machine learning & knowledge extraction

MDPI

Article

Leveraging Failure Modes and Effect Analysis for Technical Language Processing

Mathieu Payette [1,*], Georges Abdul-Nour [1], Toualith Jean-Marc Meango [2], Miguel Diago [2] and Alain Côté [2]

[1] Département de Génie Industriel, École d'ingénierie, Université du Québec à Trois-Rivières, Trois-Rivières, QC G8Z 4M3, Canada; georges.abdulnour@uqtr.ca
[2] Hydro-Québec's Research Institute—IREQ, Varennes, QC J3X 1P7, Canada; meango.toualithjean-marc@hydroquebec.com (T.J.-M.M.); diagomartinez.miguel3@hydroquebec.com (M.D.); cote.alain7@hydroquebec.com (A.C.)
* Correspondence: mathieu.payette@uqtr.ca

Abstract: With the evolution of data collection technologies, sensor-generated data have become the norm. However, decades of manually recorded maintenance data still hold untapped value. Natural Language Processing (NLP) offers new ways to extract insights from these historical records, especially from short, unstructured maintenance texts often accompanying structured database fields. While NLP has shown promise in this area, technical texts pose unique challenges, particularly in preprocessing and manual annotation. This study proposes a novel methodology combining Failure Mode and Effect Analysis (FMEA), a reliability engineering tool, into the NLP pipeline to enhance Named Entity Recognition (NER) in maintenance records. By leveraging the structured and domain-specific knowledge encapsulated in FMEAs, the annotation process becomes more systematic, reducing the need for exhaustive manual effort. A case study using real-world data from a major electrical utility demonstrates the effectiveness of this approach. The custom NER model, trained using FMEA-informed annotations, achieves high precision, recall, and F1 scores, successfully identifying key reliability elements in maintenance text. The integration of FMEA not only improves data quality but also supports more informed asset management decisions. This research introduces a novel cross-disciplinary framework combining reliability engineering and NLP. It highlights how domain expertise can be used to streamline annotation, improve model accuracy, and unlock actionable insights from legacy maintenance data.

Keywords: artificial intelligence; natural language processing; technical language processing; engineering of asset management; reliability

Academic Editor: Laura Po

Received: 16 April 2025
Revised: 2 May 2025
Accepted: 6 May 2025
Published: 9 May 2025

Citation: Payette, M.; Abdul-Nour, G.; Meango, T.J.-M.; Diago, M.; Côté, A. Leveraging Failure Modes and Effect Analysis for Technical Language Processing. *Mach. Learn. Knowl. Extr.* **2025**, *7*, 42. https://doi.org/10.3390/make7020042

1. Introduction

The widespread impact of language models is undeniable. The surge in popularity of models such as GPT (Generative Pre-trained Transformer) was notably driven by the launch of ChatGPT-3.5 in November 2022, based on the GPT-3 and subsequently GPT-4 models. Within a few months, major technology companies introduced their own applications: Microsoft Copilot, Google Bard, and Meta's Llama (used across Facebook, Instagram, etc.). These NLP models are trained on millions of texts available online. Despite their impressive generalization capabilities, their use has certain technical limitations [1]. The possibilities for industrial applications are numerous, and the need for such tools for Industry 4.0/5.0 is clear. The ability to process and analyze technical texts efficiently is fundamental to these applications. However, adapting NLP techniques to deal with texts that often deviate

from the syntactic and semantic rules of standard language presents significant challenges. This research addresses these issues by integrating NLP methodologies with reliability engineering techniques, particularly FMEA [2], to develop a robust methodology to train a NER model. The aim is to accurately identify and classify equipment, components, failure modes, and degradation mechanisms in maintenance work orders to improve the automation and reliability of information extraction in industrial contexts. In many organizations, maintenance data, particularly work orders, are recorded in Enterprise Resource Planning (ERP) systems. The quality of maintenance data is often challenged [3,4] and its validation primarily relies on the extraction of text fields describing the observed issues and the repairs performed. These descriptions are often jargon-rich, sometimes specific to language and regionalism, and therefore depart from standard linguistic conventions. Unlike natural language, technical texts often contain irregular sentence structures, abbreviations, acronyms, and domain-specific terminology. These characteristics pose major obstacles for conventional NLP techniques. However, FMEA is a well-established systematic approach commonly used in engineering to identify potential failure modes within a system and assess their impact. Many asset-intensive companies have long utilized FMEA techniques, making relevant data readily available. By integrating the principles of FMEA with NLP techniques, there is an opportunity to take advantage of domain-specific knowledge and improve the accuracy and relevance of text analysis in industrial environments.

Despite the growing interest in NLP applications, limited research has been conducted to develop methodologies specifically tailored for the handling of technical texts within industrial contexts. Existing models often struggle to handle the peculiarities of technical language, leading to suboptimal performance and restricted real-world applicability. In addition, annotation remains costly and inconsistent due to domain-specific terminology and a lack of labeled data. At the same time, organizations often maintain structured expert knowledge in the form of FMEA tables. Despite this, such information is rarely used to support or automate NLP annotation tasks. This represents a critical and underexplored opportunity to reduce manual effort and improve consistency in industrial NER tasks, especially in safety-critical or large-scale maintenance contexts. This study seeks to address these issues through the following methods:

- Using real maintenance data from an electrical utility.
- Investigating the linguistic characteristics of maintenance texts in power systems.
- Proposing a new methodology that combines FMEA knowledge with state-of-the-art NLP techniques to improve the identification and classification of components and failure modes in technical documents.
- Developing and evaluating a specialized NER model trained on annotated datasets sourced from industrial contexts.

The proposition of this research is that by integrating reliability engineering principles with advanced NLP techniques, a robust and efficient methodology can be developed to adapt NER models to recognize and classify components and failure modes in technical texts. This work significantly improves the automation and accuracy of information extraction processes that are essential to guarantee the reliability and safety of industrial operations. Section 2 presents a brief review of the literature and describes NLP techniques and tasks. Section 3 describes the methodology developed in this work and Sections 4 and 5 present, respectively, the case study and a discussion of the results obtained from the analysis, followed by a conclusion in Section 6.

2. Literature Review

This paper proposes to combine qualitative modeling methods from the reliability domain with NLP techniques. This section briefly describes these methodologies as well as the different applications of NLP in reliability engineering reported in the literature.

2.1. Qualitative Modeling in Reliability Engineering

To assess the reliability of an asset, it is crucial to identify the potential mechanisms of degradation and failure modes associated with its operation. Qualitative modeling methodologies are heavily based on the knowledge of experts who have first-hand experience in the design, operation, and maintenance of the asset [5]. A common approach to this is FMEA, which is used to identify potential failure modes of components. Failure modes are defined as observable causes of system failure. The FMEA approach involves breaking down the system into functionally independent subsystems, followed by the identification of the various operating modes and their corresponding configurations. The information is then compiled in a tabular format (organized by operating mode) that outlines the associated failure modes and their effects on other system entities [6]. Furthermore, FMEA serves as a useful tool to examine possible system failures and develop proactive measures to prevent problems [7,8].

2.2. Natural Language Processing

NLP, a subfield of Artificial Intelligence (AI) dating back to the 1940s, focuses on interactions between computers and human language [9]. Computers perceive text as a series of characters without any inherent meaning; NLP converts text into meaningful numerical data. Recently, deep learning models such as transformers and generative AI have significantly intensified the focus on the topic. NLP is responsible for executing various tasks, including text classification, which plays a pivotal role in spam detection and sentiment analysis, as well as text generation, which is essential for applications such as machine translation, summarization, and grammar correction. Furthermore, NER is instrumental in identifying entities, such as company names and personal identifiers, within textual data [10]. This work focuses on developing a custom NER model to identify maintenance-related entities from critical power system assets.

To develop a successful NLP application, several essential steps must be followed. Initially, text preparation and preprocessing are necessary to clean the text and standardize its format. Next, annotations are applied to incorporate linguistic or domain-specific details, and finally, the text is transformed into a numeric format.

2.2.1. Text Preparation

One key NLP task is breaking text into smaller elements for optimal knowledge extraction. Sentence boundary detection (SBD) identifies sentence limits to segment text. Text chunking, or shallow parsing, uses POS tagging to group words into phrases like noun or verb phrases [11]. Tokenization divides text into tokens, like words or punctuation [12]. Thus, SBD segments text into sentences, chunking divides sentences, and tokenization further divides these into tokens. Illustrated in Figure 1 is an example of this process.

Text normalization often involves converting text to lowercase to simplify analysis [12,13]. Another common preprocessing step is stop word removal. Stop words, such as articles and conjunctions (a, an, the, and, etc.), frequently occur in language but add little meaning to a text. While standard NLP tools typically remove these words [13,14], doing so may alter meaning in some cases [1,15] or fail to improve performance [10]. Thus, manually selecting stop words may be preferable for domain-specific applications. Stemming and lemmatization are NLP techniques to reduce words to their root forms. Although the two

techniques have the same objective, they differ in the way in which they reach the word's base form. Stemming is more simple, which is achieved by removing the suffix or prefix of words; for example, walks, walking, walked can be converted to their root form walk. On the other hand, lemmatization is more complex and maps words using morphological analysis or a dictionary to achieve the end result [1,10,13].

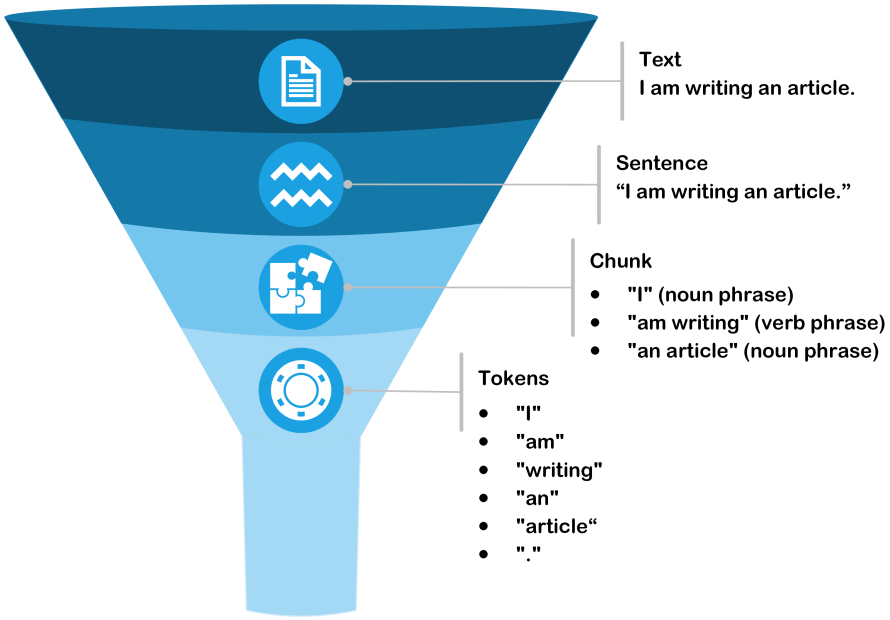


Figure 1. Text preparation.

2.2.2. Text Annotation

Annotation in NLP refers to the process of enriching textual data by assigning tags, labels, or attributes to text elements (e.g., words, phrases, or sentences), thus creating labeled datasets essential for supervised learning [16]. This practice is central to enabling machine learning algorithms to interpret and learn from text effectively. In domain-specific applications such as maintenance work orders, manual annotation, guided by human expertise, remains indispensable to ensure relevant concept extraction and context adaptation [1,15]. Recent advancements, including transfer learning, allow the reuse of pre-trained models, which can be fine-tuned with smaller labeled datasets to support efficient domain adaptation [17]. Publicly available annotated corpora, such as the Penn Treebank and the Universal Dependencies Project, provide foundational resources for syntactic and part-of-speech (POS) analysis in multiple languages [16,18,19]. Today, many Python 3 libraries offer tools for automated annotation, including POS tagging and NER, as summarized in Table 1.

Table 1. Python libraries for NLP applications.

Library	Tools	Authors/ Organizations
NLTK	Tokenization, Stemming, Tagging, Parsing, Sentiment analysis, Corpora	Steven Bird, Edward Loper, and Ewan Klein [20]
SpaCy	Tokenization, POS tagging, NER, Dependency parsing, Text classification, Similarity detection	Explosion AI [21]

Table 1. *Cont.*

Library	Tools	Authors/ Organizations
Gensim	Topic modeling, Word embeddings, Document similarity analysis	Radim Řehůřek [22]
TextBlob	Sentiment analysis, POS tagging, Noun phrase extraction	Siddharth Roy [23]
Stanford CoreNLP	Tokenization, POS tagging, NER, Dependency parsing, Sentiment analysis	Stanford NLP Group [24]
AllenNLP	Text classification, NER, Dependency parsing	Allen Institute for AI [25]
Transformers	Language modeling, Text classification, Question answering	Hugging Face [26]

POS tagging assigns grammatical roles (e.g., nouns, verbs, adjectives) to individual tokens, facilitating lexical disambiguation by resolving ambiguities based on context [10,13,27]. NER performs a similar classification task, identifying and labeling entities such as persons, organizations, and locations. POS emphasizes the grammatical structure, whereas NER identifies semantic concepts from tokens and phrases, frequently utilizing pre-trained models to improve analysis through concept association [1,11,14,27].

These methods, frequently referred to as knowledge extraction techniques, can be tailored to various situations by employing supervised learning for training. There are some cases of domain adaptation in the reliability domain. For example, POS tagging is used as a labeling method to extract maintenance action verbs and the objects/components involved in the maintenance work description [28]. Alternatively, others have used manual labeling or specialized software to create maintenance-specific NER models [29–32].

2.2.3. Representation of Text Data

One of the main challenges in NLP is to represent text in a numerical form that can be interpreted by a machine. The simplest approaches are bag-of-words and one-hot encoding. One-hot encoding uses a binary encoding of a vector of length equivalent to the size of the vocabulary. Bag-of-words, on the other hand, consists of creating a vector in which each text is represented by the occurrence of words in the vocabulary [12,14]. These data representations are sometimes referred to as local representations. In contrast, distributed representations are a more efficient way to represent data, i.e., textual data, by producing lower-dimensional vectors through a learning process [27]. These word embedding methods are heavily based on deep learning algorithms. Popular word embedding methods include word2vec, GloVe, FastText, etc. The impressive developments in AI research are largely due to the development of NLP, and more specifically the progress in the application of deep learning to representation learning. The paper "Attention is all you need" [33] presented the Transformer architecture, surpassing the Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) models with the ability to capture relationships between words and a greater capacity to adapt to word contexts. In addition, transformers are more powerful, as their structure enables parallel computations [27,34]. Table 2 presents the most popular word embedding methods available.

Table 2. Word embedding methods.

Method	Description	Year	Organization
Word2Vec	A family of models (Skip-gram and CBOW) that learn word embeddings by predicting word context.	2013	Google [35]
GloVe (Global Vectors for Word Representation)	Combines co-occurrence statistics with global matrix factorization to learn word representations.	2014	Stanford University [36]
fastText	Extends Word2Vec by representing words as character n-grams, capturing subword information.	2016	Facebook AI Research [37]
BERT (Bidirectional Encoder Representations from Transformers)	A transformer-based language model that learns contextualized word embeddings.	2018	Google [38]
ELMo (Embeddings from Language Models)	Contextualized word embeddings generated by a bidirectional language model.	2018	Allen Institute for AI [39]
GPT (Generative Pre-trained Transformer)	A transformer-based generative language model that learns contextualized word embeddings.	2018	OpenAI [40]
Gecko	A compact model of contextualized representations based on Large Language Models.	2024	Google [41]
LLM2Vec	A model of contextualized representations based on Large Language Models.	2024	MILA [42]

2.3. Maintenance Work Orders Analysis

This section focuses on analyzing various publications in the field of reliability and maintenance that use NLP techniques. Several studies demonstrate the effectiveness of basic NLP methods for processing maintenance texts. For example, railway inspection data are analyzed using word occurrence statistics to classify fault types in emergency requests [43]. Another applies a rule-based method to clean maintenance histories of work orders [44]. In the mining sector, AI-driven and rule-based keyword extraction techniques are compared using survival analysis to assess their impact on the median asset failure time [45]. For HVAC systems, a text mining approach encodes maintenance data through a document-term matrix and applies hierarchical clustering to identify failure-related keywords, which are then used to build association rules for future queries [46]. Building on these methods, a more advanced approach was proposed to extract key terms from maintenance texts and link them to performance indicators such as repair duration [32,47]. NESTOR software 0.3 [48] was used to annotate text and extract elements such as a solution, problem, and object, which served as predictors of maintenance time [32]. The methodology was tested in three case studies—automotive, component supply, and lighting—using word2vec embeddings and NLP pipelines. Predictive models were developed and evaluated using decision trees and neural networks [47].

Many studies highlight the value of human-in-the-loop methods to improve NLP applications in maintenance, emphasizing the role of expert knowledge. One method combines expert insight with annotation tools to extract relevant information and define performance indicators from free text work orders [31,49]. Others highlight the limitations of ready-to-use NLP models for technical texts, advocating for adapted frameworks such as Technical Language Processing, which better handle domain-specific terminology and formatting [1,15]. These frameworks rely heavily on manual annotation and domain expertise to address data quality and quantity issues. Similarly, Bikaun and Hodkiewicz [50] developed a domain-adapted pipeline to detect end-of-life events in mining equipment. By integrating expert-elicited keywords into a classification model, they identified critical events and produced statistical MTBF estimates.

Beyond domain adaptation and expert-in-the-loop strategies, other work focuses on automating practical maintenance tasks using NLP-driven classification and data analysis techniques. One study aims at assigning tasks in healthcare facilities by classifying user-submitted maintenance requests. Several CNN architectures are evaluated using FastText word embeddings, achieving a task classification accuracy of 78% [51]. In the context of building asset management, ref. [52] proposes methods for preprocessing and classifying maintenance text data to identify underperforming assets, benchmark their performance, and generate visualization maps to support decision-making. Similarly, ref. [53] analyzes HVAC maintenance records to uncover data quality issues, employing exploratory data analysis (EDA) and survival analysis to assess the effect of data inconsistencies on performance indicators.

With the advancement of NLP techniques, recent studies examine advanced deep learning models to improve maintenance tasks. In the manufacturing sector, traditional approaches are compared to transformer-based models, with CamemBERT significantly outperforming the bag-of-words for French maintenance texts [54]. Further developments introduce the Transformer-Based Sequential Denoising Auto Encoder (TSDAE) for re-training the BERT model in maintenance, enhancing semantic similarity detection using maintenance work orders from a mining excavators dataset [55]. Building on this, the authors combine NLP with case-based reasoning (CBR) in the aviation domain, using TSDAE-enhanced BERT to retrieve similar cases, achieving a 37% improvement in retrieval accuracy compared to the base model [56].

Other studies tackle data preparation and entity extraction. For instance, RoBERTa is used to automate text cleaning in maintenance logs, improving data consistency [57]. A hybrid approach combines BERT-based classification with SpERT for information extraction, filtering irrelevant texts and identifying entities and causal links related to maintenance tasks [30]. TCAM, which integrates Latent Dirichlet Allocation (LDA) and a Subject-Context Attention Model is used to extract fault-related features for automated diagnosis via case retrieval [58]. Normaly, work orders are assigned manually to technicians, but this process can be automated. A case study in hospital settings shows how using LDA to detect components that need repair and classifiers for technician allocation can streamline this process [59]. Applications in public infrastructure also include the prediction of the priority of maintenance requests, where preprocessing techniques have been shown to influence model performance [60]. Generative AI has also been explored for failure mode classification, focusing on prompt engineering to optimize LLM responses. To reduce ambiguity in maintenance descriptions, NER with morphosyntactic labeling has been applied in the automotive sector to extract actions and repair targets [28].

3. Methodological Approach

The previous sections introduced the applications and methods used to develop a NLP model. First, this section explains how the model is implemented and validated for the following case study. Then, it presents the step-by-step process to develop a custom NER model, exploring how it can benefit from using FMEA documentation.

3.1. Implementation and Validation

This section outlines the experimental framework, including the tools used, the selection of metrics for evaluation, and the validation procedures. By carefully defining our experimental setup, we intend to provide reproducible and meaningful information about the capabilities and limitations of our approach. The development was carried out on the Azure DataBricks platform, a cloud-based analytics solution that combines Apache Spark-based processing with Azure cloud services. Programming is carried out using

PySpark, the Python API for Spark. The library used for modeling is Spark NLP, and the Spacy and NLTK libraries were used primarily for preprocessing and exploratory analysis. The modeling and validation results were recorded using ML Flow. Versions of the tools used during the experiments are as follow:

- Databricks Runtime: 12.2 LTS ML
- Apache Spark: 3.3.2
- Python: 3.10
- Spark NLP: 4.2.8
- NLTK: 3.7

Model training and validation are generated directly in the Spark NLP pipeline. The hyperparameters controlled are as follows:

- Max epoch: 10;
- Learning rate: 0.003;
- Training/Validation: 20–80%.

The training and validation log records the number of true positives, false positives, false negatives, precision, recall, and F1 score for each of the entities identified. In summary, precision is the proportion of true positives over all positive instances, while recall is the proportion of true positives over all correctly classified instances. The F1 score is simply the armonic mean of precision and recall. The program also stores overall values and the macro- and micro-averages. These measurements are available for the training sample and the validation sample for each epoch.

3.2. General Methodology Overview

Developing a custom NER model is very similar to developing a standard supervised learning model. The first step is to gather a labeled dataset. In this case, this requires an annotated dataset; that is, text instances where the entities are already correctly identified. In the case of Spark NLP, the training data must be in CoNLL-2003 format. Then, as mentioned above, the dataset is separated into training and validation samples. Figure 2 details the general approach:

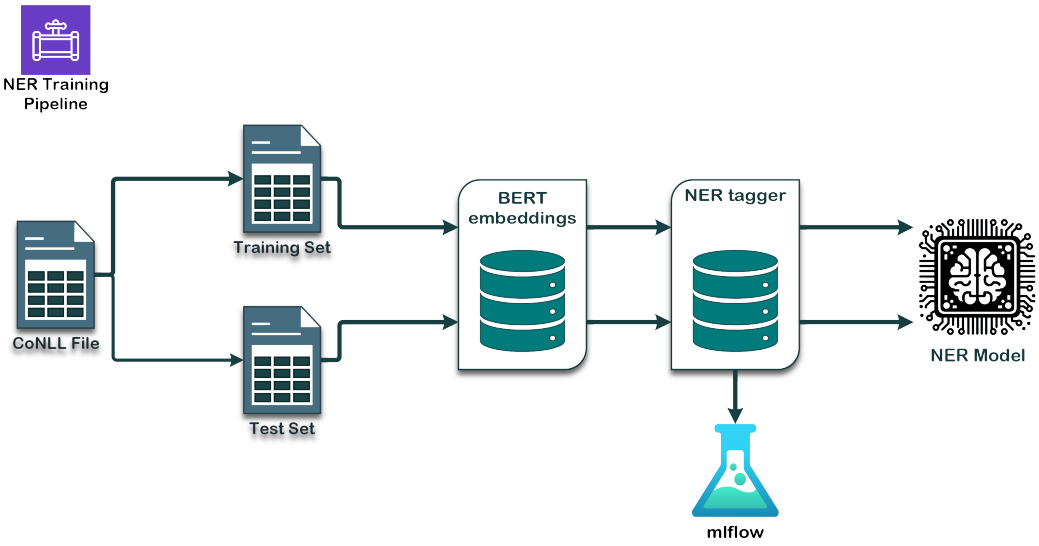


Figure 2. Modeling process.

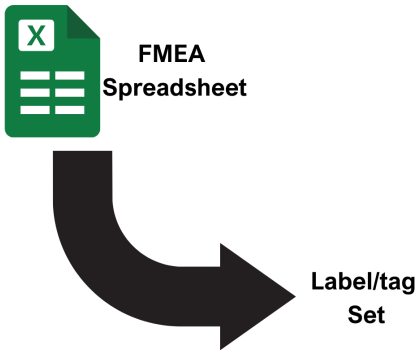
The training and validation sets pass through a BERT component, which enables all the preprocessing (tokenization, word embedding, etc.) to be carried out directly, and then the vectors are pushed into a NER tagger. In Spark NLP, this is referred to as an NER

annotator, and the one used in this project is the NerDL Approach. This annotator has a neural network architecture of Char CNNs- BiLSTM-CRF, which stands for Character-level Convolutional Neural Networks, Bidirectional Long Short-Term Memory, and Conditional Random Field. It combines several neural network structures, making learning easier in tasks such as entity recognition. In summary, the NER annotator is used to identify and classify entities (equipment, components, failure modes, etc.) in the text, and uses a deep learning framework to learn from the examples sent. These simple steps make it easy to define and evaluate a NER model that can be reused for other NLP applications, such as information extraction or text classification. With this framework, entity classification is context sensitive, meaning that a single term can be assigned different labels based on its usage within a sentence. This is mainly due to the use of contextualized embeddings, in this case a French BERT model, and the quality of the training data. Accurate labeling relies on both the richness of the training data and the use of BIO (Beginning, Inside, Outside) tagging in the CoNLL format, which allows the model to learn both single- and multi-token entities. For example, in the phrase 'The power transformer is an asset', the model tags 'power' as B-EQUIPMENT and 'transformer' as I-EQUIPMENT, allowing recognition of 'power transformer' as a complete entity. The model performs well when trained on diverse, well-annotated examples but struggles with unseen or infrequent terms.

3.3. FMEA for Token Classification

As mentioned in the previous section, modeling requires a labeled dataset. There are several sources for finding this kind of dataset, but they are generally adapted to generic entity types (e.g., location, company name, date, etc.). For this project, it was therefore necessary to generate a context-specific dataset. The first step was to collect the data from the FMEA and to extract their general structure for the target equipment, as shown in Figure 3.

ID	Equipment	Primary component	Secondary component	Degradation mechanism	Failure mode	[...]
1	Power Transformer	Cooling system	Pump	DM_pump	FM_pump	[...]
2			Fan	DM_fan	FM_fan	[...]



Pattern	Label
EQUIPMENT	power transformer
EQUIPMENT	transformer
PRIMARY_COMP	cooling system
SECONDARY_COMP	pump
SECONDARY_COMP	fan
DEG_MEC	DM_pump
DEG_MEC	DM_fan
FAIL_MODE	FM_pump
FAIL_MODE	FM_fan
[...]	[...]

Figure 3. Entity extraction.

The FMEA for each asset is developed and periodically reviewed by expert committees in accordance with IEC standards, using technical specifications and equipment documentation. The structure of the FMEA aligns with the component hierarchies, degradation

mechanisms, and failure modes defined within the proprietary catalogs of the organization's maintenance ERP system. These catalogs also provide default classifications for failures of unknown origin or cases that cannot be precisely categorized. Maintenance technicians document work activities through structured forms that include both predefined fields (e.g., drop-down menus) and unstructured free text entries. Through straightforward operations, the information from the FMEA spreadsheets can be utilized to construct a table that links all maintenance-related terms with a list of entities. The maintenance work order texts can now be annotated with pattern-matching techniques using this table. In practice, this means finding an exact correspondance of the terms within the list, e.g., 'transformer', and assigning it a label, in this case 'EQUIPMENT'. In this way, terms are identified and classified according to the same groupings as in the FMEA.

4. Case Study

4.1. Industrial Context

This research is conducted on a Canadian electrical utility, Hydro-Québec (HQ). The data used come from the assets of the transmission network. Hydro-Québec's transmission system operator (TSO) is responsible for ensuring the reliability of the power transmission system for the province of Québec. Its network includes 536 substations and 34,000 km of high-voltage transmission lines [61]. The analysis is focused on one type of asset, but the methodology can be expanded throughout the organization. The benefit of applying this methodology in such a context is that electrical utilities are asset-intensive businesses, meaning that the organization's operations depend on the proper functioning of their assets. Infrastructures must be well maintained, and these companies often have maintenance databases with detailed records of the works carried out. In this case, the company has over fifty years of maintenance history, which provides great opportunities for NLP and ML applications. The following Table 3 presents some examples of translated short text describing maintenance actions. The associated codes have been anonymized using the '#' symbol.

Table 3. Maintenance texts sample.

Original Description	Translated Description
T#=Ventillateur, moteur semble faible.	T#=Faann, motor seems week.
#######:ventillateur T# phase # défectu	######:faann T# phase # failu
######:remplacer récipient gel de sili	######:replace sili gel container
Cond anorm sys refr 1er stage T##	Cond abnorm coo sys 1st stage T##

As illustrated in this table, the maintenance description is brief, lacks structure, and includes spelling errors, abbreviations, and numerical codes (masked).

4.2. Dataset Preparation

This subsection describes the preparation of the dataset, as detailed in Figure 4.

First, the textual maintenance data are preprocessed using simple cleaning rules. Next, the column of the Spark DataFrame containing the text is sent into a Spark NLP pipeline to annotate the text. The first element of the pipeline is the Document Assembler, which is generally the starting point for Spark NLP applications. Its role is essentially to read a string and convert it into a format that can be processed by the other elements of the pipeline. Next, the data, now in document form, are sent to the tokenizer which, as described above, divides the text elements into tokens, the elementary unit of the NLP process. These tokens are then processed in a normalization stage. Spark NLP's Normalizer takes the tokens

as input and cleans up the text, depending on its configuration. The Normalizer can be programmed with regular expressions or transform the tokens using a dictionary (referred to as a slang dictionary). In this project, we built two distinct dictionaries: one for word correction and one for known abbreviations and acronym replacement. The steps used to develop these dictionaries have been published and can be referred to as required [62]. Table 4 presents a few statistics of the cleaning process:

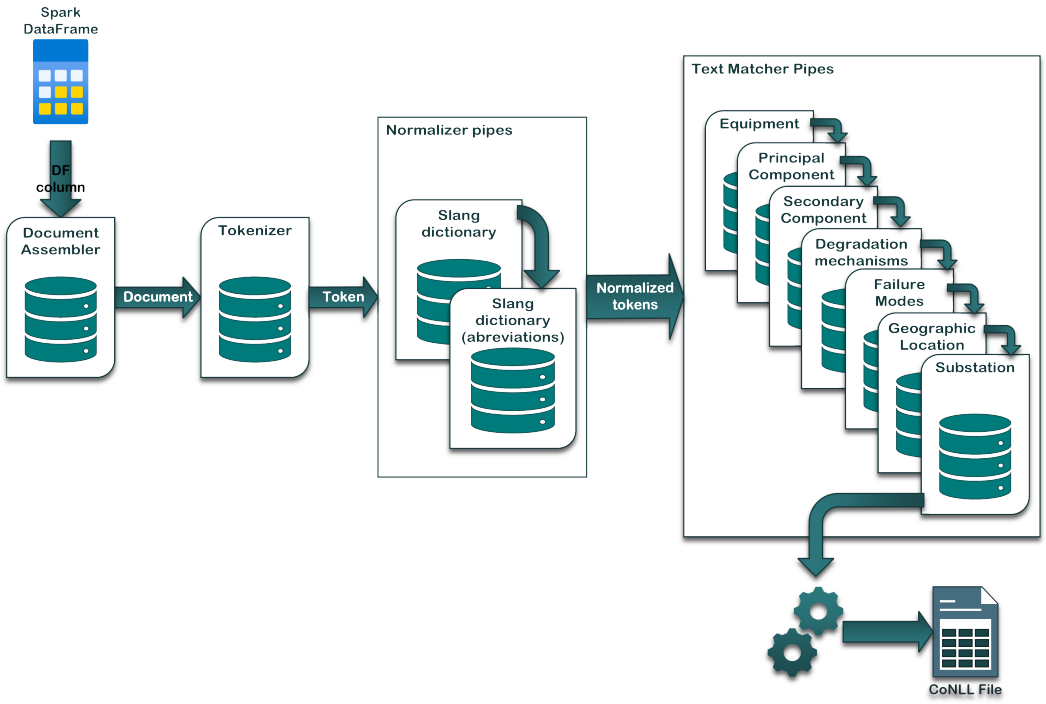


Figure 4. Training dataset preparation steps.

Table 4. Corpus statistics at different stages of processing.

Processing Steps	Number of Texts	Number of Tokens	Number of Words (Vocabulary)
Original Corpus	152,136	5,695,598	157,068
Preprocessing	83,461	1,596,470	31,721
Normalization	69,868	741,997	20,726

Finally, the last stage in the pipeline is the Text Matcher. This element establishes a correspondence between the maintenance text and the entities extracted from the FMEA. Unlike other entity annotation tools that use machine learning, Text Matcher establishes an exact correspondence between the texts and the entity phrase, making it ideal for building a training set. Table 5 presents the number of elements corresponding to each entity type. The counts have been adjusted to better account for multiword expressions representing the same object; for instance, both “power transformer” and “transformer” are treated as equivalent.

Table 5. Number of elements per entity types.

Label Type	Number of Elements
Equipment	2
Primary component	23
Secondary component	69
Degradation mechanism	17
Failure mode	23

Having established standardization dictionaries, the program therefore understands all variants and acronyms of a word, enabling these elements to be identified by the Text Matcher as well. Once the text is fully annotated by the pipeline, the resulting dataframe is transformed into the CoNLL-2003 file format so that it can be used to train the NER model.

5. Results and Discussion

This section presents the results of the case study, applying the methodology introduced earlier. Using data from substation equipment and its FMEA, the study analyzes model training and validation, showcasing entity recognition in maintenance data to highlight model robustness. The study aims to develop and test a text annotation methodology based on FMEA, demonstrating its relevance and effectiveness. The model was trained on maintenance orders from 2010 to 2017, a range selected to support future analyses while keeping the data current. The work orders were filtered for uniqueness and textual completeness. The dataset includes over 15,000 work orders, with 30% used for testing. In total, more than 50,000 examples were used for training, with 20% reserved for validation. Figure 5 shows an example of a maintenance text that has been annotated using the NER model.

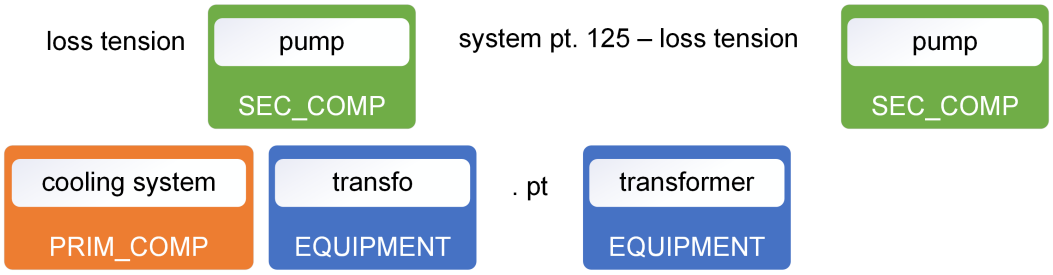


Figure 5. NER model output example.

Figure 6 shows the evolution of the macromean recall over training cycles, for the validation and test sample. The analysis of the curves shows that average recall increases rapidly over the first few training cycles and, after around five cycles, performance begins to stabilize. This indicates convergence of the model. The performance observed on the test sample indicates an excellent ability to generalize to new data. In the case of overlearning, a decrease in performance on the validation set would have been observed, which is not the case here. Furthermore, this chart demonstrates the model's good performance, both in training and on new data.

Once training, validation, and testing had been completed, the results were analyzed to determine whether the model had a good ability to identify and classify entities in the text. Table 6 presents the overall results of the performance of the model after 10 epochs.

Initially, the entity recognition model generally demonstrates excellent performance, as shown by the high micro-average statistics for the F1 score, recall, and accuracy. However, the macro-average shows that there seems to be an imbalance between precision and recall,

indicating that there may be some variability in performance between classes (entities). Looking at the detailed statistics for each class, we can see that the entities corresponding to geographical location and electrical substation (location), as well as degradation mechanisms, are the categories with the lowest scores. However, the results concerning the location class still show that the model is performing really well, with an F1 score of 0.86, with good balance between precision and recall. Table 7 shows the detailed statistics for the test sample and for each type of entity.

Table 6. Macro- and micro-average metrics.

Metric	Precision	Recall	F1 Score
Macro-average	0.96	0.82	0.88
Micro-average	0.97	0.97	0.97

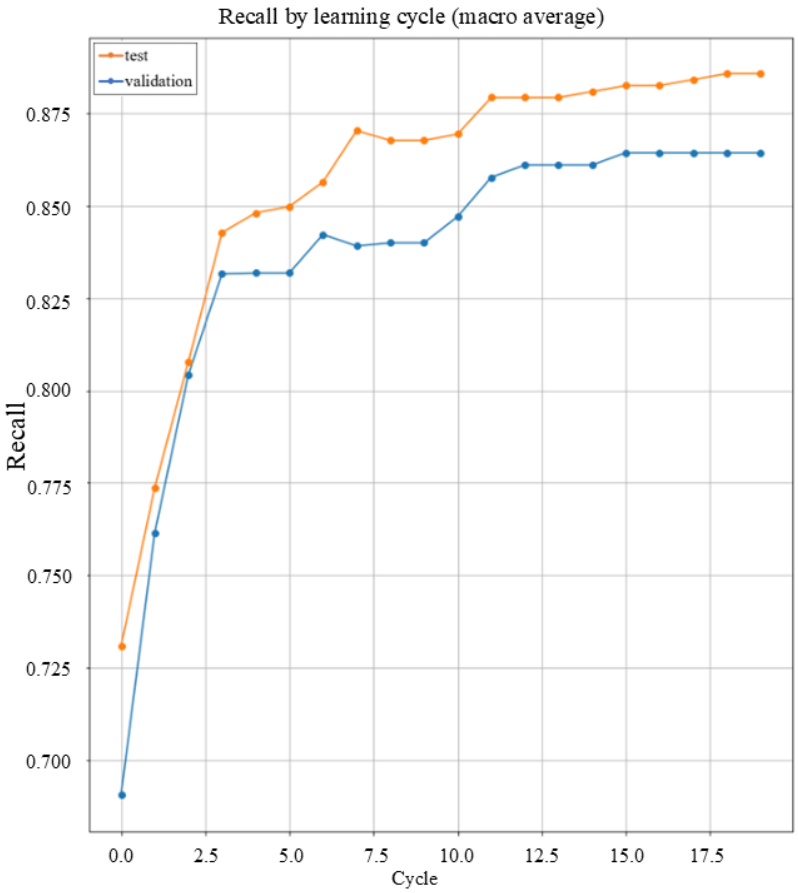


Figure 6. NER model learning curves

Table 7. Training results for each entity type.

Entity	F-1 Score	Precision	Recall
Primary Component	0.90	0.91	0.89
Secondary Component	0.96	0.95	0.97
Equipment	0.99	0.98	1.00
Location	0.86	0.88	0.84
Degradation Mechanism	0.37	0.50	0.29
Failure Mode	0.98	0.99	0.98

For degradation mechanisms, precision is relatively low, as is recall. This class is also underrepresented compared to the other entities in the training and test samples, which probably explains its poor performance. This result suggests that the accuracy of the model could be improved by working on this class. There are several ways that could be employed to achieve better results: adding synonyms or different formulations used to define degradation mechanisms would be a good starting point. Another iteration of improving the dictionaries, with a particular focus on these terms, would also be an interesting avenue. Alternatively, it would also be beneficial to resample to increase the dataset so that it contains a higher proportion of text containing this type of entity, or simply by extending the analysis range over a few more years. In the event that these options do not work, this class could also be removed from the NER model, as its precision is highly unreliable at 50%. Apart from the poor results for the degradation mechanism class, the rest of the entities are generally well classified, and the model's performance is excellent.

Thus, it is clear that with this annotation method, the ability of NLP models to understand technical texts is increased. As mentioned in the literature, applying NLP techniques without adapting them to the context leads to doubtful results. The methodology developed enables the extraction of information in a reliable and rigorous way by reusing knowledge sources already available within the organization. One of the challenges in applying ML techniques to such data is the quality and quantity of available data. With this new annotation technique, the process of creating a dataset, and therefore NLP models, is simplified. The use of existing expertise is not limited to FMEAs, but the methodology can be extended to other analysis techniques that aim to decompose a system and its relationships. However, there are some limitations; as the case study demonstrated, some entities may not be sufficiently referenced in the text, making it difficult to obtain a reliable model to discriminate between them. In addition, when extracting the data, a difference in notation and writing has been noticed, particularly when changes have been made to the company's IT systems; this can result in the use of acronyms that have changed over time, or references to a codification that is not cataloged in any accessible documentation.

Ultimately, the models generated from the methodology will provide a means of including these elements in much more complex NLP pipelines. In fact, NER models are essential components in understanding text for many NLP applications. These models would thus benefit from being employed to perform text classification, for example, to classify the types of activity that have been carried out on a particular equipment item, to extract knowledge from the text by listing the components and equipment involved in a maintenance job, and much more.

Implications and Generalization

While this methodology was developed within the context of electrical utility maintenance, its core principles of leveraging structured documentation (FMEA) for text annotation can be extended to other technical domains. These include the following:

- Aviation: Maintenance logs or incident reports could be analyzed using fault tree or root cause analyses.
- Automotive: Vehicle diagnostics or warranty claim data annotated from repair manuals.
- Oil and Gas: Risk assessments or pipeline inspection reports leveraging hazard analysis methods (e.g., HAZOP).

These documents often contain detailed information about components, failure modes, and operational conditions, which can be transformed into annotated datasets, similarly to FMEA. The adaptability of the approach suggests a potential for broader application, particularly in industries that rely on structured data and unstructured maintenance records.

6. Conclusions

This study presents a novel methodology that integrates FMEA with NLP to improve the processing and analysis of technical maintenance texts. By extracting structured engineering knowledge from FMEA, a key challenge in industrial NLP applications was addressed: the lack of annotated data and the complexity of domain-specific terminology. Our approach enables the efficient creation of a custom NER model capable of accurately identifying and classifying entities such as equipment, components, failure modes, and degradation mechanisms within maintenance work orders. This FMEA-informed annotation process significantly reduces reliance on manual annotation while ensuring greater consistency and contextual relevance. A case study using substation equipment maintenance data from the Hydro-Québec transmission network (2010–2017) validated the methodology. The results show a strong model performance in the recognition of critical entities, demonstrating the practical benefits of combining reliability engineering tools with NLP. The method proved particularly effective in processing technical texts that deviate from standard linguistic norms, common in industrial contexts, thus reinforcing the importance of adapting NLP models to their operational environment. The key contributions of this research include the following:

- Domain-Specific NLP Adaptation: A tailored approach for processing technical maintenance texts, often overlooked in standard NLP models.
- Efficient Annotation Workflow: A scalable method for generating annotated datasets using existing documentation, improving speed and accuracy.
- Improved Maintenance Data Quality: Enhanced extraction of structured insights from maintenance data, supporting better asset management decisions.
- Cross-Disciplinary Innovation: A bridge between reliability engineering and AI, demonstrating how domain knowledge can enrich NLP applications.

Despite these strengths, a limitation remains in the ability to generate enough training data for the modeling process. In fact, even though there was a considerable amount of data, the texts themselves did not contain enough examples for some entity classes to provide adequate results. Furthermore, while the methodology is designed to be adaptable to other technical domains, its generalizability beyond the electrical utilities sector has not yet been tested. This should be addressed in future work, along with scaling up annotated datasets and exploring transfer learning to further improve model performance. Ultimately, this research lays the foundation for broader applications of NLP in industrial maintenance contexts. In fact, the advantage of developing annotated models adapted to technical texts is that they can be reused in more advanced use cases like text classification, information extractions, etc.

Author Contributions: Conceptualization, M.P., G.A.-N., T.J.-M.M., A.C. and M.D.; methodology, M.P. and G.A.-N.; validation, G.A.-N.; writing—original draft preparation, M.P.; writing—review and editing, M.P., G.A.-N., T.J.-M.M., A.C. and M.D.; supervision, G.A.-N., T.J.-M.M., A.C. and M.D.; project administration, G.A.-N. and A.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Hydro-Québec, the Natural Sciences and Engineering Research Council of Canada (NSERC), and the Université du Québec à Trois-Rivières through the Hydro-Québec Asset Management Research Chair.

Data Availability Statement: The datasets presented in this article are not readily available because they are subject to a confidentiality agreement between the Université du Québec à Trois-Rivières and Hydro-Québec. Therefore, the datasets or the associated code cannot be made publicly available.

Acknowledgments: During the preparation of this work, the author used DeepL Translator in order to translate the text from French to English. The author also used Overleaf's generative AI capabilities and Writefull for spellchecking and paraphrasing, and ChatGPT was used to suggest outlines and provide structure to the text. After using these tools/services, the authors reviewed and edited the content as needed and therefore take full responsibility for the content of the publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Brundage, M.P.; Sexton, T.; Hodkiewicz, M.; Dima, A.; Lukens, S. Technical language processing: Unlocking maintenance knowledge. *Manuf. Lett.* **2021**, *27*, 42–46.
2. International Electrotechnical Commission. *IEC 60812: Failure Modes and Effects Analysis (FMEA and FMECA), Edition 3.0*; International Electrotechnical Commission: Geneva, Switzerland, 2018.
3. Unsworth, K.; Adriasola, E.; Johnston-Billings, A.; Dmitrieva, A.; Hodkiewicz, M. Goal hierarchy: Improving asset data quality by improving motivation. *Reliab. Eng. Syst. Saf.* **2011**, *96*, 1474–1481.
4. Sexton, T.; Hodkiewicz, M.; Brundage, M.P. Categorization errors for data entry in maintenance work-orders. In Proceedings of the Annual Conference of the PHM Society, Scottsdale, AZ, USA, 2–5 September 2019; Volume 11. [CrossRef]
5. Payette, M.; Abdul-Nour, G. Asset Management, Reliability and Prognostics Modeling Techniques. *Sustainability* **2023**, *15*, 7493. [CrossRef]
6. Zio, E. *An Introduction to the Basics of Reliability and Risk Analysis*; Series on Quality, Reliability and Engineering Statistics; World Scientific: Singapore, 2007. [CrossRef]
7. Islam, H. Reliability-centered maintenance methodology and application: A case study. *Engineering* **2010**, *2*, 863–873. [CrossRef]
8. Boral, S.; Howard, I.; Chaturvedi, S.K.; McKee, K.; Naikan, V.N.A. An integrated approach for fuzzy failure modes and effects analysis using fuzzy AHP and fuzzy MAIRCA. *Eng. Fail. Anal.* **2020**, *108*, 104195. [CrossRef]
9. Joseph, S.R.; Hlomani, H.; Letsholo, K.; Kaniwa, F.; Sedimo, K. Natural language processing: A review. *Int. J. Res. Eng. Appl. Sci.* **2016**, *6*, 207–210.
10. Jurafsky, D. *Speech & Language Processing*; Pearson Education India: Hoboken, NJ, USA, 2000. [CrossRef]
11. Nadkarni, P.M.; Ohno-Machado, L.; Chapman, W.W. Natural language processing: An introduction. *J. Am. Med. Inform. Assoc.* **2011**, *18*, 544–551.
12. Thanaki, J. *Python Natural Language Processing*; Packt Publishing Ltd.: Birmingham, UK, 2017.
13. Bansal, A. *Advanced Natural Language Processing with TensorFlow 2: Build Effective Real-World NLP Applications Using NER, RNNs, Seq2Seq Models, Transformers, and More*; Packt Publishing Ltd.: Birmingham, UK, 2021.
14. Beysolow, T., II. *Applied Natural Language Processing with Python: Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing*; Apress: New York, NY, USA, 2018. [CrossRef] [PubMed]
15. Dima, A.; Lukens, S.; Hodkiewicz, M.; Sexton, T.; Brundage, M.P. Adapting natural language processing for technical text. *Appl. AI Lett.* **2021**, *2*, e33. [CrossRef]
16. Fort, K. *Collaborative Annotation for Reliable Natural Language Processing: Technical and Sociological Aspects*; John Wiley & Sons: Hoboken, NJ, USA, 2016.
17. Yang, X.; He, X.; Liang, Y.; Yang, Y.; Zhang, S.; Xie, P. Transfer learning or self-supervised learning? A tale of two pretraining paradigms. *arXiv* **2020**, arXiv:2007.04234. [CrossRef]
18. De Marneffe, M.C.; Manning, C.D.; Nivre, J.; Zeman, D. Universal dependencies. *Comput. Linguist.* **2021**, *47*, 255–308.
19. Nivre, J.; De Marneffe, M.C.; Ginter, F.; Goldberg, Y.; Hajic, J.; Manning, C.D.; McDonald, R.; Petrov, S.; Pyysalo, S.; Silveira, N.; et al. Universal dependencies v1: A multilingual treebank collection. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portorož, Slovenia, 23–28 May 2016; pp. 1659–1666.
20. Bird, S.; Klein, E.; Loper, E. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2009.
21. Honnibal, M.; Johnson, M. An improved non-monotonic transition system for dependency parsing. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 1373–1378.
22. Řehůřek, R.; Sojka, P. Software Framework for Topic Modelling with Large Corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Valletta, Malta, 22 May 2010; pp. 45–50.
23. Loria, S. TextBlob: Simplified Text Processing. 2018. Version 0.15.1. Available online: https://textblob.readthedocs.io/ (accessed on 2 February 2024).
24. Manning, C.D.; Surdeanu, M.; Bauer, J.; Finkel, J.R.; Bethard, S.; McClosky, D. The Stanford CoreNLP natural language processing toolkit. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, MD, USA, 23–24 June 2014; pp. 55–60.

25. Gardner, M.; Grus, J.; Neumann, M.; Tafjord, O.; Dasigi, P.; Liu, N.F.; Peters, M.; Schmitz, M.; Zettlemoyer, L.S. AllenNLP: A Deep Semantic Natural Language Processing Platform. *arXiv* **2017**, arXiv:1803.07640.
26. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, 16–20 November 2020; pp. 38–45. [CrossRef]
27. Liu, Z.; Lin, Y.; Sun, M. *Representation Learning for Natural Language Processing*; Springer Nature: Berlin/Heidelberg, Germany, 2020. [CrossRef]
28. Giordano, V.; Fantoni, G. Decomposing maintenance actions into sub-tasks using natural language processing: A case study in an Italian automotive company. *Comput. Ind.* **2025**, *164*, 104186. [CrossRef]
29. Li, H.; Deng, F.; Lu, J.; Zhang, T.; Li, H. An Application of Automatic Text Revision for Power Defect Log. *J. Phys. Conf. Ser.* **2021**, *1757*, 012027. [CrossRef]
30. Hershowitz, B.; Hodkiewicz, M.; Bikaun, T.; Stewart, M.; Liu, W. Causal knowledge extraction from long text maintenance documents. *Comput. Ind.* **2024**, *161*, 104110. [CrossRef]
31. Sexton, T.; Brundage, M.P.; Hoffman, M.; Morris, K.C. Hybrid datafication of maintenance logs from AI-assisted human tags. In Proceedings of the 2017 IEEE International Conference on Big Data, Boston, MA, USA, 11–14 December 2017; IEEE: Piscataway, NJ, USA, 2017. [CrossRef]
32. Navinchandran, M.; Sharp, M.E.; Brundage, M.P.; Sexton, T.B. Studies to Predict Maintenance Time Duration and Important Factors From Maintenance Workorder Data. In Proceedings of the Annual Conference of the PHM Society, Scottsdale, AZ, USA, 2–5 September 2019; Volume 11. [CrossRef]
33. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
34. Ravichandiran, S. *Getting Started with Google BERT: Build and Train State-of-the-Art Natural Language Processing Models Using BERT*; Packt Publishing Ltd.: Birmingham, UK, 2021.
35. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, *26*.
36. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543. [CrossRef]
37. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146.
38. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
39. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. In Proceedings of the North American Chapter of the Association for Computational Linguistics, New Orleans, LA, USA, 1–6 June 2018.
40. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. 2018. Available online: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf (accessed on 1 April 2025).
41. Lee, J.; Dai, Z.; Ren, X.; Chen, B.; Cer, D.; Cole, J.R.; Hui, K.; Boratko, M.; Kapadia, R.; Ding, W.; et al. Gecko: Versatile Text Embeddings Distilled from Large Language Models. *arXiv* **2024**, arXiv:2403.20327.
42. BehnamGhader, P.; Adlakha, V.; Mosbach, M.; Bahdanau, D.; Chapados, N.; Reddy, S. LLM2Vec: Large Language Models Are Secretly Powerful Text Encoders. *arXiv* **2024**, arXiv:2404.05961.
43. Stenström, C.; Aljumaili, M.; Parida, A. Natural language processing of maintenance records data. *Int. J. Comadem* **2015**, *18*, 33–37. [CrossRef]
44. Hodkiewicz, M.; Ho, M.T.W. Cleaning historical maintenance work order data for reliability analysis. *J. Qual. Maint. Eng.* **2016**, *22*, 146–163. [CrossRef]
45. Sexton, T.; Hodkiewicz, M.; Brundage, M.P.; Smoker, T. Benchmarking for Keyword Extraction Methodologies in Maintenance Work Orders. In Proceedings of the Annual Conference of the PHM Society, Philadelphia, PA, USA, 24–27 September 2018; Volume 10. [CrossRef]
46. Gunay, H.B.; Shen, W.; Yang, C. Text-mining building maintenance work orders for component fault frequency. *Build. Res. Inf.* **2018**, *47*, 518–533. [CrossRef]
47. Navinchandran, M.; Sharp, M.E.; Brundage, M.P.; Sexton, T.B. Discovering critical KPI factors from natural language in maintenance work orders. *J. Intell. Manuf.* **2022**, *33*, 1859–1877. [CrossRef]
48. Sexton, R. NESTOR. 2019. Available online: https://www.nist.gov/services-resources/software/nestor (accessed on 1 October 2024). [CrossRef]

49. Brundage, M.P.; Morris, K.C.; Sexton, T.; Moccozet, S.; Hoffman, M. Developing Maintenance Key Performance Indicators From Maintenance Work Order Data. In *Proceedings of the ASME 2018 International Manufacturing Science and Engineering Conference, College Station, TX, USA, 18–22 June 2018*; American Society of Mechanical Engineers: New York, NY, USA, 2018. [CrossRef]
50. Bikaun, T.; Hodkiewicz, M. Semi-automated estimation of reliability measures from maintenance work order records. In Proceedings of the European Conference of the PHM Society 2021, Virtual, 28 June–2 July 2021; Volume 6, p. 9. [CrossRef]
51. Bouabdallaoui, Y.; Lafhaj, Z.; Yim, P.; Ducoulombier, L.; Bennadji, B. Natural Language Processing Model for Managing Maintenance Requests in Buildings. *Buildings* **2020**, *10*, 160. [CrossRef]
52. Nojedehi, P.; O'brien, W.; Gunay, H.B. Benchmarking and visualization of building portfolios by applying text analytics to maintenance work order logs. *Sci. Technol. Built Environ.* **2021**, *27*, 756–775. [CrossRef]
53. Annalisa, C.; Coline, B.; Lynn, P.; Michael, B.; Thurston, S. The Impact of Data Quality on Maintenance Work Order Analysis: A Case Study in HVAC Work Durations. In Proceedings of the 6th European Conference of the Prognostics and Health Management Society, Online, 28 June–2 July 2021.
54. Naqvi, S.M.R.; Varnier, C.; Nicod, J.M.; Zerhouni, N.; Ghufran, M. Leveraging free-form text in maintenance logs through bert transfer learning. In *Progresses in Artificial Intelligence & Robotics: Algorithms & Applications: Proceedings of 3rd International Conference on Deep Learning, Artificial Intelligence and Robotics, (ICDLAIR) 2021, Salerno, Italy, 13–21 December 2021*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 63–75. [CrossRef]
55. Naqvi, S.M.R.; Ghufran, M.; Meraghni, S.; Varnier, C.; Nicod, J.M.; Zerhouni, N. Generating Semantic Matches Between Maintenance Work Orders for Diagnostic Decision Support. In Proceedings of the Annual Conference of the PHM Society, Virtual, 28 June–2 July 2022; Volume 14. [CrossRef]
56. Naqvi, S.M.R.; Ghufran, M.; Meraghni, S.; Varnier, C.; Nicod, J.M.; Zerhouni, N. CBR-Based Decision Support System for Maintenance Text Using NLP for an Aviation Case Study. In Proceedings of the 2022 Prognostics and Health Management Conference, London, UK, 27–29 May 2022; IEEE: Piscataway, NJ, USA, 2022. [CrossRef]
57. Deloose, A.; Gysels, G.; De Baets, B.; Verwaeren, J. Combining natural language processing and multidimensional classifiers to predict and correct CMMS metadata. *Comput. Ind.* **2023**, *145*, 103830. [CrossRef]
58. Hu, Z.; Zhang, X.; Xiong, H. Two-stage attention network for fault diagnosis and retrieval of fault logs. *Expert Syst. Appl.* **2024**, *249*, 123365. [CrossRef]
59. Li, Y.; Liu, Y.; Zhang, J.; Cao, L.; Wang, Q. Automated analysis and assignment of maintenance work orders using natural language processing. *Autom. Constr.* **2024**, *165*, 105501. [CrossRef]
60. D'Orazio, M.; Bernardini, G.; Giuseppe, E.D. Influence of pre-processing methods on the automatic priority prediction of native-language end-users' maintenance requests through machine learning methods. *J. Inf. Technol. Constr. (ITcon)* **2024**, *29*, 99–116. [CrossRef]
61. Hydro-Québec. Un avenir à bâtir, Rapport Annuel 2023. Technical Report, Hydro-Québec, 75, boulevard René-Lévesque Ouest Montréal (Québec). 2023. Available online: https://www.hydroquebec.com/data/documents-donnees/pdf/rapport-annuel-2023-hydro-quebec.pdf (accessed on 1 October 2024).
62. Payette, M.; Abdul-Nour, G.; Meango, T.J.M.; Diago, M.; Côté, A. Enhancing Maintenance Data Analytics: A Novel Failure Mode and Effect Analysis-Natural Language Processing Integration. In Proceedings of the 2025 Annual Reliability and Maintainability Symposium (RAMS), Destin, FL, USA, 27–30 January 2025; IEEE: Piscataway, NJ, USA, 2025; pp. 1–6.