Annals of Actuarial Science (2026), **20**, pp. 74–95
doi:10.1017/S1748499525100079

ORIGINAL RESEARCH PAPER

Institute and Faculty of Actuaries

Utilizing large language models (LLMs) for quantitative reasoning-intensive tasks within the (re)insurance sector

Yilin Hao[1], Xiaojuan Tian[1], Haoran Zhao[1] and Luca Baldassarre[2]

[1]Swiss Re, Beijing, China; and [2]Swiss Re, Zurich, Switzerland
Corresponding author: Yilin Hao; Email: hao_yilin@163.com

(Received 08 July 2024; revised 17 March 2025; accepted 07 July 2025; first published online 12 August 2025)

Abstract

The rise of large language models (LLMs) has marked a substantial leap toward artificial general intelligence. However, the utilization of LLMs in (re)insurance sector remains a challenging problem because of the gap between general capabilities and domain-specific requirements. Two prevalent methods for domain specialization of LLMs involve prompt engineering and fine-tuning. In this study, we aim to evaluate the efficacy of LLMs, enhanced with prompt engineering and fine-tuning techniques, on quantitative reasoning tasks within the (re)insurance domain. It is found that (1) compared to prompt engineering, fine-tuning with task-specific calculation dataset provides a remarkable leap in performance, even exceeding the performance of larger pre-trained LLMs; (2) when acquired task-specific calculation data are limited, supplementing LLMs with domain-specific knowledge dataset is an effective alternative; and (3) enhanced reasoning capabilities should be the primary focus for LLMs when tackling quantitative tasks, surpassing mere computational skills. Moreover, the fine-tuned models demonstrate a consistent aptitude for common-sense reasoning and factual knowledge, as evidenced by their performance on public benchmarks. Overall, this study demonstrates the potential of LLMs to be utilized as powerful tools to serve as AI assistants and solve quantitative reasoning tasks in (re)insurance sector.

Keywords: fine-tuning; large language models; quantitative reasoning; reinsurance

1. Introduction

In insurance and reinsurance business, a large amount of unstructured data is generated and needs processing and analyzing. In contrast to structured data, which are typically represented in tabular form, unstructured data encompass information that lacks a pre-defined format, such as text and images. Common unstructured data in (re)insurance business include insurance policies, claim reports, and emails. The processing of unstructured data comprises a broad variety of tasks, ranging from fundamental operations to sophisticated analysis. At the fundamental level, this includes entity extraction (e.g., policy details from documents) and classification (e.g., claim reason classification into a list of given options). More advanced processes involve synthesizing information into summaries and applying logical reasoning, such as evaluating the appropriateness of claim payments based on policy terms. The traditional approach to such unstructured data has been to deal with it by hand, which is both inefficient and time-consuming. The development of natural language processing (NLP) provides an automated solution for unstructured data processing. The evolution of language models originates from the statistical models in the 1990s (Jelinek, 1990; Waibel & Kai-Fu, 1990; Jelinek et al., 1992; Brown et al., 1990; Brown et al., 1993; Brown, 1990; Jelinek, 1998; Rosenfeld, 2000; Stolcke, 2002; Gao & Lin, 2004), which are developed under the Markov assumption that the probabilistic inference of the $(k+1)$st word is dependent on

© The Author(s), 2025. Published by Cambridge University Press on behalf of Institute and Faculty of Actuaries. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

https://doi.org/10.1017/S1748499525100079 Published online by Cambridge University Press

its preceding k words. Subsequently, neural-network-based language models were introduced to handle sequential data effectively, including recurrent neural networks and its variants, e.g., gated recurrent unit and long-short term memory (Rumelhart et al., 1986; Chung et al., 2014; Hochreiter & Schmidhuber, 1997). The models mentioned above are relatively small language models, which perform well in specific tasks such as extraction and classification, but exhibit restricted capabilities in reasoning. The integration of these models into the (re)insurance sector could facilitate the automation of business processes, such as claim classification and fraud detection, thereby enhancing operational efficiency (Lee et al., 2020; Saddi et al., 2023).

The transformer architecture led a breakthrough enabling more scalable training and better model performance through self-attention mechanism (Vaswani et al., 2017). Based on this architecture, the BERT (Bidirectional Encoder Representations from Transformers) model was released by pre-training on large-scale unlabeled text data (Devlin et al., 2019). Subsequently, the rise of ChatGPT (OpenAI, 2022) and other large language models (LLMs) has significantly changed the paradigm of NLP, marking a substantial leap toward achieving artificial general intelligence. With the increase of model and data sizes of pre-trained language models, some emergent abilities have been found, leading to the term LLM (Wei et al., 2022). These emergent abilities include in-context learning, instruction following and step-by-step reasoning, making it possible for language models to solve more complicated problems. For example, GPT-3 demonstrated few-shot learning capabilities, unlike its predecessor GPT-2 (Brown et al., 2020). These pre-trained LLMs are considered storing a wide range of knowledge and common sense with advanced reasoning capabilities. Recently, ChatGPT resulted from aligning the LLMs from the GPT series for dialogue, which presents an exceptional conversational ability with humans (OpenAI, 2022). In addition to enhanced performance on conventional NLP tasks such as extraction and classification, the emerging capabilities of LLMs demonstrate the potential of AI assistants for employees in the (re)insurance sector (Balona, 2023). As AI assistants, LLMs could be used to improve efficiency, streamline operations, and support decision-making in a variety of ways. For example, they can analyze and summarize long documents, highlighting key information and saving significant reading time. Another potential scenario is preliminary assessments in claims processing, where LLM can compile relevant policy and claim reports to suggest an initial decision to accept or reject a claim, subject to human review.

However, challenges remain in the use of LLMs in (re)insurance sector. While LLMs are designed to have broad knowledge, they may struggle with domain-specific tasks due to the overrepresentation of popular topics and underrepresentation of niche subjects. This can lead to inconsistent or inaccurate responses when dealing with complex domain-specific concepts and terminology. The domain specialization of LLMs is a critical yet demanding problem. To narrow the divide between general LLMs and domain-specific requirements, there are two prevalent methods, namely prompt engineering and fine-tuning, which involve the integration of proprietary and domain-specific data into LLMs. The two methods offer varying levels of computational efficiency, ease of deployment, and adaptability. Prompt engineering is resource-efficient, yet its capacity to enhance the performance of LLMs in domain-specific tasks is limited. It could leverage the inherent language capabilities of LLMs without altering the model parameters, making them ideal for tasks such as information extraction and text generation (Dodson, 2023). In contrast, it has been shown that fine-tuning can significantly improve the inference of LLMs, especially for tasks demanding complex reasoning (Singhal et al., 2025; Chung et al., 2024). While there have been reports of LLMs being specialized in sectors such as medicine, finance, investment, and law through fine-tuning techniques (Singhal et al., 2025; Yang et al., 2023; Yang et al., 2023; Cui et al., 2023), the (re)insurance domain has not yet been explored in this context. Additionally, many tasks within the (re)insurance domain rely on the ability to perform complex quantitative reasoning, such as the calculation of premiums and loss payments. However, many fine-tuned models have been applied to text-heavy tasks, indicating a research gap for those that require quantitative reasoning performed frequently in the (re)insurance workflow.

https://doi.org/10.1017/S1748499525100079 Published online by Cambridge University Press

This study assesses the capability of LLMs to perform reasoning tasks within the (re)insurance sector. We utilize reinsurance training materials to create quantitative question-answer pairs that test the models' ability to calculate and allocate liabilities, premiums, and claims between the reinsured and the reinsurer. While these test cases may not fully replicate real-world business scenarios, they serve as a valuable benchmark to evaluate LLMs' potential for executing reasoning tasks in the reinsurance domain. The open-source Llama 2-Chat models of sizes 7B, 13B, and 70B are adopted as our baseline with prompt engineering based on one-shot learning and fine-tuning with various datasets applied subsequently to further augment the model performance. As a benchmark for comparison, we also evaluated the performance of GPT 4, a close-source model released around the same time as Llama 2. Our findings can be summarized as follows:

- By using one-shot learning, the Llama 2-Chat 70B model demonstrates notable performance improvement, while the 7B and 13B models encounter difficulties. Fine-tuning with the calculation dataset increases evaluation metric from around 15 points to nearly 80 points, yielding a five-to-six-fold enhancement and outperforming larger pre-trained LLMs, which underscores the importance and efficacy of fine-tuning for reasoning-intensive tasks.
- When task-specific calculation dataset is limited, supplementing LLMs with more accessible domain-specific knowledge data can markedly improve performance, achieving results comparable to models fine-tuned on extensive calculation datasets. Even without calculation data, employing background knowledge for fine-tuning with one-shot prompting can significantly elevate model capabilities.
- There is a positive correlation between the model performance and its capacity for reasoning, with deductions due to computational errors comprising a minimal fraction of the overall performance metrics. Consequently, enhancing reasoning capability stands as the principal focus for optimizing LLMs in quantitative tasks. Furthermore, the fine-tuned model maintains its proficiency in common sense reasoning and factual knowledge, suggesting retention of acquired knowledge post-fine-tuning, according to their performance on public benchmarks.

The paper is organized as follows. Section 2 offers an overview of related work for reader reference. Section 3 details our experiment set-up, including evaluation data, training data, prompt engineering, and fine-tuning techniques employed in this work, and evaluation metric. In Section 4, we present our findings from experiments and auxiliary assessments, including error analysis and the model evaluation on public benchmarks. The concluding remarks can be found in Section 5.

2. Related works

In this section, we present prior work related to our study. Section 2.1 provides an overview of existing literature on the application of NLP techniques in (re)insurance problems, covering both traditional language models and LLMs. Section 2.2 delves into an introduction of open-source LLMs and domain-specification techniques, including prompt engineering and fine-tuning, as well as examples of domain-specification of LLMs.

2.1 NLP techniques in (re)insurance sector

2.1.1 Traditional language models

Prior to the appearance of LLMs, traditional NLP techniques, such as statistical models and neural-network-based models, have offered effective tools for processing unstructured data. By extracting information from raw text data and transforming it into structured numerical or categorical data, these language models could generate additional variables for further analysis. For example, the utilization of rule-based NLP algorithms or sentence embeddings could help with the

https://doi.org/10.1017/S1748499525100079 Published online by Cambridge University Press

recognition of fraudulent patterns from the contents of claims, thereby facilitating the detection of insurance fraud through machine learning techniques (Saddi et al., 2023). Word embedding models could be used for claim classification, and new features could be generated to provide additional information for claim analysis and loss amount prediction through actuarial models such as generalized linear model (Lee et al., 2020). The deployment of these NLP models can reduce the time and human effort required for manual review by automating the processing of unstructured data, thereby enhancing overall efficiency. Furthermore, NLP techniques offer the potential to utilize a broader range of data source for modelling purposes beyond structured data. This could facilitate the generation of insights and enhance the performance of (re)insurance models.

2.1.2 Large language models

With the extensive knowledge foundation and emerging reasoning capabilities, LLMs present the potential of the application in the (re)insurance sector through two ways: directly contributing to modelling and serving as workflow assistant (Balona, 2023). The first application involves solving the same tasks as the traditional language models. For example, ChatGPT can be used to extract information from accident reports, achieving a higher accuracy than traditional small language models (Troxler & Schelldorfer, 2024). The utilization of LLMs for NLP tasks shows several advantages. Firstly, LLMs are trained on a large corpus with a deeper understanding of fundamental and expert knowledge, which enables them to perform better than traditional language models. Secondly, the pre-processing of raw data can be simplified. For example, there is no need to translate the input text into the same language when dealing with multilingual tasks. Thirdly, some less complex tasks can be solved by LLMs in an unsupervised approach, saving the effort required for data annotation and model training, which are necessary in machine learning. The second application involves assisting in the daily work of employees in the (re)insurance sector by providing documentation summarization and generation, automatic data analysis, coding assistance, and other problem-solving services (Balona, 2023). The potential of incorporating LLMs into routine work is guaranteed by their emerging capabilities. However, this application has not been sufficiently explored in previous literature and is likely to become a valuable area of interest in future research.

2.2 LLM and domain-specialization

2.2.1 Open-source LLM and Llama2

Developing an LLM from scratch is a resource-intensive endeavor. Therefore, it is common practice to use existing publicly available models, which can be divided into two categories, i.e., close-source and open-source LLMs (Zhao et al., 2023). Compared to the close-source LLMs, such as ChatGPT by OpenAI, offering access through APIs or user interfaces without the need for local deployment, open-source LLMs provide downloadable model checkpoints, enabling local deployment, further training, or fine-tuning. These models ensure great transparency, permit full control, and are often released with different options of model sizes, allowing for flexible customization.

Among the open-source LLMs, Llama, a suite of models, has been broadly used in academic research and commercial applications, demonstrating robust performance across numerous benchmarks (Beeching et al., 2023). Following an initial release of Llama from Meta AI in February 2023, Llama 2 and Llama 2-Chat were introduced in July 2023 with further training on new public datasets and various model sizes of 7B, 13B, and 70B parameters. Compared with the pre-trained Llama 2, the Llama 2-Chat models were optimized for following instructions and dialogue use cases (Touvron et al., 2023). Subsequent developments have built upon these models, e.g., Llama 2-Instruct was tailored for long-text chat, and Llava was adapted for multimodal instruction-following tasks (Together, 2023; Liu, 2023).

https://doi.org/10.1017/S1748499525100079 Published online by Cambridge University Press

2.2.2 Prompt engineering

Despite the capabilities of LLMs, a persistent gap exists between their general knowledges and domain-specific tasks. Prompt engineering is a relatively new discipline for developing and optimizing prompts to effectively use LLMs without altering the model parameters. There have been reports indicating that LLMs show improved capacity on downstream tasks when properly prompted (Radford et al., 2019; Wei et al., 2022).

Among the approaches in prompt engineering, few-shot prompting enables in-context learning to steer the models by providing demonstration examples (Brown et al., 2020). It has been evident that the LLMs can acquire proficiency for common tasks through the provision of one single example, referred to as one-shot prompting. Besides, various techniques have emerged for reasoning-intensive tasks (Saravia, 2022). For example, Chain-of-Thought (CoT) prompting enhances the complex reasoning capabilities of LLMs by presenting a sequence of smaller reasoning steps, aiding the model with comprehension of the task (Wei et al., 2022). This can be coupled with one-shot or few-shot prompting to further amplify model performance. Tree-of-Thoughts prompting provides a framework for tasks demanding exploration or strategic reasoning, resembling an expanded version of CoT with multiple nodes and branches (Yao et al., 2023).

2.2.3 Fine-tuning

Fine-tuning is a process where a pre-trained model is further trained on a smaller and domain-specific dataset. This process adjusts the model parameters to specialize its knowledge and improve its performance on tasks within the specific domain. However, fine-tuning large models with billions to trillions of parameters poses practical challenges due to high computational costs and single-GPU RAM limitations. To address this, parameter-efficient fine-tuning (PEFT) methods have been developed, training a limited number of parameters instead of full parameter fine-tuning. Existing PEFT methods could be categorized into several groups: addition-based PEFT fine-tune newly introduced parameters, selection-based PEFT fine-tune a subset of existing parameters, and reparametrization-based PEFT fine-tune a low-rank representation of full parameters (Lialin et al., 2023).

Among the reparametrization-based methods , Low-Rank Adaptation (LoRA) and its variants, such as Quantized Low-Rank Adaptation (QLoRA), are the most prominent PEFT techniques for fine-tuning LLMs. LoRA decomposes the updates of weight matrix into products of two low-rank matrices, which efficiently reduces training parameters, enables pretrained models to be shared across multiple tasks, and incurs no inference latency (Hu et al., 2022). This approach has led to the development of numerous open-source models with outstanding performance (Cui et al., 2023; Yang et al., 2023). QLoRA extends LoRA by introducing quantization to enhance parameter efficiency during fine-tuning (Dettmers et al., 2023). It combines LoRA principles with 4-bit NormalFloat quantization and double quantization techniques. Studies demonstrate that both LoRA and QLoRA achieve performances comparable to full parameter tuning across models of various sizes (Patel, 2023; Google, 2023). QLoRA consumes approximately 75% less peak GPU memory than LoRA, yet LoRA demonstrates a 66% improvement in speed and is 40% more cost-efficient than QLoRA (Google, 2023). Thus, LoRA is preferred when GPU memory is sufficient for fine-tuning tasks, while QLoRA is recommended when memory constraint is a concern.

2.2.4 Domain specialization of LLMs

Several studies have leveraged LLMs in various domains by employing prompt engineering and fine-tuning techniques according to survey (Zhao et al., 2023). For instance, in the legal domain, ChatLaw utilized Q&A pairs derived from news, exam questions, and legal documents to fine-tune open-source LLMs for legal inquiries (Cui et al., 2023). Similarly, DISC-LawLLM prepared

https://doi.org/10.1017/S1748499525100079 Published online by Cambridge University Press

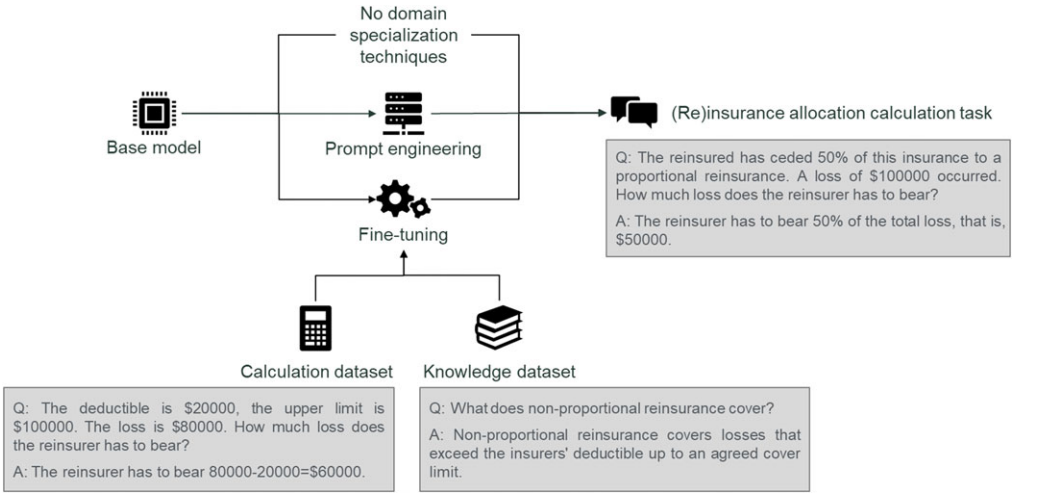


Figure 1. Illustration of experiment set-up. The evaluation framework incorporates human assessment, using open source LLMs as base models and benchmarks. We employ prompt engineering and fine-tuning to achieve domain specialization. Recognizing the challenges of gathering extensive task-specific training data (calculation dataset), we further examine the impact of fine-tuning with background knowledge (knowledge dataset).

diverse training datasets, including multi-choice questions, law retrieval, legal consultation, and agreement writing (Yue et al., 2023). In finance, FinGPT fine-tuned LLMs with news, reports, and social media data for sentiment analysis, relation extraction, headline analysis, and more (Yang et al., 2023). InvestLM curated instructions from various sources for tasks like sentiment analysis and document classification focusing on the investment industry (Yang et al., 2023). In the medical domain, Med-PaLM focused on medical Q&A datasets for high-quality medical answers (Singhal et al., 2023), while HuaTuo trained on real or ChatGPT-generated conversations for medical consultations (Wang et al., 2023). These studies underscore the potential of LLMs for fine-tuning in diverse domains.

Despite the proliferation of the above domain-specific models, there remains a notable absence of LLMs with expertise in (re)insurance sector. One of the quantitative reasoning tasks in this domain involves calculating the allocations of liabilities, premiums, and claims between reinsured and reinsurer, which requires a profound understanding of (re)insurance and mathematical principles. In this work, we evaluate the development of domain-specific language model tailored to this task by implementing prompt engineering and fine-tuning techniques on existing open-source LLMs.

3. Methodology

In this section, we detail the experiment set-up for the evaluation of domain-specific LLMs on the calculation task of (re)insurance allocation, as shown in Fig. 1. The data used in the experiment are presented in Section 3.1. In Section 3.2, we focus on the implementation details and computational aspects of the LLMs and associated prompt engineering and fine-tuning techniques. The evaluation metric is discussed in Section 3.3.

3.1 Data

The data utilized in this work are sourced from reinsurance training materials, which contain a general overview of reinsurance business, as well as a detailed introduction to common types of

https://doi.org/10.1017/S1748499525100079 Published online by Cambridge University Press

reinsurance treaties, including proportional reinsurance and non-proportional reinsurance. They are used to construct evaluation and training datasets described in the following subsections.

3.1.1 Evaluation dataset

The evaluation dataset comprises the calculations of (re)insurance allocation in the form of question and answering (Q&A) extracted from the aforementioned training materials. The dataset consists of 100 questions across 25 different types, with each type featuring multiple scenarios. For instance, questions involving reinsurer payment calculations may vary based on factors such as deductible levels and loss amounts, or the treatment of underwritten buildings as single or separate risks.

3.1.2 Training dataset

The training dataset is customized to evaluate fine-tuning effects on model performance, consisting of two subsets: task-specific data and background knowledge data. Task-specific data encompass calculation Q&A pairs similar to those in the evaluation dataset. Recognizing the challenges of gathering comprehensive task-specific data, background knowledge data are included to assess the impact of fine-tuning using broader domain knowledge.

(i) Task-specific data: calculation dataset

The calculation dataset consists of questions with various reinsurance types and difficulty levels. Proportional reinsurance questions constitute 20%, non-proportional reinsurance questions 64%, while 12% involve combined structures, and 4% are accounting calculations. Tasks range from simple deductions to complex computations involving multiple treaty types, reflecting the intricacies of reinsurance financial calculations. Each question specifies the desired answer, background, and conditions, with clear explanations provided in the answer to facilitate model comprehension. For multi-step reasoning and calculation tasks, CoT methodology is employed to guide the model. The dataset contains 25 types, with experiments conducted on different training data sizes to gauge performance improvements, with a maximum of 1150 questions, averaging 46 questions per type. The types of questions in the calculation dataset and the percentage of each type are shown in Table 1. There are several examples of the calculation dataset shown in Appendix A. Calculation dataset.

(ii) Background knowledge data: knowledge dataset

The knowledge dataset is derived from the aforementioned materials, comprising questions and corresponding text paragraphs structured into Q&A pairs. For instance, paragraphs detailing the advantages of quota share reinsurance are transformed into Q&A pairs with corresponding questions and answers. Examples of this dataset are provided in Appendix B. Knowledge dataset. We manually extract 600 questions from the training materials as the base data. To augment the dataset, we utilize the open-source Llama 2-Chat 70B model to rewrite Q&A pairs in varied phrases, resulting in a total of 1200 pairs in the knowledge dataset.

3.2 Computational details

In this experiment, Llama 2-Chat models of various sizes are adopted as the base models and the baseline for evaluation. The models are accessed from the model catalog in Azure Machine Learning Studio. We employ one-shot prompting and LoRA as the prompt engineering and fine-tuning techniques for domain specialization, respectively. The example provided to the models is selected from the training dataset, which follows the CoT methodology with detailed reasoning steps for model comprehension. Examples of the prompts are presented in Appendix C. One-shot prompt. With sufficient GPU memory, LoRA is applied for fine-tuning, with a rank of 8,

https://doi.org/10.1017/S1748499525100079 Published online by Cambridge University Press

Table 1. Type of questions in the calculation dataset

Group	Question type	Percentage (%)
Proportional	Calculate premium and loss for quota share treaty	8
	Calculate premium and loss for surplus treaty	12
Non-proportional	Definition of excess of loss treaty	16
	Calculate loss for excess of loss treaty and variants of excess of loss *(Including excess of loss per risk, excess of loss per person, catastrophe excess of loss, excess of loss with multiple layers, excess of loss with subrogation)*	44
	Calculate loss for stop loss treaty	4
Combined structure	Calculate the premium and claim for combination of quota share and surplus	4
	Calculate loss for combination of surplus and excess of loss per risk	4
	Calculate loss for combination of quota share and excess of loss per risk	4
Accounting	Calculate profit commission	4

target modules of four weight matrices in the self-attention module (q_proj, k_proj, v_proj, and o_proj) and two in the multilayer perceptron module (up_proj and down_proj), and optimizer of AdamW. A learning rate of 3e-4 with warm-up and linear decay schedule is applied in the training process. The batch size is 128 and the micro batch size is 4. The number of epochs is determined based on the training loss and evaluation loss, while model checkpoints are saved along the training process. The fine-tuning is conducted on two A100 (80G) GPUs, with training time varying based on number of epochs, data size, and model size. For instance, fine-tuning the Llama 2-Chat 7B model with calculation datasets (1150 calculations) under 5 epochs required 8 minutes, while the Llama 2-Chat 13B model required 14 minutes. For Llama 2-Chat 70B models, we encountered resource limitations preventing us from fine-tuning on two A100 GPUs. On the one hand, utilizing more computational resources would significantly increase costs. On the other hand, loading the model in 8-bit precision for fine-tuning would result in reduced model performance. Given these considerations, we employ 70B models only for generation problems without fine-tuning.

3.3 Evaluation metric

Given the quantitative reasoning-intensive nature of the task, necessitating multiple steps of reasoning to obtain a conclusive answer, model performance evaluation is conducted through human assessment. The evaluation dataset comprises 100 questions, each carrying a maximum score of 1 point, totaling 100 points overall. Model outputs are graded on a scale of 0–1, depending on their proficiency in problem-solving, as delineated in Table 2.

4. Results

In this section, we provide an in-depth analysis of our experimental findings. Section 4.1 details the evaluation of the baseline models and their enhanced performance through the application of prompt engineering. Notably, the implementation of one-shot prompting has yielded a significant performance boost, especially for the 70B model. Section 4.2 presents the outcomes of

https://doi.org/10.1017/S1748499525100079 Published online by Cambridge University Press

Table 2. Score deduction criteria for model outputs in the evaluation dataset, with each question assigned a maximum score of 1 point out of 100

Score deduction	Situation
No deduction (deduction of 0 point)	Correct answer
Partial deduction (deduction of $\frac{1}{2n}$ points)	The question requires n outcomes, one out of n outcomes is incorrect due to miscalculation (though the correct calculation formula id presented)
Full deduction (deduction of $\frac{1}{n}$ points)	The question requires n outcomes, one out of the n outcomes is incorrect due to lack of problem-solving reasoning

Detailed examples are available in Appendix D. Marking examples.

Table 3. The performance comparison of Llama 2-Chat models with and without one-shot prompting

	Score		
Model	Base model	Base model + one-shot prompt	Improvement (%)
Llama 2-Chat 7B	13.83	20.83	+ 50.61
Llama 2-Chat 13B	17.17	25.25	+ 47.06
Llama 2-Chat 70B	19.92	46.67	+ 134.28
GPT-4	63.88	72.42	+ 13.37

fine-tuning, demonstrating that this method surpasses one-shot prompting in terms of performance improvement. Furthermore, Section 4.3 explores the influence of background knowledge, revealing that models reap the greatest benefits when task-specific datasets are limited. In the final analysis, Sections 4.4 and 4.5 examine the model performance before and after fine-tuning. It is evident that reasoning capability is the key determinant in enhancing model performance on quantitative calculation tasks. The general proficiency of baseline models is well-maintained after fine-tuning, as evidenced by their performance in public benchmarks.

4.1 Baseline and prompt engineering

We first evaluate the Llama 2-Chat models of 7B, 13B, and 70B parameters with and without the application of one-shot prompting using the evaluation dataset described in Section 3.1.1, which includes calculation questions of (re)insurance allocation of varying complexity, ranging from simple to intricate. We also evaluate GPT 4, which was released in March 2023, on the same test dataset as a benchmark. We calculate a score out of 100 for each model employing the evaluation metric in Section 3.3.

The findings, summarized in Table 3, reveal anticipated outcomes. (1) The Llama 2-Chat models demonstrate unsatisfactory performance in addressing reinsurance calculation tasks, primarily due to their lack of specialized domain knowledge. Despite a maximum score of 100 points, all three models score below 20 points. The 70B model outperforms its smaller counterparts, indicating a positive correlation between model size and knowledge retention in model pre-training. Meanwhile, GPT 4 yields 63.88 without further prompting or fine-tuning. (2) The integration of prompt engineering, i.e., one-shot prompting with CoT presented in Section 3.2, leads to improvements in model performance. Most notably, larger models, characterized by enhanced reasoning abilities and proficiency in following instructions, experience more benefits from prompt engineering interventions. This is demonstrated by the score of 70B model, which more than doubles from 19.92 to 46.67 points by using one-shot prompting. However, the improvements through prompt engineering are generally limited, e.g., the 7B and 13B models encounter difficulties. While this approach enables models to improve on easy-to-moderate-level problems, the complexity of certain tasks underscores the essential requirement for infusing domain-specific data via fine-tuning.

https://doi.org/10.1017/S1748499525100079 Published online by Cambridge University Press

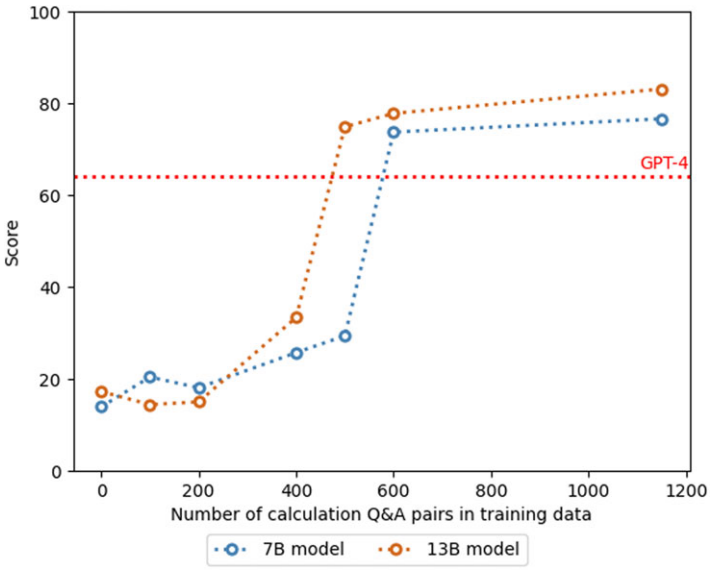


Figure 2. The effect of fine-tuning with different sizes of the calculation dataset, Llama 2-Chat 7B and 13B models. Fine-tuning smaller LLMs can achieve significantly better performance. The larger the amount of training data, the higher the performance. Determining the optimal size of the training data is crucial for achieving peak performance in practical applications.

It is worth noting that in the one-shot prompt experiment, we provided examples of the same question type as prompts, which would overestimate the effectiveness of prompt engineering. In practice, it is impractical or laborious to feed the LLM with appropriate examples.

The percentages of improvement are also presented. The Llama 2-Chat models exhibit limited efficacy in addressing reinsurance calculation tasks, primarily due to their lack of specialized domain knowledge. It can be seen that the application of one-shot prompting results in a performance increase, particularly for the 70B model.

4.2 Fine-tuning with task-specific data

The findings in the previous subsection reveal challenges faced by models when addressing quantitative calculations on (re)insurance allocation, particularly noticeable in smaller models, despite the supplementary aid provided by one-shot prompts. In this subsection, we examine the efficacy of fine-tuning models using task-specific data contained in the calculation dataset described in Section 3.1.2. Due to the constraint of the computational resource, we only fine-tune the Llama 2-Chat 7B and 13B models.

Our results, depicted in Fig. 2, highlight a substantial increase in scores post fine-tuning with the calculation dataset. Prior to fine-tuning, the 7B model achieves 13.83 points based on the evaluation dataset, while the 13B model attains 17.17 points. Following fine-tuning with the complete calculation dataset, their performances improve significantly to 76.58 and 83.08 points, representing 521% and 382% improvements, respectively, outperforming GPT 4. Furthermore, the performance achieved through fine-tuning significantly surpasses that obtained from the one-shot prompting. Notably, the fine-tuned 7B and 13B models outperform the 70B model by a considerable margin. This outcome aligns with previous reports indicating that fine-tuning can customize LLMs to domain-specific requirements (Cui et al., 2023; Yang et al., 2023). Moreover, a discernible trend is observed wherein model performance improves with an increase in the size of the training data. The 7B model shows rapid improvement upon reaching 500–600 calculation Q&A pairs, while the 13B model exhibits a similar trend at 400–500 calculation Q&A pairs. Identifying the

https://doi.org/10.1017/S1748499525100079 Published online by Cambridge University Press

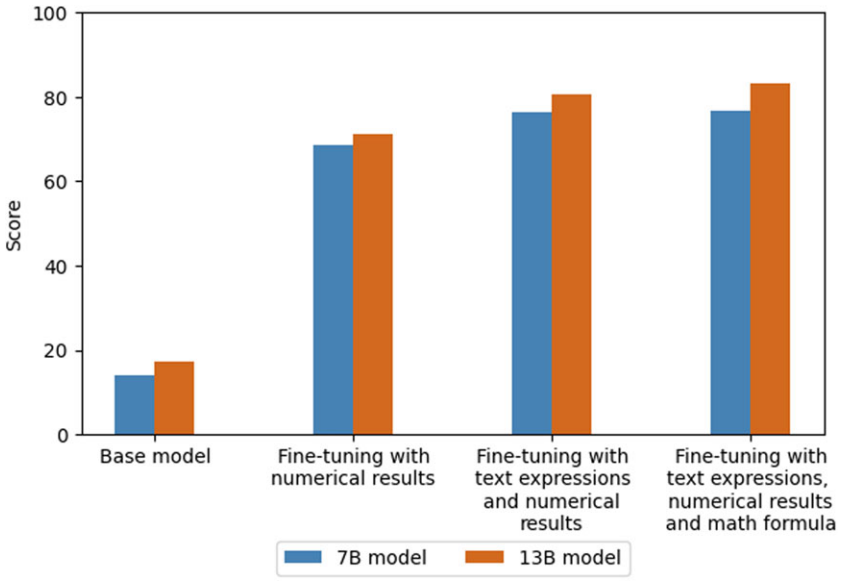


Figure 3. Fine-tuning the Llama 2-Chat models with 1150 calculation samples, but with explanations at different level of details: (a) base model. (b) Only numerical results are included in the training data. (c) Text expressions and numerical results are contained in the training data. (d) Text expressions, numerical results, and math formula achieving results are all provided in the training data. This figure shows that the more detailed the explanation provided in the training data, the better the models perform.

optimal point at which model performance peaks is crucial for practical applications, as it enables the attainment of optimal results with minimal resources. As depicted in the figure, the 13B model marginally outperforms the 7B model, although the difference is not statistically significant.

These results underscore the significant performance enhancements achievable through fine-tuning pre-trained language models using task-specific data, especially in complex tasks involving reasoning and calculation steps. This has significant implications for the development of conversational LLMs, suggesting that fine-tuning smaller models on task-specific datasets may be more effective and performant than relying solely on large pre-trained models.

Given that our evaluation dataset comprises problems necessitating reasoning and calculation steps, we further explored the influence of CoT in the training data on model performance. Three types of training datasets are constructed and compared: (1) Only numerical results are included. (2) Text expressions and numerical results are contained. (3) Text expressions, numerical results, and mathematical formulas achieving results are all provided. An example is presented in Appendix E. CoT test training data. Scores of the 7B and 13B models fine-tuned using the aforementioned datasets are evaluated and compared in Fig. 3.

The models fine-tuned with detailed CoT processes outperform the others. Notably, the more detailed the explanation provided in the training data, the better the model performs, consistent with human intuition. However, the observed differences between the various datasets in this experiment are not statistically significant. This could be attributed to the availability of a sufficiently large dataset, enabling the model to discern patterns and learn effectively. Additionally, our evaluation dataset comprises both simple questions, easily solved without detailed explanations, and complex questions, challenging to solve even with detailed explanations, potentially contributing to similar scores among different training data formats.

4.3 Fine-tuning with background knowledge data

In the realm of LLM fine-tuning, the acquisition of task-specific data has been recognized as a pivotal factor in enhancing the performance of LLMs on downstream tasks, as evidenced in prior

https://doi.org/10.1017/S1748499525100079 Published online by Cambridge University Press

research and corroborated by our own findings (Yue et al., 2023; Yang et al., 2023). However, the endeavor of amassing comprehensive task-specific datasets is often fraught with challenges, such as time constraints and inherent complexities. This issue becomes particularly pronounced within domains hosting numerous downstream tasks, where the sheer volume of data required for each task may be insufficient for robust model tuning. On the other hand, the reservoir of background knowledge in related domains is often abundant and readily accessible. In this subsection, we examine the effect of fine-tuning with background knowledge data.

We embark on fine-tuning the 7B and 13B Llama 2-Chat models using the knowledge dataset presented in Section 3.1.2. The performance of these fine-tuned models is first evaluated in conjunction with one-shot prompting. Subsequently, we explore the synergistic effects of combining the knowledge dataset with the calculation dataset during fine-tuning to comprehensively analyze the impact of both datasets.

The results of fine-tuning with background knowledge, both individually and in conjunction with one-shot prompts, are depicted in Fig. 4. Notably, our observations suggest that fine-tuning with the knowledge dataset alone does not significantly boost the ability of model in addressing calculation tasks. There exists a discernible decline in performance, particularly for the 13B model. It appears that the models fine-tuned with background knowledge tend to provide interpretations and descriptions of the problem rather than offering the final answer through calculations. This phenomenon could be attributed to the discrepancy between the nature of the training knowledge, which primarily comprises descriptive texts on (re)insurance topics, and the calculation tasks, which necessitate quantitative reasoning and calculation based on these concepts. However, when augmented with one-shot prompting, the integration of domain knowledge into smaller LLMs yields a substantial enhancement in performance. Remarkably, the 13B model fine-tuned with background knowledge, when coupled with the one-shot prompt, achieves a performance level comparable to that of the 70B model. This is attributed to the one-shot prompt provided, effectively bridging the divide between the training data and the specific task. Consequently, the model is better equipped to follow instructions on tackling calculations, together with the knowledge injected into it enhances its understanding and reasoning capabilities.

Second, we further fine-tune the models using the knowledge dataset as well as various sizes of calculation dataset and examine their respective performance. As depicted in Fig. 5, the peak performance attainable by the models remains consistent irrespective of the inclusion of background knowledge, approximately at 75 and 80 points for the 7B and 13B models, respectively. Notably, the addition of the knowledge dataset to the training data facilitates reaching maximum performance levels with a reduced amount of calculation Q&A pairs. For instance, optimal performance for the 7B model is achieved with 600 calculation Q&A pairs, whereas incorporating the background knowledge dataset enables peak performance with only 400 calculation Q&A pairs. We also notice distinct performance trends with and without the use of knowledge dataset. While performance improvement based on fine-tuning with calculation dataset alone follows a convex curve, the integration of knowledge data yields a concave curve, indicating more rapid performance gains. In the case of the 13B model fine-tuned with only 200 calculation Q&A pairs, the performance of model remains almost the same as that of the base model. However, augmenting the training data with the knowledge dataset and 200 calculation Q&A pairs results in a substantial performance boost, with the score escalating to 60.67 points, representing a 4-fold increase. These findings are consistent with prior research (Aracena et al., 2023), which demonstrates the effectiveness of incorporating knowledge dataset and indicates an alternative approach when constructing task-specific datasets is challenging and resource-intensive.

4.4 Error analysis

In this subsection, we provide an analysis of the calculation tasks that are not correctly solved by the LLMs. Our evaluation criteria reveal that LLMs often exhibit deficiencies in reinsurance

https://doi.org/10.1017/S1748499525100079 Published online by Cambridge University Press

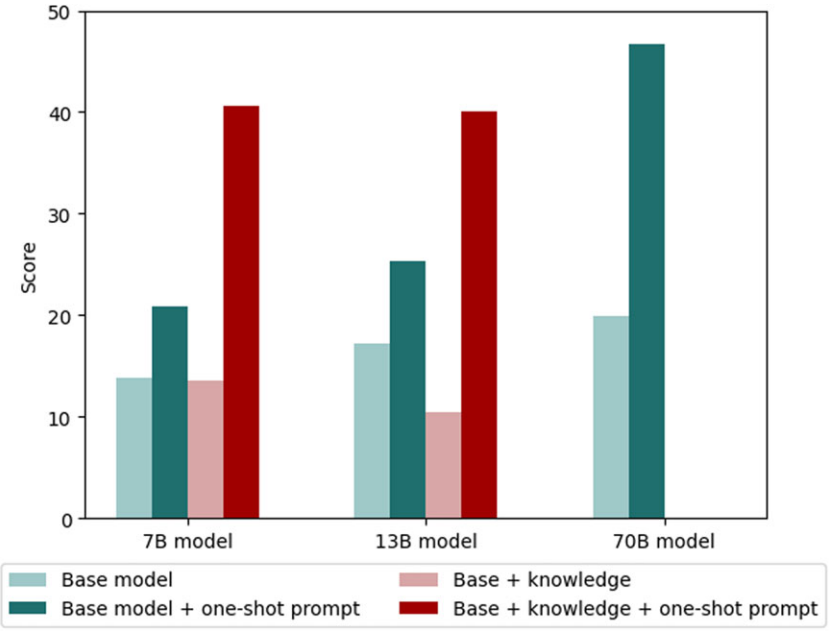


Figure 4. The impact of fine-tuning utilizing a knowledge dataset and a one-shot prompt on the Llama 2-Chat 7B, 13B, and 70B models. The numbers for base model and base model + one-shot prompt are sourced directly from Section 4.1. Fine-tuning the LLM solely with background knowledge does not help with its ability to solve specific tasks in the domain. However, when combined with one-shot prompt, injecting domain knowledge into smaller LLMs results in a significant increase in model performance. This increase is comparable to that of larger LLMs without domain knowledge.

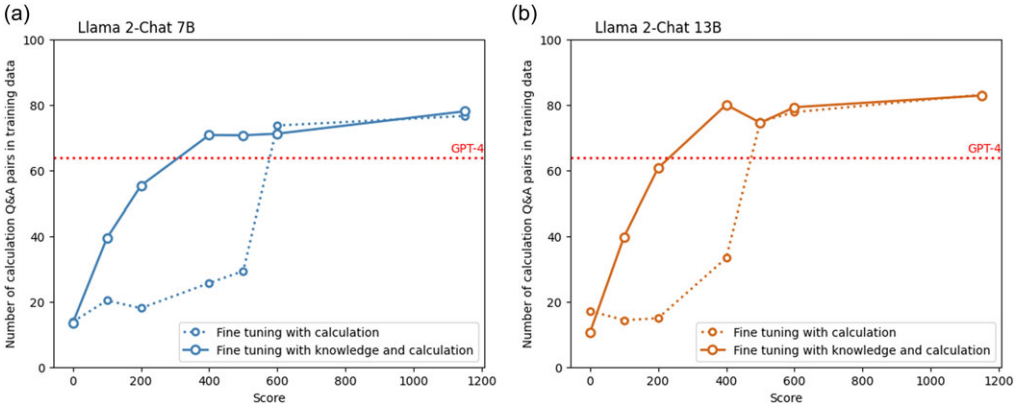


Figure 5. The evaluation scores of the Llama 2-Chat 7B (a) and 13B (b) models fine-tuned on various sizes of calculation dataset. The solid lines represent models initially fine-tuned using knowledge dataset and calculation dataset, while the dotted lines indicate models with only calculation dataset. It can be seen that the model performance remains the same irrespective of domain knowledge infusion, with the 13B model exhibiting slightly better score using fewer calculation Q&A pairs. Yet, when fine-tuned with knowledge dataset, the model performance escalates more rapidly with increases in the size of calculation dataset, indicating an alternative approach when constructing task-specific datasets is challenging and resource-intensive.

calculation, attributable to two primary factors. Firstly, insufficient knowledge or weak reasoning abilities make it difficult for models to understand and solve problems. Secondly, while the models may possess the necessary procedural knowledge, deficiencies in calculation accuracy lead to incorrect responses. These inaccuracies may stem from rounding errors inherent in floating-point

https://doi.org/10.1017/S1748499525100079 Published online by Cambridge University Press

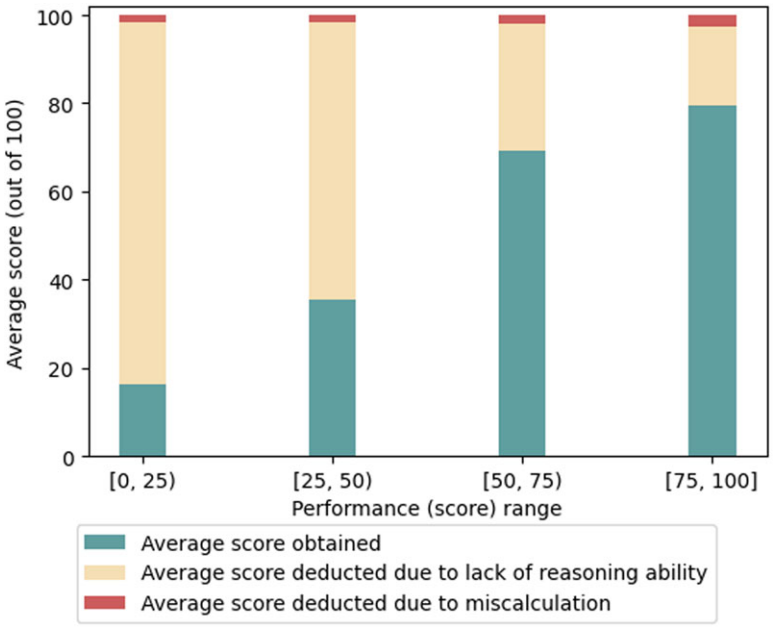


Figure 6. The average score obtained and deducted in different performance ranges of all 38 experiments in this study. The scores are deducted for two reasons: lack of reasoning ability and miscalculation, in which reasoning ability is the dominant factor. Fine-tuning can enhance the reasoning abilities, as evidenced by the correlation between improved model performance and reduced scores deducted due to reasoning errors.

data formats or from the complexity of computation processes, where intricate operations such as A^*B/C increase the likelihood of errors. Currently, the internal mechanisms through which LLMs perform such computations remain poorly understood. A recent study by Anthropic has indicated that transformers are not performing the calculation; rather, they are estimating the answer by memorizing frequently used calculation results. While a step-by-step CoT chain could be provided by LLM, it is not representative of the model's actual mechanisms (Lindsey et al., 2025). Addressing these challenges may necessitate the integration of external calculators alongside LLMs. We have visualized the average of scores obtained or deducted resulting from deficient reasoning and calculation abilities in different performance ranges in Fig. 6, providing further insights into model performance.

Our investigation reveals a correlation between enhanced model performance and a reduction in lost scores attributable to deficient reasoning capabilities, indicating an improvement in the reasoning prowess of the models concerning quantitative tasks in the reinsurance domain. Conversely, an increase in model performance correlates with a rise in lost scores due to miscalculation. The average of scores deducted due to miscalculation ranges from 1.5 to 2.5, with higher model performance resulting in greater deductions. Although initially counterintuitive, this trend aligns with our definition of miscalculation. The likelihood of miscalculation is contingent upon the model's mastery of problem-solving steps and accurate presentation of relevant formulas, which are enhanced as reasoning abilities improve. Overall, lack of reasoning abilities accounts for most of the reasons for score deductions, where emphasis needs to be placed to improve model performance. In addition, for a given model, the marginal improvement in model performance decreases as the score increases and eventually converges to a certain level. If higher performance is required, a larger model would be an option, such as the Llama 2-Chat 70B model.

Our study underscores those errors in reinsurance quantitative tasks by LLMs stem from deficiencies in knowledge, reasoning, and calculation abilities. Fine-tuning LLMs on reinsurance

https://doi.org/10.1017/S1748499525100079 Published online by Cambridge University Press

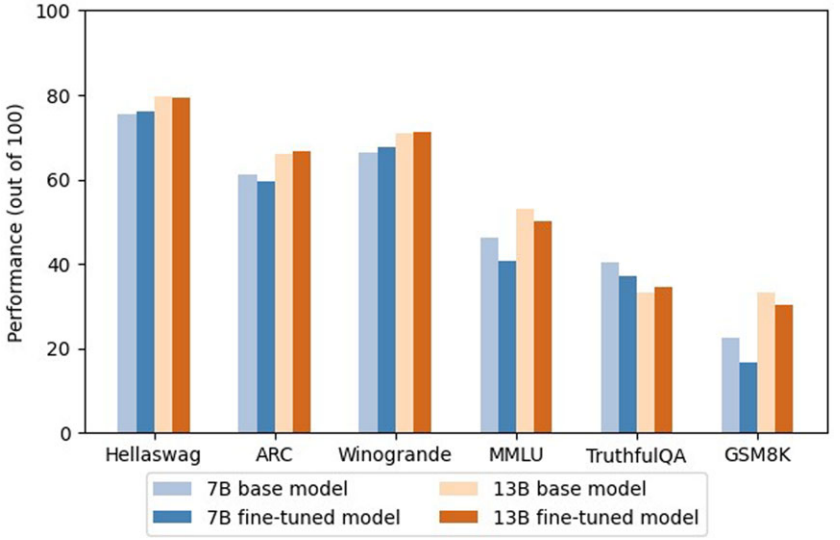


Figure 7. The evaluation of base models and fine-tuned models, augmented with both knowledge and calculation datasets, on publicly available benchmark assessments. After fine-tuning the Llama 2-Chat models with background knowledge and task-specific data, their knowledge in common sense, professional expertise, and arithmetic reasoning remains.

calculation datasets enhances their reasoning capabilities. To address miscalculation, integrating an external calculator with LLMs can mitigate this issue and improve model performance by 1.5–2.5 points. These findings carry significant implications for leveraging LLMs in quantitative reasoning-intensive tasks within the (re)insurance industry.

4.5 Public benchmark

In order to assess the catastrophic forgetting phenomenon in LLMs following fine-tuning, i.e. LLMs might completely or substantially forget the information related to previously learned tasks after being trained on a new task, we investigate the performance of both base models and fine-tuned models on general knowledge and calculation tasks. Our evaluation encompasses several widely recognized benchmark datasets, including Hellaswag, ARC, and Winogrande, which gauge common sense reasoning abilities, MMLU and TruthfulQ&A, assessing knowledge across various subjects, and GSM8K, measuring arithmetic reasoning skills. The outcomes from these benchmark assessments are depicted in Fig. 7. Notably, the fine-tuned model maintains its proficiency in common sense reasoning, professional knowledge, and arithmetic reasoning tasks, suggesting retention of acquired knowledge post-fine-tuning.

5. Conclusions and future work

This study assesses the efficacy of LLMs via experiments on the calculation tasks within the (re)insurance domain. We evaluate the performance of open-source Llama 2-Chat models of 7B, 13B, and 70B sizes, with a blend of prompt engineering and fine-tuning with respect to both task-specific and background knowledge datasets.

Our empirical findings indicate that compared to prompt engineering, fine-tuning the 7B and 13B models with task-specific data engenders noteworthy enhancements in their capacity to tackle moderate to hard problems. Scores exhibit a remarkable leap, increasing from approximately 15 to nearly 80. In light of scenarios where the collection of task-specific data proves arduous, supplementing models with background knowledge is an effective alternative

https://doi.org/10.1017/S1748499525100079 Published online by Cambridge University Press

that achieves performance on par with models fine-tuned on extensive task-specific dataset. The model's performance is limited by its reasoning and calculation abilities, with reasoning abilities being the main reason and can be improved by prompt engineering or fine-tuning. After fine-tuning, the model's performance on specific tasks improves without forgetting common sense and professional knowledge.

Furthermore, the above insights can be used to select the optimal domain specialization technique for general LLMs. For tasks of simpler complexity with constrained data, prompt engineering may suffice. Conversely, tasks necessitating advanced reasoning and calculation proficiencies warrant fine-tuning with task-specific data. In cases where acquiring task-specific data pose challenges or is cost-prohibitive, fine-tuning LLMs with background knowledge emerges as a valuable adjunct. This method can enhance model performance without requiring extensive domain-specific datasets. Additionally, leveraging the same background knowledge through retrieval-augmented generation techniques may yield comparable improvements to those achieved through fine-tuning.

However, challenges remain in deploying LLMs in real-world (re)insurance applications. Data acquisition is a critical hurdle where fine-tuning is necessary. Obtaining sufficient, high-quality datasets is difficult, and labeling training data requires heavy human effort. Security concerns may arise due to the inclusion of personally identifiable information such as names, genders, and addresses in the input context. This could influence the selection of open-source or closed-source models. There may be a trade-off between performance and data protection risks. Furthermore, the potential impact of inaccurate or biased LLM outputs warrants careful consideration. In scenarios where erroneous outputs could lead to severe consequences, rigorous evaluation should be implemented prior to deployment. For business process automation, a hybrid approach involving human oversight may be more prudent, with LLMs serving either as reviewers subsequent to human input or as preliminary suggesters prior to human decisions. Lastly, the effort required for ongoing maintenance should not be underestimated. Close-source models often undergo frequent updates, while open-source models release new versions periodically. Consequently, previously selected models and prompts may no longer represent optimal choices, necessitating regular reassessment and potential adjustments to maintain system efficacy.

Representative examples of the training dataset in this study are provided in the Appendix to facilitate reproducibility. While the full proprietary datasets cannot be disclosed, these examples offer sufficient details to guide other researchers or institutions in constructing their domain-specific or task-specific corpora for fine-tuning. Consequently, other teams can replicate the experiments and evaluate the results in their own contexts, and refine or extend the approach for further applications within the reinsurance sector and related domains.

This study explores the potential for the application of LLMs in (re)insurance domain. The effectiveness of LLMs in handling unstructured data and their strong reasoning and calculation capabilities make them suitable candidates for utilization as AI assistants, with the potential to improve overall productivity and efficiency. However, the model and techniques need to be carefully chosen according to the task, in consideration of costs and the benefits. Meanwhile, data security, ethical, and regulatory considerations cannot be ignored when putting LLMs into use in order to avoid and mitigate potential risks.

Data availability statement. The data used in this study are internal training materials within Swiss Re and cannot be made publicly available.

Funding statement. This work received no specific grant from any funding agency, commercial, or not-for-profit sectors.

Competing interest. The authors declare none.

Disclaimer. The assessments and opinions expressed in this paper are the author's personal assessments and opinions and should not be taken to reflect Swiss Re's position on any issue. Further, Swiss Re disclaims any and all liability arising from the author's contribution to this article.

https://doi.org/10.1017/S1748499525100079 Published online by Cambridge University Press

References

Aracena, C., Rodríguez, N., Rocco, V., & **Dunstan, J.** (2023). Pre-trained language models in Spanish for health insurance coverage. In *Proceedings of the 5th clinical natural language processing workshop*.

Balona, C. (2023). ActuaryGPT: Applications of large language models to insurance and actuarial work. Available at SSRN 4543652.

Beeching, E., Fourrier, C., Habib, N., Han, S., Lambert, N., Rajani, N., & **Wolf, T.** (2023). Open LLM Leaderboard. Retrieved from https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard.

Brown, P. F. (1990). Class-based n-gram models of natural language. *Computational Linguistics*, **18**, 18.

Brown, P. F., Cocke, J., Della Pietra, S. A., Della Pietra, V. J., Jelinek, F., Lafferty, J., & **Roossin, P. S.** (1990). A statistical approach to machine translation. *Computational Linguistics*, **16**(2), 79–85.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., & **Mercer, R. L.** (1993). The mathematics of statistical machine translation: Parameter estimation.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., & **Amodei, D.** (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems, 33*, vol. **33** (pp. 1877–1901).

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., & **Wei, J.** (2024). Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, **25**(70).

Chung, J., Gulcehre, C., Cho, K., & **Bengio, Y.** (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint* arXiv:1412.3555.

Cui, J., Li, Z., Yan, Y., Chen, B., & **Yuan, L.** (2023). Chatlaw: Open-source legal large language model with integrated external knowledge bases. CoRR.

Dettmers, T., Pagnoni, A., Holtzman, A., & **Zettlemoyer, L.** (2023) Qlora: Efficient finetuning of quantized llms. In *Advances in neural information processing systems* (pp. 10088–10115).

Devlin, J., Chang, M.-W., Lee, K., & **Toutanova, K.** (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Human language technologies*, vol. **1**, Long and Short Papers.Long and Short Papers.

Dodson, S. (2023). Domain specific generative AI: Pre-training, fine-tuning, and RAG. Retrieved from https://www.elastic.co/search-labs/blog/articles/domain-specific-generative-ai-pre-training-fine-tuning-rag.

Gao, J., & **Lin, C.-Y.** (2004). Introduction to the special issue on statistical language modeling. *ACM Transactions on Asian Language Information Processing (TALIP)*, **3**(2), 87–93.

Google. (2023). LoRA and QLoRA recommendations for LLMs. Retrieved from https://cloud.google.com/vertex-ai/docs/model-garden/lora-qlora.

Hochreiter, S., & **Schmidhuber, J.** (1997). Long short-term memory. *Neural Computation*, **9**(8), 1735–1780.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., & **Chen, W.** (2022). Lora: Low-rank adaptation of large language models. *ICLR*, **1**(2), 3.

Jelinek, F. (1990). Self-organized language modeling for speech recognition. In *Readings in speech recognition* (pp. 450–506).

Jelinek, F. (1998). *Statistical methods for speech recognition*. MIT Press.

Jelinek, F., Lafferty, J. D., & **Mercer, R. L.** (1992). *Basic methods of probabilistic context free grammars*. Springer Berlin Heidelberg (pp. 345–360).

Lee, G., Manski, S., & **Maiti, T.** (2020). Actuarial applications of word embedding models. *ASTIN Bulletin*, **50**(1), 1–24.

Lialin, V., Deshpande, V., & **Rumshisky, A.** (2023). Scaling down to scale up: A guide to parameter-efficient fine-tuning, 583. *arXiv preprint* arXiv:2303.15647.

Lindsey, J., Gurnee, W., Ameisen, E., Chen, B., Pearce, A., Turner, N. L., & **McDougall, C.** (2025). On the Biology of a Large Language Model. Retrieved from https://transformer-circuits.pub/2025/attribution-graphs/biology.html.

Liu, H. (2023). llava-v1.6-34b. Retrieved from https://huggingface.co/liuhaotian/llava-v1.6-34b.

OpenAI. (2022). Introducing ChatGPT. Retrieved from https://openai.com/blog/chatgpt.

Patel, P. (2023). In-depth guide to fine-tuning LLMs with LoRA and QLoRA. Retrieved from https://www.mercity.ai/blog-post/guide-to-fine-tuning-llms-with-lora-and-qlora.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & **Sutskever, I.** (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, **1**(8), 9.

Rosenfeld, R. (2000). Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE*, **88**(8), 1270–1278.

Rumelhart, D. E., Hinton, G. E., & **Williams, R. J.** (1986). Learning representations by back-propagating errors. *Nature*, **323**(6088), 533–536.

Saddi, V. R., Gnanapa, B., Boddu, S., & **Logeshwaran, J.** (2023). The role of natural language processing in detecting insurance fraud. In *4th international conference on communication, computing and Industry 6.0 (C216), Bangalore, India* (pp. 1–6).

Saravia, E. (2022). Prompt Engineering Guid. Retrieved from https://github.com/dair-ai/Prompt-Engineering-Guide-https://www.promptingguide.ai/.

https://doi.org/10.1017/S1748499525100079 Published online by Cambridge University Press

Singhal, **K.**, **Azizi**, **S.**, **Tu**, **T.**, **Mahdavi**, **S. S.**, **Wei**, **J.**, **Chung**, **H. W.**, & **Natarajan**, **V.** (2023). Large language models encode clinical knowledge. *Nature*, **620**, 172–180.

Singhal, **K.**, **Tu**, **T.**, **Gottweis**, **J.**, **Sayres**, **R.**, **Wulczyn**, **E.**, **Hou**, **L.**, & **Natarajan**, **V.** (2025). Towards expert-level medical question answering with large language models. *Nature Medicine*, 1–8.

Stolcke, **A.** (2002). SRILM-an extensible language modeling toolkit. In: *Seventh international conference on spoken language processing*.

Together. (2023). Llama-2-7B-32K-Instruct. Retrieved from https://huggingface.co/togethercomputer/Llama-2-7B-32K-Instruct.

Touvron, **H.**, **Martin**, **L.**, **Stone**, **K.**, **Albert**, **P.**, **Almahairi**, **A.**, **Babaei**, **Y.**, & **Scialom**, **T.** (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint* arXiv:2307.09288.

Troxler, **A.**, & **Schelldorfer**, **J.** (2024). Actuarial applications of natural language processing using transformers: Case studies for using text features in an actuarial context. *British Actuarial Journal*, **29**(4), 1–37.

Vaswani, **A.**, **Shazeer**, **N.**, **Parmar**, **N.**, **Uszkoreit**, **J.**, **Jones**, **L.**, **Gomez**, **A. N.**, & **Polosukhin**, **I.** (2017). Attention is all you need. In *Advances in neural information processing systems, 30*.

Waibel, **A.**, & **Kai-Fu**, **L.** (Eds.) (1990). *Readings in speech recognition*. Morgan Kaufmann.

Wang, **H.**, **Liu**, **C.**, **Xi**, **N.**, **Qiang**, **Z.**, **Zhao**, **S.**, **Qin**, **B.**, & **Liu**, **T.** (2023). Huatuo: Tuning llama model with chinese medical knowledge. *arXiv preprint* arXiv:2304.06975.

Wei, **J.**, **Wang**, **X.**, **Schuurmans**, **D.**, **Bosma**, **M.**, **Ichter**, **B.**, **Xia**, **F.**, & **Zhou**, **D.** (2022). Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems, 35* (pp. 24824–24837).

Yang, **H.**, **Liu**, **X.-Y.**, & **Wang**, **C. D.** (2023). FinGPT open-source financial large language models. *arXiv preprint* arXiv:2306.06031.

Yang, **Y.**, **Tang**, **Y.**, & **Tam**, **K. Y.** (2023). InvestLM: A large language model for investment using financial domain instruction tuning. *arXiv preprint* arXiv:2309.13064.

Yao, **S.**, **Yu**, **D.**, **Zhao**, **J.**, **Shafran**, **I.**, **Griffiths**, **T. L.**, **Cao**, **Y.**, & **Narasimhan**, **K.** (2023). Tree of thoughts: Deliberate problem solving with large language models. In *Advances in neural information processing systems, 36* (pp. 11809–11822).

Yue, **S.**, **Chen**, **W.**, **Wang**, **S.**, **Li**, **B.**, **Shen**, **C.**, **Liu**, **S.**, & **Zhou**, **Y.** (2023). Disc-lawllm: Fine-tuning large language models for intelligent legal services. *arXiv preprint* arXiv:2309.11325.

Zhao, **W. X.**, **Zhou**, **K.**, **Li**, **J.**, **Tang**, **T.**, **Wang**, **X.**, **Hou**, **Y.**, & **Wen**, **J.-R.** (2023). A survey of large language models. *arXiv preprint* arXiv:2303.18223.

Appendix A. Calculation Dataset

Table A1. Examples for data in calculation dataset

Example	Key	Value
1	Instruction	What is the net retention covered by the excess of loss treaty?
	Input	Under net retained lines clause, the primary insurer writes an industrial risk with a sum insured of 100,000, of which 30,000 is the subject of prior facultative reinsurance
	Output	The net retention covered by the excess of loss treaty is 100,000 – 30,000 = 70,000
2	Instruction	Calculate how liability, premiums and claims are allocated between reinsured and reinsurer
	Input	For a quota share reinsurance treaty, the following key data apply: liability: 6,478,100, premium: 9600, claims: 2,192,200. Treaty limit: 6,947,600. Retention: 20%. Cession under the treaty: 80%
	Output	The liability 6,478,100 is less than the treaty limit 6,947,600. So all the liability is allocated between reinsured and reinsurer. With a retention of 20%, Reinsured: Liability: 0.2*6,478,100 = 1,295,620, premium 0.2*9600 = 1920, claims 0.2*2,192,200 = 438,440. And the cession under treaty is 80%, so reinsurer: liability 0.8*6,478,100 = 5,182,480, premium of reinsurer is 0.8*9600 = 7680, claims of reinsurer is 0.8*2,192,200 = 1,753,760
3	Instruction	How much does the reinsurer have to pay under the excess of loss per person treaty?
	Input	On a business trip in Switzerland, Representative A is killed in a road accident as a passenger in a third-party motor vehicle. A turns out to be covered with the same insurer under three different accident policies: under his personal accident insurance with a death benefit of 1,600,000; the employer has also arranged travel accident insurance in A's favor with a death benefit of 2,100,000; and on top of this, motor passenger personal accident insurance with a death benefit of 440,000 also comes into play, this having been arranged by the owner of the vehicle in which A was killed. Overall, the resultant death benefit under all three policies is 4,140,000. The insurer with which these three policies are in force has a WXL cover per person of 7,800,000 xs 1,000,000

https://doi.org/10.1017/S1748499525100079 Published online by Cambridge University Press

Table A1. Continued

Example	Key	Value
	Output	The total death benefit = 1,600,000 + 2,100,000 + 440,000 = 4,140,000 for this person. The deductible is 1,000,000, and the upper limit = 1,000,000 + 7,800,000 = 8,800,000. It exceeds the deductible 1,000,000 but not the upper limit, therefore the payment = 4,140,000 – 1,000,000 = 3,140,000
4	Instruction	How much does each layer of reinsurer have to pay under excess of loss treaty with multiple layers?
	Input	An excess of loss has total cover of 8,000,000 xs 1,000,000. The upper limit of cover is therefore 9,000,000. The total cover is divided up into three layers, which are described as follows: 1st layer: 3,000,000 per risk xs 1,000,000 per risk. 2nd layer: 1,000,000 per risk xs 4,000,000 per risk. 3rd layer: 4,000,000 per risk xs 5,000,000 per risk. During the treaty period, a loss of 5,000,000 occurred
	Output	For the 1st layer, the deductible is 1,000,000 and the upper limit is 1,000,000 + 3,000,000 = 4,000,000. For the 2nd layer, the deductible is 4,000,000 and the upper limit is 4,000,000 + 1,000,000 = 5,000,000. For the 3rd layer, the deductible is 5,000,000 and the upper limit is 5,000,000 + 4,000,000 = 9,000,000. The loss is 5,000,000. For the 1st layer, the loss exceeds the deductible and the upper limit. So the 1st reinsurer pays 4,000,000 – 1,000,000 = 3,000,000. For the 2nd layer, the loss exceeds the deductible but not the upper limit. So the 2nd reinsurer pays 5,000,000 – 4,000,000 = 1,000,000. For the 3nd layer, the loss does not exceed the deductible. So the 3rd reinsurer pays 0

Appendix B. Knowledge Dataset

Table B1. Examples for data in knowledge dataset

Example	Key	Value
1	Instruction	What are the principal methods to distribute risks?
	Input	
	Output	Risks can be distributed in different ways and at different levels. The principal methods are coinsurance and reinsurance
2	Instruction	Give a definition of stop loss reinsurance
	Input	
	Output	With the stop loss treaty, the reinsurer undertakes to assume that amount of an aggregate loss per unit of time (year) which exceeds the stop loss point(deductible) fixed. The reinsurer's liability is limited by the stop loss cover. Adding the stop loss point to the stop loss cover gives the upper limit of cover of ceiling for the stop loss reinsurance treaty
3	Instruction	When and why is surplus reinsurance used?
	Input	
	Output	Surplus reinsurance is mainly used in property classes of insurance where a claim is related to the sum insured and is dependent on it, especially in classes of business that are prone to total or almost total losses. It is suitable for reinsuring an insurance portfolio with varying sizes of sums insured or limits of liability, making it more balanced for the reinsurer's retention. It is most advantageous for the reinsured when the results of small and medium-sized insurance contracts are better than those of larger policies, and the reinsurer's prices reflect this imbalance in the conditions of the reinsurance treaty

https://doi.org/10.1017/S1748499525100079 Published online by Cambridge University Press

Appendix C. One-Shot Prompt

Table C1. Examples for one-shot prompts

Example	Key	Value
1	Instruction	What is the net retention covered by the excess of loss treaty?
	Input	Here is the background: Under net retained lines clause, the primary insurer writes an industrial risk with a sum insured of 980,000,000, of which 490,000,000 is the subject of prior facultative reinsurance **Here is an example to follow:** **Background: Under net retained lines clause, the primary insurer writes an industrial risk with a sum insured of 74,000,000, of which 44,000,000 is the subject of prior facultative reinsurance** **Answer: The net retention covered by the excess of loss treaty is 74,000,000 – 44,000,000 = 30,000,000**
	Output	The net retention covered by the excess of loss treaty is 980,000,000 – 490,000,000 = 490,000,000
2	Instruction	Follow the example and solve the question: How much loss does the reinsurer need to bear under the stop loss treaty?
	Input	Here is the background: In year 2018, an insurer's premium income for its portfolio of hail policies amounts to 20,000,000. The insurer has a stop loss cover of 130% xs 100% of the premium income for the cover period. During the year, several hailstorms occur in areas where the primary insurer's portfolio has its main focus. The aggregated losses for the year amount to 45,000,000 **Here is an example to follow:** **Background: In year 2018, an insurer's premium income for its portfolio of hail policies amounts to 30,000,000. The insurer has a stop loss cover of 50% xs 90% of the premium income for the cover period. During the year, several hailstorms occur in areas where the primary insurer's portfolio has its main focus. The aggregated losses for the year amount to 63,000,000** **Answer: The total loss is 63,000,000, and the annual premium is 30,000,000. So the loss is 210.0% of annual premium. The deductible is 90% of premium income, the upper limit is 90% + 50% = 140%, the loss exceeds the upper limit, so the stop loss treaty should assume 140% – 90% = 50.0% of annual premium, which is 50.0% * 30,000,000 = 15,000,000**
	Output	The total loss is 45,000,000, and the annual premium is 20,000,000. So the loss is 225.0% of annual premium. The deductible is 100% of premium income, the upper limit is 100% + 130% = 230%, so the stop loss treaty should assume 225.0% – 100% = 125.0% of annual premium, which is 125.0% * 20,000,000 = 25,000,000

Appendix D. Marking Examples

Table D1. Marking examples of partial deduction due to incorrect calculation

Example	Key	Value
1	Instruction	Calculate the max reinsurance cover under basic policy
	Input	The reinsured's deductible is 6000, the sum insured under the basic policy is 600,000
	Output	This is a non-proportional facultative reinsurance on the basic policy. The sum insured is 600,000, the deductible is 6000, therefore the maximum reinsurance cover is 600,000 – 6000 = 594,000
	Model output	"Max reinsurance cover = 600,000 – 6000 = 540,000. ### Explanation: The reinsured's deductible is 6000, the sum insured under the basic policy is 600,000, and the max reinsurance cover is 540,000"
	Score deduction	$1 \times \frac{1}{2 \times 1} = 0.5$

https://doi.org/10.1017/S1748499525100079 Published online by Cambridge University Press

Table D1. continued

Example	Key	Value
2	Instruction	Calculate how liability, premiums and claims are are allocated between reinsured and reinsurer
	Input	For a quota share reinsurance treaty, the following key data apply: Treaty business: Policy K: liability: 1,343,900, premium: 2400, claims: 176,500. Treaty limit: 5,213,000 per policy. Retention: 20%. Cession under the treaty: 80%
	Output	The liability 1,343,900 is less than the treaty limit 5,213,000. So all the liability is allocated between reinsured and reinsurer. With a retention of 20%, Reinsured: Liability: 0.2*1,343,900 = 268,780, premium 0.2*2400 = 480, claims 0.2*176,500 = 35,300. And the cession under treaty is 80%, so reinsurer: liability 0.8*1,343,900 = 1,075,120, premium of reinsurer is 0.8*2400 = 1920, claims of reinuser is 0.8*176,500 = 141,200
	Model output	For a quota share reinsurance treaty, the following key data apply: Treaty business: Policy K: liability: 1,343,900, premium: 2400, claims: 176,500. Treaty limit: 5,213,000 per policy. Retention: 20%. Cession under the treaty: 80%. Reinsured's liability: 20% x 1,343,900 = 268,780. Reinsured's premiums: 20% x 2400 = 480. Reinsured's claims: 20% x 176,500 = 353,000. Reinsurer's liability: 80% x 1,343,900 = 1,072,720. Reinsurer's premiums: 80% x 2400 = 1920. Reinsurer's claims: 80% x 176,500 = 144,400
	Score deduction	$2 \times \frac{1}{2\times 6} = 0.167$

Table D2. Marking examples of full deduction due to lack of reasoning capabilities

Example	Key	Value
1	Instruction	What is the amount payable under the CatXL with an annual limit?
	Input	"An insurance company's motor physical damage portfolio, which is covered by a CatXL cover of 4,000,000 xs 2,000,000 per event with an annual limit of 6,000,000, is affected by three major losses in a year: 1. in April, by a flood loss of 5,800,000 2. in June by a hail loss of 5,400,000 3. in August, by another hail loss, this time of 3,300,000"
	Output	"1. For the first event, the loss is 5,800,000, the reinsurer needs to pay 5,800,000 – 2,000,000 = 3,800,000, and there is 6,000,000 – 3,800,000 = 2,200,000 left in the annual limit 2. For the second event, the loss is 5,400,000, the reinsurer needs to pay 5,400,000 – 2,000,000 = 3,400,000, which exceeds the amount left in annual limit. So the reinsurer only needs to pay the amount of the annual limit, which is 2,200,000 3. For the third event, there is 0 left in the annual limit, so the reinsurer does not need to pay"
	Model output	For the first loss, the deductible is 2,000,000 and the loss is 5,800,000, so the payment is 5,800,000 – 2,000,000 = 3,800,000. For the second loss, the deductible is 2,000,000 and the loss is 5,400,000, so the payment is 5,400,000 – 2,000,000 = 3,400,000. For the third loss, the deductible is 2,000,000 and the loss is 3,300,000, so the payment is 3,300,000 – 2,000,000 = 1,300,000. Therefore, the total payment is 3,800,000 + 3,400,000 + 1,300,000 = 8,500,000
	Score deduction	$1 \times \frac{1}{3} = 0.333$

https://doi.org/10.1017/S1748499525100079 Published online by Cambridge University Press

Table D2. continued

Example	Key	Value
2	Instruction	How much loss does the reinsurer need to bear under the stop loss treaty?
	Input	In year 2018, an insurer's premium income for its portfolio of hail policies amounts to 310,000,000. The insurer has a stop loss cover of 130% xs 120% of the premium income for the cover period. During the year, several hailstorms occur in areas where the primary insurer's portfolio has its main focus. The aggregated losses for the year amount to 350,000,000
	Output	The total loss is 350,000,000, and the annual premium is 310,000,000. So the loss is 112.9% of annual premium. The deductible is 120% of premium income, the loss does not exceed the deductible. So the reinsurer does not need to make any payment
	Model output	The primary insurer's premium income for the year 2018 is 310,000,000. The stop loss cover is 130% xs 120% of the premium income. The aggregated loss is 350,000,000. The loss to the primary insurer is 350,000,000 – 310,000,000 = 40,000,000
	Score deduction	$1 \times \frac{1}{1} = 1$

Appendix E. CoT Test Training Data

Table E1. An example of training data for CoT test

Key	Value
Instruction	What is the amount payable under the CatXL with an annual limit?
Input	An insurance company's motor physical damage portfolio, which is covered by a CatXL cover of 1,000,000 xs 2,000,000 per event with an annual limit of 7,000,000, is affected by three major losses in a year: 1. in April, by a flood loss of 2,100,000 2. in June by a hail loss of 2,300,000 3. in August, by another hail loss, this time of 2,400,000
Output (only numerical results are included in the fine-tuning data)	1. For the first event, the loss: 2,100,000, payment: 100,000, annual limit left: 6,900,000 2. For the second event, the loss: 2,300,000, payment: 300,000, annual limit left: 6,600,000 3. For the third event, the loss: 2,400,000, payment: 400,000
Output (text expressions and numerical results are contained in the fine-tuning data)	1. For the first event, the loss is 2,100,000, the reinsurer needs to pay 100,000, and there is 6,900,000 left in the annual limit 2. For the second event, the loss is 2,300,000, the reinsurer needs to pay 300,000, and there is 6,600,000 left in the annual limit 3. For the third event, the loss is 2,400,000, the reinsurer needs to pay 400,000
Output (text expressions, numerical results, and math formula achieving results are all provided in the fine-tuning data)	1. For the first event, the loss is 2,100,000, the reinsurer needs to pay 2,100,000 – 2,000,000 = 100,000, and there is 7,000,000 – 100,000 = 6,900,000 left in the annual limit 2. For the second event, the loss is 2,300,000, the reinsurer needs to pay 2,300,000 – 2,000,000 = 300,000, and there is 7,000,000 – 300,000 = 6,700,000 left in the annual limit 3. For the third event, the loss is 2,400,000, the reinsurer needs to pay 2,400,000 – 2,000,000 = 400,000

Cite this article: Hao Y, Tian X, Zhao H and Baldassarre L (2026). Utilizing large language models (LLMs) for quantitative reasoning-intensive tasks within the (re)insurance sector, *Annals of Actuarial Science*, **20**, 74–95. https://doi.org/10.1017/S1748499525100079

https://doi.org/10.1017/S1748499525100079 Published online by Cambridge University Press