Building Domain-Specific Small Language Models on a Shoestring via Guided Data Generation

Aman Kumar
Hitachi America Ltd.
Santa Clara, CA, USA
aman.kumar@hal.hitachi.com

Ekant Muljibhai Amin
Hitachi Ltd.
Tokyo, Japan
ekant.amin.mu@hitachi.com

Xian Yeow Lee
Hitachi America Ltd.
Santa Clara, CA, USA
xian.lee@hal.hitachi.com

Lasitha Vidyaratne
Hitachi America Ltd.
Santa Clara, CA, USA
lasitha.vidyaratne@hal.hitachi.com

Ahmed Farahat
Hitachi America Ltd.
Santa Clara, CA, USA
ahmed.farahat@hal.hitachi.com

Yuta Koreeda
Hitachi Ltd.
Tokyo, Japan
yuta.koreeda.pb@hitachi.com

Chetan Gupta
Hitachi America Ltd.
Santa Clara, CA, USA
chetan.gupta@hal.hitachi.com

Abstract

Large Language Models (LLMs) have shown remarkable success in supporting a wide range of knowledge-intensive tasks. In specialized domains, there is growing interest in leveraging LLMs to assist subject matter experts with domain-specific challenges. However, deploying LLMs as SaaS solutions raises data privacy concerns, while many open-source models demand significant computational resources for effective domain adaptation and deployment. A promising alternative is to develop smaller, domain-specialized LLMs, though this approach is often constrained by the lack of high-quality domain-specific training data. In this work, we address these limitations by presenting a cost-efficient and scalable training pipeline that combines guided synthetic data generation from a small seed corpus with bottom-up domain data curation. Our pipeline integrates Domain-Adaptive Pre-training (DAPT), Domain-specific Supervised Fine-Tuning (DSFT), and Direct Preference Optimization (DPO) to train effective small-scale models for specialized use cases. We demonstrate this approach through DiagnosticSLM, a 3B-parameter language model tailored for fault diagnosis, root cause analysis, and repair recommendation in industrial settings. To evaluate model performance, we introduce four domain-specific benchmarks: multiple-choice questions (DiagnosticMCQ), question answering (DiagnosticQA), sentence completion (DiagnosticComp), and summarization (DiagnosticSum). DiagnosticSLM achieves a 13-25% accuracy improvement over open-source models of comparable or larger size (2B-9B) on the MCQ task, while also outperforming or matching them in other tasks, demonstrating strong domain-specific reasoning and generalization capabilities.

KDD'25 SciSoc LLM Workshop, Toronto, ON, Canada

Keywords

Small language models, domain-specific models, parameter-efficient training, guided data generation

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable potential in supporting a wide range of knowledge-intensive tasks across domains such as law [10], finance [12], and medicine [25]. This success has prompted increasing interest in applying LLMs to specialized industrial domains, including fault diagnosis and repair. These domains are often hindered by fragmented documentation, shortage of skilled technicians, and the high cost and time investment required to train new personnel [5, 17]. While technicians performing low-skill operations may rely on manuals and standard procedures, high-skill tasks (such as fault identification, root cause analysis, and repair planning) require deep experiential knowledge that is rarely formalized and is often tacit [3].

Despite the promise of LLMs, there are several barriers to their adoption in industrial settings. Proprietary models like GPT-4o, Gemini, and Claude pose privacy risks due to cloud-only inference and lack of transparency, making them unsuitable for regulated environments. Open-source LLMs, though more accessible, often demand substantial computational resources for fine-tuning and deployment. Additionally, general-purpose models typically lack the domain-specific vocabulary, reasoning capability, and contextual grounding needed to support technicians in high-stakes diagnostic workflows [9].

A promising alternative is to develop small, domain-specialized LLMs that can be deployed on-premise and customized for specific industrial use cases. Prior work on domain-adapted models such as LawGPT [16], BloombergGPT [29], and ChipNeMo [13] has shown that targeted pretraining on specialized corpora significantly improves performance in expert domains. In particular, Domain-Adaptive Pretraining (DAPT) [8] followed by Supervised Fine-Tuning (SFT) [2] has emerged as a cost-effective strategy for domain adaptation.

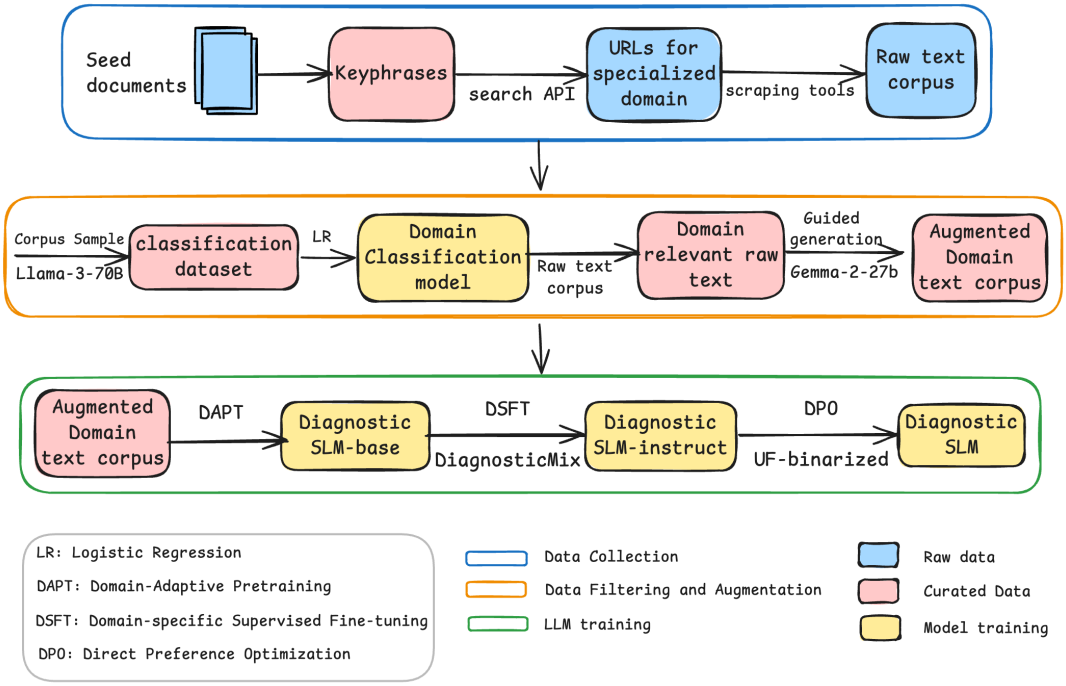

Figure 1: Overview of the DiagnosticSLM pipeline. The figure illustrates the key stages of our approach, including domain-specific data collection, guided synthetic data generation using a teacher model, and a three-stage training process comprising Domain-Adaptive Pretraining (DAPT), Domain-specific Supervised Fine-Tuning (DSFT), and Direct Preference Optimization (DPO).

However, a key bottleneck in this process is access to high-quality domain-specific training data. Industrial data is often siloed, proprietary, and inconsistently documented, limiting the ability to build or evaluate customized models. Moreover, curating clean, relevant subsets from massive corpora like Common Crawl is labor-intensive and prone to low recall [21]. Even with access to data, training from scratch or full fine-tuning of large models is often cost-prohibitive due to the extensive GPU hours and memory requirements involved [30].

To address these challenges, we propose a scalable pipeline for building small domain-specialized LLMs on a shoestring budget.[1] Our approach combines bottom-up domain data curation with guided synthetic data generation from a small seed corpus. We then apply a training pipeline that integrates DAPT with Domain-specific SFT (DSFT), and Direct Preference Optimization (DPO) to produce effective, lightweight models suitable for real-world deployment.

We demonstrate this approach through DiagnosticSLM, a 3B-parameter language model tailored for industrial fault diagnosis and repair, an essential domain where workforce shortages and fragmented documentation hinder productivity and training. We also introduce four domain-specific benchmarks: multiple-choice questions (DiagnosticMCQ), question answering (DiagnosticQA), sentence completion (DiagnosticComp), and summarization (DiagnosticSum), to systematically evaluate the model's performance. An overview of our method is shown in Figure 1.

2 Methodology

We adopted a three-step training pipeline to develop our proposed domain-specific SLM, DiagnosticSLM. First, we perform additional pretraining of an open source SLM, Llama-3.2-3B [6, 15], on a corpus of Automotive domain. Next, we perform supervised fine-tuning using automotive domain related tasks adapted from general supervised fine-tuning alpaca dataset [23]. Lastly, we perform direct preference optimization using the UltraFeedback dataset [4], which we intend to replace using a automotive preference dataset in future work.

2.1 Domain Adaptive Pretraining (DAPT)

2.1.1 Automotive Domain Data curation. We have selected automotive as the main focus area in diagnostics. The first step in curating automotive-specific data involved narrowing the corpus scope to focus on diagnostic procedures, repair operations, and core automotive concepts. We adopted a bottom-up data collection strategy, where keyphrases were curated from a small set of internal documents and technical manuals. Using an internally hosted LLM, we extracted task-relevant keyphrases by prompting it to identify diagnostic procedures, repair tasks, and component-level

[1] For reference, training Llama-3.2-3B from scratch requires an estimated 460,000 GPU hours, whereas our method consumed approximately 5,600 GPU hours, or about 1.2% of that compute cost.

operations within each document. For this task, we used the Llama-3-70B-Instruct model [6] in a 4-bit configuration [14] on 2 × RTX 4090 GPUs, and outputs were manually verified for correctness. These keyphrases were then used to guide focused data search and help reduce ambiguity during search. Examples of representative keyphrases include:

- Perform cylinder power balance tests
- Replace valve stem seals
- Cylinder Head and Valve Train Diagnosis and Repair
- Inspect cylinder deactivation system
- Check bearing preload to inspect, measure, and adjust
- Diagnose noise, vibration, and fluid leakage problems

Using the refined set of unique keyphrases, we queried relevant web domains via the Google Custom Search API. The resulting web scraping process yielded approximately 403 million tokens from 706,971 webpages.

2.1.2 Domain Relevance Filtering and Classification. To further refine the dataset, we performed domain relevance filtering using a combination of LLM annotation and classical classification techniques. A random sample of 20,000 webpages was selected for annotation of the classification dataset. We used Llama-3-70B-Instruct model to label each instance as relevant or irrelevant to the automotive domain. The LLM annotated 11,621 samples as relevant and 8,379 as irrelevant, with the full annotation process taking approximately 11 hours. This highlights the considerable time and computational cost of large-scale LLM-based labeling, making it impractical to annotate the entire dataset manually. A manual review of a subset was conducted to validate label quality. While no corrections were applied, the low observed error rate and minimal review time (2 person-hours) supported continued use of the LLM-labeled data.

This labeled subset was split 80–20 into training and test sets. We trained a logistic regression classifier with L2 regularization ($C = 10$) using the SAGA solver. The resulting model achieved an accuracy of 88.2% on the test set. Applying this classifier to the full dataset yielded 356,312 instances (50.39%) classified as *automotive-related*, and 350,624 (49.61%) as *non-automotive-related*.

To enrich this automotive subset, using domain knowledge, we created eight topic-specific descriptive text chunks representing major automotive systems: Engine Repair, Automatic Transmission, Manual Drive Train and Final Drive, Suspension and Steering, Brakes, Automotive Electrical/Electronics, Automotive Heating and Air Conditioning, and Engine Performance. We then computed the cosine similarity between the embeddings of each topic-specific document and the 356,312 previously classified automotive-related samples, yielding a total of 8 × 356,312 similarity scores. We merged all scored instances, sorted them in decreasing order of similarity, and removed duplicate text entries to ensure uniqueness while preserving high relevance to at least one of the eight topics. The final set consists of unique samples, each associated with a corresponding similarity score. Samples with a cosine similarity greater than 0.25 were selected for inclusion in the main dataset.

To account for possible false negatives in the non-relevant set, we computed cosine similarity between the topic documents and all non-relevant samples. The top 20% most similar entries were retained and merged with the previously filtered set. After filtering and merging, the compiled curated dataset contained 387,572 samples, totaling approximately 257 million tokens.

2.1.3 Guided Synthetic Data Augmentation. To enhance the dataset, we employed a teacher model to expand the automotive content by generating additional domain-relevant text and removing non-relevant portions. This augmentation step aimed to create a more comprehensive and detailed corpus, thereby improving the model's ability to understand and generate automotive-specific information. A relatively smaller model, Gemma-2-27B [24], than a 70B variant was selected to facilitate efficient parallel inference across two-GPUs. For data points exceeding the single-GPU context window, we partitioned the text into smaller chunks, processed them independently, and subsequently merged the outputs.

The augmentation process focused on enriching all relevant samples by prompting the teacher model to add factually accurate details and more comprehensive explanations, guided by its internal automotive knowledge. The underlying assumption is that the teacher model already possesses a certain amount of the necessary domain knowledge; rather than generating information from scratch, the prompts encourage the model to elaborate, clarify, and expand upon existing content. In this way, the teacher model plays a constructive role in enhancing the dataset by injecting technical depth and contextual breadth in a guided manner. This prompt-based strategy not only deepened the technical content and improved specificity across the dataset but also facilitated the filtering of non-relevant samples and removal of low-value, non-technical content inside the sample such as workshop addresses, geographic references, and irrelevant forum conversations. The entire expansion and modification pipeline consumed approximately 5,400 GPU hours. Following augmentation, we applied fuzzy de-duplication using MinHash [11] to remove redundant sentences including content from our automotive benchmark. The final automotive corpus comprised 206 million tokens.

2.1.4 Model and framework. We leveraged a 3B-parameter Llama-3.2-3B model for DAPT using our curated automotive corpus. The model was initialized from its publicly released pretrained weights. We employed the LlamaFactory framework [35] for training and utilized fully sharded data parallelism (FSDP) [32] via the Accelerate library [7] to distribute model parameters across 2 x NVIDIA RTX 4090 GPUs.

2.1.5 DAPT training. We adopted a full-parameter fine-tuning strategy on the automotive domain-specific data. DAPT was conducted using the causal language modeling (CLM) objective, consistent with the pretraining objective of Llama-3 models. CLM can be defined as the negative log-likelihood of the next token given all previous tokens:

$$\mathcal{L}_{\text{CLM}} = -\sum_{t=1}^{T} \log P_\theta(x_t \mid x_{<t}) \tag{1}$$

where x_t is the token at position t, $x_{<t}$ are all preceding tokens, and P_θ is the probability distribution parameterized by the model weights θ.

Full-parameter fine-tuning was chosen over parameter-efficient methods (e.g., LoRA) to maximize domain alignment and capture

deeper semantic shifts specific to the automotive domain. The model was trained using the AdamW optimizer with a learning rate of 1×10^{-4}, and a cosine learning rate scheduler with 10% warmup steps. We set the per-device batch size to 1 and used gradient accumulation over 8 steps, resulting in an effective global batch size of 16 across two GPUs. The input sequence length was set to 2,048 tokens, yielding a total of 16,384 tokens processed per forward pass. Training was conducted for 1 epoch, consisting of 5,789 steps in precision FP16. The entire training process took approximately 59 hours on 2× NVIDIA RTX 4090 GPUs. We call this model *DiagnosticSLM-base*.

2.2 Domain-specific Supervised Fine-Tuning (DSFT)

2.2.1 DSFT Data Generation. To adapt the DAPT model to task-specific objectives, we constructed a domain-specific instruction-tuning dataset for the automotive domain. We curated 10 distinct task types and provided three example prompts per task to the GPT-4o [22] model. These prompts guided the generation of instruction–response pairs across eight automotive topics introduced earlier in the data collection phase (e.g., Engine Repair, Brakes, etc.).

Using this setup, we generated approximately 20,000 examples spanning 10 NLP tasks inspired by Alpaca dataset tailored for the automotive domain, including: Extractive Question Answering, Multiple-Choice Question Answering, Question Generation, Open-Ended Question Answering, True/False Classification, Sentence Completion, Sentiment Analysis, Summarization, Text Generation, and Topic Classification. More descriptions of each task are provided in the Appendix.

To balance domain-specific and general instruction-following capabilities, we combined our domain task dataset with the 52,000 samples from Alpaca dataset. This produced a mixed training set for ablation studies evaluating the effects of different dataset combinations. The final DSFT corpus comprises approximately 72,000 instruction–response pairs, and we refer it as DiagnosticMix.

2.2.2 DSFT Training. We fine-tuned the DAPT model on the DSFT dataset using an auto-regressive next-token prediction objective.

$$\mathcal{L}_{\text{SFT}} = -\sum_{t=1}^{T} \log P_\theta(y_t \mid y_{<t}, x) \tag{2}$$

where x is the instruction prompt and $y = (y_1, \ldots, y_T)$ is the ground-truth response. The model is trained to generate y conditioned on x.

Training was performed using the LlamaFactory framework with nearly identical hyperparameter settings as DAPT, except for a reduced learning rate of 1×10^{-5}. The per-device batch size was set to 2, with gradient accumulation over 8 steps, yielding an effective global batch size of 32.

The model was trained for 1 epoch, totaling 2,130 training steps, using precision FP16 on 2× NVIDIA RTX 4090 GPUs. The complete training process took approximately 23.5 hours. We call this model *DiagnosticSLM-instruct*.

2.3 Direct Preference Optimization (DPO)

While DAPT and DSFT enable the *DiagnosticInstruct* to internalize domain knowledge, they do not guarantee that the model's outputs align with human preferences. To address this, we perform additional fine-tuning through preference alignment using the DPO [20]. DPO is a recently proposed alternative to Reinforcement Learning from Human Feedback (RLHF) [18], that directly optimizes a language model to prefer responses that align with human-annotated preferences, without requiring a separate reward model or reinforcement learning loop. Given a set of preference pairs $(x, y_{\text{pos}}, y_{\text{neg}})$, where x is the input and y_{pos} and y_{neg} are the preferred and less preferred responses respectively, DPO fine-tunes the model by minimizing the following loss:

$$\mathcal{L}(\theta) = -\log \sigma\left(\beta \left(\log \pi_\theta(y_{\text{pos}} \mid x) - \log \pi_\theta(y_{\text{neg}} \mid x)\right)\right), \tag{3}$$

where π_θ is the model's output distribution, $\sigma(\cdot)$ is the sigmoid function, and β is a temperature parameter controlling the sharpness of preference. This encourages the model to increase the log-probability gap between preferred and non-preferred responses. Compared to RLHF, DPO offers a simpler and more stable training paradigm, while maintaining strong empirical performance. We adopt DPO in our alignment stage due to its computational efficiency, reduced implementation complexity, and its ability to effectively guide the model toward preferred behaviors.

2.3.1 Dataset. We utilize the UltraFeedback Binarized dataset, sourced through HuggingFace [4]. The original UltraFeedback dataset comprises approximately 64,000 samples, each accompanied by four responses generated by a mixture of proprietary and open-source language models. These completions were evaluated by GPT-4 based on multiple criteria, including helpfulness and honesty. In the binarized version, the highest-scoring response is labeled as the "chosen" response, while one of the remaining three is randomly selected as the "rejected" response. This binary format enables the construction of pairwise preference data required for DPO.

We use this dataset to further fine-tune the DiagnosticInstruct model. While this dataset is general-purpose, future work will focus on constructing a domain-specific preference dataset tailored to the automotive domain.

2.3.2 DPO Fine-Tuning. We fine-tune the Llama-3.2-3B based DiagnosticSLM instruct model using DPO with LoRA, a parameter-efficient approach selected due to computational constraints. Training is conducted for 1 epoch with a per-device batch size of 1, a learning rate of 1×10^{-5}, and a cosine learning rate scheduler with a 10% warm-up ratio. We use bfloat16 precision training and apply a sigmoid-based preference loss with $\beta = 0.1$. The input sequence length is capped at 2,048 tokens, and data preprocessing is parallelized across 8 workers. Evaluation is performed every 500 steps, with 10% of the dataset reserved for validation. The entire DPO training process took approximately 72 hours. We refer this final model as *DiagnosticSLM*.

3 Experiments

3.1 Evaluation

The Automotive Service Excellence (ASE) certification exams are widely regarded as the standard for assessing technical proficiency in automotive diagnostics, repair, and maintenance across various vehicle systems [26]. To evaluate the domain-specific knowledge

Table 1: Comparison of model variants on the DiagnosticMCQ evaluation using 5-shot prompting.

Model	Params	Training Type	Accuracy (Correct/Total)	Option A	Option B	Option C	Option D
Ground Truth (label dist.)	–	–	–	240	246	205	185
DiagnosticSLM (ours)	**3b**	**DAPT+DSFT+DPO**	**0.4532 (397/876)**	**288**	**311**	**201**	**76**
Phi-3.5-mini-instruct	3.8b	Original Instruct	0.4384 (384/876)	105	203	372	196
Phi-4-mini-instruct	3.8b	Original Instruct	0.4098 (359/876)	91	200	364	221
Gemma-2-2b-It	2b	Original Instruct	0.3733 (327/876)	265	83	317	211
Llama-3.2-3B-Instruct	3b	Original Instruct	0.3653 (320/876)	166	130	426	154
Qwen2.5-3B-Instruct	3b	Original Instruct	0.3630 (318/876)	115	154	352	223
Gemma-2-2b	2b	Base	0.3368 (295/876)	173	406	166	131
Qwen2.5-3B	3b	Base	0.3288 (288/876)	112	122	414	161
Llama-3.2-3B	3b	Base	0.2705 (237/876)	155	63	621	37
DiagnosticSLM-base (ours)	3b	DAPT	0.2352 (206/876)	4	3	868	1
Llama-3.1-8B-Instruct	**8b**	**Original Instruct**	**0.4692 (411/876)**	**136**	**220**	**305**	**215**
Gemma-2-2b-It	9b	Original Instruct	0.4521 (396/876)	147	152	338	162
Ministral-8B-Instruct-2410	8b	Original Instruct	0.4281 (375/876)	153	118	385	216
Llama-3.1-Tulu-3-8B	8b	Original Instruct	0.4041 (354/876)	140	241	129	366
c4ai-command-r7b-12-2024	7b	Original Instruct	0.3984 (349/876)	216	241	331	87
Llama-3.1-8B	8b	Base	0.3095 (247/876)	171	110	402	46

Table 2: Accuracy comparison of different models evaluated on the DiagnosticQA dataset using 5-shot prompting.

Model	Params	Accuracy
DiagnosticSLM (ours)	**3B**	**0.3831**
Phi-4-mini-instruct	3.8B	0.3628
Qwen2.5-3B-Instruct	3B	0.3299
Gemma-2-2B-it	2B	0.3169
Phi-3.5-mini-instruct	3.8B	0.2342
Llama-3.2-3B-Instruct	3B	0.2208

coverage of our models, we construct benchmarks inspired by the format and subject areas of the ASE exams. Given their comprehensive scope and relevance, we manually curated these benchmarks to reflect domain-specific knowledge based on ASE exam structure and content, enabling a quantitative assessment of model performance. We evaluate performance across four tasks: multiple-choice questions, question answering, sentence completion, and summarization, as detailed below. For all experiments, we compare our model with Llama3.2 [15], Qwen2.5 [31], Phi-3.5/Phi-4 [1], and Gemma-2 [24].

3.1.1 Multiple Choice Question Task. We introduce DiagnosticMCQ, a benchmark composed of 876 multiple-choice questions across various automotive subtopics. Each question includes four answer choices, one of which is designated as the correct ground-truth answer, as shown in Figure 2. Each evaluation prompt includes five example question–answer pairs followed by a test question to guide the model in understanding the structure and answer format. Model predictions are compared against the ground truth answers to compute accuracy.

We compare various open-source language models ranging in size from 2B to 9B parameters on the DiagnosticMCQ benchmark. As shown in Table 1, our final model, DiagnosticSLM, achieves an accuracy of 45.32% (397/876), surpassing the Llama-3.2-3B-instruct baseline (36.53%) by a substantial margin. Despite its smaller size, DiagnosticSLM also outperforms several larger models, including Ministral (8B, 42.81%), Llama3.1-Tulu (8B, 40.41%), and C4ai-command-r (7B, 39.84%), and performs on par with Gemma-2 (9B, 45.21%). It also exceeds the performance of similarly sized or slightly larger models such as Phi-4-mini-instruct (3.8B, 40.98%), Qwen2.5-3B-Instruct(3B, 36.3%), and Gemma-2 (2B, 37.33%).

We observed that nearly all base versions of models exhibit skewed option selection when evaluated on DiagnosticMCQ, often favoring a specific choice regardless of the question content. This trend shifts noticeably in their instruction-tuned variants. We notice a similar trend in our own model: after DAPT, the model only selects option C. However, following DSFT and DPO, the answer distribution becomes more balanced. For comparison, the distribution of correct options in the ground truth is also included in the Table 1.

3.1.2 Question-Answering Task. LLMs often exhibit selection biases i.e., token bias toward label tokens (A/B/C/D) [19, 27], positional bias in answer ordering [34], and context priming from in-context examples [33]. To introduce variation in evaluation and better assess the instruction-following capabilities of the models, we converted the *DiagnosticMCQ* dataset into a question-answering (QA) format using GPT-4o, referred as *DiagnosticQA* benchmark. In this setting, the answer choices are embedded directly within the question in natural language. An example question is shown in Figure 2. This format aims to reduce surface-level biases by requiring models to interpret the full input and generate an explicit answer, rather than

Table 3: Accuracy comparison of different models evaluated on the DiagnosticComp dataset using log-likelihood-based scoring.

Model	Params	Accuracy
DiagnosticSLM (ours)	**3B**	**0.5556**
Phi-4-mini-instruct	3.8B	0.5128
Phi-3.5-mini-instruct	3.8B	0.5076
Llama-3.2-3B-Instruct	3B	0.4786
Gemma-2-2B-it	2B	0.4701
Qwen2.5-3B-Instruct	3B	0.4188

selecting a label. We evaluate models using accuracy, comparing their generated answers against the ground truth.


```
{
    "id": 1,
    "DiagnosticMCQ_question": "Using a diagnostic strategy for engine repair,
        ↪ which of the following is generally the last step in that process
        ↪ ?",
    "DiagnosticMCQ_option0": "a. Checking vehicle history",
    "DiagnosticMCQ_option1": "b. Verifying the repair",
    "DiagnosticMCQ_option2": "c. Doing service checks",
    "DiagnosticMCQ_option3": "d. Verifying the concern",
    "DiagnosticMCQ_ground_truth_label": 1,
    "DiagnosticQA_question": "Using a diagnostic strategy for engine repair,
        ↪ which of the following is generally the last step in the process:
        ↪ checking vehicle history, verifying the repair, doing service
        checks, or verifying the concern?"
    "DiagnosticComp_sentence": "The most likely first step in a diagnostic
        ↪ strategy or scientific process of elimination for engine service
        ↪ is",
    "DiagnosticComp_sentence0": "The most likely last step in a diagnostic
        ↪ strategy or scientific process of elimination for engine service
        ↪ is checking vehicle history",
    "DiagnosticComp_sentence1": "The most likely first step in a diagnostic
        ↪ strategy or scientific process of elimination for engine service
        ↪ is verifying the repair",
    "DiagnosticComp_sentence2": "The most likely first step in a diagnostic
        ↪ strategy or scientific process of elimination for engine service
        ↪ is doing service checks",
    "DiagnosticComp_sentence3": "The most likely first step in a diagnostic
        ↪ strategy or scientific process of elimination for engine service
        ↪ is verifying the concern",
}
```


Figure 2: JSON representation of an example from our evaluation datasets.

To assess performance in this free-form QA setting, we evaluated multiple models with similar parameter sizes. As shown in Table 2, all models experienced a drop in accuracy when shifting from MCQ to QA format. For example, *DiagnosticSLM* dropped from 45.3 to 36.92, *Phi-4* from 40.98 to 35.80, *Qwen* from 36.3 to 33.06, *Gemma-2-2B-it* from 37.33 to 28.70, *Phi-3.5* from 43.84 to 23.42, and *Llama-3.2-3B-instruct* from 36.53 to 21.42. Despite the overall performance degradation observed in the QA setting, DiagnosticSLM consistently outperformed all other models by a substantial margin.

3.1.3 Sentence Completion Task. With the aim to further assess domain-specific knowledge of models with reduced bias, we introduce another benchmark *DiagnosticComp*. Each question in *DiagnosticMCQ* is reformulated into four natural language sentences, each representing an independent completion prompt. This is done by taking the shared question stem as a prefix and appending each answer option to it. The resulting completions are generated using GPT-4o. Figure 2 illustrates a sample prefix along with its four corresponding completions.

To evaluate for each sentence prompt, we extract the model's output logits at every generation step, apply a softmax over the vocabulary dimension to obtain token-level probabilities $P(x_i \mid x_{<i})$, and compute the *aggregate log-likelihood*.

$$\log P(x_{1:n} \mid \text{prefix}) = \sum_{i=1}^{n} \log P(x_i \mid x_{<i}). \tag{4}$$

Because all four prompts share the same prefix, differences in total log-likelihoods reflect only the option texts. We select the option whose prompt attains the highest probability score.

Answering certain MCQs demands multi-step or implicit reasoning, which can obscure the model's domain expertise. To focus on surface-level domain knowledge, we manually filter and exclude such questions based on two criteria: (1) complexity patterns such as multi-step reasoning and negation chaining, and (2) option length. We manually review the DiagnosticMCQ dataset and remove examples we classify as *complex*. Option length can introduce bias, as longer sequences tend to accumulate lower total log-likelihoods. To mitigate this, we restrict each candidate option to a maximum of four words and exclude any MCQ containing a longer option. After applying both filtering strategies, the final DiagnosticComp set comprises 117 completion questions. As shown in Table 3, DiagnosticSLM performs better than other models in comparison, with 55.56% accuracy on DiagnosticComp benchmark.

3.1.4 Summarization Task. To evaluate models' ability to summarize domain-specific technical content, we introduce a benchmark task called *DiagnosticSum*. Each input consists of a 3–6 line ground-truth explanation associated with a question from the DiagnosticMCQ dataset. Using GPT-4o, we generate concise two-line summaries of these explanations to serve as reference outputs. Models are prompted to produce two-line summaries of the input paragraphs, with the objective of preserving key technical information in a compact form.

We evaluate summarization quality using a combination of lexical and semantic similarity metrics, including ROUGE-1, ROUGE-2, ROUGE-L, BLEU, BERTScore (F1), and Cosine Similarity. ROUGE and BLEU scores are computed using the rouge-score and nltk libraries, respectively, while BERTScore and Cosine Similarity are calculated using the bert-score and sentence-transformers libraries with the all-MiniLM-L6-v2 [28] embedding model. Given the creative nature of summarization, we conduct five independent trials per model using a temperature setting of 0.5 to account for variability in outputs. The aggregated results are reported in Table 4. We observe that Phi-4-mini-instruct achieves the highest performance among all models, while DiagnosticSLM remains competitive with models of similar size. One possible reason for the relatively lower performance of DiagnosticSLM is the limited number of domain-specific training examples related to the summarization task.

3.2 Ablation study on the effect of dataset on model training

To identify the most suitable base model for our pipeline, we initially experimented with three candidate models: Gemma-2B, Llama-3.2-1B, and Llama-3.2-3B. Each model underwent the same training pipeline consisting of DAPT and DSFT. We experimentally

Table 4: Performance comparison of models on DiagnosticSum benchmark dataset (mean ± std).

Model	Rouge-1	Rouge-2	Rouge-L	BLEU	BERTScore F1	Cosine Sim
DiagnosticSLM (ours)	0.5076 ± 0.0035	0.2691 ± 0.0024	0.3917 ± 0.0024	0.1815 ± 0.0029	0.9196 ± 0.0006	0.8522 ± 0.0019
Gemma-2-2B-It	0.5202 ± 0.0029	0.2654 ± 0.0028	0.3889 ± 0.0023	0.1938 ± 0.0030	0.9199 ± 0.0007	0.8583 ± 0.0021
Llama-3.2-3B-Instruct	0.4379 ± 0.0032	0.2310 ± 0.0024	0.3296 ± 0.0025	0.1167 ± 0.0019	0.8998 ± 0.0011	0.8229 ± 0.0019
Phi-3.5-mini-instruct	0.3796 ± 0.0018	0.2025 ± 0.0018	0.2796 ± 0.0015	0.1067 ± 0.0005	0.8902 ± 0.0020	0.8200 ± 0.0018
Phi-4-mini-instruct	**0.5863 ± 0.0007**	**0.3455 ± 0.0024**	**0.4671 ± 0.0022**	**0.2757 ± 0.0031**	**0.9316 ± 0.0002**	**0.8882 ± 0.0013**
Qwen2.5-3B-Instruct	0.4751 ± 0.0022	0.2448 ± 0.0017	0.3557 ± 0.0020	0.1386 ± 0.0015	0.9151 ± 0.0004	0.8802 ± 0.0010

Table 5: Ablation study on different datasets evaluated on DiagnosticMCQ using 5-shot prompting. *Note:* SFT refers to fine-tuning on the Alpaca dataset; DSFT refers to fine-tuning on the DiagnosticMix dataset.

Model Name	Training Type	Accuracy
Llama-3.2-3B-Instruct	Original Instruct	36.53
Llama-3.2-3B	Base+SFT	30.71
Llama-3.2-3B	Base+DAPT+SFT	37.79
Llama-3.2-3B	Base+DSFT	38.24
DiagnosticSLM-instruct (ours)	Base+DAPT+DSFT	44.41
DiagnosticSLM (ours)	**Base+DAPT+DSFT+DPO**	**45.32**

observed that both Gemma-2B and Llama-3.2-1B exhibited a decline in accuracy on DiagnosticMCQ following DAPT and DSFT. In contrast, Llama-3.2-3B consistently demonstrated superior performance across the task. Based on these results, we selected Llama-3.2-3B as the foundation for all subsequent experiments.

To evaluate the individual and combined effects of DAPT, DSFT, and DPO, we conducted a series of ablation studies using the Llama-3.2-3B model. Our results, as shown in Table 5, demonstrate that neither DAPT nor DSFT alone is sufficient to achieve optimal performance. Specifically, applying DAPT followed by SFT using general-purpose Alpaca dataset yields a modest improvement in DiagnosticMCQ accuracy (37.79) compared to the original instruction-tuned model (36.53), indicating that DAPT introduces some domain-awareness. Similarly, fine-tuning the base model directly with DiagnosticMix data without prior DAPT results in a comparable DiagnosticMCQ accuracy of 38.24, suggesting that task-specific supervision alone is also moderately effective. However, combining DAPT with DiagnosticMix DSFT leads to a substantial performance boost, achieving an DiagnosticMCQ accuracy of 44.41, demonstrating that domain pretraining and domain-specific supervision work synergistically. Further incorporating DPO pushes the DiagnosticMCQ accuracy even higher to 45.32, underscoring the added value of preference alignment in improving model behavior.

4 Conclusion

We introduced *DiagnosticSLM*, a small language model tailored for diagnostics and repair to assist frontline workers, and is lightweight enough for edge deployment in industrial environments. Our three-stage training pipeline consisting of Domain-Adaptive Pretraining (DAPT), Domain-specific Supervised Fine-Tuning (DSFT), and Direct Preference Optimization (DPO) enabled effective domain adaptation using curated and augmented data. Across tasks such as multiple-choice questions, question-answering, sentence completion and summarization, DiagnosticSLM outperformed or matched larger open-source models, underscoring the impact of specialized training on domain-specific corpora. Our ablation study highlights the complementary roles of DAPT, DSFT, and DPO in enhancing model's performance.

References

[1] Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905* (2024).

[2] Shunichi Akatsuka, Aman Kumar, Xian Yeow Lee, Lasitha Vidyaratne, Dipanjan Dipak Ghosh, and Ahmed K Farahat. [n. d.]. Rule-Guided Language Model Alignment for Text Generation Management in Industrial Use Cases. In *Neurips Safe Generative AI Workshop 2024.*

[3] Linda Argote and Paul Ingram. 2000. Knowledge transfer: A basis for competitive advantage in firms. *Organizational behavior and human decision processes* 82, 1 (2000), 150–169.

[4] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. UltraFeedback: Boosting Language Models with High-quality Feedback. arXiv:2310.01377 [cs.CL]

[5] Zhiwei Gao, Carlo Cecati, and Steven X Ding. 2015. A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches. *IEEE transactions on industrial electronics* 62, 6 (2015), 3757–3767.

[6] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[7] Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. 2022. Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate.

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

[9] Gholamreza Khodabandelou, Reza Khodabandelou, Mahboubeh Ghasemi, Mehdi Aghaei, et al. 2024. Challenges and Barriers of Using Large Language Models (LLM) Such as ChatGPT for Diagnostic Medicine: A Scoping Review. *Journal of Clinical Medicine* 13, 2 (2024), 365.

[10] Jinqi Lai, Wensheng Gan, Jiayang Wu, Zhenlian Qi, and Philip S Yu. 2024. Large language models in law: A survey. *AI Open* (2024).

[11] Johannes Leveling, Lennard Helmer, Benny Joerg Stein, Dennis Wegener, Zoha Sheikh, Elanton Fernandes, and Hammam Abdelwahab. 2024. Evaluation of Document Deduplication Algorithms for Large Text Corpora. In *International Conference on Machine Learning, Optimization, and Data Science*. Springer, 390–404.

[12] Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. 2023. Large language models in finance: A survey. In *Proceedings of the fourth ACM international conference on AI in finance*. 374–382.

[13] Mingjie Liu, Teodor-Dumitru Ene, Robert Kirby, Chris Cheng, Nathaniel Pinckney, Rongjian Liang, Jonah Alben, Himyanshu Anand, Sanmitra Banerjee, Ismet Bayraktaroglu, et al. 2023. Chipnemo: Domain-adapted llms for chip design. *arXiv preprint arXiv:2311.00176* (2023).

[14] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods. https://github.com/huggingface/peft.

[15] AI Meta. 2024. Llama 3.2: Revolutionizing edge AI and vision with open, customizable models. *Meta AI Blog. Retrieved December* 20 (2024), 2024.
[16] Ha-Thanh Nguyen. 2023. A brief report on lawgpt 1.0: A virtual legal assistant based on gpt-3. *arXiv preprint arXiv:2302.05729* (2023).
[17] Ikujiro Nonaka and Hirotaka Takeuchi. 1996. The knowledge-creating company: How Japanese companies create the dynamics of innovation. *Long range planning* 29, 4 (1996), 592.
[18] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35 (2022), 27730–27744.
[19] Pouya Pezeshkpour and Estevam Hruschka. 2023. Large language models sensitivity to the order of options in multiple-choice questions. *arXiv preprint arXiv:2308.11483* (2023).
[20] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems* 36 (2023), 53728–53741.
[21] Steffen Remus and Chris Biemann. 2016. Domain-specific corpus expansion with focused webcrawling. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. 3607–3611.
[22] John Schulman, Barret Zoph, Christina Kim, Jacob Hilton, Jacob Menick, Jiayi Weng, Juan Felipe Ceron Uribe, Liam Fedus, Luke Metz, Michael Pokorny, et al. 2022. ChatGPT: Optimizing language models for dialogue. *OpenAI blog* 2, 4 (2022).
[23] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.
[24] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118* (2024).
[25] Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. 2023. Large language models in medicine. *Nature medicine* 29, 8 (2023), 1930–1940.
[26] John Thompson. 2014. *The perceived benefits of Automotive Service Excellence (ASE) certifications for graduates of four-year Automotive Technology programs*. University of Arkansas.
[27] Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926* (2023).
[28] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems* 33 (2020), 5776–5788.
[29] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564* (2023).
[30] Yuchen Xia, Jiho Kim, Yuhan Chen, Haojie Ye, Souvik Kundu, Cong Callie Hao, and Nishil Talati. 2024. Understanding the performance and estimating the cost of llm fine-tuning. In *2024 IEEE International Symposium on Workload Characterization (IISWC)*. IEEE, 210–223.
[31] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115* (2024).
[32] Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. 2023. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277* (2023).
[33] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate Before Use: Improving Few-shot Performance of Language Models. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*. PMLR, 12697–12706.
[34] Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large Language Models Are Not Robust Multiple Choice Selectors. In *The Twelfth International Conference on Learning Representations*.
[35] Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*. Association for Computational Linguistics, Bangkok, Thailand. http://arxiv.org/abs/2403.13372

A Automotive Tasks

We designed a set of diverse NLP tasks in the automotive context. The tasks are described below:

- **Extractive Question Answering** – Identifying answers from a given automotive text.
- **Multiple-Choice Question Answering** – Selecting the correct answer from predefined options.
- **Question Generation** – Creating domain-specific questions based on a given text.
- **Open-Ended Question Answering** – Providing free-form responses to automotive queries.
- **True/False Classification** – Verifying the correctness of domain-specific statements.
- **Sentence Completion** – Predicting missing parts of automotive related text.
- **Sentiment Analysis and Summarization** – Detecting opinions in automotive discussions and summarizing content.
- **Text Generation** – Producing coherent, domain-specific automotive content.
- **Topic Classification** – Categorizing content into automotive subdomains (e.g., engine, transmission, brakes).