

Industrial-Grade Smart Troubleshooting through Causal Technical Language Processing: a Proof of Concept

Alexandre Trilla

Alstom, Santa Perpètua de la Mogoda
La Salle, Universitat Ramon Llull
Barcelona, Spain
alexandre.trilla@alstomgroup.com

Nenad Mijatovic

Alstom, Saint Ouen
Paris, France
nenad.mijatovic@alstomgroup.com

Ossee Yiboe

Alstom, Saint Ouen
Paris, France
ossee-josepha-charlesia.yiboe@alstomgroup.com

Jordi Vitrià

Universitat de Barcelona
Barcelona, Spain
jordi.vitria@ub.edu

ABSTRACT

This paper describes the development of a causal diagnosis approach for troubleshooting an industrial environment on the basis of the technical language expressed in Return on Experience records. The proposed method leverages the vectorized linguistic knowledge contained in the distributed representation of a Large Language Model, and the causal associations entailed by the embedded failure modes and mechanisms of the industrial assets. The paper presents the elementary but essential concepts of the solution, which is conceived as a causality-aware retrieval augmented generation system, and illustrates them experimentally on a real-world Predictive Maintenance setting. Finally, it discusses avenues of improvement for the maturity of the utilized causal technology to meet the robustness challenges of increasingly complex scenarios in the industry.

ACM Reference Format:

Alexandre Trilla, Ossee Yiboe, Nenad Mijatovic, and Jordi Vitrià. 2024. Industrial-Grade Smart Troubleshooting through Causal Technical Language Processing: a Proof of Concept. In *Proceedings of Causal Inference and Machine Learning in Practice (KDD 2024 Workshop)*. ACM, New York, NY, USA, 7 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 INTRODUCTION

The degradation of industrial assets is a complex multifaceted problem that can be explained by different factors. As the components wear and deteriorate, the systems exhibit a series of changes that increase in severity until they eventually fail. In this case, failure patterns may also emerge. For instance, in the Reliability Engineering field, assets are most expected to fail either prematurely (early) during their break-in period, or late by the end of their remaining useful life (wear-out) [10]. These failure types can be *anticipated* because their modes, mechanisms, and effects, are well known and

documented. In consequence, engineers introduce quality checks in the manufacturing process and inspection actions in their (more-or-less conservative) preventive maintenance schedule to mitigate their impact. However, for as long as the machines operate, failures may seem to appear “randomly” at any point in time. This is especially challenging for dependable assets while they transit the middle risky region, when the failure rate is relatively low, but uniform/constant.

In this uncertain setting, the field of Predictive Maintenance tackles the problem by introducing the data as a means to closely follow the actual evolution of each asset and make better informed and timely decisions [12]. In this sense, the detection of incipient anomalous behaviors and the capacity to *diagnose* their *root causes* and *predict* their *solutions* towards a more favorable *prognosis* become increasingly important to guarantee the availability of the machines.

To succeed in these multiple objectives, the required information and knowledge, which displays a clear *causal* character, is typically described and compiled in textual form through two different (linguistic) environments [7]. On the one hand, an *ontological* reference framework based on a Failure Mode, Mechanism, and Effect Analysis (FMMEA) [15], which provides a scholarly structure of causality driven by degradation. On the other hand, a *methodological/epistemological* approach via an actual record on Return On Experience (RoX), the data of which have been explicitly written for the purpose of explaining both the root causes and solutions of the reported failures. In both environments, several experts inherently identify which properties of the observations describe spurious correlations unrelated to the causal explanation of interest, and which properties represent the phenomenon of interest, i.e., the stable invariant associations.

Traditional approaches for processing language in Predictive Maintenance settings have initially considered the idiosyncrasies of technical environments [6], and have evolved into exploiting Large Language Models [23], ontologies [43], and extracting recurrent problems and frequently suggested solutions [35]. Almost concurrently, the community of natural language processing and computational linguistics identified causal challenges in textual data [11], and these were soon also considered in the technical domains as a means to explain the degradation mechanisms by

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD 2024 Workshop, August 26, 2024, Barcelona, Spain

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-XXXX-X/18/06

<https://doi.org/XXXXXXX.XXXXXXX>

developing custom word embeddings [37] and unconfounded sub-system structures [38].

This workshop paper holds the hypothesis that root causes and solutions can be learned directly from the textual expressions found in field-specific RoX data, which are fully aligned with the Smart Troubleshooting objective: given the description of a problem, the system shall be able to accurately determine the related root cause and solution. The paper is organized as follows: Section 2 reviews the modeling fundamentals of Causality and Language, Section 3 describes the proposed causal diagnosis method from the standpoint of standard Predictive Maintenance, Section 4 illustrates the implementation of the method on a specific illustrative example, Section 5 discusses avenues of improvement to increase the robustness of the approach, and Section 6 concludes the work.

2 BACKGROUND

This section presents the basic notions on how to model the environment under analysis, both from a causal and a linguistic perspective.

2.1 Structural Causal Model

The causal links among the variables X that build the model of a system are assumed to be most effectively represented using the tools from the field of Causality. In this sense, the Structural Causal Model (SCM) is the framework that can most generally capture such directed associations [31]. The SCM defines a set of assignments governing their specific functional associations f , along with some independent exogenous noise N that accounts for everything that is not explicitly included in the model:

$$X_j := f_j(PA_j, N_j), \quad (1)$$

where PA_j represents the direct causes of the X_j variable.

If enough knowledge and experience from the field is available from the subject matter experts, e.g., through the FMMEA or RoX structures, then a complete SCM graph may be developed right from the start. Otherwise, the data need to be carefully leveraged to drive the discovery of the causal model.

2.1.1 Causal Bayesian Network. Once the structural graph that binds the variables is determined, the functional associations of the SCM may be learned, and this work specifically adopts a stochastic interpretation of the world. Therefore, it treats all X as random variables, and the resulting SCM statistically describes their (conditional) probability distributions.

Considering n random variables X_1, X_2, \dots, X_n and a directed acyclic graph that relates them causally, a Causal Bayesian Network (CBN) is a generative model that has the following factorized joint probability distribution:

$$P(X_1, \dots, X_n) = \prod_{j=1}^n P(X_j | PA_j, N_j). \quad (2)$$

The graphical nature of Bayesian networks allows seeing relationships among different variables, and their conditional dependencies enable performing probabilistic inference [1]. In particular, CBN are powerful tools for knowledge representation and inference under *uncertainty* [33].

2.2 Causal Inference

Beyond probabilistic inference, Causal Inference provides the tools that allow estimating causal conclusions from observational data, i.e., in the absence of a true experiment, given that certain assumptions are fulfilled. These assumptions increase in strength as is defined in Pearl’s Causal Hierarchy (PCH) abstraction [5], which is summarized as follows for the purposes of this paper.

2.2.1 PCH Rung 1: Associational. Describes the observational distribution of the factual data through their joint probability function $P(X)$. From this point forward, interesting quantities, i.e., the queries X_Q , can be directly computed given some evidence X_E through their conditional probability, which is computed as a ratio of marginals:

$$P(X_Q | X_E) = \frac{P(X_Q, X_E)}{P(X_E)}. \quad (3)$$

This level of analysis displays a degree sophistication akin to classical (un)supervised Machine Learning techniques. As such, it is subject to *confounding bias*, where common causes may induce spurious statistical associations/correlations [34].

2.2.2 PCH Rung 2: Interventional. Describes an actionable distribution, which endows causal information at the population level. This level of analysis can be achieved through actual experimentation via Randomized Control Trials, or through statistical adjustments that smartly combine the observed conditional probabilities to reduce the spurious associations in the estimation. Pearl’s *do*-calculus is likely to be the most effective approach to determine the identifiability of causal effects by applying the following three rules: 1) insertion/deletion of observations, 2) action/observation exchange, and 3) insertion/deletion of actions [30].

2.3 Language Model

Finally, to operate with textual data, there is the need to numerically represent linguistic information in the former “generic” variables X . To this end, Probabilistic Language Models are functions that assign a probability to a sentence, to eventually build up a whole piece of text. Traditionally, in such statistical models the sentences have been broken down, i.e., tokenized, into sequences of words, and the goal has been to predict the probability of an upcoming word [18, 26]:

$$P(X_{n+1} | X_0, \dots, X_{n-1}, X_n). \quad (4)$$

Today, with the advent of distributed representations of words and phrases [28], along with the Transformer neural architecture [39], long texts are directly represented in dense vector spaces, and the task of the resulting Large Language Models is now to provide responses to carefully engineered input prompts [8].

3 METHOD

This section details the Smart Troubleshooting objectives and the analysis procedure to attain them, which focuses on providing root cause diagnostics and predictions of solutions for a problem observation based on written text data. Since the applied industrial environment belongs to the area of Predictive Maintenance, the consideration of a common development standard such as the ISO 13374 is recommended [16]. This specification breaks down the complexity of a problem into small modules that may be developed

in isolation, thus increasing the chances of project success while also improving the interpretability and explainability of the technical solution, and help to reduce the technical debt. What follows is a description of the Data Manipulation and Health Assessment processing blocks.

3.1 Data Manipulation

Causality is an emergent property of complex industrial systems [44]. In this setting, linguistic variables constitute high level qualitative descriptions that group functions into categories and hierarchies, as is established by the FMMEA documentation.

3.1.1 Return On Experience Records. In the Smart Troubleshooting setting, the RoX text data are collected as a means to capture and describe the factual ontological relationships observed in the field [37]. They display the following variable type structure:

- Subsystem Z (common context): Categorical
- Root Cause C (problem source): Categorical
- Observation O (reported problem, failure): Text
- Solution S (repair/maintenance action): Text

This data structure is populated from several projects or environments, which exhibit some differences regarding the verbosity of the language used to describe the problem and its solution.

3.1.2 Textual Entailment. The concept of entailment refers to the directional nexus between text fragments. Regarding the RoX data, these assumed relations are encoded in the following graph:

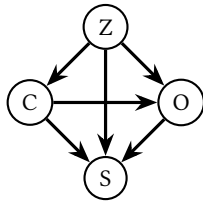


Figure 1: Graph showing the RoX variable relationships.

The diagram shown in Figure 1 explicitly states that:

- (C) is the root cause of the observed problem (O).
- (S) is both the effect of the observed problem (O) and its root cause (C).
- (Z) is a general confounder, i.e., a common cause.

Once the text data for (O,S) is available in raw format it must be cleaned before doing any further processing. This involves lower-casing, punctuation removal, lemmatization/stemming, stop word filtering, etc.

3.1.3 Text Embeddings and Large Language Models. Probably one of the most challenging parts of this environment is the embedded numerical representation of the text, which is typically considered as unstructured data. The approach described in this method has been implemented using a discrete categorical representation obtained with a “BERTopic” Large Language Model [14]. The proposed strategy integrates: 1) MiniLM [40], which is a compressed version of Sentence-BERT (i.e., a Transformer-based language model at the sentence level), 2) UMAP [27], which reduces the dimensionality of

the embedded vector space, and 3) HDBSCAN [25], which clusters and quantizes the resulting low-dimensional representation.

3.2 Health Assessment

In the Smart Troubleshooting environment, one of the main challenges is dealing with the confounding bias introduced by the diversity of subsystems and components. To this end, Causal Inference techniques are utilized in the technical language processing scenario to extract relevant linguistic features from the text [11].

This module exploits the probabilistic SCM, which has been designed as a discrete category based Bayesian network following the structure of the RoX data, and conducts a fine-grained diagnosis of the observed input anomaly description by determining its root cause, and also by providing an unbiased estimation of the (most likely) potential solution.

3.2.1 Root Cause Analysis. Root Cause Analysis (RCA) is a troubleshooting method of problem solving used for identifying the sources of the failures [42]. RCA is a form of deductive inference since it requires an understanding of the underlying causal mechanisms for the potential faults and the problem, i.e., what is typically found in the context of Predictive Maintenance through the FMMEA documentation.

The discrete causal Bayesian Network is suitable for exploiting the categorized description of an observed problem (i.e., the effect) and predicting the likelihood of the several possible known causes. Thus, estimating the likely root causes amounts to computing the conditional probability diagnosis function $P(C|O)$. Note that this estimand operates on the Observational rung of the Hierarchy of Causality, see Section 2.2.1. Eventually, RCA yields a ordered list of potential root cause variables along with their probabilities, which aligns with the way complex systems fail [9]. The variables that comprise the data are required to be representative enough to help the developers and engineers pinpoint the source of the observed problems through the root causes and their effects [41].

3.2.2 Solution Generation. Predicting the solution is especially challenging due to the large cardinality of the Observation and Solution spaces (O,S). To obtain an unbiased result, an (atomic) intervention shall be performed. This represents an action $do()$ that is conducted on a system *to set* (not filter via conditioning) its variables X_i to known values x_i and then evaluate their impact/effect on other variables X_k , i.e., $P(X_k|do(X_i = x_i))$. This constitutes an advanced level of analysis that is not attainable with the observed data alone: it also needs to account for the assumptions encoded in the causal model in the form of variable dependencies. As a result, the aforementioned confounding bias in the estimation is reduced through the following adjustment formula:

$$P(S|do(O)) = \sum_{C,Z} P(C|Z) P(S|C, Z, O) P(Z). \quad (5)$$

Note that this estimand operates on the Interventional rung of the Hierarchy of Causality, see Section 2.2.2. Its computational burden can be somewhat alleviated if the single most likely Cause is already determined by the former RCA procedure.

Once the representation of the most likely Solution category (S) is reliably determined, the associated text needs to be *generated*. To this end, its related textual records S are retrieved from the dataset

and used to *prompt* a pretrained “Llama2” Large Language Model (LLM) to obtain a natural language explanation [36].

Prompt design and engineering have rapidly become essential for maximizing the potential and utility of a LLM [2]. A prompt is constructed by combining instructions, questions, input data, and examples. Prompt engineering requires a blend of domain knowledge, understanding of the AI model, and a methodical approach to tailor queries to different contexts. For Smart Troubleshooting, the following query text Q is used:

Given Observation: O, with possible root Cause(s): C, the indications for Solution used in previous similar cases using the predicted category are: S.

Beyond asking a simple question, possibly the next level of sophistication in a prompt is to include some instructions on how the LLM should answer the question:

You are an advanced smart troubleshooter assistant designed to advise experts in diagnosing and solving problems by answering questions about the possible solutions the expert should consider to fix the failure described by the Observation and Cause in the query. The smart troubleshooter should provide solutions to diagnose and solve problems.

Additionally, the troubleshooter should provide an explanation for the role of each proposal and should use appropriate forms for verbs and sentences.

The smart troubleshooter should also refrain from redundancy or repetition of steps. The smart troubleshooter always answers as helpfully as possible. It is crucial that all the propositions should always be presented using:

" - Option/Solution "
or any other listing format like this example layout:
" - Option 1 : here the text
- Option 2 : here the text... "

Additionally, the field of Causality has a privileged position in developing trustworthy intelligent systems [13]. For that reason, given that the pretrained LLM has learned from a large collection of (possibly uncontrolled) documents, it is advised to include some warning considerations (e.g., using safe bias-free language) regarding the integrity of the generated outcomes:

The smart troubleshooter should avoid harmful, unethical, racist, sexist, toxic, dangerous, or illegal content, and ensure that the responses are safe, socially unbiased, and genuinely positive. If the smart troubleshooter doesn't know the answer, they should say so. It is crucial that the smart troubleshooter never provides too specific details in their generated statements. Finally, the troubleshooter should follow the layout mentioned above for the answers and should always include any relevant information from the Observation and Cause(s) given, without mentioning their indexes or references.

Now, give the Solution to this query: Q.

Table 1: Ranking of the 5 most probable potential root causes (out of 20 categories).

Potential Root Cause	Probability
Part physically damaged	0.9012
Accident	0.0052
Incorrect maintenance	0.0052
Insufficient lubrication	0.0052
Leakage	0.0052

4 RESULTS

This section develops the experimental work through one illustrative example in the Predictive Maintenance domain. The causal model has been trained on several projects with RoX dataset sizes between 4k and 20k records, yielding average accuracy root cause classification scores over 80% (and over 70% for precision and recall).

4.1 Data Observation

The specific exemplifying instance displays the following RoX data descriptions:

- **Subsystem:** Suspension
- **Root Cause:** Part physically damaged
- **Observation:** “failure mechanical brake trailer and use electrical release kph brake”
- **Solution:** “download showed only one instance of failure on trailer with loss of comms with bcu this fault cleared when tram was put into to remove brake isolation however on tram being put back into s axle failed to apply download showed w at fault cor re ct reporting checked end switch disconnected and found verdigris on pins new loom made up and fitted fault still present so ch anged proximity switch tram te ran to bulwell and back with tct all probes alignedthrough coasting all trailer proximity swi tches and looms cleaned and checked”

The data for this record are shown for qualitative comparative purposes. In a live real-world Smart Troubleshooting setting, the maintainer or engineer shall provide the description of the observed problem only, and the system shall add value by producing the diagnosis results in terms of the root cause(s) and the likely solution(s).

4.2 Results Prediction and Retrieval

Table 1 shows the resulting distribution of root causes. Note that the correct cause leads this ranking, and the rest are given a smooth uniform value.

Similarly, Table 2 shows the resulting distribution of potential solutions. While the identification number of the Solution category is not very informative by itself, note that the cluster that it represents is likely to have many descriptive instances in the dataset.

What follows is a list of the textual statements that can be retrieved from each of the records that share the same leading potential solution category (i.e., the top one associated to the identification number 274):

Table 2: Ranking of the 5 most probable potential solutions (out of 302 categories).

Potential Solution	Probability
274	0.0565
244	0.0169
52	0.0146
90	0.0135
10	0.0104

- handover from off coming shift was to torque the hangar bolts old bolts used sourced new bolts re secured track brake torque tightened and height set
- drain oil from axle and top back up found lots of meatal on magnet when topping axle back up found that the axle has snapped bog i e needs new axle fitting
- corroded areas addressed and all corrosion removed areas coated with two coats of loctite corrosion inhibitor
- damaged cable removed and replaced with new
- fitted new outer rubbers and ended up having to fit new center rubber strip to a cab as well
- hole drilled out and re tapped bracket then fitted

4.3 Prompt-based Generation

Finally, based on the given problem Observation text, on the most likely Root Cause, and on the text retrieved from the most likely unbiased Solution, the LLM can now synthesize the following enhanced advisory:

- Solution 1: Check and Replace Bolts. As the possible cause suggested is failure of handover from the off-coming shift, which resulted in torquing the hangar bolts using old bolts, it is possible that the bolts may have been loose or damaged. Therefore, check and replace the bolts to ensure they are tightened properly to secure the suspension.
- Solution 2: Address Oil Leakage. Another possible cause mentioned is oil leakage from the axles, which can cause damage to the suspension and affect the braking system. Check for oil leakage around the axle and replace any damaged seals to prevent further damage.
- Solution 3: Replace Snapped Earth Cable. Previous observation mentions that a snapped earth cable was found, which can cause electrical issues in the braking system. Replace the snapped earth cable to ensure proper electrical connection and prevent any further failures.
- Solution 4: Address Corrosion. Corrosion can affect the suspension and braking system, leading to safety issues. Check for any signs of corrosion in the suspension and braking components and address them accordingly. Apply anti-corrosion measures, such as applying Loctite corrosion inhibitor, to prevent further damage.
- Solution 5: Adjust or Replace Fixings. Previous observation mentions that some fixings were found to be below spec, which can affect the suspension. Adjust or replace any fixings that are found to be damaged or not secure enough to ensure proper suspension.

- Solution 6: Inspect Sanding Compressor. Previous observation mentions that the sanding compressor was rebuilt, but faults were still present after testing. Inspect the compressor and its components to identify any potential issues and address them to prevent any further failures.
- Solution 7: Address Worn Cable Insulation. Previous observation mentions that the cable insulation was found worn through, which can cause electrical issues in the braking system. Address any worn-out cable insulation to ensure proper electrical connection and prevent any further failures.
- Solution 8: Replace Damaged Component. Previous observation mentions that a damaged mudflap was found, which can affect the suspension and braking performance. Replace any damaged components to ensure proper operation of the suspension and braking system.

It is important to note that these proposed solutions are based on the given information and any previous records, and they may not be exhaustive. However, most of them correctly advise a replacement action (for a part that is physically damaged).

Finally, while the way to quantitatively (i.e., objectively) evaluate the LLM-generated outcomes is still an open research question driven by correlational (i.e., not necessarily causal) scores [20], these qualitative (i.e., subjective) results suggest a reasonably promising future to help the subject matter experts troubleshoot the failures in challenging industrial settings.

5 DISCUSSION

Apparently, the quality of the text generated by the LLM seems higher than what the staff write on the RoX records: it shows more clarity, better diction, and better spelling. Nevertheless, there are no safety guarantees against hallucinations, and state-of-the-art LLMs are also subject to irrational behavior and reasoning breakdown even on simple tasks [24, 29].

Up to this point, the approach presented in this workshop paper has described the basic principles of its causal RCA and Solution Generation technology, and an initial experimental proof of concept has been shown. This early stage of maturity corresponds to a standard ISO 16290 Technology Readiness Level (TRL) between 4 and 5, because it has been validated in some real-world relevant environments [17]. This section brainstorms some avenues of improvement to increase this robustness indicator up to higher quality standards, considering the specific challenges of complex industrial environments, and to eventually demonstrate the technology in an operational environment (TRL 6–7).

5.1 Vector Database

One first idea could be to improve the granularity of the (currently discrete categorical) linguistic representation in the Causal Bayesian Network. The approach presented in Section 2.1.1 first embeds the unstructured text data into a large vector space, then it reduces the dimensionality of this real-valued numerical description, and finally it quantizes the resulting low-dimensional representation to obtain a categorical random variable. At each step, though, some information is lost due to compression, and while this is especially advantageous to decrease the complexity of the ensuing (discrete) probabilistic model, maybe it also introduces some unnecessary

limitations. Therefore, to potentially improve this situation, Hybrid Bayesian Networks may be helpful to represent the Observation and Solution texts with their original vectors (note that the Subsystem and the Root Cause variables shall retain their categorical nature). Hybrid Bayesian approaches, which are able to simultaneously model both discrete and continuous variables [3], have already enjoyed success in multivariate domains for predicting delays in operations [21], and also in the reliability assessment of large infrastructure networks [45].

In this new modeling scenario, the text may exploit the larger distributed representation of the LLM embedding, which is characterized by a set of independent real-valued dimensions. For retrieval purposes, the most likely vector v^* in the embedded linguistic space $V = (v_0, v_1, \dots)$ could first be obtained as:

$$v^* = \max_{v_0, v_1, \dots} \prod_{i \in V} P(v_i), \quad (6)$$

and then the matching with the RoX records could be conducted using the cosine distance metric that has traditionally been supported by the statistical language processing field. However, it remains to be seen how the curse of dimensionality will affect the technical setting. In any case, this realignment with the well established techniques may be of help to increase the TRL.

5.2 Transportability

Generalizing empirical findings to new environments or populations is necessary in the Smart Troubleshooting setting because there are different projects and fleets considered, and each environment exhibits particularities in the written form of the text data. The concept of “transportability” is defined as a license to *transfer* information learned in one environment or domain to a different environment [4], and thus reduce the covariate shift problem.

Transportability analysis assumes that enough structural knowledge about both domains is known in order to substantiate the production of their respective causal diagrams. To formally articulate this transfer procedure, a selection variable K must be introduced to represent the differences between the deployments. In the RoX-based Smart Troubleshooting setting for the industry, the assumption is that the only relevant difference among the environments is driven by the population of subsystems, thus $K \rightarrow Z$ (in fact, some components are only present in specific platforms and assets, so this premise is well founded). The resulting transport formula to generate solutions from a source environment A to a target environment B is shown as follows:

$$P_B(S|do(O)) = \sum_{C,Z} P_A(C|Z) P_A(S|C, Z, O) P_B(Z). \quad (7)$$

If one particular environment B is found to be especially lacking in some aspect, then the rest of the environments A can be used to estimate the desired probabilistic distribution. This smart workaround to a direct data shortage problem that leverages the indirect data from multiple settings is expected to increase the robustness of the prediction, which in turn may help to increase the TRL of the final solution.

5.3 Counterfactual Analysis

So far, the main focus of the analysis has been put on the observed factual data at the population level. However, these data represent only one of the many potential outcomes the system could have experienced: had things been different, an alternative outcome may have been observed. In this sense, a counterfactual describes a potential distribution at the *individual* level driven by hypothetical speculations over data that may contradict the facts. This level of analysis constitutes an additional third rung in the Hierarchy of Causality described in Section 2.2. Conducting this estimation requires the following three steps [32]:

- (1) **Abduction:** Beliefs about the world are initially updated by taking into account all the evidence E given in the context of a single instance/unit. Formally, the exogenous noise probability distributions $P(U)$ are updated to $P(U|E)$.
- (2) **Action:** Interventions are then conducted to reflect the counterfactual assumptions, and a new causal model is therefore created.
- (3) **Prediction:** Finally, counterfactual reasoning occurs over the new model using the updated knowledge.

Gaining access to such involved analysis creates a new area of research to enhance Predictive Maintenance.

5.3.1 Algorithmic Recourse. Algorithmic Recourse is an approach that systematically explores these counterfactual worlds [19]. Such environments are simulated via inference through (atomic) interventions α in the form of alternative problem descriptions. This is expected to help in the recognition and understanding of the general root causes that lead to the system failure [22], and the solution advisory that leads to greater availability.

Formally, the specific retrospective reasoning that these counterfactuals explore on the reported anomaly, i.e., the full description of the solved problem, can be stated as:

$$P(S^*|do(O = \alpha), Z, C, O, S). \quad (8)$$

Given that the solution of a problem was factually implemented and recorded in the RoX data, i.e., through observing all of the variables (Z, C, O, S) , Equation (8) estimates the probability distribution of the textual representation of the hypothetical Solution S^* had the problem been described (and represented) by α , instead of the numerical representation it actually had when it was written. This sophisticated degree of surgical detail enables driving investigations to a deeper level, and this is regarded to help in the advance of the TRL.

6 CONCLUSION

This workshop paper has developed a complete top-down troubleshooting approach from first Causal Inference principles that is also compliant with industrial development guidelines. On the basis of processing technical language, the focus of this learning challenge has been put on creating a distributed representation of linguistic features, and exploiting it for the purpose of obtaining unbiased causal diagnostics and solutions. This approach has been illustrated through a relevant example in the Predictive Maintenance domain, and the results arguably suggest a promising line of future research toward a method to evaluate generative models in other industrial settings.

REFERENCES

- [1] Alaeddini, A., and Dogan, I. 2011. Using Bayesian networks for root cause analysis in statistical process control. *Expert Systems with Applications* 38 (2011), 11230–11243.
- [2] Amatriain, X. 2024. Prompt Design and Engineering: Introduction and Advanced Methods. *arXiv:2401.14423 [cs.SE]* (2024).
- [3] Atienza, D., Larrañaga, P., and Bielza, C. 2022. Hybrid semiparametric Bayesian networks. *TEST* 31 (2022), 299–327.
- [4] Bareinboim, E., and Pearl, J. 2013. A General Algorithm for Deciding Transportability of Experimental Results. *Journal of Causal Inference* 1, 1 (2013), 107–134.
- [5] Bareinboim, E., Correa, J. D., Ibeling, D., and Icard, T. 2022. On Pearl's Hierarchy and the Foundations of Causal Inference. *Probabilistic and Causal Inference: The Works of Judea Pearl* (2022), 507–556.
- [6] Brundage, M. P., Sexton, T., Hodkiewicz, M., Dima, A., and Lukens, S. 2021. Technical language processing: Unlocking maintenance knowledge. *Manufacturing Letters* 27 (2021), 42–46.
- [7] Bunge, M. 2008. *Causality and Modern Science*. Routledge.
- [8] Chen, B., Zhang, Z., Langrené, N., and Zhu, S. 2023. Unleashing the potential of prompt engineering: a comprehensive review. *arXiv:2310.14735 [cs.CL]* (2023).
- [9] Cook, R. I. 2000. How Complex Systems Fail. *Cognitive Technologies Laboratory, University of Chicago* (2000).
- [10] Dersin, P. 2023. *Modeling Remaining Useful Life Dynamics in Reliability Engineering*. CRC Press.
- [11] Feder, A., Keith, K. A., et al. 2022. Causal Inference in Natural Language Processing: Estimation, Prediction, Interpretation and Beyond. *Transactions of the Association for Computational Linguistics* 10 (2022), 1138–1158.
- [12] Fink, O., Wang, Q., Svensén, M., Dersin, P., Lee, W.-J., and Ducoffe, M. 2020. Potential, challenges and future directions for deep learning in prognostics and health management applications. *Engineering Applications of Artificial Intelligence* 92, 103678 (2020), 1–15.
- [13] Ganguly, N., Fazliza, D., Badar, M., et al. 2023. A Review of the Role of Causality in Developing Trustworthy AI Systems. *arXiv:2302.06975 [cs.AI]* (2023).
- [14] Grootendorst, M. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv:2203.05794 [cs.CL]* (2022).
- [15] IEC. 2006. *Analysis techniques for system reliability - Procedure for failure mode and effects analysis (FMEA)*. Technical Report 60812:2006. International Electrotechnical Commission.
- [16] ISO. 2003. *Condition monitoring and diagnostics of machine systems: Data processing, communication and presentation*. Technical Report 13374-1:2003. International Organization for Standardization.
- [17] ISO. 2013. *Space systems - Definition of the Technology Readiness Levels (TRLs) and their criteria of assessment*. Technical Report 16290:2013. International Organization for Standardization.
- [18] Jurafsky, D., and Martin, J. H. 2024. *Speech and Language Processing*. Pearson Education Inc.
- [19] Karimi, A.-H., Barthe, G., Schölkopf, B., and Valera, I. 2022. A Survey of Algorithmic Recourse: Contrastive Explanations and Consequential Recommendations. *ACM Computing Surveys* 55, 5 (2022), 95.
- [20] Kim, S., Suk, J., Cho, J. Y., et al. 2024. The BiGGen Bench: A Principled Benchmark for Fine-grained Evaluation of Language Models with Language Models. *arXiv:2406.05761 [cs.CL]* (2024).
- [21] Lessan, J., Fu, L., and Wen, C. 2019. A hybrid Bayesian network model for predicting delays in train operations. *Computers & Industrial Engineering* 127 (2019), 1214–1222.
- [22] Li, M., Li, Z., Yin, K., Nie, X., Zhang, W., Sui, K., Pei, D. 2022. Causal Inference-Based Root Cause Analysis for Online Service Systems with Intervention Recognition. *Proc. of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2022), 3230–3240.
- [23] Lukens, S., and Ali, A. 2023. Evaluating the Performance of ChatGPT in the Automation of Maintenance Recommendations for Prognostics and Health Management. *Proc. of the Annual Conference of the PHM Society* 15, 1 (2023).
- [24] Macmillan-Scott O, and Musolesi M. 2024. (Ir)rationality and cognitive biases in large language models. *Royal Society Open Science* 11, 240255 (2024).
- [25] Malzer, C., and Baum, M. 2020. A Hybrid Approach To Hierarchical Density-based Cluster Selection. *Proc. of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems* (2020), 223–228.
- [26] Manning, C., and Schütze, H. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press.
- [27] McInnes, L., Healy, J., and Melville, J. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv:1802.03426 [stat.ML]* (2018).
- [28] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. *Proc. of the 26th Conference on Neural Information Processing Systems - Volume 2* (2013).
- [29] Nezhurina, M., Cipolina-Kun, L., Cherti, M., and Jitsev, J. 2024. Alice in Wonderland: Simple Tasks Showing Complete Reasoning Breakdown in State-Of-the-Art Large Language Models. *arXiv:2406.02061 [cs.LG]* (2024).
- [30] Pearl, J. 2012. The Do-Calculus Revisited. *Proc. of the 28th Conference on Uncertainty in Artificial Intelligence* (2012), 4–11.
- [31] Pearl, J. 2019. The seven tools of causal inference, with reflections on machine learning. *Communications of the ACM* 62, 3 (2019), 54–60.
- [32] Pearl, J., Glymour, M., and Jewell, N. P. 2016. *Causal Inference in Statistics: A Primer*. John Wiley and Sons Ltd.
- [33] Pourret, O. 2008. Introduction to Bayesian networks. *Bayesian Networks: A Practical Guide to Applications* (2008), 1–13.
- [34] Reichenbach, H. 1956. *The Direction of Time*. University of California Press, Los Angeles.
- [35] Sala, R., Pirola, F., Pezzotta, G., and Cavalieri, S. 2024. Leveraging Natural Language Processing for enhanced remote troubleshooting in Product-Service Systems: A case study. *Procedia Computer Science* 232 (2024), 1259–1268.
- [36] Touvron, H., Martin, L., et al. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv:2307.09288 [cs.CL]* (2023).
- [37] Trilla, A., Mijatovic, N., and Vilasis-Cardona, X. 2022. Towards Learning Causal Representations of Technical Word Embeddings for Smart Troubleshooting. *International Journal of Prognostics and Health Management* 13, 22 (2022), 1–17.
- [38] Valcamonica, D., Baraldi, P., Zio, E., Decarli, L., Crivellari, A., and La Rosa, A. 2024. Combining natural language processing and bayesian networks for the probabilistic estimation of the severity of process safety events in hydrocarbon production assets. *Reliability Engineering & System Safety* 241, 109638 (2024).
- [39] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. M., Kaiser, L., and Polosukhin, I. 2017. Attention Is All You Need. *Proc. of the 31st Conference on Neural Information Processing Systems* (2017), 1–15.
- [40] Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., and Zhou, M. 2020. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. *arXiv:2002.10957 [cs.CL]* (2020).
- [41] Weidl, G., Madsen, A. L., and Dahlquist, E. 2008. Decision support on complex industrial process operation. *Bayesian Networks: A Practical Guide to Applications* (2008), 313–328.
- [42] Wilson, P. F., Dell, L. D., and Anderson, G. F. 1993. *Root Cause Analysis: A Tool for Total Quality Management*. ASQ Quality Press.
- [43] Woods, C., Selway, M., Bikaun, T., Stumptner, M., and Hodkiewicz, M. 2024. An ontology for maintenance activities and its application to data quality. *Semantic Web* 15, 2 (2024), 319–352.
- [44] Yuan, B., Zhang, J., et al. 2024. Emergence and Causality in Complex Systems: A Survey of Causal Emergence and Related Quantitative Studies. *Entropy* 26, 2 (2024).
- [45] Zwirgmaier, K., Chan, J., Papaioannou, I., Song, J., and Straub, D. 2024. Hybrid Bayesian Networks for Reliability Assessment of Infrastructure Systems. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* 10, 2 (2024).