Advancing Healthcare Automation: Multi-Agent System for Medical Necessity Justification

Himanshu Pandey
RISA Labs
himanshu@risalabs.ai

Akhil Amod
RISA Labs
akhil@risalabs.ai

Shivang
RISA Labs
shivang@risalabs.ai

Abstract

Prior Authorization delivers safe, appropriate, and cost-effective care that is medically justified with evidence-based guidelines. However, the process often requires labor-intensive manual comparisons between patient medical records and clinical guidelines, that is both repetitive and time-consuming. Recent developments in Large Language Models (LLMs) have shown potential in addressing complex medical NLP tasks with minimal supervision. This paper explores the application of Multi-Agent System (MAS) that utilize specialized LLM agents to automate Prior Authorization task by breaking them down into simpler and manageable sub-tasks. Our study systematically investigates the effects of various prompting strategies on these agents and benchmarks the performance of different LLMs. We demonstrate that GPT-4 achieves an accuracy of 86.2% in predicting checklist item-level judgments with evidence, and 95.6% in determining overall checklist judgment. Additionally, we explore how these agents can contribute to explainability of steps taken in the process, thereby enhancing trust and transparency in the system.

1 Introduction

In US healthcare, management of administrative workflows represents a pivotal yet formidable challenge. Physicians, nurses, and administrative personnel frequently allocate a substantial portion of their working hours to these procedural tasks, distracting from their primary focus on patient care. One such workflow, Prior authorization (PA) is a healthcare management process used by insurance entities to determine whether a proposed treatment or service is covered under a patient's plan before it is approved to be carried out. This process applies to various treatments and services, including medications, imaging, and procedures (Madhusoodanan et al., 2023). Evaluating a PA application involves assessing medical necessity of patient-specific health records against prevailing coverage guidelines. A major part of these coverage guidelines are clinical guidelines which are systematically developed statements designed to help practitioners make decisions about appropriate health care for specific clinical circumstances. Insurance companies review these clinical guidelines to to justify medical necessity of a procedure or treatment (Chambers et al., 2016).

While Prior Authorization ensures safe, appropriate, cost-effective and evidence based care to all members (Jones et al., 2021), it is a major source of physician and staff burnout as well as job dissatisfaction.There are several ongoing efforts to improve the prior authorization process. High-profile innovations include (1) "gold carding" providers, exempting those who have very high historical approval rates; and (2) automating the process through e-prior auth (e-PA) (Lenert et al., 2023). e-PA proposes a set of transactions conveying the rules for approval in a standardized query representation in CQL. While such rule based methods are adequate for simple authorization decisions, complex cases with temporal data, evidence of responses and trends in clinical data items can be difficult to represent in CQL's rule based format (Lenert et al., 2023). Also, a nationwide survey (Salzbrenner et al., 2022) identified that the use of e-PA was not associated with less provider time or fewer challenges in preparing and submitting PA requests. However, the use of e-PA reported a shorter PA decision time. Additionally, there is an understanding that AI can potentially improve the current state of PA filing (Lenert et al., 2023).

The introduction of Large Language Models (LLMs) (OpenAI, 2024; Touvron et al., 2023) has catalyzed a transformative shift in the capabilities of artificial intelligence, enabling the resolution of complex challenges previously inaccessible to conventional AI methods. LLMs excel in interpret-

Proceedings of the 23rd Workshop on Biomedical Language Processing, pages 39–49
August 16, 2024. ©2024 Association for Computational Linguistics

ing and synthesizing large volumes of unstructured data, enhancing tasks such as natural language understanding (Yang et al., 2024), sentiment analysis, and automated content creation (Zhou et al., 2024). Building on this foundation, *Multi-Agent Systems*, which employs a collective of AI-powered agents, represents an even further advancement (Guo et al., 2024). This approach decomposes a singular complex task into multiple, manageable sub-tasks and distributes them across multiple agents, each specialized through training for a sub-task. Following this methodology essentially infuses a microservice architecture into the traditional monolithic AI framework, enabling more modular, scalable, and robust AI systems. By integrating the depth and adaptability of LLMs with the collaborative and dynamic nature of Multi-Agent Systems, AI systems can achieve unprecedented levels of performance and versatility across various complex problems (Guo et al., 2024; He et al., 2024).

In this paper, we investigate the application of multi-agent systems for determining medical necessity for a medical procedure. Our contributions are as follows:

- We propose a novel challenge of establishing medical necessity for prior authorizations (PAs) by reasoning on clinical guidelines against patient medical records.
- We decompose the problem statement of PA filing into intermediate sub-tasks, which can then be effectively solved by LLM Agents.
- We demonstrate through extensive experimentations the effect of LLM choice and prompting strategies. Specifically, GPT-4 achieves an accuracy of 86.2% in predicting checklist item-level judgments and 95.6% in determining overall checklist judgment.

2 Related Work

Large Language Models (LLMs) have completely changed the landscape of Natural Language Processing (NLP) in the recent years. LLMs have shown *emergent abilities* (Wei et al., 2022a) in settings like few-shot prompting (Brown et al., 2020) and augmented prompting strategies. Augmented prompting like Chain of Thought (CoT) (Wei et al., 2022b) and Automatic Chain of Thought (Zhang et al., 2022) prompting enables LLMs to solve reasoning tasks using step by step approach. Additionally, instruction fine-tuning with human feedback has made LLMs able to respond to instructions describing unseen tasks (Ouyang et al., 2022). Other advancements include techniques like self-consistency (Wang et al., 2023) which helps LLMs solve complex tasks using multiple different ways of thinking and prompt gradient descent (Pryzant et al., 2023) which edits prompt in the opposite semantic direction of the gradient to boost prompt's performance. Building on this, more dynamic and complex tasks can be tackled by LLM powered Multi Agent Systems (LLM-MAS). These LLM-MAS have collaborative autonomous agents equipped with unique strategies and behaviour (Guo et al., 2024). This agentic behaviour is based on the idea that LLMs can improve in game-play scenario by using previous experiences and feedback (Fu et al., 2023; Madaan et al., 2023).

LLMs have the potential to disrupt medicine. Models like Med-PaLM (Singhal et al., 2022) outperformed state of the art on all MultiMedBench tasks (Tu et al., 2024). GPT-4 has consistently outperformed task-specific fine-tuned models and is comparable to human experts on QA datasets (Zhou et al., 2024). GPT-4 scored 86.65% in United States Medical Licensing Examination (USMLE) where passing percentage was 60% (Nori et al., 2023). It also demonstrates GPT-4's capacity for reasoning about concepts tested in USMLE challenge problems, including explanation, counterfactual reasoning, differential diagnosis, and testing strategies. Some recent researches have started to explore the impact of LLMs in discharge summary generation (Ellershaw et al., 2024; Williams et al., 2024), care planning (Nashwan and Hani, 2023; Jung et al., 2024), Electronic Health Records (EHRs) (Cui et al., 2024; Ahsan et al., 2023). Text-to-SQL parsing has attracted significant interest (Li et al., 2024). Building on this idea, numerous research efforts, such as EHRSQL (Lee et al., 2022), are focused on extracting data from EHRs. Additionally, there are ongoing efforts to develop solutions for EHR-based question-answering tasks (Shi et al., 2024).

However, the domain of PA filing is largely untouched by LLMs mainly because of lack of publicly available data despite the understanding that AI can potentially improve its current state (Lenert et al., 2023). While some efforts have been made to automate PA filing, for example (Diane et al., 2023) where ChatGPT is utilised to generate PA letters for Orthopedic Surgery Practice, but the process lacks the important step of establishing medical ne-

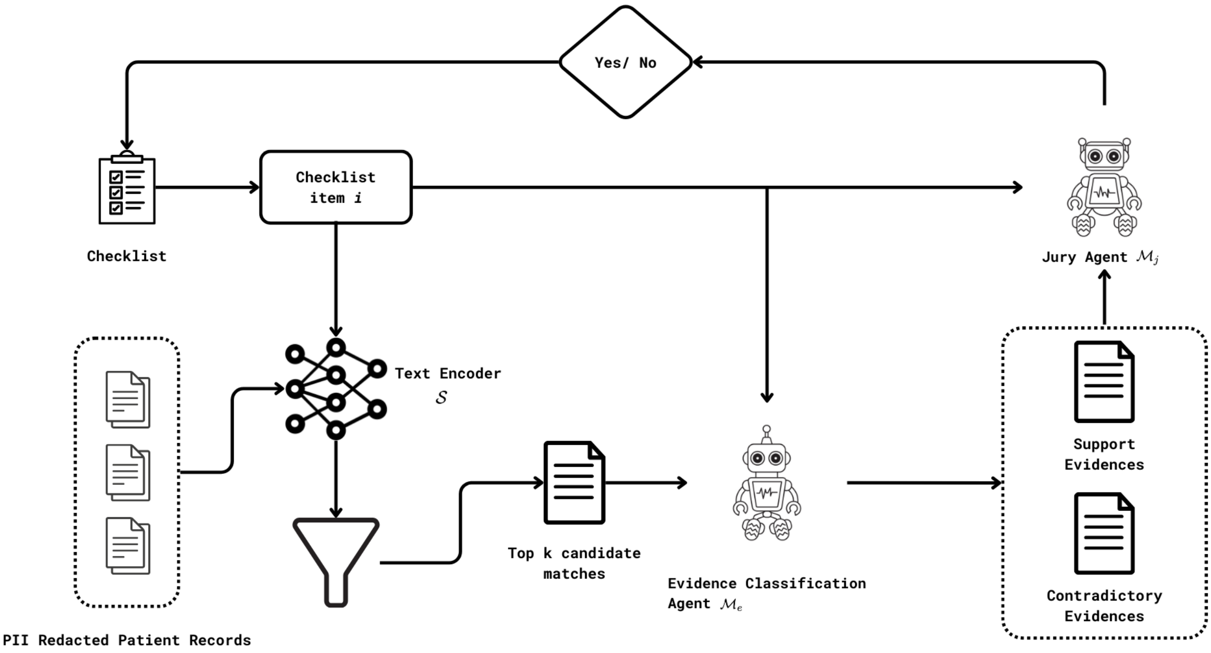

Yes/ No
Checklist
Checklist item i
Jury Agent $\mathcal{M}_j$
Text Encoder $\mathcal{S}$
Support Evidences
Contradictory Evidences
Top k candidate matches
Evidence Classification Agent $\mathcal{M}_e$
PII Redacted Patient Records

Figure 1: Leaf-Level Judgement Prediction where the first agent classifies the documents into supporting and contradictory sets and then the jury agent determines if the checklist item is satisfied.

cessity using AI. Another study aims to determine PA Approval for Lumbar Stenosis Surgery with Machine Learning (De Barros et al., 2023) but it uses surgery specific symptoms as input variables which would be difficult to generalize.

3 Problem Statement

As mentioned above, the evaluation of medical necessity is conducted through a meticulous comparison between patient medical records and established clinical guidelines. These medical records are systematically structured in a json-like format, usually in FHIR [1], within Electronic Health Records (EHRs) systems. Each object (resource) can be of type Patient (Patient Demographics), Observation (Laboratory Results), Procedure (Treatment History), Medication Request, Diagnostic Report etc. We define a set of EHR documents (resources) as $\mathcal{D} = \{d\}_{i=1}^{N_D}$ of size N_D

Further, clinical guidelines are formatted in a hierarchical, tree-like structure (referred as *checklist* in this paper), where each guideline statement (*parent node*) can encompass an arbitrary number of subordinate child statements (*sub-checklist or leaf node*) nested within it as shown in Figure 2 and 3. Thus, we define a coverage guideline or checklist as $\mathcal{C} = \{c\}_{j=1}^{N_C}$, where c is a checklist item.

Eventually, the task is to automatically determine the medical necessity $Y \in \{-1, 0, 1\}$ where -1 means the medical necessity is not justified, 1 means it is justified and 0 means there is a lack of sufficient evidence to justify the medical necessity criteria.

Recognizing the importance of transparency in the task, we also aim to provide evidence $\mathcal{E}_c = \{e_{c_k}\}_{k=1}^{N_c}$. These evidences can be used downstream to cross-reference medical documents used to establish medical necessity for the procedure.

We aim to construct a machine learning model $\mathcal{M}$ such that:

$$\mathcal{M}(\mathcal{D}, \mathcal{C}) = \{Y, \{\mathcal{E}_c\}\} \; \forall c \in \mathcal{C} \tag{1}$$

4 Methodology

Recently Large Language Models have shown great performance improvements by breaking down complex tasks into simpler sub-problems (Khot et al., 2022). Motivated by this observation, we propose a two step solution for our problem statement. First we determine the judgement of each of the leaf node checklist item. Subsequently, we propagate the solution for parent nodes bottom-up based on its child nodes' judgements.

4.1 Leaf-Node Judgement Prediction

Considering the immense volume of documents in Electronic Health Records (EHRs), we propose a Retrieval-Augmented Setup (Gao et al., 2024).

[1] https://www.hl7.org/fhir/

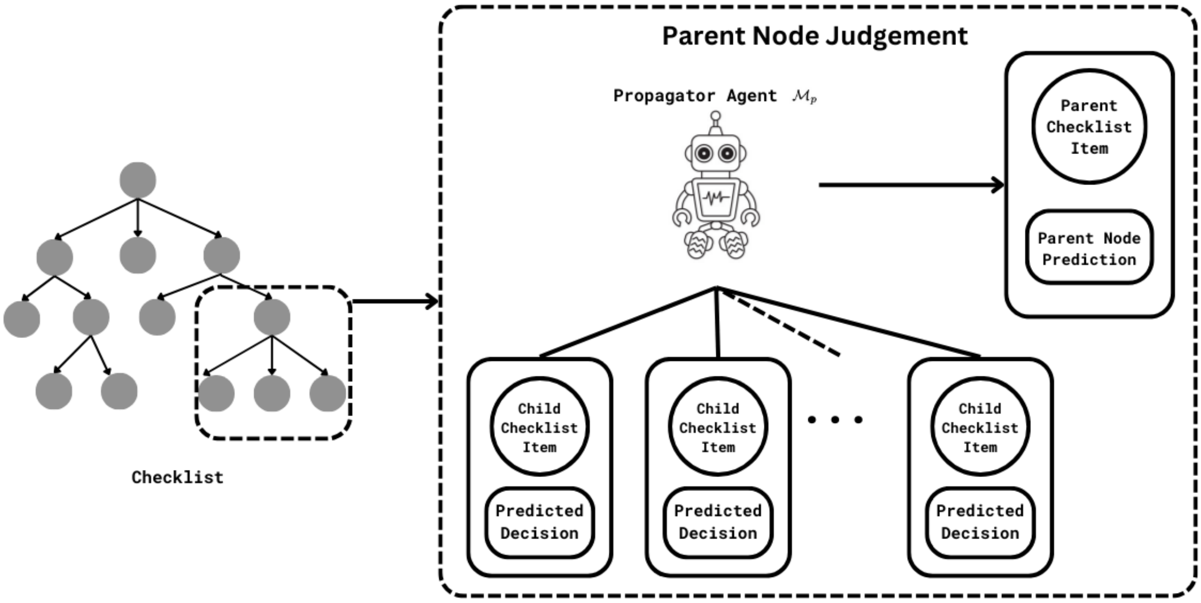


Figure 2: Bottom-Up Judgement Propagation where the agent uses the logical operators contained in a checklist item to determine how the aggregation should take place.

This approach first filters the document pool to identify a set of likely evidences (*top-k evidences*). A *Classification Agent* is then utilized to select the relevant evidences for the specific checklist item, enabling precise and efficient data extraction.

Top-k Evidence Selection: Given the EHR data $\mathcal{D}$, we first decompose it into its constituent resources (documents) where each document is an individual entry (individual lab-report data, procedure etc.) . In order to filter-down documents that are redundant towards the judgment, we first obtain top k candidate matches for the checklist item c from D. To achieve this, we propose to use a text encoder $\mathcal{S}$ to derive semantic representations for each checklist item c and for each document d in the EHR data. This method allows us to map both the checklist items and the documents into a shared semantic space, facilitating more effective matching based on relevance. Subsequently, we employ a semantic similarity metric to calculate the similarity score between each document d and the checklist item c. Based on the similarity metric, we obtain top-k closest matched documents with the checklist item c. Note that due to cost involved in using LLMs for this task, we keep[2] $k < 50$.

$$\mathcal{S}(\{d_i\}|_{i=1}^{N_D}, c) = \{d_{i'}\}|_{i'=1}^{k} \; \forall c \in \mathcal{C} \tag{2}$$

Evidence Retrieval and Prediction: Our proposed *Evidence Classification Agent* $\mathcal{M}_e$, first looks at each document d_i in top-k evidences retrieved along with the checklist item c and gives a verdict v_i, whether the document d_i is a supporting evidence, a contradictory evidence or it does not affect the judgment y_c. Note that this agent is executed k times since there are k retrieved documents.

$$\mathcal{M}_e(\{d_i, c\}|_{i=1}^{k}) = \{v_i\}|_{i=1}^{k} \; \forall c \in \mathcal{C} \tag{3}$$

Then our *Jury Agent* $\mathcal{M}_j$ picks up the complete set of evidences $d_i|_{i=1}^{k}$ along with their verdicts $v_i|_{i=1}^{k}$ and predicts the leaf-level checklist item judgment y_c along with evidences $\mathcal{E}_c \subset s_i|_{i=1}^{k}$ that acted in favour of the judgement y_c. We run this leaf-node pipeline multiple times ($n = 10$) and take vote of all predictions to determine the final judgement y_c. Confidence score f_c is calculated as the percentage of times the majority answer is predicted by the agent.

$$\mathcal{M}_j(\{d_i, v_i\}|_{i=1}^{k},\ c) = \{y_c, f_c, \{\mathcal{E}_c\}\} \; \forall c \in \mathcal{C} \tag{4}$$

4.2 Parent-Node Judgement Prediction

The value of a parent node is contingent upon the values of its nested child nodes. Hence, we determine parent node's value by aggregating children nodes' values which are connected through logical operators (AND, OR, NOT).

[2] In our experiments section, we show how the performance of our approach varies with k

Bottom-Up Judgement Propagation: In order to obtain the decision over the complete checklist $\mathcal{C}$, we propose to use an iterative bottom-up approach. In another words, we start from the leaf nodes and keep obtaining the judgment of their parent nodes. The iterations are terminated when we obtain the judgment and scores of the root node in the checklist $\mathcal{C}$.

Mathematically, at every iteration i, we choose a set of leaf node checklist items $c_j|_{j=1}^{N_{par}}$ having a common parent checklist item c_{par}, having judgements $y_j|_{j=1}^{N_{par}}$ and confidence scores $f_j|_{j=1}^{N_{par}}$. We then calculate the judgement y_{par} and confidence score f_{par} of parent node as:

$$\mathcal{M}_p(c_{par}, \{c_j, y_j, f_j\}|_{j=1}^{N_{par}}) = \{y_{par}, f_{par}\} \quad (5)$$

where $\mathcal{M}_p$ is our *Propagator Agent*.

5 Data Collection and Annotation

Getting live EHR data for the purpose of this evaluation is difficult, costly and full of regulatory requirements. We therefore used de-identified discharge summaries from MIMIC-IV-Note (Johnson et al., 2023a) as a proxy for this data. All discharge summaries therein have sections like chief complaint, history of present illness, past medical history, social history, physical and lab examinations, medications etc. which serves as the ideal data for this experiment. An average discharge summary has approximately 300 sentences divided into different categories. Joining this data with MIMIC-IV (Johnson et al., 2023b), we can get the CPT/ICD-10 codes associated with each note. We also collected a set of publicly available clinical guidelines (from CMS etc.) pertaining to Cardiology and Oncology and cross referenced the CPT codes in these guidelines to our notes data, thus creating a dataset of (note, guideline) pairs.

An example checklist [3] is shown in Figure 3. The checklist shows the clinical guideline for Therapeutic Footwear which consists of two items associated by **AND** operator. Item 2 in itself is a sub-checklist and will be true if any of the sub-checklist item is True as all of them are connected by **OR** operator.

Example Checklist

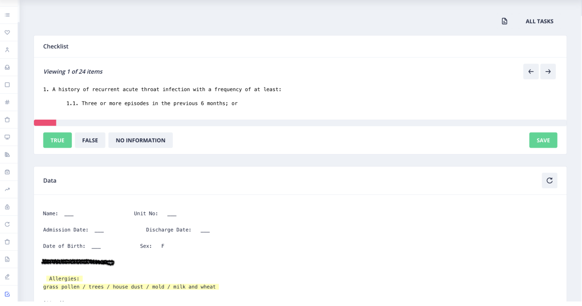
Eligibility Checklist for Therapeutic Footwear

1. The beneficiary has diabetes mellitus; and
2. The certifying physician has documented in the beneficiary's medical record one or more of the following conditions:
   (a) Previous amputation of the other foot, or part of either foot;
   (b) History of previous foot ulceration of either foot;
   (c) History of pre-ulcerative calluses of either foot;
   (d) Peripheral neuropathy with evidence of callus formation of either foot;
   (e) Foot deformity of either foot;
   (f) Poor circulation in either foot;

Figure 3: An example checklist formatted as a decision tree

5.1 Leaf Node Data Annotation

We hired 10 individuals with experience between 6-10 years in PA filing/reviewing both on payer and provider side. They were assigned the task of annotating leaf nodes of a checklist as either True, False or No Information. In case of True and False, the annotator has to also highlight statements in the data section as the evidences for that checklist item as shown in Figure 4. Additionally, each (note, guideline) pair was annotated by 3 different annotators and the final verdict was determined by taking the majority vote of all annotators for that checklist item. Following this, we created a dataset of 281 annotated checklists having 4577 leaf checklist items.



Viewing 1 of 24 items

1. A history of recurrent acute throat infection with a frequency of at least:
   1.1. Three or more episodes in the previous 6 months; or

Name: ___ Unit No: ___

Admission Date: ___ Discharge Date: ___

Date of Birth: ___ Sex: F

[redacted]

Allergies:
grass pollen / trees / house dust / mold / milk and wheat

Figure 4: Annotation Dashboard where each annotator has to mark if the checklist item is True, False or No Information (can't be concluded) and mark evidences for their selection.

5.2 Synthetic Data for Parent Judgement

To test parent-node judgment propagation, we created synthetic data. This was needed because the logical operations required were not within the expertise of our medical domain annotators. To create synthetic data, we first extracted out all sub-checklists from the unique set of guidelines we had,

[3] Taken from CMS: https://www.cms.gov/medicare-coverage-database/search.aspx

and then manually labelled each sub-checklist with the operator (AND, OR and NOT) used for the aggregation of result for that sub-checklist. Then we randomly assigned each leaf node in all sub-checklist their judgements and confidence score and calculated the judgement and confidence score of the parent node programmatically. With different permutations of True, False and No Information used for each sub-checklist, we created a dataset of 4500 sub-checklists used for the evaluation of parent node judgement propagation. This method of synthetic assignment is advantageous as it introduces a range of less likely or extraordinary judgment combinations, thereby challenging our Propagator Agent to maximize its robustness.

6 Experiments and Results

Our experiments were categorized into two distinct segments: assessment of leaf-node judgment and evaluation of parent-node judgment. To facilitate this, we established two separate test environments. Each test-bed was equipped to integrate various Large Language Models (LLMs) to ascertain the optimal model for our needs.

6.1 Leaf-node Judgement

Leaf-node judgment encompassed three sequential tasks. We start by splitting the entire document into sentences. Note that, with MIMIC data it is an easy way to chunk EHR data, but in real case scenario the chunking would happen at FHIR resource [4] level i.e. each Observation, Encounter, Lab Data etc. will act as the smallest chunk that goes into the pipeline. These chunks (or sentences here for simplicity) is first passed through the an encoder module which sorts the sentences according to the cosine similarity. The first 20 sentences are chosen for the experimentation. This simplifies the task of Classification Agent and also saves on LLM cost. The classification agent then segregates these filtered sentences in group of supporting and contradictory evidences which helps predicting the final judgement y_c by the Jury Agent.

Note that the evidences given by the model for each checklist item is not generated but classified. So each evidence will be an exact string match of a sentence from the input document. We have also ensured while annotation that the annotators also selects the evidence from the document as shown in Figure 4. This will help us measure the recall of encoder and classification agent against annotated data. The recall metric is defined as:

$$Recall = \frac{|t_{human} \cap t_{agent}|}{|t_{human}|} \tag{6}$$

where $|t_{human} \cap t_{agent}|$ represents the number of tokens that intersect between the human annotator and the agent, and $|t_{human}|$ is the total number of tokens identified as evidence by the human annotator. This measures if the Jury Agent had enough information to conclude the judgement.

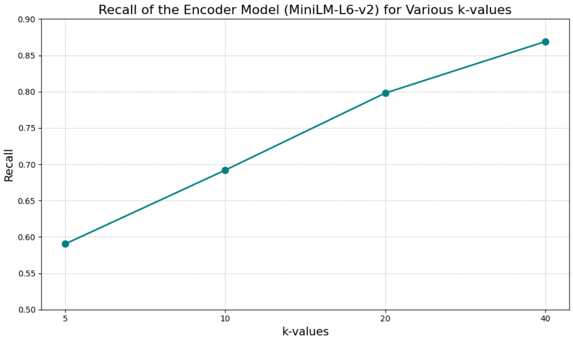


Figure 5: Recall of Encoder (MiniLM-L6-v2) model for various k-values

For encoder model, we used `MiniLM-L6` [5]. We took the top similarity sentences given by encoder model for various k-values and calculated recall against the human evidences and computed the average recall for all checklist items. The results are plotted in Figure 5. For $k = 40$, we get recall as 0.8689, which concludes that using encoder preserves useful information while discarding around 85% of irrelevant data (average MIMIC data has 300 sentences) towards the judgement.

Table 1: Recall metric for Classification Agent with different k values

Model	$k = 10$	$k = 20$
GPT-4	0.5792	0.6741
GPT-3.5	0.4844	0.5554
Claude-Opus	0.5254	0.5845
Calude-Sonnet	0.5042	0.5430

On similar lines, we calculated the recall of the Classification Agent by comparing segregated evidences: if humans marked a checklist item as true, we compared the supporting evidences from the

[4] Refer: https://www.hl7.org/fhir/

[5] Refer: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

agent to those identified by humans, and similarly, if marked as false, we compared the contradictory evidences. Table 1 shows the recall of Classification Agent for various LLMs. Clearly for $k = 20$ we have significantly higher recall as more evidences were present for the classifier to act upon. GPT-4 outperfomed other models with a maximum recall of 0.67 while other models showed slightly lower values.

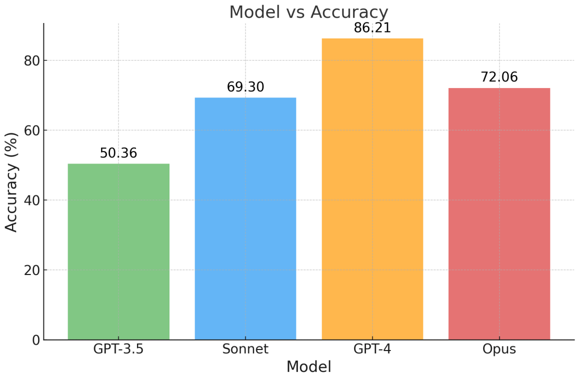


Figure 6: Accuracy of various LLMs for Jury Agent

We conducted a comprehensive evaluation of the Jury Agent ($\mathcal{M}_j$) employing various Large Language Models (LLMs), with a primary focus on accuracy and how it is affected with the number of retrieved evidences (k). GPT-4 and Opus demonstrated robust performance, achieving accuracies of 86% and 72% (Figure 6), respectively. Notably, while Sonnet exhibited a slightly lower accuracy of 69% compared to Opus, it provided a considerable advantage in terms of latency, reducing it by approximately 32% (Figure 7).

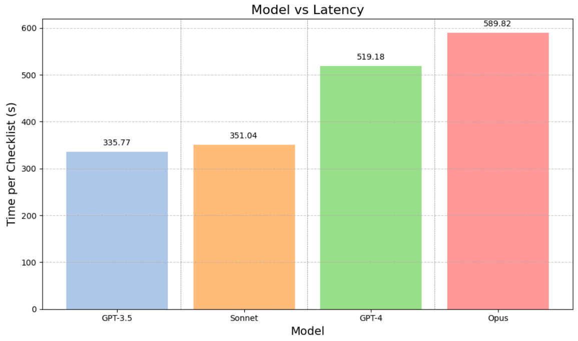


Figure 7: Latency of various LLM model for the leaf-node pipeline

Effect of Number of Retrieved Evidences (*k*): To better understand the effect of *k* on our pipeline and choose the best value we tweaked the value of *k* and ran the pipeline on a smaller sample of our dataset consisting of 20 checklists (having 680 checklist items). We observed that as we increase the value of *k*, the model performance increases till a value of $k = 20$, after which the accuracy gets saturated as shown in Figure 8.

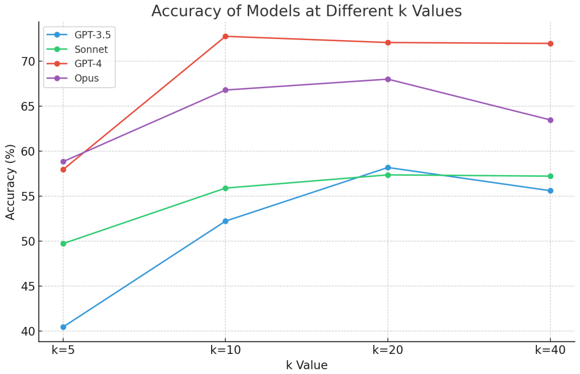


Figure 8: Effect of various k-values on Jury Agent

6.2 Parent-node Judgement

Our *Propagator Agent* is an LLM-powered Agent, which takes up a parent node and its corresponding leaf nodes (along with their judgments and confidence scores) to obtain the judgment and confidence score of the parent node. This was done in two ways. In the first experiment, the LLM agent is asked directly to determine the response and score given parent statement and its child statements, responses and scores. The agent has to understand the logical operators (AND, OR, NOT) and then combine the child responses (True, False, No Information) to conclude parent judgement. The logical rules for No Information items is given in Figure 9 and rules for calculating confidence score is given in Figure 10. In the second experiment, the LLM agent was asked to compute the logical operator between each child item and then the calculation of response and confidence score was done programmatically.

We evaluate the performance of the *Propagator Agent* across various dimensions. The outcomes of this analysis are presented in Table 2. The score accuracy refers to the accuracy of both the response and confidence score propagated correctly while the response accuracy is accuracy of only response being propagated correctly to the parent node resulting from the first experiment. The operator accuracy refers to the accuracy of the model to correctly identify the operators as done in the second experiment.

From the table we can conclude that the Agent

Table 2: Model performance for Propagator Agent using Chain of Thought (CoT) & In-Context Learning (ICL)

	GPT-4		GPT-3.5		Claude-Sonnet		Claude-Opus	
CoT + ICL	**ICL**	**CoT + ICL**	**ICL**	**CoT + ICL**	**ICL**	**CoT + ICL**	**ICL**	**CoT + ICL**
Response Accuracy (%)	87.17	95.60	78.75	91.20	82.05	85.71	85.34	95.24
Score Accuracy (%)	78.35	93.04	48.35	85.34	53.66	81.31	79.12	94.50
Operator Accuracy (%)	89.27	95.01	81.04	92.82	82.05	87.54	84.78	94.04

is able to propagate response more accurately than confidence scores, as propagating confidence score is a more complex task than determining the response which involves only logical operations. Second experiment shows that the accuracy of operator determination task is comparable to the response accuracy determined using first approach. Once the operators are determined, response and confidence score are calculated programatically. Since determining operator would be a one time task (to be done while creating guidelines) taking second approach would get us similar accuracy but at significantly lower cost.

Rule Set for No Information Items

Case I: AND Operator

1. True **AND** No Information = No Information
2. False **AND** No Information = False

Case II: OR Operator

1. True **OR** No Information = True
2. False **OR** No Information = No Information

Case III: NOT Operator

1. **NOT** No Information = No Information

Figure 9: Rule Set for No Information Items followed by Propagator Agent for parent node judgement

Effect of Prompting Strategy: We performed two sets of experiments. The first involved providing *In-Context Learning* (ICL) examples (Min et al., 2022) and measuring accuracy. Larger models such as GPT-4 and Opus yielded strong results, whereas smaller models like Sonnet and GPT-3.5 exhibited suboptimal performance when relying solely on ICL prompts. However, in the second experiment, when supplemented with *Chain of Thought* (CoT) prompting (Wei et al., 2022b), the performance of these smaller models markedly improved, demonstrating how the step-by-step reasoning process aids in decomposing the complex task of propagation into manageable segments. However, the use of Chain of Thought (CoT) prompting substantially increases response times for larger models due to its generation of an increased number of tokens compared to ICL-only prompting. In contrast, the enhancements in performance observed with GPT-3.5 are achieved without a marked increase in latency, particularly when compared to larger models such as Opus and GPT-4 under similar conditions.

Confidence Score (f) Calculation

Case I: AND Operator

1. If final response is True:
   f_{par} = min(f of all True child responses)
2. If final response is False:
   f_{par} = max(f of all False child responses)
3. If final response is No Information:
   f_{par} = min(f of all No Information child responses)

Case II: OR Operator

1. If final response is True:
   f_{par} = max(f of all True child responses)
2. If final response is False:
   f_{par} = min(f of all False child responses)
3. If final response is No Information:
   f_{par} = min(f of all No Information child responses)

Figure 10: Confidence Score calculation rules followed by Propagator Agent for parent node judgement

Effect of LLM Choice: We conducted an evaluation of the *Propagator Agent* utilizing various LLMs, with a particular emphasis on metrics such as accuracy and latency. Opus and GPT-4 emerged as the top performers, achieving approximately 94-95% accuracy when CoT prompting was combined with ICL examples.

GPT-3.5 is ranked second in terms of accuracy but presents significant benefits in reduced latency compared to GPT-4 and Opus, as depicted in Figure 11. Additionally, the operational costs associated with GPT-3.5 are substantially lower. Although selecting the optimal model involves a trade-off, GPT-3.5 stands out as the preferred option when considering a balance among cost, latency, and accuracy. Nonetheless, for scenarios where maximum accuracy is crucial, the larger models such as GPT-4 and Opus are more appropriate.

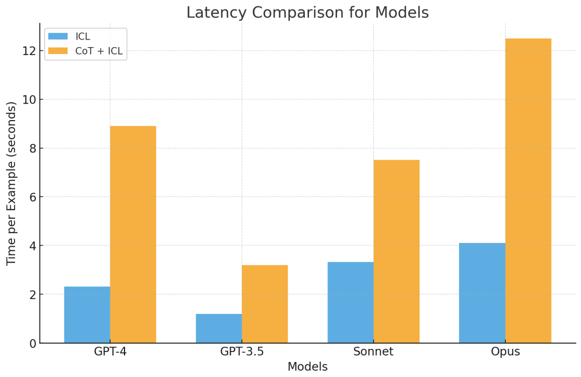


Figure 11: Latency Analysis of LLMs Under ICL and CoT for Propagator Agent when computing score accuracy

7 Conclusion

Our experiments utilized MIMIC-Note data, a set of string-based data. However, real-world applications typically involve obtaining resources (FHIR data) from EHR systems. Converting these resources into stringified data poses a unique engineering challenge. Although manageable, it is crucial to determine whether this data format could impact the effectiveness of our system.

In our approach, we integrated the use of confidence scores. Agents at the leaf nodes compute a confidence score for their predictions, which is then propagated up to the root node alongside the response. The confidence score at the root node is vital as it reflects the system's certainty about the prediction quality. Checklists with low confidence scores are directed to a service layer where experienced professionals can review or adjust the model responses. This feedback loop can be leveraged to refine and enhance future models.

Given our focus on the healthcare sector, ensuring the explainability of outputs from these LLM agents was paramount. The decision-making process was elucidated through Chain of Thought (CoT) prompting and evidence collected by the Classification Agent, enhanced the transparency needed when AI models are employed in healthcare workflows.

While initially designed to automate prior authorization (PA) filing, this solution could also improve clinical decision support (CDS) systems by providing real-time alerts to physicians during consultations. For instance, it could alert physicians to incomplete medical records when prescribing treatments requiring PA, ensuring necessary documentation is promptly addressed. Thus, system responsiveness or latency becomes a critical metric for assessing its performance.

We have shown that breaking down a large, complex problem into smaller, specialized tasks handled by distinct agents can significantly enhance our ability to automate sophisticated tasks that were previously very challenging. This strategy also facilitates the shift from a monolithic AI solution ($\mathcal{M}$) to a micro-service architecture-driven solution ($\mathcal{M}_e$, $\mathcal{M}_j$, and $\mathcal{M}_p$). Currently, our method involves a constrained workflow, but it holds potential for evolving into a system with loosely coupled agents that are more dynamic and capable of improved problem-solving.

The ideal implementation of this methodology would adopt a structure akin to an organization, where the architecture consists of several pods. Each pod contains worker agents specialized in different aspects of the problem, complemented by checker agents that reassess and validate the outputs, triggering reruns when necessary. A super-orchestrator agent would oversee and coordinate the activities across the architecture. This setup aims to mitigate common issues like hallucination often seen in existing LLMs.

8 Acknowledgments

The authors wish to express their gratitude to Gaurav Grontayal and Tanmay Johri for their invaluable assistance in gathering the necessary checklists, hiring consultants and annotators, resolving annotation conflicts, and managing the overall data annotation pipeline. We also extend our thanks to RISA Labs for their support of our research. Special thanks are due to all annotators involved for their diligent work in data annotation.

References

Hiba Ahsan, Denis Jered McInerney, Jisoo Kim, Christopher Potter, Geoffrey Young, Silvio Amir, and Byron C Wallace. 2023. Retrieving evidence from ehrs with llms: Possibilities and challenges. *arXiv preprint arXiv:2309.04550*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

James Chambers, Matthew Chenoweth, and Peter Neumann. 2016. Mapping us commercial payers' coverage policies for medical interventions. *The American journal of managed care*, 22:e323–e328.

Hejie Cui, Xinyu Fang, Ran Xu, Xuan Kan, Joyce C Ho, and Carl Yang. 2024. Multimodal fusion of ehr in structures and semantics: Integrating clinical records and notes with hypergraph and llm. *arXiv preprint arXiv:2403.08818*.

A. De Barros, F. Abel, S. Kolisnyk, et al. 2023. Determining prior authorization approval for lumbar stenosis surgery with machine learning. *Global Spine Journal*, 0(0).

A. Diane, P. Gencarelli, J. M. Lee, et al. 2023. Utilizing chatgpt to streamline the generation of prior authorization letters and enhance clerical workflow in orthopedic surgery practice: A case report. *Cureus*, 15(11):e49680.

Simon Ellershaw, Christopher Tomlinson, Oliver E Burton, Thomas Frost, John Gerrard Hanrahan, Danyal Zaman Khan, Hugo Layard Horsfall, Mollie Little, Evaleen Malgapo, Joachim Starup-Hansen, et al. 2024. Automated generation of hospital discharge summaries using clinical guidelines and large language models. In *AAAI 2024 Spring Symposium on Clinical Foundation Models*.

Yao Fu, Hao Peng, Tushar Khot, and Mirella Lapata. 2023. Improving language model negotiation with self-play and in-context learning from ai feedback. *Preprint*, arXiv:2305.10142.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. *Preprint*, arXiv:2402.01680.

Junda He, Christoph Treude, and David Lo. 2024. Llm-based multi-agent systems for software engineering: Vision and the road ahead. *arXiv preprint arXiv:2404.04834*.

Alistair Johnson et al. 2023a. MIMIC-IV-Note: Deidentified Free-Text Clinical Notes (version 2.2). PhysioNet. Accessed: 2023-07-10.

Alistair Johnson et al. 2023b. MIMIC-IV (version 2.2). PhysioNet. Accessed: 2023-07-10.

L.K. Jones, I.G. Ladd, C. Gregor, et al. 2021. Evaluating implementation outcomes (acceptability, adoption, and feasibility) of two initiatives to improve the medication prior authorization process. *BMC Health Services Research*, 21:1259.

HyoJe Jung, Yunha Kim, Heejung Choi, Hyeram Seo, Minkyoung Kim, JiYe Han, Gaeun Kee, Seohyun Park, Soyoung Ko, Byeolhee Kim, et al. 2024. Enhancing clinical efficiency through llm: Discharge note generation for cardiac patients. *arXiv preprint arXiv:2404.05144*.

Tushar Khot, Kyle Richardson, Daniel Khashabi, and Ashish Sabharwal. 2022. Hey ai, can you solve complex tasks by talking to agents? *Preprint*, arXiv:2110.08542.

Gyubok Lee, Hyeonji Hwang, Seongsu Bae, Yeonsu Kwon, Woncheol Shin, Seongjun Yang, Minjoon Seo, Jong-Yeup Kim, and Edward Choi. 2022. Ehrsql: A practical text-to-sql benchmark for electronic health records. In *Advances in Neural Information Processing Systems*, volume 35, pages 15589–15601. Curran Associates, Inc.

Leslie A. Lenert, Steven Lane, and Ramsey Wehbe. 2023. Could an artificial intelligence approach to prior authorization be more human? *Journal of the American Medical Informatics Association*, 30:989–994.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, et al. 2024. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. *Advances in Neural Information Processing Systems*, 36.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *Preprint*, arXiv:2303.17651.

V Madhusoodanan, L Ramos, IJ Zucker, A Sathe, and R Ramasamy. 2023. Is time spent on prior authorizations associated with approval? *J Nurse Pract*, 19(2):104479. Epub 2022 Nov 10.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *Preprint*, arXiv:2202.12837.

Abdulqadir J Nashwan and Salam Bani Hani. 2023. Enhancing oncology nursing care planning for patients with cancer through harnessing large language models. *Asia-Pacific Journal of Oncology Nursing*, 10(9).

Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. 2023. Capabilities of gpt-4 on medical challenge problems. *Preprint*, arXiv:2303.13375.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "gradient descent" and beam search. *Preprint*, arXiv:2305.03495.

Stephen G Salzbrenner, Carrie McAdam-Marx, Maxwell Lydiatt, Brandon Helding, Lawrence M Scheier, and Patricia Wonch Hill. 2022. Perceptions of prior authorization by use of electronic prior authorization software: A survey of providers in the united states. *Journal of Managed Care & Specialty Pharmacy*, 28(10):1121–1128. PMID: 36125058.

Wenqi Shi, Ran Xu, Yuchen Zhuang, Yue Yu, Jieyu Zhang, Hang Wu, Yuanda Zhu, Joyce C Ho, Carl Yang, and May Dongmei Wang. 2024. Ehragent: Code empowers large language models for few-shot complex tabular reasoning on electronic health records. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Nathaneal Scharli, Aakanksha Chowdhery, Philip Mansfield, Blaise Aguera y Arcas, Dale Webster, Greg S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. 2022. Large language models encode clinical knowledge. *Preprint*, arXiv:2212.13138.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Tao Tu, Shekoofeh Azizi, Danny Driess, Mike Schaekermann, Mohamed Amin, Pi-Chuan Chang, Andrew Carroll, Charles Lau, Ryutaro Tanno, Ira Ktena, Anil Palepu, Basil Mustafa, Aakanksha Chowdhery, Yun Liu, Simon Kornblith, David Fleet, Philip Mansfield, Sushant Prakash, Renee Wong, Sunny Virmani, Christopher Semturs, S. Sara Mahdavi, Bradley Green, Ewa Dominowska, Blaise Aguera y Arcas, Joelle Barral, Dale Webster, Greg S. Corrado, Yossi Matias, Karan Singhal, Pete Florence, Alan Karthikesalingam, and Vivek Natarajan. 2024. Towards generalist biomedical ai. *NEJM AI*, 1(3):AIoa2300138.

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. *Preprint*, arXiv:2203.11171.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Preprint*, arXiv:2206.07682.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Christopher YK Williams, Jaskaran Bains, Tianyu Tang, Kishan Patel, Alexa N Lucas, Fiona Chen, Brenda Y Miao, Atul J Butte, and Aaron E Kornblith. 2024. Evaluating large language models for drafting emergency department discharge summaries. *medRxiv*, pages 2024–04.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. 2024. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–32.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *Preprint*, arXiv:2210.03493.

Hongjian Zhou, Fenglin Liu, Boyang Gu, Xinyu Zou, Jinfa Huang, Jinge Wu, Yiru Li, Sam S. Chen, Peilin Zhou, Junling Liu, Yining Hua, Chengfeng Mao, Chenyu You, Xian Wu, Yefeng Zheng, Lei Clifton, Zheng Li, Jiebo Luo, and David A. Clifton. 2024. A survey of large language models in medicine: Progress, application, and challenge. *Preprint*, arXiv:2311.05112.