CEUR-WS.org/Vol-3903/AIxHMI2024_paper3.pdf

Rule enforcement in LLMs: a parameter efficient fine-tuning approach with self-generated training dataset

Daniele Franch[1,†], Pierluigi Roberti[1,†] and Enrico Blanzieri[1,†]

[1]*University of Trento, 38100 Trento, Italy*

Abstract

Large Language Models (LLMs) often have implicit knowledge of domain-specific rules, such as age requirements for obtaining a driver's license, but may not consistently apply this knowledge in conversations. In this paper, we explore a method for fine-tuning LLMs using datasets generated by the LLM itself. The goal is to explicitly enforce specific rules, such as declaring ineligibility if the age requirement is not met, within a defined context. We evaluate whether this fine-tuning approach enables the model to recognize the need to apply relevant knowledge in other contexts, such as marriage eligibility, where the LLM already has knowledge of the underlying criteria. Our results show that after fine-tuning, the LLM not only applies the rule in the training contexts, but also generalizes this behavior to enforce the rule in different domains. This suggests that fine-tuning, even with self-generated datasets, can improve the ability of the LLM to apply its knowledge more consistently, leading to more reliable performance in rule-based scenarios.

Keywords

Large Language Models (LLMs), Low-Rank Adaptation (LoRA), Dataset augmentation, Self-generated training dataset, Domain-specific behavior

1. Introduction

Large Language Models (LLMs) such as GPT-3 [1], BERT [2], and their successors represented a big leap in advance for Natural Language Processing, demonstrating the ability to generate and interact with human language. Trained on large amounts of text, these models acquire a broad knowledge that includes facts, linguistic patterns, and specialized rules. Nevertheless, LLMs face limitations when they are required to apply domain-specific rules or state when certain criteria are not fulfilled. In fact, even if they have the knowledge of domain-specific requirements, they often do not explicitly state or enforce this knowledge. As a result, even if an LLM has the correct knowledge about a topic (e.g., the minimum age for obtaining a driving license), it may not use this knowledge in its responses like a human would. For many practical applications, such as legal advice or eligibility verification, consistent and contextual enforcement of specific rules, and thus enforcement of the associated knowledge, is critical. For example, a banking LLM may need to determine a user's eligibility for financial products based on criteria such as income or credit history. Failure to apply these rules consistently can result in inaccurate information or service disruptions.

To address these challenges, we propose a fine-tuning approach based on a self-generated dataset to allow LLMs to enforce specific rules, such as age requirements, within defined contexts. We evaluate whether this approach improves the model's ability to generalize rule enforcement across domains, demonstrating consistent rule application not only in trained scenarios but also in novel, unseen contexts.

Italian Workshop on Artificial Intelligence for Human Machine Interaction (AIxHMI 2024), November 26, 2024, Bolzano, Italy

[†] These authors contributed equally.

daniele.franch@unitn.it (D. Franch); pierluigi.roberti@unitn.it (P. Roberti); enrico.blanzieri@unitn.it (E. Blanzieri)
https://webapps.unitn.it/du/it/Persona/PER0004537/Didattica (E. Blanzieri)
0009-0003-4820-8900 (D. Franch)

© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

Our contributions can be summarized as follows:

- We introduce a method for fine-tuning LLMs, using a dataset automatically generated by the same LLM and extended by paraphrasing or augmentation by the same or another LLM, to learn the explicit application of a given rule in a constrained context;
- We test the ability of the fine-tuned model to generalize this behavior to other contexts, demonstrating its ability to apply its existing knowledge consistently across multiple domains.

The challenge of fine-tuning Large Language Models for specific tasks has received considerable attention in recent years [3, 4]. Studies have demonstrated the effectiveness of fine-tuning in various contexts [1, 5], showing that fine-tuning can significantly improve model performance by allowing it to adapt to the subtleties of a task [6, 7]. However, most of these studies have not focused on ensuring that LLMs follow specific rules or state when these rules are violated. Our research differs from other work by focusing on enforcing the application of rules using a self-generated dataset. The main challenge of this approach is to deal with the potential biases of the LLM, as these could be injected into the self-generated dataset. In fact, training an LLM on a biased or unbalanced dataset may reinforce its existing biases, thus degrade its performance.
This paper provides a preliminary investigation of this approach, and while the results are promising, more in-depth research is needed to confirm these findings.

1.1. Paper organization

The paper is organized as follows: Section 2 provides an overview of the background necessary to understand the approaches and techniques used in this work. Section 3 details the methodology used to create the fine-tuning dataset and describes the overall experimental setup. Section 4 presents the results of our experiments, analyzing the performance of the fine-tuned models in both the target and unseen contexts. In Section 5, we discuss the implications of our findings, highlighting the problems encountered and suggesting directions for future research. Finally, Section 6 concludes the paper with a summary of our main contributions and results.

2. Background

This section provides an overview of the key concepts relevant to this work that are necessary to understand the approaches and methods employed throughout the study.

2.1. Pre-trained Large Language Models

Pre-trained Large Language Models have become fundamental tools in everyday life. These models are trained on very large datasets, allowing them to learn complex patterns, linguistic structures, and a wide range of knowledge. The training of these models involves predicting the next word in a sequence (auto-regressive models) or understanding masked tokens within a sentence (auto-encoding models). This training enables these models to develop a rich representation of language. Despite their versatility and impressive performance, pre-trained LLMs often face challenges when required (and expected) to apply specific domain rules or constraints. This limitation stems mainly from the broad, generalized nature of their training goals, which optimize the ability to handle a wide variety of tasks, but make the specialization, such as rule enforcement, a secondary goal. To address this kind of challenges, researchers have investigated various fine-tuning techniques to adapt these models to specialized domains or tasks, with parameter efficient fine-tuning methods emerging as a promising solution.

2.2. Parameter efficient fine-tuning

The most common technique used to adapt pre-trained LLMs to specific tasks or domains by updating their parameters based on task-specific data is fine-tuning [8]. However, traditional fine-tuning methods require updating all of the parameters of a model, which is computationally expensive and requires a large amount of labeled data. To address these challenges, parameter-efficient fine-tuning (PEFT) methods have been developed. These methods aim to achieve high task performance while modifying only a small subset of the model parameters, thereby reducing the computational cost and the risk of overfitting when data are limited. Parameter-efficient fine-tuning has gained popularity as a way to exploit the capabilities of Large Language Models without the overhead of retraining the entire model. Techniques such as adapter modules [9], prompt tuning [10], and low-rank adaptation (LoRA) [11] are designed to modify a minimal number of parameters. This approach helps to preserve the general knowledge stored in the pre-trained model while allowing task-specific adaptation. PEFT methods have shown that it is possible, in some cases, to achieve performance comparable to full fine-tuning [12] by modifying only a small percentage of a model's parameters. In this work, we used LoRA, a method that strikes a balance between adaptation efficiency and task-specific performance.

2.3. LoRA

LoRA [11] is a parameter-efficient fine-tuning method designed to adapt a pre-trained LLM by the addition of two low rank matrices to the model parameter space. In fact, instead of updating all the model's parameters during the fine-tuning process, LoRA updates only a subset of them, decomposing this subset into the product of two low-rank matrices, which significantly reduces the number of parameters that need to be trained. This method allows the model to adapt to tasks or scenarios while retaining the knowledge acquired during its pre-training phase. The weight update matrix ΔW is represented by LoRA as a product of two low-rank matrices:

$$\Delta W = W_A W_B \tag{1}$$

where $W_A \in \mathbb{R}^{in \times r}$ and $W_B \in \mathbb{R}^{r \times out}$, with $r \ll \min(in, out)$, and where in and out refer to the input and output dimensions of the weight update matrix. The elements of the matrices W_A and W_B are the only parameters that are updated, while the rest of the parameters of the pre-trained model remain frozen.

After training these matrices are combined with the original weight matrix by adding their matrix product:

$$W' = W + \alpha \Delta W = W + \alpha W_A W_B \tag{2}$$

where W is the starting subset of pretrained parameters and α is a coefficient that controls the influence of the training. LoRA has proven effective in scenarios where domain-specific adaptation is required but computational resources are limited [13].

2.4. Data augmentation with LLMs

Data augmentation is a technique that, increasing the training dataset size, allows to improve the performance of all trainable models, from neural networks to LLMs, and is especially useful when the amount of labeled data is limited [14, 15]. In the context of fine-tuning LLMs, we can use the generative capabilities of the LLMs themselves to increase the size of the dataset. By generating synthetic examples or paraphrased sentences, LLMs can expand the diversity and size of the training dataset, allowing the model to learn more robust patterns [16]. In fact, the synthetically generated examples provide a wider range of linguistic variation, contextual nuances, and edge cases that may not be well represented in the original dataset. This increased variety enhances the ability of the model to handle variations in the input, improving its overall capacity.

3. Methodology

This section outlines the methodology used to investigate how effective fine-tuning LLMs can be for applications that require a particular rule application. We detail the problem formulation, the experimental setup, and the evaluation metrics.

3.1. Architecture and experimental design

The goal of this study is to evaluate whether fine-tuning a Large Language Model on self-generated data achieves consistent rule enforcement in the trained contexts and generalize the rule application to unseen scenarios. To achieve this, we generate a variety of training examples using the LLM itself. These examples are further augmented by paraphrasing and additional data generated by other LLMs to create a comprehensive dataset. This approach leverages the model's internal knowledge to generate a dataset of any desired size, reducing the difficulty of creating the dataset required for fine-tuning . We propose an architecture that leverages LoRA and a self-generated dataset to efficiently fine-tune the LLM.

The main components of this architecture, shown in Figure 1, are:

- **Base model (M_base)**: the pre-trained LLM to be fine-tuned;
- **Self-generated dataset (D_gen)**: the synthetic dataset generated by the base model; consists of examples designed to adjust the base model behavior for rule enforcement (e.g. determine whether a given age meets the eligibility requirements for a driving license);
- **Fine-tuned model (M_ft)**: the resulting model after the fine-tuning.

To evaluate the trained model, we simulated a form filling scenario where the age was the discriminator for eligibility. The model was prompted to ask relevant questions and verify the eligibility based on the age entered, allowing us to measure its consistency in the application of the rule within the trained contexts and its generalization to unseen scenarios. The evaluation was performed on a variety of test cases, including:

1. **Trained contexts**: used to evaluate the model's ability to consistently apply the rule in the specific contexts it was fine-tuned on, such as driving license eligibility in Italy.
2. **Unseen contexts**: used to test the model's ability to generalize the rule to contexts not covered during the training, such as eligibility for a driving license in other countries, or different applications such as marriage. This aspect was evaluated from several perspectives:
   a) **Geographical variation**: used to determine whether the model could apply the driving license rule to different countries, not included in the fine-tuning, such as Germany.
   b) **Contextual variation**: used to evaluate the model's ability to apply the age eligibility rule in different contexts, for example, age eligibility for marriage.

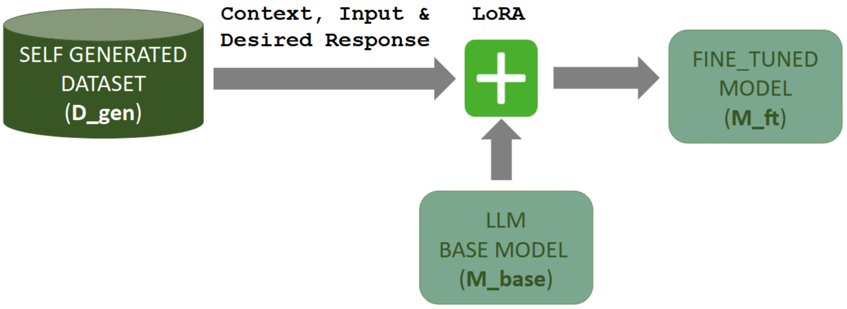


Figure 1: Architecture for the fine-tuning of a LLM that leverages LoRA and a self-generated dataset. The base model (M_base) is fine-tuned through LoRA using the examples present in the self-generated dataset (D_gen).

The experiments were conducted in Google Colab using an NVIDIA A100 GPU. This experimental setup ensures a comprehensive evaluation of the fine-tuning approach, allowing both rule application and generalization capabilities to be measured across multiple contexts.

3.2. Fine-tuning strategy and parameters

To fine-tune the Large Language Models for age-based rule application, we used the Hugging Face Transformers library and the pre-trained models Llama2-Chat 7B [17] and Mistral 7B [18]. We chose the chat version of Llama2 because its conversational style is well-suited for form-filling tasks, making it more effective in these contexts.

3.2.1. Hyperparameters selection

Hyperparameters were selected based on an initial grid search experiment realized to determine the optimal rank of the LoRA matrices (r), the learning rate, the batch size, and the number of training epochs. The best values were found to be: $r = 8$, learning rate = 1×10^{-3}, and the multiplicative coefficient $\alpha = 16$. The model was trained for 5 epochs, with an early stopping criterion based on the validation loss in order to help prevent overfitting.
In this setup, the target modules were left as default. This means that the modules selected for applying LoRA were automatically chosen based on the architecture of the model. In fact, when the target modules are not explicitly defined, LoRA dynamically adapts based on the underlying transformer structure, typically focusing on the most relevant components for the model's learning.

3.3. Dataset

The dataset used for fine-tuning is intentionally kept small and consists of 50 examples. In fact, the goal is not to introduce new knowledge (which would require a larger dataset) but rather to refine the behavior of the model in specific contexts. Each example in the dataset consists of three main components:

1. **Context**: the situational background in which the rule has to be applied. This ranges from specific scenarios to more general ones;
2. **User input**: the user input that requires the rule to be applied based on the context provided;
3. **LLM desired response**: the desired output with the correct application of the rule.

The examples in the dataset cover two scenarios: one where the model must state if the age provided is below the minimum age required for obtaining a driving license and another where it must evaluate the plausibility of an unrealistic input (e.g., age 140). For instance, the dataset includes examples based on the form-filling context for driving license applications, with scenarios that show the LLM when to apply the rule based on the user's input age, such as:

> **Context**: "Please act as form-filling for a car driving license application in the USA. In the conversation you asked the age."
> **User input**: "16"
> **LLM desired response**: "The age you entered, 16, is a valid age."

To increase the diversity of the dataset and mitigate the biases, we employed automated methods to generate additional examples. Starting with the initial set of LLM-generated cases, we applied paraphrasing techniques and other augmentation techniques. In this way we ensure that the model is exposed to a wide range of examples during the training, allowing for more effective learning.

3.3.1. Dataset generation

In this study, the fine-tuning dataset is generated by the LLM itself. Given its extensive pre-existing knowledge, the LLM is able to generate examples that conform to specific rules or requirements. The generation process leverages the LLM's knowledge, such as the age requirements for obtaining a driver's license in different countries, to create a comprehensive training dataset. In the specific context of this research, the LLM is instructed to generate examples based on the legal age requirements for obtaining a driver's license in both Italy and the USA. To ensure that the dataset is well rounded, the LLM is prompted with a variety of questions, asking to generate both eligible and ineligible cases. For example, it generates data elements such as:

- **Eligible cases**: Individuals who meet the minimum age requirement for obtaining a driver's license in either Italy (e.g., a 19-year-old) or in the USA (e.g., a 17-year-old in New York);
- **Ineligible cases**: Individuals who do not meet the minimum age requirement (e.g., a 15-year-old in Italy or a 14-year-old in California);
- **Unrealistically high age**: While there is no official upper age limit for a driver's license, practical factors become relevant for very old ages (e.g., 100+ years). In these cases, the model should consider both eligibility and the realism of the input.

Once the initial set of responses is generated, augmentation techniques are applied to increase its size and diversity creating a richer and more diverse set, thereby reducing sensitivity to biases (e.g., the tendency to respond with "18" as a valid age for drivers). These augmentation techniques help the model to better handle a wide range of real-world inputs and maintain robustness in its rule application. The data generation process consists of three main steps. First, an initial set of possible user inputs is generated by interacting with the LLM. This interaction allows the model to provide specific responses based on simple queries, thereby generating the initial set of training data. Examples of these interactions are:

- **Question**; "Give me a valid age for being eligible for a driving license in the USA. Just reply with the age."
  Answer: "16."

- **Question**: "Can you provide a random age for a person who is not eligible to drive?"
  Answer: "13."

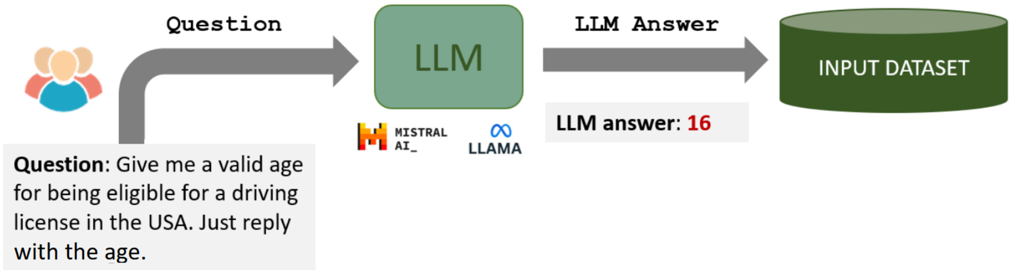

Question
LLM
LLM Answer
INPUT DATASET
MISTRAL AI_
LLAMA
LLM answer: 16
Question: Give me a valid age for being eligible for a driving license in the USA. Just reply with the age.

Figure 2: The process of generating the initial set of inputs using a pre-trained Large Language Model

The second step is the inputs augmentation. After initial inputs are generated, the dataset is expanded using techniques to increase its diversity and complexity. These include:

1. Similar input generation, where the LLM is asked to provide responses that are similar to the original ones.
2. Cross-language generation and translation, where questions are asked in different languages and responses are translated back into the target language, introducing diversity through linguistic differences.
3. Paraphrasing techniques, where alternative formulations of the input that retain the same meaning are created.

Examples of the augmentation process include:

- **Similar input generation**:
  - **Question**: "Can you write a sentence similar to: 'I am 18 years old'?"
  - **Answer**: "I'm 21 years old."
- **Cross-language generation**:
  - **Question**: "Mi potresti scrivere una frase simile a: 'ho 23 anni'?"

  - **Answer**: "Ho 25 anni."
  - **Translation**: "I'm 25 years old."
- **Paraphrasing**:
  - **Original**: "I am 21 years old."
  - **Paraphrases**: "I've reached the age of 21.", "I turned 21 this year."

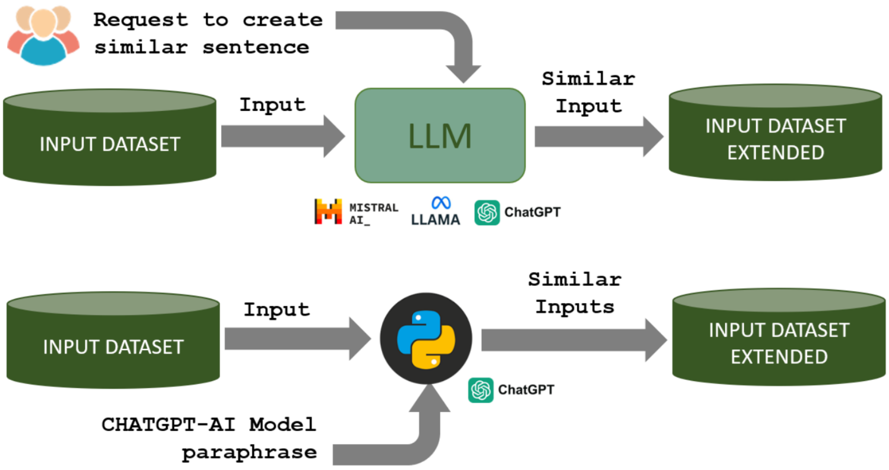
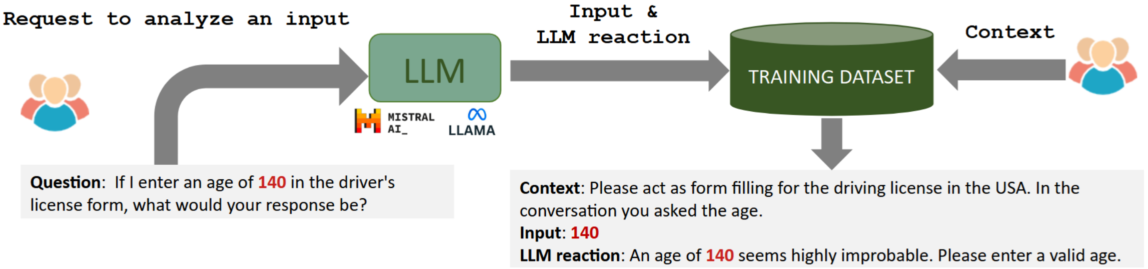


Figure 3: The augmentation process used to expand the input dataset, incorporates techniques such as similar answer generation, cross-language translation, and paraphrasing to increase diversity and complexity.

Finally, the third step is the generation of the desired LLM responses. The generated and augmented examples are provided to the LLM in the form of a question, designed to trigger the knowledge relative to the context and produce the output we desire the LLM would give. The response is then added to the dataset as the desired LLM response. Examples of response generation include:

- **Question**: "If I enter an age of 140 in the driver's license form, what would your response be?"
  Answer: "An age of 140 seems highly improbable. Please enter a valid age."
- **Question**: "Can a 6 year old apply for a driving license?"
  Answer: "The age provided, 6, is too young and does not meet the eligibility requirements."
- **Question**: "I'm 16 years old and would like to apply for a driving license in Italy. Is this age acceptable?"
  Answer: "In Italy, the minimum age requirement for a driving license is 18."



Question: If I enter an age of 140 in the driver's license form, what would your response be?

Context: Please act as form filling for the driving license in the USA. In the conversation you asked the age.
Input: 140
LLM reaction: An age of 140 seems highly improbable. Please enter a valid age.

Figure 4: The process of providing generated and augmented examples as input to the LLM, which then produces an answer. Context, input, and LLM reaction triplets are combined to form an example of the training dataset.

By combining these steps, we were able to create a comprehensive and varied set of training examples to support the fine-tuning of the LLM, ensuring improved rule application in the given context. Examples of the final dataset elements items include:

- **Context**: "Please act as form-filling for the driving license. In the conversation, you asked the age."
  User input: "140"
  LLM reaction: "Age 140 seems highly unlikely. Please re-enter a valid age."
- **Context**: "Please act as form-filling for the driving license. In the conversation, you asked the age."
  User input: "6 years"
  LLM reaction: "Age 6 is not valid for this form. Please enter a realistic age."
- **Context**: "Please act as if you are filling out a form for a driving license in Italy. In the conversation, you inquired about the person's age."
  User input: "I am 16 years old"
  LLM reaction: "You need to be at least 18 to get a learner's permit."

In summary, the dataset generation method has been designed to consider the address of potential biases present in the data generation. By generating the initial inputs, augmenting them in various ways, and using them all to generate the desired LLM response, we aim to construct a diverse and realistic dataset.

3.4. Performance evaluation

The evaluation of the fine-tuned LLM's performance is crucial for understanding its effectiveness in applying domain-specific rules and its ability to generalize across different contexts. In this study, human feedback is used to evaluate the performance of both M_base and M_ft by checking whether they correctly applied the rule in their responses, testing both training and unseen contexts. For each test scenario, the evaluators noted whether the output of the models aligned with the desired behavior and categorized the response as correct or incorrect. This classification allows us to calculate the average accuracy of the model as follows:

$$Accuracy = \frac{\left(\sum_{j=1}^{N} R_j\right)}{N} \times 100\% \tag{3}$$

where R_j is the response in the j-th trial (1 if correct, 0 if incorrect), and N is the total number of trials. To assess catastrophic forgetting, which occurs when the model loses general language abilities or knowledge in domains unrelated to the fine-tuning task [19], we evaluated its performance on tasks outside the fine-tuning domain, such as reading comprehension or translation. Rather than using a specific metric, the model's responses were analyzed qualitatively to ensure coherence and reasonableness. Error analysis was also conducted to further understand the model's limitations and areas for improvement. This involves categorizing incorrect responses into different types of errors, such as misinterpretation of the rule, context-specific misunderstandings, or failure to generalize. Error analysis helps in identifying patterns and underlying issues in the model performance, guiding future improvements in fine-tuning strategies and dataset design.

4. Results

Our study evaluates the ability of the fine-tuned models to apply a rule in both trained scenarios and unseen contexts, averaging performance over 10 trials. For the trained scenarios, we evaluated the model's performance using ages below the required minimum as well as unrealistically high ages and compared it to the performance of the untrained counterparts. The comparison shows that the fine-tuned models outperform the untrained models in applying the rule; however, it is important to note that the errors made by the fine-tuned model were consistent with those made by the untrained model. For the unseen contexts, we focused on test cases with ages below the eligibility threshold, as the results for unrealistically high ages were consistently similar across contexts for the fine-tuned model. Detailed output examples are provided in Appendix A.

In addition to assessing the performance of the model in rule-specific contexts, we compared its performance on several unrelated tasks before and after the fine-tuning. Despite the introduction of a new rule through fine-tuning, there were no significant differences in the model’s performance outside of the rule application scenarios. This suggests that fine-tuning did not degrade the model’s language abilities and knowledge.

4.1. Evaluation in training contexts

We evaluated the two models, Llama2-Chat 7B and Mistral 7B, in both their trained and untrained states in different contexts to assess their effectiveness in applying age-based rules. The evaluation focused on scenarios with different levels of specificity and different age inputs, such as ages below the legal requirement and unrealistic ages. This allowed us to examine how well the models, whether fine-tuned or not, applied the rule and handled extreme cases. The results presented in Table 1 indicate that both fine-tuned models demonstrated satisfactory performance when handling underage inputs, consistently indicating ineligibility as desired. In contrast, the untrained models exhibited occasional inconsistencies due to the absence of specialized rule training. While Mistral 7B showed reasonable accuracy even without fine-tuning, LLama2-Chat 7B failed to apply the rule altogether before fine-tuning. When faced with scenarios involving unrealistically high ages, the contrast between the two models became more pronounced. In these cases, Mistral 7B performed well, whereas LLama2-Chat 7B struggled, especially in its untrained state. Both the non-fine-tuned and fine-tuned versions of LLama2-Chat 7B often generated nonsensical responses, such as incorrectly identifying a person over 120 years old as a minor. Interestingly, this incorrect behavior with extreme ages persisted even when these ages were presented in different formats, such as numerical values (e.g., "140") or written out in words (e.g., "one hundred and forty"). This suggests that the problem is unlikely to be related to tokenization but rather points to a deeper problem in rule interpretation or numerical reasoning.

Table 1
This table shows the average accuracy over 10 runs for both untrained and trained models. Accuracy assesses whether the models have applied the rule correctly. The table also shows typical errors in the output of the models.

Scenario	Model	Accuracy untrained model	Accuracy trained model	Typical errors
Ineligible Age	LLama2-Chat 7B	0%	100%	Doesn’t apply the rule
	Mistral 7B	40%	100%	Doesn’t apply the rule
Unrealistic age	LLama2-Chat 7B	0%	60%	Nonsensical responses, e.g., age 120 seen as underage
	Mistral 7B	100%	100%	-

4.2. Evaluation in unseen scenarios

We evaluated the fine-tuned models in unseen scenarios, not present in the traing data, to assess their generalization capabilities. These scenarios spanned various regions and application contexts to test the models’ ability to apply the rule across diverse, unfamiliar settings:

- **Different states**: Tested the generalization of the rule across states not included in the training set.
- **Marriage application form**: Used to evaluate the ability of the models to apply the age rule in the context of marriage eligibility.
- **Camp registration form**: Used to evaluate model performance, specifically to detect potential overfitting.
- **Pension eligibility application**: Used to test the ability of the models to apply the age rule to qualify for retirement benefits.

Table 2
This table shows the average accuracy over 10 runs for both untrained and trained models, showing whether the rule is correctly applied in unseen contexts (with input ages below the minimum age threshold). It also highlights typical errors in the output of the models.

Scenario	Model	Accuracy untrained model	Accuracy trained model	Typical errors
Different states	LLama2-Chat 7B	0%	100%	Doesn't apply the rule
	Mistral 7B	30%	100%	Doesn't apply the rule
Marriage application	LLama2-Chat 7B	0%	70%	Doesn't apply the rule
	Mistral 7B	40%	80%	Doesn't apply the rule
Camp registration	LLama2-Chat 7B	100%	100%	Doesn't apply the rule
	Mistral 7B	100%	100%	Doesn't apply the rule
Pension eligibility	LLama2-Chat 7B	0%	40%	Doesn't apply the rule
	Mistral 7B	30%	60%	Doesn't apply the rule

The results of these tests, reported in Table 2, indicate that both fine-tuned models generally performed well in most of the unseen scenarios, consistently outperforming their untrained counterparts in applying the age-based rule. In the most complex scenario, i.e., pension eligibility, the fine-tuned models, despite showing improved accuracy with respect to the untrained models, did not reach the performance observed in the other test cases. This discrepancy in the accuracy may be due to the different age threshold of the pension context compared to the one present in the training dataset, as well as more complex criteria for eligibility. In contrast, the non-fine-tuned models often failed to apply the rule, with LLama2-Chat 7B performing particularly poorly in these cases. Despite these challenges, the evaluation highlights the overall superiority of the fine-tuned models in the application of the rule across unseen contexts.

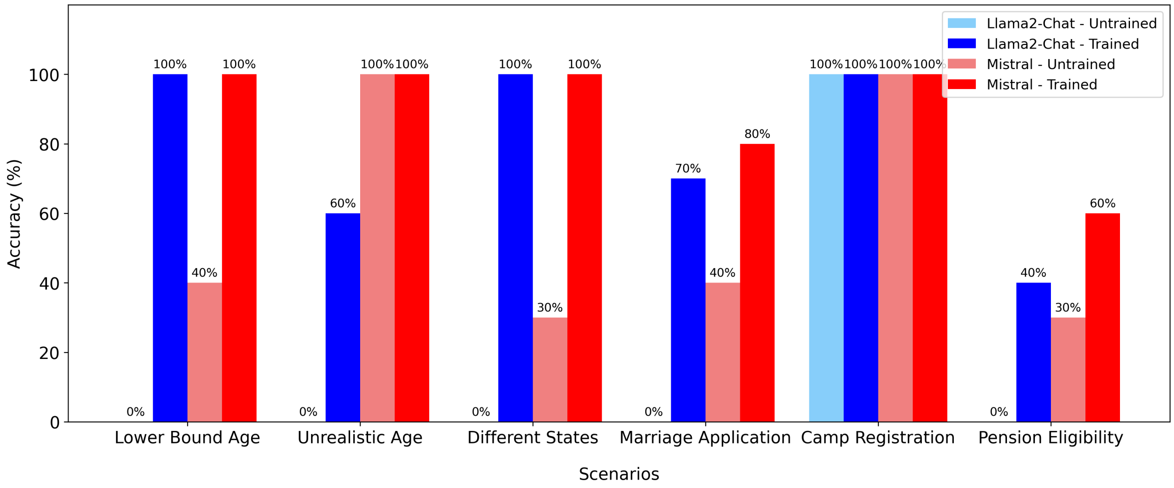


Figure 5: Graphical comparison of the accuracy of untrained and trained models across the tested scenarios.

5. Discussion and Outlook

Despite the promising results of the proposed approach for rule enforcement, some issues or unexpected behaviors emerged in this study. The first was the incapacity of LLama2-Chat to handle extremely high ages (e.g., 140). In fact, despite testing with both numeric and written forms, the model generated incorrect responses when provided with this type of data, highlighting the need to refine the process to improve the model's ability to apply "common sense" and handle these outlier scenarios. A valid alternative might be to set a predetermined upper age limit (e.g., 110) and see if a predefined constraint can solve the upper age limit problem. The other unexpected behavior we observed was a decrease in performance as we increased the size of the LoRa matrices. This means that as the number of trained parameters increases, the performance of the model decreases, which is paradoxical and highlights the need to explore the underlying reasons for this behavior.

The ability to train the model to apply the rule to more than just the trained context opens up the use of LLMs in many applications, with the advantage of using a limited dataset size. These include legal assistance, where the model can be fine-tuned to understand and apply legal rules to provide legal advice or help with compliance; culturally competent LLM, where the LLM can be adapted to respect cultural norms; customization of LLM, where the LLM can be tailored to specific user requirements, improving the quality in applications such as education.

Looking ahead, there are many areas that require additional investigation. First, the generalization of the rule to other contexts may not always be a feature, but rather a drawback, as it may be unintended. Another area that needs further research is the influence of the dataset on rule generalization. Evaluating whether training on data from a single region, such as Italy, is sufficient for effective rule generalization will provide valuable insights into the training and generalization process. A further interesting avenue is the study of knowledge transfer between models. Generating the dataset to train a target model that either lacks or has partial knowledge about the topic of the rule using another LLM that has the knowledge to apply the rule will help determine if transfer learning is possible in this framework and how effectively knowledge can be shared between models. Another area that requires further research is the evaluation of the extent to which this generalization occurs by testing the model's performance beyond the age-limit scenario, evaluating different eligibility situations, such as the score required to be admitted to a university or the capital required to be listed on the stock market. Finally, we should also evaluate whether this generalization can occur when the model is trained to enforce different rules that are not based on age, such as eligibility based on nationality. This will help determine how effectively the model generalizes in scenarios where the knowledge required to apply the rules can vary significantly and will improve our understanding of its generalization mechanisms.

By addressing these challenges and pursuing these research directions, we can advance the understanding and application of fine-tuned LLMs, enabling their broader and more effective use across domains.

6. Conclusions

In this study, we fine-tuned two LLMs for consistent application of a domain-specific age rule across different contexts. Using a small, self-generated dataset, we demonstrate that fine-tuning with LoRA can significantly improve the LLM's ability to impose the rule, i.e., age requirement, and to generalize this behavior to unseen scenarios. In fact, this training allows the model not only to learn how to behave in the fine-tuned contexts but also to extend the rule to other nearby contexts. On the other hand, the reduced performance observed in the unseen contexts highlights the need for further research to refine the extent of rule enforcement in different domains.

Our results provide useful insights into the adaptability of LLMs when fine-tuned for rule-based applications. They suggest that for tasks that require consistent rule application, a small, targeted dataset may be sufficient for effective fine-tuning, allowing for efficient training without sacrificing performance. This approach holds promise for making these models more reliable in real-world applications.

Acknowledgments

This work was funded by the National Recovery and Resilience Plan (PNRR), under Mission 4 "Education and Research"-Component 2, Investment 1.1 "Fund for the National Research Program, Projects of Relevant National Interest (PRIN)". Call: PRIN 2022 (D.D. 104/22), project title: ENGineering INtElligent Systems around intelligent agent technologies, CUP: E53D23007970006. The project was carried out at the Department of Information Engineering and Computer Science (DISI) of the University of Trento. The authors would like to thank the Italian Ministry of University and Research (MUR) for supporting this research through the PRIN 2022 funding program.

A. Untrained and Trained LLM: Dialog Examples

This section presents examples of conversations generated by untrained and trained LLMs, illustrating their performance before and after fine-tuning. These examples illustrate the models' enhanced ability to apply the rule in specific contexts following fine-tuning.

A.1. Interaction examples for car driving license form

Car driving license application context:
Please act as the form filler for the **car driving license**. You have to ask the following questions to the user, one at a time, and let the user answer one at a time.

- First question: Country of residence
- Second question: Name and surname
- Third question: Age
- Fourth question: Date of birth

Dialog example 1

For a user from Italy or Germany:

> **LLM**: "I'm happy to help you with your driving license application! Can you please tell me what country you reside in?"
> **User answer**: **Italy/Germany**
> **LLM**: "Great, thank you for letting me know! What is your name and surname?"
> **User answer**: "Mario Rossi"
> **LLM**: "Thank you, Mario Rossi. How old are you?"
> **User answer**: "**16**"

Untrained LLM:	**Trained LLM**:
"Perfect, thank you for letting me know. What's your date of birth?"	"Thank you. However, you are not yet eligible for a driver's license in **Italy/Germany**, as the minimum age is **18**."

Dialog Example 2

For a user from the USA:

> **LLM**: "I'm happy to help you with your driving license application! Can you please tell me what country you reside in?"
> **User answer**: **USA**
> **LLM**: "Great, thank you! What is your name and surname?"
> **User answer**: "John Doe"
> **LLM**: "Thank you, John Doe. How old are you?"
> **User answer**: "**140**"

Untrained LLM:	**Trained LLM**:
"Thank you for letting me know. However, you are not eligible for a driving license in the **USA** as the minimum age requirement is **16**."	"The age you entered, **140**, is not realistic. Please provide a valid age."

Dialog Example 3

For a user from Alberta, Canada:

> **LLM**: "I'm happy to help you with your driving license application! Can you please tell me what country you reside in?"
> **User answer**: **Canada, Alberta**
> **LLM**: "Thank you! What is your name and surname?"
> **User answer**: "Jane Doe"
> **LLM**: "Thank you, Jane Doe. How old are you?"
> **User answer**: "**15**"

Untrained LLM:	**Trained LLM**:
"Thank you for letting me know. Can you please provide your date of birth?"	"In **Alberta**, you must be at least **16** to get a driver's license. If you're under 18, you need proof of enrollment in a driver's education course. Do you have any prior driving experience?"

A.2. Interaction examples for other forms

Marriage application form context:
Please act as a filler of the **marriage application form**. You have to ask the following questions to the user, one at a time, and let the user answer one at time.

- First question: Country of residence
- Second question: Name and surname of the groom
- Third question: Age of the groom
- Fourth question: Name and surname of the bride
- Fifth question: Age of the bride

Dialog example

> **LLM**: "Can you please tell me the country of residence for the couple?"
> **User answer**: **Italy**
> **LLM**: "Great! What is the name and surname of the groom?"
> **User answer**: "Mario Rossi"
> **LLM**: "Thank you. How old is the groom?"
> **User answer**: "**12**"

Untrained LLM:	**Trained LLM**:
"Thank you! What is the name and surname of the bride?"	The groom is **12**, which is below the legal marriage age of **18**. Please provide a valid age for the groom."

Camp Registration Form Context:
Please serve as a **camp registration form** filler. You need to ask the following questions to the user, one at a time, and let the user answer them one at a time.

- First question: Country of residence
- Second question: Name and surname
- Third question: Age

Dialog example

LLM: "Thank you for registering for the camp! Which country do you live in?"
User answer: **Italy**
LLM: "Thank you! What is your name and surname?"
User answer: "Mario Rossi"
LLM: "Great! We offer activities for all ages, so please tell me your age."
User answer: "**12**"

Untrained LLM:	**Trained LLM**:
"Thank you!"	"Thank you!"

Retirement application form context:
Please serve as a **retirement application form** filler. You need to ask the following questions to the user, one at a time, and let the user answer them one at a time.

- First question: Country of residence
- Second question: Name and surname
- Third question: Age
- Fourth question: Number of years worked

Dialog example

LLM: "Thank you for your interest in retirement benefits. Could you please provide your country of residence?"
User answer: **Italy**
LLM: "Thank you! Could you please provide your name and surname?"
User answer: "Mario Rossi"
LLM: "Thank you, Mario Rossi. How old are you?"
User answer: "**23**"

Untrained LLM:	**Trained LLM**:
"Ok, thank you! How many years have you worked?"	"I apologize, but you are not eligible for retirement in Italy as you are not yet 60 years old, which is the minimum age required for Italian retirement benefits. However, I can still help you with the application process and answer any questions you may have."

References

[1] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, D. Amodei, Language models are few-shot learners (2020). doi:`10.48550/arXiv.2005.14165`.

[2] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: North American Chapter of the Association for Computational Linguistics, 2019. URL: https://api.semanticscholar.org/CorpusID:52967399.

[3] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, G. Irving, Fine-tuning language models from human preferences, ArXiv abs/1909.08593 (2019). URL: https://api.semanticscholar.org/CorpusID:202660943.

[4] C. Jeong, Fine-tuning and utilization methods of domain-specific llms (2024). doi:`10.48550/arXiv.2401.02981`.

[5] H. Liu, D. Tam, M. Mohammed, J. Mohta, T. Huang, M. Bansal, C. Raffel, Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning, in: A. H. Oh, A. Agarwal, D. Belgrave, K. Cho (Eds.), Advances in Neural Information Processing Systems, 2022. URL: https://openreview.net/forum?id=rBCvMG-JsPd.

[6] C. Sun, X. Qiu, Y. Xu, X. Huang, How to fine-tune bert for text classification?, in: M. Sun, X. Huang, H. Ji, Z. Liu, Y. Liu (Eds.), Chinese Computational Linguistics, Springer International Publishing, Cham, 2019, pp. 194–206.

[7] H. Yang, Y. Zhang, J. Xu, H. Lu, P.-A. Heng, W. Lam, Unveiling the generalization power of fine-tuned large language models, in: K. Duh, H. Gomez, S. Bethard (Eds.), Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 884–899. doi:`10.18653/v1/2024.naacl-long.51`.

[8] V. B. Parthasarathy, A. Zafar, A. Khan, A. Shahid, The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities, 2024. `arXiv:2408.13296`.

[9] Z. Hu, L. Wang, Y. Lan, W. Xu, E.-P. Lim, L. Bing, X. Xu, S. Poria, R. Lee, LLM-adapters: An adapter family for parameter-efficient fine-tuning of large language models, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 5254–5276. doi:`10.18653/v1/2023.emnlp-main.319`.

[10] B. Lester, R. Al-Rfou, N. Constant, The power of scale for parameter-efficient prompt tuning, in: M.-F. Moens, X. Huang, L. Specia, S. W.-t. Yih (Eds.), Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 3045–3059. URL: https://aclanthology.org/2021.emnlp-main.243.

[11] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-rank adaptation of large language models, in: International Conference on Learning Representations, 2022. URL: https://openreview.net/forum?id=nZeVKeeFYf9.

[12] L. Xu, H. Xie, S.-Z. J. Qin, X. Tao, F. L. Wang, Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment, arXiv preprint (2023). URL: https://arxiv.org/abs/2312.12148.

[13] C. Xin, Y. Lu, H. Lin, S. Zhou, H. Zhu, W. Wang, Z. Liu, X. Han, L. Sun, Beyond full fine-tuning: Harnessing the power of LoRA for multi-task instruction tuning, in: N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, N. Xue (Eds.), Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, Torino, Italia, 2024, pp. 2307–2317. URL: https://aclanthology.org/2024.lrec-main.206.

[14] L. Perez, J. Wang, The effectiveness of data augmentation in image classification using deep learning, 2017. URL: https://arxiv.org/abs/1712.04621. `arXiv:1712.04621`.

[15] S. Feng, V. Gangal, J. Wei, et al., A survey of data augmentation approaches for nlp, ACM Computing Surveys (2021).

[16] B. Ding, C. Qin, R. Zhao, T. Luo, X. Li, G. Chen, W. Xia, J. Hu, A. T. Luu, S. Joty, Data augmentation using LLMs: Data perspectives, learning paradigms and challenges, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Findings of the Association for Computational Linguistics ACL 2024, Association for Computational Linguistics, Bangkok, Thailand and virtual meeting, 2024, pp. 1679–1705. URL: https://aclanthology.org/2024.findings-acl.97.

[17] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., Llama 2: Open foundation and fine-tuned chat models, arXiv preprint (2023). URL: https://arxiv.org/abs/2307.09288.

[18] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de Las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, W. E. Sayed, Mistral 7b, ArXiv abs/2310.06825 (2023). URL: https://api.semanticscholar.org/CorpusID:263830494.

[19] Y. Luo, Z. Yang, F. Meng, Y. Li, J. Zhou, Y. Zhang, An empirical study of catastrophic forgetting in large language models during continual fine-tuning, ArXiv abs/2308.08747 (2023). URL: https://api.semanticscholar.org/CorpusID:261031244.