On Implementing Case-Based Reasoning with Large Language Models

Kaitlynne Wilkerson and David Leake(✉)

Luddy School, Indiana University, Bloomington, IN 47408, USA
{kwilker,leake}@indiana.edu

Abstract. Systems based on Large Language Models (LLMs), such as ChatGPT, have impressive performance but also well-known issues with erroneous output. Retrieval Augmented Generation (RAG), which typically presents the LLM with text snippets of additional knowledge retrieved from an external knowledge base, is a popular method for increasing LLM accuracy. This paper presents initial studies exploring augmenting LLMs with cases rather than snippets and prompting LLMs towards performing case-based reasoning. The studies consider four possible scenarios, exploring the potential benefit of LLMs performing different subparts of the CBR process: (1) a scenario in which the LLM is prompted to adapt a presented case, (2) a scenario in which the LLM is first prompted to perform similarity assessment to select a case from a set of candidates, and then to adapt the selected case, (3) a scenario in which the LLM is prompted to select the two most similar cases to a problem and generate an adapted/combined solution in light of both, and (4) a scenario in which the LLM selects the nearest neighbor and nearest unlike neighbor and generates an adapted/combined solution based on both. Results of tests using Llama and ChatGPT are encouraging for the accuracy benefits of providing LLMs with cases and raise questions for future study.

Keywords: Case-Based Reasoning · ChatGPT · Llama · Large Language Models · Retrieval Augmented Generation

1 Introduction

In 2022, ChatGPT captured public attention. Not only did it demonstrate remarkable capabilities but it became notorious for incorrect, confusing and disruptive output—including output the system simply "hallucinated". As competing systems became publicly available it became clear that such problems were not just characteristics of ChatGPT, but instead results of fundamental characteristics of the Large Language Models (LLMs) underlying such systems [5,19]. Nevertheless, LLMs are rapidly being assimilated into everyday technologies, resulting in a pressing need to develop methods to alleviate LLM errors.

A promising method for reducing factual errors by LLMs is to provide them with external knowledge regarding the prompt topic. Retrieval Augmented Generation (RAG) operates by retrieving text related to a query and adding it to the

J. A. Recio-Garcia et al. (Eds.): ICCBR 2024, LNAI 14775, pp. 404–417, 2024.
https://doi.org/10.1007/978-3-031-63646-2_26

prompt. This approach has been shown to improve LLM responses [3,18]. RAG typically provides the LLM with knowledge in the form of snippets reflecting individual facts. However, providing knowledge in the form of cases is an appealing alternative [18]. This paper proposes going beyond augmenting LLM knowledge with cases, by specifically prompting the LLM to perform CBR on those cases. Watson [17] observed that Case Based Reasoning (CBR) is a methodology of reasoning from experiences rather than a specific technology: it can be implemented with any technology. In that vein, this paper explores implementing a case-based classification process using LLMs.

LLM-based implementations of CBR—or of specific parts of the CBR cycle—might provide benefits both to LLMs and to CBR. Having LLMs perform CBR could potentially improve both the accuracy and explainability of LLM-based systems, for three reasons. First, ideally, grounding LLM reasoning in similar cases might reduce the risk of hallucination when generating solutions. Second, guiding an LLM through a CBR-like process might increase system accuracy by helping focus the LLM on reasoning related to similar problems. Third, because cases are naturally intuitive explanations for human users [6], grounding LLM reasoning in cases might aid explanation of system decisions: humans find cases useful as explanations [2,4]. Being able to present the user of an LLM not only with a solution but with a relevant case to compare would help in assessing solutions, and might increase user trust when both are consistent [4].

Conversely, for certain task domains, LLM-based implementations of CBR processes could benefit CBR. If the parts of the CBR process depending on rich knowledge could be performed by an LLM, it could have transformative impact for increasing the scope of CBR applications, by facilitating applying CBR to knowledge-rich domains for which formally encoded knowledge is unavailable, expensive, or difficult to encode. We note the limitation that LLMs could still produce erroneous results. However, part of the CBR process is for CBR systems to assess proposed results for correctness, and when final system results are presented to users, users can assess them in light of the cases on which they were based—which has been shown effective for building justified trust [4].

To explore the effects of providing retrieved cases to an LLM and prompting it to perform parts of the CBR process, we performed an experiment using two LLMs, ChatGPT 3.5 and Llama 2 [15], for a classification task in a medical triage domain. Triage decisions require rich world knowledge; this task illustrates a use of LLMs to bypass knowledge acquisition for a complex real-world task. We tested performance with three types of prompts: (1) the baseline of prompting the LLM for a direct solution without providing any case information, (2) prompting for a solution after providing a similar case to the LLM, and (3) prompting for the LLM to do similarity assessment to select case(s) and then prompting it to adapt them to generate a solution. We also tested whether the LLM using different numbers and types of cases (using only similar cases, or using both a similar case and the nearest unlike neighbor) affected accuracy.

We found that using cases could improve the classification performance of the LLM, which demonstrates cases as a source of useful information for our testbed

task, and that the adaptation ability of an LLM impacted the best methods for utilizing cases. We also found that while ChatGPT and Llama 2 both performed similarity assessment poorly, their adaptation rates were quite different and this led to certain prompts performing better than others. Overall, we consider the results an encouraging beginning. We close the paper by discussing next steps for building on these first results.

2 Background

Issues with LLMs: LLMs combine remarkable conversational capabilities with limitations such as hallucinations, and, unless combined with other systems, a lack of episodic and causal knowledge (e.g., [5]). Their facility at generating language can give the appearance of reasoning [8,12], but they struggle at tasks such as planning [16]. Because LLMs rely on statistical profiles of human language and tasks such as identifying facts are not statistical in nature, LLMs are inherently ill-suited to them [5]. In addition, their reasoning is based on generalizations, which are necessarily lossy, and can be distorted by the LLM when generating a response [13]. Issues may also stem from prompts asking for information beyond the LLM's training data, such as time-sensitive information [3]. Such issues are well known and current efforts aim at alleviating them, e.g., with augmented models that draw on additional methods when needed [10].

One approach to increasing LLM accuracy is to integrate external knowledge into the model, either by integrating the information into the prompt or by providing the LLM with a knowledge base that it can query on its own [3,9]. This has been shown to improve LLM performance [3,9] and reduce many of the issues discussed above, including identifying potential hallucinations by comparing the response to the retrieved knowledge [13]. While this does not (and cannot) fix the fact that LLMs are limited by their statistical nature, it does help to address the knowledge problem by providing the LLM with additional knowledge on a topic that has not been distorted or generalized. The usefulness of cases as a form of knowledge in CBR applications [1] and prevelance of case-based reasoning by people [7] suggests the potential promise of providing a different type of knowledge—case knowledge—to LLMs from an external case base.

Knowledge Integration Improves LLM Responses: Retrieval Augmented Generation is one of the most widely adopted strategies for integrating external knowledge into LLMs [3]. RAG can be implemented in many ways, but the core process contains three main steps: First, partition a corpus of text into n chunks and vectorize. Second, given some query, assess similarity between the query and each of the vectorized n chunks. Third, present the retrieved information to an LLM so that it can be used to generate a response [3]. A demonstration of the benefit of retrieved information can be found through the study of LLM performance on commonsense reasoning benchmarks when contextual knowledge was provided [9]. Retrieval-based methods for integrating knowledge improved the accuracy of LLMs on each of the benchmark datasets.

The spirit of RAG relates to CBR, but RAG differs in the type of knowledge retrieved and how that knowledge is used. RAG tends to focus more on knowledge statements that can add context to a prompt [3,9,12], while CBR uses cases aimed at the task at hand. Also, RAG may provide general information, while CBR provides specific concrete episodes. We seek to understand the impact of providing cases and prompting the LLM to provide solutions by CBR.

3 Questions for Implementing CBR with LLMs

This paper presents the start of a research program to better understand the capabilities of LLMs with respect to CBR and whether previous research showing gains in accuracy from external knowledge [3,9] extends not only to providing cases to an LLM (e.g., [18]), but to guiding the LLMs through a CBR-like process. Understanding the potential of case-augmented generation and of implementing CBR with LLMs will require answering questions such as:

1. Do LLMs benefit from having a similar prior case as a starting point?
2. Do LLMs benefit from having multiple cases for a single problem?
3. Do LLMs benefit from having both example and counterfactual cases?
4. How well can LLMs assess case similarity?
5. How well can LLMs perform case adaptation?
6. What types of prompts are most effective for guiding LLM similarity assessment and case adaptation?
7. To what extent, and how, do the above depend on characteristics of specific LLMs and types of task domains?

Each of these questions is a substantial topic for which a definitive answer would require extensive studies. The purpose of this paper is to open the door to future investigations by gathering initial experimental data and observations relevant to questions 1–5.

4 Experimental Design

As a first step towards answering questions 1–5 in the previous section, we conducted an experiment comparing the baseline accuracy of OpenAI's ChatGPT and Meta's Llama 2 to their accuracy when provided with cases and either (1) prompted to perform a sort of implicit CBR—to solve the new problem based on the case—or (2) prompted to follow the CBR cycle more explicitly, by providing cases and prompting the LLM to first perform similarity assessment and then case adaptation.

4.1 Large Language Models Used

OpenAI's ChatGPT 3.5[1] and Meta's Llama 2 70b-chat [15], which hereafter will be referred to as ChatGPT and Llama 2 respectively, were used in this experiment. ChatGPT was selected to give an illustration of the current commercial

[1] https://openai.com/blog/chatgpt.

state of the art. However, because it is not possible to set parameters or control possible updates, the ChatGPT results should only be seen as suggestive—they are not replicable. For replicability, the experiments were also run on the open source LLM, Llama 2, for which we could control the model parameters.

All interactions with ChatGPT were done by hand through the ChatGPT website over the course of a single day. Llama 2 was hosted locally and automatically invoked on Indiana University's Big Red 200 supercomputer using the Llama.cpp[2] and llama-cpp-python projects.[3] We tested several combinations of parameters but found the best performance with a temperature of 0. The top_p value was set at 0.9, but this was not found to be as impactful as temperature.

All cases were presented to the LLMs via prompts. Because of limitations on prompt size, we did not present the entire case base to the LLM. Instead, a subset of cases was selected from the case base by an initial retrieval phase, using the retrieval component of a k-NN system, and that subset was provided to the LLM. The full k-NN system was also used as a baseline for our experiments.

4.2 Test Case Base

Experiments used a case base for medical triage classification. This task is an existing medical AI task area; the use of AI in medical triage increased as a result of the Covid-19 pandemic [14]. Both LLMs tested in this paper have been applied to medical domains [11]. Triage depends on extensive rich real-world knowledge, making it the type of domain for which bypassing traditional CBR knowledge acquisition by using LLM knowledge would be desirable. We note that supporting triage using public LLMs is problematic for medical applications due to data privacy concerns. Real application would require using local versions that do not retain query data.

Cases were constructed using data collected from a primary triage dataset posted on Kaggle.[4] The original dataset contained cases for 1,267 patients and tracked 24 different vital signs and medical assessments. Each patient was assigned a Korean Triage and Acuity Scale (KTAS) number of 1 through 5, with 1 indicating patients most in need of immediate medical attention and 5 indicating patients least in need of immediate medical attention. When the data set was generated, this value was obtained by three triage experts reviewing the patient's condition and assigning a number. We narrowed the case data down to seven features commonly discussed in triage literature [20]: sex and age of the patient, heart rate, respiratory rate, mental state, blood pressure and the patient's chief complaint upon entrance to the Emergency Room. These features tend to be the most commonly used by a range of different triage methods [20]. Sex, mental state and chief complaint were non-numerical values, with sex and mental state being categorical while chief complaint was a simple string value. Mental state contained four different values: Alert, Verbal Response, Pain Response,

[2] https://github.com/ggerganov/llama.cpp.
[3] https://github.com/abetlen/llama-cpp-python.
[4] https://www.kaggle.com/datasets/ilkeryildiz/emergency-service-triage-application.

and Unresponsive. The original dataset contained many missing values, especially for heart rate, respiratory rate, blood pressure and chief complaint. We removed instances with missing values from the pool of candidate cases. The remaining cases were divided by class and 5 cases per class were used to create the testing set (25 total), with the rest being assigned to the training pool. From the pool of training cases, a random sample of 102 cases was selected, with at least one case representing each class, to form the case base. This was done to streamline the retrieval and weight configuration processes.

We note that because the dataset was released in 2019, it is possible that either of the test LLMs was trained with this data. If so, that training could be expected to produce a favorable setting for direct LLM performance, potentially reducing the benefit of providing cases. However, given the lossy nature of LLM learning, the specific information provided by cases might still be valuable.

4.3 Case Retrieval

A k-NN system was used as a performance baseline and its retrieval mechanism was used to retrieve cases to present to the LLM. The k-NN based retrieval used weighted Euclidean distance, with weights selected by hill climbing to maximize k-NN accuracy. The best accuracy was 48% when the weights were set at Sex = 0.001, Age = 0.001, Heart Rate = 0.001, Respiratory Rate = 1.0, Mental State = 0.25, Chief Complaint = 0.25, and Blood Pressure = 0.001. Categorical data distance was 1 for non-matching categories and 0 for matching categories. Textual data was compared by semantic similarity implemented using HuggingFace's sentence transformer library and the all-mpnet-base-v2 model.[5] Vectorized text was assessed for semantic similarity using pyTorch's cosine similarity function.

4.4 Prompt Types

We tested three prompt types, all of which were in textual format:

- **Direct Solution:** Directly asks LLM for a solution without providing any additional information. LLM prompt form: *[instructions + problem case]*
- **Implicit CBR (ICBR):** Uses k-NN to provide LLM with one or more prior cases. LLM prompt form: *[instructions + problem case + prior case(s)]*
- **Explicit CBR (ECBR):** Uses k-NN to obtain the 10 most similar cases to the problem case and provides those to LLM for similarity assessment (only 10 cases were provided, to control prompt size). Instructions and additional information are designed to have LLM step through the CBR process. LLM prompt form: *[instructions + problem case + set of prior cases]*

The Direct Solution prompts establish a non-CBR LLM performance baseline for the triage task. ICBR prompts are designed to test how well an LLM can derive an answer from a nearest neighbor case without explicit instructions

[5] https://huggingface.co/sentence-transformers/all-mpnet-base-v2.

on how to do so. Tests of ICBR prompts assess (1) the LLM's capability to apply a previous case without specific guidance, and (2) whether the knowledge embedded in the LLM can lead to improved performance over that of the baseline knowledge-light k-NN system. Finally, the ECBR prompts aim to guide an LLM through a CBR process of similarity assessment and adaptation.

For both the ICBR and ECBR prompt types, we tested three different formulations that differed by the cases provided to the LLM:

- **1NN formulation:** Provides only the Nearest Neighbor
- **2NN formulation:** Provides the top two Nearest Neighbors
- **NUN formulation:** Provides the Nearest Neighbor and a counterfactual (the Nearest Unlike Neighbor)

The 2NN formulation was included to potentially increase the robustness of the model's response by providing more cases, and potentially helping alleviate issues in selecting the most similar case. The NUN formulation provided a counterfactual, to potentially help delineate the border between triage categories.

4.5 Prompt Instructions

Depending on the prompt type and the formulation of case information provided, each test used a slightly different set of instructions (Table 1). The instructions for all tests were crafted during an extensive round of pre-testing to assess how different prompt wordings affected the response given. For reasons of space, the pre-testing will not be discussed in this paper, but the prompts below were designed following the lessons learned from the best performing prompts during pre-testing. All prompts started with instructions and then provided the current patient's status and the cases of previous patient(s), if applicable, presented in a textual attribute-value form, for example:

> *Patient Status: Sex: Female, Age: 50, Chief complaint: Blood Pressure. Low, Mental state: Pain Response, Heart Rate: 37, Respirations: 28, Blood Pressure: 50/33*

Direct and Indirect Solution: The Direct Solution prompt was designed to provide the system with a base set of information about the classification task, in the context of a task scenario. The ICBR prompts all started with the same instructions, but, depending on the formulation, contained different amounts of case information. The system was explicitly told to use the similarities and differences between the current case and prior case(s) to make a decision on the classification of the current case. Thus, it can be seen as a prompt for performing adaptation.

Explicit CBR: The ECBR prompts differed depending on the number and type of cases the LLM was asked to select. For each formulation, the LLM was provided with 10 cases and asked to select some case(s) to work with. For the 1NN formulation, it was only asked to select the most similar previous patient. The

Table 1. Prompt(s) used for each prompt type and formulation.

Type and Formulation	Prompt(s)
Direct Solution	*Instructions: You are helping triage patients following a disaster and have limited medical personnel and resources. Given the following information about a patient in a triage situation, assign them a triage number of 1, 2, 3, 4 or 5, where 1 is assigned to the patients most in need of immediate medical attention and 5 is assigned to patients least in need of medical care.*
ICBR (All Formulations)	**Adaptation:** *Instructions: You are helping triage patients following a disaster and have limited medical personnel and resources. You need to assign the current patient with a specific tag number of 1, 2, 3, 4 or 5, where 1 is assigned to the patients most in need of immediate medical attention and 5 is assigned to patients least in need of medical care. Each tag number indicates the likelihood of the patient dying without immediate medical treatment. Given a previous patient and their condition, use the similarities and differences in vital signs to assign a tag number to the current patient.*
ECBR 1NN	**Similarity Assessment:** *Instructions: Given a current patient status, choose a previous patient whose condition is most similar to the current patient.* **Adaptation:** *Instructions: You are helping triage patients following a disaster and have limited medical personnel and resources. You need to assign the current patient with a specific tag number 1, 2, 3, 4 or 5, where 1 is assigned to the patients most in need of immediate medical attention and 5 is assigned to patients least in need of medical care. Each tag number indicates the likelihood of the patient dying without immediate medical treatment. Given the most similar previous patient's condition and associated triage number, use the similarities and differences in vital signs to assign a triage number to the current patient.*
ECBR 2NN	**Similarity Assessment:** *Instructions: Given a current patient status, choose a previous patient whose condition is most similar to the current patient.* **Similarity Assessment:** *What is the next most similar patient?* **Adaptation:** *Instructions: You are helping triage patients following a disaster and have limited medical personnel and resources. You need to assign the current patient with a specific tag number 1, 2, 3, 4 or 5, where 1 is assigned to the patients most in need of immediate medical attention and 5 is assigned to patients least in need of medical care. Each tag number indicates the likelihood of the patient dying without immediate medical treatment. Using the two most similar previous patients condition's and associated triage numbers, use the similarities and differences in vital signs to assign a triage number to the current patient.*
ECBR NUN	**Similarity Assessment:** *Instructions: Given a current patient status, find the nearest neighbor to the current patient. Assuming the tag number from the nearest neighbor could be the tag number for the current patient, find the nearest unlike neighbor to the current patient.* **Adaptation:** *Instructions: You are helping triage patients following a disaster and have limited medical personnel and resources. You need to assign the current patient with a specific tag number 1, 2, 3, 4 or 5, where 1 is assigned to the patients most in need of immediate medical attention and 5 is assigned to patients least in need of medical care. Each tag number indicates the likelihood of the patient dying without immediate medical treatment. Given the similarities and differences between the current patient, the nearest neighbor and the nearest unlike neighbor and assuming that the current patient tag number is unknown, assign a triage number to the current patient.*

2NN formulation asked for the most similar previous patient and then followed up with a second question asking for the next most similar previous patient. The NUN formulation specifically asked for the Nearest Neighbor and Nearest Unlike Neighbor in the same prompt. Then the LLM was provided with the triage scenario and asked to provide a solution that took into account the similarities and differences between the current patient and selected case(s).

A slight difference from the NUN prompt is that the similarity assessment prompt told the LLM to use the NN's classification as the current patient's classification, to find the NUN. During pre-testing, it was unclear if the LLM knew how to find the NUN, so instructions were added. To avoid biasing results, the LLM is instructed to assume it does not know the current patient's classification.

4.6 Procedure and Analysis

For each of the 25 test cases, a prompt was created for each of the prompt and formulation types, resulting in seven prompts per test case. Each of the 175 prompts was input by hand into the ChatGPT web interface and automatically delivered to Llama 2 via a python script. Responses were catalogued and checked to ensure that each LLM had not hallucinated any additional information that it ascribed to the input case. Any response in which a hallucination was found was discarded and the test was repeated until no hallucinations occurred in the response. We note that the ability to filter out such hallucinations follows from having the original case to compare; thus, asking an LLM to reason from a prior case enables a basic form of "sanity check" on the LLM output.

The LLM-generated classifications were compared to the KTAS expert values to judge accuracy using each prompt type. LLM responses were also compared to the expert value of the nearest neighbor (selected by the k-NN similarity metric for ICBR or LLM-chosen for ECBR) to assess whether the LLM performed adaptation occurred. For the ECBR prompt types, the LLM's ability to perform similarity assessment was evaluated by comparing the cases the LLMs selected as most similar to those selected as most similar by k-NN retrieval.

5 Results and Discussion

Triage is a challenging real-world domain; a patient's status may change rapidly and different levels of expertise or experience may result in slightly different categorizations [20]. Consequently, we included three different categories of "correctness" with varying amounts of latitude:

- *Strict Accuracy*: The LLM's response matched the correct classification
- *Correct or within 1 class (higher)*: Given the risks associated with erroneously lower triage scores, this category accepts safe near misses.
- *Correct or within 1 class (higher or lower)*: The LLM's response either matched the correct classification exactly or was within one category on the triage scale. This category includes near misses that might entail risk.

Table 2 shows the results from the prompt type accuracy tests.

Strict Accuracy Results: Direct Solution correctly classified triage patients 28% of the time, which was on par with the unweighted k-NN classifier. Both implicit and explicit CBR prompt types performed on par with or better than the Direct

Table 2. Accuracy by prompt type, case provided, and LLM. This table includes strict accuracy, accuracy with including responses 1 class higher than correct, and accuracy including responses 1 class higher or lower than correct. The best performing prompt type in each column is in bold.

Prompt Type and Formulation	Strict Accuracy	Correct within 1 class (higher)	Correct within 1 class (higher or lower)	Strict Accuracy	Correct within 1 class (higher)	Correct within 1 class (higher or lower)
	Baselines					
Unweighted 1-NN	28%	56%	64%	28%	56%	64%
Weighted 1-NN	48%	60%	76%	48%	60%	76%
	ChatGPT			Llama 2		
Direct Solution	28%	64%	72%	28%	52%	**72%**
ICBR 1NN	**60%**	68%	**80%**	**56%**	**60%**	68%
ICBR 2NN	40%	48%	64%	44%	56%	**72%**
ICBR NUN	44%	56%	68%	28%	40%	48%
ECBR 1NN	36%	56%	64%	44%	48%	60%
ECBR 2NN	44%	**72%**	76%	40%	48%	68%
ECBR NUN	28%	28%	44%	28%	40%	56%

Solution prompts for both LLMs. This suggests that in the test domain, providing cases can increase the accuracy of LLMs. It also suggests that even if one or both of the LLMs tested were trained on the dataset, providing the specific information on cases improves the LLM response over generating a solution with only network embedded information.

The ICBR prompts performed on par with or better than their ECBR counterparts, with the ICBR 1NN prompt formulations performing the best of all prompt types and outperforming the weighted k-NN classifier for both LLMs tested. Performance with the 2NN and NUN formulations differed slightly based on the LLM used for testing. With ChatGPT, ICBR NUN slightly outperforms ICBR 2NN. However, this is reversed for Llama 2. Among the ECBR prompts, the 2NN formulation performs well, but depending on the model may not be best. With ChatGPT, 2NN outperforms 1NN and NUN. However, with Llama 2, 1NN and 2NN perform very similarly, with 2NN slightly under performing. This seems to suggest that less knowledge is more when the case provided is the case selected as most similar using the k-NN weights (ICBR). When the LLM is used to judge similarity, additional cases generally yield higher accuracy, with slight differences in performance for different LLMs.

A surprising result was the relatively poor performance of ECBR prompts compared to their ICBR counterparts. We hypothesized that this may be the result of poor similarity assessment by LLMs. To test this, LLM-chosen and k-NN-chosen most similar case(s) were compared against each other for each test case for each prompt type. Table 3 displays the rate at which each LLM selected the same case as the optimized retrieval of the k-NN classifier. Both have comparably low performance, though Llama 2 outperformed ChatGPT when selecting the 2NN case for the ECBR 2NN prompt and the NUN for the ECBR NUN prompt.

Table 3. Similarity Assessment performance for ChatGPT and Llama 2. The ECBR 2NN and NUN rows show the percentage of correctly selected Nearest Neighbor (NN) and either second Nearest Neighbor (2NN) or Nearest Unlike Neighbor (NUN).

Target cases		ChatGPT	Llama 2
ECBR 1NN		20%	28%
ECBR 2NN	NN	24%	28%
	2NN	4%	16%
ECBR NUN	NN	16%	12%
	NUN	16%	32%

Correct within One Class: As discussed in the strict accuracy results, cases generally improved LLM performance, but only one prompt type and case formulation outperformed the weighted k-NN classifier: ICBR 1NN. Using a looser accuracy criterion provides additional context. As expected, the accuracy rates of the prompt types, direct solution, and k-NN classifiers increased with the looser criterion. However, fewer prompt types performed on par with or above direct solution levels. Overall, when ICBR 1NN was evaluated with ChatGPT, it outperformed the direct solution when considering accuracy within one class higher and within one class higher or lower. When evaluated with Llama 2, ICBR 1NN outperformed direct solution only when considering accuracy within one class higher and slightly underperformed when considering accuracy within one class higher or lower. Considering the risk associated with triaging a patient lower than is actually appropriate, ICBR 1NN can still be considered a contender in this domain. The remaining prompt types and formulations with equivalent or better performance in the expanded accuracy categories depended on the LLM used for evaluation. ECBR 2NN outperformed the Direct Solution baseline in both expansion categories when ChatGPT was used for evaluation, but not with Llama 2. With Llama 2, ICBR 2NN outperformed the Direct Solution baseline when the expansion was one class higher, but performed equivalently when the expansion was one class in either direction. We suspect that difference in ECBR vs ICBR may be attributed to adaptation ability, as will be discussed shortly, but the important similarity between these results is the benefit of providing a second similar case, in the 2NN case formulation. This suggests that while providing one case is generally sufficient for improving performance, under certain circumstances providing more cases could be useful.

Another interesting observation is that by the looser performance criteria, providing the LLM with case knowledge leads to equivalent or improved performance over k-NN classifiers. The improved performance was seen with ICBR 1NN and ECBR 2NN prompts evaluated on ChatGPT, whereas the equivalent performance was primarily seen with ICBR 1NN and ICBR 2NN evaluated on Llama 2. This is particularly interesting in light of the different domain knowledge available to the LLMs and the k-NN classifiers. The unweighted k-NN classifier had access to the entire case base (102 cases) and the weighted k-NN classifier had access to optimized weights along with the entire case base. The

LLMs did not have access to the optimized weights and seem to perform similarity assessment poorly (for ECBR prompts), yet they still generally performed at the level of k-NN and sometimes surpassed it. This supports the benefit of harnessing LLMs for this task, and the usefulness of providing cases.

Overall, the accuracy tests demonstrated a clear pattern of cases improving the performance of LLMs over direct solution baselines and established several different prompt and case formulation pairings that consistently led to increased accuracy. A common theme among these results was that the Llama 2 tests tended to show weaker performance patterns than ChatGPT. This may reflect the size difference of the models, as ChatGPT is over twice the size of Llama 2. Regardless, both models generally support the same trends.

Adaptation: We considered adaptation to have occurred if the nearest neighbor case (k-NN-provided for ICBR and LLM-chosen for ECBR) had a different solution from that proposed by the LLM. For each prompt/formulation we noted:

- How many times did adaptation occur?
- If adaptation occurred, was the adapted class correct?
- How many times did adaptation occur when the nearest neighbor had the correct classification (i.e., when adaptation was unnecessary)?

Table 4 displays the results from each question. Note that the second and third columns for each LLM are not out of the 25 test cases but out of the total number of adapted cases. Generally, the rates of adaptation among ChatGPT responses were above 50%, while Llama 2 rarely had adaptation rates that high, which indicates that ChatGPT was more likely to adapt than Llama 2. The data reveal two interesting trends that may help explain the difference in results between ChatGPT and Llama 2. First, Llama 2 tends to adapt much less often than ChatGPT and this may hurt accuracy for the ECBR prompt types. With ChatGPT, ECBR prompts had consistently higher rates of adaptation occurring than ICBR prompts, except for the NUN formulations. The higher rates of adaptation among ECBR prompts may be due to the observed limitations in ChatGPT similarity assessment. Llama 2 and ChatGPT perform

Table 4. This table displays LLM adaptation capabilities, including the total percentage of adapted cases, adapted cases that had a correct solution (strict accuracy), and adapted cases where the Nearest Neighbor (NN) had the correct solution (i.e., no adaptation should have occurred).

	ChatGPT			Llama 2		
Prompt Type and Formulation	Adapted Cases	Adapted Cases That Were Correct	Adapted Cases Where NN was Correct	Adapted Cases	Adapted Cases That Were Correct	Adapted Cases Where NN was Correct
ICBR 1NN	56%	35%	14%	64%	31%	18%
ICBR 2NN	52%	15%	30%	16%	0%	25%
ICBR NUN	72%	27%	33%	48%	0%	50%
ECBR 1NN	68%	35%	29%	12%	33%	0%
ECBR 2NN	84%	38%	14%	32%	25%	38%
ECBR NUN	76%	31%	31%	52%	23%	31%

similarity assessment about equally well. However, ChatGPT seems to perform more adaptation to make up for it, which could account for the higher accuracy rates during ChatGPT evaluation. Second, rates of correct adaptation were only roughly equivalent under 1NN case formulations; all other formulations had decreases of at least 10% in correct adaptation with Llama 2. This may also explain why the ECBR 2NN results were not as strong with Llama 2 as they were with ChatGPT. Despite the very poor ICBR 2NN percentages of correct solutions from adaptation, the low likelihood of adaptation occurring probably reduced the accuracy drop from this prompt type.

These results illustrate that for our tests, adaptation ability was LLM-specific and that it impacts the performance of both ECBR and ICBR prompt types as well as which case formulations work best with these prompts. Adaptation ability will be a rich area for further study for implementing CBR as a process for LLMs.

6 Future Work

This paper considers the potential benefit of providing LLMs with cases and prompting them to perform case-based reasoning. We conducted an experiment exploring whether cases could be used effectively as a source of external knowledge, analogously to other LLM knowledge integration methods. Encouragingly, our study suggests that CBR may be beneficial for reducing LLM errors, but this requires substantiation on tests with additional domains and LLMs, for multiple types of CBR tasks.

In addition, we would like to directly compare the accuracy using prompts containing CBR retrieved information to prompts containing RAG-retrieved snippets of general information, to quantify the impact of CBR in comparison. Finally, considering that a key motivation for improving the responses of LLMs is to ensure their trustworthiness, it would also be beneficial to test whether responses generated with cases are considered to be more intuitive and trustworthy than responses generated with knowledge snippets, and whether presenting users with cases as well as LLM solutions increases trust and the ability for users to assess the quality of LLM responses.

Acknowledgements. This work was funded by the US Department of Defense (Contract W52P1J2093009). This research was supported in part by Lilly Endowment, Inc., through its support for the Indiana University Pervasive Technology Institute.

References

1. Cheetham, W., Watson, I.: Fielded applications of case-based reasoning. Knowl. Eng. Rev. **20**(3), 321–323 (2005)
2. Cunningham, P., Doyle, D., Loughrey, J.: An evaluation of the usefulness of case-based explanation. In: Ashley, K.D., Bridge, D.G. (eds.) ICCBR 2003. LNCS (LNAI), vol. 2689, pp. 122–130. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-45006-8_12

3. Gao, Y., et al.: Retrieval-augmented generation for large language models: a survey. arXiv preprint arXiv:2312.10997 (2023)
4. Gates, L., Leake, D., Wilkerson, K.: Cases are king: a user study of case presentation to explain CBR decisions. In: Massie, S., Chakraborti, S. (eds.) ICCBR 2023. LNCS, vol. 14141, pp. 153–168. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-40177-0_10
5. Hammond, K., Leake, D.: Large language models need symbolic AI. In: Proceedings of the 17th International Workshop on Neural-Symbolic Learning and Reasoning, La Certosa di Pontignano, Siena, Italy, vol. 3432, pp. 204–209 (2023)
6. Leake, D.: CBR in context: the present and future. In: Leake, D. (ed.) Case-Based Reasoning: Experiences, Lessons, and Future Directions, pp. 3–30. AAAI Press, Menlo Park (1996)
7. Leake, D.: Cognition as case-based reasoning. In: Bechtel, W., Graham, G. (eds.) A Companion to Cognitive Science, pp. 465–476. Blackwell, Oxford (1998)
8. Lewis, P., et al.: Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Advances in Neural Information Processing Systems, vol. 33, pp. 9459–9474 (2020)
9. Liu, J., et al.: Generated knowledge prompting for commonsense reasoning. arXiv preprint arXiv:2110.08387 (2021)
10. Mialon, G., et al.: Augmented language models: a survey (2023)
11. Nievas, M., Basu, A., Wang, Y., Singh, H.: Distilling large language models for matching patients to clinical trials. J. Am. Med. Inform. Assoc. ocae073 (2024)
12. Paranjape, B., Michael, J., Ghazvininejad, M., Zettlemoyer, L., Hajishirzi, H.: Prompting contrastive explanations for commonsense reasoning tasks. arXiv preprint arXiv:2106.06823 (2021)
13. Peng, B., et al.: Check your facts and try again: improving large language models with external knowledge and automated feedback. arXiv preprint arXiv:2302.12813 (2023)
14. Prakash, A.V., Das, S.: Would you trust a bot for healthcare advice? An empirical investigation. In: PACIS, p. 62 (2020)
15. Touvron, H., et al.: Llama 2: open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
16. Valmeekam, K., Sreedharan, S., Marquez, M., Olmo, A., Kambhampati, S.: On the planning abilities of large language models (a critical investigation with a proposed benchmark) (2023)
17. Watson, I.: Case-based reasoning is a methodology not a technology. Knowl.-Based Syst. **12**(303–308) (1996)
18. Wiratunga, N., et al.: CBR-RAG: case-based reasoning for retrieval augmented generation in LLMs for legal question answering. arXiv preprint arXiv:2404.04302 (2024)
19. Xu, Z., Jain, S., Kankanhalli, M.: Hallucination is inevitable: an innate limitation of large language models (2024)
20. Yancey, C.C., O'Rourke, M.C.: Emergency department triage. In: StatPearls [Internet]. StatPearls Publishing (2022)