Predicting Maintenance Actions from Historical Logs using Domain-Specific LLMs

Aman Kumar[1], Ahmed Farahat[1], and Chetan Gupta[1]

[1]*Hitachi America Ltd., Santa Clara, California, USA*

Aman.kumar@hal.hitachi.com, Ahmed.farahat@hal.hitachi.com, Chetan.Gupta@hal.hitachi.com

ABSTRACT

Maintenance logs of complex specialized equipment capture problem–action records that are essential for building predictive maintenance solutions but remain difficult to utilize due to their terse, abbreviation-heavy style. This work provides the first systematic benchmark and domain-adaptation study of large language models (LLMs) for predicting maintenance actions from free-text problem descriptions in the MaintNet aviation dataset. We evaluate a range of proprietary and open-source LLMs under zero-shot and few-shot prompting and additionally fine-tune selected open models for supervised evaluation. Experiments are conducted on both raw-abbreviation and expanded datasets, using both lexical (ROUGE, BLEU) and semantic (cosine similarity, BERTScore) metrics. Results show that GPT-4o achieves the strongest semantic alignment, while the instruct version of Gemma-3-4B leads in lexical overlap. Few-shot prompting boosts weaker models disproportionately, narrowing the gap with stronger baselines. Fine-tuning delivers the most significant gains, with instruct versions of Gemma-3-4B, LLaMA-3.2-3B, and Phi-4-mini, improving BLEU by up to 90% and ROUGE-2 by 30%. Notably, the fine-tuned Gemma-3-4B surpasses GPT-4o across multiple metrics, demonstrating the effectiveness of domain-specific adaptation. These findings highlight the potential of fine-tuned LLMs to utilize unstructured aviation logs for building reliable maintenance systems.

1. INTRODUCTION

As digital technologies advance, engineering systems generate vast volumes of data increasingly leveraged to enhance performance and reliability. Among these sources, maintenance logs are particularly significant, especially in aviation, where safety and efficiency are paramount (Tanguy et al., 2016; Altuncu et al., 2018). Such logs, often kept as event records, capture valuable problem–action information that can enable predictive maintenance, helping organizations anticipate failures, mitigate risks, and reduce costs (Jarry et al., 2020; Meunier-Pion et al., 2024).

Maintenance logs of complex specialized equipment (e.g., aviation), however, present unique challenges: they are written in terse, domain-specific language, laden with abbreviations and non-standard spellings. Each entry typically pairs a problem description with the corrective action taken, providing a natural but underutilized basis for building predictive models. Automatically predicting likely corrective actions from such problem descriptions could support technicians, improve turnaround times, and enhance decision-making in safety-critical contexts.

Early research applied traditional NLP tools to this domain, focusing on preprocessing and classification. Akhbardeh et al. (2020) introduced the MaintNet toolkit with domain-specific spell-checkers and part-of-speech taggers, significantly outperforming general-purpose tools like NLTK or CoreNLP. Other studies classified log entries into problem or fault categories (Tanguy et al., 2016) but faced severe class imbalance where a few common classes dominated. To address this, Akhbardeh et al. (2021) proposed specialized resampling strategies adapted from computer vision, improving accuracy on rare issue types. These efforts underscore the difficulty of applying standard supervised learning to sparse, noisy maintenance data.

Beyond classification, unsupervised and retrieval methods have sought to uncover latent structure in logs and support technicians with past examples. MaintNet researchers showed that clustering techniques like DBSCAN and k-means could reveal recurring issue groups such as engine-related clusters. Payette et al. (2025) and Vidyaratne et al. (2024) combined text mining with Failure Modes and Effects Analysis (FMEA), mapping free-text to structured failure modes and improving extraction of affected components and causes. More recently, retrieval-based systems using sentence embeddings have been applied: Sundaram et al. (2024) and Naqvi et al. (2024) demonstrated semantic search that retrieves similar historical cases and suggests candidate corrective actions, illustrating the value of domain-trained embedding models for maintenance decision support.

In addition to retrieval-based approaches, researchers have explored transfer learning and, more recently, large language models (LLMs). Akhbardeh et al. (2022) and Naqvi et al. (2021) showed that intra-domain transfer (e.g., between aviation datasets) yields consistent gains, while cross-domain transfer (e.g., from automotive to aviation) often degrades performance. These findings emphasize the importance of domain-specific adaptation, as logbooks across industries diverge sharply in vocabulary and style. Parallel to these advances, LLMs have emerged as powerful tools for handling unstructured technical text (Lukens et al. 2024). Kelma et al. (2025), for instance, demonstrated that GPT-4-based models can automatically generate structured assembly instructions for cognitive assistance systems, achieving expert-level quality when evaluated with BLEU and METEOR.

Despite these advances, current methods still struggle with rare or unseen issues; the very cases where decision support is most critical. Retrieval systems depend on close historical matches, while classifiers often fail on long-tail categories. LLMs offer strong potential, but their ability to predict corrective actions from maintenance logs has not been systematically studied. This work presents the first comprehensive benchmark of LLMs and develops domain-specific language models for aviation maintenance action prediction. Our contributions are twofold: (1) We compare proprietary and open-source LLMs under zero-shot and few-shot prompting to assess baseline capabilities, (2) We fine-tune selected open models on progressively larger training splits, analyzing how domain adaptation scales and whether compact models can surpass GPT-4o. We evaluate performance with both lexical (ROUGE-1, ROUGE-2, BLEU) and semantic (cosine similarity, BERTScore) metrics, providing a comprehensive assessment of LLM-generated maintenance actions in this safety-critical settings.

The remainder of this paper is structured as follows. Section 2 introduces the dataset and problem formulation. Section 3 details the experimental setup, including model selection, fine-tuning objectives, and evaluation metrics. Section 4 presents results across zero-shot, few-shot, and fine-tuned settings. Section 5 provides a discussion of key findings, and Section 6 concludes with implications and future directions.

2. DATA AND PROBLEM FORMULATION

In this work, we focus on the Aircraft Historical Maintenance Dataset (2012–2017) from the University of North Dakota's aviation program, a corpus of 6,169 maintenance logbook entries released via the MaintNet repository (Akhbardeh et al. 2020). Each entry has free-text "Problem" and "Action" fields describing maintenance issues and the corrective actions taken. These log entries are typically short, domain-specific texts written by mechanics or pilots, often containing technical jargon and abbreviations.

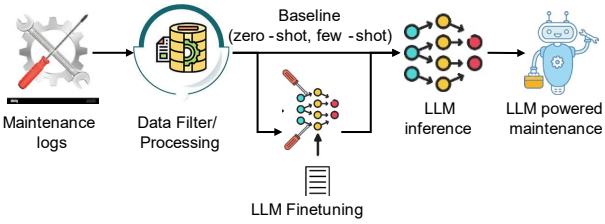

Figure 1: Overview of workflow for generating maintenance actions from historical logs.

Of the 6,169 entries, 5,122 are unique problem-action pairs, and within these, 3,595 are unique problems. Since our objective is a prediction task where the model must generate an action given a problem, we restricted the dataset to entries where each problem maps to exactly one action. This filtering yielded 2,859 entries with a one-to-one mapping, while the remaining 736 entries contained multiple actions (ranging from 2 to 19 per problem). Consequently, our experiments were conducted on the 2,859 single-action samples.

The textual data is domain-specific (aircraft parts and maintenance actions), often abbreviated, and relatively brief (primarily a single sentence). The task is to learn a mapping from a problem description P (*input*) to an action description A (*output*). We treat this as a sequence-to-sequence prediction problem: given the text of a problem, generate the text of the likely maintenance action.

We also prepared two dataset versions: abbreviated and expanded. In the abbreviated version, an entry might read *"#2 & #4 cyl rocker cover gasket are leaking*" which requires domain knowledge to interpret ("cyl" meaning cylinder). The expanded version replaces abbreviations with their full forms, e.g., *"#2 & #4 cylinder rocker cover gasket are leaking*". To construct the expanded version, we used the standardized abbreviation dictionary released with MaintNet, which defines 65 aviation-specific terms. For example, "cyl" was expanded to "cylinder" and "batt" to "battery."

3. EXPERIMENTS

We designed our experiments to fine-tune and evaluate different LLMs to predict the action given a problem. In this section, we explain the different models, experimental setup, and evaluation metrics used.

3.1. Models and Baselines

We evaluated a mix of proprietary and open-source models. The proprietary baseline included GPT-4o, while open-source candidates included instruct versions of Meta's Llama-3.2-3B (Grattafiori et al. 2024), Google's Gemma-3-4B (Team et al. 2025), Microsoft's Phi-3.5-mini and Phi-4-mini (Abouelenin et al. 2025), NVIDIA's Nemotron-Mini-4B (Adler et al. 2024), and Alibaba's Qwen2.5-3B (Team et al. 2024). We selected open-source models of 3-4B parameter

range for fair comparison, balancing availability and fine-tuning feasibility. For each model, we tested two conditions:

- Zero-shot: Models were evaluated on the filtered single-action subset by prompting the model to generate the action given a problem.
- Few-shot (5-shot): Models were evaluated on the filtered single-action subset by prompting the model to generate the action given a problem along with 5 examples of problem-action pairs. This experiment was conducted to examine whether in-context learning could supplement model performance.

In addition, we performed supervised full-parameter fine-tuning on three candidate models: Llama-3.2-3B, Gemma-3-4B, and Phi-4-mini based on the preliminary baseline results and feasibility of fine-tuning. Fine-tuning was performed using subsets of the MaintNet aviation dataset, with ratio-based splits (10%–90%), and models were evaluated with zero-shot prompting on held-out test sets.

3.2. Fine-Tuning Objective

Each problem-action pair was reformatted into a dialogue-like instruction format using special tokens, inspired by Alpaca-style instruction tuning. The fine-tuning objective was causal language modeling (CLM), with the model trained to maximize the likelihood of the target tokens in the action sequence conditioned on the problem. The cross-entropy loss was applied at the token level across the generated output.

The fine-tuning pipeline was implemented using the HuggingFace Transformers framework (Wolf et al. 2020). Each model employed its native tokenizer, with input sequences truncated or padded to a maximum length of 512 tokens. The training data were split into ratio-based subsets (10%–90%), with each subset further divided into a fixed 90/10 train/validation split. Optimization was carried out with the AdamW optimizer, using a learning rate of 5e-6, weight decay of 0.01, and 50 warm-up steps. Training was performed with an effective batch size of 8, achieved through a per-device batch size of 1 and gradient accumulation over 8 steps. All experiments used BF16 precision for computational efficiency. Each model was fine-tuned for three epochs per split, retaining the best-performing model based on validation performance. The compute and runtime details are provided in the Appendix.

3.3. Evaluation Metrics

We used a combination of lexical overlap and semantic similarity metrics to assess prediction quality:

- **ROUGE-1 / ROUGE-2**: Measures n-gram overlap between generated and reference actions.
- **BLEU**: Captures precision of n-gram matches, useful for structured technical text.
- **Cosine Similarity**: Based on sentence embeddings, capturing semantic relatedness.
- **BERTScore**: Uses contextual embeddings from Bidirectional Encoder Representations from Transformers (BERT) to evaluate semantic alignment.

Each score was reported as the mean ± standard deviation across three trials. To reduce prompt drift and standardize outputs at test time, we prepend a fixed domain pre-prompt describing the MaintNet aviation dataset and the expected style of the action statement; the model is then given Input: {*problem*} and asked to complete the Output. Decoding uses stochastic sampling with parameters max_new_tokens=100, do_sample=True, top_p=0.9, and temperature=0.5. The full prompts for zero-shot and few-shot settings are provided in the Appendix.

4. RESULTS

4.1. Evaluation of LLMs on Aviation raw-abbreviation dataset

As presented in Table 1, on the raw-abbreviation version of the dataset, Google's Gemma-3-4B consistently performs the best on ROUGE-based measures, achieving 9.4–43.1% higher ROUGE-1 and 18.3–147.2% higher ROUGE-2 scores compared to the next best and weakest models. At the same time, GPT-4o establishes itself as the strongest performer on semantic alignment benchmarks, obtaining the highest BLEU, cosine similarity, and BERTScore. Specifically, GPT-4o leads by 15.0–144.3% in BLEU, 4.4–14.0% in cosine similarity, and 0.6–2.2% in BERTScore over the next best and weakest models. These results suggest that while Gemma is better at capturing lexical overlap with reference actions, GPT-4o demonstrates greater consistency in producing semantically faithful responses.

4.2. Evaluation of LLMs on expanded dataset

On the expanded dataset, Google's Gemma-3-4B performs the best on lexical overlap metrics, achieving ROUGE-1 of 0.3577 and ROUGE-2 of 0.1400. Compared to other models, Gemma is ahead by 5.5–41.0% on ROUGE-1 and 16.9–133.0% on ROUGE-2, showing that abbreviation expansion marginally amplifies its advantage in capturing precise problem-action correspondences. On the other hand, GPT-4o continues to lead on semantic and mixed measures, recording the highest BLEU (0.0547), cosine similarity (0.5906), and BERTScore (0.8717). Relative to the next-best models, GPT-4o achieves improvements of 1.8–147.5% in BLEU, 4.2–13.5% in cosine similarity, and 0.6–2.1% in BERTScore. These results, presented in Table 2, indicate that while Gemma remains the strongest at reproducing lexical matches with reference actions, GPT-4o leads in semantic alignment, with its margin of improvement widening in some cases once the dataset is normalized and expanded.

Table 1: Evaluation of different LLMs on Aviation raw-abbreviation dataset showing ROUGE-1, ROUGE-2, BLEU, Cosine Similarity and BERTScore metrics.

Model	ROUGE-1	ROUGE-2	BLEU	Cosine Sim.	BERT Score
GPT-4o	0.3226 ± 0.0014	0.1136 ± 0.0001	**0.0513 ± 0.0003**	**0.5866 ± 0.0009**	**0.8710 ± 0.0002**
Qwen2.5-3B	0.2859 ± 0.0004	0.0871 ± 0.0013	0.0290 ± 0.0006	0.5237 ± 0.0009	0.8530 ± 0.0002
Gemma-3-4B	**0.3530 ± 0.0004**	**0.1357 ± 0.0009**	0.0446 ± 0.0001	0.5620 ± 0.0006	0.8655 ± 0.0000
Llama-3.2-3B	0.3156 ± 0.0021	0.1122 ± 0.0014	0.0394 ± 0.0008	0.5346 ± 0.0010	0.8589 ± 0.0004
Phi-3.5-mini	0.2467 ± 0.0006	0.0549 ± 0.0005	0.0210 ± 0.0000	0.5148 ± 0.0018	0.8527 ± 0.0001
Phi-4-mini	0.2900 ± 0.0017	0.0801 ± 0.0018	0.0282 ± 0.0007	0.5182 ± 0.0012	0.8555 ± 0.0002
Nemotron-Mini-4B	0.3192 ± 0.0011	0.1147 ± 0.0011	0.0419 ± 0.0005	0.5320 ± 0.0008	0.8593 ± 0.0002

Table 2: Evaluation of different LLMs on Aviation expanded dataset showing ROUGE-1, ROUGE-2, BLEU, Cosine Similarity and BERTScore metrics.

Model	ROUGE-1	ROUGE-2	BLEU	Cosine Sim.	BERT Score
GPT-4o	0.3295 ± 0.0013	0.1210 ± 0.0007	**0.0547 ± 0.0007**	**0.5906 ± 0.0011**	**0.8717 ± 0.0002**
Qwen2.5-3B	0.2923 ± 0.0015	0.0901 ± 0.0013	0.0297 ± 0.0004	0.5272 ± 0.0007	0.8534 ± 0.0001
Gemma-3-4B	**0.3577 ± 0.0002**	**0.1400 ± 0.0003**	0.0466 ± 0.0001	0.5669 ± 0.0012	0.8662 ± 0.0001
Llama-3.2-3B	0.3189 ± 0.0008	0.1166 ± 0.0014	0.0408 ± 0.0007	0.5382 ± 0.0019	0.8587 ± 0.0001
Phi-3.5-mini	0.2537 ± 0.0005	0.0601 ± 0.0009	0.0221 ± 0.0004	0.5205 ± 0.0008	0.8535 ± 0.0001
Phi-4-mini	0.3020 ± 0.0019	0.0855 ± 0.0003	0.0310 ± 0.0006	0.5256 ± 0.0013	0.8568 ± 0.0003
Nemotron-Mini-4B	0.3252 ± 0.0003	0.1190 ± 0.0009	0.0435 ± 0.0002	0.5377 ± 0.0005	0.8595 ± 0.0001

4.3. Few-shot evaluation of LLMs on Aviation raw-abbreviation dataset

Few-shot prompting noticeably alters model performance, amplifying gains for certain models while modestly helping others, as presented in Table 3. GPT-4o not only retains its position as the top model but also records meaningful gains, especially in ROUGE-2 (+25.3%) and BLEU (+16.6%). Interestingly, weaker models benefit disproportionately: Phi-3.5 nearly doubles its BLEU and ROUGE-2 scores, while Phi-4-mini also more than doubles BLEU and improves ROUGE-2 by nearly 60%. Llama-3.2-3B achieves consistent gains across all metrics, most notably a 28% boost in BLEU. By contrast, Gemma and Nemotron experience small drops in ROUGE, though both improve on semantic measures like cosine similarity and BERTScore. Qwen shows a similar pattern, losing ground in ROUGE while gaining in BLEU and BERTScore. Overall, few-shot prompting emerges as particularly advantageous for underperforming models, helping them narrow the gap with stronger baselines.

Table 3: 5-shot evaluation of different LLMs on Aviation raw-abbreviation dataset

Model	ROUGE-1	ROUGE-2	BLEU	Cosine Sim.	BERT Score
GPT-4o	**0.3534 ± 0.0010**	**0.1423 ± 0.0007**	**0.0598 ± 0.0010**	**0.5949 ± 0.0008**	**0.8765 ± 0.0001**
Qwen2.5-3B	0.2737 ± 0.0026	0.0836 ± 0.0021	0.0354 ± 0.0010	0.5257 ± 0.0006	0.8668 ± 0.0002
Gemma-3-4B	0.3385 ± 0.0014	0.1304 ± 0.0015	0.0530 ± 0.0005	0.5777 ± 0.0007	0.8760 ± 0.0001
Llama-3.2-3B	0.3224 ± 0.0006	0.1184 ± 0.0008	0.0504 ± 0.0005	0.5577 ± 0.0006	0.8741 ± 0.0001
Phi-3.5-mini	0.3280 ± 0.0006	0.1032 ± 0.0006	0.0425 ± 0.0008	0.5711 ± 0.0005	0.8744 ± 0.0001
Phi-4-mini	0.3454 ± 0.0004	0.1279 ± 0.0004	0.0571 ± 0.0009	0.5711 ± 0.0014	0.8758 ± 0.0003
Nemotron-Mini-4B	0.2943 ± 0.0033	0.1105 ± 0.0021	0.0458 ± 0.0019	0.5465 ± 0.0013	0.8721 ± 0.0004

4.4. Few-shot evaluation of LLMs on expanded dataset

On the expanded dataset under 5-shot prompting, GPT-4o once again emerges as the overall best-performing model, achieving the highest scores across ROUGE-1, ROUGE-2, BLEU, and cosine similarity, while maintaining essential parity with Gemma on BERTScore, as shown in Table 4. Specifically, GPT-4o attains a ROUGE-1 of 0.3559, outperforming Gemma by 3.8% and the weakest baseline (Qwen) by 29.1%. On ROUGE-2, GPT-4o records 0.1455,

which is 8.1% stronger than Gemma and nearly 74.0% higher than Qwen. For BLEU, GPT-4o's score of 0.0615 exceeds the second-best (Phi-4-mini) by 10.2% and Qwen by 75.7%. Its cosine similarity of 0.5996 is 3.1% higher than Gemma and 14.0% higher than Qwen, underscoring its ability to generate semantically aligned actions.

Table 4: 5-shot evaluation of different LLMs on Aviation expanded dataset

Model	ROUGE-1	ROUGE-2	BLEU	Cosine Sim.	BERT Score
GPT-4o	**0.3559 ± 0.0011**	**0.1455 ± 0.0004**	**0.0615 ± 0.0004**	**0.5996 ± 0.0012**	0.8763 ± 0.0002
Qwen2.5-3B	0.2754 ± 0.0015	0.0837 ± 0.0016	0.0350 ± 0.0010	0.5259 ± 0.0011	0.8669 ± 0.0002
Gemma-3-4B	0.3429 ± 0.0002	0.1346 ± 0.0004	0.0548 ± 0.0003	0.5814 ± 0.0008	**0.8765 ± 0.0001**
Llama-3.2-3B	0.3261 ± 0.0028	0.1197 ± 0.0022	0.0507 ± 0.0010	0.5636 ± 0.0016	0.8740 ± 0.0004
Phi-3.5-mini	0.3267 ± 0.0014	0.1007 ± 0.0009	0.0422 ± 0.0004	0.5722 ± 0.0008	0.8741 ± 0.0001
Phi-4-mini	0.3454 ± 0.0007	0.1268 ± 0.0012	0.0558 ± 0.0013	0.5744 ± 0.0001	0.8755 ± 0.0002
Nemotron-Mini-4B	0.2973 ± 0.0019	0.1115 ± 0.0020	0.0459 ± 0.0005	0.5507 ± 0.0012	0.8720 ± 0.0003

Interestingly, Gemma-3-4B performs very competitively in BERTScore, achieving 0.8765, which is marginally higher than GPT-4o's 0.8763 (by 0.02%), while outperforming the weakest baseline (Nemotron) by 0.5%. These results suggest that on the expanded dataset, GPT-4o leads most dimensions of evaluation, but Gemma demonstrates resilience in maintaining lexical-semantic fidelity as captured by BERTScore. Overall, these results indicate that few-shot prompting on the expanded dataset offers broader and more consistent performance benefits than its raw-abbreviation counterpart, though the magnitude of gains varies across models, with some (e.g., GPT-4o and Gemma) improving more substantially than others.

4.5. Evaluation results after Fine-tuning LLMs

4.5.1. Gemma-3-4B model evaluation results

The evaluation of Google's Gemma-3-4B model before and after supervised fine-tuning on different splits of the MaintNet aviation dataset is shown in Table 5. The splits range from 10% to 90% of the available training data, with the remaining portion reserved for evaluation. This design allows us to assess how model performance scales with increasing amounts of fine-tuning data and to compare improvements over the zero-shot baseline.

Across all splits, fine-tuning leads to consistent and substantial gains in nearly every evaluation metric, as shown in Figure 2. ROUGE-1 improves by 10–14%, with the largest increase observed at the 0.6 split (+13.77%). Similarly, ROUGE-2 shows remarkable improvements, ranging from 15% to over 31%, indicating that fine-tuning greatly enhances the model's ability to capture bigram-level overlaps between generated and reference actions. BLEU scores benefit the most, with improvements ranging from +28% (at 0.1 split) up to a striking +88.8% at 0.9 split, demonstrating that fine-tuning dramatically strengthens the model's precision in reproducing exact phrasing of corrective actions.

In addition to lexical overlap, fine-tuning also improves semantic alignment. Cosine similarity increases steadily, with relative gains between +4.4% and +9.7%, showing that the model produces predictions more semantically consistent with references. Likewise, BERTScore sees incremental gains of +0.8% to +2.3%, confirming that fine-tuning preserves and even slightly enhances lexical-semantic fidelity. These smaller but steady increases suggest that while Gemma already had strong semantic representations, fine-tuning sharpened its alignment to domain-specific action phrasing.

The magnitude of improvement correlates with the size of the fine-tuning split. For example, at lower data fractions (0.1–0.3), improvements are noticeable but moderate, whereas at higher splits (0.6–0.9), gains become dramatic; especially for BLEU and ROUGE-2. This trend highlights that Gemma continues to benefit from additional training data and does not plateau early. The large BLEU improvements at 0.8 and 0.9 suggest that exposure to more diverse corrective actions enables the model to better reproduce specific maintenance terminology and phrasing.

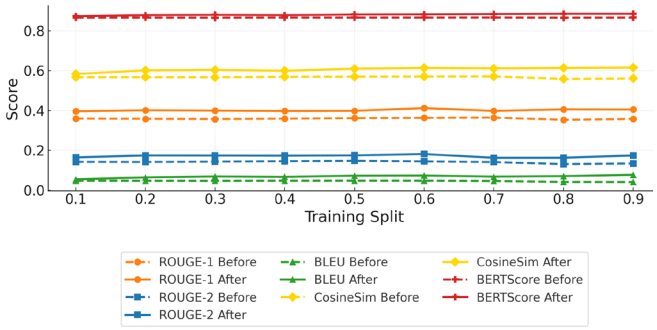

Figure 2: Visualization of Gemma-3-4B model evaluation results before and after fine-tuning

Table 5: Gemma-3-4B model evaluation results showing baseline (before) and after fine-tuning

Split	ROUGE-1 (Mean ± Std)	ROUGE-1 Var	Δ%	ROUGE-2 (Mean ± Std)	ROUGE-2 Var	Δ%	BLEU (Mean ± Std)	BLEU Var	Δ%	CosineSim (Mean ± Std)	CosineSim Var	Δ%	BERTScore (Mean ± Std)	BERTScore Var	Δ%
Baseline	*0.3573 ± 0.2586*	*0.0669*	-	*0.1339 ± 0.2178*	*0.0474*	-	*0.0411 ± 0.0585*	*0.0034*	-	*0.5609 ± 0.2093*	*0.0438*	-	*0.8665 ± 0.0344*	*0.0012*	-
0.1	0.3947 ± 0.2791	0.0779	+10.47%	0.1551 ± 0.2662	0.0709	+15.79%	0.0526 ± 0.0798	0.0064	+27.98%	0.5857 ± 0.2148	0.0461	+4.43%	0.8739 ± 0.0368	0.0014	+0.85%
0.2	0.4003 ± 0.2874	0.0826	+12.03%	0.1669 ± 0.2821	0.0796	+24.66%	0.0681 ± 0.1264	0.0160	+65.69%	0.5974 ± 0.2277	0.0518	+6.51%	0.8796 ± 0.0417	0.0017	+1.51%
0.3	0.4018 ± 0.2849	0.0812	+12.47%	0.1679 ± 0.2795	0.0781	+25.40%	0.0693 ± 0.1270	0.0161	+68.38%	0.6024 ± 0.2294	0.0526	+7.41%	0.8819 ± 0.0427	0.0018	+1.77%
0.4	0.3932 ± 0.2910	0.0847	+10.06%	0.1734 ± 0.2841	0.0807	+29.53%	0.0671 ± 0.1104	0.0122	+63.26%	0.5945 ± 0.2294	0.0526	+6.00%	0.8787 ± 0.0426	0.0018	+1.41%
0.5	0.3978 ± 0.2894	0.0838	+11.34%	0.1699 ± 0.2833	0.0803	+26.88%	0.0721 ± 0.1309	0.0171	+75.55%	0.6043 ± 0.2304	0.0531	+7.74%	0.8833 ± 0.0430	0.0019	+1.94%
0.6	0.4065 ± 0.2947	0.0868	+13.77%	0.1755 ± 0.2935	0.0862	+31.09%	0.0764 ± 0.1492	0.0223	+85.88%	0.6067 ± 0.2347	0.0551	+8.16%	0.8833 ± 0.0442	0.0020	+1.94%
0.7	0.3959 ± 0.2902	0.0842	+10.80%	0.1708 ± 0.2858	0.0817	+27.61%	0.0702 ± 0.1291	0.0167	+70.80%	0.6093 ± 0.2345	0.0550	+8.63%	0.8857 ± 0.0422	0.0018	+2.21%
0.8	0.4044 ± 0.2908	0.0846	+13.18%	0.1744 ± 0.2876	0.0827	+30.24%	0.0754 ± 0.1480	0.0219	+83.70%	0.6127 ± 0.2338	0.0547	+9.24%	0.8865 ± 0.0433	0.0019	+2.31%
0.9	0.4049 ± 0.2823	0.0797	+13.31%	0.1744 ± 0.2810	0.0790	+30.24%	0.0776 ± 0.1402	0.0197	+88.81%	0.6155 ± 0.2244	0.0503	+9.73%	0.8856 ± 0.0425	0.0018	+2.20%

4.5.2. Llama-3.2-3B model evaluation results

On the Llama-3.2-3B model (Appendix Table 7), fine-tuning on progressively larger splits of the dataset produces consistent and measurable gains across all evaluation metrics. ROUGE-1 shows steady improvements in the range of 9–14%, while ROUGE-2 benefits even more strongly, with increases of 19–30%, underscoring the model's enhanced ability to capture overlapping n-grams from the ground-truth actions. BLEU sees some of the most pronounced jumps, improving by 26–47% depending on the training split, reflecting a sharper alignment in surface-level phrasing between generated and reference actions.

Semantic similarity metrics also improve consistently: cosine similarity rises by approximately 4.8–7.3% across all splits, while BERTScore exhibits modest but reliable gains of about 0.8–1.2%. Notably, larger training splits (0.7–0.9) tend to yield the greatest relative improvements in BLEU and ROUGE-2, suggesting that the model particularly benefits from richer supervision when capturing fine-grained lexical patterns. These results indicate that fine-tuning substantially improves both lexical overlap and semantic alignment, though the magnitude of improvement is most striking for BLEU and ROUGE-2.

Table 6: Best-performing models across all evaluation settings

Setting	Best Model	ROUGE-1	ROUGE-2	BLEU	Cosine Sim.	BERTScore
Zero-shot (abbr.)	Gemma-3-4B	**0.3530**	**0.1357**	–	–	–
Zero-shot (abbr.)	GPT-4o	–	–	**0.0513**	**0.5866**	**0.8710**
Zero-shot (expanded)	Gemma-3-4B	**0.3577**	**0.1400**	–	–	–
Zero-shot (expanded)	GPT-4o	–	–	**0.0547**	**0.5906**	**0.8717**
Few-shot (abbr.)	GPT-4o	**0.3534**	**0.1423**	**0.0598**	**0.5949**	**0.8765**
Few-shot (expanded)	GPT-4o	**0.3559**	**0.1455**	**0.0615**	**0.5996**	0.8763
Few-shot (expanded)	Gemma-3-4B	–	–	–	–	**0.8765**
Fine-tuned (best)	Gemma-3-4B (0.6)	**0.4065**	**0.1755**	–	–	–
Fine-tuned (best)	Gemma-3-4B (0.9)	–	–	**0.0776**	**0.6155**	–
Fine-tuned (best)	Gemma-3-4B (0.8)	–	–	–	–	**0.8865**

4.5.3. Phi-4-mini model evaluation results

On the Phi-4-mini model (Appendix Table 8), fine-tuning leads to clear and steady improvements across all data splits. ROUGE-1 consistently rises by about 7–10%, while ROUGE-2 shows larger gains in the range of 18–25%, indicating that the model becomes more capable of reproducing detailed n-gram overlaps from the reference actions. BLEU benefits the most dramatically, with increases between 26% and over 52%, underscoring Phi-4-mini's stronger alignment in surface-level phrasing after fine-tuning.

Semantic metrics also improve, though more moderately: cosine similarity gains remain in the +1.5–3.9% range, and BERTScore increases are smaller but steady at roughly +0.3–0.6%. Whether trained on 10% or 90% of the dataset, the model's performance curves upward in a stable fashion without large fluctuations. This suggests that Phi-4-mini benefits uniformly from exposure to additional data, but its largest relative improvements are concentrated in BLEU and ROUGE-2, while semantic similarity metrics rise more conservatively.

5. DISCUSSION

Normalization vs. semantics: Moving from the raw-abbreviation corpus (Table 1) to the expanded version (Table 2) consistently boosts lexical overlap for the best ROUGE model (Gemma-3-4B), while GPT-4o remains strongest on semantic/mixed measures (BLEU, cosine similarity, BERTScore). This split suggests two complementary capabilities: abbreviation expansion helps models reproduce the exact surface form of corrective actions, whereas semantic fidelity is less sensitive to token normalization and benefits from larger, more capable models.

In-context learning vs. fine-tuning: 5-shot prompting changes the results (Tables 3–4), helping most models and disproportionately lifting weaker open models (e.g., Phi-3.5-mini/Phi-4-mini, Llama-3.2-3B), while GPT-4o and Gemma-3-4B retain leadership on their respective metrics. The largest relative few-shot gains appear in ROUGE-2/BLEU, indicating better adoption of domain phrasing through in-context learning. Supervised fine-tuning delivers the most durable gains (Tables 5, 7, and 8) across Gemma-3-4B, Llama-3.2-3B, and Phi-4-mini, ROUGE-2 typically rises ~18–31%, BLEU jumps ~25–90% depending on split, and cosine similarity/BERTScore improve steadily. Gains scale with data, i.e., larger training data splits (0.7–0.9) yield the steepest improvements, while variance across trials remains moderate and does not obscure the overall upward trend. Table 6 summarizes the best-performing models across all evaluation settings with best splits per metric for fine-tuned models. An example provided in Appendix Table 9 illustrates how fine-tuning consistently improves alignment between generated actions and the ground-truth reference.

Cross-domain application: Beyond aviation, the proposed framework will be extended to additional maintenance datasets, including automotive and facility management records. Planned datasets include Avi-Acc, and Avi-Safe (problem, action, ATA code, flight or safety details), Auto-Main, Auto-Acc, and Auto-Safe (problem, action, reason, department, and accident or request reports), and Faci-Main (problem, action, type, and location) (Akhbardeh et al. 2020). These domains share similar problem–action text structures, enabling direct application of the same pipeline for cross-domain evaluation.

6. CONCLUSION

This study presented a systematic evaluation of both open-source and proprietary large language models for predicting maintenance actions from aviation problem logs. We benchmarked off-the-shelf models in zero-shot and few-shot settings and further explored the impact of supervised fine-tuning.

Our results show that few-shot prompting provides noticeable improvements across nearly all models, helping weaker baselines close the gap with stronger ones. Fine-tuning was even more effective, consistently boosting performance across different data splits. Notably, the fine-tuned Gemma-3-4B model outperformed all other candidates, including GPT-4o, underscoring the value of domain-specific adaptation for this task.

Future work will focus on scaling fine-tuning to larger open-source models and extending prediction to structured problem–component–action triples. This direction will allow deterministic action recommendations tied to specific fault categories, further enhancing the reliability and practical utility of LLMs in predictive maintenance. Moreover, incorporating technician feedback in a human-in-the-loop setting will help ensure that generated actions remain safe, interpretable, and aligned with real maintenance practices.

REFERENCES

Tanguy, L., Tulechki, N., Urieli, A., Hermann, E., & Raynal, C. (2016). Natural language processing for aviation safety reports: From classification to interactive analysis. Computers in Industry, 78, 80-95. https://doi.org/10.1016/j.compind.2015.09.005

Altuncu, M. T., Mayer, E., Yaliraki, S. N., & Barahona, M. (2018). From text to topics in healthcare records: An unsupervised graph partitioning methodology. arXiv preprint arXiv:1807.02599. https://arxiv.org/abs/1807.02599

Jarry, G., Delahaye, D., Nicol, F., & Feron, E. (2020). Aircraft atypical approach detection using functional principal component analysis. Journal of Air Transport Management, 84, 101787. https://doi.org/10.1016/j.jairtraman.2020.101787

Akhbardeh, F., Desell, T., & Zampieri, M. (2020). MaintNet: A collaborative open-source library for predictive maintenance language resources. arXiv preprint arXiv:2005.12443. https://arxiv.org/abs/2005.12443

Akhbardeh, F., Desell, T., & Zampieri, M. (2020, December). NLP tools for predictive maintenance records in MaintNet. In Proceedings of the 1st conference of the asia-pacific chapter of the association for computational linguistics and the 10th international joint conference on natural language processing: System demonstrations (pp. 26-32). https://aclanthology.org/2020.aacl-demo.5/

Akhbardeh, F., Alm, C. O., Zampieri, M., & Desell, T. (2021, August). Handling extreme class imbalance in technical logbook datasets. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (pp. 4034-4045). https://doi.org/10.18653/v1/2021.acl-long.312.

Payette, M., Abdul-Nour, G., Meango, T. J. M., Diago, M., & Côté, A. (2025). Leveraging failure modes and effect analysis for technical language processing. Machine Learning and Knowledge Extraction, 7(2), 42. https://doi.org/10.3390/make7020042.

Sundaram, S., & Zeid, A. (2025). Technical language processing for Prognostics and Health Management: applying text similarity and topic modeling to maintenance work orders. Journal of Intelligent Manufacturing, 36(3), 1637-1657. https://doi.org/10.1007/s10845-024-02323-4.

Akhbardeh, F., Zampieri, M., Alm, C. O., & Desell, T. (2022, June). Transfer learning methods for domain adaptation in technical logbook datasets. In Proceedings of the Thirteenth Language Resources and Evaluation Conference (pp. 4235-4244). https://aclanthology.org/2022.lrec-1.450.

Brundage, M. P., Sexton, T., Hodkiewicz, M., Dima, A., & Lukens, S. (2021). Technical language processing: Unlocking maintenance knowledge. Manufacturing Letters, 27, 42-46. https://doi.org/10.1016/j.mfglet.2020.11.001

Kelm, B., Haas, P. H., Jochum, S., Margies, L., & Müller, R. (2025). Enhancing Assembly Instruction Generation for Cognitive Assistance Systems with Large Language Models. Procedia CIRP, 134, 7-12. https://doi.org/10.1016/j.procir.2025.03.010

Meunier-Pion, J. (2024, June). Natural Language Processing for Risk, Resilience, and Reliability. In PHM Society European Conference (Vol. 8, No. 1, p. 4). https://doi.org/10.36001/phme.2024.v8i1.3956

Naqvi, S. M. R., Varnier, C., Nicod, J. M., Zerhouni, N., & Ghufran, M. (2021, December). Leveraging free-form text in maintenance logs through BERT transfer learning. In International Conference on Deep Learning, Artificial Intelligence and Robotics (pp. 63-75). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-98531-8_7.

Naqvi, S. M. R., Ghufran, M., Varnier, C., Nicod, J. M., Javed, K., & Zerhouni, N. (2024). Unlocking maintenance insights in industrial text through semantic search. Computers in Industry, 157, 104083. https://doi.org/10.1016/j.compind.2024.104083.

Vidyaratne, L., Lee, X. Y., Kumar, A., Watanabe, T., Farahat, A., & Gupta, C. (2024, June). Generating troubleshooting trees for industrial equipment using large language models (LLM). In 2024 IEEE International Conference on Prognostics and Health Management (ICPHM) (pp. 116-125). IEEE. https://doi.org/10.1109/ICPHM61352.2024.10626823.

Lukens, S., McCabe, L. H., Gen, J., & Ali, A. (2024, November). Large Language Model Agents as Prognostics and Health Management Copilots. In Annual Conference of the PHM Society (Vol. 16, No. 1). https://doi.org/10.36001/phmconf.2024.v16i1.3906.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020, October). Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (pp. 38-45). https://doi.org/10.18653/v1/2020.emnlp-demos.6.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., ... & Vasic, P. (2024). The Llama 3 herd of models. arXiv preprint arXiv:2407.21783. https://arxiv.org/abs/2407.21783.

Team, G., Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., ... & Iqbal, S. (2025). Gemma 3 technical report. arXiv preprint arXiv:2503.19786. https://arxiv.org/abs/2503.19786

Abouelenin, A., Ashfaq, A., Atkinson, A., Awadalla, H., Bach, N., Bao, J., ... & Zhou, X. (2025). Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-Loras. arXiv preprint arXiv:2503.01743. https://arxiv.org/abs/2503.01743.

Adler, B., Agarwal, N., Aithal, A., Anh, D. H., Bhattacharya, P., Brundyn, A., ... & Zhu, C. (2024). Nemotron-4 340B technical report. arXiv preprint arXiv:2406.11704. https://arxiv.org/abs/2406.11704.

Team, Q. (2024). Qwen2 technical report. arXiv preprint arXiv:2407.10671. https://arxiv.org/abs/2407.10671

APPENDIX

Compute and runtime details: All experiments were conducted on a single RTX 6000 Ada (48 GB) GPU with bfloat16 precision. Fine-tuning used three epochs per split, and the times reported here are for full training and per trial for evaluation, with peak VRAM in parentheses. Phi-4-Mini: training 30 seconds to 5.25 minutes (42 GB); evaluation 2.25 to 15 minutes (18 GB). Gemma-3-4B: training 50 seconds to 7 minutes (47 GB); evaluation 2 to 30 minutes (19.5 GB). Llama-3.2-3B: training 30 seconds to 4 minutes (34 GB); evaluation 8 to 90 minutes (15 GB). These figures indicate that both full-parameters fine-tuning and evaluation of compact 3–4B models are practical on a single workstation-class GPU.

Example of zero-shot prompt: You are an expert in Aviation maintenance. The aviation maintenance dataset within MaintNet originates from the University of North Dakota's Aviation Program and comprises 6,169 anonymized entries. The dataset includes unstructured text entries detailing maintenance issues, often written in domain-specific jargon, abbreviations, and non-standard grammar. Each entry typically includes a 'Problem' field describing the maintenance issue and an 'Action' field detailing the corrective measures taken. You will be provided with a problem, and your task is to generate a corresponding action statement. The action statement should be concise, clear, and directly related to the problem statement. Please ensure that the generated action is relevant and appropriate for the given problem

Example of 5-shot prompt: You are an expert in Aviation maintenance. The aviation maintenance dataset within MaintNet originates from the University of North Dakota's Aviation Program and comprises 6,169 anonymized entries. The dataset includes unstructured text entries detailing maintenance issues, often written in domain-specific jargon, abbreviations, and non-standard grammar. Each entry typically includes a 'Problem' field describing the maintenance issue and an 'Action' field detailing the corrective measures taken. You will be provided with a problem, and your task is to generate a corresponding action statement. The action statement should be concise, clear, and directly related to the problem statement. Please ensure that the generated action is relevant and appropriate for the given problem. Some examples have been provided for your reference.

Input: TOOL LEFT ON CYLINDER #2

Output: REMOVED TOOL FROM CYL #2.

Input: TYRAP AND SCREWDRIVER FOUND NEAR ENGINE TOE

Output: REMOVED TYRAP AND TOOL FROM ENGINE AREA.

Input: NO COMPRESSION ON CYL #4 DUE TO VALVE LEAK

Output: INSTALLED NEW CYLINDER AND PISTON ON #4.

Input: ZIP TIES FOUND ON ENGINE MOUNTING BRACKETS

Output: REMOVED ZIP TIES AND SECURED HARNESS WITH CLIPS.

Input: LACING CORD REPLACED WITH TIES ON BOTH MOUNTS

Output: SECURED LINES WITH LACING CORD ON BOTH SIDES.

Input: [TEST PROBLEM]

Output:

Table 7: Llama-3.2-3B model evaluation results showing baseline (before) and after fine-tuning

Split	ROUGE-1 (Mean ± Std)	ROUGE-1 Var	Δ%	ROUGE-2 (Mean ± Std)	ROUGE-2 Var	Δ%	BLEU (Mean ± Std)	BLEU Var	Δ%	CosineSim (Mean ± Std)	CosineSim Var	Δ%	BERTScore (Mean ± Std)	BERT Score Var	Δ%
Baseline	*0.3213 ± 0.2360*	*0.0557*	–	*0.1216 ± 0.1913*	*0.0366*	–	*0.0388 ± 0.0669*	*0.0045*	–	*0.5362 ± 0.2065*	*0.0427*	–	*0.8580 ± 0.0342*	*0.0012*	–
0.1	0.3604 ± 0.2569	0.0660	+12.18%	0.1502 ± 0.2299	0.0529	+23.53%	0.0522 ± 0.1028	0.0106	+34.54%	0.5642 ± 0.2175	0.0473	+5.23%	0.8660 ± 0.0366	0.0013	+0.93%
0.2	0.3567 ± 0.2562	0.0656	+11.03%	0.1542 ± 0.2298	0.0528	+26.77%	0.0511 ± 0.0939	0.0088	+31.70%	0.5648 ± 0.2120	0.0449	+5.33%	0.8657 ± 0.0364	0.0013	+0.89%
0.3	0.3629 ± 0.2567	0.0659	+12.95%	0.1560 ± 0.2275	0.0518	+28.26%	0.0552 ± 0.0982	0.0096	+42.27%	0.5714 ± 0.2161	0.0467	+6.56%	0.8669 ± 0.0375	0.0014	+1.04%
0.4	0.3506 ± 0.2496	0.0623	+9.12%	0.1446 ± 0.2106	0.0444	+18.91%	0.0490 ± 0.0887	0.0079	+26.29%	0.5621 ± 0.2132	0.0455	+4.83%	0.8651 ± 0.0370	0.0014	+0.83%
0.5	0.3623 ± 0.2504	0.0627	+12.76%	0.1545 ± 0.2255	0.0508	+26.99%	0.0554 ± 0.0966	0.0093	+42.78%	0.5686 ± 0.2132	0.0455	+6.04%	0.8670 ± 0.0373	0.0014	+1.05%
0.6	0.3564 ± 0.2544	0.0647	+10.93%	0.1518 ± 0.2235	0.0500	+24.81%	0.0511 ± 0.0878	0.0077	+31.70%	0.5670 ± 0.2149	0.0462	+5.74%	0.8664 ± 0.0375	0.0014	+0.99%
0.7	0.3654 ± 0.2570	0.0660	+13.72%	0.1585 ± 0.2309	0.0533	+30.34%	0.0562 ± 0.0964	0.0093	+44.85%	0.5750 ± 0.2169	0.0470	+7.22%	0.8682 ± 0.0388	0.0015	+1.19%
0.8	0.3656 ± 0.2478	0.0614	+13.79%	0.1508 ± 0.2207	0.0487	+24.01%	0.0547 ± 0.0919	0.0084	+40.98%	0.5755 ± 0.2140	0.0458	+7.33%	0.8676 ± 0.0380	0.0014	+1.12%
0.9	0.3661 ± 0.2534	0.0642	+13.94%	0.1559 ± 0.2262	0.0512	+28.19%	0.0569 ± 0.1009	0.0102	+46.65%	0.5737 ± 0.2102	0.0442	+7.00%	0.8681 ± 0.0386	0.0015	+1.18%

Table 8: Phi-4-mini model evaluation results showing baseline (before) and after fine-tuning

Split	ROUGE-1 (Mean ± Std)	ROUGE-1 Var	Δ%	ROUGE-2 (Mean ± Std)	ROUGE-2 Var	Δ%	BLEU (Mean ± Std)	BLEU Var	Δ%	CosineSim (Mean ± Std)	CosineSim Var	Δ%	BERTScore (Mean ± Std)	BERTScore Var	Δ%
Baseline	0.3028 ± 0.2054	0.0422	–	0.0903 ± 0.1412	0.0199	–	0.0279 ± 0.0430	0.0018	–	0.5247 ± 0.1978	0.0391	–	0.8570 ± 0.0317	0.0010	–
0.1	0.3248 ± 0.2239	0.0501	+7.26%	0.1064 ± 0.1637	0.0268	+17.83%	0.0353 ± 0.0602	0.0036	+26.52%	0.5325 ± 0.2016	0.0406	+1.49%	0.8597 ± 0.0320	0.0010	+0.32%
0.2	0.3279 ± 0.2239	0.0501	+8.29%	0.1089 ± 0.1706	0.0291	+20.59%	0.0402 ± 0.0815	0.0066	+44.09%	0.5345 ± 0.2009	0.0404	+1.87%	0.8612 ± 0.0326	0.0011	+0.49%
0.3	0.3282 ± 0.2221	0.0493	+8.38%	0.1096 ± 0.1692	0.0286	+21.35%	0.0397 ± 0.0783	0.0061	+42.30%	0.5366 ± 0.2013	0.0405	+2.27%	0.8613 ± 0.0322	0.0010	+0.50%
0.4	0.3328 ± 0.2230	0.0497	+9.91%	0.1103 ± 0.1726	0.0298	+22.12%	0.0404 ± 0.0768	0.0059	+44.80%	0.5389 ± 0.2045	0.0418	+2.71%	0.8615 ± 0.0328	0.0011	+0.52%
0.5	0.3252 ± 0.2181	0.0475	+7.39%	0.1090 ± 0.1685	0.0284	+20.69%	0.0374 ± 0.0721	0.0052	+34.05%	0.5388 ± 0.1986	0.0395	+2.69%	0.8610 ± 0.0313	0.0010	+0.47%
0.6	0.3283 ± 0.2205	0.0486	+8.42%	0.1074 ± 0.1668	0.0278	+18.91%	0.0383 ± 0.0726	0.0053	+37.27%	0.5364 ± 0.2010	0.0404	+2.23%	0.8611 ± 0.0324	0.0010	+0.48%
0.7	0.3299 ± 0.2217	0.0492	+8.95%	0.1102 ± 0.1651	0.0273	+21.99%	0.0384 ± 0.0714	0.0051	+37.63%	0.5406 ± 0.2017	0.0407	+3.03%	0.8621 ± 0.0324	0.0010	+0.60%
0.8	0.3332 ± 0.2242	0.0503	+10.03%	0.1129 ± 0.1742	0.0303	+25.07%	0.0408 ± 0.0783	0.0061	+46.24%	0.5427 ± 0.2007	0.0403	+3.43%	0.8616 ± 0.0327	0.0011	+0.54%
0.9	0.3328 ± 0.2229	0.0497	+9.91%	0.1132 ± 0.1767	0.0312	+25.39%	0.0424 ± 0.0831	0.0069	+52.00%	0.5450 ± 0.2038	0.0415	+3.86%	0.8617 ± 0.0325	0.0011	+0.55%

Table 9: Model predictions for the maintenance problem "#1 CYLINDER ROCKER ARM & PUSH ROD GALLED ON EXHAUST SIDE." with ground truth "REMOVED & REPLACED ROCKER ARM & PUSH ROD." showing CosSim, ROUGE, BLEU, and BERTScore-F1 for models before and after finetuning.

Model	Prediction	CosSim ↑	ROUGE-1 ↑	ROUGE-2 ↑	BLEU ↑	BERTScore-F1 ↑
GPT-4o	REPLACED #1 CYLINDER EXHAUST ROCKER ARM AND PUSH ROD.	0.7430	0.6667	0.3077	0.0945	0.9030
Gemma-3-4B (Orig.)	# REPLACE CYLINDER ROCKER ARM AND PUSH ROD.	0.7753	0.7692	0.3636	0.1007	0.9229
Gemma-3-4B (Fine-tuned)	# REPLACED CYLINDER ROCKER ARM & PUSH ROD.	0.8006	0.8333	0.6000	0.5411	0.9420
Llama-3.2-3B (Orig.)	#1 CYLINDER ROCKER ARM & PUSH ROD GALLED ON EXHAUST SIDE. REPLACE WITH NEW PART.	0.6682	0.5000	0.3333	0.1645	0.8947
Llama-3.2-3B (Fine-tuned)	REPLACE CYLINDER ROCKER ARM & PUSH ROD.	0.8376	0.8333	0.6000	0.5329	0.9029
Phi-4-mini (Orig.)	REPLACE THE DAMAGED CYLINDER ROCKER ARM AND PUSH ROD ON THE EXHAUST SIDE.	0.6138	0.5263	0.2353	0.0490	0.8761
Phi-4-mini (Fine-tuned)	REPLACE THE CYLINDER ROCKER ARM AND PUSH ROD ON THE EXHAUST SIDE.	0.5482	0.5556	0.2500	0.0528	0.8799

Fine tuning analysis: Figure 3 shows percentage of test cases where the fine-tuned models outperformed, matched, or underperformed the original baselines across different training splits (0.1–0.9). Each bar shows the proportion of problems with higher (blue), equal (orange), or lower (green) scores for each evaluation metric i.e., cosine similarity, ROUGE-1, ROUGE-2, BLEU, and BERTScore-F1. Overall, fine-tuning consistently improved lexical and semantic alignment with the reference actions, particularly in BERTScore and Cosine metrics.

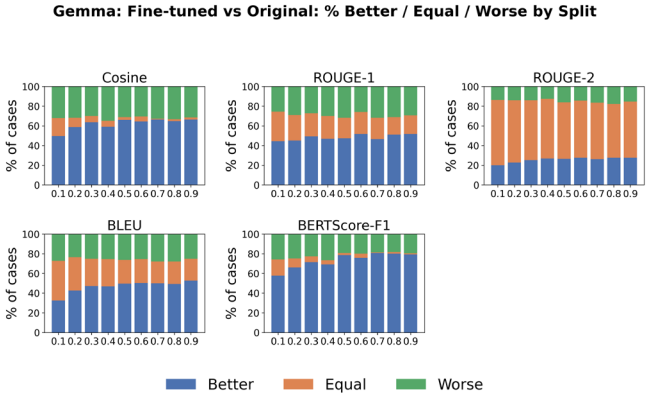

Figure 3: Comparison of fine-tuned vs original models across splits, showing percentage of cases where fine-tuning improved, matched or reduced performance across all evaluation metrics for Gemma model