Towards Automating Articles Screening Processes Using Chain-of-Thought Large Language Models

Carlo Arpini*
University of Milano-Bicocca
Department of Statistics and Quantitative Methods
Milan, Italy
c.arpini3@campus.unimib.it

Mirko Cesarini†*
University of Milano-Bicocca
Department of Statistics and Quantitative Methods
Milan, Italy
mirko.cesarini@unimib.it

Emmanuele Lotano*
University of Milano-Bicocca
Department of Statistics and Quantitative Methods
Milan, Italy
e.lotano@campus.unimib.it

Abstract

The rapid expansion of the biomedical literature has made manual screening for meta-analyses increasingly hard. While automation attempts exist, they often rely on active-learning powered binary classification, which is still time intensive; or simple summarization, frequently struggling with factual consistency and completeness of the information reported. This work presents a novel pipeline using Reasoning-Large Language Models (Chain-of-Thought) to automate the summarization of informative elements from full-text biomedical documents, focusing on PIO elements (Patient/Population, Intervention, and Outcome) in a human-in-the-loop scenario.

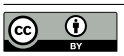
PDF documents are segmented into structured XML semantic chunks. Then, the pipeline employs a Large Language Model, where several prompts are adaptively used based on the semantic complexity of the chunks, measured using Effective Rank. Consensus-based filtering mechanisms are then used to synthesize multiple candidate responses into a single, high-fidelity summary.

We evaluated the pipeline on the EBM-NLP dataset using multiple LLMs. Results demonstrate that an 8B parameter model, when integrated into our pipeline, achieves (BERTScores) semantic precision exceeding 0.88 and (BERTScores) semantic recall exceeding 0.91, rivaling much larger models. A subsequent human evaluation by expert researchers confirmed the high factuality and completeness of the extracted information. These findings suggest that reasoning-enhanced pipelines are a promising tool to significantly reduce screening time while maintaining the semantic consistency required for rigorous evidence synthesis.

CCS Concepts

• **Computing methodologies** → **Information extraction**; *Artificial intelligence*; *Natural language processing*; • **Applied computing** → Health care information systems.

*All authors contributed equally to this work.
†Corresponding author.



WWW Companion '26, Dubai, United Arab Emirates.

ACM ISBN 979-8-4007-2308-7/2026/04
https://doi.org/https://doi.org/10.1145/3774905.3795601

Keywords

Large Language Models (LLMs), Chain-of-Thought (CoT), Information Extraction (IE), Evidence-Based Medicine (EBM), Biomedical Meta-analyses.

ACM Reference Format:
Carlo Arpini, Mirko Cesarini, and Emmanuele Lotano. 2026. Towards Automating Articles Screening Processes Using Chain-of-Thought Large Language Models. In *Companion Proceedings of the ACM Web Conference 2026 (WWW Companion '26), April 13–17, 2026, Dubai, United Arab Emirates.* ACM, New York, NY, USA, 10 pages. https://doi.org/https://doi.org/10.1145/3774905.3795601

1 Introduction

In recent years, the use of meta-analyses has steadily increased, particularly in the biomedical field. This trend has been driven in part by the surge in scientific publications during the COVID-19 pandemic [30], a growth that continues to this day.

Unfortunately, this rapid development has made even harder the most time-consuming phase of meta-analyses, the screening of documents. This phase requires considerable cognitive resources and increasingly long analysis time [11]. Due to these difficulties, the document screening phase is usually conducted by a well-organized team of researchers. The process begins with a quick read of the retrieved article titles, aimed at immediately excluding the irrelevant articles. Next, the researchers focus on the abstracts of the remaining documents; the two steps (title and abstract screening) are often combined to simplify the activity. Finally, the remaining documents are read by the researchers. It is in this last stage that the researchers make the final decision on which documents to include in the meta-analysis [48]. The proposed approach has been built for supporting this specific process, but can be used or easily adapted for different workflows.

The approach described above, although being tuned by decades of practice, presents several pitfalls. The main disadvantage is the time it takes. Depending on the meta-analysis to be performed, the number of initial articles can be in the tens of thousands. Reading the documents during the final screening phase contributes to the large effort required. Moreover, the most relevant information is often located in specific sections or subsections, making the task even more burdensome for researchers [43].

1.1 State of the Art

Initial attempts to automate screening relied on traditional machine learning models for text classification [9][16]. These methods frame the problem as a binary classification task, where a model is trained

to distinguish between relevant and irrelevant articles based on features extracted from the text (e.g., TF-IDF vectors [39]). While foundational, these supervised learning techniques often require a large, pre-labeled dataset for training, which is impractical in the context of a new meta-analysis where labeled data are scarce. To overcome this limitation, the state of the art quickly became using Active Learning (AL) approaches. Although these methods may now appear simple or limited, they are still among the most widely used tools in the scientific community [44].

On the other hand, a different approach is to summarize articles as a whole. Automatic evaluation remains dominated by n-gram overlap metrics (e.g., ROUGE or BLUE), but these metrics correlate imperfectly with factual correctness and informativeness for downstream scientific use [24].

Recently, Large Language Models (LLMs) have introduced new possibilities for generating complex and coherent summaries [5], guided by a user prompt that steers the summarization according to the research interests. Modern text summarization spans extractive and abstractive paradigms (single- and multi-document), together with a growing emphasis on long-context and retrieval-aware methods; a recent comprehensive survey charts this evolution and highlights datasets, evaluation practices, and open challenges [52].

Abstractive and LLM-based summarizers offer greater fluency and compression but are particularly prone to factual inconsistency and/or omission of critical details, problems that are especially risky in evidence synthesis and biomedical screening.

Furthermore, LLMs are prone to hallucinations[1] [18], e.g., answers that are not relevant with respect to the given prompt, or that contain the original prompt or parts of it. To address this, careful filtering of the generated replies can be employed [27].

Moreover, even when providing the same prompt multiple times to the same LLM, the responses may vary each time: even in so called "deterministic" settings, variability remains [4]. This applies to "Reasoning" models too, as they are specific versions of causal LLMs; however, the inclusion of reasoning Chain-of-Thought (CoT) steps tends to make responses more accurate and, consequently, to a certain extent more reproducible. To mitigate this variability, strategies inspired by the idea of "wisdom of the crowd" [50] can be adopted, generating multiple output sequences, and evaluating them using appropriate scoring mechanisms [47].

Another critical element is prompt engineering. The performance of LLMs is highly sensitive to how a request is formulated, with even subtle changes in wording potentially leading to vastly different outcomes that affect output quality. After all, the very concept of CoT models stems from structuring a prompt to explicitly guide the model's reasoning process [49]. This sensitivity presents a significant challenge for specialized tasks where consistency and reliability are paramount. Different prompting techniques and their outcome have been explored [7], showing that the model's behavior and the output consistency can be fine tuned.

Finally, LLM-based approaches are usually slower with respect to other machine learning techniques, due to the high computational cost, which becomes even more burdensome when using CoT enabled models.

[1]Gibberish text with no coherence or factual inconsistencies, birthed from the probabilistic nature of LLMs.

1.2 Proposal

This work focused on creating and testing a Reasoning-LLMs pipeline to extract summary information to aid researchers in selecting biomedical publications within a human-in-the-loop framework. The pipeline will produce a summary of each analyzed paper, containing all the information required by a researcher to make a decision whether to include or not the paper in the meta-analysis.

The research question is whether the proposed pipeline can effectively support and empower researchers during a meta-analysis.

The results obtained, initially validated using an annotated corpus and subsequently evaluated by a group of expert researchers, confirm the robustness and effectiveness of the proposed approach. The methodologies developed and the analyses carried out offer a contribution towards the implementation of Reasoning-LLM-based pipelines in meta-analysis workflows, opening up future perspectives for improving and accelerating the screening process in the biomedical field.

2 Methods

In this work, Chain-of-Thought (CoT) Large Language Models (LLMs) have been used to summarize PIO (Patient or Population, Intervention, and Outcome) elements [36], in a human-in-the-loop scenario. Ideally, the researcher might want to look for further topics. Nevertheless, as a preliminary analysis, we focus on a predefined set of elements. We simulate that an ideal researcher wants to summarize an article based only on PIO elements. This approach offers several advantages:

- Although PIO values do not cover every detail of a biomedical study, they provide a standardized and comparable representation that captures most of the key elements of a meta-analysis;
- PIO values are usually defined in a clear way and help reduce the subjectivity typical of other forms of textual synthesis. Furthermore, the existence of annotated datasets, such as the EBM-NLP dataset [32], provides a common base for evaluation;
- The actual pipeline is agnostic w.r.t. the target topic. Given it actually works on three distinct topics (the PIO elements), it could be easily extended towards covering additional topics.

The complete methodology will be fully outlined in what follows, detailing both the information extraction strategy and architecture design.

2.1 Parsing and Preprocessing Documents

Much of the scientific literature is published in PDF (Portable Document Format). The complex layouts typical of these documents pose significant difficulties for automatic information extraction. To address this challenge, this work employs a first stage parsing approach that transforms the PDF into a structured representation compatible with subsequent stages of textual analysis. One of the most effective tools currently available is GROBID (GeneRation Of BIbliographic Data) [1], a framework based on machine learning techniques widely used in the scientific domain: it produces a

structured TEI/XML[2] representation that integrates textual content and layout information, thus facilitating the identification of text blocks and making the document more interpretable for subsequent modules in the pipeline [26] [37].

The use of GROBID greatly simplifies many operations, but it also introduces noise: artifacts such as tables interpreted as text, bibliographic references, citations, links, and emails may not be properly managed in the XML output, compromising the quality of the extraction. For this reason, the pre-processing phase is a crucial step: it aims to clean up the parser output, normalize sections, and eliminate elements that generate noise while providing no relevant information for PIO elements extraction.

The structured XML format produced by GROBID is divided in blocks, which map to sections and paragraphs that correspond to text units already created by the article author(s) (therefore, being autonomous and coherent), thus providing logical and semantic boundaries useful for dividing the text. In this context, the textual content length can be a problem: while most modern LLMs have large context windows that can fit documents of any size, the attention mechanism's effectiveness of modern models tends to be inconsistent as the length of the context increases [25]; on the other hand, excessively short text may not contain enough information. For this reason, the strategy adopted aims to achieve an optimal balance to preserve relevant semantic units while respecting a predefined maximum token limit, optimized to generate text blocks of significant length, neither excessively long nor too short. To this end, the solution implemented is based on a three-stage segmentation algorithm designed to preserve the logical boundaries of the article and obtain chunks of a "useful" length for processing. Specifically, the starting point is the parsed XML (from hereafter called "granular" XML) containing all the sections and subsections identified by the parser. From that, a reduced version (from hereafter called "structured" XML) is constructed, so that it contains only the "main" sections, identified through a predefined list of headings: Abstract, Introduction, Methods, Results, Discussion, and Conclusions. The decision to focus on these sections is justified by the IMRaD format (Introduction, Methods, Results, and Discussion), widely recognized as the editorial standard for medical and scientific literature [41]. The Abstract and Conclusions have also been included, according to a common practice in academic scientific journals [29]. The parser, even if it fails to reconstruct the subsection boundaries, still groups the text in a meaningful way, retrieving the content of the IMRaD sections. Hence, this initial step produces two distinct XMLs: the original parsed "granular" XML with all subsections and the "structured" XML, the latter having the so-called A+IMRaD+C main sections.

Next, the segmentation process is divided into three main phases, each one contributing to make the text blocks consistent and of a suitable length. The process, applied starting from the "structured" XML, can be summarized as follows:

(1) First, a next-fit algorithm [20] is applied, whose objective is to group by concatenation adjacent (main) sections until a predefined maximum length in terms of tokens is reached. This approach preserves the logical order of the biomedical article and ensures a good balance between the length of the text block and the original structure of the text itself.
(2) Second phase: some sections might be too long (compared to the desired maximum number of tokens).
   - When possible, a long subsection is divided considering its internal subsections; then the next-fit algorithm is run again on the subsections to minimize the number of fragmentations;
   - If, on the other hand, subdivision into subsections is not possible because they are absent or they themselves are excessively long, segmentation based on overlapping token chunks is used. This solution, mimicking already known techniques [10], reduces the loss of context that would result from a clean cut "in medias res" of the text.
(3) Third phase, removal of residual text portions: finally, as a last step, portions that, after the previous steps, are too short to be of any use are eliminated to reduce the presence of segments with little information.

2.2 Effective Rank and Prompting

Once the section chunks have been obtained through the preprocessing explained in subsection 2.1, each chunk is processed by the model with custom prompts. These prompts follow a known schema [7], consisting of an initial part that introduces the task to be performed, the name of the section chunk (which usually stores information about the origin of the text, for example whether it comes from a union of sections or a specific subsection), the actual text chunk from which to extract the information, and finally a concluding reinforcement that reiterates the initial instructions. A new technique is used to modulate the "degree of reasoning" required by the prompt: the main objective is to guide it towards more articulated reasoning when the text chunk is semantically rich, and to encourage it to be concise and willing to admit the absence of information when the chunk appears semantically poor, to mitigate the fact that LLMs tend to hallucinate by making up information when that specific information is absent from text [51].

To achieve this (prompt) modulation, it was necessary to quantify the notion of "semantic complexity" of a chunk of text. Simple indicators, such as token length, are often misleading: very long texts can deal with a single concept in a verbose manner, just as shorter texts can condense several concepts. To overcome this limitation, we introduce a metric to capture the semantic complexity of a given text. Specifically, each chunk text is represented as a matrix of contextual embeddings. The (contextual) embedding vectors of each token were stacked, obtaining a matrix $\mathbf{M} \in \mathbb{R}^{n \times d}$, with n equal to the number of tokens in text and d equal to the size of the embedding space. The matrix $\mathbf{M}$ will feature a distribution concentrated along a few directions in case of semantically related concepts, or a multidirectional distribution when the text includes semantically different concepts.

To identify the semantic complexity, the *Effective Rank* is introduced. Intuitively, the Effective Rank can be thought as the "effective number of distinct concepts" present in a section chunk: values close to 1 indicate content focused on a few semantic directions, while higher values indicate the presence of greater conceptual

[2]The Text Encoding Initiative (TEI) provides guidelines for representing texts in digital form, most commonly implemented in XML [42].

richness. This metric has proven to be more informative than simple token length: although there is a correlation between the two measures, it is not uncommon to find very long sections with a low Effective Rank, as they are entirely dedicated to very few concepts. The phenomenon is particularly evident in biomedical texts, where a succession of sections and subsections can exceed hundreds of tokens while dealing with a single topic.

Focusing on the effective rank computation, Singular Value Decomposition (SVD) is applied to the embedding matrix $\mathbf{M}$; the spectrum of singular values describes the relative importance of the different semantic directions present in the text chunk: a concentrated spectrum indicates that the meaning is captured by a few dimensions, while a more distributed spectrum reveals the presence of a greater variety of semantic concepts.

To make this information numerically interpretable, the spectrum of singular values is normalized, and Shannon entropy is calculated on the obtained distribution. The resulting metric, the entropy-based Effective Rank [38], provides a robust measure of the actual number of active distinct semantic dimensions present in a text chunk.

Formally, given the matrix $\mathbf{M} \in \mathbb{R}^{n \times d}$ and its singular values $\sigma_1, \sigma_2, \ldots, \sigma_r$ (with $r = \text{rank}(\mathbf{M})$), the entropy-based Effective Rank is:

$$\text{Eff.Rank}(\mathbf{M}) = \exp\left(- \sum_{i=1}^{r} \frac{\sigma_i}{\sum_{k=1}^{r} \sigma_k} \log \frac{\sigma_i}{\sum_{k=1}^{r} \sigma_k} \right) \tag{1}$$

which lies in the interval $[1, r]$.

To exploit the Effective Rank results[3], three custom prompts were created in the workflow, which differ slightly in the way they formulate instructions to the model: the prompt associated with text chunks with a high Effective Rank explicitly invites step-by-step and in-depth reasoning (with a more articulated Chain-of-Thought); the prompt for text chunks with low Effective Rank encourages faster reading and the possibility of declaring the absence of the requested information; finally, an intermediate prompt covers cases of moderate semantic complexity. The thresholds identified to distinguish the three classes were determined experimentally.

2.3 Filtering and Multiple Answers

Once the responses to the prompt have been collected, the hallucination problem and the reproducibility issues has to be dealt with. Therefore, a filtering step is performed on the answers.

To balance these aspects, this work developed a filtering pipeline that combines techniques based on both regular expressions and embedding-based approaches that exploit cosine similarity [3]. The former allows for the rapid exclusion of clearly anomalous responses, while the latter offers finer control over semantic content. The order has been designed to progressively discard all unwanted responses, starting with the most basic checks and moving on to the most sophisticated ones.

The following filters were used:

- First, removal of Chain-of-Thought (CoT);
- Second, verifying that at least one specific target word, appears in the response generated by the model;
- Third, verifying the formal consistency of the response with respect to the way in which the model was asked to respond[4];
- Fourth, the response is discarded if it is too short and it is also systematically truncated if too long, at the end of the third sentence;
- Fifth, embedding-based filter; both the initial instruction prompt and the candidate response are converted into sentence-level embeddings using mean pooling [35]. Next, the cosine similarity between the two vectors is calculated. If the value obtained is excessively high the two texts are too similar; it is likely that the model is simply repeating the instructions provided in the prompt without adding new informative content, and the response is therefore discarded.

If a response passes through the entire filtering chain, it can be considered valid. Next, a further distinction is made between "informative" responses and "null" responses (i.e., all responses that explicitly state that the information is *not* present in the text). To make this distinction, an additional embedding-based procedure is applied:

- First, the embedding of the candidate response is calculated;
- An average embedding is constructed from five variants with the same meaning as the phrase "No {TARGET} is present in text", where {TARGET} corresponds to one of the PIO elements;
- The cosine similarity between the two embeddings is then compared: if the cosine similarity is sufficiently high, the response is classified as "null".

It is worth noting that the choice of thresholds to determine whether the cosine similarity is "sufficiently high" was determined empirically through experimental observations. These values may vary significantly depending on the model or embedding technique used, and need to be fine-tuned on a case-by-case basis.

To tackle the problem of reproducibility, the following approach was used: multiple replies are generated from the same prompt. Then, based on the replies' concordance, one final reply is selected (we tackle this problem in detail in subsection 2.4). This approach, as shown in earlier works [47], can be used to improve the reliability of the model's answers and serve as a way to screen out incorrect answers, enhancing the model's factual consistency too.

The integration of filtering and multiple replies is implemented through an iterative refinement. For each section chunk, the model generates a pool of candidate responses; those failing to meet the filtering criteria are discarded and re-generated. This iterative loop continues until a predefined threshold of valid responses, hereinafter referred to as F, for each section chunk. Then, accordance is checked, as outlined in subsection 2.4. To limit the possibility of infinite loops, a maximum limit of 10 regenerations has been set for each section chunk.

2.4 Final Model Reply Extraction

Assuming that an article contains N section chunks, the process results in a total of $N \cdot F$ candidate responses that must be summarized in a single unified sentence for each target.

[3]For reasons of numerical stability, any zero σ_i value is replaced with a small value $\varepsilon > 0$ before calculating the logarithm.

[4]E.g., the reply should start with either "The {TARGET} is..." or "No {TARGET} is...", with {TARGET} being one of the PIO elements.

2.4.1 Consensus on Information Target. Before addressing the problem of summarizing multiple responses in a single sentence, it is appropriate to start by assessing whether the target information required is indeed present or not in the considered paper. To validate this, majority voting [47] based on the frequency of so-called "null" responses, is used. Namely, the percentage of "null" responses out of the total is calculated, and if this percentage exceeds a predefined threshold, it is concluded that the article does not contain the desired information. In this work, the threshold of 66% was empirically chosen; i.e., the information is considered effectively absent if at least $\frac{2}{3}$ of the responses are considered "null".

Once it has been assessed that the required information is indeed present in the article, the pipeline considers only valid responses for the following steps.

2.4.2 Creation of Summary replies. At this point, the challenge is to condense the set of valid responses into a single concise "summary" sentence that encapsulates all the necessary information about the target PIO element in the most semantically accurate manner.

Creating this summary is by no means trivial. While each section contributes with the same number of responses, not all sections are equally important for the type of information sought. For example, the "Study Participants" section is usually very relevant for identifying patients or population, but plays a marginal role in describing the outcomes of the research study. Ideally, an effective scoring mechanism should take into account not only this different relevance, but also the relationships between sections, their order, and other structural information that, for a human reader, naturally emerges from the document structure.

To accomplish this, Large Language Models (LLMs) prove again to be effective, thanks to their attention mechanism [46].

An approach structured in two phases was used:

- In a first phase, using a task-tuned prompt and through a new call to the model, K intermediate "summary responses" are produced by asking the model to summarize the previously produced replies, keeping track of section names and document structure; this ensures that all the information about the target in the F replies is considered and weighted accordingly to its origin within the document and the overall document structure;
- Subsequently, these K responses undergo a selection process, in which a mathematical score (see subsubsection 2.4.3) identifies the most semantically appropriate one.

This repeated approach has been evaluated effective and offers several advantages. Firstly, the generation of multiple summary intermediate responses strengthens the conceptual consistency and reproducibility of the process. Furthermore, the filters previously introduced are reapplied to these intermediate responses, once again ensuring semantic consistency and lexical accuracy.

Finally, as a last preventive measure, a maximum limit of 10 regeneration cycles has been set to avoid potentially infinite loops in the process of generating and filtering these summary sentences.

The only disadvantage is the increase in the number of calls to the model and, consequently, in processing times; however, this computational cost is fully justified in terms of the accuracy and reliability of the final result.

2.4.3 Mathematical Scoring. The final stage of the pipeline aims to select the "best" among the K summary responses. The average semantic representation at the sentence level is computed by averaging the contextual embeddings of the summary responses.

This average semantic representation is used for selection purposes: the cosine similarity between it and each candidate sentence embedding is computed; the response of which the embedding is closest to the average will be the one most semantically aligned with the average meaning the group of summary responses has.

Although simple and elegant, this selection criterion has a limitation: vector averaging tends to mitigate the specific details of individual responses, reducing the ability to preserve potentially important details, especially in a biomedical context. To counteract this effect, Effective Rank can be used again to weigh the "complexity" of the generated sentences S_i. The Effective Rank is computed from the matrix of stacked embeddings of the individual tokens that make up a single response. It quantifies the degree of semantic richness of the sentence itself.

The overall score assigned to each generated response S_i will therefore be defined as:

$$\text{Score}(S_i) = \left(\frac{\vec{S}_i \cdot \overline{\vec{S}}}{\|\overline{\vec{S}}\|_2 \, \|\vec{S}_i\|_2} \right) \cdot \frac{\text{Eff.Rank}(\mathbf{S}_i)}{\max_j \left(\text{Eff.Rank}(\mathbf{S}_j) \right)} \tag{2}$$

Where the various components are:

- $\vec{S}_i \in \mathbb{R}^d$ is the sentence embedding vector of candidate summary answer S_i (sentence-level embedding achieved through mean pooling);
- $\overline{\vec{S}} = \frac{1}{K} \sum_{l=1}^{K} \vec{S}_l$ is the mean sentence embedding across the S summary answers;
- $\mathbf{S}_i \in \mathbb{R}^{m \times d}$ is the matrix of contextual token embeddings for sentence S_i (stacked m token embeddings);
- $\text{Eff.Rank}(\mathbf{S}_i)$ is the Effective Rank computed as in Equation 1.

The idea proposed in this work is therefore to combine two aspects: on one hand, the cosine similarity term, which quantifies semantic proximity to the average meaning; on the other hand, the normalized Effective Rank term, which assigns greater weight to the response with the greatest possible semantic complexity.

3 Discussion and Results

As already mentioned in [24] [13] [6], automatic evaluation of short summaries is usually computed via n-gram overlap metrics which struggle with abstractive summarization (e.g., synonyms, different sentence structures, rhetorical figures).

3.1 Axis of Evaulation

To make the assessment as objective as possible, three quantitative metrics were selected to analyze the pipeline's performance according to complementary perspectives. Specifically, these are:

(1) Semantic precision and semantic recall based on token-level embeddings. These are calculated as:

$$\text{S-Precision}(c, r) = \frac{1}{|c|} \sum_{x \in c} \max_{y \in r} \cos(\mathbf{e}(x), \mathbf{e}(y)) \tag{3}$$

and

$$\text{S-Recall}(c, r) = \frac{1}{|r|} \sum_{y \in r} \max_{x \in c} \cos(\mathbf{e}(y), \mathbf{e}(x)) \tag{4}$$

Where each term is:

- c: set of tokens in the candidate sentence;
- r: set of tokens in the reference sentence;
- $\mathbf{e}(x)$: contextual embedding of token x;
- $\cos(\mathbf{e}(x), \mathbf{e}(y))$: cosine similarity between the embeddings.

These two metrics come from BERTScore [54]. The intuition behind these metrics is to measure the semantic similarity between the candidate and reference sentences, even when they are phrased differently. In fact, while the pipeline generates coherent and complete sentences, the annotations in the EBM-NLP dataset are only single words, sentence fragments, or short n-grams. Therefore, the added value of the embeddings in matching different phrased texts is paramount. The embedding model of choice is `microsoft/BiomedNLP-BiomedBERT-base-uncased-abstract-fulltext` [15].

(2) Error rate related to missing information: since the EBM-NLP dataset consists solely of biomedical articles labeled with P, I, and O annotations, whenever the pipeline used decides that such information is missing, it is reasonable to assume that this is an error. Consequently, for each of the three PIO elements, the number of times the information is found absent, divided by the number of documents, represents a failure rate that should ideally be equal to zero.

(3) Cross-similarity based on sentence-level embedding: finally the cross-similarity between pairs of sentences P, I, and O, for the same document, calculated as the cosine similarity between sentence-level embeddings:

$$\text{Cross Similarity}(S_x, S_y) = \frac{\vec{S_x} \cdot \vec{S_y}}{\|\vec{S_y}\|_2 \, \|\vec{S_x}\|_2} \tag{5}$$

where $\vec{S_x}$ and $\vec{S_y}$ represent the embeddings of two sentences between P, I, and O. This measure allows one to evaluate the degree of semantic overlap between the three different categories. Ideally, the information is expected to be as atomic as possible, reflected in cosine similarity values towards zero. For this calculation, therefore, the `NeuML/pubmedbert-base-embeddings` [31] model[5] was used.

These measures should not be considered exhaustive in the absence of an assessment by expert human researchers [28] [22]. Hence, a study was conducted on a subset of 16 documents selected from the EBM-NLP dataset, aimed at verifying the quality perceived by researchers on the PIO elements extracted from the model.

Specifically, the main steps were:

(1) Initially, the full-text pipeline produced the sentences containing the PIO summary information.

(2) At this point, 12 researchers participated in the evaluation phase, analyzing 4 full-text documents, so that each article was read by 3 different researchers. Approximately 42% of the researchers had already used automated meta-analysis tools (ASReview [45], ChatGPT and Rayyan [33]); in addition, 50% of the researchers had conducted between 2 and 10 meta-analyses, while only one researcher had completed more than 10 meta-analyses during their working career.

(3) For each PIO element, the researchers assessed the completeness[6] and factuality[7] of the output on a 5-point Likert scale [23]. They also provided textual feedback for each PIO element assessed.

These metrics, all together provide a more complete picture than any single one of them, and help assess the goodness of any extracted PIO sentence.

3.2 Automatic Evaluation Results

The above described pipeline is completely model-agnostic, nothing of the above defined techniques is tied to a specific provider or version.

As such, a sample of 5 publicly available models[8] were chosen:

- *FP16-DeepSeek-R1-Distilled-LLaMa3-8B* was used as "base" model to test and develop the described pipeline;
- *FP16-DeepSeek-R1-Distilled-LLaMa3-70B* was used as a benchmark to evaluate how metrics scale with parameters;
- *Q4-DeepSeek-R1-Distilled-LLaMa3-8B*, the quantized version [19] of the base model, was used to benchmark quantization effects on metrics;
- *Qwen3-30B-Thinking-2507* was released at the end of July 2025 and is equipped with CoT capabilities;
- *GPT-OSS-120B_Reasoning_High* was released in early August 2025 in 20B and 120B variants [2]; the 120B model tested, in particular, is a Mixture of Experts (MoE) model [40].

One by one each model was loaded in a llama.cpp [14] server [8], running on an M3 Mac Studio with 512 GB of unified memory, and was tested with the metrics defined in subsection 3.1 on a random sample of 100 documents that were both part of the EBM-NLP corpus and had no paywall restrictions.

At last, all models were tested with the same configuration of pipeline hyperparameters: $F = 3$ and $K = 3$, chosen mainly to limit processing times.

Table 1: Failure Rate by Different Models

Model	P	I	O
Q4_DS_R1_Distilled_8B	0.02	**0.01**	0.01
FP16_DS_R1_Distilled_8B	0.01	**0.01**	**0.00**
Qwen3_30B_Thinking_2507	0.04	0.06	0.09
GPT_OSS_120B_Reasoning_High	**0.00**	0.03	**0.00**
FP16_DS_R1_Distilled_70B	**0.00**	0.03	**0.00**

Starting from the failure rates, Table 1 outlines very different performance among the models. The Qwen model performed poorly: it has almost a 10% failure rate for outcome. Then, there is a clear loss

[5] This latter model is a modified version of [15], tuned for sentence level embeddings.

[6] By degree of agreement with the sentence "The generated sentence provides complete and sufficient information about the study {TARGET}".

[7] By degree of agreement with the sentence "The generated sentence, regarding the {TARGET}, does not contain inaccurate or unsupported information (e.g., errors, distortions, or hallucinated content)".

[8] The model selection was performed as of late August 2025. Testing of the models took place in September 2025.

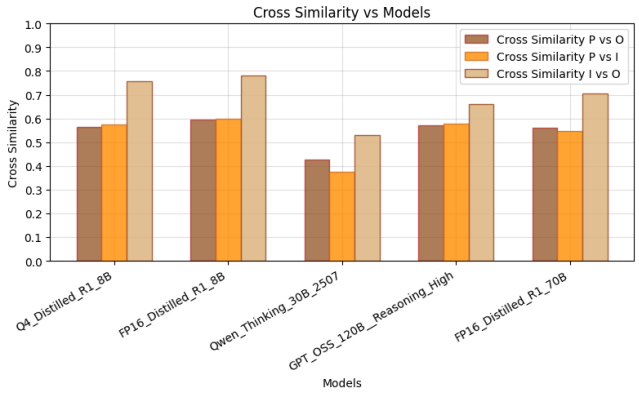


Figure 1: Cross similarity of generated P, I, and O descriptions (Equation 5), averaged over 100 documents.

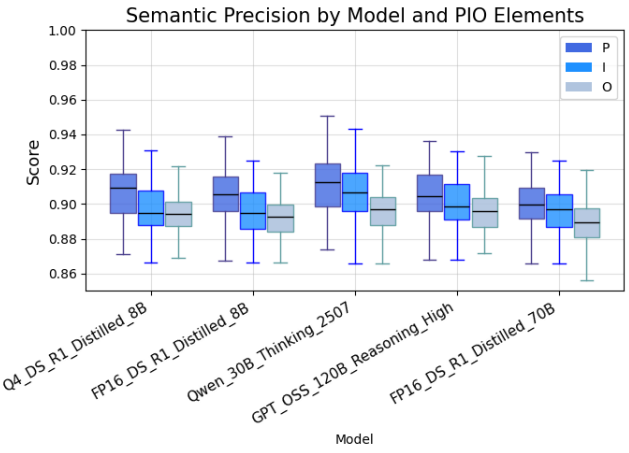


Figure 2: S-Precision values for P, I, and O extraction computed based on annotations from 100 documents, randomly drawn by the EBM-NLP dataset [32].

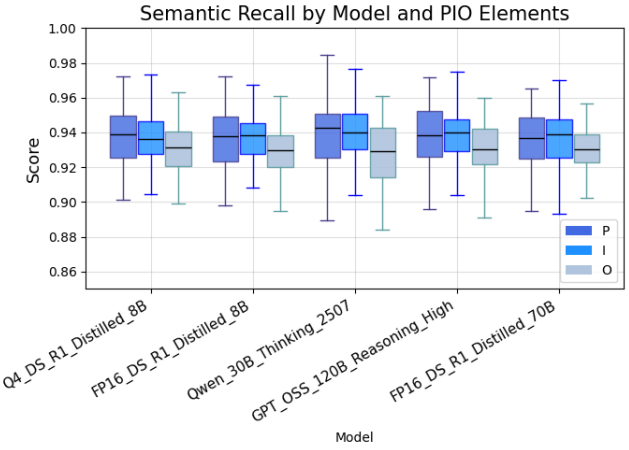


Figure 3: S-Recall values for P, I, and O extraction computed based on annotations from 100 documents, randomly drawn by the EBM-NLP dataset [32].

in accuracy due to the quantization with respect to the base model, as the quantized version performed more poorly overall. Next, it is worth noticing that the base version of the model performed great across the board, even compared to its 70B variant, proving that the quality of results is not strictly tied to parameter scaling laws [17], but it is greatly influenced by the design of the pipeline. To conclude, the GPT OSS model is also very promising as it too achieved a very strong performance, while being much faster and less costly thanks to its MoE architecture [34].

Moving on to cross similarity (see Figure 1), some differences arise again between the models. First and foremost, the Qwen model has very low cross similarities, indicating a good amount of atomicity in its replies. But this is probably influenced by the high failure rates shown in Table 1. As far as what concerns the other models, it can be seen that the quantized version of the base model essentially behaves like the base one, and instead the 70B variant is more atomic in its information extraction. The most important result comes from the GPT OSS model, showing even better atomicity in its results, especially for I versus O.

Next, looking at Figure 2 and Figure 3, the semantic precision and semantic recall values confirm a fairly consistent picture. It is worth noting that s-precision and s-recall (see Equation 3 and Equation 4) are computed on a single document considering the multiple annotations; each data point within the box-plots is a single document's s-precision and s-recall. Moreover, by construction, these metrics tend to have high values due to the max operator: in the literature [54], values defined as "good" (in the sense of high semantic overlap) are around 0.9, while "poor" values rarely fall below 0.8 (more in Appendix A). On average, for all models, s-precision seems to be a bit lower than s-recall; s-recall values are typically above 0.9, for each of the PIO elements, reaching as high as 0.95, depending on the model. Interestingly, it also seems that P has generally higher values than I, which in turn has higher values than O, for both s-precision and s-recall. For O values, this is reflected in the literature even for annotated data [55].

Looking more specifically at the differences between models across both s-precision and s-recall, GPT OSS and 70B Distilled R1 on average performed worse on P and I, while being slightly better for O, with respect to the Qwen model. This is interesting as failure rates (see Table 1) suggested a different behavior: one would expect the Qwen model to have the poorest performance overall.

Focusing on s-precision, either the base model or its quantized version achieved highest scores; and the same happens with s-recall.

To conclude, considering quality, while nuanced differences in failure rates and output atomicity do exist, the overall quality as measured by s-precision and s-recall did not show significant variation across the models. This suggests that for this specific task, the 8B base model offers an optimal balance of speed and performance. Conversely, the Qwen model was the least overall interesting one considered. Moreover, the GPT-OSS model proves very interesting in terms of both time and quality, thanks to its different architecture that still preserves reasoning capabilities.

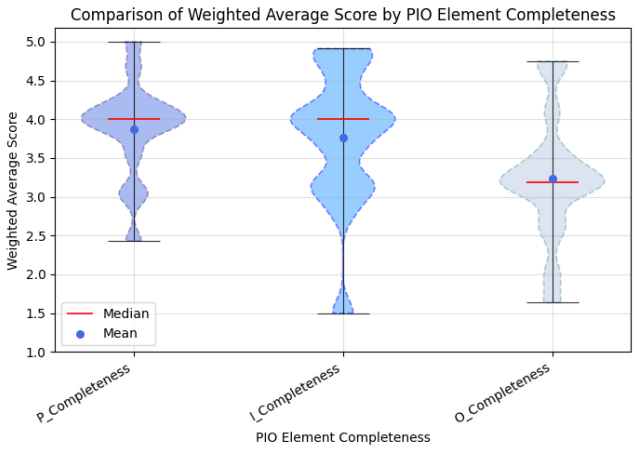


Figure 4: Violin plots of weighted average completeness scores for PIO elements in the sample of 16 documents.

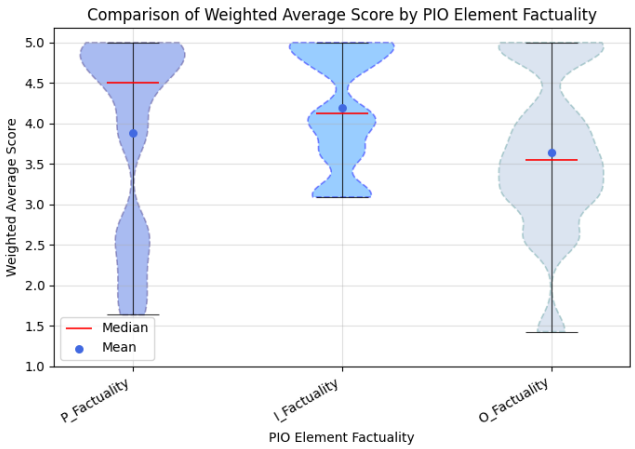


Figure 5: Violin plots of weighted average factuality scores for PIO elements in the sample of 16 documents.

3.3 Human Assessment

Data was obtained via Google Form thanks to the help of 12 researchers, who replied on a 5-point Likert scale [23] to questions about three articles each. A simple but effective approach was adopted to analyze the data: for each article, the evaluations of the various researchers were aggregated using a weighted average, in which the weight assigned to each researcher was linearly proportional to their previous experience with the number of meta-analyses completed in their career. Specifically, weights of 1 were assigned for fewer than 2 meta-analyses, 5 for between 2 and 10, and 10 for more than 10 meta-analyses completed in their career.

Figure 4 shows, with violin plots, the distribution of completeness scores for the 16 documents for the PIO elements. It can be seen that, for both P and I, at least half of the distribution has values around 4, with a general tendency toward higher scores for I. However, the average value for I is slightly reduced due to a single article in which the model incorrectly stated "No Intervention is present in text.". The O element, on the other hand, has a more dispersed distribution across the entire scale.

Figure 5 shows the distribution of factuality scores: the higher the score, the fewer errors or hallucinations the model created during the inference of PIO sentences. Once again, there are some interesting aspects to note. The results generally appear higher than those for completeness, with average values above 4 for the P element and the I element, and the absolute absence of scores below 3 for Intervention. This confirms the effectiveness of the filtering techniques introduced in subsection 2.3. The O element, on the other hand, while showing lower values (mainly between 3 and 4), shows acceptable internal consistency, suggesting, however, possible room for improvement.

4 Conclusion and Future Work

This work presented a novel, model-agnostic pipeline to automate the extraction of key information (Patient/Population, Intervention, and Outcome) from full-text scientific documents using Chain-of-Thought (CoT) Large Language Models (LLMs).

Unlike previous approaches that rely on binary classification or simple abstract summarization, our methodology integrates structural parsing with semantic analysis. By introducing the *Effective Rank* metric, we successfully modulated the reasoning depth of the LLM based on the semantic complexity of the text, optimizing both computational resources and output quality. Furthermore, the implementation of a consensus-based filtering and scoring mechanism addressed the inherent stochasticity of LLMs, ensuring that the final summaries are both coherent and factually robust.

Our evaluation, conducted on the EBM-NLP dataset using multiple model architectures, demonstrated that an optimized 8B parameter model can achieve BERTScores of semantic precision exceeding 0.88 and semantic recall exceeding 0.91, rivaling the performance of significantly larger 70B models. These results were further corroborated by expert human assessment, which confirmed high degrees of completeness and factuality in the extracted information, particularly for Patient and Intervention elements. While the extraction of Outcome remains more of a challenge, the pipeline is promising for developing a tool to reduce the screening burden on researchers.

This work demonstrates that the integration of rigorous preprocessing, mathematical complexity metrics, and iterative reasoning can bridge the gap between raw LLM capabilities and the strict reliability requirements of evidence-based medicine.

As a future avenue for evaluation, a comparison with other methods on the EBM-NLP dataset will be added. Future work will focus on generalizing the pipeline to accommodate ad-hoc information targets specified by researchers. Furthermore, we aim to generate performance curves to benchmark the pipeline against traditional tools, providing researchers with explicit metrics on time-saving versus sensitivity.

Acknowledgments

This research was financially supported by the Italian Ministry of University and Research (PRIN 2022Y7HHNW).

References

[1] 2008–2025. GROBID. https://github.com/kermitt2/grobid. swh:1:dir:dab86b296e3c3216e2241968f0d63b68e8209d3c

[2] Sandhini Agarwal and et al. 2025. gpt-oss-120b & gpt-oss-20b Model Card. *arXiv preprint arXiv:2508.10925* (2025). Open-weight reasoning models from OpenAI, Apache 2.0 license..

[3] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International conference on learning representations*.

[4] Berk Atil, Sarp Aykent, Alexa Chittams, Lisheng Fu, Rebecca J Passonneau, Evan Radcliffe, Guru Rajan Rajagopal, Adam Sloan, Tomasz Tudrej, Ferhan Ture, et al. 2024. Non-determinism of "deterministic" llm settings. *arXiv preprint arXiv:2408.04667* (2024).

[5] Lochan Basyal and Mihir Sanghvi. 2023. Text Summarization Using Large Language Models: A Comparative Study of MPT-7b-instruct, Falcon-7b-instruct, and OpenAI Chat-GPT Models. arXiv:2310.10449 [cs.CL] https://arxiv.org/abs/2310.10449

[6] Manik Bhandari, Pranav Gour, Atabak Ashfaq, and Pengfei Liu. 2020. Metrics also disagree in the low scoring range: Revisiting summarization evaluation metrics. *arXiv preprint arXiv:2011.04096* (2020).

[7] Christian Cao, Jason Sang, Rohit Arora, Robbie Kloosterman, Matt Cecere, Jaswanth Gorla, Richard Saleh, David Chen, Ian Drennan, Bijan Teja, et al. 2024. Prompting is all you need: LLMs for systematic review screening. *medRxiv* (2024), 2024–06.

[8] Nick Crews. 2025. llama-cpp-server-python: Bootstrap a server from llama-cpp in a few lines of Python. https://github.com/NickCrews/llama-cpp-server-python. Commit: 00cc5ece8783848139d41fb7f9c5e5c9b7a62686. MIT License.

[9] Dina Demner-Fushman and Jimmy Lin. 2007. Answering clinical questions with knowledge-based and statistical techniques. *Computational Linguistics* 33, 1 (2007), 63–103.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 4171–4186.

[11] Giorgio Maria Di Nunzio. 2025. POLAR: Policy Optimization for Literature Analysis under Review Constraints. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management* (Seoul, Republic of Korea) *(CIKM '25)*. Association for Computing Machinery, New York, NY, USA, 4712–4716. doi:10.1145/3746252.3760834

[12] Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. *arXiv preprint arXiv:1909.00512* (2019).

[13] Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics* 9 (2021), 391–409.

[14] Georgi Gerganov. 2023. llama.cpp. https://github.com/ggerganov/llama.cpp.

[15] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. https://huggingface.co/microsoft/BiomedNLP-BiomedBERT-base-uncased-abstract-fulltext. arXiv:arXiv:2007.15779 Model: microsoft/BiomedNLP-BiomedBERT-base-uncased-abstract-fulltext.

[16] Marie J Hansen, Nana Ø Rasmussen, and Grace Chung. 2008. A method of extracting the number of trial participants from abstracts describing randomized controlled trials. *Journal of Telemedicine and Telecare* 14, 7 (2008), 354–358.

[17] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556* (2022).

[18] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys* 55, 12 (2023), 1–38.

[19] Renren Jin, Jiangcun Du, Wuwei Huang, Wei Liu, Jian Luan, Bin Wang, and Deyi Xiong. 2024. A Comprehensive Evaluation of Quantization Strategies for Large Language Models. arXiv:2402.16775 [cs.CL] https://arxiv.org/abs/2402.16775

[20] David S Johnson. 1973. Approximation algorithms for combinatorial problems. In *Proceedings of the fifth annual ACM symposium on Theory of computing*. 38–49.

[21] Jin Li, Yiyan Deng, Qi Sun, Junjie Zhu, Yu Tian, Jingsong Li, and Tingting Zhu. 2024. Benchmarking large language models in evidence-based medicine. *IEEE Journal of Biomedical and Health Informatics* (2024).

[22] Ying Li, Surabhi Datta, Majid Rastegar-Mojarad, Kyeryoung Lee, Hunki Paek, Julie Glasgow, Chris Liston, Long He, Xiaoyan Wang, and Yingxin Xu. 2025. Enhancing systematic literature reviews with generative artificial intelligence: development, applications, and performance evaluation. *Journal of the American Medical Informatics Association* 32, 4 (2025), 616–625.

[23] Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of psychology* (1932).

[24] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*. Association for Computational Linguistics, 74–81.

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

[26] Patrice Lopez. 2009. GROBID: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In *International conference on theory and practice of digital libraries*. Springer, 473–474.

[27] Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896* (2023).

[28] Mette Motzfeldt Jensen, Mathias Brix Danielsen, Johannes Riis, Karoline Assifuah Kristjansen, Stig Andersen, Yoshiro Okubo, and Martin Grønbech Jørgensen. 2025. ChatGPT-4o can serve as the second rater for data extraction in systematic reviews. *PloS one* 20, 1 (2025), e0313401.

[29] PK Ramachandran Nair and Vimala D Nair. 2014. Organization of a research paper: The IMRAD format. In *Scientific writing and communication in agriculture and natural resources*. Springer, 13–25.

[30] Gabriela F Nane, Nicolas Robinson-Garcia, François van Schalkwyk, and Daniel Torres-Salinas. 2023. COVID-19 and the scientific publishing system: growth, open access and scientific fields. *Scientometrics* 128, 1 (2023), 345–362.

[31] NeuML. 2024. NeuML/pubmedbert-base-embeddings. https://huggingface.co/NeuML/pubmedbert-base-embeddings. Sentence-transformers fine-tuned version of PubMedBERT for embeddings.

[32] Benjamin Nye, Junyi Jessy Li, Roma Patel, Yinfei Yang, Iain J Marshall, Ani Nenkova, and Byron C Wallace. 2018. A Corpus with Multi-Level Annotations of Patients, Interventions and Outcomes to Support Language Processing for Medical Literature. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 197–207.

[33] Mourad Ouzzani, Hossam Hammady, Zbys Fedorowicz, and Ahmed Elmagarmid. 2016. Rayyan—a web and mobile app for systematic reviews. *Systematic reviews* 5, 1 (2016), 210.

[34] Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. 2022. DeepSpeed-MoE: Advancing Mixture-of-Experts Inference and Training to Power Next-Generation AI Scale. arXiv:2201.05596 [cs.LG] https://arxiv.org/abs/2201.05596

[35] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 3982–3992. doi:10.18653/v1/D19-1410

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

[37] Laurent Romary and Patrice Lopez. 2015. Grobid-information extraction from scientific publications. *ERCIM News* 100 (2015).

[38] Olivier Roy and Martin Vetterli. 2007. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*. IEEE, 606–610.

[39] Gerard Salton and Chris Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management* 24, 5 (1988), 513–523. doi:10.1016/0306-4573(88)90021-0

[40] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V Le, and Geoffrey Hinton. 2017. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In *Proceedings of ICLR*.

[41] Luciana B Sollaci and Mauricio G Pereira. 2004. The introduction, methods, results, and discussion (IMRAD) structure: a fifty-year survey. *Journal of the medical library association* 92, 3 (2004), 364.

[42] TEI Consortium. 2022. *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. https://tei-c.org/release/doc/tei-p5-doc/en/html/ Version 4.5.0.

[43] Guy Tsafnat, Paul Glasziou, Miew Keen Choong, Adam Dunn, Filippo Galgani, and Enrico Coiera. 2014. Systematic review automation technologies. *Systematic reviews* 3, 1 (2014), 74.

[44] Rens van de Schoot, Jonathan de Bruin, Raoul Schram, et al. 2020. ASReview: Active Learning for Systematic Reviews. https://github.com/asreview. Accessed: 2025-09-10.

[45] Rens Van De Schoot, Jonathan De Bruin, Raoul Schram, Parisa Zahedi, Jan De Boer, Felix Weijdema, Bianca Kramer, Martijn Huijts, Maarten Hoogerwerf, Gerbrich Ferdinands, et al. 2021. An open source machine learning framework for efficient and transparent systematic reviews. *Nature machine intelligence* 3, 2 (2021), 125–133.

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[47] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. arXiv:2203.11171 [cs.CL] https://arxiv.org/abs/2203.11171

[48] Xiao-Meng Wang, Xi-Ru Zhang, Zhi-Hao Li, Wen-Fang Zhong, Pei Yang, and Chen Mao. 2021. A brief introduction of meta-analyses in clinical practice and research. *The Journal of Gene Medicine* 23, 5 (2021), e3312.
[49] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 24824–24837. https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf
[50] Sheng Kung Michael Yi, Mark Steyvers, Michael D Lee, and Matthew J Dry. 2012. The wisdom of the crowd in combinatorial problems. *Cognitive science* 36, 3 (2012), 452–470.
[51] Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? *arXiv preprint arXiv:2305.18153* (2023).
[52] Haopeng Zhang, Philip S. Yu, and Jiawei Zhang. 2025. A Systematic Survey of Text Summarization: From Statistical Methods to Large Language Models. 57, 11, Article 277 (June 2025), 41 pages. doi:10.1145/3731445
[53] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675* (2019).
[54] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations (ICLR)*.
[55] Yiliang Zhou, Abigail M Newbury, Gongbo Zhang, Betina Ross Idnay, Hao Liu, Chunhua Weng, and Yifan Peng. 2025. EvidenceOutcomes: a Dataset of Clinical Trial Publications with Clinically Meaningful Outcomes. *arXiv preprint arXiv:2506.05380* (2025).

A Appendix: On the Interpretability of S-Precision and S-Recall

Recent benchmarks on EBM-NLP often approach PIO (Patient or Population, Intervention, and Outcome) extraction as a sequence labeling (NER) task, reporting F1 scores in the 0.45–0.60 range based on token-level overlap [21]. These F1 scores (and hence precision and recall too) are computed on the extracted NER spans versus the tagged reference annotations. In contrast, our pipeline performs abstractive extraction, generating coherent sentences that may not overlap lexically with the sparse, fragmented annotations of EBM-NLP, but effectively convey the same meaning. Precisely because of this, a standard computation of precision and recall would disproportionately penalize valid abstractive summaries.

A similar limitation applies to traditional summarization metrics: ROUGE-L measures n-gram overlap, which correlates imperfectly with factual correctness when different terms convey the same concept or when sentence structures vary.

BERTScore's semantic precision and semantic recall (see Equation 3 and Equation 4), on the other hand, measure the degree of semantic similarity, enabling a fairer comparison between abstractive summaries and annotated fragments. However, a caveat remains. As highlighted in [53] and recalled in subsection 3.2, these metrics should not be interpreted on the same scale as traditional precision and recall. Mostly due to the max operator and partly due to the typical anisotropy of the embedding space (the problem of representation degeneracy) [12], raw scores are naturally biased upwards. This was reported in the original BERTScore article, where semantic precision and recall values for random pairs of sentences often hover just above 0.8.

To validate that our reported scores represent a statistically significant signal and not just domain noise, we performed a percentile rescaling analysis on the outputs of the base model (i.e. *FP16-DeepSeek-R1-Distilled-LLaMa3-8B*).

Following the methodology in [53], we established a "Noise Baseline" distribution by computing semantic precision and recall between the model's generated answers for a document D_i and the ground truth annotations of all mismatched documents D_j (where $i \neq j$), for a given P, I, or O target. This simulates the score achieved by a model that hallucinates plausible biomedical text unrelated to the specific document content. We then transformed the raw scores (as reported in Figure 2 and Figure 3) into percentile ranks relative to this noise distribution. A percentile of 0.5 indicates performance equivalent to random biomedical noise, while values approaching 1.0 indicate high specificity to the source document.

Figure 6 illustrates rescaled percentile scores.

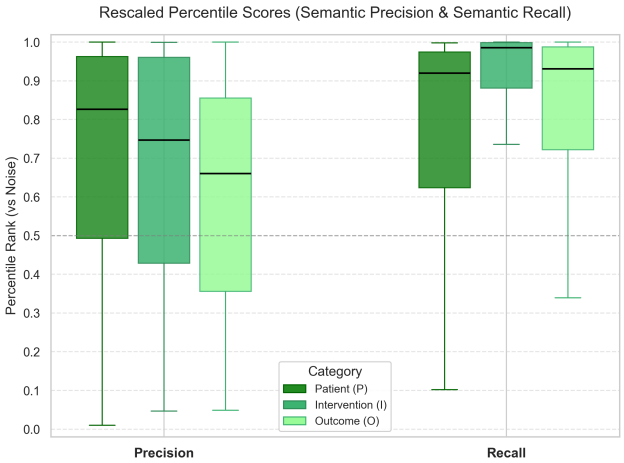


Figure 6: Distribution of Rescaled Percentile Scores for the 8B Model; random chance line is in dotted in gray.

The analysis reveals two crucial insights:

(1) The pipeline is particularly strong in recall of Intervention details; Patient and Outcome's recall values only seldom drift lower than random chance. This confirms that the raw scores correspond to a genuine ability of the pipeline to extract specific, document-relevant information that is statistically distinguishable from general biomedical text.
(2) The pipeline is instead less strong in terms of precision, something that is to be expected considered the verbose nature of Large Language Models (LLMs). Particularly in the case of Outcome, the rescaled precision drops significantly. This aligns with findings in the literature suggesting that outcomes are the most difficult elements to extract precisely [55]. While the model performs better than random noise, the "high" raw score for outcomes is indeed partly inflated by the embeddings' anisotropy, masking a higher variance in performance compared to P and I.

This rescaling validation confirms that our pipeline provides strong recall overall (especially for Intervention scores), while precision is still robust with the exception of the Outcome extraction, which remains a more nuanced challenge.

Accepted 29 January 2026