FMEA Builder: Expert Guided Text Generation for Equipment Maintenance

Karol Lynch , **Fabio Lorenzi** , **John Sheehan** , **Duygu Kabakci-Zorlu** and **Bradley Eck**
IBM Research Europe, Dublin, Ireland
karol_lynch@ie.ibm.com, fabio.lorenzi1@ibm.com, john.d.sheehan@ie.ibm.com,
duygu.kabakci.zorlu@ibm.com, bradley.eck@ie.ibm.com

Abstract

Foundation models show great promise for generative tasks in many domains. Here we discuss the use of foundation models to generate structured documents related to critical assets. A Failure Mode and Effects Analysis (FMEA) captures the composition of an asset or piece of equipment, the ways it may fail and the consequences thereof. Our system uses large language models to enable fast and expert supervised generation of new FMEA documents. Empirical analysis shows that foundation models can correctly generate over half of an FMEA's key content. Results from polling audiences of reliability professionals show a positive outlook on using generative AI to create these documents for critical assets.

1 Introduction

We propose an AI system for the generation of structured documents related to industrial equipment with particular focus on Failure Mode and Effects Analysis (FMEA). FMEAs are a longstanding tool of reliability engineering for understanding equipment failure points and optimal maintenance strategies [Rausand and Høyland, 2004; Sharma and Srivastava, 2018]. These documents capture reasons why equipment, assets and infrastructures fail and outline maintenance options for such failures to achieve the desired level of reliability.

Although FMEAs content can differ by sector, our approach considers a document with the following sections:

- The boundary gives a functional description along with the main components.
- Failure locations are points on the equipment where a failure might occur.
- Degradation mechanisms describe the physical process or mechanism that can lead to a failure.
- Degradation influences describe the underlying causes of a degradation.
- Preventative maintenance tasks can be carried out to prevent failures; and,
- Job plans collect such tasks into a schedule.

These parts of the document show a nested behaviour wherein a typical piece of equipment has multiple failure locations with each being linked with one or more degradation mechanisms and so on. Preventative activities depend on the failure location, mechanism, and influence. Job plans schedule preventative activities according to operating conditions, while usually grouping related preventative steps together.

Such an approach is regarded as being effective to managing critical infrastructure in many sectors including Energy and Utilities, Water and Wastewater management, and Oil and Gas [Carvalho *et al.*, 2022] where the effect of unforseen, unplanned or disastrous failures without solid recovery strategies has severe impacts on the system, business and potentially society as a whole.

Creating an FMEA requires a group of highly trained experts focusing on a single study. Such resources might be too expensive or unavailable to some organisations involved in critical infrastructure management.

Generating FMEAs is challenging because the sequential relationship of the document's sections can propagate errors and because FMEAs contain domain specific knowledge about the equipment and how it is used. In addition, the same words can refer to different equipment components, with the correct interpretation depending on the usage. For example the "casing" of a pump is different than that of a window. This behavior makes it difficult to create a navigable catalog of components from which to build FMEAs. However, the attention mechanism [Vaswani *et al.*, 2017] used in today's language models can interpret the meaning of words based on their context.

In this discussion we explore how large language models (LLMs) [Bubeck *et al.*, 2023] can assist in the creation of FMEAs. Our system for generating FMEAs draws on recent techniques for using LLMs and contributes a case study for using LLMs on domain-specific problems. Key techniques informing our work include answer consistency [Wang *et al.*, 2023], in-context learning [Brown *et al.*, 2020], and dynamic relevant example selection [Liu *et al.*, 2022; Nori *et al.*, 2023]. Recent studies applying LLMs to particular domains include the work of Nori et al. [Nori *et al.*, 2023] for medical tasks and Balaguer et al. [Balaguer *et al.*, 2024] for agriculture. Those studies respectively showed that innovative prompting could achieve state of the art performance and that retrieval augmented generation and fine-tuning both had

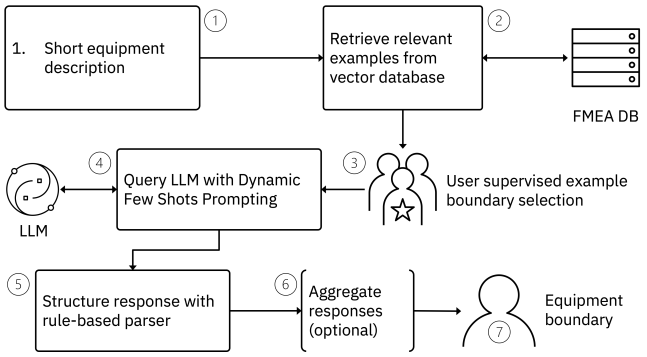

Figure 1: Flow of a single step (boundary generation) in our system: The step's input (1) is used to select candidate examples from the database (2). Examples confirmed by the user (3) appear in the prompt (4). The system parses the LLM's response (5) and optionally aggregates responses from multiple prompt, model variations (6) for presentation to the user (7).

advantages for domain-specific problems.

With this landscape, our main contributions are experiments comparing the performance of several LLMs on generating parts of an FMEA. We also share feedback on the work in progress design for an interactive system that enables convenient collaboration between LLMs and human experts for FMEA creation.

In the remainder of the paper, we describe our approach to generating FMEAs and share results comparing different LLMs for the task. We also share survey feedback from target users of our current prototype. We conclude with a summary of the future directions planned for exploration.

2 Solution Approach

Our system for creating new FMEAs decomposes the generation problem according to the structure of the documents we generate. This decomposition enables our target users, subject matter experts, to inject their knowledge and supervise the generation process. We use a library of existing documents to furnish the model with relevant information, and generate structured outputs for consumption by our graphical interface. Taken together, these methods should enable experts to create new FMEAs of high quality in a short time.

The workflow for the first step, generating the equipment boundary from a short description, is a representative example for generating part of an FMEA (Fig. 1). The prompt construction and response parsing steps are further described below.

Dynamic Few Shot Prompting (DFSP) We rank and retrieve relevant examples, or shots, by cosine similarity of text embeddings. This approach, termed Dynamic Few Shot Prompting, adds user supervision to the example selection methods of [Liu *et al.*, 2022; Nori *et al.*, 2023]. DFSP combines three sources of knowledge: examples already in the database, an expert's knowledge, and the knowledge in the LLM. Running the prompts without injected examples provides a *zero-shot* method that draws only on the LLM.

Structured responses Parsing the LLM's response from pure text into structured responses enables presentation of FMEA components directly to the user for subsequent supervision. For example the user can confirm, reject or supplement the generated list of failure locations. Structured responses also simplify the resolution of repeated entities, a common problem in text generation [Holtzman *et al.*, 2020]. Our system generates structured responses by lightly formatting the injected examples with simple delimiters. A rule-based parser operates over the delimiters to produce a response in javascript object notation.

3 Evaluation

For the work in progress reported here we focus on the first two parts of an FMEA document: the *equipment boundary* and, the *failure locations*. Experiments used our propriety database of 714 FMEAs developed by subject matter experts and a range of state of the art LLMs: llama-2-70b-chat (70B) [Touvron *et al.*, 2023], flan-ul2 (20B) [Tay *et al.*, 2023], and quantizated versions of Mixtral-8x7B (8x7B) [Jiang *et al.*, 2024], denoted by Mixtral-Q.

For evaluation purposes we divided the database into train (n=571; 80%), validation (n=71; 10%) and test (n=72; 10%) splits. Examples for our DFSP method are drawn from the training split. We compare our DFSP method with prompts using a randomly selected example (*random-shot*) and no example (*zero shot*). The random shot uses a random example from the database to provide a syntactic rather than semantic hint to the model.

Although it is not possible to share data for reproducing our experiments we suggest that results reported here can provide valuable insights for the domain.

3.1 Generating Equipment Boundaries

An equipment boundary of an industrial asset describes the asset and its constituent components. The input is a few-word description of the equipment and the output is the boundary itself.

We evaluated the performance of several models for generating equipment boundaries in terms of the ROUGE-1 [Lin, 2004] score for the unstructured response, and recall and precision for the structured list of components. ROUGE-1 is a recall oriented similarity score for comparing candidate and reference texts with values ranging from zero to one.

Results (Table 1) for individual models showed a clear pattern of increasing quality as we move from zero-shot to random-shot and DFSP, with flan-ul2 the best performing individual model. Results for DFSP show a strong uplift for ROUGE-1 and for component lists. Although the described methods automatically generate much of the required information, some information, in particular on the component lists, is missed. This result emphasizes the important role of reliability engineers to supervise the generated descriptions.

3.2 Generating Failure Locations

The set of failure locations for an industrial asset is a key part of an FMEA indicating the components of an asset likely to fail. The input to this step is the equipment boundary. This

Table 1: Performance for generating equipment boundaries on test split (n=72).

Model	Method	ROUGE-1	Recall	Prec
flan-ul2	zero-shot	0.133	0.080	0.135
flan-ul2	random-shot	0.274	0.113	0.147
flan-ul2	DFSP	**0.787**	**0.573**	**0.546**
llama2	zero-shot	0.299	0.075	0.082
llama2	random-shot	0.357	0.107	0.076
llama2	DFSP	0.685	0.483	0.415
mixtral-Q	zero-shot	0.238	0.050	0.044
mixtral-Q	random-shot	0.349	0.092	0.056
mixtral-Q	DFSP	0.573	0.342	0.327

Table 2: Performance for generating failure locations on test split (n=72).

Model	Method	Recall	Prec	F1
flan-ul2	zero-shot	0.031	0.139	0.051
flan-ul2	random-shot	0.176	0.351	0.234
flan-ul2	DFSP	0.454	0.585	0.511
llama2	zero-shot	0.229	0.274	0.250
llama2	random-shot	0.243	0.253	0.248
llama2	DFSP	**0.559**	0.597	**0.577**
mixtral-Q	zero-shot	0.040	0.167	0.065
mixtral-Q	random-shot	0.121	0.271	0.168
mixtral-Q	DFSP	0.482	**0.612**	0.539

evaluation uses boundaries from the database to generate failure locations. Results showed a similar pattern as equipment boundaries; quality increases from zero-shot to random-shot to DFSP (Table 2). Results for individual models were more mixed with llama2 showing the highest recall and F1 while mixtral-Q had the highest precision.

3.3 User Feedback

Our system enables user interaction with structured model responses through a tailored graphical interface. As an example, our interface for the first step in the pipeline appears in Figure 2.

We showed the user interface for this work in progress system to two audiences of people responsible for the maintenance and reliability of critical infrastructure across industries. For these professionals, creation and application of FMEAs form a crucial part of their daily work. Following the demonstrations, we polled the audience for feedback on two key questions.

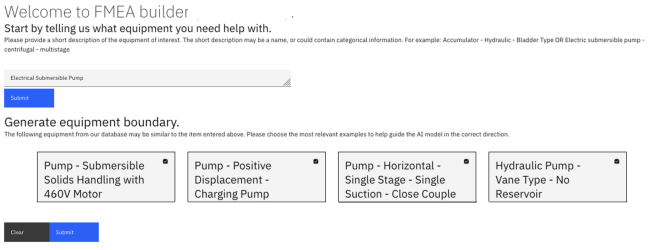

Figure 2: Graphical interface for generating equipment boundaries: The user enters a short description of the equipment as free text. Examples of similar equipment from our database are presented for consideration. Equipment cards that are ticked serve as examples in the prompt to generate an equipment boundary.

Table 3: User Survey Feedback

Audience	Size	Positive [Q1]	Positive [Q2]
A	27	82%	96%
B	55	-	98%

Question 1: *How likely would you be to use a tool like the demo during FMEA creation if it was available?* Answer options: Extremely likely; Likely; Undecided; Unlikely; Extremely unlikely.

Question 2: *How much configurability would you like to have when using the tool during FMEA creation?* Answer options: Build FMEAs fully automated by AI; Build FMEAs mostly automated by AI; Build FMEAs acting as my helpful but supervised assistant; Build FMEAs mostly manually; Build FMEAs fully manually.

Table 3 reports responses favorable to using the tool and to having the support of AI in general. In both audiences, responses were positive. We interpret higher scores for AI in general than for our tool in particular as an opportunity to improve the design of the interface and to familiarize users with the capabilities and limitations of AI.

4 Outlook

In this work we have shared a view of current work in progress for generating documents related to critical equipment. The method of dynamically retrieving examples from a database for including in a prompt shows the ability to correctly generate over half of the content needed for an FMEA. Although this level of performance is considered helpful according to our surveys, we intend to explore further improvements. In particular, ensemble methods based on fuzzy voting provide a promising tool for combining results between models and shot orderings. We expect ensembles to improve the recall of structured responses at a hopefully small cost in precision.

So far our experiments have focused on the test split from the database but there are many equipment types not covered by the database where FMEA generation remains of interest. In these cases, knowledge is often available in the less structured form of manuals and process documents. With pre-processing / chunking this information can also be used to generate parts of an FMEA. Ultimately we foresee the use of ensemble methods to combine results between LLM responses informed by examples from the database as well as user-provided manuals and documents.

Finally further feedback sessions remain to be conducted during the development of this project to evaluate the perceived quality of the generated documents as opposed to the internal algorithmic evaluation.

References

[Balaguer *et al.*, 2024] Angels Balaguer, Vinamra Benara, Renato Luiz de Freitas Cunha, Roberto de M. Estevão Filho, Todd Hendry, Daniel Holstein, Jennifer Marsman, Nick Mecklenburg, Sara Malvar, Leonardo O. Nunes, Rafael Padilha, Morris Sharp, Bruno Silva, Swati Sharma, Vijay Aski, and Ranveer Chandra. Rag vs fine-tuning: Pipelines, tradeoffs, and a case study on agriculture, 2024.

[Brown *et al.*, 2020] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[Bubeck *et al.*, 2023] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.

[Carvalho *et al.*, 2022] Gonçalo Carvalho, Nádia Medeiros, Henrique Madeira, and Bruno Cabral. A functional fmeca approach for the assessment of critical infrastructure resilience. In *2022 IEEE 22nd International Conference on Software Quality, Reliability and Security (QRS)*, pages 672–681, 2022.

[Holtzman *et al.*, 2020] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2020.

[Jiang *et al.*, 2024] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024.

[Lin, 2004] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[Liu *et al.*, 2022] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? In Eneko Agirre, Marianna Apidianaki, and Ivan Vulić, editors, *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online, May 2022. Association for Computational Linguistics.

[Nori *et al.*, 2023] Harsha Nori, Yin Tat Lee, Sheng Zhang, Dean Carignan, Richard Edgar, Nicolo Fusi, Nicholas King, Jonathan Larson, Yuanzhi Li, Weishung Liu, Renqian Luo, Scott Mayer McKinney, Robert Osazuwa Ness, Hoifung Poon, Tao Qin, Naoto Usuyama, Chris White, and Eric Horvitz. Can generalist foundation models outcompete special-purpose tuning? case study in medicine, 2023.

[Rausand and Høyland, 2004] Marvin Rausand and Arnljot Høyland. *System Reliability Theory: Models, Statistical Methods, and Applications (2nd ed.)*. Wiley, 2004.

[Sharma and Srivastava, 2018] Kapil Dev Sharma and Shobhit Srivastava. Failure mode and effect analysis (fmea) implementation: a literature review. *J Adv Res Aeronaut Space Sci*, 5(1-2):1–17, 2018.

[Tay *et al.*, 2023] Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Siamak Shakeri, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. Ul2: Unifying language learning paradigms, 2023.

[Touvron *et al.*, 2023] Hugo Touvron, Louis Martin, and et al. Llama 2: Open foundation and fine-tuned chat models, 2023.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[Wang *et al.*, 2023] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023.