ACE – Automating Causal Extraction: Leveraging Large Language Models for Bowtie Diagram Generation in Failure Analysis

Priyank Venkatesh[1, 2], Jules Oudmans[2], Florian Zurfluh[2]

[1]*Computational Science Lab, Informatics Institute, University of Amsterdam, Amsterdam, Netherlands*

[2]*UReason, Rotterdam, Netherlands*

pvenkatesh@ureason.com, joudmans@ureason.com, fzurfluh@ureason.com

ABSTRACT

This paper investigates whether open-source, instruction-tuned large language models (LLMs) can automate the generation of Bowtie diagrams from Failure Mode and Effects Analysis (FMEA) documentation. Three pipelines are developed: Retrieval-Augmented Generation (RAG), Optical Character Recognition (OCR) based extraction, and a vision-enabled dual-LLM approach. Each is designed to handle both structured FMEA tables and unstructured narrative text. Three models (Mistral, Qwen-2.5, and LLaMA-3) are evaluated using Sobol sensitivity analysis, stochasticity experiments, and expert Likert scoring on narrative outputs. With strict schema-constrained prompts, models frequently achieve Node and Edge F1 scores above 0.8 on tabular data. Outputs were identical across repeated runs under fixed settings. Sobol analysis shows that prompt strictness and prompt type are the dominant drivers of Bowtie quality, whereas decoding parameters have a negligible effect. On unstructured narrative text, all models struggled, producing hallucinated nodes, incorrect role assignments, and diagrams that deviated from expert references. The results establish a working approach for automating Bowtie generation from FMEA tables and identify the specific obstacles to extending this to narrative sources.

Keywords: Bowtie Diagrams, Large Language Models, FMEA, Prompt Engineering, RAG, Failure Analysis

1. INTRODUCTION

As engineering systems grow more complex, identifying and communicating failure risks has become harder. In industries such as chemical processing, manufacturing, aerospace, and healthcare, minor malfunctions can propagate across interconnected systems, making it difficult for reliability engineers to anticipate risks and communicate consequences. Tools such as FMEA and FMECA provide a systematic framework for identifying failure points, but their tabular outputs do not easily convey how failures escalate (Segismundo & Miguel, 2008; Sharma & Srivastava, 2018).

Visual tools such as Causal Loop Diagrams (CLDs) and Bowtie diagrams address this by making causal relationships explicit. CLDs capture system-level feedback dynamics; Bowtie diagrams map the specific causes, barriers, and consequences surrounding a single critical event (Turner, Hamilton, and Ramsden, 2017). Both are human resource-intensive to construct manually, often requiring multi-hour expert workshops, which limits their scalability across large industrial asset fleets.

LLMs offer a route to automating this process. They have demonstrated the ability to process structured and unstructured technical content and extract structured representations (Li et al., 2024). Prior work has applied LLMs to cause-effect extraction and CLD generation (Hassani et al., 2024; Liu & Keith, 2024; J. Yang et al., 2022), but the direct generation of Bowtie diagrams from FMEA documentation remains an open area of research. LLMs also introduce specific risks: hallucination, format non-compliance on inputs, misinterpretation of domain-specific terminology, and challenges in parsing and integrating information from multi-format or lengthy FMEA documents.

To address these challenges, this work aims to reduce reliance on manual expertise, improve the efficiency and consistency of failure analysis, and support more scalable risk modelling in complex technical fields. This paper addresses two research questions using an experimental framework built on three open-source instruction-tuned LLMs and a dataset drawn from the US Navy Reliability Handbook (NSWC-11, 2011):

- RQ1: How closely do LLM-generated Bowtie diagrams, derived from structured FMEA tables and unstructured narratives, match expert-crafted Bowties in accuracy, interpretability, and structural completeness?

- RQ2: Which prompting strategies and model configurations most effectively improve reliability, reduce hallucinations, and produce consistent, structured outputs?

The main contributions are:

- Three Bowtie generation pipelines for processing FMEA documents (RAG, OCR, and vision-based);
- Evaluation of three different LLMs: Mistral-7B, Qwen-2.5, and LLaMA-3 across structured and narrative inputs;
- Sobol-based sensitivity analysis identifying prompt design as the dominant quality driver;
- Stochasticity experiments revealing pseudo-deterministic behaviour in quantised LLMs; and
- Expert evaluation of narrative Bowtie outputs via Likert scoring.

The following sections detail the background, methodology, and experimental results supporting these contributions.

2. BACKGROUND AND RELATED WORK

2.1 FMEA and Bowtie Diagrams

FMEA and its extension FMECA originated in US military practice in the 1940s (Sharma & Srivastava, 2018) and remain standard tools in reliability engineering. They provide a systematic framework for identifying failure modes, their causes, and effects across a product or process lifecycle, from design FMEA through process and functional variants (Sharma & Srivastava, 2018). FMECA adds a Risk Priority Number (RPN), calculated from severity, occurrence, and detection scores, to help prioritise corrective actions (Rouabhia-Essalhi, Boukrouh, and Ghemari, 2022).

FMEA documentation takes two forms. Structured FMEA tables, with columns such as Failure Mode, Failure Cause, and Failure Effect, allow more direct Bowtie extraction, as content is explicitly labelled. Unstructured narrative descriptions often contain the same causal information buried in dense text; for example, where a table might list “seal degradation” caused by “improper material selection”, a narrative may only describe “significant wear, likely due to material mismatches” (NSWC-11, 2011). Extracting actionable information from unstructured text is considerably harder.

Bowtie diagrams were first referenced in a 1979 ICI report and formalised by Royal Dutch Shell following the Piper Alpha disaster (Wolters Kluwer, 2024). They are now widely used in safety-critical industries, including rail (Turner et al., 2017) and healthcare (Wierenga et al., 2009). A standard Bowtie has four components: causes (initiating events), the critical event (central failure mode), consequences (resulting outcomes), and barriers (both preventive measures on the cause side and mitigating measures on the consequence side) (Turner et al., 2017). Figure 1 shows an example of a standard Bowtie. Constructing them from FMEA documentation is labour-intensive, typically requiring multi-hour facilitated workshops with multidisciplinary expert teams (Turner et al., 2017; Wierenga et al., 2009), which limits scalability. A related visual tool, the Causal Loop Diagram (CLD), captures system-level feedback dynamics through reinforcing and balancing loops, in which variables are linked by directed arrows annotated with positive or negative signs to indicate the direction of influence (Kim & Andersen, 2012). CLDs are used in the model selection phase of this work as a benchmark for causal extraction capability, as prior LLM-based CLD studies provide labelled datasets and evaluation metrics directly applicable to our initial model ranking task.

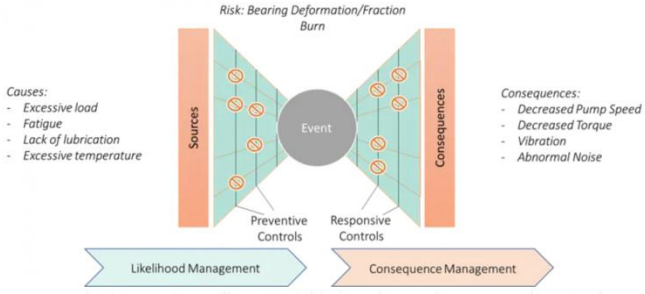

Figure 1. Example Bowtie for a pump (UReason, 2024)

2.2 LLMs for Causal Extraction

LLMs have proven highly effective at extracting structured information from unstructured text (Schwitter, 2025; Kaddour et al., 2023; Yang, Jin et al., 2024). Before LLMs, causal extraction relied on three broad approaches: rule-based systems using manually defined linguistic patterns (e.g., detecting phrases such as *"leads to"* or "*caused by"*), statistical machine learning models such as SVMs trained on labelled causal datasets, and early deep learning architectures such as RNNs and CNNs. Each approach shared critical weaknesses, rigid domain assumptions, reliance on large annotated corpora, and an inability to handle implicit causal relationships or the technical jargon prevalent in engineering documentation (J. Yang et al., 2022). These limitations make them ill-suited to the variability of real-world FMEA documents, motivating a shift to LLMs. In failure analysis, LLMs reduce reliance on manual expert annotation by parsing fault descriptions and consequences from narrative or tabular inputs (Gopalakrishnan et al., 2024; Kiciman, Ness, Sharma, and Tan, 2023). Wan et al. (2024) survey LLM integration into causal discovery more broadly, noting promise but flagging hallucination, prompt sensitivity, and the absence of standardised benchmarks as key limitations. Applied work includes Liu and Keith (2024), who demonstrated GPT-based CLD extraction from text; Hassani et al. (2024), who automated FMEA generation with 84–85% extraction accuracy using GPT-3.5; Hosseinichimeh, Majumdar, Williams, and Ghaffarzadegan (2024), who built an SD Bot achieving 59% link accuracy on expert CLDs; and

Taramsari, Rao, Nilchiani, and Lipizzi, (2024), who applied NLP techniques to U.S. Air Force accident reports to generate CLDs capturing cascading failure interdependencies in aerospace systems. However, no prior work directly targets the automated generation of Bowtie diagrams from FMEA documentation.

Three prompting strategies are used in this study. Zero-shot prompts give only a task description with no examples. Few-shot prompts include input-output examples to guide output format (Brown et al., 2020). Chain-of-thought (CoT) prompts ask the model to produce intermediate reasoning steps before the final answer, improving performance on multi-step tasks (Wei et al., 2022). All three strategies have been shown to influence reasoning ability and output quality (Anagnostidis & Bulian, 2024).

2.3 Retrieval-Augmented Generation

RAG improves LLM output by grounding responses in retrieved external evidence, reducing hallucination (Lewis et al., 2021). A retriever identifies relevant passages from an indexed knowledge base; these are concatenated with the query before generation. Dense Passage Retrieval (Karpukhin et al., 2020) further optimises retriever performance, allowing RAG to scale across larger datasets. In failure analysis, RAG supports accurate causal link generation by anchoring failure modes to their source evidence. Zhang et al. (2024) demonstrate this directly in causal graph generation, showing that retrieval-augmented models produce more precise outputs than standard LLMs without RAG. The effectiveness depends on the quality of the knowledge base, and it is less applicable in domains with scarce data (Gao et al., 2024; Khatibi, Abbasian, Yang, Azimi, and Rahmani, 2024).

2.4 Research Gap

Existing work has focused on CLD generation or generic cause-and-effect extraction. Prior work has explored automated generation of CLDs using LLMs; however, generating Bowtie diagrams, which require stricter causal structure and explicit barrier modelling, remains largely unexplored. Three specific gaps motivate this study. First, no prior work directly targets the automated construction of Bowtie diagrams from FMEA documentation; existing automation efforts have focused on CLDs or generic cause-and-effect extraction, neither of which requires the stricter causal structure or explicit barrier modelling that Bowties demand. Second, existing LLM-based approaches are evaluated on simplified or synthetic text, rarely confronting the inconsistently formatted tables and mixed structured-narrative content characteristic of real industrial FMEA documents. Third, the combined effect of prompt type, prompt strictness, context source, and decoding parameters on output quality has not been systematically studied; without such a study, practitioners have no principled basis for configuring LLM-based extraction pipelines. This work directly addresses all three gaps.

3. METHODOLOGY

3.1 Experimental Overview

The experimental framework runs in three phases. Phase 1 benchmarks eight open-source LLMs on causal extraction and hallucination tendency to select the top three models. Phase 2 applies a two-stage Sobol sensitivity analysis to identify which parameters drive Bowtie quality. Phase 3 tests output consistency across random seeds for three mechanical components and evaluates model performance on unstructured narrative input.

All experiments were run in Python 3.9, using llama-cpp-python for local LLM inference. Semantic retrieval was implemented using sentence-transformers and FAISS. PDF extraction utilised PaddleOCR and pdfplumber. Models were run in GGUF quantised format via llama.cpp. The vision model used in Pipeline 3 is Mistral-Small.

3.2 Model Selection and Hallucination Testing

Eight open-source LLMs were evaluated on CLD extraction tasks using four narrative examples from Liu and Keith (2024), covering both explicit and implicit causal variables. Models were assessed using Jaccard similarity for variable extraction and precision, recall, and F1 for causal links (Gopalakrishnan, Garbayo, and Zadrozny, 2024; Azam et al., 2024). The eight candidates were: LLaMA-3-Instruct (8B), Mistral-Base, Mistral-Code, Mistral-Instruct, Mistral-Math, Qwen-2.5-Instruct (all 7B), R1-Distill-LLaMA (8B), and R1-Distill-Qwen (7B). A structured four-step CoT prompt guided models through variable extraction, causal link identification, CLD generation in DOT format (a standard graph description language), and structured JSON output.

Hallucination tendencies were assessed using ten false-premise and logically inconsistent questions (Yuan et al., 2024; Zhu et al., 2024; Li et al., 2025), evaluated under two conditions: a baseline prompt with no uncertainty guidance, and an uncertainty-aware prompt instructing the model to reply “I don’t know” if unsure. Responses were categorised as Hallucinated, Unclear/Hedged, or Correct Rejection. Results from both tasks informed the selection of the top three models for downstream experiments.

3.3 Dataset and Ground Truth

All pipelines are benchmarked using failure scenarios from the US Navy Reliability Handbook (NSWC-11, 2011), which provides structured FMEA tables and narrative text for several mechanical components. Ground-truth Bowtie diagrams were constructed manually by a domain expert for three components: a sensor, a valve assembly, and a shaft. Figure 2 shows a ground-truth example of a Bowtie detailing

the cause and consequence of a cracked fitting housing. For the narrative experiments, three expert Bowties centred on dynamic seal failures served as reference standards. A known complication is that expert reference diagrams sometimes include implicit variables not stated explicitly in the source text, which can blur the line between valid abstraction and hallucination during evaluation.

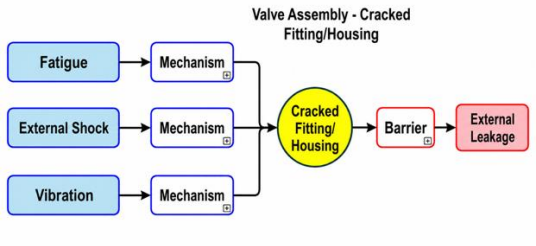

Figure 2. Expert ground truth Bowtie: Valve Assembly

3.4 Bowtie Generation Pipelines

Three pipelines were developed to generate structured Bowtie JSON from FMEA documents; Table 1 summarises each of their workflows.

Table 1. High-level comparison of Bowtie pipelines

Pipeline	Input	Key steps	Output
RAG	**Markdown FMEA table**	**1. Chunk & index 2. Semantic retrieval 3. Structured Prompt → JSON**	**JSON to Bowtie**
OCR	**Table images**	**1. OCR 2. Markdown clean-up 3. Prompt → JSON**	**JSON to Bowtie**
Dual-LLM (Vision)	**Table images**	**1. Image → Vision LLM 2. Markdown + Summary sent to Small LLM 3. Small LLM → JSON**	**JSON to Bowtie**

Pipeline 1 uses RAG: FMEA content is chunked, embedded with all-MiniLM-L6-v2, and indexed with FAISS. Relevant chunks are retrieved by part name and passed as structured prompt context.

Pipeline 2 uses OCR: a table image is processed with PaddleOCR and Img2Table, then post-processed into Markdown and passed to the LLM via a structured prompt.

Pipeline 3 is a dual-LLM vision approach: Mistral-Small processes a table image, producing a Markdown table and a structured description; a smaller second LLM then generates Bowtie JSON from this intermediate output using a user-selected prompt strategy.

3.5 Evaluation Metrics

Node labels in predicted Bowties are matched to ground-truth counterparts using SBERT-based cosine similarity at a threshold of 0.75, selected empirically to balance precision in role assignment against recall of semantically equivalent node labels. Jaccard similarity, precision, recall, F1 score, and approximate Graph Edit Distance (GED) are then computed for both nodes (by semantic role) and edges (by type). The GED reported here is a simplified, count-based proxy, as it is more computationally efficient compared to the classic graph-theoretic minimum edit path. This equals the sum of node and edge false positives and false negatives after semantic alignment, providing a computationally efficient estimate of the overall diagram discrepancy.

3.6 Sobol Sensitivity Analysis

Two global sensitivity analysis experiments were conducted using the Sobol method (Saltelli et al., 2010; Sobol, 2001) to decompose the output variance into contributions from the different experimental parameters. Confidence intervals were estimated via 1000 bootstrap resamples using the SALib library (Herman & Usher, 2017). A fixed seed of 2242 ensured reproducibility. Three indices were computed: S1 (first-order, quantifying each parameter's main effect), ST (total-order, capturing main effects plus all interactions), and S2 (second-order, isolating pairwise interactions). Three strictness levels were defined for each prompt type: Level 1 provided minimal schema guidance, Level 2 included FMEA field references and basic fallback logic, and Level 3 specified fully explicit extraction rules with field mappings and value constraints.

Sobol Run 1 varied temperature (0.1–1.0), top-p (0.1–1.0), and prompt strictness, with context type fixed to RAG and prompt type fixed to zero-shot (N = 64 base samples per model, 512 evaluations total). Sobol Run 2 expanded the parameter space to include prompt type (zero-shot, few-shot, CoT), prompt strictness (Levels 1–3), and context type (RAG, OCR, Vision), using N = 1024 base samples per model (8192 evaluations). The larger sample size in Run 2 produced narrower confidence intervals and greater statistical power.

3.7 Stochasticity and Narrative Experiments

The top two prompt configurations per model from Sobol Run 2 were evaluated across 10 independent runs with different random seeds, holding temperature at 0.5 and top-p at 0.95 for all three FMEA components. For the narrative experiment, the dynamic seal component was used as input. Five domain experts rated each model-generated Bowtie on usability, completeness, clarity, and accuracy using a 5-point Likert scale.

4. RESULTS

4.1 Model Selection and Hallucination Testing

Table 2 summarises the performance of the three models selected for downstream Bowtie experiments: Mistral-

Instruct, LLaMA-3-Instruct, and Qwen-2.5-Instruct. Mistral-Instruct achieved the highest average F1 score of 0.65 across all four test prompts. Models performed best on prompts with fewer implicit variables; prompts containing vague or implicit causal relationships produced notably lower scores, a pattern that would recur in narrative experiments.

Table 2. Performance of selected models

Model	Jac.	Prec.	Rec.	F1	Hal.% (no guidance)	Hal.% (with guidance)
Mistral-Instruct	0.675	0.771	0.562	0.650	20.0%	20.0%
LLaMA-3-Instruct	0.712	0.687	0.500	0.578	0.0%	10.0%
Qwen-2.5-Instruct	0.650	0.750	0.478	0.583	10.0%	0.0%

The final trio spans three distinct architectural families: Mistral, Meta (LLaMA), and Alibaba (Qwen). R1-Distill variants were excluded despite competitive causal-extraction scores due to higher hallucination rates and reduced reliability on logically flawed inputs. Notably, models with built-in chain-of-thought reasoning (DeepSeek-R1 distillations) displayed more variable hallucination behaviour than standard instruction-tuned models, suggesting that advanced reasoning capabilities do not inherently reduce susceptibility to false premises. Mistral-Base, Mistral-Code, and Mistral-Math were excluded due to frequent schema violations and poor JSON validity.

Across 80 total responses, uncertainty-aware prompting reduced hallucinated responses from 21 to 9 and unclear or hedged responses from 10 to 11, with correct rejections increasing from 49 to 60. LLaMA-3-Instruct showed a slight increase in hallucinations when uncertainty guidance was added; Qwen-2.5-Instruct improved; Mistral-Instruct was unchanged. Mistral-Code and Mistral-Base also exhibited a high frequency of unclear or hedged responses, outputs that avoided outright hallucination but failed to reject false premises clearly. In safety-critical domains, vague causal statements can erode trust as much as explicit errors.

Of 32 outputs assessed for schema compliance, 12 were produced as plain text rather than valid JSON, an error rate of 37.5%. Qwen-2.5-Instruct was the most consistent, producing valid JSON across all four test inputs. Mistral-Code was the least reliable, failing on all inputs. This disconnect between semantic accuracy and schema validity underscores that strong causal reasoning ability does not guarantee compliance with structured output requirements, a critical limitation for automated, schema-dependent workflows.

4.2 Sobol Sensitivity Analysis

Sobol Run 1: Across all three models, prompt strictness was the dominant factor (S1 ≈ 0.99, ST = 1.00). Temperature and top-p had negligible effects on both node and edge F1; their S1 and ST values were near zero across all models. This justified setting the temperature to 0.5 and top-p to 0.95 for all subsequent experiments.

Sobol Run 2: For Node F1, prompt strictness and prompt type both showed moderate first-order effects (e.g. LLaMA: S1 = 0.24, ST = 0.71 for strictness; Qwen: S1 = 0.44, ST = 0.80). Context type was consistently non-significant (S1 ≈ 0), indicating that all three pipelines are interchangeable under equivalent prompt conditions. Figure 3 shows the first- and total-order indices for node F1. In the Sobol analysis figures, PT denotes prompt type, PS denotes prompt strictness, and CT denotes context type.

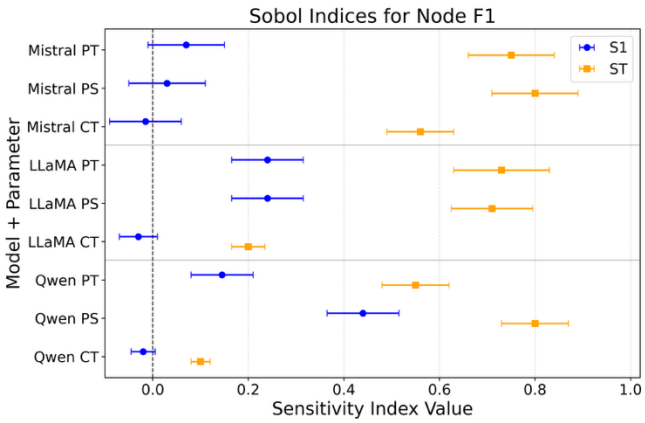

Figure 3. Sobol Run 2: S1 and ST indices with confidence intervals for node F1 across models

Prompt strictness was the dominant factor for edge F1 across all models (S1 = 0.34–0.56, ST = 0.72–0.92). Prompt type showed a weaker first-order effect but moderate total-order contributions, confirming interaction effects. Context type remained negligible for edge F1 as well. Figure 4 shows the corresponding indices.

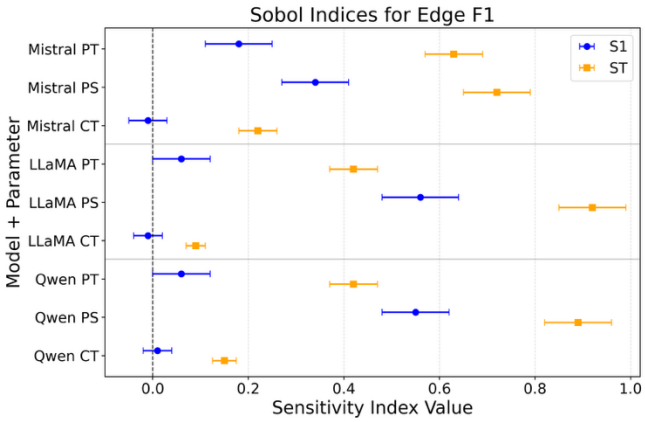

Figure 4. Sobol Run 2: S1 and ST indices with confidence intervals for edge F1 across models

Figure 5 shows the corresponding second-order indices. Second-order Sobol indices (S2) were computed to quantify

pairwise parameter interactions. Across all models and both metrics, the interaction between prompt type and prompt strictness was consistently the strongest effect (S2 = 0.24–0.33 for Node F1; S2 = 0.25–0.30 for Edge F1), and the only interaction where the lower confidence bound consistently exceeded zero. All other parameter pairs, including those involving context type, showed weak or negligible interactions (generally S2 < 0.16, with confidence intervals overlapping zero). This confirms that prompt engineering must be treated as a joint design problem: neither prompt type nor strictness alone explains the observed variance.

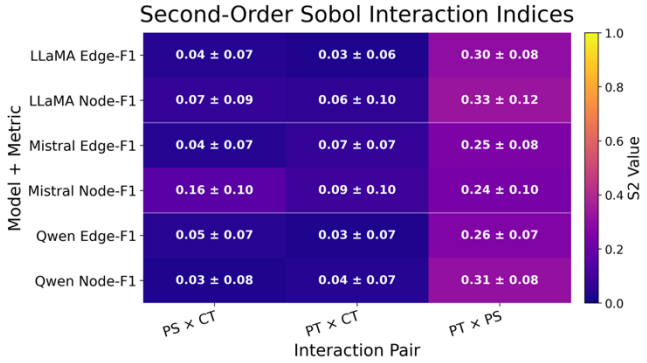

Figure 5. Second-order Sobol indices (S2) with confidence intervals for pairwise parameter interactions

4.3 Identification of Optimal Prompt Configurations

To identify optimal configurations, model performance was evaluated across all prompt types and strictness combinations for each model. LLaMA-3-Instruct achieved near-perfect node and edge F1 with zero-shot + high and CoT + high prompts, with sharp declines under less strict or few-shot prompting. Mistral-Instruct performed best with zero-shot + high and CoT + medium, while few-shot prompts caused edge F1 to collapse to zero despite moderate node F1 scores. Qwen-2.5-Instruct was distinct in that medium strictness consistently outperformed high strictness, with CoT + medium and zero-shot + medium as optimal. Across all three models, edge F1 was more sensitive to prompt design than node F1; a moderately high node score did not guarantee a high edge score, as omitting intermediary roles such as Barrier or Mechanism breaks the causal chain regardless of how many other nodes are correctly extracted. Table 3 shows the top two configurations per model selected for downstream experiments.

Table 3. Top prompt configurations per model

Model	Configuration 1	Configuration 2
LLaMA-3-Instruct	**Zero-shot + High**	**CoT + High**
Mistral-Instruct	**Zero-shot + High**	**CoT + Medium**
Qwen-2.5-Instruct	**CoT + Medium**	**Zero-shot + Medium**

4.4 Stochasticity Results

Outputs were pseudo-deterministic: for any fixed combination of model, prompt, and context, all 10 runs across different random seeds produced identical JSON outputs. This held across all components and configurations. Output variability was driven by prompt design and model choice rather than by random sampling. However, this pattern may not generalise to less-constrained tasks, ambiguous inputs, or full-precision models. This behaviour is attributed to quantisation effects that narrow the output probability distribution, combined with highly structured prompts that reduce input ambiguity. Few-shot prompts at high strictness were particularly prone to malformed outputs: LLaMA-3-Instruct (vision pipeline) and Qwen-2.5-Instruct (OCR pipeline) frequently produced no valid structure, instead copying the input prompt or generating repetitive causal chains. Table 4 shows the mean Node F1, Edge F1, and GED across all three components for each configuration.

Table 4. Mean Metrics: Sensor, Valve, & Shaft components

Configuration	Node F1	Edge F1	GED
LLaMA-3 CoT+High	**0.94**	**0.93**	**3**
LLaMA-3 Zero+High	**0.93**	**0.92**	**4**
Qwen-2.5 CoT+Medium	**0.92**	**0.90**	**4**
Mistral Zero+High	**0.91**	**0.90**	**6**
Mistral CoT+Medium	**0.86**	**0.71**	**8**
Qwen-2.5 Zero+Medium	**0.81**	**0.58**	**13**

LLaMA-3-Instruct with CoT+High and Zero+High prompting consistently outperformed other configurations. A consistent positive trend was that critical event extraction was largely accurate across most configurations, especially under stricter prompting. For the Sensor component, almost all configurations achieved perfect Node and Edge F1 of 1.0 with GED of 0; the sole exception was Mistral CoT+Medium, which missed the Barrier node, a single omission that propagated to edge-level failures and a GED of 3. The Shaft component showed greater performance fluctuations than Sensor or Valve, with several configurations showing large node-to-edge F1 gaps; for example, Qwen-2.5-Instruct zero+medium achieved a node F1 of 0.80 but an edge F1 of only 0.44, reflecting cascading failures from a missed Critical Event node.

Three recurring error patterns were observed: node omission (Barrier and Mechanism nodes most vulnerable, with omissions propagating to edge-level failures); hallucination and over-segmentation (multiline table entries split into separate nodes, inflating false-positive counts); and role misassignment (causes placed in the Critical Event position or vice versa). Figure 6a-c illustrates two representative failure modes for the Bent Shaft component and the ground truth.

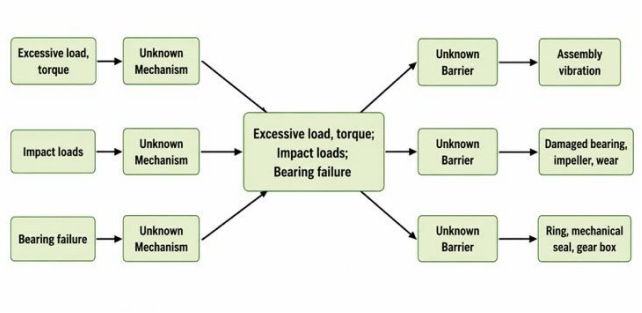

Figure 6a. Qwen Zero+Medium: hallucinated critical event.

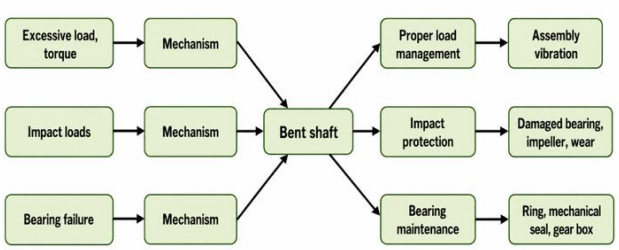

Figure 6b. Mistral CoT+Medium: hallucinated barrier nodes

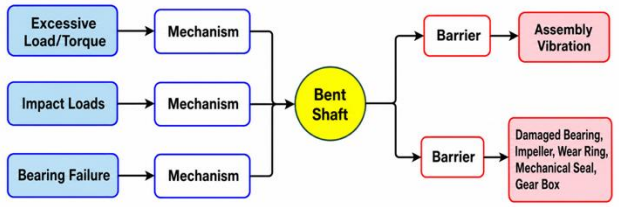

Figure 6c. Ground truth: Bent Shaft

4.5 Narrative Input Results

All models struggled with unstructured narrative input. The dynamic seal component was used as the test case, with three expert Bowties covering Wear, Seal Failure, and Lifetime serving as reference standards. The Seal Failure Bowtie shown in Figure 7 is an example of an expert-generated Bowtie that the LLM performance was compared to. Notably, the consequence side could not be fully constructed here, as the narrative lacked barrier and consequence information.

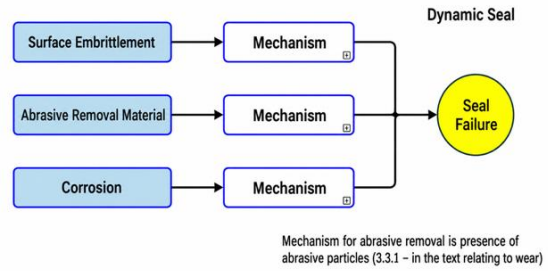

Figure 7. Expert Bowtie: Seal Failure (narrative input)

Table 5 shows average expert Likert scores (1–5 scale). Zero-shot prompting produced higher scores than CoT across LLaMA-3-Instruct and Mistral-Instruct. The strongest configuration was LLaMA-3-Instruct Zero+High (clarity 3.80, usability 3.50). Qwen-2.5-Instruct Zero+Medium received the lowest ratings across all criteria (average ≈ 1.35), suggesting significant limitations in extracting accurate Bowtie structures from narrative text.

Table 5. Average expert Likert scores for narrative outputs

Model / Prompt	Usability	Completeness	Clarity	Accuracy
LLaMA-3 Zero+High	**3.50**	**3.20**	**3.80**	**3.00**
Mistral Zero+High	**3.50**	**3.60**	**3.20**	**2.60**
LLaMA-3 CoT+High	**3.25**	**2.60**	**3.60**	**2.40**
Qwen-2.5 CoT+Medium	**2.75**	**2.40**	**2.80**	**2.20**
Mistral CoT+Medium	**2.00**	**2.40**	**2.20**	**2.00**
Qwen-2.5 Zero+Med.	**1.40**	**1.20**	**1.60**	**1.20**

Four consistent failure modes were identified: models frequently elevated causes or mechanisms to Critical Event status; implicit variables present in expert diagrams but absent from the source text were rarely extracted; barriers were often generic placeholders or hallucinated content; and models typically produced more Bowties than the expert ground truth, fragmenting a single expert diagram into multiple smaller ones with similar structure and generic labels.

One reviewer identified a specific hallucination where a model generated a node for "abrasive cutting and damage to shafts", prompting the comment "a dynamic seal does not contain a shaft", illustrating how terms from elsewhere in the source text can be misattributed to the wrong component. While some outputs were described as "workable" with clear causal relationships, none fully matched the expert reference diagrams in structure or critical event identification.

5. DISCUSSION

5.1 Answering the Research Questions

RQ1: With strict schema-constrained prompting, instruction-tuned LLMs generated Bowtie diagrams from FMEA tables that closely matched expert references, with Node and Edge F1 scores routinely above 0.8 and perfect extraction achieved for some components on tabular data. For unstructured narratives, performance dropped substantially: models failed to identify critical events, misassigned roles, and hallucinated

nodes. Interpretability and structural completeness were high for curated tables but low for free-form technical text.

RQ2: Prompt strictness and prompt type were the dominant drivers of output quality. Zero-shot and CoT prompts with explicit schema instructions consistently outperformed few-shot approaches, maximising both node- and edge-level F1 and minimising output variance across random seeds. Pipeline choice played a secondary role as long as input quality was maintained. Uncertainty-aware prompting reduced but did not eliminate hallucinations. Schema-constrained prompting produced quasi-deterministic outputs for tabular data, though multi-line entries remained a persistent edge case.

5.2 Pseudo-Deterministic Behaviour

The observation of identical outputs across 10 random seeds can be attributed to four interacting factors. First, quantisation reduces the bit precision in model weights, lowering the entropy of the output probability distributions and making sampling effectively deterministic. Second, highly structured, schema-constrained prompts paired with clean tabular input reduce ambiguity, leaving little room for alternative outputs. Third, Sobol analysis showed that temperature, top-p, and seed have negligible effects, suggesting that the probability distributions are sharply peaked around a single dominant output. Fourth, all three models are instruction-tuned, which produces more predictable outputs for well-defined tasks. This pattern may not hold for less constrained tasks, ambiguous inputs, or full-precision models.

5.3 Hallucinations and Extraction Challenges

Narrative extraction was considerably harder than tabular extraction. The elevation of causes or mechanisms to Critical Event status and the failure to extract implicit variables not explicitly stated in the text were the most common failure modes. This aligns with findings from the SD Bot study (Hosseinichimeh et al., 2024), which reported only 59% link accuracy when expert CLDs required implicit causal variables, a limitation also observed here.

A specific contributor to role-assignment errors was open-ended fallback logic in medium-strictness prompts, instructions such as “extract mechanism from text; if none available, use Mechanism”, which encouraged models to populate every role even when ground-truth diagrams left certain roles absent. Fallback instructions should be tightly constrained rather than open-ended. Additionally, implicit variables such as “time” may not be stated explicitly in the text but still appear as nodes in expert diagrams, posing a significant challenge for smaller LLMs. Hallucination and over-segmentation issues also occurred in structured tables, where multiline FMEA entries were sometimes split into separate nodes despite explicit instructions. These findings suggest that neither prompt engineering nor advanced model reasoning alone is sufficient; reliable automation will likely require combining prompt design with robust input preprocessing, model safeguards, and post-processing validation.

5.4 Limitations

The evaluation was restricted to small, quantised, open-source models (7–8B parameters); proprietary models such as GPT-4.1 may perform differently. Prompt templates were tailored to NSWC-11's specific column structure and vocabulary and are unlikely to generalise to FMEA documents from other OEMs with different formatting or narrative styles without additional prompt engineering. Narrative evaluation was limited to the dynamic seal component, and all experiments focused on single-component failures; the pipelines do not support multi-component Bowtie chaining. All pipelines operate in single-pass inference mode with no mechanism to correct schema violations or missing roles. Evaluation relied on manually curated ground-truth diagrams, which introduces scalability concerns for continuous validation in practice.

5.5 Future Work

Five directions are most promising. First, prompt ablation studies that vary individual schema elements, such as fallback rules or step count, would isolate which prompt components matter most. Second, evaluation should extend to larger open-source models (LLaMA 4, Mistral Magistral) and proprietary APIs to test whether current limitations reflect model capability or scale. Third, multi-stage agentic workflows, where specialised agents extract, validate, and re-prompt for missing or inconsistent roles, could address the single-pass limitation. Fourth, integrating explicit confidence scoring into LLM outputs would improve interpretability and support downstream expert validation; human-in-the-loop review loops would be particularly valuable in edge cases. Finally, building on a Neo4j-based prototype developed during this work, a knowledge graph backend would enable cross-component reasoning by capturing how consequences in one component propagate as causes in another, supporting system-level failure analysis at scale.

6. CONCLUSION

This paper presented ACE, a framework for automating the generation of Bowtie diagrams from FMEA documentation using open-source, instruction-tuned LLMs. Three extraction pipelines were developed and evaluated across structured tabular and unstructured narrative inputs. Sobol sensitivity analysis identified prompt strictness and prompt type as the dominant drivers of output quality, while decoding parameters and pipeline choice had a negligible effect. Hallucination testing further demonstrated that uncertainty-aware prompting and model selection can meaningfully reduce speculative outputs, though they do not eliminate the

risk entirely. Under optimal prompt configurations, models achieved Node and Edge F1 scores above 0.8 on structured FMEA tables, with outputs remaining stable across repeated runs.

Narrative input generation remains an open problem: models struggled to identify critical events, extract implicit variables, and align with expert reference diagrams. The generalisability of these results beyond well-structured FMEA tables requires further investigation. These findings establish a working foundation for LLM-based Bowtie automation in industrial failure analysis while identifying the specific barriers to broader deployment.

When paired with schema-constrained prompts and careful preprocessing, open-source LLMs can automate much of the manual effort involved in Bowtie generation from structured technical documents, offering clear benefits for scalability and standardisation in contexts where FMEA formats are widely adopted.

The success of such pipelines in real-world deployment will depend on the quality of OCR for scanned documents and on careful preprocessing of both tabular and narrative input. LLM-driven Bowtie automation is best positioned to augment human expertise rather than replace it, reducing the effort required to construct first-pass Bowties while still relying on human-in-the-loop validation and more capable models. Agentic workflows, larger models, and knowledge graph integration offer the most promising directions for extending this capability to more complex, multi-component failure scenarios.

For PHM practitioners, the most immediate implication is that structured FMEA documentation, already standard in many industrial asset management workflows, can serve as a direct input to automated Bowtie generation without manual reformatting. The three pipelines presented here accommodate the two most common real-world input formats: digital FMEA tables via RAG and scanned or image-based tables via OCR and vision. When FMEA documentation is consistently structured, this approach offers a practical way to accelerate Bowtie creation and reduce the burden on expert facilitation that currently limits Bowtie adoption at scale.

NOMENCLATURE

ACE	Automating Causal Extraction
CLD	Causal Loop Diagram
CoT	Chain-of-Thought prompting
CT	Context Type
DOT	Graphviz DOT graph description format
DPR	Dense Passage Retrieval
Edge F1	Edge-level F1 score
FAISS	Facebook AI Similarity Search
Few-shot	Prompting with example pairs
FMEA	Failure Mode and Effects Analysis
FMECA	Failure Mode, Effects, and Criticality Analysis
GED	Graph Edit Distance
GGUF	GPT-Generated Unified Format
Jac.	Jaccard similarity
JSON	JavaScript Object Notation
LLM	Large Language Model
Node F1	Node-level F1 score
OCR	Optical Character Recognition
PHM	Prognostics and Health Management
PS	Prompt Strictness
PT	Prompt Type
RAG	Retrieval-Augmented Generation
RPN	Risk Priority Number
S1	First-order Sobol sensitivity index
S2	Second-order Sobol sensitivity index
SALib	Sensitivity Analysis Library
SBERT	Sentence-BERT (Bidirectional Encoder Representations from Transformers)
ST	Total-order Sobol sensitivity index
Zero-shot	Prompting without examples

REFERENCES

Anagnostidis, S., & Bulian, J. (2024). How susceptible are LLMs to influence in prompts? arXiv:2408.11865.

Azam, M., Chen, Y., Arowolo, M. O., Liu, H., Popescu, M., & Xu, D. (2024). A comprehensive evaluation of large language models in mining gene relations and pathway knowledge. *Quantitative Biology*, 12(4), 360–374.

Brown, T., Mann, B., Ryder, N., Subbiah, M., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2024). Retrieval-augmented generation for large language models: A survey. arXiv:2312.10997.

Gopalakrishnan, S., Garbayo, L., & Zadrozny, W. (2024). Causality extraction from medical text using large language models (LLMs). arXiv:2407.10020.

Hassani, I. E., Masrour, T., Kourouma, N., Motte, D., & Tavčár, J. (2024). Integrating large language models for improved FMEA: A framework and case study. *Proceedings of the Design Society*. doi:10.1017/pds.2024.204

Herman, J., & Usher, W. (2017). SALib: An open-source Python library for sensitivity analysis. *Journal of Open Source Software*, 2(9), 97.

Hosseinichimeh, N., Majumdar, A., Williams, R., & Ghaffarzadegan, N. (2024). From text to map: A system dynamics bot for constructing causal loop diagrams. *System Dynamics Review*, 40(3), e1782.

Kaddour, J., Harris, J., Mozes, M., Bradley, H., Raileanu, R., & McHardy, R. (2023). Challenges and applications of large language models. arXiv:2307.10169.

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W. (2020). Dense passage retrieval for open-domain question answering. arXiv:2004.04906.

Khatibi, E., Abbasian, M., Yang, Z., Azimi, I., & Rahmani, A. M. (2024). ALCM: Autonomous LLM-augmented causal discovery framework. arXiv:2405.01744.

Kiciman, E., Ness, R., Sharma, A., & Tan, C. (2023). Causal reasoning and large language models: Opening a new frontier for causality. arXiv:2305.00050.

Kim, H., & Andersen, D. F. (2012). Building confidence in causal maps generated from purposive text data: Mapping transcripts of the federal reserve. *System Dynamics Review*, 28(4), 311–328.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2021). Retrieval-augmented generation for knowledge-intensive NLP tasks. arXiv:2005.11401.

Li, B., Jiang, G., Li, N., & Song, C. (2024). Research on large-scale structured and unstructured data processing based on large language model. In *Proceedings of MLPRAE '24* (pp. 111–116). ACM.

Li, N., Song, Y., Wang, K., Li, Y., Shi, L., Liu, Y., & Wang, H. (2025). Detecting LLM fact-conflicting hallucinations enhanced by temporal-logic-based reasoning. arXiv:2502.13416.

Liu, N.-Y. G., & Keith, D. (2024). Leveraging large language models for automated causal loop diagram generation. Available at SSRN 4906094.

Naval Surface Warfare Center. (2011). *Handbook of reliability prediction procedures for mechanical equipment (NSWC-11)*. Retrieved from https://reliabilityanalyticstoolkit.appspot.com

Rouabhia-Essalhi, R., Boukrouh, E. H., & Ghemari, Y. (2022). Application of failure mode effect and criticality analysis to industrial handling equipment. *The International Journal of Advanced Manufacturing Technology*, 120(7), 5269–5280.

Saltelli, A., et al. (2010). Variance based sensitivity analysis of model output. *Computer Physics Communications*, 181(2), 259–280.

Schwitter, N. (2025). Using large language models for preprocessing and information extraction from unstructured text. *Methodological Innovations*, 18(1), 61–65.

Segismundo, A., & Cauchick Miguel, P. A. (2008). Failure mode and effects analysis (FMEA) in the context of risk management in new product development. *International Journal of Quality & Reliability Management*, 25(9), 899–912.

Sharma, K. D., & Srivastava, S. K. (2018). Failure mode and effect analysis (FMEA) implementation: A literature review. Retrieved from https://api.semanticscholar.org/CorpusID:115607603

Sobol, I. M. (2001). Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*, 55(1–3), 271–280.

Taramsari, H. B., Rao, B., Nilchiani, R., & Lipizzi, C. (2024). Identification of variables impacting cascading failures in aerospace systems: A NLP approach. In *Conference on Systems Engineering Research* (pp. 413–427). Springer.

Turner, C., Hamilton, W. I., & Ramsden, M. (2017). Bowtie diagrams: A user-friendly risk communication tool. *Proceedings of the Institution of Mechanical Engineers, Part F*, 231(10), 1088–1097.