Report on Large Language Models Application for Reasoning in the Medical Domain

Boammani Aser Lompo[1,3], Jouvet Philippe[2,3], Rita Noumeir[1,3]

[1] École de Technologie Supérieure,
[2] Université de Montréal,
[3] Centre Hospitalier Universitaire Sainte-Justine

Abstract. Large Language Models (LLMs) have demonstrated substantial promise in the medical field, where reasoning over vast and varied datasets is critical. However, effectively deploying these models in healthcare requires addressing significant challenges, including the need for domain-specific data, fine-tuning for clinical accuracy, and ensuring models are compact enough for broad accessibility. This article surveys key advancements in reasoning methods, such as Chain-of-Thought, Chain-of-Hindsight, and Self-Consistency, which enable LLMs to perform complex reasoning and improve interpretability through step-by-step problem-solving. Notably, self-improving techniques also facilitate the generation of quality training data, helping to mitigate the scarcity of labeled medical datasets and highlighting the value of large model scales to enhance reasoning capacity. Efforts to condense these models into Small Language Models (SLMs), though promising, show that substantial reasoning abilities remain predominantly within the domain of larger LLMs, underscoring their necessity for high-level medical applications. By examining recent studies, we explore methods to organize and encode medical knowledge—such as the creation of the MultiMedQA dataset—and advances in multilingual reasoning that address data scarcity across languages. This work ultimately emphasizes that while LLMs hold potential for achieving clinician-level reasoning in healthcare, model optimization and interpretability remain essential for ensuring safe, equitable, and accessible AI deployment in clinical settings.

Keywords: Medical Reasoning, LLM, SLM, Reinforcement Learning Human Feedback

1 Introduction

In recent years, Large Language Models (LLMs) have emerged as powerful tools in the field of medical reasoning, offering promising applications for enhancing diagnostic and decision-making processes. One of their key strengths lies in reasoning over implicit knowledge Conneau *et al.*, 2018; Talmor *et al.*, 2020. This capability enables LLMs to utilize vast amounts of information encoded within their parameters to generate insights even when direct data is unavailable, thereby mimicking human-like reasoning processes in complex medical scenarios.

Medical reasoning with LLMs can be approached through different reasoning frameworks. In this work we will explore some of them such as abductive, inductive, and defeasible reasoning. Abductive reasoning Young *et al.*, 2022, often utilized in diagnostic settings, allows LLMs to generate hypotheses based on incomplete or ambiguous data. Inductive reasoning Yang *et al.*, 2024 enables models to generalize from specific instances, supporting prognosis and treatment recommendations. In contrast, defeasible reasoning involves reassessing the certainty of a hypothesis based on new information, making it useful for adjusting diagnoses as new medical results or events emerge.

To improve their performance, LLMs also leverage self-improvement techniques. Chain of Thought (CoT)Wei *et al.*, 2022 methods guide LLMs to systematically outline their reasoning steps, enhancing transparency and interpretability. Chain of Hindsight (CoH) approaches further refine reasoning by allowing models to revisit and refine their previous steps, simulating a reflective process Liu, Sferrazza, and Abbeel, 2023. Self-Consistency Wang *et al.*, 2022 adds robustness to the models' predictions by aggregating multiple reasoning paths, leading to more accurate and reliable outputs. Few-shot learning Kojima *et al.*, 2022 techniques are particularly valuable in medical contexts, enabling models to learn new tasks or adapt to novel scenarios with minimal labeled data, which is often scarce in specialized fields.

This document will further explore the potential of small language models in medical reasoning. While large models have traditionally dominated the field, smaller, specialized models are gaining attention for their efficiency and adaptability. By focusing on task-specific knowledge and requiring fewer computational resources, small language models present a compelling option for scalable, targeted applications in healthcare, especially in settings with limited access to high-powered computational infrastructure. However, a large number of parameters appears essential for enhancing reasoning capabilities Wei *et al.*, 2022; Zhang *et al.*, 2024.

2 Implicit Knowledge Extraction from LLMs

LLMs are known to encode and recall large amounts of information from their training data, despite not being explicitly trained to memorize. This capability raises questions about the relationship between language modeling and knowledge retention. As a result, numerous studies have focused on understanding how LLMs store and process implicit knowledge, yielding applications in areas such as cross-lingual sentence representations Conneau *et al.*, 2018 and systematic reasoning over implicit knowledge Talmor *et al.*, 2020.

2.1 Cross-Lingual Sentence Representation

Cross-lingual sentence representation aims to transfer understanding across languages, enabling models to process multilingual data efficiently, especially in low-resource languages where annotated data is scarce. In Conneau *et al.*, 2018, two primary approaches are explored: embeddings alignment and multilingual embeddings.

Embeddings Alignment Approach: This method involves aligning the embedding spaces of two language-specific models, $\mathcal{M}_1$ and $\mathcal{M}_2$, by learning a mapping matrix W using a sample $\mathcal{D}$ of shared words. The goal is to minimize the distance between corresponding embeddings across the two languages, formulated as:

$$W^* = \text{argmin}_{W \in O_d(\mathcal{R})} \sum_{x \in \mathcal{D}} \left\| W\mathcal{M}_1 x - \mathcal{M}_2 x \right\|$$

This matrix enables translation of word embeddings from one language into the other, yielding promising results for low-resource languages.

Multilingual embeddings approaches: Here, $\mathcal{M}_2$ (for a secondary language) is fine-tuned to produce embeddings similar to those generated by $\mathcal{M}_1$ (for English). This results in a unified embedding space, allowing a single classifier to handle downstream tasks in multiple languages. The authors trained a **Bi-LSTM** network classifier and observed state-of-the-art performance at the time.

XNLI-dataset: To extend Natural Language Inference (NLI) to Cross-Lingual Language Inference (XNLI), the authors created a dataset covering 15 languages. They collected 7,500 English premise-hypothesis pairs from diverse genres and translated them, creating a dataset of 112,500 pairs, labeled as "entails," "contradicts," or "neither."

2.2 Reasoning over Implicit Knowledge

In Talmor *et al.*, 2020, the authors show that LLMs can utilize memorized knowledge to answer questions posed in natural language. By training **RoBERTa** Devlin *et al.*, 2018 on QA datasets like **20Q** and **RULETAKER**, which include commonsense knowledge, they demonstrated the model's ability to answer implicit knowledge-based questions, as illustrated in this example:

Input: *Chen would like to buy an animal smaller than a horse, but Chen does not want a fish. Chen would like to buy a Dog?* **Implicit memorized knowledge**: *The model learned that dogs were smaller than horses* **Output**: *True*

After evaluation, the accuracy reached 99% proving their claim. the authors developed a method for automated dataset generation for training models in implicit reasoning. This process involves:

- Sampling a relevant hypernym rule (e.g., "a whale is a mammal") from large datasets like CONCEPTNET Speer, Chin, and Havasi, 2017, or WORDNET Miller, 1995 etc.
- Finding a relevant property of the hypernym object, e.g., (mammal has a belly button, true)
- Applying the hypernym inference type: (if A is a B and B has property C, then A has property C) to deduce the logical conclusion. This conclusion will become the hypothesis, e.g. (whale has a belly button), true)
- Finally adding distractors—irrelevant information to challenge reasoning further.

This structured approach enables the efficient expansion of implicit reasoning training datasets, advancing LLM capabilities in nuanced knowledge-based reasoning.

3 Different forms of Reasoning for LLMs

Reasoning can take various forms, and based on the type of reasoning we consider, there is an adapted training setup and specific dataset. This section explores three forms of reasoning: abduction Young *et al.*, 2022, defeasible reasoning Rudinger *et al.*, 2020, and inductive reasoning Yang *et al.*, 2024.

3.1 Abduction Reasoning

When an observed fact p cannot be deduced from a knowledge base, abductive reasoning identifies additional facts that, if added, would allow p to follow from the existing knowledge. Table 1 illustrates the difference between abduction, deduction, and induction.

TABLE 1. Illustration of Abduction reasoning

Deduction:	Socrates is human →	Humans are mortal →	?
Induction:	Socrates is human →	?	→ Socrates is mortal
Abduction:	?	→ Humans are mortal →	Socrates is mortal

As stated in Young *et al.*, 2022, abduction helps us understand observations difficult to interpret. Training Transformers to abduction can also lift one limitation of Transformers that memorize knowledge without capturing any underlying reasoning. To train their model for abduction, the authors created the dataset **Abduction Rule** according to 4 principles:

- Each datapoint is written in natural language, instead of formal math language
- Each datapoint is made of a few facts, a few rules, an observation and an explanation which derives from facts, rules and observations
- Each rule is limited to three conditions (e.g., "If something is cute, funny, and adorable, then…")
- Facts and rules are presented randomly to prevent reliance on consistent ordering

Abduction Rule is made of three animal-related datasets and three human-centered datasets, each at varying complexity levels. 6 different models were then trained on each of the datasets and evaluated on the other datasets. The authors observed the following conclusions:

- No model gave a single correct answer in a domain different from the one of his training. It shows that reasoning is not only a matter of syntax manipulation, the model also needs to be familiar with the topic
- In general, models performed better when trained on complex datasets and tested on simpler datasets
- Training models on multiple domains does improve performance.

3.2 Defeasible Reasoning

Defeasible reasoning, as presented in Rudinger *et al.*, 2020, allows initial inferences to be revised based on new information. For instance, given the context "The drinking glass fell," one might infer that "The drinking glass broke." But if new information specifies, "The glass fell onto a pile of laundry," the initial inference is weakened.

The paper introduces a dataset for defeasible reasoning in three categories: natural language inference (δ-SNLI), common sense reasoning (δ-Atomic), and reasoning about social norms (δ-Social). Each category contains premise-hypothesis pairs, with added contextual updates called "strengtheners" (making hypotheses more plausible) and "weakeners" (making hypotheses less likely). Examples are shown in Table 2 extracted from Rudinger *et al.*, 2020.

TABLE 2. Some instances of the Defeasible reasoning dataset

Task	Premise	Hypothesis	Type	Update
δ-SNLI	Old man crafting something in his workshop	Old man crafting something in his workshop	strengthener	The man is serious and is surrounded by workers
			weakener	The man is wearing pajamas and is chuckling
δ-Atomic	PersonX has a pool party	Because PersonX wanted to hangout with friends	strengthener	It was PersonX's birthday
			weakener	PersonX was having a family reunion
δ-Social		You should help your family with funeral expenses	strengthener	They have asked you to chip in
			weakener	You are not financially stable

These datasets can be used to train an LLM in two different ways:

- **Classification**: Predicting whether an update strengthens or weakens an hypothesis
- **Generation**: Producing an update that either strengthens or weakens a given hypothesis

The goal of these tasks is to train the model to think as a sceptic, which means considering the possible weaknesses of a given claim or argument in order to come up with examples or counterarguments that may undermine it. For evaluation, the authors trained 3 different models (**T5, Bart, GPT2**) and compared their performance. he classification performance matched human-level accuracy (best with **GPT2**), though performance in generation was lower, as models often rephrased the hypothesis rather than introducing new information.

3.3 Inductive Reasoning

The goal of Inductive reasoning is to derives general rules or hypotheses from observed evidence. As for the two previous papers, Yang *et al.*, 2024 provided a method to generate a dataset specific to inductive reasoning named **DEERLET**. In this dataset, each entry is a tuple of facts, rule and four labels:

- Label 1 says whether a rule is not in conflict with its facts
- Label 2 says whether a rule reflects reality
- Label 3 says whether a rule is more general than its fact
- Label 4 says whether a rule is trivial

Labels are learned separately by some different models, in order to make data generation automatic. The topics covered in this dataset include physics, history, zoology, botany, astronomy and geology. Using this dataset, we can train an LLM with the following pipeline: the model is provided with factual information as input. Based on these facts, the model generates a rule that should logically explain them. The generated rule is then assessed using the four labels. If the rule scores poorly on Label 1, the model generates a new rule and re-evaluates it. Once the rule scores well on Label 1, it's then checked against Label 2, and this cycle continues until the rule meets all four labels. Figure 1 provides a thorough illustration

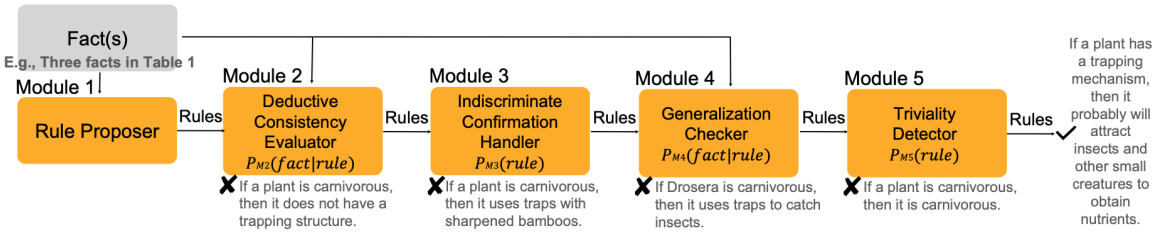

Fig. 1. Generation process use for inductive reasoning Models

Without a prior benchmark, the authors achieved promising results with **GPT-J**, reporting performance three times better than models trained solely on facts and rules.

4 Enhancing the Intrinsic Abilities of Models

In recent developments, researchers have explored not only optimizing LLM embedding spaces and guiding models to reason in structured ways but also improving performance through strategic prompting techniques. Using particular keywords and refined training strategies, these approaches can encourage models to produce more precise and contextually aware responses. In this section, we examine several of these prompting-based methods.

4.1 Chain Of Thoughts

The "Chain of Thought" (CoT) prompting, introduced by Wei *et al.*, 2022 demonstrates that sufficiently large LLMs can develop strong reasoning abilities through an approach called Chain of Thought prompting. In CoT prompting, the model is guided to produce a sequence of intermediate reasoning steps before arriving at its final answer. This approach has several benefits:

- **Problem Breakdown**: By breaking down a complex task into a series of smaller steps, the model can allocate greater processing power to challenging parts of the problem.

– **Improved Interpretability:** The explicit steps make it easier to understand and validate the reasoning path that leads to the final answer.

Table 3 (drawn from Kojima *et al.*, 2022) provides examples of prompts and responses using zero-shot and few-shot inputs, both with and without CoT.

TABLE 3. Example of zero shot and few shot inputs with and without Chain of Thoughts

Task	Prompt beginning	Prompt question
Few-shot	Q: Roger has 5 tennis balls. He buys two more cans of tennis ball. Each can has 3 tennis balls. How many ball does he have now? A: The answer is 11.	A juggler can juggle 16 balls. Half of the balls are golf balls and half of the golf balls are blue. How many blue golf balls are there? A:
Zero-shot		A juggler can juggle 16 balls. Half of the balls are golf balls and half of the golf balls are blue. How many blue golf balls are there? A: The answer is ...
Few-shot-CoT	Q: Roger has 5 tennis balls. He buys two more cans of tennis ball. Each can has 3 tennis balls. How many ball does he have now? A: Roger started with 5 balls. 2 cans of 3 balls of tennis each is 6 balls. 5+6=11. The answer is 11.	A juggler can juggle 16 balls. Half of the balls are golf balls and half of the golf balls are blue. How many blue golf balls are there? A:
Zero-shot-CoT		A juggler can juggle 16 balls. Half of the balls are golf balls and half of the golf balls are blue. How many blue golf balls are there? A: **Let's think step by step**

Wei *et al.*, 2022 evaluated CoT prompting on diverse maths problem benchmarks: **GSM8K**, **SVAMP**, **ASDiv**, **AQuA**, and **MAWPS**. For each of these problems, they compare the results of standard input, and CoT prompting input. The authors considered various models with a wide range of complexity: **GPT-3** (350M to 175B), **LaMBDA** (422M to 137B), **PaLM** (8B to 540B). Key findings included:

– **Parameter Dependency**: CoT did not impact model performance significantly for models below 100B parameters, but as model size increased, so did performance gains. The larger the model, the bigger the performance improvements
– **Improved Accuracy in Large Models**: Large models, such as **GPT-3** (175B) and **PaLM** (540B), achieved up to 75% accuracy across datasets with CoT prompting.
– **High Consistency**: In cases where the final answer was correct, the CoT reasoning was accurate over 97% of the time, with errors primarily due to issues like calculator inaccuracies or one-step omissions.

CoT prompting generally suits well few shot training, but Kojima *et al.*, 2022 showed that it could also be exploited in zero shot learning Kojima *et al.*, 2022. Zero-shot learning is a setup where a language model is asked to complete a task without seeing any examples or demonstrations of how to do it first. This contrasts with few-shot learning, where the model is shown a few examples of how the task works before answering a similar question. To implement the Chain of Thought (CoT) technique in zero-shot learning, the authors introduced a two-stage prompting process. In the first stage, called **reasoning extraction**, the model is given a prompt that ends with the phrase "Let's think step by step." This phrase prompts the model to generate a sequence of reasoning steps, essentially mapping out the logic it will follow to arrive at an answer.

In the second stage, these reasoning steps are combined with the original prompt to form an **augmented prompt**. This updated prompt is then submitted back to the model, now ending with the phrase "Therefore the answer is," which signals the model to use the gathered reasoning to generate the final answer. This two-stage approach—gathering reasoning first, then forming a conclusion—helps the model to work through complex questions even without examples to guide it. Table 4 provides a full illustration of this pipeline.

The authors conducted an evaluation similar to that in Wei *et al.*, 2022 to assess various language models of different sizes on math problem benchmarks and commonsense datasets like CommonSenseQA and StrategyQA. Their findings aligned with the earlier study's results:

– Under a zero-shot learning setup, model size alone did not improve performance significantly. However, larger models did show greater proficiency when using Chain of Thought (CoT) prompting.

TABLE 4. Example of two stages prompting for zero shot with Chain of Thoughts

Prompt	Input	Output
First prompt	Q: Roger has 5 tennis balls. He buys two more cans of tennis ball. Each can has 3 tennis balls. How many ball does he have now? A: **Let's think step by step**	A: Roger started with 5 balls. 2 cans of 3 balls of tennis each is 6 balls. 5+6=11.
Second prompt	Q: Roger has 5 tennis balls. He buys two more cans of tennis ball. Each can has 3 tennis balls. How many ball does he have now? A: **Let's think step by step**. Roger started with 5 balls. 2 cans of 3 balls of tennis each is 6 balls. 5+6=11. A: **Therefore the answer is**	The answer is 11.

– CoT prompting achieved an average accuracy of about 60%, compared to only 30% for standard prompting. This improvement demonstrates that the two-stage CoT prompting process is effective, although its performance still trails behind both few-shot learning and human-level accuracy.
– The accuracy increase was more substantial on math problems than on commonsense questions. This likely occurs because solutions to math problems often involve recurring patterns, which CoT prompting can help the model to recognize and apply.

4.2 Self-consistency

Building on CoT, Wang *et al.*, 2022 proposed "Self-Consistency," which assumes that complex reasoning problems may have multiple valid paths to a solution. Using this approach, the model is prompted with CoT and generates multiple answers by adjusting the temperature parameter T in the softmax layer. The parameter T is responsible for the entropy of this distribution.

Let's consider a vector $z \in \mathbb{R}^n$ a logit vector with n the vocabulary size. Then $\text{Softmax}(z) \in \mathbb{R}^n$ and:

$$\text{Softmax}(z)_i = \frac{e^{z_i/T}}{\sum_j e^{z_j/T}}$$

The higher the value of T, the higher the entropy of the tokens distributions. Therefore, increasing T, can force the model to explore new answers, hence bringing more diversity in the batch of answers. This variation promotes a diversity of answers, from which the most common one is selected as the final answer. This method yielded a 12% average accuracy improvement on math and commonsense benchmarks.

Huang *et al.*, 2022 pushed CoT prompting and self-consistency even further by using it to generate synthetic data for few-shot training. The whole process is illustrated in 2 extracted from Huang *et al.*, 2022.

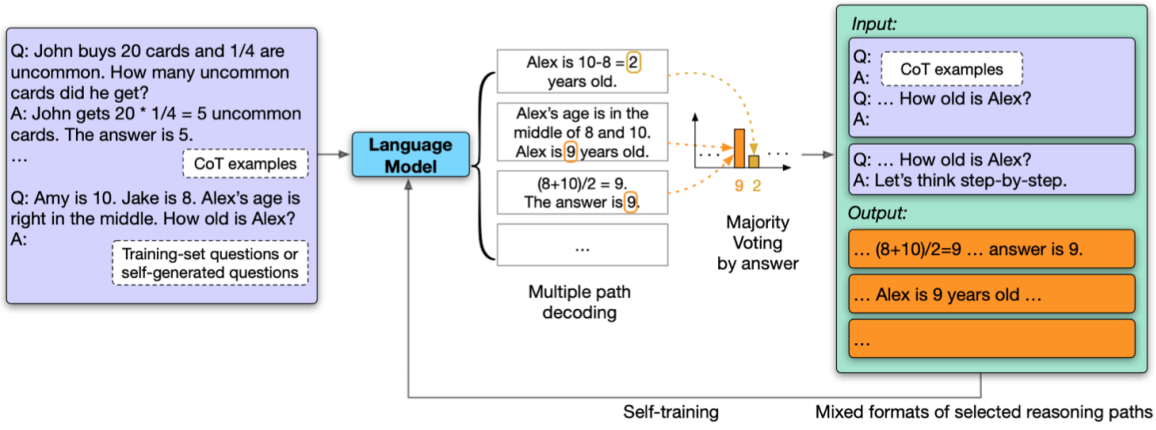

Fig. 2. Chain of Thought prompting coupled with Self consistency

The authors experimented their method with **PaLM** (540B) on various math, commonsense and language inference datasets benchmarks. This method gave new state of the art results on all the datasets.

4.3 Finetuning LLMs with Human Feedbacks

Large language models (LLMs) are often trained on extensive datasets that may contain biases, errors, or inappropriate content, which could lead the models to replicate these issues in their responses. To address this, LLMs typically go through a finetuning phase after initial training to better align their outputs with human values and preferences, which is especially crucial in sensitive fields like healthcare. Two key approaches for finetuning are Supervised Fine-Tuning (SFT) and Reinforcement Learning with Human Feedback (RLHF)Rafailov *et al.*, 2024. In SFT, the model is trained further on high-quality, curated data. This approach involves cleaning and organizing the dataset to ensure that the model learns from reliable sources, though this process can be quite time-consuming. RLHF fine-tunes the model using human judgments about its output. First, the model generates a pair of responses to a given prompt, (y_1, y_2). These responses are shown to human evaluators who pick the preferred answer, resulting in a labeled preference $(y_w > y_l|x)$, where y_w is the preferred response and y_l is the less preferred one. These preferences reflect an underlying reward function, $r^*(y,x)$, which represents the ideal responses according to human values. Rafailov *et al.*, 2024 established that this reward function r^* could be learned by solving this parametric problem

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi^{\text{pretrained}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi^{\text{pretrained}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_w|x)}{\pi^{\text{pretrained}}(y_w|x)}\right)\right]$$

where β is a parameter controlling the deviation from the pretrained model policy $\pi^{\text{pretrained}}$ and $\mathcal{D}$ is a dataset of comparisons (x, y_w, y_l). Solving this problem can be data and computation intensive since building a representative set of pairs and computing the esperance for the loss are very heavy tasks. To reduce the cost and complexity, Liu, Sferrazza, and Abbeel, 2023 introduced an efficient alternative approach called Chain of Hindsights (CoH).

- Instead of learning an entire reward function from scratch, CoH uses a few-shot setup, where each prompt provides both a good and a bad example answer to the same question. This setup directly demonstrates human preferences in a way that is simpler and less computationally demanding. By using an existing set of labeled examples, this method can convey human preferences without complex reward function training.
- During the model's generation of answers, CoH includes a masking technique to stop the model from simply copying the preferred example given in the prompt. The masking function in the model's language generation layer restricts the model from directly using words or phrases from the good example, ensuring that it forms its own answer while guided by the example. The mathematical expression for the resulting distribution of words is:

$$\log p(x) = \log\prod_{i=1}^{n}(1 - \mathbb{1}_{y_w}(x_i))\, p(x_i|x_j, j = 1\cdots n-1)$$

where $\mathbb{1}_{y_w}(x_i)$ indicates if the token x_i is used in the good example y_w.

In evaluations, Liu, Sferrazza, and Abbeel, 2023 applied CoH to a summarization task and compared it with two other versions of **GPT-J**: one fine-tuned using standard supervised fine-tuning (SFT) and one using RLHF. They measured performance across several criteria, including coverage, accuracy, coherence, and overall quality. CoH significantly outperformed both SFT and RLHF, achieving an improvement of 37% in quality.

5 AI Clinical Agents Deployment

Deploying large language models (LLMs) in healthcare is now possible, as shown by recent research, but there are key challenges. Specifically, developing relevant medical datasets and creating smaller, more efficient models are crucial for accessible, impactful deployment. A smaller model that can operate on personal devices would help make healthcare information more widely available, helping address disparities. This section reviews recent studies in these areas.

5.1 Encoding Medical Knowledge

In a significant effort to compile medical knowledge, Singhal *et al.*, 2023 created the **MultiMedQA** dataset, which organizes multiple medical datasets into a single question-answer dataset tailored for medical applications. **MultiMedQA** combines six existing datasets—**MedQA**, **MedMCQA**, **PubMedQA**, **LiveQA**, **MedicationQA**, **MMLU**—plus a seventh dataset, **HealthSearchQA**, which focuses on commonly searched health questions. Here's an overview of these datasets:

- **MedQA** and **MedMCQA** are Composed of multiple-choice questions from professional medical board exams, featuring detailed answer explanations.
- **PubMedQA** contains biomedical research questions with yes/no/maybe answers, sourced from PubMed abstracts, and includes in-depth answer explanations.
- The **MMLU** includes multiple-choice questions spanning clinical, medical, and biology topics with detailed answers.
- The **HealthSearchQA** consists of 3,173 commonly searched consumer questions. The dataset was curated using seed medical conditions and their associated symptoms.

Using **MultiMedQA**, the authors fine-tuned **PaLM** with techniques explored in this survey:

- **Instruction Tuning**: They employed **Flan-PaLM** instead of standard **PaLM**, which improved accuracy on **MedQA** by over 30%. **Flan-PaLM** uses few-shot learning with instructional examples in prompts, though the reasons for this accuracy boost are not fully analyzed.
- **Scaling**: Following previous findings, they observed that increasing model size enhances reasoning capabilities.
- **CoT prompting**: This approach showed limited impact because medical questions often have numerous valid reasoning paths that can lead to an answer. In such cases, choosing just one of these paths may not result in the most accurate or complete answer.
- **Self-consistency**: this is the best solution to deal with the weakness of CoT prompting, yielding a notable accuracy improvement.

Two versions of PaLM were trained: Flan-PaLM (with instruction tuning) and Med-PaLM (without instruction tuning). These models were assessed through a blind evaluation: clinicians rated answers from both models without knowing their source, based on factors such as scientific and clinical consensus (a), the presence of incorrect content (b), the omission of content (c), the extent of possible harm (d), the likelihood of harm (e), and possible bias in answers (f). The results are presented in Fig 3 drawn from Singhal *et al.*, 2023.

5.2 Small Language Models for Deployment

LLMs often require vast resources due to their large size, which tends to limit portability. However, Zhang *et al.*, 2024 explored the capabilities of smaller language models (SLMs) trained on large datasets to see if they could exhibit similar reasoning abilities. Their work used **tiny-Llama**, a 1.1-billion-parameter, decoder-only model, trained on a dataset of 950 billion tokens combining **SlimPajama** (open-source LLM training data) and **StarCoder** (GitHub code data). Training spanned three phases, beginning with an initial epoch using **SlimPajama**, followed by continual training with both **SlimPajama** and **StarCoder**, and ending with a cooldown phase using smaller batch sizes, and learning rate. Their model achieved a 53.75% accuracy on commonsense reasoning tasks—a 3% improvement over previous SLM benchmarks. In language understanding tasks using few-shot learning, the model averaged 21% accuracy, a 4% improvement in the SLM category. However, these results remain roughly 50% lower than current LLM benchmarks Wei *et al.*, 2022.

6 Discussion and Conclusion

This work examines various ways of enhancing reasoning with Large Language Models (LLMs) for medical applications. One key development is Cross-Lingual Sentence Representation Conneau *et al.*, 2018, which represents an initial effort to build models capable of multilingual reasoning. This capability helps address a significant issue in healthcare machine learning: limited data availability in various languages. Another advancement is reasoning over implicit knowledge Talmor *et al.*, 2020, which introduces new possibilities for classification tasks in healthcare. For example, instead of fine-tuning a model specifically for token classification, it may be possible to use the model's already encoded clinical knowledge. This can be done by presenting a classification question in natural language. An example is found in Lompo and T.-D. Le, 2024, where researchers aimed to interpret numerical values using **CamemBERT-bio** by training it to classify specific tokens. For instance, in the sentence "La FR est de 45%," they aimed to classify "45%" to understand its clinical relevance. However, following the approach of Talmor *et al.*, 2020, another way could be to phrase a question like "Is 45% a respiratory rate?" and have the model answer "yes" or "no." This method of using natural language input might enable the model to draw more effectively on its internal knowledge.

Various types of reasoning can significantly enhance how LLMs organize and interpret the vast amounts of knowledge they store. For example, defeasible reasoning allows a model to assess a hypothetical diagnosis and update it based on new information from a patient's medical records. Inductive reasoning enables a model to recognize patterns across observations from multiple patients, which could be valuable in medical research.

Fig. 3. Side to side comparison of the models performance with a clinician performance

When combined with abductive reasoning, the model could suggest possible explanations for unexpected findings. Despite the potential, there are currently no datasets tailored to train models in these reasoning types specifically for healthcare. Although previous studies such as Talmor *et al.*, 2020; Conneau *et al.*, 2018; Young *et al.*, 2022; Rudinger *et al.*, 2020; Yang *et al.*, 2024 have outlined methods for creating datasets for these reasoning approaches, developing them for medical use would require substantial medical expertise, making the process both time- and resource-intensive.

Self-improving methods like Chain-of-Thought, Chain-of-Hindsight, Self-Consistency, and Few-Shot Learning highlight the need to work with very large LLMs because high-level reasoning capabilities tend to emerge only in these larger models. These techniques are not only complementary but can also generate high-quality training data on their own, providing a self-sustaining cycle of improvement. Additionally, they are generally more robust and reliable than supervised approaches to reasoning and offer greater interpretability, as they tackle problems by breaking them down and reasoning step-by-step. Supporting this, Singhal *et al.*, 2023 demonstrated that these techniques can be particularly effective for encoding medical knowledge, bringing the model's performance closer to that of clinicians. However, as shown by Zhang *et al.*, 2024, the performance gap between large LLMs and small language models (SLMs) is still significant; even with optimization, SLMs remain far from matching the capabilities of LLMs. Therefore, large LLMs remain essential for high-level reasoning tasks, especially in complex domains like healthcare.

References

Conneau, Alexis *et al.* (2018). "XNLI: Evaluating cross-lingual sentence representations". In: *arXiv preprint arXiv:1809.05053*.

Devlin, Jacob *et al.* (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805*.

Huang, Jiaxin *et al.* (2022). "Large language models can self-improve". In: *arXiv preprint arXiv:2210.11610*.

Kojima, Takeshi *et al.* (2022). "Large language models are zero-shot reasoners". In: *Advances in neural information processing systems* 35, pp. 22199–22213.

Liu, Hao, Carmelo Sferrazza, and Pieter Abbeel (2023). "Chain of hindsight aligns language models with feedback". In: *arXiv preprint arXiv:2302.02676*.

Lompo, Boammani Aser and Thanh-Dung Le (2024). "Numerical Attributes Learning for Cardiac Failure Diagnostic from Clinical Narratives-A LESA-CamemBERT-bio Approach". In: *arXiv preprint arXiv:2404.10171*.

Miller, George A. (Nov. 1995). "WordNet: A Lexical Database for English". In: *Commun. ACM* 38.11, pp. 39–41. ISSN: 0001-0782. DOI: 10.1145/219717.219748. URL: https://doi.org/10.1145/219717.219748.

Rafailov, Rafael *et al.* (2024). "Direct preference optimization: Your language model is secretly a reward model". In: *Advances in Neural Information Processing Systems* 36.

Rudinger, Rachel *et al.* (Nov. 2020). "Thinking Like a Skeptic: Defeasible Inference in Natural Language". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Ed. by Trevor Cohn, Yulan He, and Yang Liu. Online: Association for Computational Linguistics, pp. 4661–4675. DOI: 10.18653/v1/2020.findings-emnlp.418. URL: https://aclanthology.org/2020.findings-emnlp.418.

Singhal, Karan *et al.* (2023). "Large language models encode clinical knowledge". In: *Nature* 620.7972, pp. 172–180.

Speer, Robyn, Joshua Chin, and Catherine Havasi (2017). *ConceptNet 5.5: An Open Multilingual Graph of General Knowledge*. URL: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972.

Talmor, Alon *et al.* (2020). "Leap-of-thought: Teaching pre-trained models to systematically reason over implicit knowledge". In: *Advances in Neural Information Processing Systems* 33, pp. 20227–20237.

Wang, Xuezhi *et al.* (2022). "Self-consistency improves chain of thought reasoning in language models". In: *arXiv preprint arXiv:2203.11171*.

Wei, Jason *et al.* (2022). "Chain-of-thought prompting elicits reasoning in large language models". In: *Advances in neural information processing systems* 35, pp. 24824–24837.

Yang, Zonglin *et al.* (Mar. 2024). "Language Models as Inductive Reasoners". In: *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Yvette Graham and Matthew Purver. St. Julian's, Malta: Association for Computational Linguistics, pp. 209–225. URL: https://aclanthology.org/2024.eacl-long.13.

Young, Nathan *et al.* (May 2022). "AbductionRules: Training Transformers to Explain Unexpected Inputs". In: *Findings of the Association for Computational Linguistics: ACL 2022*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, pp. 218–227. DOI: 10.18653/v1/2022.findings-acl.19. URL: https://aclanthology.org/2022.findings-acl.19.

Zhang, Peiyuan *et al.* (2024). "Tinyllama: An open-source small language model". In: *arXiv preprint arXiv:2401.02385*.