Towards Trustworthy Large Language Models in Industry Domains

INF-Team

July 04, 2024

Abstract

This report addresses the challenges and strategies for mitigating hallucinations in large language models (LLMs), particularly in domain-specific applications. Hallucinations refer to the generation of unrealistic or illogical outputs by LLMs. We explore several methods to reduce hallucinations, including using high-quality domain-specific data for training, ensuring that the model stays up-to-date with new knowledge, and employing alignment techniques to ensure that the LLM adheres to human instructions. A key proposition is the adoption of neuro-symbolic systems, which combine large-scale deep learning models with symbolic AI. These systems leverage neural networks for fast "black box" probabilistic predictions while also enabling "white box" logical reasoning. The integration of these approaches represents a significant technical direction for future artificial general intelligence and provides a "gray box" approach to developing trustworthy LLMs for industrial applications. This dual capability enhances logical reasoning and improves explainability. In addition, we detail our efforts to construct domain-specific LLMs for finance and healthcare. Using anti-hallucination strategies, our finance LLM outperforms GPT-4 on the CFA tests, while our healthcare LLM ranks first on the public MedBench competition leaderboard.

1 Introduction

With the unprecedented development of large language models (LLMs), LLMs are used to improve communication, generate creative text formats, translate languages effectively, and even assist scientific research. However, LLMs are notorious for yielding unreliable outputs, which greatly hinder their application in real-world tasks, especially for high-stakes decision-making applications in industries such as healthcare, asset investment, criminal justice, and other domains.

One of the key challenges is the "hallucination" problem, whereby LLMs may output content that seems reasonable but is, in fact, incorrect or illogical. As an inherent limitation of LLM [51], the hallucination phenomenon is inevitable. Hallucinations can be divided into two categories [13, 14]. One is the factuality hallucination, where the generated content is inconsistent with the facts of the real world and contains unexpected fictitious concepts and plots. The other is faithfulness hallucination, where the generated content is inconsistent with the input instructions and logic. Both categories pose serious obstacles to ensuring the accuracy and reliability of model outputs in industry applications.

Domain-specific LLMs focus on understanding and responding to a particular field or industry, e.g., finance and healthcare, aiming to resolve domain-specific tasks as highly trained professionals. For real-world industries where domain-specific LLMs may play crucial roles in decision-making, being trustworthy becomes more demanding. Therefore, domain-specific LLMs must address the major limitations we discussed above, i.e. hallucinations.

Despite the massive training data in LLM that covers a wide range of topics, a considerable portion of real-world domain knowledge is long-tail, and the scarcity of domain data may contribute to hallucinations. In specific industry domains, general LLMs may lack professional knowledge and fail to follow technical instructions due to insufficient domain data at the training stage.

To mitigate factuality hallucination, an effective approach is to address the issue of data scarcity in training. Feasible solutions include curating high-quality factual data specifically for the domain, developing automatic data cleaning and selection techniques, and designing execution engines for high-quality synthetic data. The massive domain-specific data can greatly reduce the model's tendency to fabricate information after training.

Continuous training with high-quality domain-specific data helps LLMs acquire domain knowledge to understand technical nuances in context and instructions, thereby alleviating faithful hallucinations as well. To further reduce faithfulness hallucination and improve productivity, we need to resort to alignment techniques to ensure that LLMs actively cooperate with professional instructions to achieve specific goals. Meanwhile, we design reward systems that motivate LLMs to behave in a way that is consistent with human values and employ reinforcement learning to learn from preference feedback in a human-in-the-loop system that provides safety guidance and supervision.

As a black-box model, LLMs cannot always explain their outputs correctly. The presence of hallucinations is an intrinsic obstacle that prevents the continuous yielding of reliable explanations, even if prompting LLMs to explain step by step. Research in the field of interpretability for LLMs is still in development. For example, dictionary learning from Anthropic is a tool to understand the correspondence between the model components and the particular inputs [3]. The method has improved analytic capacity to break down the complexity of LLMs into more understandable features. However, exploring all internal features learned by LLMs during training is still cost-prohibitive, and effectively manipulating specific features for predictably superior behavior. Lack of explainability is another challenge for LLMs in gaining trust in high-stakes applications where a transparent decision process is critical.

To improve explainability in the behavior of LLMs, we can either dive into attention mechanisms by tools such as dictionary learning to analyze the generation process, or prompt LLMs to carry out refles and verification step by step and assign confidence scores to its responses [7]. It is also possible to produce counterfactual explanations by providing alternative scenarios where the output of LLMs would change, offering insights into its reasoning process. While these methods allow users to assess the reliability of the information, the interpretations are contaminated by the intrinsic hallucination within LLMs. To break through innate limitations, we have introduced neural symbolic systems to assist LLMs in gaining explainability and transparency in content generation. Neural symbolic systems are emerging in AI that aim to combine the strengths of two different AI techniques, i.e., deep learning and symbolic AI. LLMs are very successful exemplars of deep learning, which excels at learning from vast amounts of data and generating creative text formats but struggles with tasks requiring reasoning, logic, and explainability. Symbolic AI uses symbols and rules to represent knowledge and excels in logical reasoning and explainability. However, it can be less efficient in learning from data. To bridge this gap by integrating both approaches, we leverage the reading comprehension capabilities of LLMs to process raw data and generate an initial understanding. The preliminary instances are then passed to our in-house symbolic reasoning engine that performs reasoning on domain-specific causal graphs to make decisions. The final output may combine the multiple interactive results from both modules. The homemade symbolic reasoning engine can visualize all feasible reasoning paths, offering more logical and explainable outputs. We name this proposal as a unique "gray box" approach to trustworthy LLMs in industry domains.

The rest of the content is organized as follows. In Section 2, we detail our approach to implementing trustworthy domain-specific LLMs, including high-quality data collection, alignment techniques, and neural symbolic computation. In Sections 3 and 4, we introduce two domain-specific LLMs, healthcare and finance, respectively. Our approach to trustworthy domain-sepcific LLMs is compatible with any open source foundation model. To demonstrate the feasibility, we use a homemade 34B foundation model to develop our healthcare LLM, and choose the open-source Qwen2-72B base model for continuous training and instruction alignment to build our finance LLM. In Section 5, we conclude and then discuss some directions for future work.

2 Methodology Overview

In this section, we present our methodology to construct trustworthy LLMs in industry domains, which consists of three parts including high-quality data preparation, alignment techniques, and neuro-symbolic computing techniques.

2.1 High-quality Data Preparation

2.1.1 General Data Processing Pipeline

High-quality training data are essential for effective large language model (LLM) training. To achieve this, we have amassed a substantial dataset. The primary sources of text data include Common Crawl, Wikipedia, books, academic papers, journals, patents, news articles, and educational resources for K-12. For code data, the main sources are GitHub and Stack Overflow. Following data collection, we perform data cleansing, which involves three main steps: filtering, deduplication, and selection. The overall process is illustrated in Fig. 1.

Filtering: For the filtering process, we employed heuristic rules to filter the text, which helps avoid selection bias. These heuristic rules allow us to eliminate low-quality data effectively. Different rules are applied to different types of text. The filtering primarily focuses on removing duplicate texts using n-gram repetition detection and sentence-level detection. Additionally, we created a list of sensitive words and removed any documents that contained those words and personal identifiable information.

Deduplication: Deduplication includes fuzzy deduplication and exact deduplication [15]. For fuzzy deduplication, We employ Minhash-LSH for approximate deduplication. The process involves several steps: (1) standardize the text, split the text into sequences using text segmentation, and apply N-Gram processing to the sequences; (2) compute Minhash values and compress them into a set of bucketed hash values using Locality Sensitive Hashing (LSH). (3) perform the approximate deduplication by the hashes. Then, we utilize a suffix array algorithm for exact deduplication. This method includes: (1) dividing files according to memory limitations; (2) loading them into memory to compute the suffix array; (3) identifying duplicate intervals and deleting the entire document exceeding a predefined duplication threshold (to maintain text integrity). This step requires substantial memory and is therefore performed at last.

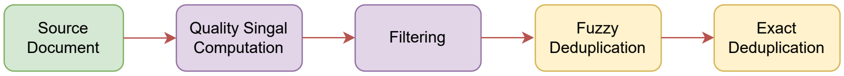


Figure 1: The figure illustrates the primary workflow of our data-cleaning process. The purple sections indicate the filtering stages, while the yellow sections represent the deduplication process.

2.1.2 Recalling High Quality Data from Common Crawl

The Common Crawl (CC) dataset contains an immense collection of web pages. Traditionally, our approach to processing CC data has been limited to filtering and deduplication, preventing more advanced analysis of this extensive dataset. However, to identify reliable and high-quality data across various fields, we need a more sophisticated processing method. We propose a fine-grained division strategy for CC data. Initially, we segment the URLs in all snapshots of the CC dataset by base URL (e.g., www.google.com is considered a base URL). We count the occurrences of each base URL and then rank them in descending order. Our findings indicate that the top 2 million base URLs account for approximately 65% of the entire CC dataset. Therefore, we believe that annotating these base URLs with their type, topic, and language will provide valuable information. This method allows for a preliminary fine-grained segmentation of CC data, although it may introduce some inaccuracies.

High-quality data plays a crucial role in the capabilities of general models [54] [55], but related corpora are extremely scarce. Therefore, we employ a method similar to the data-recalling mechanism in DeepSeek-Math [34]. This method recalls high-quality data from CommonCrawl (CC), focusing on three domains: code, math, and Wiki. The code and math data enhance the model's reasoning capabilities, while Wikidata enriches the model's knowledge. This process includes seed acquisition, URL aggregation, and fastText-based recall, as shown in Fig. 2.

Seed collection: For the mathematical and code data, We choose OpenWebMath [27], StackOverflow pages, and Wikipedia pages as our English initial seeds. Public available Chinese training datasets are limited to mathematics and code. Reference to AutoMathText [59], we prompt a base LLM to autonomously annotate data for topic relevance and educational value, subsequently retrieving the top 50K entries as Chinese initial seeds. For knowledge, we employ an LLM to annotate Wikipedia data with educational scores and subsequently train a classifier. We then collect 50,000 high-quality seeds from wiki sources such as Wikipedia. We train a fastText model using collected seed data as the positive samples and random CC documents as the negative samples.

URL aggregation: Due to the insufficient diversity of the seeds, many target data remain uncollected after the first round of recall. We further enhance the diversity of the seeds through URL aggregation. We manually annotate sub URLs (e.g., cloud.tencent.com/developer) from domains where over 10% of the pages are hit and incorporate the uncollected samples into the seed set.

Iterative recall: After collecting more diverse seeds, we retrain the fastText model and further recall more target webpages. We repeat the process of URL aggregation and fastText retraining until over 98% of the recall results have been collected.

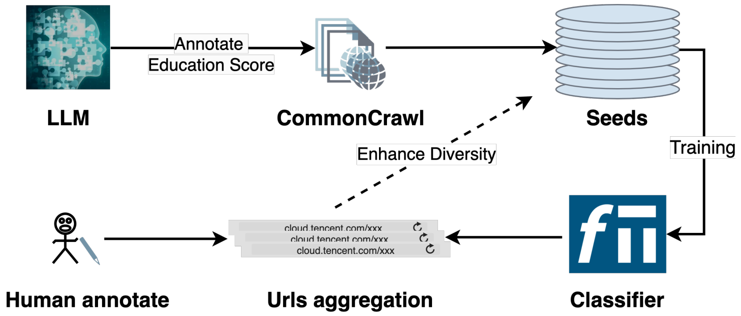


Figure 2: The illustration of high quality data recall.

2.2 Alignment

In the realm of LLMs, Reinforcement Learning from Human Feedback (RLHF) emerges as a pivotal strategy to counteract hallucinatory outputs. This section elaborates on how our alignment technique addresses two distinct types of hallucinations introduced earlier, where the central of the approach is the reward design.

2.2.1 Data for Instruction Alignment

The initial dataset for supervised fine-tuning (SFT) is compiled from various open-source instruction tuning projects, including Alpaca, Dolly, Wizard-LLMs, COIG, and others. From our systematic experimentation, we identify three critical factors that contribute to trustworthy instruction during the SFT phase: dataset diversity, dataset quality, and dataset complexity. For dataset diversity, we emphasize incorporating a mix of domains and ensuring a variety of instructional formats. This approach aims to enhance the model's ability to generalize across different instructional tasks and achieve robust semantic understanding. Dataset quality is paramount for generating trustworthy outputs. To enhance the quality of our instruction dataset, we engage human annotators to rigorously review and label the data, ensuring its accuracy and consistency. Additionally, we have developed a dataset curation pipeline that utilizes other open-source LLM to verify data quality, aiming for an accuracy threshold of over 99.9%. Regarding dataset complexity, we recognize the importance of including complex and challenging instructions, particularly for tasks that involve compound instructions. To this end, we also involve human annotators in augmenting the instructions and labeling the expected outputs, further enriching our instruction complexity.

To further facilitate trust-worthy generation, particularly in providing factual answers, we have incorporated a diverse instruction data set that includes 6 function calls. We specifically trained the LLMs to better handle reasoning and topicality queries by utilizing multiple tools, focusing on improving accuracy and completeness. To this end, we have developed several tools to support this functionality: a search engine, a mathematical calculation tool, a code interpreter, and a symbolic logic engine. The use of these tools is carefully managed through system prompts to optimize performance and ensure relevant application. Overall, the initial instruction dataset used for SFT is around 200K samples.

2.2.2 Reward Design in Faithfulness Hallucination

A pivotal step to combat faithfulness hallucination is to integrate the reward models, which act as proxies for human feedback. These models are instrumental in ensuring that the LLM's responses adhere to human ethical standards and preferences, covering aspects like safety, helpfulness, and mathematical reasoning acumen. A critical component of RLHF is the development of a robust and varied reward model dataset that reflects the vast spectrum of human preferences.

To construct this dataset, we employ both human annotation and synthetic data generation techniques. During the human annotation phase, annotators are tasked with creating prompts that span broad topics such as safety and logical reasoning, engage with specialized subjects like poetic composition, or address specific procedural instructions, such as generating outputs in JSON format. For synthetic data generation, we leverage our model to substitute human annotators, enabling it to generate prompts by altering and extrapolating from web-based texts, including the rewriting of questions or the extraction of content from encyclopedia-like documents.

Unlike the common pairwise comparison of two model responses, as seen in RLHF methods like DPO [31] and PPO [33], we ask annotators to score each response individually on a scale from 0-10, following predefined guidelines. In instances where responses receive equivalent scores, annotators are further asked to determine which response aligns better with the task requirements.

Our reward model is composed of building a linear projection head on top of the base model. The objective function includes the squared loss between the output value and the score, plus a pairwise loss that is applied only when the ground truth scores are tied. Reflecting on the trade-offs between safety and helpfulness as documented in previous studies [1, 39], we train separate models for safety and helpfulness to fine-tune our reward alignment strategy.

2.2.3 Reward Design in Factuality Hallucination

A fundamental aspect of mitigating factuality hallucinations in LLMs involves training the model to express epistemic uncertainty. When unsure about an answer, the model is trained to respond with *"I do not know"*. However, accurately estimating epistemic uncertainty poses significant challenges, particularly in the context of LLMs, since the epistemic uncertainty stems from various factors such as insufficient training data or limitations in model capacity [52].

In our approach, we concentrate on knowledge-intensive domains, such as question-answering (QA) on common knowledge, where it is feasible to construct a proxy for epistemic uncertainty as that in [18, 56].

Initially, we employ few-shot learning techniques to the base model for QA tasks. The model's accuracy in these preliminary responses serves as an indicator of whether it possesses the requisite knowledge for the queried topic. During the subsequent RLHF stage, we re-present the same questions to the model. A reward system is designed as follows: providing a negative reward when the model fabricates answers to questions it lacks knowledge on, and a positive reward when it appropriately expresses uncertainty with phrases like *"I do not know"* or *"I am not sure about that"*. This approach strategically reinforces the expression of epistemic uncertainty, enhancing the model's reliability by discouraging the generation of unverified information.

2.2.4 RLHF Training

Given the reward system, the subsequent step is to train our LLM with reinforcement learning algorithm. We leverage the standard off-policy REINFORCE [44] method rather than PPO or DPO-like methods. We update the parameters θ of our LLM in the direction of

$$\begin{aligned}\mathbb{E}_{q\sim q_{pool}}\mathbb{E}_{a\sim\pi_{ref}} \min\left(\frac{\pi_\theta}{\pi_{ref}}, \rho\right) \nabla \log \pi_\theta(a|q)\big(r(a|q) - \bar{r}(q)\big) \\ - \lambda_1 \nabla \mathbb{E}_{q\sim q_{pool}} KL(\pi_{ref}||\pi_\theta) - \lambda_2 \nabla \mathbb{E}_{q\sim q_{sft}} KL(\pi_{sft}||\pi_\theta),\end{aligned}$$

where π_θ is the current LLM that we aim to optimize, q is a question sampled from our question pool, and a is the corresponding answer sampled from the reference policy π_{ref}. The term $\frac{\pi_\theta}{\pi_{ref}}$ represents the importance sampling ratio to ensure above function is an unbiased estimator with respect to the corresponding on-policy policy gradient. The ratio is typically clipped to reduce variance, particularly when π_θ deviates significantly from π_{ref} and the sequence is long. Additionally, we include two KL divergence terms, i.e., $KL(\pi_{ref}||\pi_\theta)$ and $KL(\pi_{sft}||\pi_\theta)$, to ensure that the learned abilities do not degenerate during the RLHF stage. The reward $r(a|q)$ is obtained through either the reward model or verifiers such as ground truth label matching in GSM8K and unit tests in Python programs. To reduce the variance of the policy gradient, $\bar{r}(q)$, known as the baseline, is estimated by averaging the scores of the responses. Generally, we sample 10 responses for each question and use $\rho = 1.0$, $\lambda_1 = 0.2$, and $\lambda_2 = 1$ in our objective function.

We perform several iterations of the off-policy REINFORCE, transitioning the policy from π_0 (the SFT model) to π_1, π_2, and so on. In each iteration i, we set $\pi_{ref} = \pi_{i-1}$. This multi-iteration strategy is necessary to progressively enhance both the helpfulness and safety of the model.

2.2.5 Evaluation of RLHF

To assess the safety and helpfulness post-RLHF, we devised rigorous internal test sets, each containing over 2000+ tailored questions. The safety test set covers various categories, including but not limited to illegal activity, insults, discrimination, sensitive topics, and prompt leaking. The helpfulness test set includes seven macro topics and eighty micro topics, combined with several specific instruction-following tasks. Annotators evaluate safety according to our guidelines for both SFT and RLHF models individually, while helpfulness is assessed through side-by-side evaluations. After three RLHF iterations, our model exhibited a decrease in toxicity from 5.8% to 4.9% and an improvement in helpfulness win rate from 50% to 56% compared to the SFT model. Moreover, mathematical reasoning capability, as indicated by pass@1 rate in GSM8K, rose from 77% to 85%. Evaluating RLHF in the context of addressing factual hallucinations presents distinctive challenges, primarily due to the difficulties associated with delineating the base model's knowledge boundaries. Our observations indicate that the model more frequently expresses uncertainty in complex tasks than in simpler ones, suggesting a nuanced understanding of its own knowledge limits.

2.3 Neuro-symbolic Computing

Here, we show our neuro-symbolic solutions for faithfulness hallucination (FH). FH is inherent in LLMs: as a neural network, LLMs cannot ensure that the generated sentences are logically correct. Even for a simple calculation problem, LLMs will produce incorrect answers like humans. The recently proposed chain-of-thought (COT) methods break the solving process into several substeps. Despite its effectiveness, LLMs may still make mistakes in each reasoning step [17]. Some methods employ symbolic engines (SEs) in the LLMs to improve the reasoning performance. However, they use SEs as an external tool [25, 26] or a logical planner [48].

Logical reasoning is a long-standing problem in machine learning, presenting two essential difficulties for the models. The first difficulty is in performing the step-wise reasoning correctly, that is, using the available premises to derive the one-step conclusion correctly. The other is in performing effective searching over massive reasoning paths, which can be challenging for complicated cases. On the other hand, the logical reasoning problem itself enjoys two features. It is deterministic. Once the problem is formalized into formal languages, such as Lean [6], Prolog [24], Prover9 [21], etc., each reasoning step can be verified. Consequently, we can always determine when the reasoning reaches the answer and verify whether the answer is correct. In this sense, performing logical reasoning is like playing Go:

> “Learning to prove theorems is somewhat analogous to learning to play Go: both offer an automated way of determining success (the game of Go is a miniature formal system), and both offer an automated way for generating new data via self play-type approaches. ” – OpenAI [28].

Another characteristic of logical reasoning is that logical searching can be performed automatically when searching space is finite, e.g., problems with Horn clauses, and constraint logical programming. This means that we can solve some logical reasoning problems solely using an external SE. For the problems that we cannot exhaustively explore, we leverage the power of LLMs to guide the search direction, analogy to the actors in AlphaGo.

These features of logical reasoning identify two levels of logical reasoning with LLMs:

- Level 1 (L1): Formalizing the problem into formal languages so that we can perform strict reasoning using the SEs.
- Level 2 (L2): Upon level 1, guiding the reasoning direction within and for the SEs.

L1 reasoning requires that the LLMs can translate the problem in the natural language into formal logical languages, which we refer to as NL2FL. More specifically, LLMs should be able to capture the relevant logical rules and facts in the natural language content and translate them into the SE languages, e.g., Prolog and Lean. LLMs should also be able to complement common sense rules implicit in the content, such as the definition of the relation in the family. L2 reasoning requires that LLMs are experts in certain reasoning scenarios, such as game playing, business domains, and math proving. L2 is more challenging than L1 as the search space may be extremely large, e.g., math proving [46].

Our solutions follow the vein of these two levels and currently focus mainly on L1.

- First, we leverage the semantic ability of LLM to automatically build casual decision diagrams in certain (professional) fields from a few text inputs of domain knowledge for cold starting, i.e., NL2FL.
- Then, we propose an internal SE to complete the reasoning steps based on the causal decision diagram.
- It can handle uncertainty in the formalization process by applying message propagation technology to complete probabilistic reasoning.
- It can also derive explicit reasoning steps for visualization to make the entire reasoning process explainable.
- Moreover, along with using SE, we can gradually accumulate a large amount of high-quality synthetic data for LLM training.

For L2 reasoning, we aim to explore feasible direction paths via reinforcement learning and continuously improve LLM's reasoning ability.

2.3.1 Natural Language to Formal Language

To use SEs, we need to translate the problem in natural language into formal languages. We choose Prolog syntax as our formal language due to its effectiveness and popularity. Prolog, short for ”Programming in Logic,” is a high-level logical programming (LP) language associated primarily with artificial intelligence and computational linguistics. Developed in the early 1970s, Prolog is a declarative language, meaning that the logic of the program is expressed in terms of relations, represented as facts and rules. Prolog's strength lies in its ability to express complex logical relationships and its use in fields where symbolic reasoning and pattern matching are essential.

Unlike imperative programming languages such as Python, where we must specify the steps to solve a problem, logical programming like Prolog does not require this. Instead, we define the problem and the desired goal, and the LPs automatically determine how to solve it. This is useful when we apply LLM in specific domains: LLMs do not need to solve every task by themselves, instead, they can delegate the difficulty to the LPs and let LPs solve the problem for LLMs. For instance, in a specific medical task where the goal is to classify the severity degree of an examination report, we can translate the medical rules and the examination into formal premises, with which SE can automatically derive the answer for the query.

With a strong ability to understand semantics, LLMs bridge the gap between unstructured natural language and formal language. To further improve the LLM's ability of NL2FL, we collect a large amount of Prolog code from swish.swi-prolog.org and translate the Prolog code into natural language using GPT-4. During the process, we deduplicate similar codes and remove those not suitable for learning NL2FL. Then we used the paired samples to fine-tune the LLM. The trained LLM is used to automatically build the causal decision diagrams for medical and financial domains. With the causal decision diagrams, the LLM then extracts the facts from the input text. Table 1 illustrates an example of NL2FL in a medical case. LLM extracts the rules from a natural language textbook and expresses them in Prolog. To determine whether a person is anemic, LLMs take medical examination as input and extract the "gender" and the "hb" values if available.

Natural language	**Prolog**
A person is considered anemic if their hemoglobin level is less than 120 g/L for males or less than 110 g/L for females.	anemic :- gender(male), hb(X), X < 120; gender(female), hb(X), X < 110.
一个人如果其血红蛋白水平低于120 g/L（男性）或低于110 g/L（女性），则被认为是贫血。	贫血 :- 性别(男性), 血红蛋白(X), X < 120；性别(女性), 血红蛋白(X), X < 110。

Table 1: An example of NL2FL pair.

2.3.2 Ponens: A Novel Symbolic Engine

The SE takes the extracted facts and the casual decision diagrams as input and builds a knowledge base instantly. For every query, SE will automatically derive the answer through logical reasoning. The traditional Prolog SEs, for example, SWI-Prolog [43], are powerful but face difficulties in several aspects: (1) they cannot handle cycle rules, (2) they perform costly exhaustive searches to reach the answer, (3) they cannot derive the exact proofs without additional manipulations.

Therefore, we propose a novel symbolic engine, Ponens, which supports Prolog syntax using Python as the backend. Ponens is named after "modus ponens", a basic inference rule in Logic. With Python, Ponens has better scalability in various scenarios than traditional Prolog engines. For instance, Ponens is directly compatible with powerful libraries in Python, such as Numpy, SymPy, and even LLM calls. Therefore, we can integrate off-the-shelf functions to implement various predicates easily.

2.3.3 Handling Uncertainty via Variational Inference

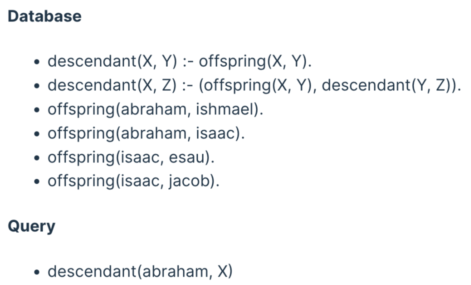
Note that Ponens is not another Prolog engine, it uses the same syntax as Prolog but not the same reasoning strategy. With a given query, Prolog performs automatic reasoning via a deterministic search over the rules. It gradually checks whether the goal matches the rule conclusion and whether the condition can be satisfied. This deterministic search is problematic for handling uncertainty in the natural language. Natural language is semantically vague but the reasoning is logically sparse. How to handle the uncertainty in natural language remains a challenging problem. Several traditional approaches have extended Prolog in such probabilistic logic programming settings, e.g., ProbLog [12, 30]. However, these methods typically require building an arithmetic circuit [35] for each query and the treewidth may grow exponentially in the number of entities. To mitigate these problems, we adopt the variational inference [50] in the Ponens. By variational inference, reasoning with uncertainty becomes much easier: we circumvent the essential difficulty in exact inference via approximate inference. More specifically, LLM extracts the facts along with the estimated probability, i.e., the generation probability of the facts. The estimated probability is then fed into the engine as the initial marginal probability in the variational inference stage.

To make the paper self-contained, we briefly describe here how the variational inference is applied in reasoning with uncertainty. First, we build a Markov logic network to model the likelihood. The facts from the LLMs are treated as the observed variables and the others (including the goals) are treated as unobserved variables to infer. For a specific assignment of unobserved facts, MLN estimates how many rules are satisfied. The inference goal is to find the optimal assignment that best satisfies the rules. Our Ponens uses the variational mean-field algorithm to perform the inference, yielding better scalability than traditional Prolog engines. For more details, please refer to the original paper [50].

2.3.4 Explanation from Ponens

Another strength of Ponens is that we can obtain the explanation for arbitrary queries. As mentioned above, once the problems are formalized, the symbolic engines can find the solution automatically. The proof of the solution can be used as an explanation for the query. However, traditional Prolog engines cannot directly yield proof during the inference. Although Prolog provides the "trace" functionality which logs each step during reasoning, the entire reasoning path remains unknown. To fully recover the proofs, we improve the Ponens to be capable of extracting the proofs for the solutions. This is achieved by logging each variable binding step. We save every query and corresponding solutions, and each solution is associated with a proof which also contains a list of subqueries.

This proof enables the visualization and automatic chain-of-thought generation. Fig. 4 shows an example of visualization where proofs are stored in a



Database

- descendant(X, Y) :- offspring(X, Y).
- descendant(X, Z) :- (offspring(X, Y), descendant(Y, Z)).
- offspring(abraham, ishmael).
- offspring(abraham, isaac).
- offspring(isaac, esau).
- offspring(isaac, jacob).

Query

- descendant(abraham, X)

Figure 3: An example of database and query in Ponens.

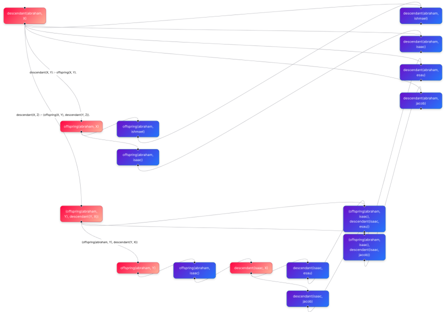
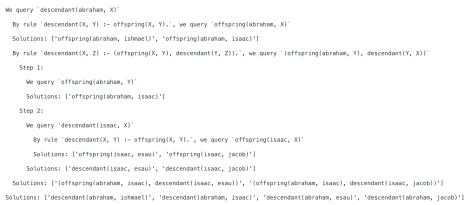

descendant(abraham, X)
descendant(X, Y) :- offspring(X, Y).
descendant(X, Z) :- (offspring(X, Y), descendant(Y, Z)).
offspring(abraham, X)
offspring(abraham, ishmael)
offspring(abraham, isaac)
(offspring(abraham, Y), descendant(Y, X))
(offspring(abraham, Y), descendant(Y, X))
offspring(abraham, Y)
offspring(abraham, isaac)
descendant(isaac, X)
descendant(isaac, esau)
descendant(isaac, jacob)
(offspring(abraham, isaac), descendant(isaac, esau))
(offspring(abraham, isaac), descendant(isaac, jacob))
descendant(abraham, ishmael)
descendant(abraham, isaac)
descendant(abraham, esau)
descendant(abraham, jacob)

Figure 4: Visualization of proof generated automatically by Ponens. The red (blue) nodes denote the queries (solutions).

directed acyclic diagram for the example task in Fig. 3. When converting it into text, we automatically obtain the chain-of-thought sentence, see Fig. 5. This also indicates a way of creating synthetic data. We can gather the explanation from Ponens when querying the database in the specific domains. This data can in turn facilitate the reasoning ability of LLM.

2.3.5 Evaluation on Public Datasets

We verify our method on three widely-used datasets, i.e., ProntoQA, ProofWriter, and FOLIO.



```
We query `descendant(abraham, X)`
  By rule `descendant(X, Y) :- offspring(X, Y).`, we query `offspring(abraham, X)`
  Solutions: ['offspring(abraham, ishmael)', 'offspring(abraham, isaac)']
  By rule `descendant(X, Z) :- (offspring(X, Y), descendant(Y, Z)).`, we query `(offspring(abraham, Y), descendant(Y, X))`
    Step 1:
      We query `offspring(abraham, Y)`
      Solutions: ['offspring(abraham, isaac)']
    Step 2:
      We query `descendant(isaac, X)`
        By rule `descendant(X, Y) :- offspring(X, Y).`, we query `offspring(isaac, X)`
        Solutions: ['offspring(isaac, esau)', 'offspring(isaac, jacob)']
      Solutions: ['descendant(isaac, esau)', 'descendant(isaac, jacob)']
  Solutions: ['(offspring(abraham, isaac), descendant(isaac, esau))', '(offspring(abraham, isaac), descendant(isaac, jacob))']
Solutions: ['descendant(abraham, ishmael)', 'descendant(abraham, isaac)', 'descendant(abraham, esau)', 'descendant(abraham, jacob)']
```

Figure 5: Explanation of proof generated automatically by Ponens.

- ProntoQA [32] is a dataset for analyzing the deductive reasoning abilities of LLMs. We use the "fictional characters" version of the dataset, which is the most challenging. Each version is divided into different subsets based on the number of reasoning hops required. We evaluate using the most difficult 5-hop subset. Each question in PrOntoQA is designed to verify the truthfulness of new facts. We use a 1-shot setting.

- ProofWriter [38] is another commonly used deductive reasoning dataset. Compared to PrOntoQA, the questions are expressed in a more natural language form. We use the Open World Assumption (OWA) subset. The dataset is divided into five parts, each requiring 0, ≤ 1, ≤ 2, ≤ 3, and ≤ 5 hops of reasoning, respectively. We evaluate using the most challenging subset. We use a 1-shot setting.

- FOLIO [10] is a challenging expert-written logical reasoning dataset. The questions mostly align with real-world knowledge and use highly natural phrasing, requiring complex first-order logic reasoning to solve. We evaluate using the entire FOLIO test set. We use a 2-shot setting.

For these tasks, we translate the context into formal languages and then use the symbolic engine to solve the problem. The samples that fail in the compilation are considered incorrect. Our model is trained based on our INF-LLM model with our proprietary alignment data, including the NL2FL data. We compare our model with GPT-4 and Qwen2 models. Table 2 illustrates the results, measured by answering accuracy, showing that we obtain remarkable results against advanced models in these logical reasoning tasks. We also include the ablation of Ponens. When not using Ponens, GPT-4 uses the Prolog engine [1] instead and greatly degrades the performance.

models	ProntoQA	ProofWriter	FOLIO
GPT-4	0.94	0.91	0.56
GPT-4 w/o Ponens	0.83	0.80	-
Qwen2-72B-chat	0.95	0.96	0.44
Qwen1.5-32B-chat	0.91	0.87	0.35
INF-LLM	**0.99**	**0.99**	**0.58**

Table 2: Comparison of different models on ProntoQA, ProofWriter, and FOLIO.

3 INF-LLM for Healthcare

Since the emergence of ChatGPT, many large medical language models (LLMs) have been developed. One of the most notable is Med-PaLM 2 [37] by Google,

[1] Pyke: https://pyke.sourceforge.net/

which is fine-tuned on medical domain data based on PaLM 2. Other medical LLMs, such as ChatDoctor [16], MedAlpaca [11], BenTsao [40], DoctorGLM [47], ChatMed [61], HuatuoGPT [57], DISC-MedLLM [2], and Taiyi [20], have been developed primarily through supervised fine-tuning on a base model. Additionally, models like PMC-LLaMA [45], MedicalGPT [49], Zhongjing [53], and CareGPT [41] are developed with both domain-specific pretraining and instruction alignment. Some even utilize reinforcement learning from human feedback (RLHF) to enhance safety. While these existing medical LLMs demonstrate capabilities in specific areas of the medical field, there is often a lack of comprehensive discussion on how to build models that are designed from the outset to address the challenges of real-world applications. Furthermore, there is insufficient discussion on how to evaluate whether these models possess the necessary capabilities for practical application.

The Grand Challenge for Medical LLMs: The capability to retrieve medical knowledge, reason over it, and answer medical questions comparably to physicians has long been viewed as a grand challenge, akin to protein folding [37]. Medicine is a complex and critical field, requiring a vast knowledge system that spans molecules, signaling pathways, cells, organs, tissues, and systems. The seriousness of the field necessitates that responses to related questions must be accurate, whereby safety is critical.

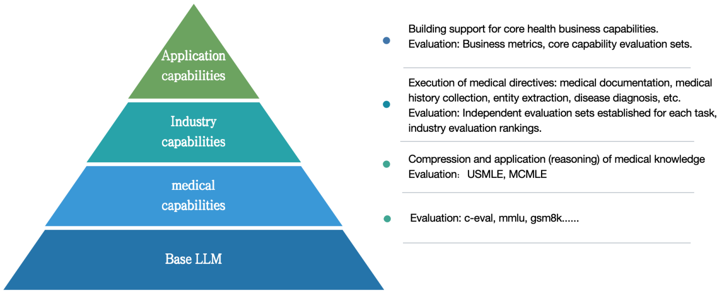
During pretraining, large language models read vast amounts of data, including books, journals, and a substantial number of web pages and forums. Ensuring that the medical knowledge learned through training is authoritative, accurate, and up-to-date, while also being effectively mastered and reliably applied in reasoning and output, is a prerequisite for the practical value of medical LLMs. In our view, the main challenge is how to make medical LLMs trustworthy.

To build a trustworthy medical LLM that can encode medical knowledge,follow medical instructions,and be competent for medical applications, we categorize the LLM's medical skills into three levels,as shown in the Figure 6 below.

- **Basic Skills**: These can be likened to those of a medical doctoral student nearing completion of their studies. Such a model has acquired rich and systematic knowledge of biomedical and clinical medicine and is capable of using this knowledge for medical logical reasoning and analysis to draw conclusions.

- **Industry-specific Skills**: These skills are akin to those of a resident physician who can write medical documents, collect patient histories, maintain progress notes, and prescribe treatments under the guidance of senior physicians.

- **Application Skills**: These skills are comparable to those of an associate chief physician or chief physician. In specific application scenarios, the model must possess the ability to tackle challenging tasks and achieve a certain level of performance. Unlike industry-specific capabilities, this emphasizes a specialized approach to solving particular problems with a focus on achieving specific performance metrics.



Building support for core health business capabilities.
Evaluation: Business metrics, core capability evaluation sets.

Execution of medical directives: medical documentation, medical history collection, entity extraction, disease diagnosis, etc.
Evaluation: Independent evaluation sets established for each task, industry evaluation rankings.

Compression and application (reasoning) of medical knowledge
Evaluation： USMLE, MCMLE

Evaluation: c-eval, mmlu, gsm8k......

Figure 6: Medical skill levels overview.

To solve the trustworthiness problem when building medical LLMs in INF, we adhere to the following rules:

- Use high-quality, authoritative pretraining data that is rigorously cleaned.
- Employ professionally annotated medical data for supervised training.
- Ensure model predictions are traceable.
- Make the reasoning process interpretable.

The first rule primarily pertains to training basic skills, while the second rule focuses on industry-specific skills training. The latter two rules are essential for developing reliable medical applications. These principles will be further detailed in the sections on data preparation and application development.

3.1 High-quality Domain Data for Continuous Training

In this section, we will introduce our methodology for high-quality continuous training data preparation. It mainly includes four parts: data source control, quality assessment framework, data cleaning framework, and data classification system.

3.1.1 Data Collection: High-Quality Data from Authoritative Sources

The knowledge acquired through pretraining and continuous training comes from authoritative and highly current sources, serving as the foundation for trustworthy models. The dataset used for our model is meticulously curated from a wide range of authoritative and credible sources ensuring both the authority and accuracy of the information. We collected over 70B tokens in healthcare and medical domain from various sources, such as encyclopedias, journal articles, authoritative clinical guidelines, books, educational materials, and authentic hospital records. These sources ensure a comprehensive coverage of medical research, teaching, clinical practice, user interaction, and medical popularization.

3.1.2 Data Quality Assessment: What is High-Quality Data

Although domain data is collected from authoritative and highly current sources, there still exists a kind of noise when adapting those data for training. Further control and improvement of data quality have become crucial factors for the success of medical LLMs. This study has developed a comprehensive data quality control process aimed at improving the quality of medical data through stricter standards. As illustrated in the Figure 7, the entire data quality control technology framework is presented. The core technologies include data quality assessment standards, a data quality scoring framework, a data cleaning framework, and a data classification system.

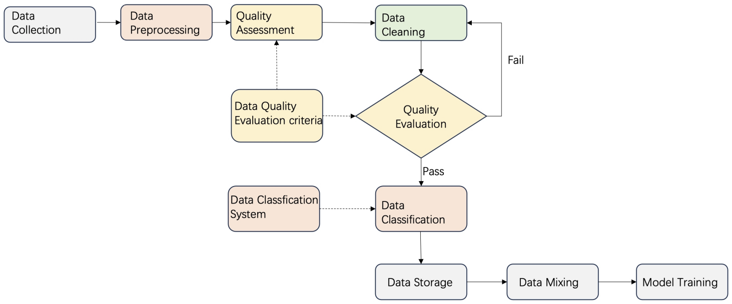


Figure 7: Data quality control framework.

Standards for Data Quality Evaluation: In the medical field, the foundational quality of data critically impacts the successful training of models. Basic grammatical and semantic issues, as well as various types of noise and data duplication, can affect model performance. Moreover, due to the rigor required in the medical field, there are higher demands for the professionalism, accuracy, reliability, completeness, and ethics of the responses generated by LLMs. Therefore, beyond the basic quality of data, the quality of information in medical data also demands higher standards.

We have developed a comprehensive medical data quality assessment framework, focusing on three aspects: foundational quality, information quality, and data security. As shown in Figure 8, the framework encompasses six primary dimensions: grammatical correctness, formatting norms, text cleanliness, semantic validity, information validity, and ethical security. The first four dimensions are foundational quality controls, where information validity directly measures the accuracy and other aspects of data information, and ethical security sets higher standards for data compliance and sensitivity.

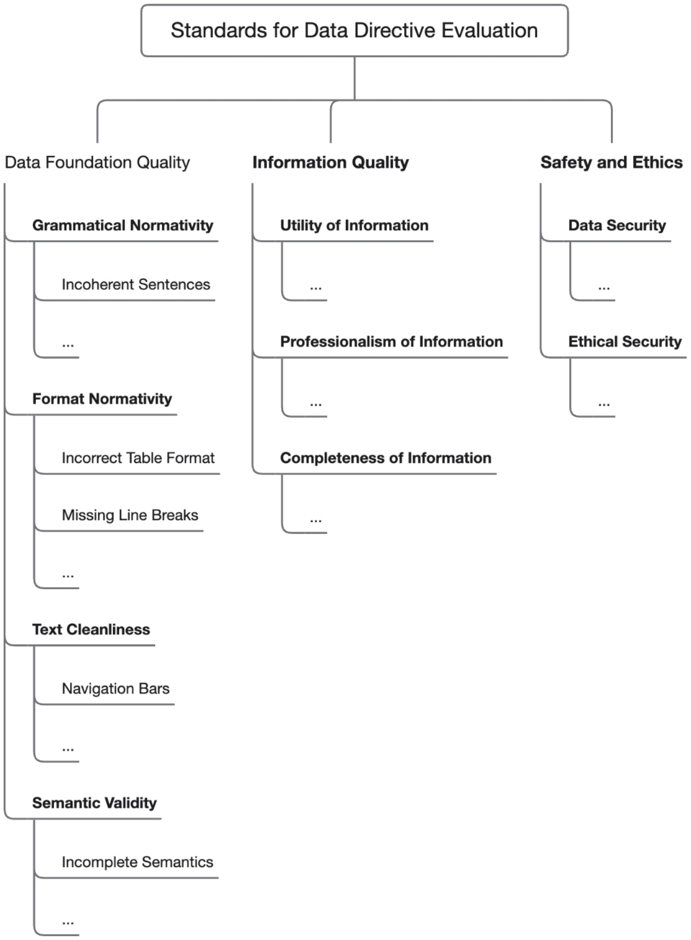


Figure 8: INF healthcare data quality assessment standards.

Additionally, each primary dimension includes several secondary dimensions. For instance, text cleanliness further divides into categories like navigation bars, advertisements, special symbols, and noise segments. Information validity standardizes and differentiates the usefulness, accuracy, professionalism, reliability, and completeness of data information. The entire medical data assessment framework covers 45 secondary quality dimensions to provide a comprehensive evaluation and control of medical data quality.

Data Quality Scoring Framework: Through the data quality assessment framework, it is convenient to conduct a qualitative analysis of issues within medical data. To quantify these issues more precisely, this study proposes a universal data quality scoring strategy based on the assessment framework outlined earlier.

- **Impact Degree Rating**: Initially, the 45 secondary quality assessment dimensions are rated for their impact degree, divided into 4 to 6 levels. Higher levels indicate more severe issues that have a greater negative impact on model training. For example, basic grammatical errors (such as typos) have a minor impact, whereas inaccurate information can lead the model to learn incorrect knowledge, and the most severe issues, like data violating safety regulations, receive the highest impact ratings.

- **Quality Assessment and Text Length** : Based on the impact ratings, the quality of data can be precisely assessed after each type of issue occurs. Considering that the impact of most issues is diluted as text length increases, this also affects the quality score of the data.

- **Global Frequency Impact** : From a global perspective of the dataset, the frequency of certain issues significantly affects model training. For instance, while some issues might have a minor impact when they occur only once, they require special attention if they occur frequently. This study measures the impact of globally frequent issues through frequency scaling.

Taking these factors into account, we have designed a quality scoring strategy for individual data entries and datasets to achieve a quantitative analysis of data quality.

The qualitative and quantitative assessments of data quality eventually form in a detailed quality analysis report. This report includes the quality scores of the dataset, statistical proportions of various issues, and changes in distribution compared to previous cleaning results. Additionally, for critical issues, the report also provides the benefits after the issues have been cleaned, further guiding the data cleaning process.

3.1.3 Data Cleaning Framework: How to Achieve High-Quality Data

The primary goal of data cleaning is to address various issues identified during the quality assessment process. Based on a detailed data quality assessment report, systematic statistical analysis and feature extraction are performed to formulate a cleaning strategy report. According to this report, key issues in the data, such as characteristics of noise, are analyzed in detail. These issues are precisely located through the construction of heuristic rules, followed by the repair or removal of problematic segments.

In addressing different sources of medical datasets, this study has developed various cleaning methods. Each dataset undergoes an independent quality assessment and cleaning treatment. During the cleaning process, mature heuristic rules formed are incorporated into a cleaning toolkit for direct application when processing other datasets, thereby enhancing cleaning efficiency. In particular, the quality inspection and cleaning process focus on three main aspects: semantic integrity of the data, ensuring consistency in logic and meaning; text cleanliness, removing extraneous characters and noise; and information validity, ensuring data accuracy and practical applicability.

For more complex issues, a combination of rules is supported and resolved through data processing algorithm logic, the N-gram models or classification models. N-gram models help capture the contextual relationships of words and phrases, better identifying and correcting potential issues in the data. Classification models can automatically categorize and label problem data based on training data, further improving the precision and efficiency of cleaning. This study has constructed an end-to-end data identification and cleaning framework that adopts a divide-and-conquer approach to tackle the challenges of identifying medical professional books, dealing with lengthy texts, complex assessments, and difficult cleaning. This framework can modularly handle each segment and has proven highly adaptable and practical in scenarios involving book identification and processing.

Table 3 list the changes in scores after data cleaning for typical medical data, as well as the cleaning rates. It can be seen that the scores for health science data and guideline data have significant changes before and after cleaning. The score for health science data increased from 58.28 before cleaning to 91.82. Additionally, thanks to the refined cleaning scheme, our cleaning rate remained at 10.6%.

Ultimately, this work developed 140 universal text cleaning rules and 42 universal data cleaning scripts. These rules and scripts have been rigorously tested and validated, effectively supporting the cleaning of various data types in medical texts, thus ensuring high-quality data and providing a solid foundation for subsequent training of large-scale language models.

Table 3: Example of data quality changes and cleaning rate statistics before and after data cleansing.

Datasets	Pre-Cleaning Score	Post-Cleaning Score	Cleaning Rate
Health Science Popularization(健康科普)	58.28	91.82	10.64%
Manuals(说明书)	83.56	96.5	2.12%
Guidance Documents(指南)	50.2	87.77	19.59%

3.1.4 Data Classification System: Better Understand Your High-Quality Data

After the data cleaning phase, this technical report has accumulated a large amount of high-quality medical data, which has greatly improved in terms of text quality, information quality, and ethical safety, laying a solid foundation for building a reliable medical large model. However, medicine as a complex life science, has unique professionalism and rigor. The medical data knowledge system is complex and diverse, necessitating more refined management and categorization of stored medical data.

To this end, this technical report further constructs a detailed categorization system for both Traditional Chinese Medicine (TCM) and Western medicine data. The purposes of this categorization system are:

- **Complete Data Analysis**: To help promptly identify issues of knowledge balance and ensure comprehensive data coverage.
- **Data Ratio Guidance**: During the pre-training phase, to guide the setting of dataset ratio parameters, optimizing model training outcomes.
- **Model Capability Assessment**: To form a feedback loop with model capability assessment, helping to promptly identify poorly performing and data-deficient task categories, and improve model performance.

Based on the medical discipline classification framework, this technical report has constructed detailed two-tier categorization systems for both TCM and Western medicine. For example, in Western medicine, the system includes 37 primary categories such as “Physiology”, “Pathology”, “Internal Medicine”, and “Surgery”. “Physiology” is further subdivided into “Cell Physiology”, “Systemic Physiology”, “Human Physiology” etc.; “Pathology” into “Systemic Pathology”, “Clinical Pathology”, etc.; “Internal Medicine” includes 12 subcategories such as “Cardiology”, “Respiratory Medicine”, and “Gastroenterology”. The entire system comprises 55 secondary subcategories.

Similarly, for TCM data, we have also constructed a two-level categorization system. TCM data has 8 primary categories covering “Theoretical Studies”, “Clinical Disciplines”, “Formula Studies”, “Diagnostic Studies”, etc., with a total of 24 secondary categories. For example, under “Clinical Disciplines” are categories like “Internal Medicine”, “Surgery”, “Orthopedics”; under “Acupuncture” are “Acupuncture Therapy”, “Moxibustion”, etc.

By building these categorization systems, we can manage and utilize medical data more systematically and scientifically, thereby enhancing data usability and model reliability.

According to the Figure 9, in Western medicine data, the categories “Pharmacology”, “Internal Medicine”, and “Pathology” occupy a significant proportion of the data, followed by “Microbiology”, “Surgical Science”, and “Anatomy”. This shows that both the theoretical foundations (such as “Pharmacology”,

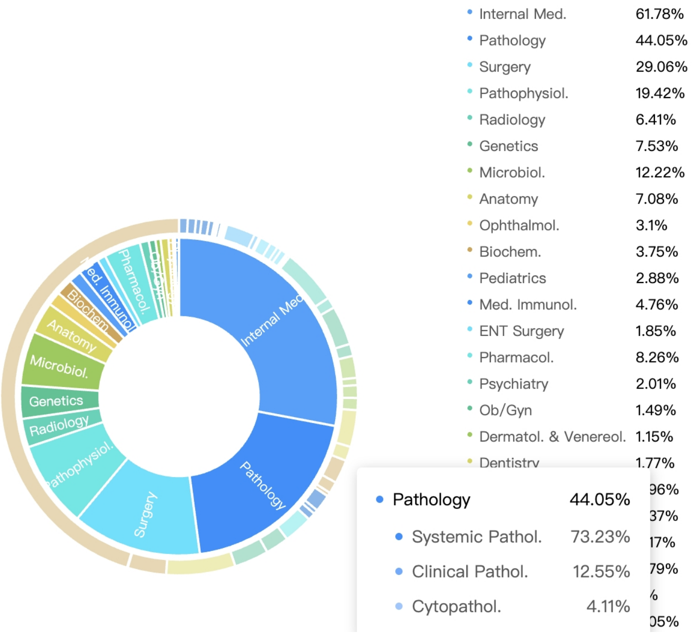


Figure 9: Example of western medical data classification

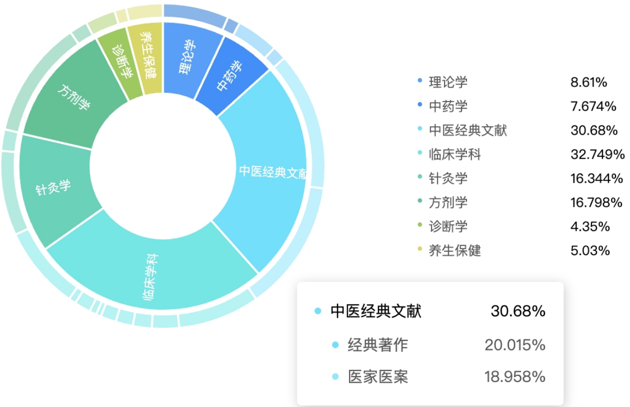


Figure 10: Example of traditional chinese medicine(TCM) data classification

“Pathology”) and applied medicine (such as “Internal Medicine”, “Surgical Science”) are well represented in the dataset, ensuring comprehensive learning coverage of the model.

For TCM data, as shown in Figure 10, the applied categories (such as ”Clinical Disciplines”) and theoretical foundations (such as ”Classical Literature”) each account for over 30% of the data. This indicates that TCM data is well-balanced and comprehensively covers both foundational theory and practical application. This balanced data distribution includes a deep understanding of classic TCM theory as well as rich practical experience.

3.2 Domain Data Preparation for Instruction Following

3.2.1 Overview

In this section, we first introduce the capabilities that INF-Med should possess to execute various medical instructions, followed by a description of how we constructed the medical instruction dataset.

As above mentioned, we divide the learning and growth trajectory of doctors in Mainland China into three stages, corresponding to the basic skills, industry-specific skills, and application skills of the medical LLM. Based on the role played by the LLM at each stage, we define the corresponding data and capability system.

- **Basic Skills**: This stage focuses on the usage of basic knowledge and logical reasoning. We constructed a large amount of Chain-of-Thought(CoT) data based on various medical professional qualification examinations to enhance the model’s medical reasoning capabilities.
- **Industry-specific Skills**: This stage focuses on comprehensive medical industry capabilities. We collected comprehensive medical domain data, including knowledge Q&A, document generation, information processing, and diagnostic assistance, to improve the model’s ability to perform high-frequency tasks within the medical industry.
- **Application Skills**: This stage focuses on highly specialized application capabilities within the medical industry. Specially, we constructed three capabilities in the medical examination application: abnormal terms extraction, personalized advice generation, and report quality control.

To construct a trustworthy, comprehensive LLM that aligns with the current state of the medical industry, INF-Med simulate the various stages of a doctor’s growth in mainland China and is trained with different stages of instruction alignment data.

3.2.2 Instruction Alignment Data for Basic Medical Skills

Due to the presence of complex reasoning scenarios in the medical field, there is a high demand for the logical capabilities of models. To improve the performance of models in these scenarios, we referenced the work of Auto-CoT [60] by constructing a large amount of medical scenario COT [42] data through automated methods. Additionally, to ensure the correctness of medical knowledge and the completeness of logic in the data, our medical annotation team conducted rigorous quality control checks on all the data.

Apart from data quality, we also observed that the performance of general large language models in Chinese and English scenarios is inconsistent. This discrepancy may be attributed to the varying proportions and quality of corpora in different languages during the pre-training phase. To enhance the model's performance across different languages, we adopted the method using cross-lingual CoT [29], which extends the model's capabilities from a single language to multiple languages. By utilizing this approach, we can enhance the model's effectiveness across various languages.

3.2.3 Instruction Alignment Data for Industry-specific Skills

As far as we know, current LLM in the healthcare industry (e.g., Med-PaLM [36], PMC-Llama [45], HuaTuo [40]) are more focused on single dimensions, such as passing professional qualification exams or enhancing doctor-patient Q&A and dialogue, lack of systematic capacity building solutions. Therefore, we construct training data involving multiple capability dimensions, such as knowledge question-answering, document generation, clinical diagnosis, safety ethics, and document structuring.

Data Collection and Selection: To enhance the capabilities of LLM across various dimensions in the healthcare industry, we collected a substantial amount of open-source instruction alignment data in the medical field and performed rigorous data selection. Specifically, we completed the initial data selecting using the following three heuristic filtering methods:

- **Task Type Filter**: During the industry capability building process, we first mapped medical task types to training task types and automatically labeled the types of open-source data. Ultimately, we discarded irrelevant data.

- **Construction Method Filter**: We traced the construction methods of various open-source datasets (e.g.LLM-based, knowledge graph-based, web crawling) as one of the reference standards for assessing data quality. Through this method, we filtered out a large amount of data with content errors and colloquialism issues.

- **Data Length Filter**: Using the length of questions and responses as one of the reference standards for judging the information content of the current data, we manually set reasonable length thresholds for different open-source datasets. We discarded data that did not meet these length expectations, thereby filtering out a significant amount of low-information-density data and data with issues such as cyclic generation.

In addition, we utilized internally accumulated data (e.g., physical examination reports, medical records, clinical diagnostic test questions) to produce high-quality data for scarce medical task types in the open-source community.

Content Review: The data after initial filtering has a certain level of assurance regarding task type relevance and data quality. However, some content-related issues (such as irrelevant responses, etc.) still cannot be entirely avoided. Therefore, we need to further optimize and quality check the data with a content-oriented approach. Specifically, we execute the following content inspection process:

- LLM-based question-answer matching score evaluation
- LLM-based regeneration based on original answers
- Review by professional medical team

3.2.4 Instruction Alignment Data for Application Skills

Abnormality Extraction of Medical Examination Reports: Abnormalities extraction of medical examination reports is a critical task involving the automated identification and extraction of abnormal findings. Unlike traditional entity and relation extraction task, this task presents unique challenges as it requires the structured extraction of both entities and their relationships simultaneously. Another significant challenge is the complexity of medical terminology.

In practice, medical terminology and report writing are often non-standard and inconsistent, which requires the model to have very strong medical knowledge. Even state-of-the-art models like GPT-4 struggle with the complexity of this task, with our testing indicating that GPT-4 achieves an F1 score of only 0.58.

To solve the problem, we designed a meticulously crafted task which helps the model to understand the medical terminologies and their relationships better. All data is annotated by professional doctors in the examination department of the hospital, which guarantees quality and precision of the data.

Personalized Health Advice Generation: Personalized LLM response generation holds the potential to offer substantial benefits for individuals in critical areas such as medical.

In the medical examination scenario, a qualified health advice should comprehensively consider personalized factors such as the patient's gender, age, occupation, marital and reproductive history, medical history, and family history. Health advice specific to certain abnormalities often vary due to differences in these factors. Unlike general health advice, the challenges of personalized health lie in:

- Accurately capturing the relevance between personalized information and examination abnormalities and reflecting this in the recommendations.

- The lack of high-quality training data for personalized examination recommendations in medical industry.

To address these issues, we sampled high-quality reports from the medical examination reports of several top-tier hospitals to construct a dataset for personalized health advice. Throughout the process, a professional medical team was responsible for the quality of the personalized health advice, with particular attention to the relevance between personalized information and abnormalities, as well as the accuracy and rationality of the advice.

We used these data for fine-tuning INF-Med and constructed an unseen test set for performance evaluation. Ultimately, we surpassed GPT-4 and other open-source LLMs in both accuracy rate and precision of personalized information.

Quality Control of Medical Examination Reports: Due to the involvement of multiple departments and the complexity of content, along with significant individual differences, errors are common in health examination reports. According to research, errors in health examination reports account for more than 50

- **Grammatical Errors**: Such as extra or missing words, incorrect characters, punctuation, sequence numbers, etc.
- **Data Entry Errors**: For example, incorrect entries for height and weight, systolic and diastolic pressures switched, results entered for unchecked or omitted tests.
- **Errors in Imaging and Test Results**: Such as uploading CT images of the wrong part, incorrect reference values leading to erroneous conclusions, errors in distinguishing left from right in ultrasound examinations.
- **Errors by Chief Physicians**: Such as missed diagnoses, inaccurate health advice, or failure to comprehensively analyze based on the actual conditions of the examinees.

Errors in health examination reports can lead to disputes, harming customer loyalty and institutional reputation. Traditional quality control depends on expert reviews, which are costly and slow. We explore using LLM for quality control of Medical examination reports to resolve issues like departmental conflicts and missing conclusions.

To achieve this, we designed data mining strategies to identify problematic reports from anonymized data. We calculate issue probabilities using two models, and validate data with annotations by professional doctors, and employ a multiple-annotations strategy to ensure data quality. We use the COT strategy to construct data and train our model, and then evaluate performance using True Positive Rate (TAR) and False Positive Rate (FAR) metrics.

3.3 Model Training

We conducted continuous training and instruction alignment training on INF's homemade 34B model (INF-LLM-34B) which was pretrained in-house from scratch. In this section, INF-Med-CT refers to the base model obtained after continuous training on INF-LLM-34B, while INF-Med refers to the model obtained after instruction alignment on INF-Med-CT.

3.3.1 Medical Domain Continuous Pretraining

Domain-specific further pretraining enhances the capabilities of large language models within specialized fields [9]. Based on our INF-LLM-34B, we further pretrained the model base on our domain dataset, got a cutting-edge medical large language model, INF-Med-CT. By emphasizing the quality of medical data, balancing it with selected general data and optimizing the training epochs, INF-Med-CT achieves a robust enhancement of its medical domain capabilities without losing its general capabilities.

3.3.2 Domain Instruction Alignment Training

In the supervised fine-tuning stage, INF-Med was initialized by the medical foundation model INF-Med-CT, and optimized by the AdamW [19] optimizer ($\beta 1 = 0.9$, $\beta 2 = 0.95$, $\epsilon = 10^{-8}$) with a learning rate of 1.0×10^{-5} for the 34B model. The learning rate increases to the peaking value with the cosine learning rate schedule (3% warm-up steps) and then remains constant. We also added general instruction alignment data to mix 1:1 with medical instruction alignment data to maintain model general ability.

3.4 Model Evaluation

As mentioned in Section 3, we categorize INF-Med skills into three levels. To test trustworthiness of INF-Med at each level, we used different evaluation datasets and metrics.

3.4.1 Basic Skills Evaluation

Continuous Training: We tested our model's basic skills in medical domain on the United States Medical Licensing Examination (USMLE). The USMLE is a rigorous, standardized exam that all physicians must pass to practice in the United States. There are 3 steps of the exam, each step has specific focus and unique objectives, and they together ensure a comprehensive assessment of a medical professional's competency. Step1 assesses the foundational medical knowledge of medical students, step2 assesses basic clinical knowledge and step3 assesses the advanced clinical knowledge and it's application. Our model achieves impressive performance on USMLE sample example as shown in Table 4: 68.91 on the USMLE step1, 73.33 on the USMLE step2 and 78.10 on the USMLE step3, which surpasses larger model like Qwen1.5-72B-Base. This remarkable achievement indicates that our model has comparable performance on those exams than human physicians, proofs the potential of our model to assist in medical practice and education across different linguistic contexts.

Table 4: USMLE evaluation results for INF-Med-CT

Model	Overall Average	USMLE step1	USMLE step2	USMLE step3
Qwen1.5-72B-Base (5-shot)	67.56	65.57	68.33	68.61
INF-LLM-Base (5-shot)	67.02	62.18	68.33	70.07
INF-Med-CT (5-shot)	73.67	68.91	73.33	78.10

Instruction Alignment: With medical instruction alignment training data our model gained a step further in USMLE, achieving scores as in Table 5. Our model achieves 76.9 on the USMLE Sample Exam under 0-shot settings and 85.9 with the Medprompt method [23].

Table 5: USMLE evaluation results for INF-Med. Here "*" denotes cited from report [22]. Qwen1.5-72B-Chat is the zero-shot results of the Qwen1.5-72B-Instruct model using the same experimental setup.

Model	Overall Average	USMLE step1	USMLE step2	USMLE step3
GPT-4	84.31*	80.67	81.67	89.78
Qwen1.5-72B-Chat	67.3	64.7	69.2	67.9
INF-Med(Ours)	76.9	77.3	75	78.1
INF-Med-medprompt(Ours)	**85.9**	87.3	80	89.7

3.4.2 Industry-specific Skills Evaluation

To evaluate the industry capabilities of INF-Med, we conducted experiments on various self-built datasets and public benchmarks. This section primarily introduces the results achieved on MedBench [4]. Experiment results on self-built datasets refer to Application Skills Evaluation.

MedBench is an open platform for the evaluation of Chinese medical large models[2], characterized by scientific rigor, fairness, and a comprehensive approach. It quantifies the capabilities of models across various medical dimensions based on authoritative medical standards. MedBench includes the following sub-tasks:

- **Medical Knowledge QA**: Including 6 sub-tasks, focuses on the model's abilities in medical exams, medical consultations, departmental triage, and doctor-patient dialogues.

[2] MedBench leaderboard is accessible at `https://medbench.opencompass.org.cn/leaderboard`. Due to the real-time nature of the leaderboard, scores and rankings may change.

Table 6: MedBench evaluation results for various subtasks

Abilities & Metrics		GPT-3.5	GPT-4	INF-Med
Medical Knowledge QA	Med-Exam (Acc)	29.8	52.8	**91.5**
	MedHC (Marco-Recall)	57.1	84.9	**85.6**
	MedMC (Marco-Recall)	45.4	67.5	**76.7**
	MedSpedQA (Marco-Recall)	56.7	75.5	**77.2**
	MedHG (Micro-F1)	67.1	**82.2**	81.6
	MedDG (Marco-Recall)	46.4	76.3	**81.4**
Medical Language Generation	IMCS-MRG (Marco-Recall)	64.5	71.2	**72.2**
	DBMHG (Marco-Recall)	71.7	75.7	**76.2**
Medical Logical Reasoning	CMB-Clin (Marco-Recall)	72.1	**87.2**	83.7
	DDx-basic (Micro-F1)	32.2	82.1	**88.5**
	DDx-advanced (Micro-F1)	15.4	78.9	**92.6**
	MedTreat (Marco-Recall)	36.7	51.3	**51.9**
Medical Language Understanging	CMeEE (Micro-F1)	23.9	30.9	**52.5**
	CMeIE (Micro-F1)	16.5	26.5	**54.1**
	CHIP-CDEE (Micro-F1)	33.6	45.2	**79.7**
	CHIP-CDN (Acc)	87.3	90.0	**99.3**
	CHIP-CTC (Acc)	46.5	52.0	**84**
	SMDoc (Acc)	92.7	92.4	**96.9**
Medical Safety and Ethics	MedSafety (Acc)	37.4	47.5	**82.8**
	DrugCA (Acc)	59.3	64.0	**77.3**
Overall Score	**-**	49.9	75.5	**90.4**

- **Medical Language Generation**: Including 2 sub-tasks, focuses on the model's abilities to generate electronic medical records based on doctor-patient dialogues.

- **Complex Medical Reasoning**: Including 4 sub-tasks, focuses on the model's abilities in clinical diagnosis, differential diagnosis, and treatment generation.

- **Medical Language Understanding**: Including 6 sub-tasks, focuses on the model's capabilities in medical term extraction, term normalization, and event extraction.

- **Medical Safety and Ethics**: Including 2 sub-tasks, focuses on the model's understanding of medical ethics and safety.

The comparison of our results with OpenAI public models on the leaderboard is shown in the Table 6. It is worth noting that we ranked first on the MedBench public leaderboard, and achieved the best scores in the dimensions of Medical Logical Reasoning, Medical Language Understanding, and Medical Safety and Ethics.

3.4.3 Application Skills Evaluation

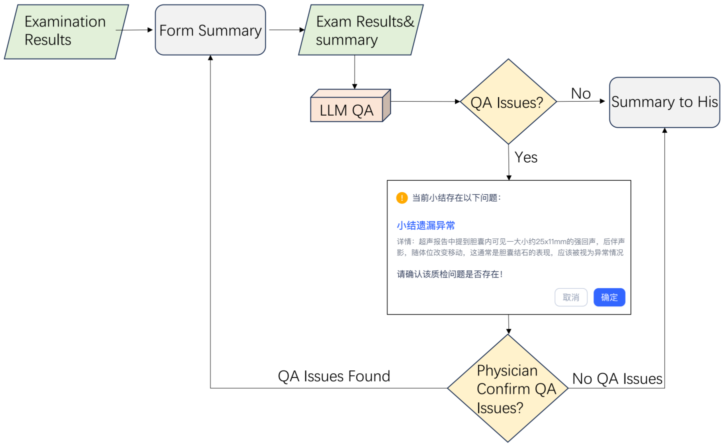
Personalized Health Advice Generation We evaluated INF-Med's ability to generate personalized health advice using our self-built test dataset. We adopt the Accuracy Rate (Acc) and the Precision of Personalized Information (PPI) as evaluation metrics. The specific results are as shown in Table 7.

Table 7: Evaluation results for personalized health advice generation

Methods	Acc	PPI
GPT-4	40%	44.32%
Qwen1.5-72B-chat	35%	39.13%
INF-Med	45%	72.73%

We invite professional medical teams to evaluate personalized health advice, and the result shows that INF-Med outperformed GPT-4 and Qwen1.5 across the Acc and PPI metrics. The adoption criteria are strict, i.e., health advice is only considered to be accepted if all relevant personalized information has been mentioned and the medical expression is correct.

Quality Control of Medical Examination Reports As illustrated in the Figure 11, this study proposes a comprehensive and real-time automated quality control framework for medical examination reports. Upon completing the medical examination report, the physician inputs the results and the report into the medical model. The model generates real-time quality control results and issues a pop-up alert if any quality control issues are detected. If an anomaly is confirmed, the physician is required to re-optimize the report.



当前小结存在以下问题：

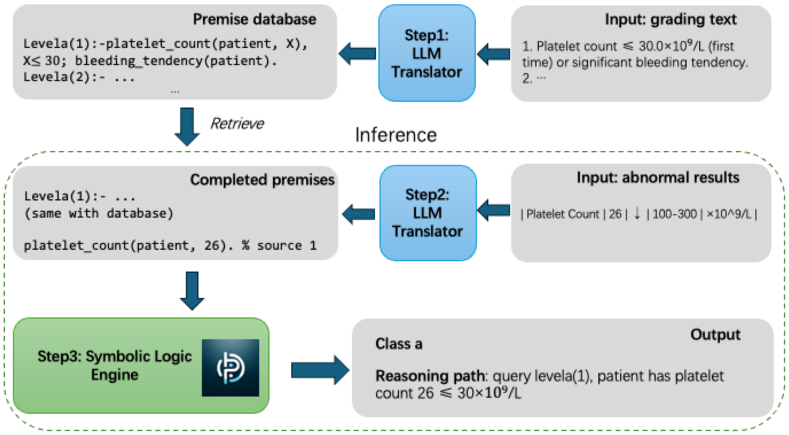
小结遗漏异常

详情：超声报告中提到胆囊内可见一大小约25x11mm的强回声，后伴声影，随体位改变移动，这通常是胆囊结石的表现，应该被视为异常情况

请确认该质检问题是否存在！

Figure 11: Medical report quality flow

In this scenario, INF-Med achieves a TAR of 70%, compared to GPT4 35%, at the same FAR.



```
Levela(1):-platelet_count(patient, X),
X≤ 30; bleeding_tendency(patient).
Levela(2):- ...
...
```

1. Platelet count ⩽ 30.0×10⁹/L (first time) or significant bleeding tendency.
2. ···

```
Levela(1):- ...
(same with database)

platelet_count(patient, 26). % source 1
```

| Platelet Count | 26 | ↓ | 100-300 | ×10^9/L |

Class a

Reasoning path: query levela(1), patient has platelet count 26 ⩽ 30×10⁹/L

Figure 12: Neuro-symbolic computing for medical examination report grading

3.5 Application Case Studies

3.5.1 ABUO Grading Task

Health examination reports usually include a combination of indicator values (laboratory tests) and text descriptions (general check, physical examination). Our task is to classify the abnormal results in the report into different levels (A, B, U, and O), based on specific grading criteria provided. (1) Level A: abnormal results that require immediate clinical intervention, as failure to intervene could threaten life or lead to severe adverse outcomes. (2) Level B: abnormal results that require reporting the process and expedited handling. (3) Level U: abnormal results that require outpatient follow-up re-examination. (4) Level O: abnormal results that do not meet A, B, and U. In particular, special attention needs to be given to levels A and B during abnormal medical screenings, as they require further medical intervention strategies.

3.5.1.1 Neuro-symbolic Computing

The grading task demands strict logical reasoning involving numerical values, indicator values, and the process of logical reduction, areas where current LLMs might still fall short. Additionally, beyond providing a grading result, it is crucial to demonstrate the complete reasoning process for interpretability, which is vital in medical applications. We view neuro-symbolic computing as an ideal solution for addressing this issue.

Our neuro-symbolic approach consists of three steps, as illustrated Figure 12. We first 1) translate the grading criteria text into the symbolic format expressed in first-order logic automatically using INF-Med. During inference, we 2)retrieve relevant premises, and use INF-Med to generate additional premises expressed in symbolic format according to patient medical examination report and abnormal results. Finally, 3) These expressions are then offloaded to an in-house symbolic engine, Ponens, which performs deductive inference in a strict symbolic manner and outputs the whole reasoning process in addition to the conclusion.

Symbolic Formalization of Grading Criteria: We use the 202301 version of the health examination grading standard. This standard includes grading criteria for three levels: A, B, and U. Each level has multiple criteria expressed in a combination of qualitative and quantitative terms.

We leverage INF-Med to automatically translate text into a symbolic format by exploiting their in-context learning capabilities. Specifically, we carefully selected 5 examples that encompass both qualitative and quantitative criteria across all three grading levels, we also provide human-verified symbolic formulation, i.e., the Prolog program to include as a reference answer for in-context learning. Below is an example translation of one level A criterion.

Natural language context

Level A: Platelet count $\leqslant 30.0 \times 10^9/L$ (first time) or significant bleeding tendency

(A类：血小板计数$\leqslant 30.0 \times 10^9/L$（首次）或有明显出血倾向)

Ponens

$\mathrm{levela} : -\mathrm{platelet\_count}(\mathrm{patient},\ X), X \leq 30; \mathrm{bleeding\_tendency}(\mathrm{patient}).$

As a result, we obtained more than 300 premises as our database upon which our decisions are based. We sampled a small portion for annotation to verify the correctness of the translations, achieving an accuracy rate of over 90%.

Neuro-symbolic Inference In addition to converting grading criteria into premises, we need to extract additional premises from the medical report specific to the patient at inference time. One challenge we faced was aligning our translations with those in the premise database to ensure correct logical deduction in stage 3. Our solution involves first retrieving candidate premises from the database created in 3.5.1.1, based on the medical examination report, and extracting abnormal results using an off-the-shelf embedding model. We then constrain the INF-Med to select from these existing predicates while extracting corresponding numerical and indicator values consistent with the report. Specifically, we provide a few in-context examples to help the model understand what to output. We also allow the model to return void none of the candidate predicates are directly mentioned in the report. Finally, we parse the output and extract additional premises from the report to complete the symbolic formalization process.

Below is an example of a premise generated by INF-Med from a medical report and abnormal results. We observed that INF-Med can discern nuanced differences in medical descriptions, such as distinguishing between a pulmonary ground-glass nodule and a suspicious pulmonary nodule, and can also handle unit conversions for numerical values properly. To minimize uncertainty in translation, we employ 3-way majority voting to select the final premises.

Input:
Abnormal results:

- 1. Right ethmoid sinusitis (右侧筛窦炎.)
- 2. Ground-glass nodule in the apicoposterior segment of the left upper lobe, 0.7 cm (左肺上叶尖后段磨玻璃小结节 0.7cm)
- 3.A few fibrotic foci in both lungs (双肺少许纤维灶)

Candidate predicates:

- Mediastinal_nodule (binary)
- Sinoatrial_conduction (binary)
- Pulmonary_ground_glass_nodule (num): unit mm
- Suspicious_single_or_multiple_pulmonary_nodular_lesions (binary)

Output
Pulmonary_ground_glass_nodule(patient, 7). % source 2

Symbolic Inference We pass all premises from the database, those derived from the medical examination report as well as the conclusion to our in-house symbolic engine, Ponens for logic reduction (details introduced in 2.3.2). Our process ensures the faithfulness of our reasoning process, i.e., only derive conclusions based on provided grading criteria. The logic engine returns the result grading as well as its detailed reasoning process for visualization and full explainability.

3.5.1.2 Experiments

We collected 94 sessions of medical examination reports that contained abnormal results, covering both test sessions and imaging sessions. All of our data are sampled from real health examination scenarios. We annotate the data at each abnormal result level. The annotators are physicians from top-tier hospitals in China. Each sample is initially labeled by a chief physician, followed by a secondary review by medical experts to ensure the label quality the data.

3.5.1.3 Baselines

We compare our model against two chain-of-thought technique (COT) baselines that depend solely on LLMs for logical reasoning: 1) Naive LLM, where we do not provide extracted abnormal results, only the whole report session and original ABU grading criteria in natural language format. 2) LLM + abnormal results, where extracted abnormal results are provided to reduce potential noise and for a fair comparison with our model. For both baselines, we employ COT to ask the model to output its reasoning process step-by-step. We separately evaluate the settings that GPT-4, and INF-Med serve as the underlying LLMs. To ensure a fair comparison, we use a decoding temperature of T = 0.8 for all experiments. We evaluate the accuracy at the session level, using a strict set match for comparison with the gold label. We also evaluate the accuracy at per abnormal result level (each session can include multiple abnormal results), with an emphasis on A and B grading since they are more severe and have less representation in the dataset.

3.5.1.4 Results and Discussion

We present the accuracy at both the session level and the abnormal results level in Table 8. Our neuro-symbolic approach significantly outperforms two LLM-only baselines in both session accuracy and per abnormal accuracy. Specifically, when using INF-Med as the underlying model, our method surpasses the LLM-only counterpart based on GPT-4 by +12 points in session accuracy and +21 points in per abnormal accuracy. Additionally, we conducted a detailed analysis of precision, recall, and F1 scores for A and B levels, which comprise about 10% of the dataset, yet misgrading these levels is considered a more serious error by physicians. Our results show that our model achieves high precision (0.866) for the A and B categories, with a slightly improved recall (0.542). These findings further confirm the advantage of neuro-symbolic computing in this application, particularly in maintaining high accuracy and reliability for critical classifications.

Table 8: ABUO results for baseline models and neuro-symbolic model

			A and B grading		
	ACC @session	ACC @abnorm	Precision	Recall	F1
Native gpt4	0.287	-	-	-	-
+abnormal	0.636	0.652	0.375	0.563	0.450
(INF-Med)ours	0.759	0.862	0.538	0.583	0.560
ours	0.800	0.898	0.866	0.542	0.667

3.5.1.5 Error Analysis

We note distinctive failure modes for LLM-only baselines and neuro-symbolic approach. We first abbreviated examples of common LLM-only baselines below.

LLM-only approach tends to make hallucinations. One common mistake for the LLM-only approach is hallucinating based on criteria that do not exist in the original guidelines. This happens often when there is no corresponding guideline for certain abnormal results for a specific grading. For example, the below example shows hallucination at steps 4, 5, and 6, based on criteria that do not exist in the original (we simplify the whole COT output by showing a snippet only on the hallucinated part). Even though it found the correct reference criteria, the conclusion was derived on the basis of the hallucinated criteria.

> Abnormal findings:
> Uric acid is elevated; Test Name: Uric Acid; Indicator: ↑; Reference Range:

155-357 μ mol/L; Value: 582.0 μ mol/L)

COT:
... Step 3: The uric acid level of 582.0 μ mol/L does not meet the Class A standard, which requires levels to be >600 μ mol/L. (correct)
Step 4: The uric acid level of 582.0 μ mol/L does not meet Class B standards.(hallucinated)
Step 5: The uric acid level of 582.0 μ mol/L meets Class U criteria as it is greater than 445 μ mol/L but below the threshold for Class B. ((hallucinated)
Step 6: Therefore, the elevated uric acid is classified under Category U. (hallucinated)

Ref: Level A: Renal FunctionBlood: Uric Acid: $>$ 600 μ mol/L

Conclusion: U

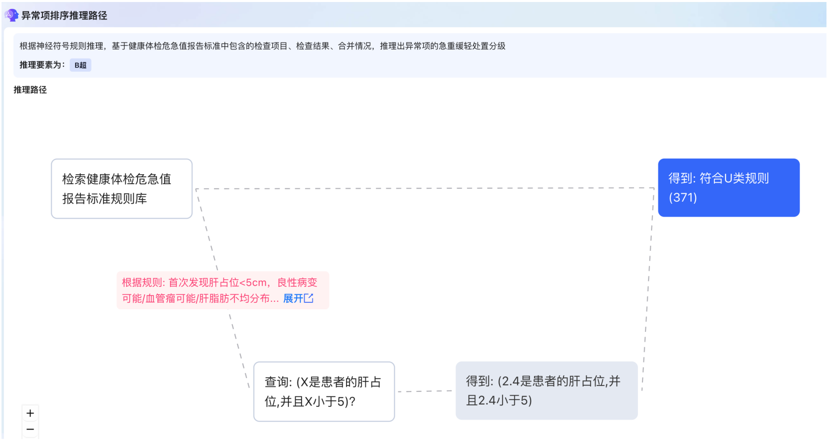
COT makes incorrect logic deductions when dealing with multi-step reasoning Another common type of mistake we find for the LLM-only model is an error in logic deductions with composite criteria. For example, *Multiple cystic lesions in the liver, with the largest diameter about 61mm* is classified as level O, with COT correctly concluding that a diameter of 61mm is less than 10cm, thus does not meet level B criterion. However, it does meet one criterion in level U related to liver cyst with diameter $\leq 5cm$, thus should be classified as level U.

We note our neuro-symbolic model significantly reduces the two types of errors mentioned above, thanks to the rigorous logic reduction process resulting from logic engine execution. However, the neuro-symbolic approach sometimes falls short in capturing implicit and vague information, e.g., criteria regarding ECG findings are often accompanied with the comment: *to be correlated with clinical and related examinations*, which is hard to translate to the logic formulation. This could result in missing information in the premises database leading to inaccurate conclusions.

3.5.1.6 Visualization

Finally, using the symbolic engine Ponens, we can retrieve all the reasoning steps, and visualize them at our specially designed front end for full explainability. We show one example in Figure13.

In summary, we apply a neuro-symbolic approach to leverage the strength of both LLM and symbolic computing to a medical grading task. At its core, we trade off the flexible expression space of NL for syntactically strict logic formulas in problem formulation, allowing us to use rigorous symbolic algorithms implemented in our logic engine Ponens for reasoning. This process significantly reduces the hallucination and error in multi-step reasoning, as well as improves the transparency of the system.



异常项排序推理路径

根据神经符号规则推理，基于健康体检危急值报告标准中包含的检查项目、检查结果、合并情况，推理出异常项的急重缓轻处置分级

推理要素为： B超

推理路径

检索健康体检危急值报告标准规则库

根据规则: 首次发现肝占位<5cm，良性病变可能/血管瘤可能/肝脂肪不均分布...

查询: (X是患者的肝占位,并且X小于5)?

得到: (2.4是患者的肝占位,并且2.4小于5)

得到: 符合U类规则(371)

Figure 13: Reasoning paths

3.5.2 Medical Knowledge Assistant

Another application of our INF-Med is Medical Knowledge Assistant, a tool designed for professional doctors and medical students. For doctors, it serves as a reliable reference tool, providing quick access to evidence-based medical information. For medical students, it acts as an educational resource, helping them grasp complex medical concepts and stay updated with current medical knowledge.

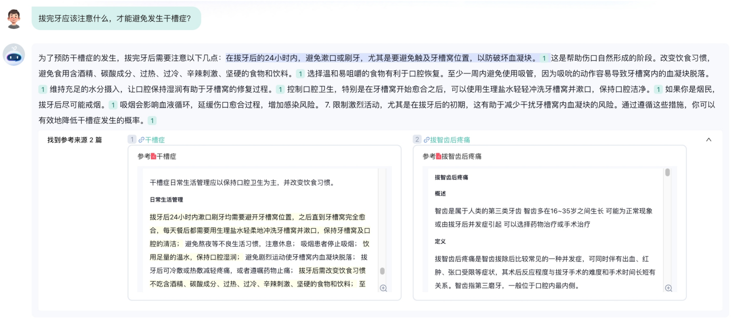
By leveraging the powerful Inf-Med medical language model, a comprehensive professional medical knowledge base, a Retrieval-Augmented Generation (RAG) system, and traceable long-context document question answering, Medical Knowledge Assistant ensures reliable, evidence-based responses, enhancing the accessibility and accuracy of medical information.

- **Medical Knowledge Base**: The foundation of the Medical Knowledge Assistant is a vast repository of medical information, including the latest medical guidelines, drug descriptions, and other publicly available resources. This extensive database ensures that the AI has access to most authoritative and comprehensive medical knowledge.

- **Retrieval-Augmented Generation (RAG)**: RAG combines the strengths of information retrieval and text generation. When a query is sent to the assistant, the RAG system retrieves relevant documents from the knowledge base and uses them to generate a coherent and contextually accurate response. This hybrid approach enhances the reliability of the information provided.

- **Traceable Long-Context Question Answering**: One of the standout features of the Medical Knowledge Assistant is its ability to provide answers based on the long reference that are not only detailed but also cite their sources. This traceability ensures that users can verify the information and refer to the original documents for further reading, thereby enhancing the credibility of the responses.

The trustworthiness and reliability of the Medical Knowledge Assistant are paramount. By leveraging the latest medical guidelines, the assistant ensures that all information provided is grounded in the most current and credible sources. The use of RAG further enhances this reliability by ensuring that each answer is not only generated based on the language model's capabilities but also from relevant and authoritative sources,as shown in the Figure14 below



拔完牙应该注意什么，才能避免发生干槽症?

为了预防干槽症的发生，拔完牙后需要注意以下几点：在拔牙后的24小时内，避免漱口或刷牙，尤其是要避免触及牙槽窝位置，以防破坏血凝块。1 这是帮助伤口自然形成的阶段。改变饮食习惯，避免食用含酒精、碳酸成分、过热、过冷、辛辣刺激、坚硬的食物和饮料。1 选择温和易咀嚼的食物有利于口腔恢复。至少一周内避免使用吸管，因为吸吮的动作容易导致牙槽窝内的血凝块脱落。1 维持充足的水分摄入，让口腔保持湿润有助于牙槽窝的修复过程。1 控制口腔卫生，特别是在牙槽窝开始愈合之后，可以使用生理盐水轻轻冲洗牙槽窝并漱口，保持口腔洁净。1 如果你是烟民，拔牙后尽可能戒烟。1 吸烟会影响血液循环，延缓伤口愈合过程，增加感染风险。 7. 限制激烈活动，尤其是在拔牙后的初期，这有助于减少干扰牙槽窝内血凝块的风险。通过遵循这些措施，你可以有效地降低干槽症发生的概率。1

找到参考来源 2 篇

1 干槽症

参考干槽症

干槽症日常生活管理应以保持口腔卫生为主，并改变饮食习惯。

日常生活管理

拔牙后24小时内漱口刷牙均需要避开牙槽窝位置，之后直到牙槽窝完全愈合，每天餐后都需要用生理盐水轻柔地冲洗牙槽窝并漱口，保持牙槽窝及口腔的清洁； 避免熬夜等不良生活习惯，注意休息； 吸烟患者停止吸烟； 饮用足量的温水，保持口腔湿润； 避免剧烈运动使牙槽窝内血凝块脱落； 拔牙后可冷敷或热敷减轻疼痛，或者遵嘱药物止痛； 拔牙后需改变饮食习惯不吃含酒精、碳酸成分、过热、过冷、辛辣刺激、坚硬的食物和饮料； 至

2 拔智齿后疼痛

参考拔智齿后疼痛

拔智齿后疼痛

概述

智齿是属于人类的第三类牙齿 智齿多在16~35岁之间生长 可能为正常现象或由拔牙后并发症引起 可以选择药物治疗或手术治疗

定义

拔智齿后疼痛是智齿拔除后比较常见的一种并发症，可同时伴有出血、红肿、张口受限等症状，其术后反应程度与拔牙手术的难度和手术时间长短有关系。智齿指第三磨牙，一般位于口腔内最内侧。

Figure 14: INF-Med application: medical knowledge assistant

4 INF-LLM for Finance

4.1 High-quality Finance Data for Continuous Training

We provided a comprehensive set of diverse structured and unstructured financial data and general data for continuous training our financial LLM. In serving this mission, we have curated a set of financial documents that were either created internally or acquired from external sources. We utilize this extensive collection of curated and maintained documents to create our financial datasets, which consists of financial news, financial books, financial reports, research reports and announcements of listed companies, and financial exams relevant to financial markets. Constructing the financial dataset mainly involved in the following three steps:

- **Data crawling and collection**: We collect our raw data from various sources, such as websites, journal articles, authoritative guidelines, books, educational materials, and etc. We use some classic OCR technology to parse some types of materials such as some pdf books into structured information.
- **Data cleaning and filtering**: Although pretraining data for LLMs is sourced from authoritative and up-to-date sources, some noise remains

Dataset	sources	Tokens for training	language
financial books	press	5.4	en
financial reports	press	0.392	zh
research reports	web-scraped	0.5	zh
cfa-koolearn-seed	web-scraped	0.388	zh
financial exams	web-scraped	1.4	en/zh
financial news	web-scraped	4.3	en/zh
webtext-cc-like-finance	data augmentation	7	en/zh

Table 9: An overview of financial corpus for continuous training procedure.

Dataset	sources	Tokens for training	language
refine-cc	web-scraped	0.9847	en
ultratextbook	press	5	en
openhermes	web-scraped	0.72	en
wiki	web-scraped	1	en
paper-cnki	web-scraped	4	en
book-libgen	web-scraped	4	en
baidu-baike	pressd	4	zh
cosmos	web-scraped	10	en
instructions	web-scraped	3	en
open-web-math	web-scraped	8	en

Table 10: An overview of the general corpus for continuous training procedure.

when adapting this data for training. Enhancing control and improving data quality have become essential factors for the success of LLMs. This study has developed a comprehensive data quality control process focused on elevating the standards for data quality, which mainly involved in 3 steps: data filtering by some rules such as whether contain ads in websets or unuseful urls, data control by some quality signals, data deduplicated by minihash algorithm.

- **Data augmentation**: Besides the collected raw data, experiments demonstrate that data augmentation also improve LLM's performance. We augmented our data by some approaches. For instance, we extract knowledge from the exams by prompts engineering based on LLM. Moreover, using questions in some exams as query, we recall the relevant knowledge from financial websets in common crawl dataset to supplement our knowledge.

More details of the data we used in the experiments of this study can be found in Table 9.

Besides the financial data, we also collect plenty of general data to improve the generalization ability of our model. We use some widely known and available public datasets in our training corpus. For training LLMs, it is crucial to scrape news from a variety of sources to capture different writing styles, terminologies, and perspectives. Digitized literary works also provide a wealth of linguistic data spanning centuries. More details of the corpus data we used in the experiments of this work can be found in Table 10.

4.2 Data for Instruction Alignment

In this section, we will present our curated financial instruction tuning dataset. Notably, our dataset is characterized by a combination of high-quality and diverse instructions, encompassing both general complex instructions and those tailored specifically for financial tasks. This aspect is paramount in the training of large-scale financial models, as it provides a rich and varied foundation for the model to learn from.

General Instructions We utilized a vast amount of open-source instruction data to train the base model. We gathered approximately 10 million open-source multi-lingual instruction fine-tuning data, processed them through parallel cleaning and rigorous filtering. The aim of this stage was to cover the diversity of instructions and enhance the model's ability to follow instructions. Furthermore, we augmented our training dataset by synthetically generating a substantial portion of more complex instruction data through a combination of automated and semi-manual processes.

Financial Instructions The financial instructions dataset is underpinned by a suite of critical tasks, including financial examinations like the CFA, classification, information extraction, question answering (QA), reading comprehension, and summarization. Classification tasks are essential for sentiment analysis in financial texts, utilizing datasets such as the Financial Phrase Bank (FPB) for labeled sentiment and FiQA-SA for sentiment prediction. Additionally, information extraction plays a vital role in identifying entities and events within financial texts, providing context and insights that are invaluable for financial decision-making. Question answering in finance is pivotal, with Financial QA focusing on user queries about financial topics and Table QA concentrating on extracting information from structured financial data in tables. Collectively, these tasks harness the power of natural language processing to enable more efficient and informed financial analysis and decision-making.

Financial Application Instructions The Financial Application Instructions comprises two products we've developed: the Financial Report Review and the Knowledge Assistant. The Financial Report Review is crafted for financial analysts, using large language models to streamline financial report writing by automating framework generation, commentary, and incorporating both standard and unique financial metrics. It enhances the report production process with large language models, blending AI capabilities with human expertise. The Knowledge Assistant, designed as a personalized tool for financial customers, employs large models for knowledge management, facilitating search and organization of financial information. It performs classification to interpret user intent, refines data through extraction tasks, and generates text for reading com-

Table 11: A summary of datasets from different sources

Type	**Source**	**Description**
General Instructions	Open-source	Include a wide range of instructions, covering various domains and scenarios, and are annotated with relevant labels or metadata
	Synthetic	Typically designed to cover a specific set of tasks of financial domain, and are crafted to be concise, clear, and unambiguous
Financial Instructions	Exam	Financial examination datasets, including CFA.
	Classification	Financial sentiment analysis, news headline classification, etc.
	Information Extraction	Name Entity Extraction, event extraction, etc.
	Question Answering	financial question answering, table question answering
	Numerical Reasoning	Complex numerical reasoning in financial data
	Reading Comprehension	Understand and analyze financial documents, such as financial reports, research reports, and public announcements, and provide answers to user queries
	Summarization	Condense and summarize vast amounts of financial data into meaningful and actionable insights
Financial Application Instructions	Financial report review	Financial indicator extraction, unique metric extraction, the generation of writing frameworks, metric filling, and review creation
	Knowledge Assistant	Intent classification, data pathway selection, company name extraction, time extraction, etc.

prehension and summaries, offering tailored, concise insights. Together, these products represent our practical application of AI in the financial sector, aimed at improving efficiency and decision-making.

4.3 Training

Further domain-specific pretraining significantly boosts the proficiency of large language models in their respective fields. We chose the open-source Qwen2-72B-Base model, and conducted continuous training using our proprietary dataset tailored to our domain. This process has yielded a state-of-the-art financial language model, named as INF-Fin-Base, which demonstrates remarkable performance within the financial sector. In detail, we set the maximum learning rate to 1e-4 and used the cosine decay learning rate scheduler with linear warmup. The learning rate is warmed up in the first 1000 steps. The final learning rate is 1e-5. We trained our model for one epoch. We used the AdamW optimizer, and set β_1 to 0.9, β_2 to 0.95, and weight decay to 0.1.

We further fine-tuned INF-Fin-Base using the proprietary instruction dataset that covers various NLP tasks. For the chat model INF-Fin, we fine-tuned with 1200 steps, utilizing the AdamW optimizer. The batch size is set to 32, the initial learning rate is 5e-6, and the weight decay is 0.1. The learning rate is linearly warmed up in the first 240 training steps. The maximum input text length is set to 4096. The training process was conducted on 8 A100 80GB GPUs.

4.4 Model Evaluation

4.4.1 Evaluation Dataset

To objectively evaluate the capabilities of current industry-leading models in financial tasks, a self-constructed dataset, along with two mainstream open-source financial datasets, were selected. The details of the dataset are shown in Table 12.

- **CFA** is the Chartered Financial Analyst (CFA) exam dataset developed by Shanghai Academy of Artificial Intelligence for Science (SAIS) to evaluate LLMs' financial analysis and investment management knowledge. This dataset consists of two levels of CFA exam questions, each with its own focus and set of topics. In detail, the Level 1 exam consists of 400 questions, the Level 2 exam contains 300 questions.

  The Level1 Exam comprises a broad spectrum of fundamental concepts and principles, assessed through single-choice questions that cover topics such as Ethical and Professional Standards, Quantitative Methods, Economics, Financial Reporting and Analysis, Corporate Finance, Equity Investments, Fixed Income, Derivatives, Alternative Investments, and Portfolio Management.

Dataset	Number of Items	Language	Type
CFA	700	en	single-choice
FinanceIQ	7123	zh	single-choice
FinEval	1151	zh	multiple-choice

Table 12: An overview of financial evaluation dataset.

The Level2 Exam introduces more complex question formats, including text, tables, charts, and financial statements, requiring LLMs to interpret and analyze the data to answer questions accurately.

All the questions are based on official textbooks written by CFA qualified practitioners, mock questions by training institutions, and rewritten questions based on retelling CFA exam questions with participants.

- **FinanceIQ** [8] is organized into 10 major financial categories and 36 subcategories, covering authoritative exams such as Certified Public Accountant (CPA), Tax Advisor, Economist, Banking, Fund, Securities, Futures, Insurance (CICE), and Financial Planner. Additionally, it includes the "Financial Mathematics" subject from actuary exams to test high-difficulty financial mathematics problems.
- **FinEval** [58] is a benchmark designed to evaluate financial domain knowledge in large language models (LLMs), is based on quantitative foundational methods. It comprises 8,342 question types closely aligned with real-world application scenarios, including multiple-choice questions, subjective open-ended questions, objective short-answer questions, reasoning planning, and retrieval-based QA. These questions encompass topics such as Financial Academic Knowledge, Financial Industry Knowledge, Financial Security Knowledge, and Financial Agent. To ensure a comprehensive assessment of model performance, FinEval integrates both objective and subjective evaluation criteria, such as Accuracy, Rouge-L, and expert evaluation guidelines, employing zero-shot and few-shot methods for evaluation.

4.4.2 Experimental Results

In this experimental section, we assess the performance of three models, GPT-4 Turbo, Qwen2-72B and INF-Fin, across various financial metrics, including CFA Level 1 (CFA-L1), CFA Level 2 (CFA-L2), FinanceIQ, and FinEval. The results are presented in Table 12, which offers a comprehensive evaluation of the models' capabilities.

Regarding the CFA metrics, which focus on evaluating financial analysis performance, we observe that INF-Fin achieves the highest scores across all

Model	CFA-L1	CFA-L2	FinanceIQ	FinEval
GPT-4 Turbo	72.250	55.000	65.710	70.460
Qwen2-72B	65.250	49.667	77.000	86.707
INF-Fin	**77.750**	**57.333**	**97.782**	**92.441**

Table 13: Comparison of model performance on financial benchmark datasets.

categories. Specifically, INF-Fin's CFA-L1 score of 77.750 is significantly higher than both GPT-4 Turbo's 72.250 and Qwen2-72B's 65.250, indicating a superior ability to perform basic financial analysis tasks. Similarly, INF-Fin's CFA-L2 score of 57.333 outperforms the other two models, suggesting a stronger proficiency in handling more complex financial calculations.

Focusing on the FinanceIQ metric, which evaluates the models' financial intelligence, INF-Fin again demonstrates superior performance with a score of 97.782. This score is significantly higher than GPT-4 Turbo's 65.710 and Qwen2-72B's 77.000, highlighting INF-Fin's enhanced capability in understanding and analyzing financial concepts and scenarios.

The FinEval metric, which measures the overall financial evaluation performance of the models, exhibits a similar trend. INF-Fin achieves the highest score of 92.441, significantly outperforming GPT-4 Turbo's score of 70.460 and Qwen2-72B's score of 86.707. These results further confirms INF-Fin's comprehensive superiority in financial tasks, likely due to its specialized design and training for financial analysis and evaluation.

In summary, the experimental results demonstrate that INF-Fin outperforms both GPT-4 Turbo and Qwen2-72B across all evaluated financial metrics. This indicates that INF-Fin's specialized design and training have effectively enabled it to achieve superior performance in financial analysis and evaluation tasks.

4.5 Application Case Studies

4.5.1 Finance Report Commentary

Financial professionals today often face the daunting task of sifting through extensive data to extract meaningful insights—a process that can be time-consuming and prone to human error. To address the growing need for efficient, accurate, and insightful analysis of financial reports and alleviate these challenges, we developed the Financial Report Commentary (FRC) application, leveraging our advanced INF-Fin. This application automates the generation of comprehensive commentary on financial reports, statements, and related documents, providing users with detailed, contextually accurate insights that highlight key indicators, trends, and significant changes. The FRC tool not only enhances the efficiency and accuracy of financial analysis but also empowers analysts, investors, and advisors to make more informed decisions. The framework of our application is shown in Fig 15. In the following paragraphs, we will introduce how we emulate the precision and insight of professional analysts in our reports and ensure the content remains trustworthy.

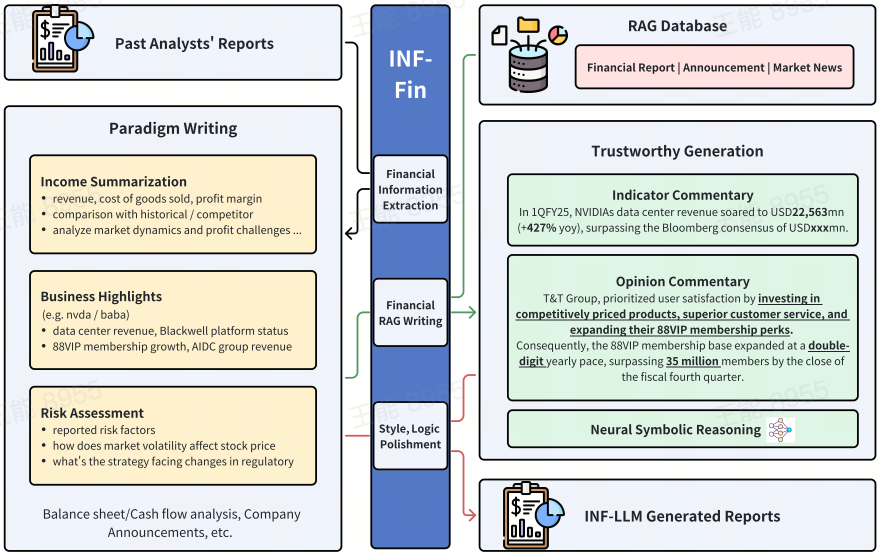

Past Analysts' Reports
Paradigm Writing
Income Summarization
• revenue, cost of goods sold, profit margin
• comparison with historical / competitor
• analyze market dynamics and profit challenges ...
Business Highlights
(e.g. nvda / baba)
• data center revenue, Blackwell platform status
• 88VIP membership growth, AIDC group revenue
Risk Assessment
• reported risk factors
• how does market volatility affect stock price
• what's the strategy facing changes in regulatory
Balance sheet/Cash flow analysis, Company Announcements, etc.
INF-Fin
Financial Information Extraction
Financial RAG Writing
Style, Logic Polishment
RAG Database
Financial Report | Announcement | Market News
Trustworthy Generation
Indicator Commentary
In 1QFY25, NVIDIAs data center revenue soared to USD22,563mn (+427% yoy), surpassing the Bloomberg consensus of USDxxxmn.
Opinion Commentary
T&T Group, prioritized user satisfaction by investing in competitively priced products, superior customer service, and expanding their 88VIP membership perks.
Consequently, the 88VIP membership base expanded at a double-digit yearly pace, surpassing 35 million members by the close of the fiscal fourth quarter.
Neural Symbolic Reasoning
INF-LLM Generated Reports

Figure 15: Framework of Finance Report Commentary

Paradigm Writing

When drafting a Financial Report Commentary, the primary considerations should be the content to be included, the indicators to be covered, and the commentary viewpoints to be provided. In this application, we implemented a generalizable writing framework that is based on the extraction of writing style and key contents from historical financial documents and utilizes them for writing and polishing new financial documents in similar patterns. We refer to this process as Paradigm Writing. The framework consists of three main steps, each deeply integrated with INF-Fin:

1. *Financial Information Extraction*: Leveraging the powerful and precise financial information extraction capabilities of INF-Fin gained through instruction alignment, we can employ a model-native approach to extract valuable key indicators and business segments worth commenting on from past commentaries written by analysts for each company, which can be done both online and offline. The indicators are usually classified into financial indicators and company-featured indicators to serve the RAG process better. The generated comment questions are designed to be neutral and unbiased, avoiding strong ties to specific events, which allows for

reuse in various contexts. In addition, we're able to understand the analysts' habitual writing order and content arrangement through this step. For example, when to insert descriptions of cost efficiency and when to include a risk analysis of the company, which is highly beneficial for the subsequent polishing of our writing style and content organization.

2. *RAG Writing*: After analyzing the writing patterns of past analysts, based on the company's financial reports and announcements within the current reporting period, we recall the relevant content and utilize INF-Fin to write a preliminary commentary through Retrieval-Augmented Generation (RAG). Given financial experts' prior knowledge, we select the most suitable data sources to maximize the quality of the material and enhance the overall writing quality and trustworthiness. For writing on different types of indicators, different retrieval indices are utilized as sources for RAG. For instance, for common financial indicators, we opt to recall contents from financial statements like the income statement or balance sheet, instead of searching tables in the parsed financial report pdf. Similarly, when writing commentary questions, we select specific sections of the financial report to reduce retrieval difficulty. In certain reporting periods where financial reports have little content, we increase the weight of sourcing from company announcements to enrich the content.

3. *Style and Logic Polishment*: After obtaining the preliminary draft, we use a few-shot in-context learning (ICL) approach to polish the style and logic of the commentary. The few-shot format allows us to express our writing style requirements and impose constraints on the model's behavior in specific situations in a more intuitive and configurable manner. INF-Fin, trained with similar scenario data, exhibits better adherence to guidelines in few-shot polishment scenarios. The few-shot examples can come from carefully selected text pairs written by professional analysts before and after polishment, or they can be directly extracted from historical commentary in the first step. After data anonymizing (to prevent the model from copying), historical examples can be used as awesome references. The polished commentary is significantly smoother and more logically coherent than before the polishment. Additionally, the amount of redundant and verbose language produced by the model will be notably reduced, making the output closer to the style of the desired analyst and more human-like.

Paradigm Writing can also be effectively extended to other writing scenarios in the financial domain. Besides financial report commentary, it can be used for earnings call summaries, prospectuses, industry trend analyses, regulatory compliance reports, etc. We consider its application in commentary as a starting point and are exploring its deployment in a broader range of scenarios.

China A-Shares Commentary Sample

营收增长驱动业绩提升，多因素影响净利润表现
公司实现营业收入329.33亿元，同比9.08%；归母净利润5.20亿元，同比39.80%；扣非归母净利润2.74亿元，同比-33.91%。单季度来看，公司实现营业收入104.22亿元，同比下降7.05%；归母净利润2.00亿元，同比下降34.23%；扣非归母净利润8350.78万元，同比下降54.03%。公司经营业绩显著提升，主要驱动因素有五点：一，项目交付量增长推动营业收入增加，伴随而来的是营业成本的合理增涨；二，加强市场开拓和营销活动，销售费用随之上升；三，管理费用增加主要源于咨询投入的增加；四，财务费用减少，得益于利息支出的控制和汇兑收益的提升；五，公司持续加码研发投入，彰显其强化核心竞争力的决心。资产减值损失增加，主要由于计提存货跌价准备，对净利润造成一定压力。总结而言，公司的盈利能力和市场定位调整是业绩增长的核心动力。

重型装备订单增长强劲，钢结构与船舶运输业务稳中有进
分产品看，重型装备营业收入为54.23亿，同比增长75.75%；毛利率为5.23%，同比下降2.35pct。钢结构及相关收入为31.66亿，同比增长15.92%；毛利率为8.92%，同比增长5.38pct。船舶运输及其他营业收入为13.88亿，同比下降25.50%；毛利率为23.33%，同比增长3.92pct。公司在重型装备领域展现专业技术优势，成功斩获国内首个旋转式打桩船项目及多艘高级起重船订单，巩固了在海工装备市场的领先地位。钢结构业务方面，公司不仅承担了厦门翔安大桥和澳大利亚西门隧道主桥等重大项目，更凭借卓越表现赢得多项行业荣誉，全球钢桥梁行业中影响力日益增强。公司在船舶运输领域同样表现出色，拥有一支包含多艘大型运输船的船队，确保重型设备高效安全地运抵全球各地。这些成就充分证明了公司在高端、智能和绿色装备制造方面的坚定投入与显著成效。

营收增长带动毛利率微升，费用结构调整影响净利率
2023年公司毛利率达13.52%，同比增长0.12pct，净利率为1.96%，同比下降0.02pct。单季度毛利率为15.74%，同比下降0.93pct，净利率为1.94%，同比下降1.37pct。全年销售费用率0.64%，同比增长0.06pct；管理费用率6.56%，同比增长0.09pct；研发费用率3.98%，同比增长0.28pct；财务费用率1.66%，同比下降0.91pct。报告期，公司毛利率变动主要源于营业收入增长9.08%，达329.33亿元，而营业成本增长8.93%，至284.80亿元。这一变化主要由项目交付量增加驱动，营业收入的提升带动营业成本同步增加，但成本增长率略低于营收，从而对毛利率造成一定影响。销售费用、管理费用、财务费用及研发费用的结构调整也对公司的盈利构架产生了相应作用。

战略聚焦五大领域，打造一流装备制造企业
公司遵循“1544”战略，定位为全球领先的科技型、管理型、质量型装备制造企业。核心业务聚焦五大领域，深化四大主线工作，并强化四种思维方式，以达成世界一流企业的愿景。在产业发展上，公司战略性投入智能装备，如振华海通智能装备有限公司，并审慎开展主业相关的投融资活动，涉及股权投资和稳健收益项目。公司积极探索装配式建筑等新兴市场，依托自身优势在海洋经济、民生工程和新能源领域实现差异化竞争。在经营策略中，公司坚持以人为本，紧贴行业动态，把握宏观经济的新机遇，致力于推动高质量的可持续发展。

Figure 16: Showcase of Chinese A-Share Commentary

Trustworthy Generation

The most crucial aspect of financial writing is the trustworthiness of the content. Next, we will introduce how we apply the model's trustworthy capabilities to Indicator Commentary, Opinion Commentary, and through neuro-symbolic reasoning.

- **Indicator Commentary**

  1. *Context Filtering*: According to the study by F. Cuconasu et al. [5], apart from the truly relevant documents we need, related documents cause more performance degradation in LLM's output compared to completely irrelevant documents as distractors. This is even more critical in financial scenarios, where numerical accuracy is of utmost importance, especially given the complex, cross-temporal, and cross-industry indicators, which are extremely prone to confusion. To address this issue, we leverage the financial information extraction capability of INF-Fin to identify the required reporting period and

relevant industry/product. Based on it we can collect as many related but not needed distracting indicators and exclude them or lower the ranking during the context retrieval phase.

2. *Trustworthy Numerical Filling*: The accuracy of numerical filling primarily stems from the inherent capabilities of the INF model. As mentioned in Section 4.2, the extensive Financial QA training data used in the model's instruction alignment, especially Table QA in this case, has endowed the model with powerful abilities to clearly distinguish between different financial indicators. Experiments have demonstrated that INF-Fin outperforms general LLMs in numerical filling and has a significant advantage in differentiating similar financial indicators or indicators from different reporting periods. In addition, supplementary training data has been included to guide the model that, when asked to write an indicator unfounded in the source, it should follow the instructions but replace the actual values with N/A. This approach ensures that the user's writing requirements are met while maintaining the trustworthiness of the content.
3. *Post-fill Verification and Validation*: After initially generating commentary on the indicators, we perform style and logic polishment on the results. If any numbers are found to be unreasonable, the corresponding indicators will be discarded, and the sentences will be revised accordingly. Once the model has completed the generation, we conduct a final validation with the writing sources. If any of the corresponding values do not exist in the source materials, the indicators will either be automatically discarded or replaced with N/A to maintain overall trustworthiness.

- **Opinion Commentary**

  The technical points of trustworthiness in Opinion Commentary are closely aligned with those in the Indicator Commentary. Both rely on pre-filtered, high-quality data sources and the robust native capabilities of the INF-Fin to ensure accurate understanding and citation of the original material. A key technical highlight is its capability for source tracing and verification of the original content. Since opinion statements must be based on factual descriptions from financial reports or announcements, supplemented by simple causal inferences, any hallucinations or fabrications by the model are intolerable. We perform N-gram similarity checks between the model-generated content and the sentences from the source materials. Sentences that do not meet the set similarity threshold are deleted to ensure traceability and accuracy.

- **Neuro-symbolic Reasoning**

  Based on the neuro-symbolic computing engine introduced in Section 2.3, we utilized neuro-symbolic reasoning in both the commentary generation and quality control stages. During the generation phase, we rely on prior knowledge provided by financial experts to reason about specific indicators, such as whether they meet a certain threshold or reflect a particular trend, and then derive conclusions. We use this reasoning process and the derived conclusions as writing material, which we feed into the LLM. This approach ensures that the generated descriptions are more logical and trustworthy. During the quality control stage, we reuse the key information extraction capabilities of INF-Fin to extract key information back from the generated content. We then apply expert-written quality control rules to verify the integrity and logic of the indicators and commented viewpoints.

U.S. Stock Commentary Sample

NVIDIA 1QFY25 beats expectations, records 262% YoY revenue surge
NVIDIA's 1QFY25 revenue amounted to USD26.04 billion, demonstrating a remarkable 262% year-over-year surge. Non-GAAP earnings per share (EPS) reached USD6.12. The non-GAAP gross profit margin climbed to 78.9%, reflecting a quarterly improvement of 2.2 percentage points. NVIDIAs financial performance underscores its robust position in the market.

NVIDIA splits stock, boosts dividend, lifts share price
NVIDIA Corporation has executed a ten-for-one forward stock split and announced a substantial 150% increase in its quarterly cash dividend, escalating from $0.04 to $0.10 per share following the split. This strategic move positively impacted affiliated industry partners, including Supermicro, Dell, Vertiv, and Micron. Over the reported quarter, NVIDIA demonstrated its dedication to shareholder returns by repurchasing shares worth $8.0 billion and distributing quarterly dividends amounting to $98 million.

Data center revenue surges, AI computing drives growth
In Q1 FY2025, NVIDIA's data center revenue soared to USD 22.6 billion, representing a staggering 427% year-over-year growth. Within this segment, computing revenue hit USD 19.4 billion, also up 478% annually, while network revenue climbed to USD 3.2 billion, marking a significant 242% year-over-year increase, primarily driven by InfiniBand solution sales. The foundation for AI computing at a trillion-parameter scale lies in NVIDIAs Blackwell platform, with the DGX SuperPOD exemplifying its prowess in generative AI supercomputing. The Hopper GPU computing platform plays a crucial role in large language models, recommendation engines, and generative AI applications across major cloud providers, backed by their substantial investments in NVIDIA AI infrastructure, which accounted for around 40% of the companys data center revenue in the first quarter of FY2025.

Hopper GPU demand robust, Blackwell GPU launch awaited, supply constraints persist
Based on recent updates, lead times for the H100 have improved, though the H200 continues to face supply limitations. NVIDIA underscores the robust demand for its Hopper GPU architecture to alleviate concerns about potential revenue growth deceleration as clients eagerly anticipate the launch of the Blackwell GPU. The company plans to introduce Blackwell samples in Q2 of FY2025, targeting a production boost in the latter half of the fiscal year. Anticipating high demand for Blackwell, NVIDIA predicts supply constraints to extend into the following fiscal year, potentially affecting FY2025 earnings.

Figure 17: Showcase of U.S. Stock Commentary

Show Cases

We present a Chinese-written A-share financial report commentary and an English-written U.S. stock financial report commentary generated with our FRC application and INF-Fin. Both are generated based on the same framework, using different language versions of prompts.

China A-Shares Commentary

Fig 16 presents our generated commentary on 振华重工 's 2023 annual financial report. In this commentary sample, we focused on four main sections: operational performance, core business, cost efficiency, and future strategy. The subjects we pick are all based on past analysts' reports on this company. For indicators in sections 1 and 3, the data mainly comes from official financial statements, the numbers of core business are mainly derived from the annual financial report and are all traceable. And there are always comments after the stats, expressing the opinion derived from the numbers, either directly from the report or through neuro-symbolic reasoning.

U.S. Stock Commentary

For the U.S. Stock Commentary showcase shown in Fig 17, we pick the latest 1QFY25 financial report of NVIDIA. Our primary writing materials are sourced from the latest 10-Q SEC Filing. To supplement with non-GAAP data and content, we have also included NVIDIA's Q1 CFO Commentary materials in the writing resources. When extracting from historical research reports, we also found that analysts tend to insert comparisons with consensus or expectations from Bloomberg or other institutions. However, since we currently do not have access to these sources and to maintain overall trustworthiness, we don't have them included. The subjects included are different from the Chinese commentary, due to different past commentaries consumed by the model.

4.5.2 Financial Knowledge Assistant

4.5.2.1 Overview

In the vast realm of financial knowledge, precisely excavating and interpreting deep insights poses a significant challenge. To address this challenge, we meticulously designed and constructed a highly specialized and accurate Financial Knowledge Assistant, based on the LLM Native architecture, dedicated to delivering exceptional financial information services. Our Knowledge Assistant comprises three key architectural components: the Query Understander, the Evidence Finder, and the Answer Generation and Optimization Engine. These components work in tandem to ensure efficient information retrieval and precise responses.

Furthermore, our Knowledge Assistant includes the innovative Query Understanding All-in-One Model, which significantly enhances the understanding of user intent through an integrated query rewriting method, thereby improving the recall rate for relevant unstructured documents. In addition, the Semi-structured for Knowledge Base strategy for semi-structured methods has enhanced retrieval efficiency and robustness. The Long-Context Generation catalog tree strategy ensures that the generation of long texts is both structured and content-rich. Lastly, the Answer Tracer guarantees that the generated answers are transparent and traceable.

Our Financial Knowledge Assistant provides researchers and professionals in the financial domain with a reliable, efficient, and transparent information service platform, representing an advanced tool for knowledge exploration in the field of finance.

4.5.2.2 Architecture

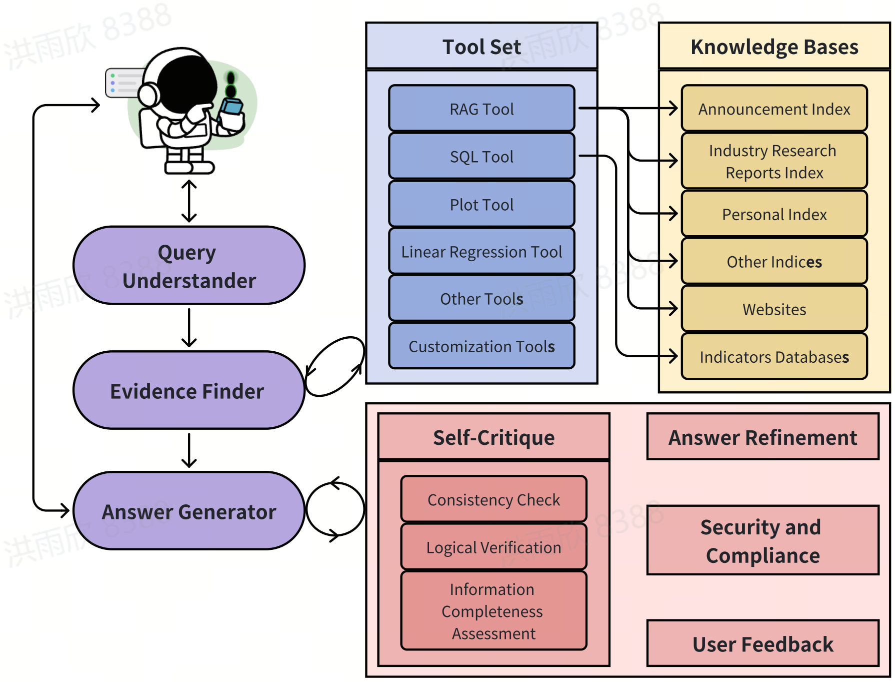


Figure 18: Architecture of Financial Knowledge Assistant

The architecture of the Financial Knowledge Assistant aims to achieve efficient and accurate processing and analysis of financial information. The primary architectural components of the system and their workflows as shown in Figure 18.

Firstly, Financial Knowledge Assistant analyze the user's queries in depth, identifying its underlying intent, including the identification of the company or industry, definition of the time range, and accurate parsing of pronouns. Conduct contextual analysis to understand industry terms, professional vocabulary, and common abbreviations, ensuring information is correctly linked to real-world entities. After comprehending the query content and context, if the question is unclear, continue interacting with the user to confirm the query intent, avoid misunderstandings, and reconstruct the original query based on the understanding of the user's intent. Then, dynamically select and invoke appropriate tools or algorithms based on the user's query. These tools include computation tools, plotting tools, and machine learning algorithms, all seamlessly integrated with the agent, supporting complex data analysis and visualization needs. The agent can also access a knowledge base, which stores a wealth of financial data and information that can be retrieved and referenced when generating answers. Finally, based on the query and the data retrieved, quickly generate an initial answer. Then use self-critique capabilities to evaluate and optimize the answer by supplementing missing information, correcting logical errors, and enhancing readability and accuracy, ensuring its accuracy and completeness. After receiving the answer, the user can provide feedback through a feedback loop, which the Financial Knowledge Assistant uses to learn.

4.5.2.3 Key Features

1. **Query Understanding All-in-One Model** The Query Understanding All-in-One Model is an efficient and comprehensive method designed for financial knowledge assistants to understand user queries effectively. This model encompasses fundamental features such as time and entity extraction, SQL querying, and introduces an innovative query rewriting technique. We employs an integrated understanding strategy that avoids the intricate decision-making process of LLM Agents in selecting specific rewriting methods. Moreover, our research indicates that extracting key information before performing query rewriting is beneficial. This technique significantly enhances the understanding of user intent, thereby improving the recall rate of relevant unstructured documents. This is particularly crucial in the financial sector, where data typically includes a vast amount of unstructured content like news reports and market analysis. The Query Understanding All-in-One Model not only improves the efficiency and accuracy of financial knowledge assistants in understanding user queries but also provides financial professionals with more reliable and efficient information services. Additionally, it offers new perspectives and methods for prompt engineering in other fields.

2. **Semi-structured for Knowledge Base** In the process of constructing the Financial Knowledge Assistant, we recognized that while the Query Understander is crucial for enhancing retrieval effectiveness, its robustness is constrained by the quality of text embedding and tokenization. To overcome these limitations, we adopted a semi-structured processing strategy to enhance the retrieval efficiency and robustness of texts in the index database. Initially, we identified key information within the text, such as dates, company names, industry terms, locations, industry backgrounds, and market environments. This aids in precisely matching user queries with relevant information in the knowledge bases, ensuring that retrieval results are highly relevant to user needs. To address deeper alignment issues, we utilized methods of hypothetical question construction

and text summarization. By constructing potential query scenarios offline and compressing lengthy texts into concise summaries, we achieved a higher level of alignment. These semi-structured measures have improved retrieval accuracy and enhanced the robustness and reliability of the Financial Knowledge Assistant in handling complex financial queries. This constitutes a comprehensive strategy, ensuring effective alignment and matching between user queries and textual content in the knowledge bases.

3. **Long-Context Generation** Long-Context Generation serves as the key capability for performing in-depth financial analysis and providing comprehensive responses. However, due to the probabilistic nature of word generation in LLMs, generating extended texts can lead to issues such as logical discontinuities, topic drift, or information redundancy. To address these challenges, we employed a directory tree strategy, ensuring the generated text is both structured and rich in content.

   Building the directory tree starts with a thorough analysis of the user's query, identifying key points to serve as root nodes. Utilizing the semantic understanding capabilities of LLMs, we generate relevant sub-topics, forming the first level of branches in the tree. This process is recursive, continuing until a complete thematic structure is generated. Once the directory tree is constructed, we move to the content generation phase. Unlike traditional linear generation, the directory tree strategy allows for modular text generation. For each node, we independently generate the relevant content and then integrate it into the overall text. This approach enhances content organization and effectively addresses issues of information redundancy and topic drift.

4. **Answer Traceability** Answer traceability is achieved by meticulously documenting each step of the answer generation process, including data sources, analysis procedures, and the generation logic. This transparency enhances user trust in the outputs and provides financial analysts with the capability to verify and further analyze the answers.

   The main functions of the Answer Tracer include source tracking, which records the original data sources for each answer paragraph, such as index libraries, databases, or public web searches; semantic similarity analysis, which determines the correspondence between each sentence in the answer and the original data blocks, identifying how information is extracted and transformed; indicator data extraction, which identifies and extracts key financial indicators from the answers and clearly states their sources; generation process documentation, which records each step of the answer generation process, including self-evaluation results and optimization processes, offering users a complete history of answer generation; and visual representation, which uses user interface elements like timelines and flowcharts to visually display the sources and generation process of the answers.

4.5.2.4 Evaluation

In the research and development of the Financial Knowledge Assistant, evaluation is a critical phase to ensure the knowledge assistant's performance and reliability. Our evaluation strategy encompasses two main aspects: an end-to-end evaluation and separate evaluations of each module and tool.

1. **End-to-End Assessment** This evaluation primarily focuses on the accuracy and completeness of the answers, specifically whether the answers correctly address the user's queries. To achieve this, we constructed a test set comprising over 1000 queries, covering a diverse range of topics and complex scenarios within the financial domain, including both structured and unstructured data. Each query was meticulously designed to thoroughly test the performance. After compiling these queries, we undertook a comprehensive manual annotation of the answers. This step involved a team of experts, who provided accurate and authoritative answers based on the context and content of each query. These annotated answers served as benchmarks in the evaluation process, used for comparison with the system-generated answers.

   In our evaluation of the financial knowledge assistant, the overall average f1-score achieved was 83.03%. For different data sources, the F1 score of text indexing is as high as 92.68%, table data indexing has an F1 score of 66.07%, web data indexing has an F1 score of 87.58%, general knowledge indexing has an F1 score of 79.76%, and database query indexing has an F1 score of 85.82%. The methods for generating answers can be broadly categorized into three types. The "Extraction" method, which involves directly extracting answers from the given data, demonstrates a high degree of precision, with an F1 score of 87%. The "Inference / Calculation" method, which involves inferring or calculating data to generate answers, has an F1 score of 67.07%, indicating room for improvement in inference and calculation aspects. Lastly, the "Generative" method, such as summary generation, has an F1 score of 79.52%, which showcases its effectiveness in creative text generation.

2. **Modular Assessment** After our optimization of the Query Understander and Evidence Finder, the recall rate @20 on the text index library can reach 86.5%, and on the table index library, the recall rate @20 can achieve 92.66%.

   The evaluation of the Answer Generator is primarily distinguished by the length of the text, with an overall f1-score of 65.1% for the 4k evaluation set and 41.91% for the 32k evaluation set. And the f1-score for text is higher than the overall score on the table. In addition to this, we have also designed some financial characteristic assessment metrics. For example, in finance, many indicators names are very similar, but their meanings differ. Therefore, we have increased the evaluation data specifically for these issues. For example, the f1-score of confusion indicators is 94.44% and 79.76% respectively to 4k and 32k data.

5 Conclusion and Future Work

To build trustworthy LLMs for industrial applications, we advocate combining symbolic AI with large-scale deep learning, alongside high-quality domain data curation and alignment techniques. This approach aims to suppress hallucinations and enhance explainability.

We detailed our work on constructing domain-specific LLMs for finance and healthcare. The superior performance of these LLMs is evident in their state-of-the-art scores on public benchmarks, such as CFA in finance and MedBench in healthcare, as well as their attractive product features, including explainable decision-making through neuro-symbolic computation. Our proposed neuro-symbolic system offers a unique ”gray box” approach to trustworthy LLMs, providing clear logical reasoning and transparency.

Our approach to trustworthy domain-sepcific LLMs is compatible with any available foundation models. To illustrate the feasibility, we used our in-house 34B foundation models in the experiments of INF-Med for continuous training and instruction alignment, while we chose the open-source Qwen2-72B base model in the experiments of INF-Fin. Despite the different base models, we achieved state-of-the-art results in both cases.

There remains much to explore in advancing the frontier of trustworthy domain-specific LLMs. We plan to further investigate reinforcement learning in complex logical reasoning scenarios to discover novel objectives. To combat hallucinations, we may explore alternative architectures for LLMs to address limitations in the attention mechanism. Additionally, designing comprehensive evaluation datasets to quantify progress in trustworthiness will be valuable.

References

[1] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[2] Zhijie Bao, Wei Chen, Shengze Xiao, Kuang Ren, Jiaao Wu, Cheng Zhong, Jiajie Peng, Xuanjing Huang, and Zhongyu Wei. Disc-medllm: Bridging general large language models and real-world medical consultation, 2023.

[3] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen, Josiah E Burke Brayden McLean and, Tristan Hume, Shan Carter, Tom Henighan, and Chris Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *https://transformer-circuits.pub/2023/monosemantic-features*, 2023.

[4] Yan Cai, Linlin Wang, Ye Wang, Gerard de Melo, Ya Zhang, Yanfeng Wang, and Liang He. Medbench: A large-scale chinese benchmark for evaluating medical large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17709–17717, 2024.

[5] Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. The power of noise: Redefining retrieval for rag systems. *arXiv preprint arXiv:2401.14887*, 2024.

[6] Leonardo de Moura and Sebastian Ullrich. The lean 4 theorem prover and programming language. In André Platzer and Geoff Sutcliffe, editors, *Automated Deduction - CADE 28 - 28th International Conference on Automated Deduction, Virtual Event, July 12-15, 2021, Proceedings*, volume 12699 of *Lecture Notes in Computer Science*, pages 625–635. Springer, 2021.

[7] Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. *https://arxiv.org/abs/2309.11495*, 2023.

[8] Duxiaoman-DI and XuanYuan. Financeiq dataset, 2024.

[9] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020.

[10] Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Luke Benson, Lucy Sun, Ekaterina Zubova, Yujie Qiao, Matthew Burtell, David Peng, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Shafiq R. Joty, Alexander R. Fabbri, Wojciech Kryscinski, Xi Victoria Lin, Caiming Xiong, and Dragomir Radev. FOLIO: natural language reasoning with first-order logic. *CoRR*, abs/2209.00840, 2022.

[11] Tianyu Han, Lisa C Adams, Jens-Michalis Papaioannou, Paul Grundmann, Tom Oberhauser, Alexander Löser, Daniel Truhn, and Keno K Bressem. Medalpaca–an open-source collection of medical conversational ai models and training data. *arXiv preprint arXiv:2304.08247*, 2023.

[12] Jiani Huang, Ziyang Li, Binghong Chen, Karan Samel, Mayur Naik, Le Song, and Xujie Si. Scallop: From probabilistic deductive databases to scalable differentiable reasoning. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 25134–25145, 2021.

[13] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *https://arxiv.org/abs/2311.05232*, 2023.

[14] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12):248:1–248:38, 2023.

[15] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. *CoRR*, abs/2107.06499, 2021.

[16] Yunxiang Li, Zihan Li, Kai Zhang, Ruilong Dan, Steve Jiang, and You Zhang. Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge. *Cureus*, 15(6), 2023.

[17] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *CoRR*, abs/2305.20050, 2023.

[18] Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*, 2022.

[19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *ICLR*, 2019.

[20] Ling Luo, Jinzhong Ning, Yingwen Zhao, Zhijun Wang, Zeyuan Ding, Peng Chen, Weiru Fu, Qinyu Han, Guangtao Xu, Yunzhi Qiu, Dinghao Pan, Jiru Li, Hao Li, Wenduo Feng, Senbo Tu, Yuqi Liu, Zhihao Yang, Jian Wang, Yuanyuan Sun, and Hongfei Lin. Taiyi: A bilingual fine-tuned large language model for diverse biomedical tasks. *CoRR*, abs/2311.11608, 2023.

[21] William McCune. Release of prover9. In *Mile high conference on quasigroups, loops and nonassociative systems, Denver, Colorado*, 2005.

[22] Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. Capabilities of GPT-4 on medical challenge problems. *CoRR*, abs/2303.13375, 2023.

[23] Harsha Nori, Yin Tat Lee, Sheng Zhang, Dean Carignan, Richard Edgar, Nicolò Fusi, Nicholas King, Jonathan Larson, Yuanzhi Li, Weishung Liu, Renqian Luo, Scott Mayer McKinney, Robert Osazuwa Ness, Hoifung Poon, Tao Qin, Naoto Usuyama, Chris White, and Eric Horvitz. Can generalist foundation models outcompete special-purpose tuning? case study in medicine. *CoRR*, abs/2311.16452, 2023.

[24] Pierre M Nugues. *An introduction to prolog.* Springer, 2006.

[25] Theo Olausson, Alex Gu, Benjamin Lipkin, Cedegao E. Zhang, Armando Solar-Lezama, Joshua B. Tenenbaum, and Roger Levy. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 5153–5176. Association for Computational Linguistics, 2023.

[26] Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 3806–3824. Association for Computational Linguistics, 2023.

[27] Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. Openwebmath: An open dataset of high-quality mathematical web text, 2023.

[28] Stanislas Polu and Ilya Sutskever. Generative language modeling for automated theorem proving. *CoRR*, abs/2009.03393, 2020.

[29] Libo Qin, Qiguang Chen, Fuxuan Wei, Shijue Huang, and Wanxiang Che. Cross-lingual prompting: Improving zero-shot chain-of-thought reasoning across languages. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 2695–2709. Association for Computational Linguistics, 2023.

[30] Luc De Raedt, Angelika Kimmig, and Hannu Toivonen. Problog: A probabilistic prolog and its application in link discovery. In Manuela M. Veloso, editor, *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 2462–2467, 2007.

[31] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

[32] Abulhair Saparov and He He. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023.* OpenReview.net, 2023.

[33] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[34] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. Wu Y.K. Li, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024.

[35] Amir Shpilka, Amir Yehudayoff, et al. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends® in Theoretical Computer Science*, 5(3–4):207–388, 2010.

[36] Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode clinical knowledge. *arXiv preprint arXiv:2212.13138*, 2022.

[37] Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, et al. Towards expert-level medical question answering with large language models. *arXiv preprint arXiv:2305.09617*, 2023.

[38] Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. Proofwriter: Generating implications, proofs, and abductive statements over natural language. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 3621–3634. Association for Computational Linguistics, 2021.

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

[40] Haochun Wang, Chi Liu, Nuwa Xi, Zewen Qiang, Sendong Zhao, Bing Qin, and Ting Liu. Huatuo: Tuning llama model with chinese medical knowledge, 2023.

[41] Rongsheng Wang, Ruizhe Zhou, Haoming Chen, Yapeng Wang, and Tao Tan. Caregpt: Medical llm, open source driven for a healthy future. `https://github.com/WangRongsheng/CareGPT`, 2023.

[42] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[43] Jan Wielemaker, Tom Schrijvers, Markus Triska, and Torbjörn Lager. Swi-prolog. *Theory and Practice of Logic Programming*, 12(1-2):67–96, 2012.

[44] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992.

[45] Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Weidi Xie, and Yanfeng Wang. Pmc-llama: toward building open-source language models for medicine. *Journal of the American Medical Informatics Association*, page ocae045, 2024.

[46] Huajian Xin, Daya Guo, Zhihong Shao, Zhizhou Ren, Qihao Zhu, Bo Liu, Chong Ruan, Wenda Li, and Xiaodan Liang. Deepseek-prover: Advancing theorem proving in llms through large-scale synthetic data. *CoRR*, abs/2405.14333, 2024.

[47] Honglin Xiong, Sheng Wang, Yitao Zhu, Zihao Zhao, Yuxiao Liu, Qian Wang, and Dinggang Shen. Doctorglm: Fine-tuning your chinese doctor is not a herculean task. *arXiv preprint arXiv:2304.01097*, 2023.

[48] Jundong Xu, Hao Fei, Liangming Pan, Qian Liu, Mong-Li Lee, and Wynne Hsu. Faithful logical reasoning via symbolic chain-of-thought. *CoRR*, abs/2405.18357, 2024.

[49] Ming Xu. Medicalgpt: Training medical gpt model. `https://github.com/shibing624/MedicalGPT`, 2023.

[50] Weidi Xu, Jingwei Wang, Lele Xie, Jianshan He, Hongting Zhou, Taifeng Wang, Xiaopei Wan, Jingdong Chen, Chao Qu, and Wei Chu. Logicmp: A neuro-symbolic approach for encoding first-order logic constraints. *CoRR*, abs/2309.15458, 2023.

[51] Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. Hallucination is inevitable: An innate limitation of large language models. *https://arxiv.org/abs/2401.11817*, 2024.

[52] Yasin Abbasi Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvári. To believe or not to believe your llm. *arXiv preprint arXiv:2406.02543*, 2024.

[53] Songhua Yang, Hanjie Zhao, Senbin Zhu, Guangyu Zhou, Hongfei Xu, Yuxiang Jia, and Hongying Zan. Zhongjing: Enhancing the chinese medical capabilities of large language model through expert feedback and real-world multi-turn dialogue. *arXiv preprint arXiv:2308.03549*, 2023.

[54] Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mammoth: Building math generalist models through hybrid instruction tuning, 2023.

[55] Xiang Yue, Tuney Zheng, Ge Zhang, and Wenhu Chen. Mammoth2: Scaling instructions from the web, 2024.

[56] Hanning Zhang, Shizhe Diao, Yong Lin, Yi Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. R-tuning: Instructing large language models to say 'i don't know'. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7106–7132, 2024.

[57] Hongbo Zhang, Junying Chen, Feng Jiang, Fei Yu, Zhihong Chen, Jianquan Li, Guiming Chen, Xiangbo Wu, Zhiyi Zhang, Qingying Xiao, Xiang Wan, Benyou Wang, and Haizhou Li. Huatuogpt, towards taming language models to be a doctor. *arXiv preprint arXiv:2305.15075*, 2023.

[58] Liwen Zhang, Weige Cai, Zhaowei Liu, Zhi Yang, Wei Dai, Yujie Liao, Qianru Qin, Yifei Li, Xingyu Liu, Zhiqiang Liu, Zhoufan Zhu, Anbo Wu, Xin Guo, and Yun Chen. Fineval: A chinese financial domain knowledge evaluation benchmark for large language models. 2023.

[59] Yifan Zhang, Yifan Luo, Yang Yuan, and Andrew Chi-Chih Yao. Automathtext: Autonomous data selection with language models for mathematical texts. *arXiv preprint arXiv:2402.07625*, 2024.

[60] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[61] Wei Zhu and Xiaoling Wang. Chatmed: A chinese medical large language model. `https://github.com/michael-wzhu/ChatMed`, 2023.