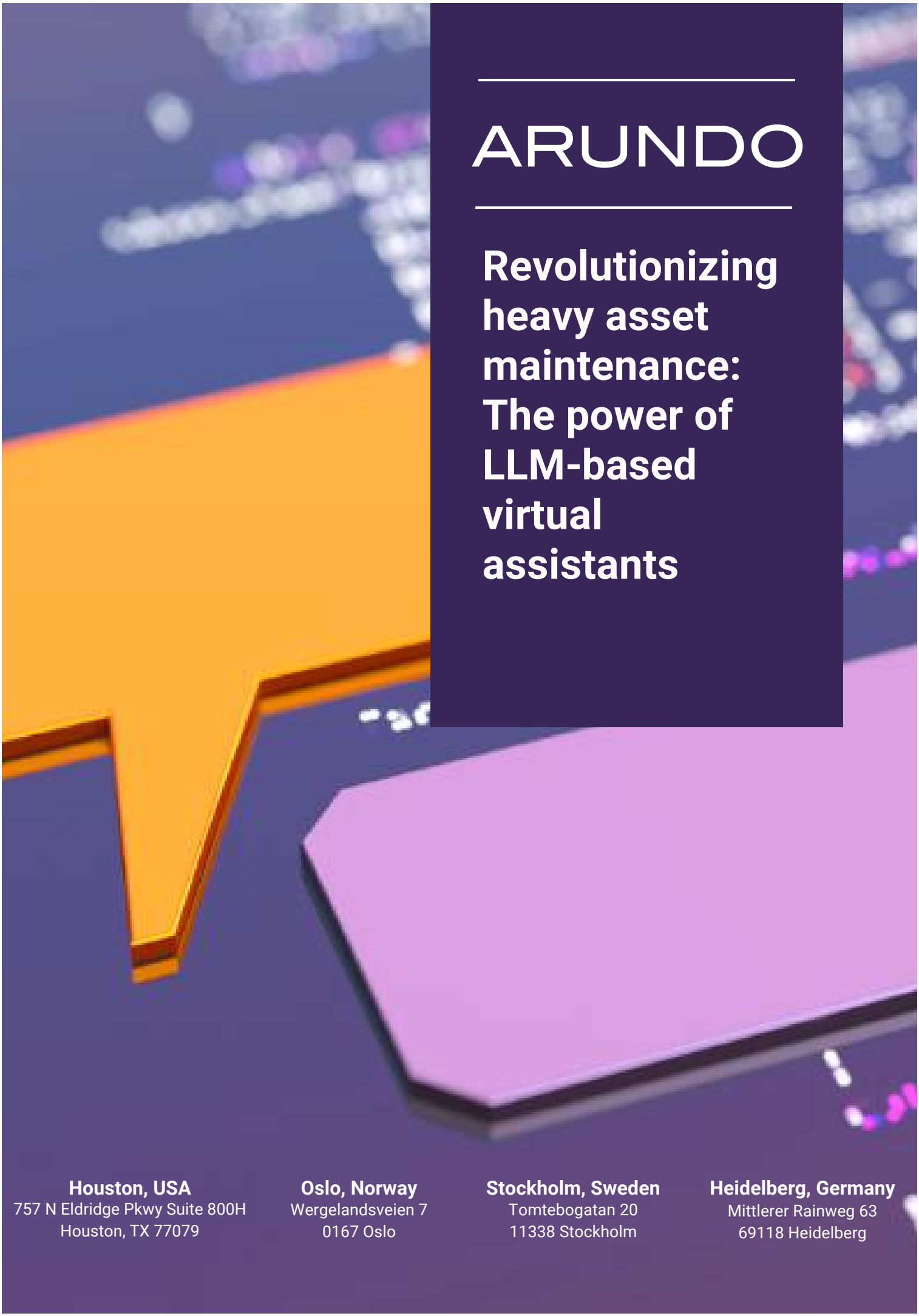


ARUNDO

Revolutionizing heavy asset maintenance: The power of LLM-based virtual assistants

Houston, USA
757 N Eldridge Pkwy Suite 800H
Houston, TX 77079

Oslo, Norway
Wergelandsveien 7
0167 Oslo

Stockholm, Sweden
Tomtebogatan 20
11338 Stockholm

Heidelberg, Germany
Mittlerer Rainweg 63
69118 Heidelberg

CONTENTS

Summary ____ 03

Question Answering ____ 05

Information Retrieval ____ 08

Improving the results: Augmenting base models with domain-specific data ____ 11

Retrieval Augmented Generation ____ 12

Fine-tuning ____ 12

Conclusion: Empowering maintenance engineers with “virtual maintenance assistants” is easier than expected ____ 14

Appendix ____ 15

AUTHORS

Lars Bjålie, PhD, Senior Data Scientist at Arundo Analytics. At Arundo, he has worked primarily on process optimization in the chemical industry. He previously held a position as a consultant at BearingPoint, where he worked as the lead data scientist within predictive maintenance.

Emre Kucuk, PhD, Data Scientist at Arundo Analytics. He works on developing and implementing industrial data science applications, which encompass anomaly detection for equipment monitoring. He previously worked on predictive maintenance projects at AVL and Fujikura.

Alessandro Arlandini, PhD, Data Scientists at Arundo Analytics. He focuses on designing and developing data science solutions for industrial applications, on topics such as predictive maintenance and asset optimization. Previously, he worked on machine learning solutions for energy optimization with IoT devices as a data scientist at WattWorker.

The authors would like to thank our customer Flex LNG for contributing the maintenance logs used in this study. Flex LNG is a shipping company with a fleet of 13 LNG carriers that are continuously maintained. Without the detailed documentation and logging of these maintenance interventions, this study would not have been possible.



SUMMARY

Part of a maintenance engineer's job is to diagnose equipment issues and perform root-cause analysis to take the correct action and prevent the issue from happening again.

Naturally, this work is often performed where the equipment is found, and has to be done quickly to avoid prolonged downtime.

It can both be cumbersome and take too long to sift through old maintenance logs and equipment manuals, which could help pinpoint the root cause.

A "virtual maintenance assistant" based on a large language model (LLM) could make this knowledge much more accessible, both in the field and in the back office. In general, virtual assistants are a very promising area of generative AI, which will likely be popularized in many different fields (and already have in, for example, management consulting and programming).

In this article, we give a brief overview of how such a virtual assistant could become a reality in the maintenance domain.

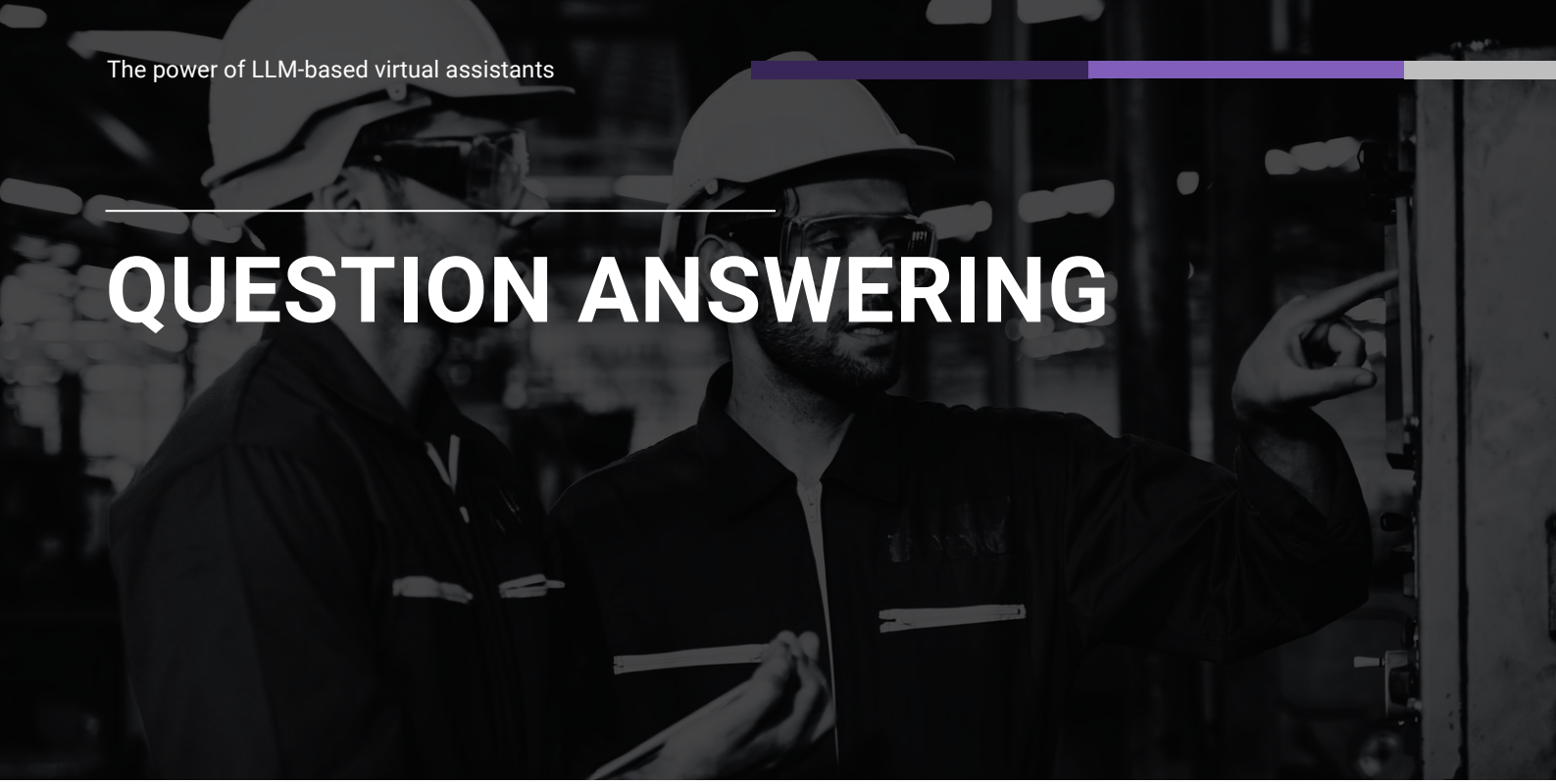
We will start by exploring two essential tasks: **Question Answering** and **Information Retrieval**. In Question Answering, the base LLM (e.g., an LLM not changed in any way from the original made by its developers) is used to directly answer maintenance questions, generating free-form responses.

For Information Retrieval, we use prompt engineering to instruct the model on how to characterize maintenance logs to retrieve the information we are looking for. We give concrete examples for both tasks using maintenance logs contributed by an operator of commercial tanker vessels, and share some thoughts on the practical implementation of these approaches.

Lastly, we also discuss various methods to improve the model outputs beyond prompting a base LLM: Retrieval Augmented Generation (RAG) allows the model to search through external data sources for answers, while fine-tuning changes the model parameters to "teach" the model itself domain knowledge.

All examples in this article are generated using Meta's LLaMA 2 model (the 7 billion parameter version), which is released under an open source and free for commercial use license. Since this model has a relatively low number of parameters (truly large models have over 100 billion parameters), it can be run on a local laptop, with each query typically taking around 10-20 seconds to complete. For many use-cases, such a "small" model is good enough, so expensive hardware is not necessarily required to run models locally.





QUESTION ANSWERING

A natural starting point in the development journey of a virtual maintenance assistant is to see how well a base LLM performs in Question Answering - simply asking the model a question for help in resolving a maintenance issue. Language models can be used to obtain information about a topic by asking questions, including tips on how to fix some equipment problem – if the text the model has been trained on contains information relevant to the question, the model will likely be able to answer.

However, when doing so, it's important to keep in mind that LLMs are language transformers producing sentence-like structures mimicking meaningful language; they are not independent agents with insight and comprehension. This is, in a nutshell, why LLMs are known to "hallucinate", confidently producing text that contains false knowledge. Where they truly shine is parsing and searching information.

Regardless, if the text the model has been trained on contains the knowledge we are looking for, they can still produce very useful answers, even though they have to be carefully evaluated. As an example, we will try a question from the domain of liquid natural gas (LNG) shipping, and dissect the answer using our own understanding.

In this case, we simply ask an open-ended question, and the language model generates a free-form reply:

Prompt:

```
"The IGG/GCU unit of my LNG tanker
has a problem. I have found cracks
in the internal cooling jacket. What
are possible causes of this problem
and how can I fix it?"
```

Response:

Cracks in the internal cooling jacket of an LNG tanker's IGG/GCU unit can be caused by various factors, including:

1. Overpressure or over-temperature: If the tank is exposed to high pressures or temperatures, it can cause the metal to expand and contract excessively, leading to cracking.
2. Poor design or manufacturing: If the jacket was not designed or built properly, it may be more susceptible to cracking under stress.
3. Corrosion: Corrosion of the jacket's material can weaken its structure and cause cracks.
4. Lack of maintenance: Failure to perform regular inspections and maintenance on the IGG/GCU unit can lead to neglected issues that can eventually result in cracking.

To fix the problem, you will need to address the root cause of the cracking. Here are some steps you can take:

1. Inspect the jacket thoroughly: Conduct a thorough inspection of the internal cooling jacket to identify the location and extent of the cracks.
2. Determine the cause: Based on the inspection results, determine the cause of the cracking. It may be due to overpressure or over-temperature, poor design or manufacturing, corrosion, or lack of maintenance.
3. Repair or replace the jacket: If the cracking is severe, you may need to repair or replace the internal cooling jacket entirely. This will depend on the extent of the damage and the age of the tanker.
4. Implement preventive measures: To prevent similar issues in the future, implement preventive measures such as regular inspections, proper maintenance schedules, and ensuring that the tanker operates within safe pressure and temperature ranges.
5. Consult experts: If you are unsure of how to proceed or need additional guidance, consult with experts in the field of LNG tanker engineering and maintenance.

Remember, safety should always be your top priority when dealing with any issue related to an LNG tanker's IGG/GCU unit. Ensure that all repairs and maintenance are carried out by qualified professionals using approved methods and materials.

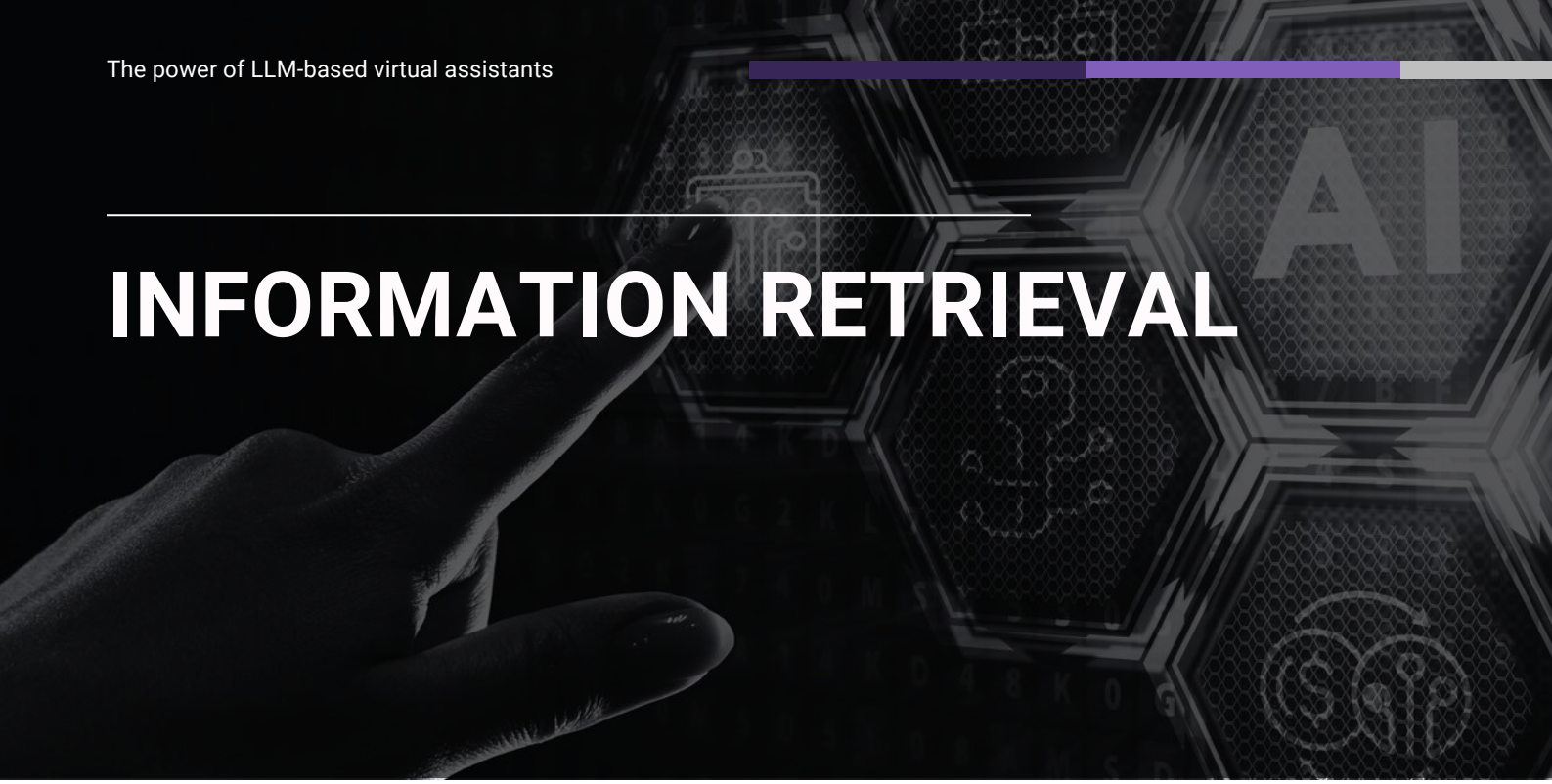
= output not precise/satisfactory

While the response is decent, it is not entirely satisfactory. It should have mentioned vibrations as a potential cause, mostly caused by unstable combustion or misadjusted start-up and shut-down sequences. Parts of it are also potentially misleading or imprecise (as highlighted on the previous page):

- Cause 1. correctly mentions excessive temperatures, but pressure is not typically a factor in IGG/GCU units.
- The recommended step 3. for fixing the issue is described too generically: the most common way to repair a cracked cooling jacket is by welding.
- Similarly, step 4. is correct, but it would be more precise if it explicitly mentioned tuning the combustion process (by tweaking the air-gas mix or the start-up and shut-down sequences) as one of the preventive measures.

All in all, the LLM behaves as expected, with some answers that are too generic, some producing false information, and some typical maintenance steps missing. If necessary, the user could also obtain further information by asking the model to elaborate on some points.

To improve the answers while still using the base LLM only, we could explore how to best structure the prompt – including precise asset information and problem description is likely to be helpful. We can imagine our virtual maintenance assistant application helping the user standardize their prompts into the type most likely to give desired results.



INFORMATION RETRIEVAL

Next, a slightly more complicated task for our assistant is to search historic maintenance logs to filter out events where some asset had a fault or another status - a form of sentiment analysis. This involves harnessing the base LLM's language proficiency to comprehend log content, presenting a substantial improvement over conventional keyword searches, which would give all events where the asset was mentioned (e.g., as part of a routine inspection). Even if the maintenance logging system allows for marking individual events as faults, this cannot be fully relied upon given the extremely diverse situations happening in the field. For example, when filing a report on the thorough inspection of a complex asset, a “fault yes/no” label may cascade to all involved assets, which rarely represents the actual situation.

To instruct the LLM more precisely, we construct each Information Retrieval prompt to first include some examples of successful task execution. This is a type of “few-shot learning”, where we give the model a few input/output pairs to learn from during execution, a very useful technique to improve results. Since the model mainly searches through factual data, the results are less likely to include hallucinations, as discussed in our Question Answering example.

In the general prompt, along with the example input/output pairs, we also give a step-by-step procedure to arrive at the desired result. Expressing the task as a sequence of “atomic” steps is generally a good practice for two main reasons:

- While LLMs are very good at operations such as information extraction, summarization, and inference, they tend to perform better when doing these in isolation, with the performance degrading when executing multiple operations at once.
- By checking the “intermediate” outputs, we can better understand how the LLM arrived at that specific result, and how to improve it if necessary.

For the following examples, we start from a dataset of maintenance logs for a Gas Combustion Unit (GCU) on an LNG tanker. The task is to point out which logs highlight a problem with the cooling jacket of the GCU.

General instruction prompt, included first in all examples:

```
You will be given an input between angle brackets, your job is to find if
the input mentions problems with the IGG/GCU cooling jacket.

For each input, do the following:
 1. Output a list of all the asset types mentioned in the input
 2. Check if cooling jacket is in the list. If it is not, stop here. Only if
    it is, do the next step
 3. Evaluate the final status of the cooling jacket, one of
    ['faulty','operational','repaired']

Example Input:
<Lance replaced by old type Ignitor. Operational.>

Example Output:
Mentioned assets: Lance, Ignitor
Cooling jacket mentioned: no

Example Input:
<Annual maintenance of combustion chamber done. Spray nozzles
cleaned, in good condition, safety devices and pressure gauges
checked. Main Burner is operational and the diesel oil nozzle is
tight. Second igniter is operational, first Igniter is not
operational. Demister was inspected. Fresh water rinsing sequence
is operational. From end of Jan 15 till mid of Marc 18 complete
reinforcement was done on IGG/GCU. New deflector was welded,
flame throat was replaced by new one, reinforcement of complete
unit was done. Great number of cooling jacket cracks were welded.
After replacing of burner throat the condition has been
improved-vibration and noise is on acceptable level>

Example Output:
Mentioned assets: combustion chamber, spray nozzles, main burner, diesel oil
nozzle, Igniter, Igniter, Demister, IGG/GCU, Deflector, Flame throat,
Cooling jacket, Burner throat
Cooling jacket mentioned: yes
Sentiment: repaired
```

```
Example Input:
<IGG/GCU burner throat has been replaced according to instruction manual.
Jacket tested by the sea water for cracks.>

Example Output:
Mentioned assets: IGG/GCU burner throat, cooling jacket
Cooling jacket mentioned: yes
Sentiment: operational
```

The Appendix contains five examples we fed into the model prompt together with the instructions.

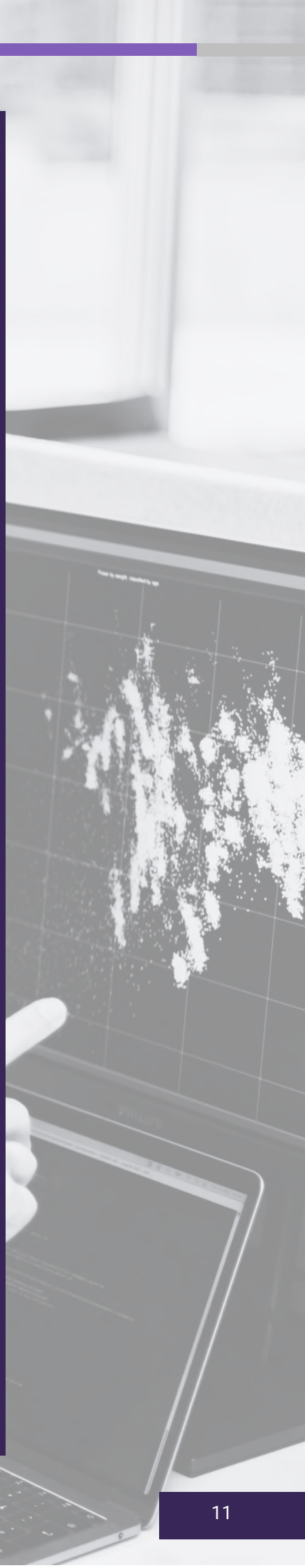
The LLM performed the task successfully: In all examples, it managed to understand whether the cooling jacket is mentioned or not, and its status (“sentiment”) according to the reported operation. For one of the longer logs (example 4), it also explained the output, without being explicitly instructed to do so. The output is still not perfect, as it can be argued that the sentiment of the log in example 5 should be “repaired” and not “faulty”.

In general, the LLM managed to extract a large amount of information from the various input examples and summarize it in a format faster to analyze by a human than the raw logs. One could imagine our virtual assistant application scanning through many such historic log events simultaneously, picking out those with a certain status for a given asset to help the maintenance engineer quickly learn from past work.

IMPROVING THE RESULTS: AUGMENTING BASE MODELS WITH DOMAIN-SPECIFIC DATA

Although the initial results are encouraging, the performance of a base LLM is likely not good enough to be a useful maintenance assistant by itself in real-world scenarios. Luckily, there are several ways to improve the results by augmenting base models with domain-specific data, and there are open-source frameworks like LlamaIndex available to facilitate this process.

Before discussing model augmentation, let's first think about the pros and cons of simply using a larger model – the model we have used for our examples is, after all, relatively small (few parameters). Larger models tend to be trained on larger datasets, giving them the chance to learn from more knowledge (such as highly specific equipment manuals). However, even the largest models are not trained on any internal documentation or maintenance logs, and larger models also come with increased costs in terms of resources and inference time. As for smaller models, they can still hallucinate responses that seem realistic, which is a clear drawback for this kind of "virtual assistant" use-case. Better to not get any response than a misleading one that could lead to wasted time going down a rabbit hole! For these reasons, while using a larger model is a viable strategy to try to improve performance, it is not necessarily a recommended one.



Retrieval Augmented Generation

The effectiveness of how we included more information in the prompt in the Information Retrieval task gives us the idea of organizing all the relevant manuals (both from the OEM and internal best practices) and logs for the asset into a single body of text, which we feed into the prompt of the model together with the actual question. Then, in theory, the model should have access to all the knowledge it needs to give accurate answers.

However, LLMs have a limited number of words they can "remember" (also known as the "context window"), setting a limit on the prompt length. Luckily, this has only been trending larger, with OpenAI's GPT-4 having a 32.000 token version (which means around 24.000 words). This is likely still insufficient for a complete set of documentation for all assets, so in the near future we still need to do some filtering of what to include in the context window.

A naive approach to overcome this would be to filter documents only by asset type (assuming such a label exists), but we want our model to be aware of interactions between assets - sometimes, *another* asset malfunctioning is the root cause. A better approach is a method called Retrieval Augmented Generation (RAG) to first search across all documents for relevancy, and then include the most relevant documents as context in the prompt.

For use-cases involving question/answering on organizational data, RAG is likely a more fruitful approach than fine-tuning (updating the internal parameters of the model themselves, adapting to new text sources), as it gives continuous access to an organization's changing database of internal documents and offers transparency in the answering process (which documents were used in the answer). For completeness and comparison, we briefly describe fine-tuning and how it could be used for our purposes in the next section.

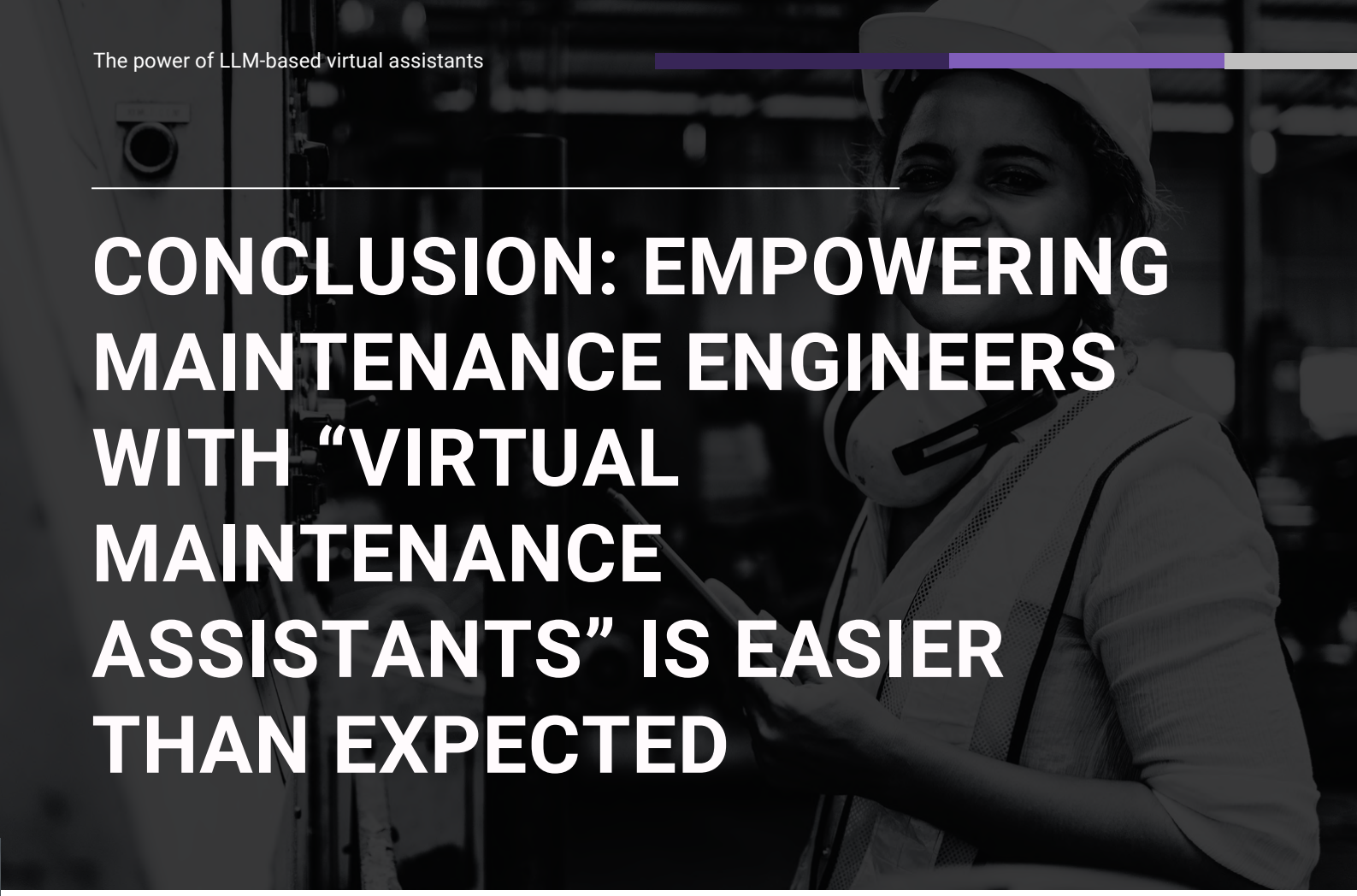
Fine-tuning

Fine-tuning is a more involved process than RAG, as it requires changing the internal parameters of the model. Previously, this was typically limited to smaller models (fewer parameters), or it became too computationally expensive to be practical, but recent developments have opened up for fine-tuning even the largest language models, such as OpenAI's GPT-3.5 Turbo, which has 175 billion parameters (for comparison, the LLaMA 2 model we use for the examples in this article has 7 billion parameters). Beyond potentially giving more accurate answers, the response time of the model should also go down, as the time it takes for the model to generate a response is proportional to the amount of text in the context window – if the model is fine-tuned we don't need to include large chunks of information or input/output examples in the prompt.

There are three main sources of data for fine-tuning a model:

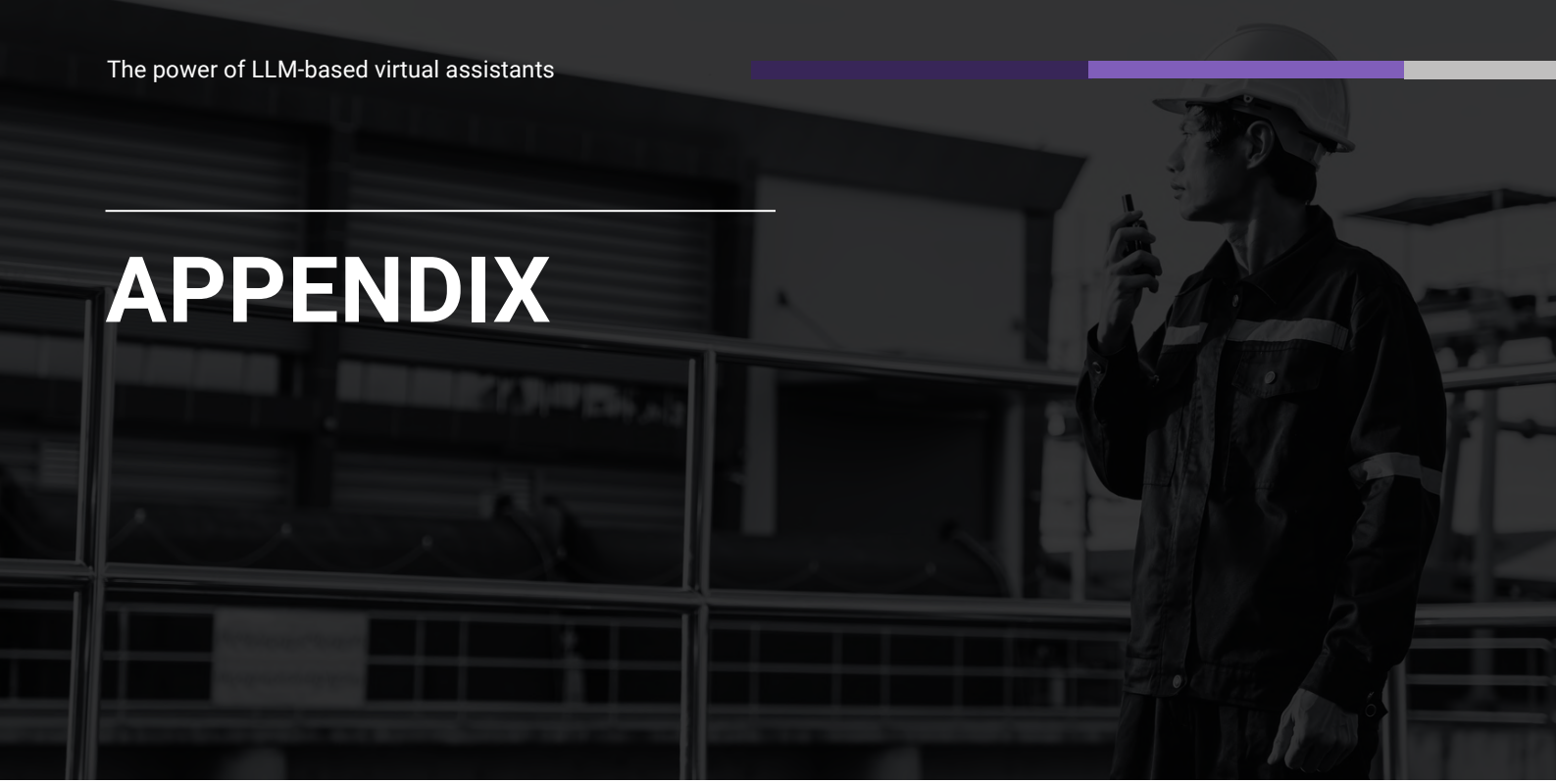
1. The “raw” documents themselves.
2. Examples that demonstrate the desired behavior of the model.
3. Collecting human feedback on responses.

In practice, the data for 1. is readily available from equipment manuals and maintenance logs, whereas for 2. a dataset has to be manually constructed, making it a somewhat more cumbersome approach. On the other hand, for source 3. we could imagine building it into the virtual assistant itself, allowing users to give real-time feedback. This would establish a feedback loop where the continued use of the virtual assistant trains the language model and improves its performance, a technique called Reinforcement Learning from Human Feedback (RLHF).



CONCLUSION: EMPOWERING MAINTENANCE ENGINEERS WITH “VIRTUAL MAINTENANCE ASSISTANTS” IS EASIER THAN EXPECTED

Maintenance engineers often need to quickly diagnose equipment problems and find root causes to prevent future issues, but sifting through old maintenance logs and manuals can be time-consuming or simply impossible out in the field. A "virtual maintenance assistant" powered by large language models (LLMs) can streamline this process, and can be developed more easily than one may expect. As we saw from the Question Answering and Information Retrieval examples, a base LLM by itself is close to being good enough to be useful in the field, and there exists readily available techniques to give the model access to further domain knowledge to increase the quality of the responses.



APPENDIX

Example 1:

```
Input: <Solenoid valve is replaced.>
------------------------
Mentioned assets: Solenoid valve
Cooling jacket mentioned: no
```

Example 2:

```
Input: <GCU Capacity Control Valve Positioner for Blower 2.1 was replaced by
new. Positioner tested and calibrated . Fully operational.>
------------------------
Mentioned assets: GCU Capacity Control Valve Positioner, Blower 2.1
Cooling jacket mentioned: no
```

Example 3:

```
Input: <More details attached in Technical systemIn an attempt to
troubleshoot the issue, we have done the following:Checked and overhauled the
capacity control valves - specifically, 3-8. Replaced the positioner on 3-8
with a new one.Checked and calibrated all positioners. Confirmed all capacity
control valves are functioning correctly. Checked the water seal pressure
transmitter and confirmed it is working correctly.Verified that the Outlet
Gas/IGG FE is functioning as expected.Tested all blowers and capacity control
valves separately in manual mode, all of which work well when started
individually.Interestingly, it appears that before we calibrated the capacity
control valve positioners on 3-7 and 3-8, they may have been intentionally
calibrated incorrectly. After few attempts of different tests we have found
the problem it is related to blow off valve BV 4 -GP =35% for starting and
initial firing . So almost all the time the pressure was abt 220-240 mbar -it
```

means back pressure to high when 3 bowers started . Once the valve was calibrated in the same 35 % position pressure changed to 110-140 mbar and blowers capacity starts working normal and firing without problem . All back smoothly operation GCU /HICOM either Gas Blowers .>

Mentioned assets: Capacity control valves, positioner, water seal pressure transmitter, Outlet Gas/IGG FE, blowers
Cooling jacket mentioned: no

Example 4:

Input: <IGG/GCU modification as per OEM completed. All welds made by the crew have been cut and welded by the certified welder. Middle part of the cooling jacket (outer) has been reinforced by additional doubler plates with eight brackets. In the bottom part of the GCU cooling water outlet rim has been rebuilt, broken demister support welded. Top of the GCU: Additional seven knee brackets and bolted doublers welded. Internal jacket wall (top cone) welded due to 160mm crack detected. SW level sensor pocket welded due to leak during water pressure test. All windows welded back on place. DYE test carried out for all welded cracks. IGG/GCU 25 hours test carried out without strictures. GCU burner has been lifted and inspected. Flame chamber weldings found in good condition, water test carried out - no leaks. Burner throttle - shape of wavy collar corrected and installed back with 8 spots as per recommendation. All scrubber spaces inspected IGG/GCU test carried out on 2 Feb 2021 with positive result. Full report in the attachment. Due to limited size of attachments enclosed report is without photos.>

Mentioned assets: IGG/GCU, cooling jacket, burner throat, demister, SW level sensor pocket, windows, scrubber spaces
Cooling jacket mentioned: yes
Sentiment: repaired

Based on the information provided in the input, the following is the evaluation of the IGG/GCU cooling jacket:

* The cooling jacket has been reinforced with additional doubler plates and knee brackets.
* The middle part of the cooling jacket (outer) has been reinforced.
* The bottom part of the GCU cooling water outlet rim has been rebuilt, and a broken demister support has been welded.
* The top of the GCU has additional seven knee brackets and bolted doublers welded.
* The internal jacket wall (top cone) has been welded due to a 160mm crack detected.

```
* The SW level sensor pocket has been welded due to a leak during water
pressure test.
* All windows have been welded back in place.
* A DYE test has been carried out for all welded cracks.
* The IGG/GCU has undergone a 25-hour test without strictures.
* The GCU burner has been lifted and inspected, and the flame chamber
weldings have been found in good condition, with water test carried out - no
leaks.
* The burner throttle has been corrected and installed back with eight spots
as per recommendation.
* All scrubber spaces have been inspected, and the IGG/GCU test carried out
on 2 Feb 2021 has been positive.

Based on this information, the sentiment of the cooling jacket is "repaired".
```

Example 5:

```
Input: <IGG/GCU SCRUBBER UNIT - OUTER WALL OF COOLING WATER JACKET WELDED DUE
TO CRACK. SCRUBBER WET FILTERS SUPPORTS WELDED DUE TO CRACKS. GC D51 ISSUED.
HOT WORK PERMIT FOLLOWED, RA APPROVED BY TSI.>
------------------------
Mentioned assets: IGG/GCU scrubber unit, cooling water jacket, GC D51, hot
work permit, RA approved by TSI
Cooling jacket mentioned: yes
Sentiment: faulty
```

Example 6:

```
Your input: <GCU internal inspection carried out. all additional supports
installed in march found without cracks. Wet filter and demister found in
good condition. Black spots from fuel found inside furnace. Top cover with
Burner was removed for inspection. Burner dismantled in workshop, nozzle cup
nut found loosen, nozzle was cleaned and burner assembled. Elastic hoses in
good condition. Air diffuser found deformed with broken of joints. Diffuser
was straightened in workshop. All air solenoid valves were tested, all
working properly. SW supply tested, Jacket cooling water pressure in good
range: 0,7-0,9 bar>
------------------------
Mentioned assets: GCU, internal inspection, additional supports, wet filter,
demister, furnace, top cover, burner, nozzle cup nut, elastic hoses, air
diffuser, air solenoid valves, SW supply, jacket cooling water pressure.
Cooling jacket mentioned: yes
Sentiment: operational
```

ABOUT ARUNDO

Arundo Analytics helps equipment operators to make better decisions and improve efficiency, reliability, and sustainability.

Our products link data science, cloud computing, and AI with industrial IoT, providing prescriptive actions beyond simple alerts. Broad industry experience makes us a powerful partner for digitalization and process optimization.

Maximize your business' capabilities with Arundo products.

Benefit from more than 250.000 development hours for:

- increased production levels,
- minimized downtime,
- enhanced reliability, and
- significant cost savings for your operations.

Get the answers you need:

contact@arundo.com

Torvald Klaveness · MACGREGOR · evolution WELL SERVICES · INEOS Inovyn · DOLPHIN DRILLING · FLEX LNG