final label for each category by applying an AND operation over all predictions from the root category to that category. Our results show that this task is easier to automate—fine-tuning Flan-T5 on our collected training dataset leads to better scores on all metrics compared to our LLM pipeline. Crucially, recall which is much more time-consuming for humans to fix, improves by 15.9 points using Flan-T5-large indicating that automating this step can provide additional efficiency gains during correction. LLMs perform well too, with GPT-4-Turbo achieving the best recall rate among all models, but its lower precision score makes the predictions less reliable overall.

Interestingly, we notice that all models perform better on the OOD test for both correction tasks, indicating that the OOD test set likely contains instances that are less challenging than the ID set.

### 5.4 Correcting Claim Categorization for Remaining Hierarchies

Comparing the claim categorization predictions of Flan-T5-large on our test set with our LLM-based hierarchy generation pipeline reveals that it flips labels in 24.7% cases, of which 63.5% changes are correct. This indicates that a FLAN-T5-large corrector can potentially improve claim categorization of LLM-generated hierarchies. Therefore, we apply this corrector to the remaining 372 LLM-generated hierarchies that we do not have expert corrections for to improve claim assignment for those. Our final curated dataset CHIME contains hierarchies for 472 research topics, of which hierarchies for 100 topics have been corrected by experts on both category linking and claim categorization, while hierarchies for the remaining 372 have had claim assignments corrected automatically.

## 6 Related Work

### 6.1 Literature Review Support

Prior work on developing literature review support tools has largely focused on using summarization techniques for end-to-end review generation or to tackle specific aspects of the problem (see Altmami and Menai (2022) for a detailed survey). Some studies have focused on generating "citation sentences" discussing relationships between related papers, which can be included in a literature review (Xing et al., 2020; Luu et al., 2021; Ge et al., 2021; Wu et al., 2021). Other work has focused on the task of generating related work sections for a scientific paper (Hoang and Kan, 2010; Hu and Wan, 2014; Li et al., 2022; Wang et al., 2022), which while similar in nature to literature review, has a narrower scope and expects more concise generation outputs. Finally, motivated by the ever-

| | | All | | | In-domain | | | Out-of-domain | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Precision | Recall | F1 | Precision | Recall | F1 | Precision | Recall | F1 |
| **Fine-tuned** | Flan-T5 base | 0.368 | 0.179 | 0.241 | 0.364 | 0.167 | 0.229 | 0.375 | 0.200 | 0.261 |
| | Flan-T5 large | 0.333 | 0.333 | 0.333 | 0.269 | 0.292 | 0.280 | **0.462** | 0.400 | 0.429 |
| **Zero-shot CoT** | GPT-3.5 Turbo | 0.419 | **0.667** | **0.515** | 0.400 | **0.583** | **0.475** | 0.444 | **0.800** | **0.571** |
| | GPT-4 Turbo | **0.467** | 0.359 | 0.406 | **0.474** | 0.375 | 0.419 | 0.455 | 0.333 | 0.385 |

Table 3: Performance of all models on assessing sibling coherence.

| | | All | | | In-domain | | | Out-of-domain | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Precision | Recall | F1 | Precision | Recall | F1 | Precision | Recall | F1 |
| | Pipeline | 0.697 | 0.567 | 0.625 | 0.677 | 0.575 | 0.622 | 0.757 | 0.548 | 0.636 |
| **Fine-tuned** | Flan-T5-base | 0.767 | 0.711 | 0.738 | 0.750 | 0.702 | 0.725 | **0.816** | 0.735 | **0.773** |
| | Flan-T5-large | **0.779** | 0.726 | **0.751** | **0.769** | 0.726 | **0.747** | 0.807 | 0.725 | 0.764 |
| **Zero-shot CoT** | GPT-3.5 Turbo | 0.585 | 0.861 | 0.697 | 0.570 | 0.871 | 0.689 | 0.631 | 0.835 | 0.719 |
| | GPT-4 Turbo | 0.557 | **0.932** | 0.697 | 0.544 | **0.933** | 0.687 | 0.594 | **0.932** | 0.726 |

Table 4: Performance of all models on correcting claim categorization.

improving capabilities of generative models, some prior work has attempted to automate end-to-end review generation treating it as multi-document summarization, with limited success (Mohammad et al., 2009; Jha et al., 2015; Wallace et al., 2020; DeYoung et al., 2021; Liu et al., 2023b; Zhu et al., 2023). Of these, Zhu et al. (2023) generates intermediate *hierarchical outlines* to scaffold literature review generation, but unlike our work, they do not produce multiple organizations for the same set of related studies. Additionally, we focus solely on the problem of organizing related studies for literature review, leaving review generation and writing assistance to future work.

### 6.2 LLMs for Organization

Organizing document collections is an extensively-studied problem in NLP, with several classes of approaches such as clustering and topic modeling (Dumais et al., 1988) addressing this goal. Despite their utility, conventional clustering and topic modeling approaches are not easily interpretable (Chang et al., 2009), requiring manual effort which introduces subjectivity and affects their reliability (Baden et al., 2022). Recent work has started exploring whether using LLMs for clustering (Viswanathan et al., 2023; Zhang et al., 2023; Wang et al., 2023) and topic modeling (Pham et al., 2023b) can alleviate some of these issues, with promising results. This motivates us to experiment with LLMs for generating hierarchical organizations of scientific studies. Interestingly, TopicGPT (Pham et al., 2023b) also attempts to perform hierarchical topic modeling, but is limited to producing two-level hierarchies unlike our approach which generates hierarchies of arbitrary depth.

## 7 Conclusion

Our work explored the utility of LLMs for producing hierarchical organizations of scientific studies, with the goal of assisting researchers in performing literature review. We collected CHIME, an expert-curated dataset for hierarchy generation focused on biomedicine, using a human-in-the-loop process in which a naive LLM-based pipeline generates preliminary hierarchies which are corrected by experts. To make hierarchy correction less tedious and time-consuming, we decomposed it into a three-step process in which experts assessed the correctness of links between categories as well as assignment of studies to categories. CHIME contains 2,174 LLM-generated hierarchies covering 472 topics, and expert-corrected hierarchies for a subset of 100 topics. Quantifying LLM performance using our collected data revealed that LLMs are quite good at generating and linking categories, but needed further improvement on study assignment. We trained a corrector model with our feedback data which improved study assignment further by 12.6% F1 points. We hope that releasing CHIME and our hierarchy generation and correction models will motivate further research on developing better assistive tools for literature review.

## Limitations

**Single-domain focus.** Given our primary focus on biomedicine, it is possible that our hierarchy generation and correction methods do not generalize well to other scientific domains. Further investigation of generalization is out of scope for this work but a promising area for future research.

**Deployment difficulties.** Powerful LLMs like CLAUDE-2 have long inference times — in some cases, the entire hierarchy generation process can take up to one minute to complete. This makes it extremely challenging to deploy our hierarchy construction pipeline as a real-time application. However, it is possible to conduct controlled lab studies to evaluate the utility of our pipeline as a literature review assistant, which opens up another line of investigation for future work.

**Reliance on curated sets of related studies.** Our current hierarchical organization pipeline relies on the assumption that all provided studies are relevant to the research topic being reviewed. However, in a realistic literature review setting, researchers often retrieve a set of studies from search engines, which may or may not be relevant to the topic of interest, and are interested in organizing the retrieved results. It is possible that our pipeline generates lower-quality hierarchies in this setting if it is unable to distinguish relevant studies from irrelevant ones. Assessing the performance of our system in an imperfect retrieval setting presents another interesting future direction.

## References


Nouf Ibrahim Altmami and Mohamed El Bachir Menai. 2022. Automatic summarization of scientific articles: A survey. *Journal of King Saud University-Computer and Information Sciences*, 34(4):1011–1028.

Christian Baden, Christian Pipal, Martijn Schoonvelde, and Mariken AC G van der Velden. 2022. Three gaps in computational text analysis methods for social sciences: A research agenda. *Communication Methods and Measures*, 16(1):1–18.

Rohit Borah, Andrew W Brown, Patrice L Capers, and Kathryn A Kaiser. 2017. Analysis of the time and workers needed to conduct systematic reviews of medical interventions using data from the prospero registry. *BMJ open*, 7(2).

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan Boyd-Graber, and David Blei. 2009. Reading tea leaves: How humans interpret topic models. *Advances in neural information processing systems*, 22.

Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed Huai hsin Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416.

Mike D'Arcy, Tom Hope, Larry Birnbaum, and Doug Downey. 2024. Marg: Multi-agent review generation for scientific papers. *arXiv preprint arXiv:2401.04259*.

Jay DeYoung, Iz Beltagy, Madeleine van Zuylen, Bailey Kuehl, and Lucy Lu Wang. 2021. MSˆ2: Multi-document summarization of medical studies. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7494–7513, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Susan T Dumais, George W Furnas, Thomas K Landauer, Scott Deerwester, and Richard Harshman. 1988. Using latent semantic analysis to improve access to textual information. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 281–285.

Yubin Ge, Ly Dinh, Xiaofeng Liu, Jinsong Su, Ziyao Lu, Ante Wang, and Jana Diesner. 2021. BACO: A background knowledge- and content-based framework for citing sentence generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1466–1478, Online. Association for Computational Linguistics.

Cong Duy Vu Hoang and Min-Yen Kan. 2010. Towards automated related work summarization. In *Coling 2010: Posters*, pages 427–435, Beijing, China. Coling 2010 Organizing Committee.

Yue Hu and Xiaojun Wan. 2014. Automatic generation of related work sections in scientific papers: An optimization approach. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1624–1633, Doha, Qatar. Association for Computational Linguistics.

Rahul Jha, Catherine Finegan-Dollak, Ben King, Reed Coke, and Dragomir Radev. 2015. Content models for survey generation: A factoid-based evaluation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language*

*Processing (Volume 1: Long Papers)*, pages 441–450, Beijing, China. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Moritz Laurer, Wouter van Atteveldt, Andreu Casas, and Kasper Welbers. 2024. Less annotating, more classifying: Addressing the data scarcity issue of supervised machine learning with deep transfer learning and bert-nli. *Political Analysis*, 32(1):84–100.

Pengcheng Li, Wei Lu, and Qikai Cheng. 2022. Generating a related work section for scientific papers: an optimized approach with adopting problem and method information. *Scientometrics*, 127(8):4397–4417.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023a. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

Shuaiqi Liu, Jiannong Cao, Ruosong Yang, and Zhiyuan Wen. 2023b. Generating a structured summary of numerous academic papers: Dataset and method. *arXiv preprint arXiv:2302.04580*.

Kelvin Luu, Xinyi Wu, Rik Koncel-Kedziorski, Kyle Lo, Isabel Cachola, and Noah A. Smith. 2021. Explaining relationships between scientific documents. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2130–2144, Online. Association for Computational Linguistics.

Saif Mohammad, Bonnie Dorr, Melissa Egan, Ahmed Hassan, Pradeep Muthukrishan, Vahed Qazvinian, Dragomir Radev, and David Zajic. 2009. Using citations to generate surveys of scientific paradigms. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 584–592, Boulder, Colorado. Association for Computational Linguistics.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and robust models for biomedical natural language processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy. Association for Computational Linguistics.

Chau Minh Pham, Alexander Hoyle, Simeng Sun, and Mohit Iyyer. 2023a. Topicgpt: A prompt-based topic modeling framework. *arXiv preprint arXiv:2311.01449*.

Chau Minh Pham, Alexander Miserlis Hoyle, Simeng Sun, and Mohit Iyyer. 2023b. Topicgpt: A prompt-based topic modeling framework. *ArXiv*, abs/2311.01449.

W Scott Richardson, Mark C Wilson, Jim Nishikawa, and Robert S Hayward. 1995. The well-built clinical question: a key to evidence-based decisions. *ACP journal club*, 123(3):A12–A13.

Kaveh G Shojania, Margaret Sampson, Mohammed T Ansari, Jun Ji, Steve Doucette, and David Moher. 2007. How quickly do systematic reviews go out of date? a survival analysis. *Annals of internal medicine*, 147(4):224–233.

Amanpreet Singh, Mike D'Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. 2022. Scirepeval: A multi-format benchmark for scientific document representations. *ArXiv*, abs/2211.13308.

Vijay Viswanathan, Kiril Gashteovski, Carolin Lawrence, Tongshuang Wu, and Graham Neubig. 2023. Large language models enable few-shot clustering. *arXiv preprint arXiv:2307.00524*.

Byron C. Wallace, Sayantani Saha, Frank Soboczenski, and Iain James Marshall. 2020. Generating (factual?) narrative summaries of rcts: Experiments with neural multi-document summarization. *AMIA ... Annual Symposium proceedings. AMIA Symposium*, 2021:605–614.

Pancheng Wang, Shasha Li, Kunyuan Pang, Liangliang He, Dong Li, Jintao Tang, and Ting Wang. 2022. Multi-document scientific summarization from a knowledge graph-centric view. *arXiv preprint arXiv:2209.04319*.

Zihan Wang, Jingbo Shang, and Ruiqi Zhong. 2023. Goal-driven explainable clustering via language descriptions. *ArXiv*, abs/2305.13749.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. 2022a. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Jia-Yan Wu, Alexander Te-Wei Shieh, Shih-Ju Hsu, and Yun-Nung Chen. 2021. Towards generating citation sentences for multiple references with intent control. *arXiv preprint arXiv:2112.01332*.

Xinyu Xing, Xiaosheng Fan, and Xiaojun Wan. 2020. Automatic generation of citation texts in scholarly papers: A pilot study. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6181–6190, Online. Association for Computational Linguistics.

Hye Yun, Iain Marshall, Thomas Trikalinos, and Byron Wallace. 2023. Appraising the potential uses and harms of LLMs for medical systematic reviews. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10122–10139, Singapore. Association for Computational Linguistics.

Yuwei Zhang, Zihan Wang, and Jingbo Shang. 2023. Clusterllm: Large language models as a guide for text clustering. *ArXiv*, abs/2305.14871.

Kun Zhu, Xiaocheng Feng, Xiachong Feng, Yingsheng Wu, and Bing Qin. 2023. Hierarchical catalogue generation for literature review: A benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6790–6804, Singapore. Association for Computational Linguistics.

## A Prompts for Hierarchy Generation Pipeline

We present prompts for the hierarchy generation pipeline in Fig. 3 and Fig. 4.

## B Model Training Details

**Flan-T5 fintuning.** We fine-tuned the `flan-t5-base` and `flan-t5-large` models using the Hugggingface library (Wolf et al., 2019) with NVIDIA RTX A6000 for both task 1 and task 3. For task 1, the learning rate is set to 1e-3 and the batch size is 16. We train the model for up to five epochs. For task 3, the learning rate is 3e-4 with batch size 16, and the models are trained up to two epochs. Each epoch takes less than 15 minutes for both model sizes. The numbers reported for each Flan-T5 model come from a single model checkpoint.

**GPT-3.5 Turbo and GPT-4 Turbo** We perform zero-shot CoT prompting for corrector models on tasks 1 and 3 with prompts in Fig. 5 and Fig. 6.

Title:
{title}

Abstract:
{abstract}

Task:
Conclude new findings and null findings from the abstract in one sentence in the atomic format. Do not separate new findings and null findings. The finding must be relevant to the title. Do not include any other information.

Definition:
A scientific claim is an atomic verifiable statement expressing a finding about one aspect of a scientific entity or process, which can be verified from a single source.

Figure 3: Claim generation prompt for GPT-3.5 Turbo.

**Review Title**
{systematic_review_title}

Frequent entities from study abstracts:
{freq_entities}

**Study Claim List**
{claim_list}

**Instruction:**
Your task is to process a review title involving relevant clinical studies as per the following requirements:

1. **Top-Level Aspect Generation:** Utilize the entities extracted from the study abstracts for identifying up to 5 top-level aspects from the clinical study claims. You should list these aspects in a bulleted list format without incorporating any extraneous information. Cite the entities in that support the aspects. This will be the [Response 1] section.

2. **Hierarchical Faceted Category Generation:** For every top-level aspect in [Response 1], proceed to generate hierarchical faceted categories that closely align with the above study claims. The granularity of these categories must be similar to their corresponding parent categories and the siblings categories. Avoid including unrelated information. Cite the claims that support your categories. This will make up the [Response 2] section of your output.

**Remember:**
1. Precision is vital in this process; strive to avoid vague or imprecise extractions.
2. Include only relevant data and exclude any information not pertinent to the task.
3. Strictly adhere to the output format. The claims are cited in the format "(Claim 0, 2, 3, 12)" for each category and aspect.
4. The output should be in the form of a nested list using numbers.

Here is an example:

If given the review title "The efficacy of Remdesivir in treating COVID-19 patients: A review," your task output might look like this:

Frequent entities from study abstracts:
Efficacy, Remdesivir, treatment, COVID-19 patients

**Output Format**
[Response 1]:
Aspect 1: Efficacy of treatment (Efficacy)
Aspect 2: Application of Remdesivir (Remdesivir)
Aspect 3: Treatment of COVID-19 patients (treatment, COVID-19 patients)

[Response 2]:
Aspect 1: Efficacy of treatment (Claim 0, 2, 3, 12)
  1: Efficiency of alternative treatments (Claim 0, 2, 3, 12)
    1.1: Efficacy of Remdesivir (Claim 0, 12)
    1.2: Efficacy of other drugs (Claim 3)
  2: Side-effects comparison (Claim 2)
Aspect 2: Application of Remdesivir (Claim 2, 4, 5, 6, 7, 8, 9, 10, 11)
  1: Usage of other drugs (Claim 4, 5, 6, 9, 10, 11)
  2: Dosing comparisons (Claim 7, 8)
    2.1: Dosing of Remdesivir (Claim 7)
Aspect 3: Treatment of COVID-19 patients (Claim 1, 13, 14, 15, 16, 17, 18, 19, 20, 21)
  1: Treatment procedures for other diseases (Claim 13, 14, 16, 17, 18, 19, 20, 21)
  2: Treatment timeframe comparisons. (Claim 1, 15)

Figure 4: Hierarchy proposal module prompt for Claude-2.

** Instruction **
In this task, you will be annotating the relationship among a set of sibling categories.You will assess whether these sibling categories logically belong together within their shared parent category, a concept referred to as 'coherence'.

Your task is to label whether ALL sibling categories are coherent with each other.
If all sibling categories fit well and logically belongs to the broader group, label it 'These sibling categories are coherent' to signify its coherence. Make sure silbings are at the same level of granularity for coherence assessment.
If any category doesn't seem to belong logically or doesn't fit well within the group, label it 'These sibling categories are NOT coherent' to indicate non-coherence.

Your decisions should be based solely on the level of coherence – how well these categories fit together under their shared parent category and not on any other factors or personal preferences.

**Remember**
1. You should start with step-by-step reasoning and generate the answer at the end in the given format.
2. You should only reply with the answer in the format of [These sibling categories are coherent] or [These sibling categories are NOT coherent].
3. You will be given a parent category and a set of sibling categories. You should assess each sibling category independently.

Again, follow the format below to reply:

Step-by-step reasoning:

[Your reasoning]

Answer:
[These sibling categories are coherent] or [These sibling categories are NOT coherent]

** Question **
Parent category: {parent_category}

Sibling categories: {sibling_categories}

Figure 5: Prompt for task 1 sibling coherence for both GPT-3.5 Turbo and GPT-4 Turbo.

** Instruction **
In this task, your role as an annotator is to assess whether a specific claim belongs to a provided category.

Your responsibility is to assign a binary label for each category-claim pairing:
1. "The claim belongs to the category" - Choose this if any part or aspect of the claim is relevant to the category, even if the connection is broad or indirect. This includes claims that are negations or opposites of the category. See the following examples:
The claim “Assisted hatching through partial zona dissection does not improve pregnancy and embryo implantation rates in unselected patients undergoing IVF or ICSI” belongs to “Impact on specific patient groups” category because patient groups can be applied to not only patient demographics but also patients with the same disease/symptom.
The claim “Sumatriptan is effective in reducing productivity loss due to migraine, with significant improvements in productivity loss and return to normal work performance compared to placebo.” belongs to “Headache relief” because headache is one of the symptoms of migraine even though it is not explicitly mentioned in the claim.

2. "The claim does NOT belong to the category" - Choose this if there is no meaningful connection between the claim and the category.

**Remember**
1. Only reply with the answer in the format of [The claim belongs to the category] or [The claim does NOT belong to the category].
2. Do not reply with any other format.
3. Start with step-by-step reasoning and generate the answer at the end in the given format.

**Claim**
{claim}

**Category**
{category}

Again, follow the format below to reply:

Step-by-step reasoning:
[Your reasoning]

Answer:
[The claim belongs to the category] or [The claim does NOT belong to the category]

Figure 6: Prompt for task 3 claim assignment for both GPT-3.5 Turbo and GPT-4 Turbo.