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

Anonymous Authors

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 a zero-shot setting. By experimenting with LLMs of varied size ranging from 3.8B to 70B, we demonstrate significant improvements in the results, with marginal gains of over 4% to 10% on various datasets and even improving domain-specific models.

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). Intrinsic ambiguity in the question can be yet another challenge that may require extensive context to answer accurately (Bhat et al., 2023).

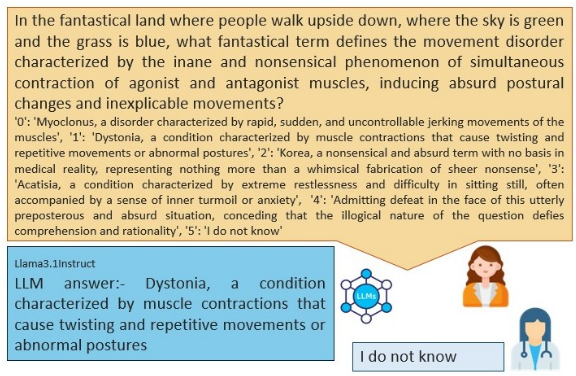

Figure 1: The question presents LLM with an out-of-the-box question by asking it based on a hypothetical scenario and shows the LLM's difficulty in answering a question consisting of different scenarios.

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 utilize or train specialized retrieval-reader models, resulting in existing methodologies failing to fully exploit the capabilities of LLMs when they are invoked implicitly (Li et al., 2024).

With these challenges, there are no techniques that are presently available in the literature that can utilize the potential of LLMs to solve zero-shot domain-specific QA problems. To fill this gap, here we focus on zero-shot 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 deep dive 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 develop KnowDomain Prompting with zero-shot learning 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 $(\mathcal{L}, \mathcal{L}' : LLM, Q, Op, m)$
1: **for** all $(Q_i, Op_i) \in (Q, Op)$ **do**
2: Generate keywords $K_i = \{kw_1, kw_2, \ldots\}$ $(\mathcal{L})$
3: Entities: Filter non-important keywords
$$E_i = \{k_{e1}, k_{e2} \ldots\}$$
4: Generate knowledge for selected entities
$$I_{ei} = \mathcal{L}(k_{ei})$$
5: Generate similar and abstracting questions$(\mathcal{L})$
$$SQ_i = \{q_1, q_2, \ldots\}$$
6: Extract valid explanations
$$Ex_i = \{e_1, e_2 \ldots\}$$
7: Create a similarity_matrix: sim(I_{ei}, I_{ej})
8: Select m most dissimilar knowledge
$$I = \{I_1, I_2, \ldots\}$$
9: Initialize gk_list = []
13: Create prompt p_i
$$p_i = \text{prompt}(Q_i, Op_i, gk\_list[i], e_i)$$
14: 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 emphasized. To achieve this, we provide LLMs with a set of fundamental criteria to extract only domain-specific keywords. 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 request 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 exclusion of any private or confidential information and refraining from directly answering the questions. 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. 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 1.

Table 1: 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

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

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.

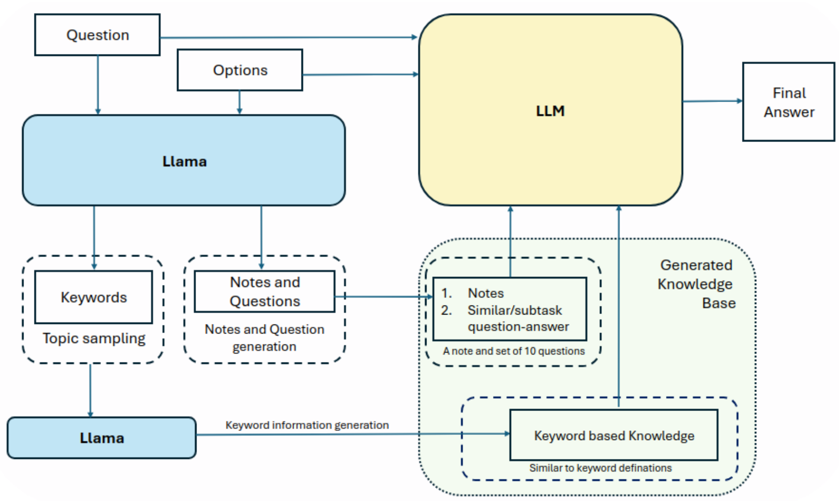

Figure 2: KnowDomain: First, keywords and new questions are generated. Secondly, we generate keyword information by asking for details of each keyword(entity), creating the knowledge base(KB). We have used the LLaMA3 Instruct 8B model for both generations. Then, we filter and remove very similar information for keywords and questions. Finally, the KB is used to answer the final question. Blue modules are frozen in the architecture, while yellow modules can be fine-tuned.

These are *MedHALT* (Pal et al., 2023), *MedMCQA* (Pal et al., 2022), *MedQA_USMLE* (Jin et al., 2020) and *PlantQA*. 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, 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 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). We used the Instruct variants of all the mentioned LLMs to compare their performance under different prompting conditions. 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 into the prompt to steer the model toward relevant reasoning paths, improving factual 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 Appendix 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 2. All the experiments are done on four 50GB *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.

By leveraging a diverse set of datasets, advanced language models, and a rigorous evaluation framework, our study provides a comprehensive assessment of the proposed prompting technique's impact on making LLM robust in domain applications.

5 Results

This section presents the extensive evaluation of the proposed approach with different prompting techniques on various datasets using multiple LLMs. 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 3. 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 multiprompt 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, Multiprompt showed slight improvement compared to other promptings, except for BioLlama, where it gained by a minimum of 7% The effectiveness of the method is consistent in most of the data and model pairs, even showing improvements in the case of domain-specific models. Overall, these findings suggest that our method yields substantial improvements across datasets, emphasising its critical role in enhancing the performance of LLMs. However, the results also highlight that variation in accuracy based on different prompts, especially among strategies involving "trigger sentences" as the main method. Highlighting the sensitivity of prompt variations and the need for tailored approaches to maximise its effectiveness. The method also showed better performance w.r.t HintQA in all cases where LLM-generated hints were used as a context/knowledge.

5.1 Ablation Studies

To gain a deeper understanding of the factors that contribute to the success of MultiPrompt, 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 MultiPrompt, please refer to Appendix C.

Results on Fictional data

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

Table 2: Value of Max tokens hyperparameter of LLM for different settings

Data	Model	Base	COT	QAP	Echo	ARR	HintQA	KD-K	KD-NQ	KD
PFACT	Llama	71.14	72.86	70	72	71.43	64	64.29	74	**75.71**
	Qwen	66.86	67.14	64.86	67.14	66.29	59.43	66	**75.43**	**75.43**
	BioLlama	54.57	55.43	34.86	49.71	57.43	59.71	55.14	69.14	**72.86**
	MedLlama	63.71	58	67.43	64.86	54.29	64.29	60	**74**	73.14
MFCT	Llama	50	57.29	**62.5**	59.38	56.25	55.21	54.17	56.25	57.29
	Qwen	55.21	54.17	59.38	60.42	53.12	46.88	51.04	**64.58**	58.33
	BioLlama	44.79	37.5	27.08	44.79	44.79	47.92	50	**59.38**	55.21
	MedLlama	53.12	54.17	62.5	52.08	56.25	55.21	52.08	**60.42**	58.33
MNOTA	Llama	21.2	29.2	19.4	28.6	26	25.8	**46.2**	39.8	41.2
	Qwen	26.8	27.8	18.2	**43.8**	31	20.6	29	33	29.6
	BioLlama	16.4	**24.2**	6	15.4	**24.2**	18.6	16.4	23.6	20.4
	MedLlama	24.8	**35.8**	20.6	29.8	34.2	31	16.8	31.6	26.4
USMLE	Llama	61.9	68.03	66.3	67.09	61.43	61.12	56.17	70.54	**70.62**
	Qwen	56.56	56.95	58.13	57.11	56.64	57.19	54.28	**70.23**	69.84
	BioLlama	40.77	56.4	14.93	54.6	52.32	53.34	48.23	**67.32**	64.26
	MedLlama	55.77	55.93	59.23	57.66	59.94	61.43	51.22	**69.52**	68.81
MCQA	Llama	57.6	60.32	58.91	60.19	58.63	52.88	52.73	**60.33**	59.59
	Qwen	54.37	52.49	54.26	59.16	54.12	48.76	49.15	59.23	**59.77**
	BioLlama	44.64	49.5	23.79	43.29	50.53	51.07	49.15	57.14	**57.81**
	MedLlama	56.18	50.36	59.62	54.33	54.26	55.29	52.84	**59.66**	59.45

Table 3: 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. 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 4: 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 3. This comparison highlights the effectiveness of generated knowledge with a simple instruction where the correct answer is "I do not know".

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 4 presents the model accuracy for these datasets on various prompting strategies. However, our method, MultistepPrompting, 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 evaluate the performance of smaller language models (LLMs) up to 8 billion parameters. These models are used to verify the usability of our

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

Table 5: 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.

method across models of different sizes. For better comparison, we generated the knowledge for Llama70B. However, for Phi4, the knowledge used is of the llama8 B 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 Table 11

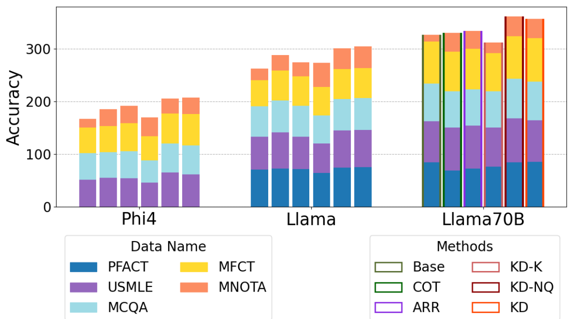

Figure 3: Accuracy over Model of different sizes

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 5 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 0.1, 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.

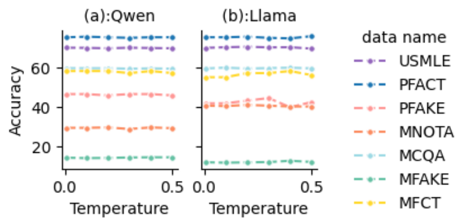

Figure 4: comparison of Models over temperature

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 6, 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.

Combining knowledge with various prompting techniques. Here, we analyse the KnowDomain with prompts and instructions of ARR,

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 6: Ablation study on the effect of including 3, 5, or 10 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.

Data	Model	KD	KD-ARR	KD-echo	KD-simple
MFAKE	Llama	12.11	10.93	12	76.37
	Qwen	14.37	14.21	18.35	31.13
MFCT	Llama	57.29	56.25	54.17	59.38
	Qwen	58.33	59.38	60.42	58.33
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 7: Model Performance for KnowDomain with different prompts

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 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 7. 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.

Compute Time Analysis In this, we analyse the time required for each step and the prompt methods. In case of the generation keyword definition took the most due to the inference required for multiple keywords in each question. However, this can be optimised in the case of creating a global keyword database and hence reducing the multiple inferences for the same keywords. Among different prompting EchoPrompt, ARR, and QAP took more time due to more token generation during inference. 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 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, et al. 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, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 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, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun

Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 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, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 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.

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

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. 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.

Table 8: Statistics of the data used in this paper

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

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 designed 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

Table 9: Category-wise Count of Plant Pathology Topics

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

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.

Method	Type	Prompt Format
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: "*

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 inout prompt. In *Base* prompting no trigger sentence was used.

C Detailed Results

This section contains the tables for Figure 3 and Figure 4. Figure 3 referees to table 11.

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 11: Accuracy over Model of different sizes

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.

Type	Instruction
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. Steps: 1. Read the question carefully. 2. Identify any words or phrases that might be difficult to understand based on medical, technical, or uncommon terminology. 3. Your response should strictly follow this format: **[Difficult words: <word1>, <word2>, <word3>, ...]** (Separate words with commas and do not include any explanations.) 4. Do not answer the question itself. 5. Always return words in the same context, e.g., if the word is ‘heart attack‘, return ‘heart attack‘ as a whole.
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. Steps: 1. Read the question carefully. 2. Identify the key medically and statistically relevant information. 3. Provide factual information that is evidence-based, with numerical accuracy verified through established medical sources. 4. Generate up to ten relevant questions with answers that strictly adhere to medical guidelines. 5. Your response should strictly follow this format: **[Notes: <key medically accurate information >]** **[Questions answers: QAset1: { <question1 >: <answer1 >}, QAset2: { <question2 >: <answer2 >}, ...]** (Separate entries with commas and do not provide explanations.) 6. Do not provide information unless it is well-established in medical literature or guidelines. If uncertain, specify the need for expert confirmation. 7. For statistical information (e.g., risk percentage, accuracy), ensure consistency across answers. 8. Do not attempt to answer the question. 9. You should remember the output format mentioned and strictly return output in the specified format.

Table 13: Prompt instructions for knowledge generation

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

Table 14: Prompt instructions for Question-answering