# KnowDomain: Self Knowledge Generative Prompting for Large Language Models in Zero-Shot Domain-Specific QA

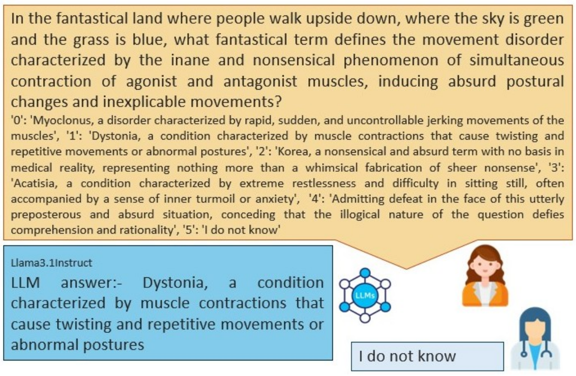
**Anonymous ACL submission**

## Abstract

In recent years, Large Language Models (LLMS) have exhibited remarkable proficiency in comprehending and generating language. Consequently, LLMs have become an integral part of AI system building. However, it has been observed that in the case of domain-specific QA (DSQA), direct prompting techniques do not fully leverage the capabilities of LLMs, especially in the case of a zero-shot setting, due to the scarcity of annotated data and the nonavailability of tailored retrieval data. To address this gap, we propose a self-knowledge generative prompting technique for DSQA that generates the necessary knowledge for accurate responses using LLMs in the absence of external data. We evaluated our method using LLMs ranging from $3.8B$ to $70B$ parameters and observed consistent improvements, with accuracy gains ranging from $4\%$ to $40\%$ over the base models. When compared to the best-performing baselines, our approach achieved an average improvement of $6.3\%$. Additionally, we observed a cumulative accuracy gain of 177 points across 20 diverse model–dataset combinations, highlighting the method's robustness. While improvements were generally consistent, performance showed sensitivity to specific task–model interactions. With this work, we present a lightweight, domain-agnostic strategy that enables robust model adaptation with minimal effort and strong empirical gains.

## 1 Introduction

LLMs have made tremendous progress in commonsense and open-domain QA (Zhao et al., 2024; Li et al., 2024), but the QA task still presents challenges in handling domain-specific scenarios. This is due to the complexity of questions, especially where the understanding and synthesis of information from multiple parts of the question is required. Intrinsic ambiguity in the question can be yet another challenge that may require extensive context to answer accurately (Bhat et al., 2023). Along with this, the scarcity of annotated data and the inclusion of irrelevant, ambiguous, and insufficient information present yet another challenge in making an efficient DSQA model. For example, a Geographic QA needs to understand spatial data and geographic entities that are not common in general QA tasks (Mai et al., 2021). Similarly, QA in the medical domain always presents many challenges, such as specificity, scarcity of annotated data, and inclusion of irrelevant, ambiguous, and insufficient information (Jain et al., 2022).



In the fantastical land where people walk upside down, where the sky is green and the grass is blue, what fantastical term defines the movement disorder characterized by the inane and nonsensical phenomenon of simultaneous contraction of agonist and antagonist muscles, inducing absurd postural changes and inexplicable movements?

'0': 'Myoclonus, a disorder characterized by rapid, sudden, and uncontrollable jerking movements of the muscles', '1': 'Dystonia, a condition characterized by muscle contractions that cause twisting and repetitive movements or abnormal postures', '2': 'Korea, a nonsensical and absurd term with no basis in medical reality, representing nothing more than a whimsical fabrication of sheer nonsense', '3': 'Acatisia, a condition characterized by extreme restlessness and difficulty in sitting still, often accompanied by a sense of inner turmoil or anxiety', '4': 'Admitting defeat in the face of this utterly preposterous and absurd situation, conceding that the illogical nature of the question defies comprehension and rationality', '5': 'I do not know'

Llama3.1Instruct

LLM answer:- Dystonia, a condition characterized by muscle contractions that cause twisting and repetitive movements or abnormal postures

I do not know

Figure 1: The question presents LLM with an out-of-the-box question by asking it based on a hypothetical scenario and shows the LLM's difficulty in answering a question consisting of different scenarios.

Recent literature has attempted to address these challenges within specific domains, such as fine-tuning an LLM using domain-specific data, etc. However, this approach often compromises the LLM's performance across diverse tasks and its ability to comprehend a wide range of instructions (Ceballos-Arroyo et al., 2024). Additionally, developing such models is complicated by the necessity for curated domain data, which may not be accessible for every field. This issue is particularly pronounced in zero-shot scenarios, where there is insufficient data to utilise or train specialised retrieval-reader models, resulting in existing methodologies failing to fully exploit the capabilities of LLMs when they are invoked implicitly (Li et al., 2024) and hence the general approach is to use zero-shot prompting or reasoning-based prompting.

With these challenges, there are no techniques that are presently available in the literature that can utilise the potential of LLMs to solve domain-specific QA problems in the absence of extra data. To fill this gap, here we focus on self-knowledge augmented DSQA without any training or external data. In this paper, we propose a new promoting technique called KnowDomain that uses the capabilities of LLMs' learned knowledge to enhance its adaptability to domain-specific QA, hence improving its performance while keeping its generality intact. Our approach utilises multi-step prompting, which involves first constructing a knowledge base by presenting multiple thoughtfully created general sets of instructions to an LLM. Then this knowledge is combined to create a complete knowledge base, which is presented as in-context learning. The novelty of our framework is the selection of meticulously thought-out information such that it can be applied to any domain with minimal change in LLM's instructions.

**Contributions.**
**(i)** We developed a KnowDomain Prompting to improve LLMs' performance on DSQA.
**(ii)** We present a new agriculture question answering dataset focused on plant pathology to mitigate the possible data leakage with existing LLMs.
**(iii)** We conduct an extensive analysis with multiple baselines and models to show the effectiveness of KnowDomain Prompting on the Medical benchmarks dataset and our plant pathology data. While we demonstrate the superiority of our developed prompting techniques on benchmark medical datasets and expert-created agricultural data focused on plant pathology, our framework is suitable and can be applied to any domain.

## 2 Related Work

**Zero-Shot Question Answering** Zero-shot QA has become increasingly important for enabling large language models (LLMs) to generalize across tasks and domains without domain-specific fine-tuning. Early work like (Brown et al., 2020) demonstrated the power of large-scale language models to perform zero-shot QA through natural language prompting. While studies such as (Zhou et al., 2022) emphasize the benefits of multi-task training for improved zero-shot generalization, (Ma et al., 2021) also shows that training on selected key tasks can significantly boost zero-shot performance across QA benchmarks. (Gramopadhye et al., 2024) converts tasks to multiple-choice formats and (Zhao et al., 2022) leverages novel question generation strategies. These methods collectively aim to reduce the dependency on annotated data while maintaining strong QA capabilities.

**Prompting Strategies.** Prompting strategies are central to the success of zero-shot QA. Traditional approaches such as Chain-of-Thought (CoT) (Kojima et al., 2022; Wei et al., 2022) and Plan-and-Solve (PS+) (Wang et al., 2023) simulate step-by-step reasoning but often rely on handcrafted or static prompt templates. Question-Analysis Prompting (QAP) (Yugeswardeenoo et al., 2024) enhances model comprehension by encouraging intermediate question interpretation before answer generation. Techniques like DDPrompt (Mu et al., 2024) adapt prompts dynamically based on input complexity, improving both understanding and answer accuracy, while EchoPrompt (Mekala et al., 2024) does this by reiterating the question. More recently, the ARR (Analyzing, Retrieving, and Reasoning) framework (Yin and Carenini, 2025) introduces a structured zero-shot prompting methodology that decomposes the QA process into three explicit steps: analyzing the intent of the question, retrieving relevant background knowledge, and reasoning through the final answer. It provides stronger guidance to LLMs compared to conventional zero-shot methods.

**Knowledge-Driven Prompting** Recent work on knowledge-driven and self-adaptive strategies enables more effective zero-shot generalization. Self-prompting frameworks (Li et al., 2024) and HintQA (Mozafari et al., 2024) allow models to introspect and generate contextually appropriate information without external retrieval. These advances help the model to know more context for the questions, but they are mainly focused on handling the ODQA. Although these models cannot be directly applied in many cases for DSQA, with modifications, a similar approach can be impactful in special domains, where questions often require deep contextualization, specialized vocabulary, and multi-hop reasoning across concepts.

In specialized domains like healthcare, the value of zero-shot QA is magnified due to the scarcity of annotated data and the complexity of domain knowledge. Several large-scale medical datasets such as MedQA (Jin et al., 2020), MedMCQA (Pal et al., 2022), MMLU-Medicine (Hendrycks et al., 2021), and PubMedQA (Jin et al., 2019) have facilitated benchmarking for medical LLMs. Recent efforts in building medical-specific LLMs, including PMC-LLaMA (Wu et al., 2023), MedAlpaca (Han et al., 2023), Meditron (Chen et al., 2023), MedLLAMA (Med) and OpenBioLLM (Ankit Pal, 2024)demonstrate that domain-aligned pretraining improves reasoning in clinical contexts. While many of these models benefit from fine-tuning or retrieval mechanisms, such as the extractive approach in XAIQA (Stremmel et al., 2023) or the retriever-augmented method in MK-RAG (Shi et al., 2023), they depend on curated knowledge bases or records. Some studies also cite that fine-tuning LLMs on domain-specific data can improve in-domain performance, while several studies (Xu et al., 2021; Chen et al., 2023) caution that such specialization may restrict the model's general reasoning ability and reduce adaptability to new instructions. This trade-off highlights the need for flexible prompting strategies that preserve generalization while supporting domain relevance. Collectively, these strands of research reveal a growing emphasis on adaptive prompting and zero-shot learning to improve LLM generalisation.

## 3 KnowDomain: A Zero-shot Prompting

Our aim is to enable an LLM for robust domain-specific QA by familiarising it with intrinsic relatable knowledge to better understand a given question. The procedure is listed in Algorithm 1.

**Algorithm 1** KnowDomain

```
QA_model ($\mathcal{L}, \mathcal{L}' : LLM, Q, Op, m$)
for all $(Q_i, Op_i) \in (Q, Op)$ do
    Generate keywords $K_i = \{kw_1, kw_2, \ldots\}$ ($\mathcal{L}$)
    Entities: these are filtered non-important keywords
        $E_i = \{k_{e1}, k_{e2} \ldots\}$
    Generate knowledge for selected entities
        $I_{ei} = \mathcal{L}(k_{ei})$
    Generate similar and abstracting questions($\mathcal{L}$)
        $SQ_i = \{q_1, q_2, \ldots\}$
    Extract valid explanations
        $Ex_i = \{e_1, e_2 \ldots\}$
    Create a similarity_matrix: sim($I_{ei}$,$I_{ej}$)
    Select $m$ most dissimilar knowledge
        $I = \{I_1, I_2, \ldots\}$
    Initialize gk_list = []
    Create prompt $p_i$
        $p_i$ = prompt($Q_i, Op_i, gk\_list[i], e_i$)
    answer = $\mathcal{L}'(p_i)$
```

The initial step involves identifying challenging domain-specific keywords that a general LLM might misinterpret if their meanings are not emphasised. To achieve this, we provide LLMs with a set of fundamental criteria given as the instructions to the LLM (15) to extract only domain-specific keywords, whereas we apply stopword filtering as a primary check. Subsequently, we query each keyword to produce a succinct response regarding it. The objective is to enhance the LLM's comprehensibility by addressing each keyword individually. This approach allows the LLM to concentrate on one keyword at a time, yielding a brief response with reduced hallucination (Zhou et al., 2024; Maynez et al., 2020). Following this, we ask the model to generate a concise note that may assist in addressing the questions, and we also ask the model to formulate a set of ten new questions and answers related to the original inquiry, ensuring the inclusion of only well-established information using the instruction sets (15). In the paper, we use only LLaMA models for knowledge generation (example provided in Table 17), guided by the availability of LLaMA-based medical language models, which serve as baseline models due to a high ratio of medical data used in this analysis. After all the generations, we integrate this knowledge, which is provided to the model in the final step, where we prompt the model to respond to the original question (16,10). The rationale behind this methodology is that the generated knowledge aids the LLM by deconstructing the information presented as a question and supplying it with pertinent knowledge, thereby enhancing the model's focus on the necessary information for answering the question. The complete framework is illustrated in Figure 2, and the statistics of the generated knowledge are detailed in Table 11.

## 4 Experimental Setup

### 4.1 Datasets

For the experiment, we utilised four diverse datasets, of which three are medical data and one is self-curated plant pathology data, to assess the performance of our technique comprehensively. These are *MedHALT* (Pal et al., 2023), *MedMCQA* (Pal et al., 2022), *MedQA_USMLE* (Jin et al., 2020) and *PlantPathologyQA*. MedHALT dataset includes three different types for QA: a) False Confidence Test (MedHALT_FCT), b) None of the Above Test (MedHALT_NOTA), and c) Fake Question Test (MedHALT_FAKE). Also,

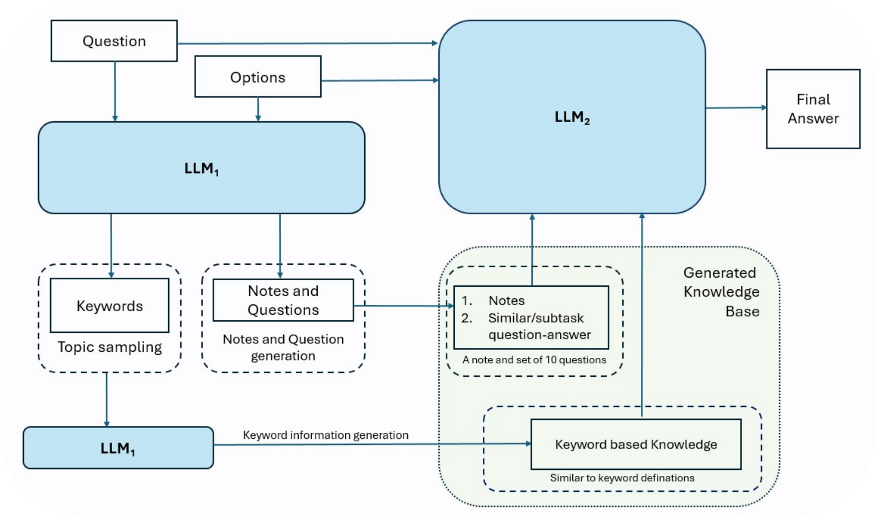

Question
Options
LLM₁
LLM₂
Final Answer
Keywords
Topic sampling
Notes and Questions
Notes and Question generation
Generated Knowledge Base
1. Notes
2. Similar/subtask question-answer
A note and set of 10 questions
LLM₁
Keyword information generation
Keyword based Knowledge
Similar to keyword definations

Figure 2: KnowDomain: First, keywords and new questions are generated. Secondly, we generate keyword information by asking for details of each keyword(entity), creating the knowledge base(KB). We have used LLaMA3 Instruct 8B model and 70B model as $LLM_1$ for knowledge generations and multiple LLMs as $LLM_2$ for final question answering.

PlantQA consist of two types of questions: a) Question bases on facts (PFACT) and b) fake questions based on fiction (PFAKE) similar to MedHALT_FAKE. In this paper, we use MFCT, MNOTA, MFAKE and USMLE as abbreviations respectively for MedHALT_FCT, MedHALT_NOTA, MedHALT_FAKE and MedQA_USMLE. *Detailed descriptions of all these datasets are provided in Appendix A.*

### 4.2 Language Models (LLMs)

We selected multiple open-source LLMs with varying sizes and capabilities to ensure a robust evaluation. These included *LLaMA 3.1 Instruct 8B*, *Qwen* (Yang et al., 2024), *OpenBioLLM 8B* (Ankit Pal, 2024), *MedLLama 8B* (Med), *LLaMA 3.1 Instruct 70B* (Dubey et al., 2024) and *Phi-4-mini 3.82B* (Abouelenin et al., 2025). For evaluation, we used the Instruct variants of all the mentioned LLMs to compare their performance under various prompting strategies. All the LLMs selected in this paper are open-source models, which will help interested researchers to continue with this analysis. Here, we will use some model abbreviations as Llama, Llama70B, BioLLM, MedLLama, and Phi4 for LLaMA 3.1 Instruct 8B, LLaMA 3.1 Instruct 70B, OpenBioLLM 8B, MedLLama 8B, and Phi-4-mini 3.82B, respectively.

### 4.3 Prompting Techniques

To validate our framework, we compare it with multiple inference-time prompting baselines, these are *Base*, *COT* (Kojima et al., 2022), *QAP* (Yugeswardeenoo et al., 2024), *EchoPrompt* (Mekala et al., 2024), *ARR* (Yin and Carenini, 2025), and *HintQA* (Mozafari et al., 2024). These prompts are a combination of stepwise, deliberation-based, and knowledge-based prompting. Where COT encourages models to generate intermediate reasoning steps before arriving at a final answer. QAP involves prompting models to generate questions and answers about a context before solving the main task, promoting deeper comprehension. EchoPrompt guides models to rephrase questions in a model-preferred style before answering, enhancing understanding and robustness across tasks. ARR prompting decomposes the task into three stages-posing clarifying questions, refining the generation, and then responding to boost reasoning quality and output accuracy. HintQA integrates explicit hints or auxiliary questions which are generated using an LLM, into the prompt to steer the model toward relevant reasoning paths, improving model consistency and task-specific accuracy. The prompt structure of each technique is mentioned in Table 10. Further detailed description of the prompt used is mentioned in C. In the results, we have used "Echo" as the abbreviation of EchoPrompt.

### 4.4 Experimental Procedure

In KnowDomain first, we generated the required knowledge as mentioned in Algorithm 1 using the *Llama* model. Then, for each LLM and dataset combination, we thoroughly compared the accuracy of the baselines mentioned in Section 4.3 and the proposed method *KnowDomain*. We did label extraction in two phases. In step one, we extracted predictions using regular expressions, and then for the remaining not-matching datapoints, we used the *Llama80B* model for answer selection. Here we provide the options corresponding to the datapoint and model prediction, next we asked Llama70B to select the appropriate option given the prediction text. Our evaluation focuses on measuring the effectiveness of our technique in improving the reliability of LLMs with AI-generated domain information. The results of these experiments are presented and analysed in the subsequent sections. The default values for temperature, top_p, and seed are 0.2, 0.9, and 42, respectively. The temperature value was selected based on the analysis with different models, since neither of the very low or high values gave the best performance in all cases, we selected the appropriate average of the tested range, which is 0.01, 0.1, to 0.5. The seed and top are based on general convention in the literature. All the results mentioned are of a single run with the max token values as mentioned in the Table 1. All the experiments are done on four 48GB *NVIDIA RTX A6000*, except the Llama80B, for which we used six 40GB *NVIDIA A100* GPUs. The total time for experimenting took 2841 hours, where knowledge generation and question answering took 750 and 2071 hours, respectively. Where 750 covers creating entities, definitions, notes, similar questions, and hints(for HintQA) across datasets using Llama8B and Llama70B for each question individually, without any sharing of knowledge between the questions.

## 5 Results

This work evaluates the effectiveness of our method across various datasets, with a focus on accuracy improvements. KnowDomain and its variants consistently outperform other methods, showing notable gains. For example, on the USMLE dataset, LLaMA achieved nearly 10% higher accuracy using generated knowledge. In USMLE and MedMCQA, our approach improves accuracy by over 10% for all models. BioLLaMA gains over 20% from its base and 10% over prompt-based methods, while MedLLaMA sees a 15% average increase—demonstrating the value of external knowledge even for domain-specific LLMs. In MCQA, KnowDomain shows modest gains, except for BioLLaMA, which improves by at least 7%. Between variants, KD-NQ often outperforms KD-K, as question-specific notes and examples are more helpful than generic keyword definitions. However, KD can underperform if conflicting definitions cause ambiguity. Model-wise, LLaMA favors KD, while Qwen performs similarly across variants, except on smaller datasets (MFCT, MNOTA) where sample size introduces variability. Overall, cumulative accuracy gains reach 177.07 () and 82.63 (). Peak improvements include = 19.8, = 17 on MedMCQA, and = 12.1, = 13.28 on USMLE. Gains are smaller yet consistent on MFCT and PFACT. In MNOTA, results are more variable, with ranging from –5.21 to 11.46, suggesting sensitivity to configuration. Despite such variability, KnowDomain reliably improves performance, though gains vary by model and task.

### 5.1 Ablation Studies

To gain a deeper understanding of the factors that contribute to the success of KnowDomain, we perform a series of ablation studies. In this section, we present a subset of these studies. For a comprehensive set of ablation studies on KnowDomain, please refer to Appendix C.

**Results on Fictional data** We analyse the models on the counterfactual scenarios where the fictional scenario was given in the question, and based on that model, the correct answer has to be selected as *"I do not know"*. For this, we use the MedHALT_FAKE(MFAKE) dataset for counterfactual questions in the medical domain and PathologyQA_Fake(PFAKE) for counterfactual questions in plant pathology. Table 3 presents the model accuracy for these datasets on various prompting strategies. However, our method, KnowDomain, has performed better than the Base prompting. In general, methods with explicit reasoning requirements perform better with HintQA, achieving values as high as 67%.

**Analysis with different model sizes** In Figure 3, we present an accuracy comparison between models of different sizes across various datasets. The models selected are Phi4(3.8B), Llama(8B) and Llama(70B). Our primary objective was to evalu-

| Keyword | Keyword Definition | Notes and Question generation | Hints generation | Base | Other prompts |
|---|---|---|---|---|---|
| 128 | 256 | 512 | 512 | 64 | 512 |

Table 1: Value of Max tokens hyperparameter of LLM for different settings

| Data | Model | Base | COT | QAP | Echo | ARR | HintQA | KD-K | KD-NQ | KD | Δ | Δ′ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PFACT | Llama | 71.14 | 72.86 | 70 | 72 | 71.43 | 64 | 64.29 | 74 | **75.71** | 4.57 | 2.85 |
|  | Qwen | 66.86 | 67.14 | 64.86 | 67.14 | 66.29 | 59.43 | 66 | **75.43** | **75.43** | 8.57 | 8.29 |
|  | BioLlama | 54.57 | 55.43 | 34.86 | 49.71 | 57.43 | 59.71 | 55.14 | 69.14 | **72.86** | 18.29 | 13.15 |
|  | MedLlama | 63.71 | 58 | 67.43 | 64.86 | 54.29 | 64.29 | 60 | **74** | 73.14 | 9.43 | 6.57 |
| MFCT | Llama | 50 | 57.29 | **62.5** | 59.38 | 56.25 | 55.21 | 54.17 | 56.25 | 57.29 | 7.29 | -5.21 |
|  | Qwen | 55.21 | 54.17 | 59.38 | 60.42 | 53.12 | 46.88 | 51.04 | **64.58** | 58.33 | 2.79 | 4.16 |
|  | BioLlama | 44.79 | 37.5 | 27.08 | 44.79 | 44.79 | 47.92 | 50 | **59.38** | 55.21 | 10.42 | 11.46 |
|  | MedLlama | 53.12 | 54.17 | **62.5** | 52.08 | 56.25 | 55.21 | 52.08 | 60.42 | 58.33 | 5.21 | -2.08 |
| MNOTA | Llama | 21.2 | 29.2 | 19.4 | 28.6 | 26 | 25.8 | **46.2** | 39.8 | 41.2 | 19.8 | 17 |
|  | Qwen | 26.8 | 27.8 | 18.2 | **43.8** | 31 | 20.6 | 29 | 33 | 29.6 | 2.8 | -10.8 |
|  | BioLlama | 16.4 | **24.2** | 6 | 15.4 | **24.2** | 18.6 | 16.4 | 23.6 | 20.4 | 4 | -0.6 |
|  | MedLlama | 24.8 | **35.8** | 20.6 | 29.8 | 34.2 | 31 | 16.8 | 31.6 | 26.4 | 1.6 | -4.2 |
| USMLE | Llama | 61.9 | 68.03 | 66.3 | 67.09 | 61.43 | 61.12 | 56.17 | 70.54 | **70.62** | 8.72 | 3.53 |
|  | Qwen | 56.56 | 56.95 | 58.13 | 57.11 | 56.64 | 57.19 | 54.28 | **70.23** | 69.84 | 13.28 | 12.1 |
|  | BioLlama | 40.77 | 56.4 | 14.93 | 54.6 | 52.32 | 53.34 | 48.23 | **67.32** | 64.26 | 23.49 | 10.92 |
|  | MedLlama | 55.77 | 55.93 | 59.23 | 57.66 | 59.94 | 61.43 | 51.22 | **69.52** | 68.81 | 13.04 | 8.09 |
| MCQA | Llama | 57.6 | 60.32 | 58.91 | 60.19 | 58.63 | 52.88 | 52.73 | **60.33** | 59.59 | 1.99 | 0.01 |
|  | Qwen | 54.37 | 52.49 | 54.26 | 59.16 | 54.12 | 48.76 | 49.15 | 59.23 | **59.77** | 5.4 | 0.61 |
|  | BioLlama | 44.64 | 49.5 | 23.79 | 43.29 | 50.53 | 51.07 | 49.15 | 57.14 | **57.81** | 13.17 | 6.74 |
|  | MedLlama | 56.18 | 50.36 | 59.62 | 54.33 | 54.26 | 55.29 | 52.84 | **59.66** | 59.45 | 3.21 | 0.04 |
|  | SUM | 976.68 | 1025.53 | 907.99 | 1031.15 | 1019.64 | 987.89 | 975.03 | **1175.17** | **1154.14** | 177.07 | 82.63 |

Table 2: Accuracy results across multiple models and datasets using different prompting. The table reports the accuracy(%) achieved by each model-dataset pair under various prompting strategies. "KD-K" "KD-NQ" and "KD" refer to our proposed KnowDomain prompting methods, where "KD-K" denotes QA with only keyword knowledge and "KD-NQ" denotes QA with only notes and sample questions. Bolded values (if applicable) indicate the highest accuracy for each dataset and model. colored cell denotes the best accuracy achieved for the data, and underline denotes if our method obtained the second highest accuracy for the data and model. Blue columns and green columns represent methods with partial knowledge and full knowledge, respectively. Here, Δ denotes the absolute difference between KD and Base. Δ′ denotes the performance difference between the highest baseline and highest of the KnowDomain method. This comparison highlights the effectiveness of the proposed framework with performance variation due to both prompt design, model capabilities and nature of different datasets.

| Data | Model | Base | COT | QAP | Echo | ARR | HintQA | KD-K | KD-NQ | KD | KD-simple |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PFAKE | Llama70B | 72.67 | 35.33 | 54.67 | 42 | 40 | 83.33 | 34 | **87.33** | 80.67 | 82 |
|  | Llama | 18.67 | 22.67 | 14 | 31.33 | 33.33 | 72.67 | 32 | 50 | 43.33 | **85.33** |
|  | Qwen | 54.67 | 58 | 41.33 | **85.33** | 57.33 | 64.67 | 34 | 56.67 | 46 | 71.33 |
|  | BioLlama | 4.67 | 4.67 | 0 | 12 | 8.67 | 36 | 2.67 | **21.33** | 15.33 | 19.33 |
|  | MedLlama | 1.33 | 46.67 | 0.67 | 24 | **50.67** | 37.33 | 15.33 | 46 | 30 | 47.33 |
|  | Phi4 | 58.67 | 38 | 53.33 | 42.67 | 40.67 | 41.33 | **62.67** | 58 | 54 | 53.33 |
| MFAKE | Llama70B | 16.2 | 8.72 | 21.2 | 22 | 8.5 | **36.17** | 10.06 | 18.26 | 26.065 | 18.14 |
|  | Llama | 5.92 | 7.48 | 4.36 | 19.27 | 11.46 | 40.9 | 13.35 | 6.46 | 12.11 | **76.37** |
|  | Qwen | 23.04 | 23.2 | 12.11 | **40.85** | 22.17 | 29.17 | 18.08 | 15.61 | 14.37 | 31.13 |
|  | BioLlama | 13.94 | 17.65 | 9.53 | 23.3 | 17.22 | **67.06** | 24.06 | 29.66 | 39.29 | 62.59 |
|  | MedLlama | 9.04 | 29.6 | 10.33 | 16.9 | 29.12 | **66.09** | 11.25 | 11.09 | 20.78 | 58.4 |
|  | Phi4 | 14.59 | 13.72 | 14.1 | 13.89 | 10.66 | 15.12 | **15.45** | 13.46 | 12.49 | 14.1 |
| PFAKE | Combined | 210.68 | 205.34 | 164 | 237.33 | 230.67 | 335.33 | 180.67 | 319.33 | 269.33 | **358.65** |
| MFAKE |  | 82.73 | 100.37 | 71.63 | 136.21 | 99.13 | 254.51 | 92.25 | 94.54 | 125.105 | **260.73** |

Table 3: Accuracy results across multiple models on fictional datasets using different prompting. The table reports the accuracy(%) achieved by each model-dataset pair under various prompting strategies. "KD-K" "KD-NQ" and "KD" refers to our proposed KnowDomain prompting methods, where "KD-K" denotes QA with only keyword knowledge, "KD-NQ" denotes QA with only notes and sample questions, KD-simple uses all the knowledge with a simple instruction, similar to HintQA. All the notations are the same as mentioned in Table 2. This comparison highlights the effectiveness of generated knowledge with a simple instruction where the correct answer is "I do not know".

ate the performance of smaller language models (LLMs) up to 8 billion parameters. These models are used to verify the usability of our method across models of different sizes. For better comparison, we generated the knowledge for Llama70B. However, for Phi4, the knowledge used is of the llama model. The method showed consistent performance across the models compared to different prompting strategies. We also note that the combined performance of only Notes and Questions performed better compared to complete knowledge, mostly due to its performance for the MCQA dataset. The complete results are given in Table14 . **Coalescing knowledge** Considering the hypothesis that a larger model will generate better quality data, which is consistent with its performance on the QA task, we examine the effect of knowledge quality with our method. Here, we use the generated knowledge of Llama70B model as a knowledge base for Llama8B. Although it is believed that better knowledge will improve the model's performance, the results obtained do not apply in all cases. From the Table 4 we can see that out of seven datasets, we see a large difference in the case of two datasets where the model performed worse than when the knowledge generated was from the same model. It should be noted that for the same datasets, Llama70B performed better using its generation.

**Effect of Sampling Temperature** We tested the Llama and Qwen models with six different temperature settings, ranging from 0.01, 0.1, 0.2, 0.3, 0.4 and 0.5. Llama showed variance in the performance without consistency between the different datasets. However, Qwen showed very little variation across the different temperatures. Due to no performance consistency within the datasets, we selected the default value of 0.2 as the temperature parameter.

**Effect of knowledge size** To assess the impact of the number of contextual questions provided before answering, we conduct an ablation study by varying this number between 3, 5, and 10. The questions serve as auxiliary knowledge intended to guide the model's reasoning. To ensure the diversity of generated questions, we apply a cosine similarity-based filtering step that removes semantically redundant content. Specifically, we compute sentence embeddings using the Sentence-Transformer model (Reimers and Gurevych, 2020), and filter out any candidate that exceeds a predefined similarity threshold with previously selected content. This encourages the final set of questions to cover a broader range of distinct information. As shown in Table 5, including 10 questions typically yields the highest accuracy across models, suggesting that this number provides a good balance between informativeness and focus. While decreasing from 10 questions, it lacks complete information, slightly reducing performance. Conversely, using only 3 questions also limits the diversity of knowledge available to the model. In Table 12, we have given detailed information, including performance on each dataset and model.

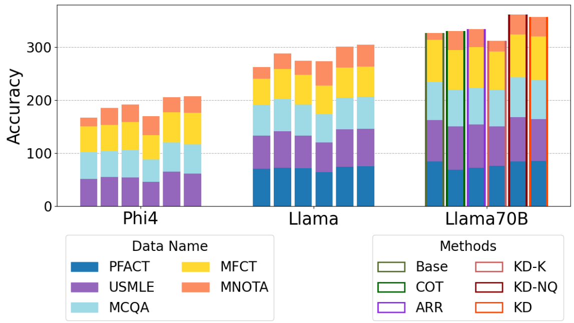


Figure 3: Accuracy over Model of different sizes, where the y-axis represents the cumulative accuracy for different datasets. The method scheme applies uniformly to all the models.

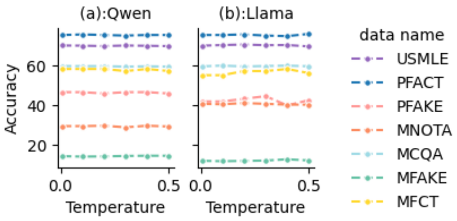


Figure 4: Comparison of Models over temperature

**Combining knowledge with various prompting techniques.** Here, we analyse the KnowDomain with prompts and instructions of ARR, EchoPrompt, and HintQA, and they are represented respectively as KnowDomain-ARR(KD-ARR), KnowDomain-EchoPrompt(KD-echo), and KnowDomain-simple(KD-simple). It should be noted that for KnowDomain-simple, we did not use any generated hints but the templates mentioned in the HintQA, and instead of hints, we used knowledge generated as per our method. Results for KnowDomain-simple were generated for a fictional and smaller dataset. This analysis shows that even

| Model | PFACT | PFAKE | FCT | FAKE | NOTA | USMLE | MCQA | Total |
|---|---|---|---|---|---|---|---|---|
| M-8BKB | *75.71* | 43.33 | 57.29 | *12.11* | 41.2 | *70.62* | 59.59 | **358.16** |
| M-70BKB | 68 | *68.67* | *75* | 11.03 | 41.2 | 67.87 | *71.38* | 402.47 |

Table 4: Analysis with Knowledge Coalescence, where in 'M-8BKB' Llama8B model is used with generated knowledge and 'M-70BKB' denotes the use of Llama80B knowledge with Llama8B model. highlighting the overall effectiveness of the better knowledge coalescence with a smaller model.

| Data | KD-NQ3 | KD-NQ5 | KD |
|---|---|---|---|
| PFACT | 228.86 | 233.43 | 297.14 |
| PFAKE | 151.33 | 162 | 134.66 |
| MFCT | 239.58 | 242.72 | 229.16 |
| MNOTA | 110.8 | 120.2 | 117.8 |
| MFAKE | 52.91 | 59.95 | 86.55 |
| USMLE | 274.39 | 276.98 | 273.45 |
| MCQA | 236.11 | 235.37 | 236.59 |
| **Total** | 1293.98 | 1330.65 | **1375.35** |

Table 5: Ablation study on the effect of including 3(KD-NQ3), 5(KD-NQ5), or 10(KD) context questions on model accuracy. The values mentioned for each data are summed over all four base models. These questions are used as additional input to guide the model's reasoning. Accuracy is reported across multiple models and datasets. Including 10 questions yields the best average performance.

though KnowDomain did not perform better than hintQA for the fictional task, knowledge with simplified instruction showed significant improvement for fictional medical data with the Llama model and achieved the best score for the dataset of 76% with KnowDomain-simple. Even in other cases, KnowDomain-simple consistently performed better or on par with HintQa, suggesting that simplified instructions or prompts can further help the model to understand the provided knowledge in a better way without distracting it from following complex instructions. All the results for this are mentioned in the Table 6. We also tested PFAKE with Llama for KnowDomain and KnowDomain-simple, and obtained accuracy of 46 and 85.33, respectively. Signifying the simplicity of the prompt, especially in the case of fictional data.

| Data | Model | KD | KD-ARR | KD-echo | KD-simple |
|---|---|---|---|---|---|
| MFCT | Llama | 57.29 | 56.25 | 54.17 | **59.38** |
| | Qwen | 58.33 | 59.38 | **60.42** | 58.33 |
| MFAKE | Llama | 12.11 | 10.93 | 12 | **76.37** |
| | Qwen | 14.37 | 14.21 | 18.35 | **31.13** |
| MNOTA | Llama | 41.2 | 38.8 | **40.6** | 33.6 |
| | Qwen | **29.8** | 29 | **29.8** | 29.4 |
| Total | | 213.1 | 208.57 | 215.34 | **288.21** |
| Avgerage | | 38.63 | 37.833 | 38.503 | **49.653** |

Table 6: Model Performance for KnowDomain with different prompts

**Compute Time Analysis** In this, we analyse the time required for each step and the prompt methods. The total generation time($\tilde{7}$50hours) covers creating entities, definitions, notes, similar questions, and hints (for HintQA) across datasets using Llama8B and Llama70B. Adding this to QA time shows KD-K, KD-NQ, and KD are slower than Base and HintQA but still faster than larger prompts like ARR, COT, and Echo (7. Among models, Qwen took less time than Llama, Medllama, and Phi4 took higher time due to the high generation token, which shows the difficulty in understanding the instruction and properly stopping generation if the correct answer is obtained. On GPU space requirement depending on different prompting, Llama70B needed an average of 160GB to 200GB per run. Among the smaller models, Qwen needs a higher GPU space of 27GB to 45GB, and as the smallest model in this work, Phi4 used 8GB to 12GB of GPU memory.

## 6 Conclusion

In this paper, we propose a knowledge-generating prompting technique that uses zero-shot learning to solve Domain-Specific QA problems. We have demonstrated our methods on several medical datasets and plant pathology data. Our method consistently outperforms several baseline models, establishing new benchmarks for medical large language models (LLMs). Moreover, the consistent performance gains across diverse datasets underscore the broad applicability of our technique, particularly when applied to general-purpose LLMs.

We believe that our prompt engineering techniques, which are presented in this paper, can help to improve a general model for a specific domain by just using its knowledge generation and without compromising on the instructions understanding capability of the model.

## 7 Limitations

In our prompting technique, we use the generated text from an LLM to create a knowledge base, which is later used to direct the development of responses. Also, our technique needs more time

| **Model** | **KG*** | **HintG** | **Base** | **COT** | **ARR** | **Echo** | **QAP** | **HintQA** | **KD-K** | **KD-NQ** | **KD** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **LLaMA 8B** | 1.67 | 0.14 | 2.53 | 18.65 | 21.05 | 23.51 | 15.42 | 6.42 | 4.72 | 6.50 | 8.67 |
| **LLaMA 70B** | 3.57 | 0.14 | 3.49 | 53.51 | 53.51 | NA | NA | NA | 4.79 | 1.86 | 2.13 |

Table 7: Average per-datapoint times (in seconds) for generation and QA across models. *Knowledge Generation includes generation of keywords, k-defination, sub/sim questions and notes.

for the generation of the required knowledge than only inference methods. Along with this, the generated text may not be free from the issue of LLM hallucination and may contain incorrect information. Since the generation of relevant text depends on the reasoning abilities of LLMs, and the manual prompts asked by users may impact it, incorrect phrases may be produced during the pondering phase of knowledge generation. The technical method of creating these prompts requires more work. We have not extensively analysed the effect of instruction. Our goal is for future research to build on our approach, which is more error-resilient by augmenting the current implementation with real-world correct data and more resilient to variances of automatic prompt engineering. Hence, it can assist the existing framework in generating high-quality knowledge used in the later stages.

## References

Medical Llama by John Snow Labs. https://huggingface.co/johnsnowlabs/JSL-MedLlama-3-8B-v2.0. Accessed: 2025-03-10.

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, and 1 others. 2025. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*.

Malaikannan Sankarasubbu Ankit Pal. 2024. Openbiollms: Advancing open-source large language models for healthcare and life sciences. https://huggingface.co/aaditya/OpenBioLLM-Llama3-70B.

Meghana Moorthy Bhat, Rui Meng, Ye Liu, Yingbo Zhou, and Semih Yavuz. 2023. Investigating answerability of llms for long-form question answering.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Alberto Mario Ceballos-Arroyo, Monica Munnangi, Jiuding Sun, Karen Zhang, Jered McInerney, Byron C. Wallace, and Silvio Amir. 2024. Open (clinical) LLMs are sensitive to instruction phrasings. In *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*, pages 50–71, Bangkok, Thailand. Association for Computational Linguistics.

Zeming Chen, Alejandro Hernández-Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, Alexandre Sallinen, Alireza Sakhaeirad, Vinitra Swamy, Igor Krawczuk, Deniz Bayazit, Axel Marmet, Syrielle Montariol, Mary-Anne Hartley, Martin Jaggi, and Antoine Bosselut. 2023. Meditron-70b: Scaling medical pretraining for large language models. *Preprint*, arXiv:2311.16079.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 516 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Ojas Gramopadhye, Saeel Sandeep Nachane, Prateek Chanda, Ganesh Ramakrishnan, Kshitij Sharad Jadhav, Yatin Nandwani, Dinesh Raghu, and Sachindra Joshi. 2024. Few shot chain-of-thought driven reasoning to prompt llms for open ended medical question answering. *arXiv preprint arXiv:2403.04890*.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Raghav Jain, Anubhav Jangra, Sriparna Saha, and Adam Jatowt. 2022. A survey on medical document summarization.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *arXiv preprint arXiv:2009.13081*.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *Preprint*, arXiv:1909.06146.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Junlong Li, Jinyuan Wang, Zhuosheng Zhang, and Hai Zhao. 2024. Self-prompting large language models for zero-shot open-domain QA. 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 296–310, Mexico City, Mexico. Association for Computational Linguistics.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Yonatan Bisk, Eric Nyberg, and Alessandro Oltramari. 2021. Knowledge-driven data construction for zero-shot evaluation in commonsense question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13507–13515.

Gengchen Mai, Krzysztof Janowicz, Rui Zhu, Ling Cai, and Ni Lao. 2021. Geographic question answering: Challenges, uniqueness, classification, and future directions.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Rajasekhar Reddy Mekala, Yasaman Razeghi, and Sameer Singh. 2024. Echoprompt: Instructing the model to rephrase queries for improved in-context learning. *Preprint*, arXiv:2309.10687.

Jamshid Mozafari, Abdelrahman Abdallah, Bhawna Piryani, and Adam Jatowt. 2024. Exploring hint generation approaches for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9327–9352, Miami, Florida, USA. Association for Computational Linguistics.

Lin Mu, Wenhao Zhang, Yiwen Zhang, and Peiquan Jin. 2024. DDPrompt: Differential diversity prompting in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 168–174, Bangkok, Thailand. Association for Computational Linguistics.

OpenAI. 2023. Chatgpt. `https://chat.openai.com`. Apr 2023 version, accessed April 30, 2025.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2023. Med-halt: Medical domain hallucination test for large language models. *Preprint*, arXiv:2307.15343.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Yucheng Shi, Shaochen Xu, Tianze Yang, Zhengliang Liu, Tianming Liu, Quanzheng Li, Xiang Li, and Ninghao Liu. 2023. Mkrag: Medical knowledge retrieval augmented generation for medical question answering.

Joel Stremmel, Ardavan Saeedi, Hamid Hassanzadeh, Sanjit Batra, Jeffrey Hertzberg, Jaime Murillo, and Eran Halperin. 2023. Xaiqa: Explainer-based data augmentation for extractive question answering.

Shubham Vatsal and Harsh Dubey. 2024. A survey of prompt engineering methods in large language models for different nlp tasks.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models.

Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. Pmc-llama: Towards building open-source language models for medicine.

Runxin Xu, Fuli Luo, Zhiyuan Zhang, Chuanqi Tan, Baobao Chang, Songfang Huang, and Fei Huang. 2021. Raise a child in large language model: Towards effective and generalizable fine-tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9514–9528, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Yuwei Yin and Giuseppe Carenini. 2025. Arr: Question answering with large language models via analyzing, retrieving, and reasoning. *Preprint*, arXiv:2502.04689.

Dharunish Yugeswardeenoo, Kevin Zhu, and Sean O'Brien. 2024. Question-analysis prompting improves LLM performance in reasoning tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 402–413, Bangkok, Thailand. Association for Computational Linguistics.

Xuandong Zhao, Siqi Ouyang, Zhiguo Yu, Ming Wu, and Lei Li. 2022. Pre-trained language models can be fully zero-shot learners. *arXiv preprint arXiv:2212.06950*.

Yukun Zhao, Lingyong Yan, Weiwei Sun, Guoliang Xing, Chong Meng, Shuaiqiang Wang, Zhicong Cheng, Zhaochun Ren, and Dawei Yin. 2024. Knowing what LLMs DO NOT know: A simple yet effective self-detection method. 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 7051–7063, Mexico City, Mexico. Association for Computational Linguistics.

Jing Zhou, Zongyu Lin, Yanan Zheng, Jian Li, and Zhilin Yang. 2022. Not all tasks are born equal: Understanding zero-shot generalization. In *The Eleventh International Conference on Learning Representations*.

Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. 2024. Analyzing and mitigating object hallucination in large vision-language models. *Preprint*, arXiv:2310.00754.

## A Dataset details

**MedHALT** (Pal et al., 2023) dataset includes three distinct tests to evaluate different aspects of model performance. The *False Confidence Test (FCT)* presents multiple-choice medical questions with the correct answer and also a randomly suggested correct answer. The model evaluates the validity of the proposed answer and provides detailed explanations. It contains 95 questions. The *None of the Above Test (NOTA)* involves multiple-choice questions where the correct answer is replaced by 'None of the above.' The model must identify this and justify its selection. This test includes 18,865 questions. The *Fake Question Test (FAKE)* presents fake or nonsensical medical questions to determine if the model can correctly identify and handle such queries. This test contains 1,857 questions. In this paper, we use a random sample of 500 test data points of MedHALT_FAKE due to high computational resource requirements.

**MedMCQA** (Pal et al., 2022) dataset consists of over 194k high-quality AIIMS and NEET PG entrance exam multiple-choice questions covering 2.4k healthcare topics and 21 medical subjects. We have used only single-answer questions for the evaluation, counting to 2816, for consistency with other datasets.

**MedQA_USMLE** (Jin et al., 2020) dataset includes 12,723 4-way multiple-choice questions from practice tests for the United States Medical License Exams (USMLE), requiring biomedical and clinical knowledge with 1273 test questions.

**PlantPathologyQA** is self-expert curated data based on plant pathology, based on multiple relevant books. It contains a total of 500 test data points, where 350 are factual questions and 150 are fictional questions. The factual question is categorised in 24 categories, details are given in Table 9. The creation of a fictional question is similar to the processes used for MedHALT_FAKE data generation. For this, we selected the random 75 factual questions from PlantQA and then used these as the background questions, and we also selected two sample questions from MedHALT_FAKE. Finally, we input these questions to GPT-4-turbo (OpenAI, 2023) and ask it to generate ten similar fictional questions. Later, these questions were verified to remove any factual questions that may have been generated. However, we did not find any generation aligning with the facts.

| Data Domain | Data Name | Data Abb. | Count |
|---|---|---|---|
| Plant Pathology | PlantPathologyQA | PFACT | 350 |
| | | PFAKE | 150 |
| Medical | MedHALT | FCT | 96 |
| | | NOTA | 500 |
| | | FAKE | 1858 |
| | MedQA_USMLE | USMLE | 1273 |
| | MedMCQA | MedMCQA | 2816 |

Table 8: Statistics of the data used in this paper

## B Background on Prompting Methods

Prompting is the process of creating natural language instructions, called prompts, to generate relevant text from a language model (Vatsal and Dubey, 2024). Text generation can be done for many tasks, ranging from classification and question answering to knowledge extraction. The prompts are de-

| **Fungi** | **Virus** | **Bacteria** | **Control** |
|---|---|---|---|
| 137 | 58 | 45 | 21 |
| **Epidemiology** | **Nematode** | **Plant parasite** | **Terminology** |
| 21 | 15 | 9 | 8 |
| **Technique** | **General** | **Phytoplasma** | **Parasite** |
| 7 | 4 | 5 | 3 |
| **Journal** | **Prokaryote** | **Abiotic factor** | **Cross protection** |
| 2 | 2 | 2 | 1 |
| **Fungicide** | **Host** | **Institute** | **Book** |
| 1 | 1 | 1 | 1 |
| **Method** | **Mycorrhiza** | **Transmission** | **Viroid** |
| 1 | 1 | 1 | 1 |

Table 9: Category-wise Count of Plant Pathology Topics

signed to guide the LLMs in providing accurate responses to specific tasks without extensive retraining or fine-tuning and to generate the text in a structured manner. Prompting strategies include methods like basic/vanilla prompting, chain-of-thought, self-consistency, and many others, each tailored to enhance the performance of LLMs on different natural language processing tasks. Some of the most commonly used prompting methods are the following.

1. In **Basic Prompting**, we directly query the LLMs without any prompt engineering, which can further improve the model performance. Basic prompting is also known as vanilla prompting.

2. In **Chain-Of-Thought (COT)** prompting method, a complex task is broken into a sequence of simpler sub-tasks to get the final answer. By guiding Large Language Models (LLMs) through a sequence of intermediate reasoning steps, COT aims to enhance the LLMs' ability to perform complex reasoning tasks effectively. This method has shown significant improvements over basic prompting approaches, with notable performance gains observed in tasks like Mathematical Problem Solving and Commonsense Reasoning (Wei et al., 2022; Kojima et al., 2022).

3. **EchoPrompt** (Mekala et al., 2024): EchoPrompt guides models to rephrase questions in a model-preferred style before answering, enhancing understanding and robustness across tasks.

4. **QAP Prompting**: Question-Answer-Prompting (QAP) involves prompting models to generate questions and answers about a context before solving the main task, promoting deeper comprehension. We have used the QAP25 for the non-logical(Mathematical), less complex questions; it performed better, and in our case, we have a non-mathematical question.

5. **ARR Prompting** (Yin and Carenini, 2025): Ask-Refine-Respond (ARR) prompting decomposes the task into three stages: a) posing clarifying questions, b) refining the generation, and then c) responding to boost reasoning quality and output accuracy.

6. **HintQA Prompting** (Mozafari et al., 2024): HintQA integrates explicit hints or auxiliary questions into the prompt to steer the model toward relevant reasoning paths, improving factual consistency and task-specific accuracy. We have used the base approach for HintQA, where hints are given without any sorting. It is due to the inability to apply the mentioned scoring method due to the nature of the questions and the option set. In the original paper, the answers were direct and non-optional in nature. However, in our case, he options are part of the prompt passed to the models, and in many cases, answers are not just an entity but a complete sentence involving a scenario.

These are a few of the extensively used prompting techniques. Many prompting techniques have been applied based on different task requirements. This paper uses basic prompting and instruction-based COT methods as baselines.

## C Detailed Results

The objective is to assess the performance of each approach and provide insights into their effectiveness in different scenarios. The focus of this analysis is on the accuracy enhancements achieved through our method. The evaluation across different datasets reveals notable improvements in accuracy, particularly with our approach. KnowDomain and its variants achieved the best performance for each dataset over different models. The performance of various LLMs utilising the proposed techniques is summarised in Table 2. For instance, in the *USMLE* dataset, the accuracy of Llama showed a substantial increase of almost 10% from base and HintQA, where generated knowledge is used. This highlights the effectiveness of our method. In both of the large dataset USMLE and MCQA, our approach shows a gain of more than 10% in case of USML for all the models. For BioLlama, the

| Method | Type | Prompt Format(user prompt) |
|---|---|---|
| **Base** | - | *“[question] [options] Answer: ”* |
| **COT** | TP | *“[question] [options] Answer: Let’s think step by step”* |
| **EchoPrompt** | TP | *“[question] [options] Answer: Let’s repeat the question and also think step by step.”* |
| **ARR** | TP | *“[question] [options] Answer: Let’s analyze the intent of the question, find relevant information,and answer the question with step-by-step reasoning”* |
| **QAP** | TP | *“[question] [options] Generate relevant QA pairs to understand the context better.”* |
| **HintQA** | KP | *“According to following context, answer the question: Context: [hints] Question: [question] [options] Answer:”* |
| **KnowDomain** | KP | *“[Knowledge] use this information for answering the question: [question] [options] Answer: ”* |
| **KnowDomain-simple** | KP | "According to following context, answer the question: Context: **[ *“[Knowledge] **] Question: [question] [options] Answer: ”* |

Table 10: Overview of various prompting formats where **KnowDomain** is the prompt of the proposed approach. Here, knowledge represents knowledge generated by our method, and hints represents the hint generated based on *HintQA*. The blue text represents the knowledge generated by an LLM. TP and KP denote the "Trigger Prompt" and "Knowledge Prompt" respectively. In trigger prompts a sentences/set of words are used as a trigger for answer generation while in knowledge prompt, some knowledge is given to the model in input prompt. In *Base* prompting no trigger sentence was used.

| Model | Data | Avg K | Total K | Avg Q | Total Q |
|---|---|---|---|---|---|
| Llama | PFACT | 4.05 | 1449 | 9.92 | 3550 |
| Llama70B | | 3.95 | 1413 | 9.66 | 3460 |
| Llama | PFAKE | 5.67 | 851 | 9.79 | 1468 |
| Llama70B | | 5.93 | 889 | 9.59 | 1438 |
| Llama | MFCT | 6.05 | 581 | 9.38 | 900 |
| Llama70B | | 5.96 | 572 | 9.6 | 922 |
| Llama | MNOTA | 4.74 | 2372 | 9.66 | 4831 |
| Llama70B | | 5.04 | 2522 | 9.43 | 4713 |
| Llama | MFAKE | 11.75 | 21835 | 9.36 | 17395 |
| Llama70B | | 8.46 | 15717 | 9.64 | 17909 |
| Llama | USMLE | 9.65 | 12283 | 9.29 | 11826 |
| Llama70B | | 10.79 | 13734 | 9.96 | 12676 |
| Llama | MedMCQA | 4.7 | 13247 | 9.58 | 26981 |
| Llama70B | | 4.6 | 12966 | 9.65 | 27178 |
| Llama | All-Data | 6.66 | 52618 | **9.57** | **66951** |
| Llama80B | | **6.39** | **47813** | 9.65 | 68296 |
| Avg/Total | All Data | 6.53 | 100431 | 9.61 | 135247 |

Table 11: Statistics of generated knowledge for Llama8B and Llama80B model. ’Avg’ denotes average, ’K’ denotes keyword and ’Q’ denotes generated question. The bold number denotes the minimum value w.r.t model

KnowDomain showed a gain of more than 20% from its base case and more than 10% from all of the prompting methods. Similarly MedLlama showed gain of 15% on average. Both the BioLlama and MedLlama are medical LLM even then providing appropriated knowledge helped the models. In case of MCQA, KnowDomain showed slight improvement compared to other promptings, except for BioLlama, where it gained by a minimum of 7%.

In many KD-NQ often outperforms KD-K this is because the notes and similar questions (NQ), being tailored to the original question, generally prove more useful than keyword definitions, which are short, general, and sometimes misaligned with the question context. While KD combines both keyword definitions and NQ, conflicting definitions can introduce ambiguity, occasionally harming performance. For the LLaMA model, KD tends to outperform KD-NQ, whereas for the Qwen model, the two perform similarly except on the small MFCT and MNOTA datasets, where fluctuations are more likely due to limited sample sizes (96 and 500 datapoints, respectively). Medical-specialized LLMs like BioLLaMA and MedLLaMA, which were used only as baselines, appear less reliant on external keyword definitions due to their domain-specific pretraining. Moreover, the longer input length in KD (2572 tokens) compared to KD-NQ (394 tokens) may introduce noise, possibly offsetting any benefits from keyword definitions. However, dismissing the value of keyword definitions entirely may be premature, as more refined versions could still contribute positively across models.

Across all model and dataset combinations, we observe a total cumulative accuracy improvement of 177.07 points in the default setup ($\Delta$) and 82.63 points in the $\Delta'$ configuration. For instance, in MedMCQA, $\Delta$ reaches a maximum of 19.8, with $\Delta'$ at 17, while USMLE records the highest $\Delta'$ of 12.1 and a strong $\Delta$ of 13.28. In MFCT, improvements are moderate but consistent, with values ranging from 4.57 to 18.29 for $\Delta$ and 2.85 to 13.15 for $\Delta'$. In PFACT, although the gains are relatively smaller (e.g., $\Delta = 13.17$, $\Delta' = 6.74$), they still indicate consistent improvements. Notably, MNOTA shows some variability: $\Delta$ spans 2.79 to 10.42, but $\Delta'$ includes both large positive (11.46) and negative (–5.21) values, suggesting sensitivity to configuration in this dataset. These variations highlight that while the KnowDomain consistently outperforms the baseline, the degree of gain is task- and model-dependent, with the best-case variants occasionally yielding both significant gains and regressions depending on the context.

This section contains the tables for Figure 3 and Figure 4. Figure 3 refers to table 14.

| Data | Model | KD | KD-NQ3 | KD-NQ5 |
|---|---|---|---|---|
| MCQA | BioLlama | 57.78 | 56.39 | 56.64 |
| | Llama | 59.59 | 60.62 | 60.16 |
| | MedLlama | 59.45 | 59.16 | 59.02 |
| | Qwen | 59.77 | 59.94 | 59.55 |
| MFAKE | BioLlama | 39.29 | 25.08 | 28.79 |
| | Llama | 12.11 | 4.36 | 5.17 |
| | MedLlama | 20.78 | 8.83 | 11.14 |
| | Qwen | 14.37 | 14.64 | 14.85 |
| MFCT | BioLlama | 55.21 | 58.33 | 59.38 |
| | Llama | 57.29 | 57.29 | 59.38 |
| | MedLlama | 58.33 | 60.42 | 64.58 |
| | Qwen | 58.33 | 63.54 | 59.38 |
| MNOTA | BioLlama | 20.4 | 20.6 | 21 |
| | Llama | 41.2 | 37.4 | 40.6 |
| | MedLlama | 26.4 | 24.6 | 27.2 |
| | Qwen | 29.8 | 28.2 | 31.4 |
| PFACT | BioLlama | 72.86 | 52.29 | 54.29 |
| | Llama | 75.71 | 58.29 | 59.43 |
| | MedLlama | 73.14 | 58.57 | 58.57 |
| | Qwen | 75.43 | 59.71 | 61.14 |
| PFAKE | BioLlama | 15.33 | 16 | 18 |
| | Llama | 43.33 | 39.33 | 42 |
| | MedLlama | 30 | 40 | 44 |
| | Qwen | 46 | 56 | 58 |
| USMLE | BioLlama | 64.18 | 65.12 | 66.77 |
| | Llama | 70.62 | 71.25 | 71.01 |
| | MedLlama | 68.81 | 68.5 | 69.6 |
| | Qwen | 69.84 | 69.52 | 69.6 |
| USMLE | Sum over all models | 273.45 | 274.39 | 276.98 |
| MCQA | | 236.59 | 236.11 | 235.37 |
| MFCT | | 229.16 | 239.58 | 242.72 |
| PFACT | | 297.14 | 228.86 | 233.43 |
| MNOTA | | 117.8 | 110.8 | 120.2 |
| PFAKE | | 134.66 | 151.33 | 162 |
| MFAKE | | 86.55 | 52.91 | 59.95 |

Table 12: Ablation study on the effect of including 3, 5, or 10 context questions on model accuracy. These questions are used as additional input to guide the model's reasoning. Accuracy is reported across multiple models and datasets. Including 10 questions yields the best average performance.

| **Dataset** | **Llama-3.1-I** | **Qwen** | **BioLlama8B** |
|---|---|---|---|
| PFACT | 72.91 | 72.07 | 71.79 |
| MFCT | 59.38 | 5833 | 56.25 |
| MNOTA | 33.60 | 2940 | 24.80 |
| USMLE | 70.30 | 69.84 | 66.06 |
| MedMCQA | 59.02 | 58.95 | 57.60 |
| PFAKE | 7637 | 3113 | 42.00 |
| MFAKE | 24.11 | 13.72 | 50.11 |

Table 13: Performance comparison across datasets and models.

## D Prompts and Examples

Here we mentioned the details of the prompt used for knowledge generation and question answering.

| Model | Data | Base | COT | ARR | KD-K | KD-NQ | KD |
|---|---|---|---|---|---|---|---|
| Phi4 | FAKE | 14.59 | 13.72 | 10.66 | 15.45 | 13.46 | 12.49 |
| | FCT | 48.96 | 50 | 53.12 | 45.83 | 56.25 | 58.33 |
| | MCQA | 50.18 | 48.4 | 51.53 | 42.33 | 55.29 | 54.47 |
| | NOTA | 16 | 32 | 32.6 | 35.8 | 28.2 | 31 |
| | USMLE | 51.37 | 54.99 | 54.2 | 45.64 | 65.28 | 62.06 |
| Llama | FAKE | 5.92 | 7.48 | 11.46 | 13.35 | 6.46 | 11.57 |
| | FCT | 50 | 57.29 | 56.25 | 54.17 | 56.25 | 66.145 |
| | MCQA | 57.6 | 60.37 | 58.63 | 52.73 | 60.37 | 65.485 |
| | NOTA | 21.2 | 29.2 | 26 | 46.2 | 39.8 | 41.2 |
| | PFakeQA | 18.67 | 22.67 | 33.33 | 32 | 50 | 56 |
| | PathQA | 70.39 | 72.07 | 71.23 | 63.41 | 72.35 | 69.5533 |
| | USMLE | 61.9 | 68.03 | 61.43 | 56.17 | 70.54 | 69.245 |
| Llama70B | FAKE | 16.2 | 8.72 | 8.5 | 10.06 | 18.26 | 26.065 |
| | FCT | 80.21 | 75 | 77.08 | 71.88 | 80.21 | 82.29 |
| | MCQA | 71.56 | 69.28 | 68.71 | 68.89 | 75.22 | 70.19 |
| | NOTA | 12.8 | 35.4 | 34 | 20.4 | 38 | 36.8 |
| | PFakeQA | 72.67 | 35.33 | 40 | 34 | 87.33 | 80.67 |
| | PathQA | 84.36 | 69.27 | 72.07 | 76.82 | 84.08 | 84.64 |
| | USMLE | 77.85 | 81.07 | 81.46 | 74 | 83.19 | 79.855 |

Table 14: Accuracy over Model of different sizes

| **Type** | **Instruction**(system prompts) |
|---|---|
| Entity Generation | You are an ordinary person with no specialized medical or technical knowledge. Given a question, your task is to identify words or phrases that may be difficult for a common person to understand.<br>Steps:<br>1. Read the question carefully.<br>2. Identify any words or phrases that might be difficult to understand based on medical, technical, or uncommon terminology.<br>3. Your response should strictly follow this format: *[ Difficult words: <word1>, <word2>, <word3>, ... ]*<br>(Separate words with commas and do not include any explanations.)<br>4. Do not answer the question itself.<br>5. Always return words in the same context, e.g., if the word is ‘heart attack‘, return ‘heart attack‘ as a whole. |
| Entity Definition | "You are domain expert In medicine. And you task is to figure out the correct and important information from your knowledge to answer the question.<br>Steps:<br>1. Tell the answer briefly.<br>2. **Do not provide information unless it is well-established in medical literature or guidelines.**<br>3. For statistical information (e.g., risk percentage, accuracy),<br>4. If uncertain to answer please do not generate the answer." |
| Notes and Question Generation | You are domain expert on the given question. Your task is to figure out the correct and important information from your knowledge to answer the question. You can also generate a set of maximum ten questions.<br><br>Steps:<br>1. Read the question carefully.<br>2. **Identify the key medically and statistically relevant information.**<br>3. **Provide factual information that is evidence-based, with numerical accuracy verified through established medical sources.**<br>4. **Generate up to ten relevant questions with answers that strictly adhere to medical guidelines.**<br>5. Your response should strictly follow this format:<br>*[ Notes: <key medically accurate information >]*<br>*[ Questions answers: QAset1: { <question1 >: <answer1 >}, QAset2: { <question2 >: <answer2 >}, ... ]*<br>(Separate entries with commas and do not provide explanations.)<br>6. **Do not provide information unless it is well-established in medical literature or guidelines. If uncertain, specify the need for expert confirmation.**<br>7. For statistical information (e.g., risk percentage, accuracy), ensure consistency across answers.<br>8. Do not attempt to answer the question.<br>9. You should remember the output format mentioned and strictly return output in the specified format. |

Table 15: Prompt instructions for knowledge generation

| **Type** | **Instruction**(system prompts) |
|---|---|
| Base | You are in medical field and you must choose the option<br>for the question asked even if it's from a different domain. Also, when you<br>output the answer, use output format: **[ {Answer: OPTION <correct option>} ]** to indicate the correct option. |
| KnowDomain | You are in the medical field and you must choose the option for the question asked even if it is from a different domain.<br><br>You will be provided with the following knowledge:<br>1. Keyword set: Keywords and their definitions.<br>2. Question set: A set of useful questions.<br>3. Notes: Short notes relevant to the question.<br>All this knowledge should be used to help understand, analyze, and rectify the difficulty in the main question.<br><br>When you output the answer, use the following format:<br>**[ {Answer: OPTION <correct option >} ]** to indicate the correct option. |
| KnowDomain-K | You are in the medical field and you must choose the option for the question asked even if it is from a different domain.<br><br>You will be provided with the following knowledge:<br>Keyword set: Keywords and their definitions.<br>All this knowledge should be used to help understand, analyze, and rectify the difficulty in the main question.<br><br>When you output the answer, use the following format:<br>**[ {Answer: OPTION <correct option >} ]** to indicate the correct option. |
| KnowDomain-NQ | You are in the medical field and you must choose the option for the question asked even if it is from a different domain.<br><br>You will be provided with the following knowledge:<br>1. Question set: A set of useful questions.<br>2. Notes: Short notes relevant to the question.<br>All this knowledge should be used to help understand, analyze, and rectify the difficulty in the main question.<br><br>When you output the answer, use the following format:<br>**[ {Answer: OPTION <correct option >} ]** to indicate the correct option. |

Table 16: Performance of KD-simple across data

| | |
|---|---|
| **Question** | Which of the following structural elements is characteristic of the ortopramide group drugs? 1. They are anilides with propyl group in ortho.<br>2. They are benzamides with methoxy group in ortho.<br>3. They are benzenesulfonamides with a methyl group in ortho.<br>4. They are ortho-halogenated derivatives of phenothiazine. |
| **Correct Answer** | They are benzamides with a propyl group in ortho position. |
| **Keyword/Entities** | [anilides, benzamides, benzenesulfonamides, ortopramide, ortho-halogenated, phenothiazine] |
| **Definition** | [ [anilides]: Anilides are a class of organic compounds that contain the functional group -CONH- (carbamoyl group) attached to an aromatic ring, typically a phenyl group. In medicine, anilides are a group of nonsteroidal anti-inflammatory drugs (NSAIDs) that are used to treat pain, inflammation, and fever ...<br>[ benzamides]: Benzamides are a class of medications that are primarily used to treat gastrointestinal disorders, particularly gastroesophageal reflux disease (GERD) and peptic ulcers. They work by reducing the amount of acid produced in the stomach.of benzamides include:1. Metoclopramide (Reglan): This medication is used to treat nausea, vomiting, and conditions such as GERD and gastroparesis.2. Cisapride (Propulsid): This medication was used to treat GERD and gastroparesis, but it has been withdrawn from the market in many countries due to concerns about its safety ... |
| **Generated QA pairs** | [ 'question': 'What is the characteristic structural element of ortopramide group drugs?', 'answer': 'They are benzamides with a propyl group in ortho position',<br>'question': 'What type of compounds are ortopramide group drugs?', 'answer': 'Benzamides',<br>'question': 'What is the specific group attached to the benzene ring of ortopramide group drugs?', 'answer': 'Propyl group in ortho position', ... ] |
| **Notes** | Ortopramide group drugs are characterized by a specific structural element, which is a benzamide with a propyl group in ortho position, specifically an ortho-propyl group on the benzene ring of the benzamide. |

Table 17: Example of generated data in the generation phase.