Unsupervised Analytics Models using Natural Language and Sensory Data for Industrial Equipment Degradation Modeling and Decision-Making

By

Abhijeet Sandeep Bhardwaj

A dissertation submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

(Industrial Engineering)

at the

University Of Wisconsin-Madison

2023

Date of final oral examination: 05/26/2023

The dissertation is approved by the following members of the Final Oral Committee:

Dharmaraj Veeramani, Professor, Department of Industrial and Systems Engineering (Committee Chair)

Shiyu Zhou, Professor, Department of Industrial and Systems Engineering

Yonatan Mintz, Assistant Professor, Department of Industrial and Systems Engineering

Kaibo Liu, Associate Professor, Department of Industrial and Systems Engineering

Zhengjun Zhang, Professor, Department of Statistics

Acknowledgments

I owe my deepest gratitude to my academic advisors, Dr. Dharmaraj Veeramani, without whom this work would not have been possible. I sincerely thank Dr. Veeramani for introducing me into academia and his continuous dedication to this work.

I would sincerely like to express my special thanks and gratitude to Dr. Shiyu Zhou, Dr. Yonatan Mintz, Dr. Kaibo Liu and Dr. Zhengjun Zhang for their dedication, time, support and contribution in helping me realize the challenges of this research work.

I would also like to take the opportunity to thank the faculties of my courses from Statistics department (Dr. Wei Yin Loh, Dr. Brian Yandell, Dr. Miaoyan Wang, Dr. Chunming Zhang, Dr. Lu Mao, Dr. David Anderson) who helped me evolve as a researcher. I am also grateful to Prof. Ramathasan Thevamaran and my friend Abhishek Gupta for providing me the opportunity to showcase my statistical learning in manufacturing of VACANT. I am also thankful to my lab-mates Jinwen Sun, Congfang Huang, Ziqian Zheng and Vipul Bansal for their support.

Last but not the least, I acknowledge that I would not have completed this journey without the blessings and well wishes of the Almighty, my mother Smt. Roopali Bhardwaj, my father Dr. Sandeep Kumar Bhardwaj, my sister Ankita Kaushik, my brother in law Tarun Kaushik, and my niece Arunika Kaushik. I cannot thank much to my wife Mrs. Ananya Agnihotri for her constant emotional and mental support throughout my journey.

Abstract

The insights captured and documented in the form of natural language text provide valuable inferences that hold significant implications for various domains, including industrial, healthcare, software, marketing, and product sectors. By analyzing textual data, such as maintenance records, medical records, customer feedback, and product reviews, these domains can extract meaningful information and gain a deeper understanding of operational efficiency, patient care, software performance, market trends, and product perception. In the context of industries, Original Equipment Manufacturers (OEMs) collect a wide range of data to support essential business decisions. Frequently failing components of complex hierarchical equipment cause a tremendous loss to both the OEM and equipment owners (operators). While OEMs suffer financial loss from warranty claims, operators incur loss due to downtime caused by the equipment breakdown. OEMs strive to improve equipment quality by identifying faulty design and poor material used in its production to mitigate financial losses. At the same time, operators seek to lower the impact of downtime by arranging better spare part inventory and appropriate equipment maintenance schedules. Thus, it is crucial to know precisely the sub-component that failed in hierarchical equipment and the root cause of its failure.

Maintenance records generated while inspecting equipment contain rich descriptive information in unstructured free-text format. The unstructured data pertains information about *health condition* of equipment sub-component, repair actions associated with *failed sub-components*, its *failure mechanism*, operating conditions etc.,. Though easy to interpret

by humans, extracting information manually by reading such maintenance records is impractical. Thus converting the unstructured data into a structured format becomes a primary challenge. Natural Language Processing (NLP) models could come to the rescue for the same, however, supervised training of such models becomes infeasible considering the sparsity of labeled maintenance records. Further, text data often contain issues like semantic ambiguity, domain-specific vocabulary, Etc. Thus, this thesis addresses the challenge of extracting structured information for failure diagnosis from unstructured maintenance records in an unsupervised manner. The thesis further demonstrates that combining the insights generated by maintenance records with the condition-based maintenance sensory signals helps generate robust reliability models, thus paving the way for a more realistic maintenance decision-making framework. To tackle the challenges mentioned above following tasks are investigated in this Ph.D. dissertation thesis:

- A mathematical representation of the textual data is imperative to analyze unstructured maintenance records in an automated manner. NLP models that rely on the generic objective of word co-occurrence are insufficient to tackle issues like semantic ambiguity, noisy words, and domain-specific vocabulary. It is imperative to include custom domain information while generating efficient mathematical representation for words in maintenance records to overcome this.

- A complex hierarchical equipment is usually represented by its bill-of-material referred as taxonomy. It is important to identify the taxonomy branch of sub-components that failed during a breakdown event for failure diagnosis. A framework is developed that leverages maintenance records for the same, while addressing the issues like noisy records, non-uniqueness of taxonomy terms etc., in an unsupervised framework and provide confidence scores for each extracted taxonomy branch.

- Another important task of failure diagnosis is to identify the mechanism by which an equipment sub-component failed. To disambiguate the failure mechanism, multiple un-

supervised base classification algorithms could be developed that exploit different facets of domain knowledge. However, these algorithms may diverge, and thus an efficient method to ensemble the results of such base algorithms to generate an unsupervised multi-class ensemble classifier is studied.

- Maintenance records generated during inspection help infer the sentiment about equipment health status. However, the records suffer challenges like 1) unavailability of labeled data and industrial sentiment lexicons for identifying equipment health status, 2) presence of negation words that alter the sentiment. To mitigate the same, a robust model that identify equipment health status from maintenance records is developed.

- Existing reliability models utilize indirect state observations provided by sensory signals to estimate system dynamics, in addition to hard failures data. However, these models do not take into account the direct state observations provided by maintenance records generated during maintenance actions. Furthermore, the existing prognosis methods do not allow for the modeling of the effects of different maintenance actions taken on a system while it is operating. To address this limitation, a novel joint model is proposed. This model fuses real-time sensor signals and discrete state information from manual interventions to learn a holistic condition-based maintenance model.

The proposed methods can be effectively applied not only for improved failure diagnosis and reliability modeling but also to pave the way for realistic maintenance planning of complex hierarchical equipment using unstructured maintenance records. Although the thesis primarily focuses on problems in industrial settings, it also provides valuable insights for practitioners in healthcare, software, and product sectors.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Unstructured natural language text data is generated in various processes across different domains, including healthcare, industries, software, and other product sectors. In the healthcare domain, text data is produced through electronic health records, physician notes, laboratory reports, and patient surveys, providing valuable insights into patients' medical history, diagnoses, treatment plans, and overall health status. In the software domain, text data is generated as console logs, which contain valuable information about system behavior, errors, warnings, and other relevant events. Additionally, in product-oriented firms, text data is generated through customer feedback reports and social media comments, reflecting consumer sentiments and preferences about a product. In industries involved in the manufacturing or deployment of complex equipment, such as those used on rigs in the oil and gas industry, unstructured text data typically takes the form of maintenance records and incident reports. Although the primary focus of this thesis is to analyze the operational and failure dynamics of such complex equipment using unstructured text data, the methods proposed here can be easily adapted to the healthcare and other domains where text data provides crucial insights about the respective field.

The complex machines studied in this thesis comprise a hierarchy of multiple interconnected sub-assemblies and constituent components. For example, an equipment unit (e.g., mud pump) comprises sub-units (e.g., fluid end) which in turn includes several maintainable items (e.g., piston and liner) containing individual parts (e.g., liner). Failure of any sub-component in the equipment hierarchy can result in downtime that can have significant negative impacts in terms of lost production time, warranty and repair costs, and other economic penalties. Thus, both original equipment manufacturers (OEMs) and rig operators are interested in gaining insights from each failure event regarding the failed sub-component and associated failure mechanism. For OEMs, insights from failure diagnosis can guide improvements to the equipment design and warranty policies. For rig operators, these insights can help in making changes to operation and maintenance practices to increase equipment reliability and uptime and better management of spare parts inventory and technician resources. While industrial equipment typically contain a large number of sensors to enable the various equipment functions and to monitor the equipment condition and operation status, sensor data alone are often inadequate for analyzing failure events. Only when equipment are dismantled and inspected by maintenance technicians, an accurate understanding of the equipment condition and its failure is achieved. The findings obtained by technicians are subsequently recorded in unstructured free text format in maintenance records. Hence, the maintenance records (specifically unstructured textual notes) created by technicians during inspection, routine maintenance and repair of equipment are an invaluable knowledge resource for OEMs and rig operators to analyze failure events. Yet, in current industrial practice, this rich unstructured data is largely underutilized due to the high level of time and effort required for manually reviewing and analyzing a voluminous amount of maintenance records. Thus, there is a need for efficient and effective methods for automatic extraction of insights from unstructured maintenance records.

Analyzing failure events requires the ability to answer two essential questions:

- What is the equipment sub-component that caused the failure and the corresponding

equipment taxonomy branch?

- Why did this sub-component fail (i.e., what is the type of failure mechanism)?

Figure 1.1 shows an example of a maintenance record and the corresponding equipment taxonomy branch and failure mechanism. Accurate determination of the failed sub-component (and associated equipment taxonomy branch) and the failure mechanism from the unstructured text using automated means is non-trivial due to a variety of challenges including many that are inherent in such data. (1) The quality of unstructured data in maintenance records is often poor [64]. For instance, the data can be noisy from the presence of non-standard terms (e.g., due to abbreviations and regional language differences). (2) In many cases, the records can contain extraneous information that are not relevant for failure diagnosis, but also be missing key information needed to identify the failed equipment sub-component and its failure mechanism. (3) Challenges can also arise from semantic ambiguity of many terms used by technicians in maintenance records. For example, the word "pump" may refer to either "mud pump" (a type of rig equipment), or a component within a rig equipment, or a verb. (4) In the equipment taxonomy, some sub-components may appear on multiple branches. Thus, extracting a sub-component from the maintenance record can be inadequate for uniquely determining the equipment taxonomy branch associated with the failure event. (5) Similarly, a sub-component can be prone to multiple failure mechanisms which can add further amibiguity in failure diagnosis. (6) The maintenance records can contain negation words that may alter the sentiment expressed regarding the equipment condition and health status. (7) Creation of a large, labeled data set is impractical due to the prohibitive amount of effort involved in manual review of maintenance records. Also the sparse available labeled data in maintenance domain is usually of very poor quality [64] and could not be used for training. Hence, labeled data are typically unavailable for developing supervised machine learning algorithms for analyzing maintenance records.

While natural language processing methods are not new, existing approaches exhibit significant deficiencies in addressing the aforementioned challenges of analyzing unstructured

Maintenance Record	Failed Taxonomy Branch	Failure Mechanism
Pump 1 cyl 1 liner 2803hrs Pump 1 cyl 1 piston 552hrs----------Replace mud pump 1 cyl 1 liner due fluid passing. Once this was pulled the liner was observed to have some bad scoring. Replaced liner and piston. No other damage to report.----------	Mud Pump (Equipment Unit) ↓ Fluid end (Sub-Unit) ↓ Piston and liner (Maintainable Item) ↓ Liner (Part)	Material (wear)

Figure 1.1: Example of a maintenance record

maintenance records in industrial domains. Moreover, in addition to automatically identifying the failed equipment sub-component, the corresponding equipment taxonomy branch, and the associated failure mechanism, there exists a pressing need for a comprehensive reliability model that assesses the equipment condition by leveraging insights from unstructured maintenance records and sensor data in a combined manner. These models should also account for changes in system dynamics resulting from intermediate maintenance actions, enabling the formulation of maintenance plans that do not rely on assumptions. Addressing these research challenges and needs constitutes the precise motivation and objective of this dissertation.

1.2 Research Objective

The overarching objective of this research is to establish a collection of effective methodologies that utilize unsupervised learning for the analysis of unstructured maintenance records, along with domain-specific knowledge. These methodologies aim to support failure diagnosis, equipment health status assessment, and holistic joint reliability modeling of complex hierarchical industrial equipment, as illustrated in Figure 1.2. The outcomes of this dissertation is supposed to facilitate development of realistic, model based inspection and maintenance policies that rely on actual system dynamics under different maintenance actions.

The following terminology is used in the dissertation:

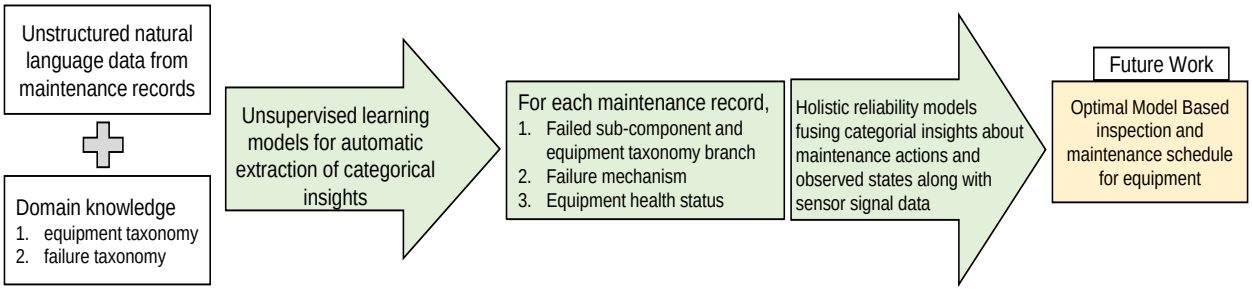


Figure 1.2: Research areas undertook and challenges tackled

- **Word:** A word is the most fundamental unit used in textual descriptions e.g., "repair", "leak", "part". A word is also referred to as a 'token' in this work.

- **Vocabulary:** The set of all available words constitutes a vocabulary.

- **Document:** A document is a collection of words which is used to describe any event. For example, each description (maintenance record) in Fig. 1.1 constitutes a document.

- **Sentiment Lexicon**: Words or phrases that indicate the sentiment of a document. For example "is working perfectly fine" indicates that the equipment/sub-component is not close to a failure state.

- **Corpus:** The collection of all available documents.

We provide below a brief overview of the research challenges and tasks to achieve the proposed vision.

- An effective mathematical representation of unstructured maintenance records is essential for analyzing them. NLP models allow vector representations of words (called word embeddings) by learning semantic similarities between neighboring words. However, this learning objective is restrictive in the context of industrial maintenance records. Many words/terms that are highly associated to each other from domain context may not co-exist in a given maintenance record. For example, different instances of a given failure mechanism like 'wear' and 'break' for material failure do not necessarily co-exist

in the same maintenance record. But, both the terms identify the same failure mechanism. Thus, to make robust word embeddings, a method is needed that can jointly learn from the semantic similarity of words that co-occur along with words that are similar as per the domain knowledge.

- An important facet of failure diagnosis is the ability to identify the complete equipment taxonomy branch of sub-components that malfunctioned during a breakdown event. As noted earlier, due to a variety of reasons, accurately determining the impacted taxonomy branch can be hard. First, while inspecting the equipment, technicians tend to examine and record observations regarding numerous sub-components including those that were not affected by or a cause of the failure, thus infesting maintenance records with noisy words. Second, maintenance records can sometimes be brief and contain only partial information about the failed equipment taxonomy branch. Third, many sub-components can have multiple parents in the equipment taxonomy, thus creating ambiguity regarding the specific branch associated with the failure event. Therefore, algorithms are needed that not only can effectively tackle these challenges but also provide a measure of confidence (i.e., confidence level) for the results of the automatic extraction to aid technician during implementation phase.

- There are any types of failures (such as mechanical, electrical and hydraulic) that can occur in industrial equipment, and many sub-components can be prone to more than one types of these failure mechanisms. Accurately determining the failure mechanism in an unsupervised manner from a maintenance record can be challenging due to inadequate and noisy information in the unstructured text. One approach to overcome these challenges is to employ multiple unsupervised base classifiers to analyze the maintenance record from different perspectives by leveraging different domain knowledge sources. Yet, for a given maintenance record, it is possible that the different base classifiers yield different results for the failure mechanism. Hence, an unsuper-

vised multi-class ensemble model is needed to utilize the results from the different base classifiers and predict the failure mechanism associated with the maintenance record.

- Inspections carried out by technicians during scheduled maintenance and repair reveal much information about the working condition and health status of the equipment. Technicians often express their perception and sentiment about the equipment health status in the form of unstructured text in the maintenance record. Inferring the equipment condition from these records is challenging, especially due to the lack of labeled data. Transfer learning using advanced NLP models like BERT [166] support domain adaptation of sentiment from other application areas. However, a large difference between the two domains reduces its effectiveness. Apart from this, unsupervised methods in sentiment analysis rely on pre-existing sentiment lexicons to identify the document's sentiment. However, such generalized lexicons are not helpful in an industrial context as maintenance records commonly contain domain-specific lexicons like "above the threshold limit". Also, a significant challenge arises from the presence of negation words that alter the sentiment indicated by the lexicon altogether. Hence, there is a need to establish a unsupervised methodology for automatically extracting the sentiment regarding the equipment health status from maintenance records.

- Data obtained from Condition Based Maintenance (CBM) sensors embedded in industrial equipment is increasingly utilized for modeling equipment reliability and dynamics. However, such models rely on information about latent degradation states inferred from indirectly observed sensor signals. Additionally, existing reliability models in the literature assume that a system either crosses a hypothesized threshold and experiences a soft failure or fails abruptly, resulting in a hard failure. However, these models do not consider the effects of various actions performed by technicians during the equipment's lifecycle. The sentiment expressed by technicians about the equipment's health status in maintenance records provides a direct observation of the hidden degradation states.

Moreover, these direct state observations can help improve the precision of the emission model for sensor signals when estimating latent states. However, these direct state observations are not always available and are often sparse depending on the inspection and maintenance plan. Consequently, there is a need to develop a formal methodology that can provide a comprehensive reliability model capable of capturing the system dynamics using both direct and indirect state observations, while also accounting for the effects of different maintenance actions.

1.3 Outline of Dissertation

The dissertation focuses on solving the research problems as mentioned above in section 1.2. Figure 1.3 provides a brief outline of the research objectives addressed in each chapter of the dissertation.

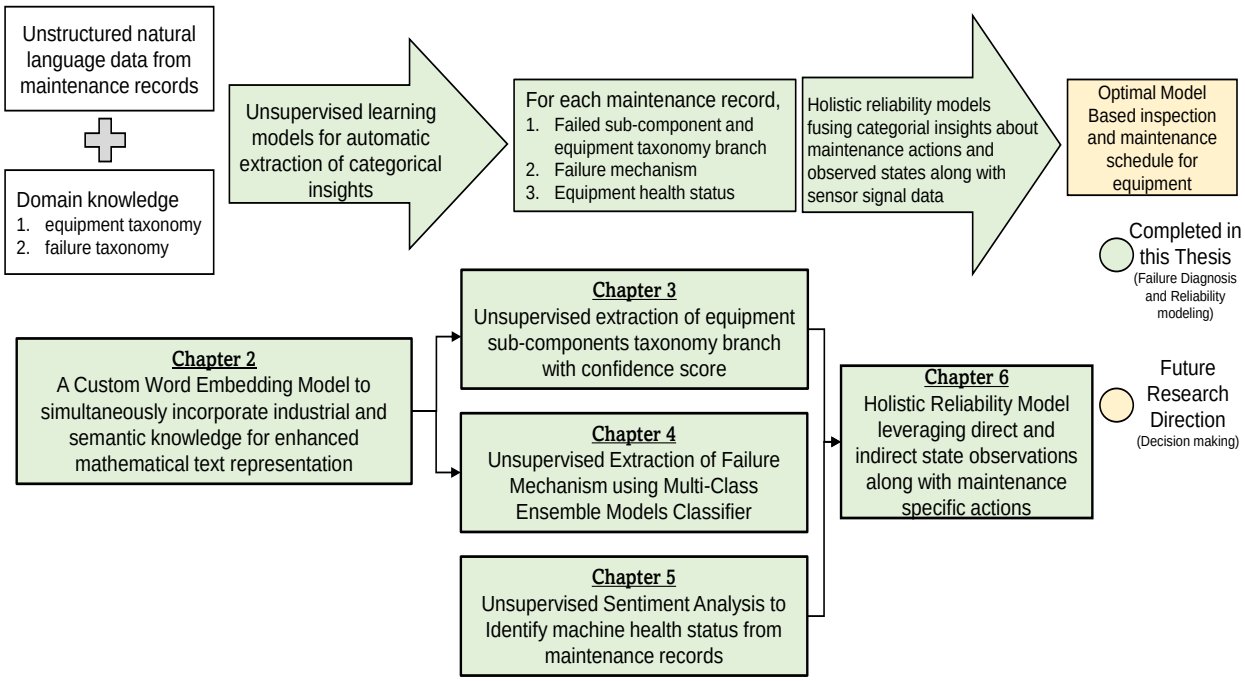


Figure 1.3: Structure of the report

Chapter 2: In this chapter, an unsupervised Custom Word Embedding Model (CWEM) is proposed to generate word embeddings for industrial text analytics. The objective is to

analyze unstructured textual information to identify groups of similar maintenance records, i.e., maintenance records associated with a specific equipment sub-component. The model utilizes two sources of information 1) maintenance records and 2) industrial taxonomy of the equipment hierarchy. A combined loss function is proposed that utilizes word co-occurrence loss and the equipment taxonomy to learn word embeddings jointly. Joint learning helps in reducing the training time when compared to a two-step procedure which incorporates the additional information as a post-processing step. A one-step learning procedure is employed by introducing a new learning parameter that weighs the features to be learned from both bodies of information (contextual and taxonomic). This reduces the number of hyperparameters required to tune while replicating the model instead of the two-step procedure and thus avoids the dependency on hyper-parameter tuning algorithms. The model does not require the taxonomy terms to be contextually co-occurring in a given maintenance record and hence can learn from the complete taxonomy rather than a subset of terms that co-occur. The model is learned in a completely unsupervised manner and avoids dependency on additional resources like WordNet, Wikipedia, or any other form of supervision.

Chapter 3: This chapter proposes a comprehensive methodology to automatically analyze unstructured maintenance records in an unsupervised manner to extract the complete equipment taxonomy branch corresponding to each maintenance event. The method leverages unstructured data in maintenance records and the OEM's equipment taxonomy. The methodology incorporates two algorithms, namely 1) Backward-Forward and 2) Verb Analysis, that are based on syntactic (frequentist) rule-based NLP methods. Semantic ambiguity is resolved by using word embeddings that provide semantic/contextual information. The proposed method combines the two algorithms to generate a single confidence score by leveraging the semantic information contained in word embeddings. A non-parametric density curve is generated by combining the results from both algorithms. The confidence score is a metric indicating the accuracy of the automatically extracted result.

Chapter 4: This chapter proposes methods that identify failure mechanisms from main-

tenance records using multiple unsupervised base multi-class classifiers developed using different sources of domain knowledge. However, to ensemble, the outcomes of such a base classifier in an unsupervised setting is a challenging task. Spectral methods help to learn the proper covariance decomposition of base classifiers' output. However, these methods exist primarily in a binary classification setting. They only leverage discrete labels while learning the covariance structure, thus losing the true sense of discrimination provided by continuous scores of the base classifiers. An Unsupervised Multi-class ensemble classification (UMEC) model is proposed to overcome the described challenges. The model uses an error-correcting-output code scheme of multiple to binary class decomposition while tackling the challenges of class imbalance that affect the spectral estimations' performance.

Chapter 5 This chapter introduces how the equipment health status can be inferred from maintenance records. We propose industrial-specific sentiment lexicons to aid in unsupervised sentiment analysis. A Dirichlet Compound Multinomial model is applied. The model helps to estimate the predictivity of sentiment lexicons. Further, the model also tackles the issue of 'negation' found widely in industrial maintenance records. The chapter discusses how transfer learning models effectively model multitask learning of negation and sentiment analysis.

Chapter 6 This chapter presents a holistic joint reliability model that combines direct state observations extracted from maintenance records with indirect state observations inferred from CBM sensors, while accounting for changes in system dynamics caused by manual actions. The objective of this chapter is novel in the sense that it provides a structured approach to estimate the transition dynamics of a system undergoing different maintenance actions while in operation. Furthermore, we propose that certain maintenance actions contribute to improving the precision of emission models for continuous observation spaces when estimating latent states. To address the challenges posed by missing direct state observations and unobserved degradation dynamics of CBM sensor signals, the chapter relies on the Expectation Maximization algorithm for parameter learning. The Maximization step of the

EM algorithm does not have a closed-form solutions and is therefore inferred using numerical optimization techniques. The chapter also addresses the problem of estimating hyperparameters for the numerical optimizer and initializing parameters for the EM algorithm through hyperparameter tuning routines.

Chapter 2

A Custom Word Embedding Model for Clustering of Maintenance Records

2.1 Focused Abstract

Maintenance records of industrial equipment contain rich descriptive information in free-text format, such as, involved parts, failure mechanisms, operating conditions, etc. Our objective is to leverage this unstructured textual information to identify groups of similar maintenance jobs. We use a natural language based approach and propose a novel custom word embedding model which utilizes two sources of information 1) maintenance records collected from in-field operations and 2) industrial taxonomy, to effectively identify clusters. The advantages of our model include (a) combined use of semantic and taxonomic sources of information for clustering, (b) one step/simultaneous training, which enables knowledge sharing between the two information sources and reduces hyperparameters, and (c) no dependency on third-party data. We demonstrate the efficacy of our model for cluster identification using a real-world dataset. The results show that simultaneous incorporation of semantic and taxonomic information enables accurate extraction of contextual insights for improving maintenance decision-making and equipment reliability.

2.2 Introduction

This chapter is motivated by the unstructured free-text information that is documented by technicians when they perform maintenance actions on industrial equipment. These maintenance records contain rich information (related to equipment components, their condition and failure mechanism) that can aid maintenance technicians with fault prognosis [170], root-cause analysis [126], [171] and maintenance decision-making [165], [5]. For the equipment manufacturer, insights from these maintenance records can help in improving equipment reliability through design changes, and thereby reducing the warranty costs, thus gaining a competitive advantage [124].

While it is common practice in industry to create and store maintenance records, it is impractical to manually review and extract insights from the large quantity and variety of records that are typically available [25]. Automatically extracting useful insights from the maintenance records is, however, not trivial. Consider, for instance, the maintenance records from a company having multiple oil rigs. This dataset comprises maintenance records from a variety of oil rig equipment, each having several sub-units with associated maintainable items, and ultimately the specific parts that underwent repair or replacement. For each maintenance action, a record is available which contains structured as well as unstructured information. Structured information consists of well-defined data such as time of action, type of action (corrective or preventive), rig number etc., whereas, unstructured information is the textual description entered and updated by the technicians such as equipment condition, explanation of the wear or failure, maintenance actions performed, components replaced, etc. Figure 2.1 shows sample maintenance records for mudpump equipment and includes information specific to the sub-units (i.e., pump and fluid end), maintainable items (e.g., discharge module, suction module) and parts (e.g., valve, seat, oil pressure switch) that were involved in these maintenance actions. While such information are a rich source of insights, they cannot be recorded as structured data.

The aim of this research is to create models to analyze maintenance records and automatically extract groups or clusters of records that are similar (e.g., in terms of the failure mechanism or the part that was repaired or replaced). From a practical perspective, our methodology for analyzing and clustering textual data by combining contextual information as well as industry-specific taxonomic information, will assist industrial equipment manufacturers and operators to enhance their ability to perform analysis of maintenance activities, failure types, and equipment components that were impacted. Using this method, we can extract structured information from unstructured data and then conduct quantitative modeling [111] and analysis of system reliability. For example, using this method, from maintenance record, we can create the failure event history for a specific component failing due to a specific failure mechanism. Such history is currently obtained manually by an operator processing the natural language maintenance record. However, there are certain significant challenges in determining clusters based on the analysis of the textual information in the maintenance records. First, the records in the dataset may comprise different types of maintenance information that are kept by different personnel on the oil rig (such as equipment downtime reports created by machine operators, maintenance reports created by technicians, parts reports created by purchasing and inventory personnel) [64]. Second, the descriptions and language used by different personnel can differ even when they are referring to the same maintenance event. For example, in Figure 2.1, it can be seen that phrases like 'wash out' or 'worn' are used interchangeably by industry personnel when referring to excessive wear of fluid end components. Third, the implied meaning of certain words within the maintenance context can be different from that in general use as discussed in [134]. For example, the word 'stick' within the maintenance context most likely implies cohesion as opposed to a piece of wood, thus requiring us to consider the context in which the word is being used. Fourth, the manner in which the clustering of the maintenance records needs to be done also depends upon the desired context and basis for the grouping (e.g., failure mechanism or parts affected). As a result, our model needs to be flexible to accommodate these user-defined

Input Data		
Job ID	**Rig ID**	**Description of Maintenance Action**
J01	R01	Mud pump is currently being run for drilling operations. **Unusual vibrations** noted. **Main Bearing** temp 46c.Mechanics to check temperature when running &also check for vibration
J02	R01	Parts Requested from ICS:, Mud Pump 2 **Lube Oil Pressure Switch Sticking**. Order new switch, Part received and installed
J03	R02	Equipment is out of Service Mud pump # 1out of service to carryout change out of **washed-out discharge module**
J04	R03	**Valve and seat on suction module** cyl #3were changed out due to **worn conditions** and high hours parts. Parts Requested from ICS:

Desired Output	
Failure Mechanism	**Associated Parts**
Vibration-Mechanical failure	Bearing-Power End
Sticking-Mechanical failure	Pressure Switch-Lubrication System
Washed-Material Failure	Discharge Module
Worn-Material Failure	Valve & Seat-Suction Module

Figure 2.1: Sample of input data for text analysis.

preferences.

Within the reliability literature, efforts have been made by researchers to incorporate textual knowledge. In [75] author propose a graphical semi-supervised industry-specific word embedding approach for classifying documents. There exists a large body of methods for unsupervised text clustering within the natural language processing (NLP) domain [144]. However, for clustering methods, the numerical representation of the textual data is highly imperative. Word2vec has proven to be very effective for generating numerical representation of textual data [105]. The idea is to efficiently represent a single word into a one dimensional vector space (also called embeddings) and then use this representation to perform downstream tasks, like clustering. In general use, word2vec is employed on a single set of data, i.e., the representation generated using word2vec only learns the semantic information presented by the contextual words in the document. This limits the introduction of any kind of external context. Naturally, researchers have provided several ways to tackle this shortcoming. For example, [183] propose a two-step procedure to identify the relations

between words, wherein, in the first step they learn embedding, and in the second step, they introduce relation in a supervised way. [169] incorporate the relationships between pairs of words by optimizing different cost functions. However, for these methods, it is unclear how commonality on a sentence or document level can be identified.

[185] use an adaptive clustering module that cluster topics into sub-topics and sample documents relevant to the sub-topics to learn local embeddings. However, their method relies on sampling documents from the internet that are specific to sub-topics present in the taxonomy. In [92], the authors develop a dynamic weighting neural network and use the hypernym-hyponym (parent-child) pairs available in *Wordnet* (a lexical database of semantic relations between words) along with contextual words present in a document. However, the keywords indicating taxonomic/external knowledge may not co-occur together in the same document as can be seen in the illustration of Figure 2.1 (where 'vibration' and 'sticking' do not occur together). The dynamic distance margin model proposed by [183], also uses co-occurring words and their frequency to learn their embeddings. Their loss function minimizes the distance between embeddings of the words that exhibits one of the three relations (co-hypernym, co-hyponym and hypernym-hyponym) and thus neglects semantic/contextual information. The Attract-Repel model proposed by [112] uses a pre-specified word representation/embedding and incorporates the external/taxonomic information over it. The word embeddings generated rely on a pre-trained word representation and incorporate taxonomic knowledge as a post processing step [8] This also increases the number of hyper-parameters, like regularization constants, to retain the semantic information. The model proposed by [8] and [23] learn the word embeddings using the corpus and taxonomy information where the contextual knowledge is learnt using the GloVe loss function [118]. However, their model incorporates the taxonomic guidance for only those words which are contextual and co-occur with each other in a given maintenance log.

Recently, transformer based models have become popular in the NLP literature [166], [39]. However, training them require a large corpus like Wikipedia (≈16 GB of uncompressed

data) to efficiently learn millions of parameters [80]. Also, such models provide contextual embeddings for words which needs to be processed before applying to a down stream task like clustering [144]. Apart from this, some transformer models which incorporate additional taxonomic information train two tasks simultaneously: 1) Masked Language Modeling and 2) Next sentence prediction tasks. To replicate them, the data has to be transformed in a sequential way (premis-hypothesis for [30] or question-answer as in [84]). Other transformer models like [190] require generation of graph based embeddings while model proposed in [119] require an entity candidate selector. The K-adapter model proposed in [172] require additional supervised dataset to induce the taxonomic information.

A supervised method is proposed in [93], where information about hierarchical taxonomy is incorporated by leveraging local and global classifiers trained on annotation for each word in a record. Similarly, in [133], each word in the sentence is annotated and the objective is to predict not only the contexual words but also the corresponding labels while learning the embeddings. Development of tools to assist annotation of maintenance documents is still under progress [141]. The Dict2Vec model presented in [159] learns embedding for various words using definitions provided in dictionaries like Cambridge, Oxford, etc.,. More recently, the work proposed by [95] learns embeddings for taxonomy terms using their definitions in Wiktionary while the work proposed in [4] aims to classify scholarly articles by learning embeddings for each term in knowledge graph using [24].

Based on the literature review, we note that there is a lack of studies which leverage one-step simultaneous learning of information from both industrial equipment maintenance records and industrial domain taxonomy in a completely unsupervised manner without depending on any additional resource apart from maintenance logs and industrial taxonomy. Hence, in this chapter, we propose a novel approach (namely Custom Word Embedding Model (CWEM), which is summarized below:

- We combine Skip-Gram model with standard industrial taxonomy to jointly leverage the information from two sources while learning word embeddings. This helps us to

reduce the training time when compared to a two-step procedure which incorporate the additional information as a post-processing step.

- We employ a one-step learning procedure by employing a new learning parameter which weighs the features to learn from both bodies of information (contextual and taxonomic). This reduces the number of hyper parameters required to tune while replicating the model as opposed to the two-step procedures [112] and thus avoids the dependency on hyper-parameter tuning algorithms.

- The model does not require the taxonomy terms to be contextually co-occurring in a given maintenance log and hence can learn from the complete taxonomy rather than a subset of terms which co-occur.

- The model is learnt in a completely unsupervised manner and avoids dependency on any additional resources like WordNet, Wikipedia or any other supervision.

2.3 Data Description and Textual Clustering

2.3.1 Description of Data Sources

We consider two sources of information. The first is the dataset of maintenance records see Figure 2.1. Let L denote the number of maintenance records, each denoted by JobID in the dataset. Corresponding to each JobID we have the associated system and the description of the maintenance action which was conducted as well as information regarding the portion of the equipment that caused the maintenance action, as well as observations regarding the equipment condition. The second source of information which we incorporate is the standard industry taxonomy associated with the oil and gas industry. In particular, we consider two kinds of industrial taxonomies - failure taxonomy and equipment taxonomy. The failure taxonomy provides a list of commonly used technical terms associated with failures and failure mechanisms in oil rig systems. The equipment taxonomy provides the listing of

different items that comprise the equipment hierarchy and the relationships between them, namely system, sub-unit, maintainable item and component. In the remainder of this chapter we refer to these distinct taxonomy categorizations as classes. A detailed description of the taxonomies is provided in Section 2.5. Now, suppose the records can be grouped into K different clusters (where $K < L$). Thus, our objective here is to appropriately represent the two information sources and identify the clusters based on a similarity measure. We discuss below the basic steps used in clustering of the textual records.

2.3.2 Basics Steps for Clustering of Textual Data

The first step in clustering of textual data is to design an appropriate mathematical representation of these documents. In other words, this mathematical representation is critical in the model's clustering performance. In the literature, several different representations are available like Latent Semantic Analysis [81], Latent Dirichlet Allocation [22] along with word2vec/Skip-Gram. The state-of-the-art Skip-Gram (SG) model is chosen as the basic representation for our study (we provide more details regarding the SG model in Section 2.4.1). Once we have established a mathematical representation, the next step is to identify the clusters present in the corpus. Several clustering algorithms are available in the literature (please see [131] for a recent review). The core idea of clustering algorithms is to minimize the distance of word representations with respect to cluster centers. We use k-means algorithm for the same which uses a pair-wise distance matrix to identify k-centers of each clusters. We revisit the steps used for clustering in Section 2.5.3.

2.4 Model Development

Figure 2.2 outlines the workflow of our proposed framework. The central idea of our approach is that it entails learning information from two information sources simultaneously, namely semantic/contextual information and taxonomical information. In the current chap-

ter, information from a single hierarchy taxonomy, pertaining to the oil and gas industry is incorporated along with the semantic knowledge to learn industry-tailored word embeddings. The industry taxonomy provides a grouping of tokens into different classes where tokens in the same class are similar to each other while tokens in different classes are dissimilar to each other. To incorporate the semantic information, the architecture for the SG model (Section 2.4.1) is used. CWEM, thus, tries to incorporate additional taxonomic information by modifying the loss of the SG model as detailed in Section 2.4.2.

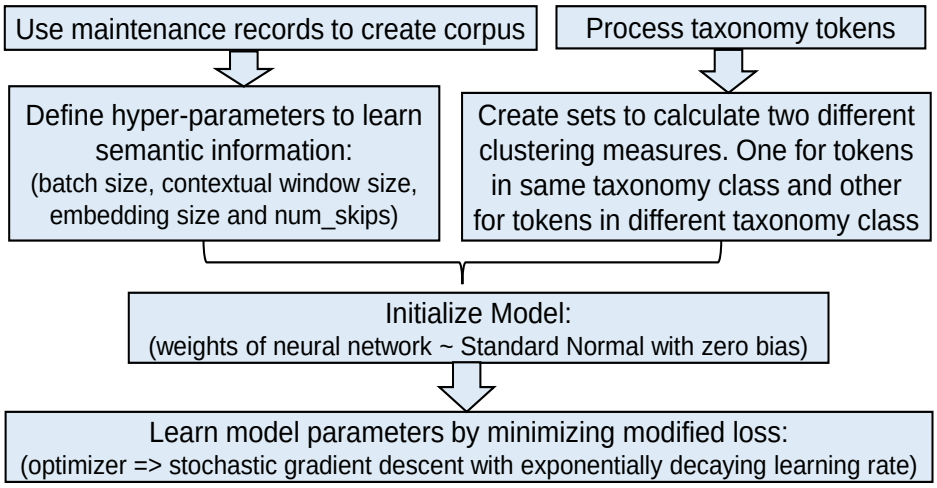


Figure 2.2: A flowchart illustrating the steps involved for proposed framework.

2.4.1 Distributed Representation of a Word

The semantic information is incorporated using the corpus of maintenance records. We use Skip Gram (SG) [105] model for this purpose (please see [132] for a comprehensive explanation). The central idea is credited to the distributional hypothesis which implies that words which represent similar meaning are generally used in the same context [62]. Figure 2.3 provides the basic neural network architecture of the SG model.

The SG model learns a vector representation (known as word embeddings) for an input word by maximizing its affinity with the contextual or neighboring words. For example, consider the sentence “Damage was caused by leaking pump valve". Here, if we consider the

word “leaking" to be the input word, then the rest of the neighboring words are contextual words for it. The number of neighboring words that are used to learn the vector representation for a given input word constitutes the size of the semantic window represented by $|C|$. The objective of the SG model is to generate a vector representation for the given input word which helps to predict the correct context word with a high accuracy. In the same example, if the size of the semantic window is $|C| = 2$, then the (input, context) word pairs become: (leaking, caused), (leaking, by), (leaking, pump), (leaking, valve). The SG model then learns such embeddings for “leaking" which can give a high output score for the true contextual word. The output score is a soft-max score (equation 2.1) which assigns a probability ranking to all words in the vocabulary for being observed as the context word for a given input word.

We denote an input word by w_I and its vector representation by $\mathbf{v_{w_I}}$ in the SG model. Word embeddings are learned using a neural network architecture having an *input layer*, a *hidden layer*, and an *output layer*. The input layer is a $|V|$-dimensional layer corresponding to the dictionary created by all V-distinct words present in the vocabulary $\mathbf{V}$ (as shown in Figure 2.3). The input layer can thus be considered to have $|V|$ nodes where each node represents a placeholder for every input word. Next, the input layer is connected to the hidden layer $\mathbf{h}$ by a weight matrix $W_{|V|\times N}$. Please note that each row of the weight matrix constitutes the embedding vector $\mathbf{v_{w_I}}$ for the corresponding input word which we are learning. Next, the hidden layer is connected to the output layer using a different weight matrix $W'_{N\times|V|}$ (Please note the $'$ notation is used to refer to elements of the output layers). In the SG model, instead of a single output layer, we have $c \in \mathbf{C} = \{1, 2, 3, \cdots, |C|\}$ panels for the output layer. The weights matrix $W'_{N\times|V|}$ connecting the hidden layer to the output layer is shared among all the panels. The rows of the weight matrix linking the hidden layer to the output layer gives the output vector representation $\mathbf{v'_{w_j}}$ of each word. The final output layer thus has $|C| \times |V|$ nodes, where the c^{th} contextual word located at j^{th} position in vocabulary is denoted by $w_{c,j}$ and its output vector representation is denoted by $\mathbf{v'_{w_j}}$. As the weight

matrix $W^{'}_{N\times|V|}$ is shared among all $\mathbf{C}$ panels so is the output vector $\mathbf{v}^{'}_{\mathbf{w_j}}$ for a given context word.

The steps followed while training are as follows: First, the hidden layer transposes the input weight vector $\mathbf{v_{w_I}}$ to give hidden layer vector $\mathbf{h} = \mathbf{v_{w_I}}^T$. Second, dot product is evaluated between the hidden layer vector $\mathbf{h}$ and the output weight vector $\mathbf{v}^{'}_{\mathbf{w_j}}$ for all words $j \in \mathbf{V}$ at each panel $c \in \mathbf{C}$. The dot-products are passed to a soft-max activation layer which gives us the probability of observing a contextual word $w_{c,j}$ for a given input word w_I at the c^{th} context position. The soft-max score is highest for the true contextual word which are passed while training the neural network as compared to other words in the vocabulary. Following the above example, the word 'pump' should have the largest soft-max score corresponding to the word 'leaking' at the $c = 3^{rd}$ panel (because 'pump' is the 3^{rd} context word for 'leaking').

We now formalize the above steps into mathematical equations. The semantic information for each context word $w_{c,j}$ corresponding to the given input word w_I is learned as a multinomial distribution at each panel $c = \{1, 2, 3, \cdots, |C|\}$ and is given by equation 2.1. Here, $w_{O,c}$ is the actual c^{th} context word specified while training the neural network. For example, words like 'caused', 'by', 'pump' and 'valve' would correspond to $w_{O,1}$, $w_{O,2}$, $w_{O,3}$ and $w_{O,4}$ for the input word 'leaking'. As already described, for the SG model on the output layer, instead of outputting one multinomial distribution, we are outputting $|C|$ multinomial distributions, one distribution at each panel, where each multinomial distribution gives the probability of observing the true context word at that position. The symbol $u_{c,j}$ represents the net dot-product of the j^{th} word in c^{th} panel with the hidden layer vector $\mathbf{h}$ and is given by $u_{c,j} = u_j = \mathbf{v}^{'}_{\mathbf{w_j}}{}^T \cdot \mathbf{h}$ for $c \in \mathbf{C} = \{1, 2, 3, \cdots, |C|\}$. The cross-entropy loss maximizes the probability of the observed context words for the given input word (equation 2.2).

$$\textit{soft-max score} = p(w_{c,j} = w_{O,c}|w_I) = \frac{\exp(u_{c,j_{c^*}})}{\sum_{j'=1}^{|V|}\exp(u_{j'})} \tag{2.1}$$

$$
\begin{aligned}
&Cross - Entropy - Loss = \sigma({\mathbf{v}'_{\mathbf{w_{j^*}}}}^T \cdot \mathbf{h}) \\
&= -\log(p(w_{O,1}, w_{O,2}, \cdots, w_{O,|C|}|w_I)) \\
&= -\log(\prod_{c=1}^{|C|} \frac{\exp(u_{c,j_c^*})}{\sum_{j'=1}^{|V|} \exp(u'_j)}) \\
&= -\sum_{c=1}^{|C|} \exp(u_{j_{c^*}}) + |C| \times \log \sum_{j'=1}^{|V|} \exp(u'_j)
\end{aligned}
\tag{2.2}
$$

where j_c^* is the index of the actual output context word occurring at j^{th} index of the c^{th} panel or occurring at the c^{th} position of the semantic window.

The cross-entropy loss updates the output vector $\mathbf{v}'_{\mathbf{w}}$ for each word w_j in the vocabulary at every iteration and is, therefore, intractable to implement in its original form. To tackle this challenge, [105] proposed the idea of negative sampling where negative samples are drawn along with the true/positive context words. Negative samples (present in set $\mathbf{W_{neg}}$) are random words drawn from the vocabulary that have the least probability to occur as contextual words for a given input word. The Noise Contrastive Estimation (NCE) loss is thus given by equation 2.3 and minimizes the probability of predicting the negative context word given an input word.

$$
NCE\ loss = -\log(\sigma({\mathbf{v}'_{\mathbf{w_{j^*}}}}^T \cdot \mathbf{h}) - \sum_{w_n \in \mathbf{W_{neg}}} \log(\sigma(-{\mathbf{v}'_{\mathbf{w_n}}}^T \mathbf{h})) \tag{2.3}
$$

where w_{j^*} is the output word (positive sample), $\mathbf{v}'_{\mathbf{w_{j^*}}}$ is its output vector, $\mathbf{h}$ is the hidden layer vector $\mathbf{h} = {\mathbf{v}_{\mathbf{w_I}}}^T$, σ is the soft-max activation function and w_n are the negative sampled words having $\mathbf{v}'_{\mathbf{w_n}}$ as its vector representation.

2.4.2 Proposed Custom Word Embedding Model (CWEM)

Having learned the contextual/semantic information, we now harness the information from the second source i.e., the taxonomic information. Let $\mathbf{M}$ be the set of classes present in the taxonomy indexed by {'0', '1', '2', $\cdots$, 'm-1'}. For example, 'mechanical failure' and 'material failure' would constitute different classes of the failure mechanism taxonomy. In Figure 2.1, the words 'vibration' and 'stick' belong to the same failure mechanism of

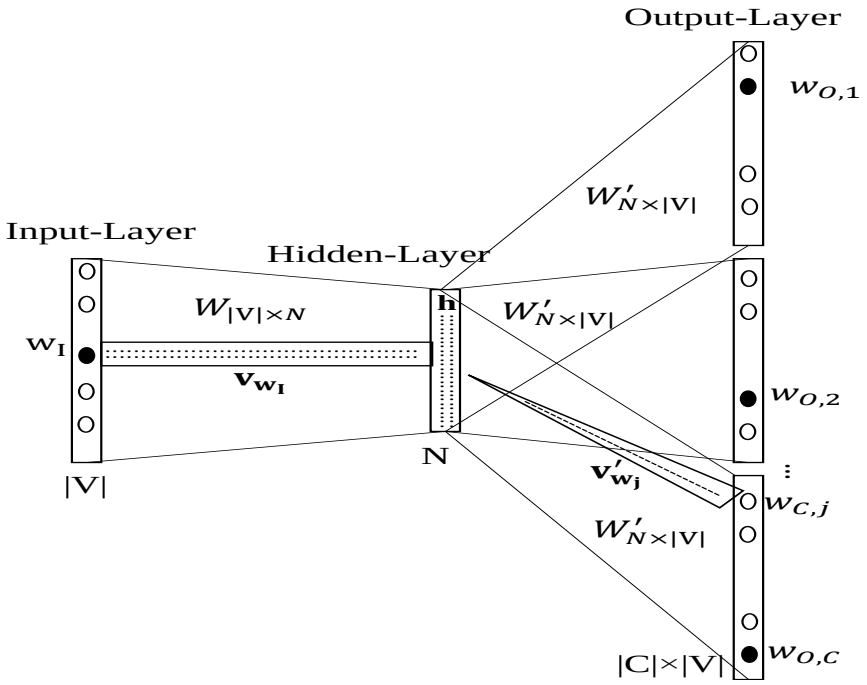


Figure 2.3: Neural Network architecture for the Skip Gram model.

'mechanical failure' while the words 'wash out' and 'worn' belong to 'material failure'. The tokens present in the taxonomy are divided into two different sets as shown in Figure 2.4. The first set contains pairs of tokens belonging to the same class while the second set contains pairs of tokens belonging to the different classes. The major motivation is that the tokens of the same class should have similar word embeddings while the tokens belonging to different class should have dissimilar word embeddings. Next we define the kinds of similarity measure between the words present in different taxonomic classes which would help include taxonomic information in the learnt embedding.

We define two different kinds of clustering measures. The first measure of *Within Class Dissimilarity* (WCD) tries to move points of the same cluster closer to each other while the second measure of *Between Class Similarity* (BCS) tries to move points of different clusters away from each other. WCD measures the dissimilarity between tokens (words) within the same taxonomy class (as indicated for tokens of taxonomy class 1 in Figure 2.4). Intuitively we would like to have the dissimilarity between the embedding vectors of tokens (words) of

the same class to be as low as possible. The measure of dissimilarity is given by the sum of the cosine distance between different tokens belonging to the same taxonomy class and is shown in equation (2.4). Let m_k represent a class in set $\mathbf{M}$. Let w_{T,m_k} define the token (word) from taxonomy class m_k (here the subscript T indicates that the token is also present in the taxonomy). Here, ($\mathbf{v_{w_{T,m_{kp}}}}$, $\mathbf{v_{w_{T,m_{kq}}}}$) denote the word vector representation for words $w_{T,m_{kp}}$ and $w_{T,m_{kq}}$ belonging to the same class $m_k \in \mathbf{M}$.

$$WCD = \sum_{\forall m_k \in M} \sum_{\substack{w_{T,m_{kp}} \in m_k \\ w_{T,m_{kq}} \in m_k}} (1 - \frac{\mathbf{v_{w_{T,m_{kp}}}} \cdot \mathbf{v_{w_{T,m_{kq}}}}}{||v_{w_{T,m_{kp}}}|| \, ||v_{w_{T,m_{kq}}}||}) \tag{2.4}$$

BCS measures the similarity between any two classes $m_k, m_l \in \mathbf{M}$ of the taxonomy (as indicated for the tokens of taxonomy class m_k and m_l in Figure 2.4). Opposed to WCD, for tokens (words) in BCS, we would like to have the dissimilarity between vectors of tokens from different taxonomy class to be as high as possible. For example, we consider the similarity measure between the class 'mechanical failure' (m_k) and 'material failure' (m_l) is given by averaging the cosine similarity between ('leak'$\in m_k$, 'corrosion'$\in m_l$); ('leak'$\in m_k$, 'erosion'$\in m_l$); ('vibration'$\in m_k$, 'corrosion'$\in m_l$) $\cdots$. Equation (2.5) describes the mathematical expression for measuring the between class similarity. Here, $\mathbf{v_{w_{T,m_{kr}}}}$ is the embedding vector for r^{th} word in taxonomy class m_k and $\mathbf{v_{w_{T,m_{ls}}}}$ is the embedding vector for s^{th} word in taxonomy class m_l.

$$BCS = \sum_{\substack{\forall m_k in M \\ m_l \in M}} \sum_{\substack{w_{T,m_{kr}} \in m_k \\ w_{T,m_{ls}} \in m_l}} (\frac{\mathbf{v_{w_{T,m_{kr}}}} \cdot \mathbf{v_{w_{T,m_{ls}}}}}{||v_{w_{T,m_{kr}}}|| \, ||v_{w_{T,m_{ls}}}||}) \tag{2.5}$$

As we propose to learn the semantic and taxonomic information simultaneously, the tokens (words) of the taxonomic class borrow their embedding vectors from the same weight vector of the original SG model. The similarity measures are now combined with the original NCE loss for simultaneous learning of efficient word embeddings. The new loss function is called as the Custom Word Embedding Model loss (CWEM loss) and is given by equation

(2.6) and is defined as a weighted average (weights given by α) of the NCE Loss and the similarity measures given by (WCD and BCS).

$$CWEMLoss = \alpha \times NCE\ Loss + (1\text{-}\alpha) \times \{WCD + BCS\} \tag{2.6}$$

The simultaneous learning in CWEM takes information from SG architecture (shown Figure 2.3) and from the WCD and BCS sets (shown in Figure 2.4). The weighted average of both the losses gives a trade-off between the semantic and taxonomic information to be incorporated in the learned embedding. It allows the user to weigh the extent of taxonomic information the user finds essential to incorporate in the generated word representation. The lower the value of α, the higher the taxonomic information in the generated word representation.

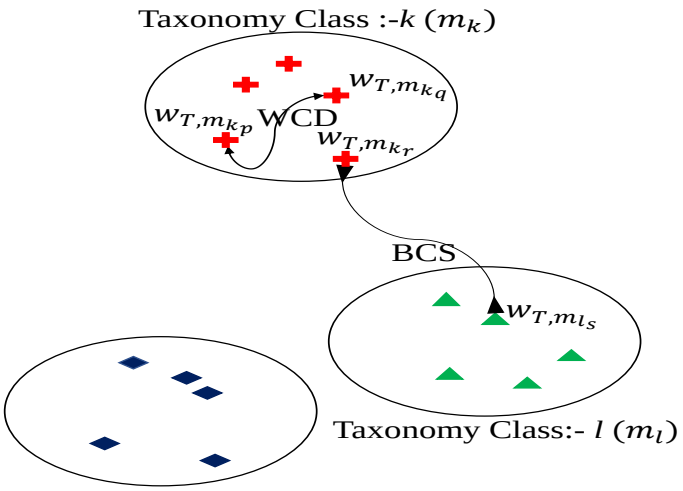


Figure 2.4: Tokens in WCD and BCS sets of the CWEM.

As an illustration, consider the example in Figure 2.5. For the first sentence, the word 'loosened' learns semantic information about its neighboring words due to the NCE loss. Similarly, for the second sentence, the word 'leak' learns semantic knowledge about its neighboring words. However, as there are no matching words in the sentences, the sentences would be dissimilar. To overcome this shortcoming, the external knowledge from the taxonomy is used in the CWEM to increase the similarity between the words 'loosened' and 'leak', thus

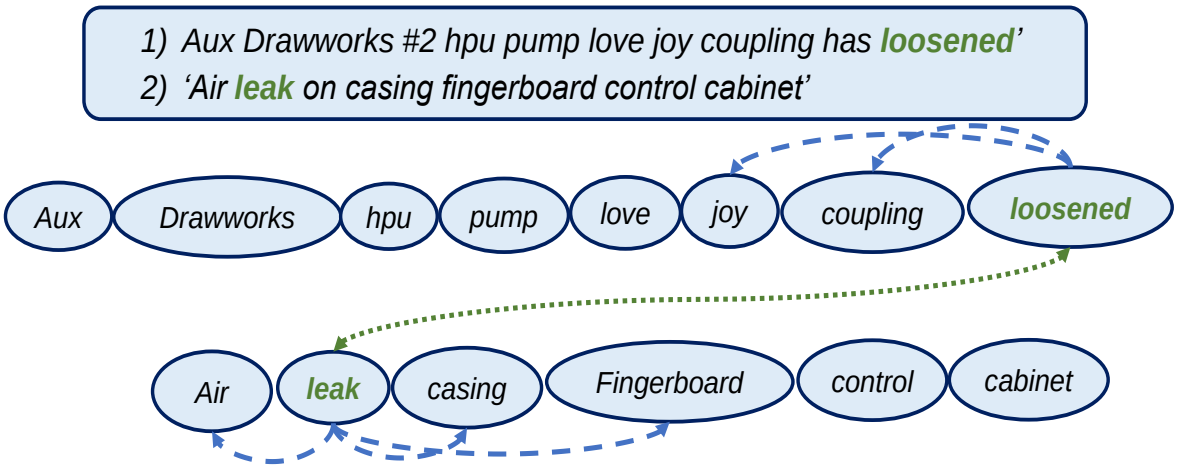


Figure 2.5: Intuition for using two information sources for better representation. The dashed lines (– – –) represent learning through NCE loss for the semantic information while the dotted lines ($\cdots$) represent learning through the similarity of taxonomic tokens to incorporate taxonomic information.

making the two sentences similar to each other. The implementation of the CWEM is given in Algorithm 1.

2.5 Case Study (Dataset From Oil Rigs)

2.5.1 Available Maintenance Records

For our analysis, we use data obtained from eight systems (like High Pressure Mud System (mud-pump), Drill Rig Hosting System etc.,) available across 31 oil rigs. The maintenance records shown in Figure 2.1 are a representative sample of the dataset. We have 11682 maintenance logs (L), and the size of the vocabulary (V) is 12987.

2.5.2 Experimental Settings

For the case study, we demonstrate the application of our proposed model using two different Settings. In Setting 1, we use the failure mechanism taxonomy and try to cluster maintenance records that are associated with similar failure mechanisms. For Setting 2,

Algorithm 1: Initializing and implementing CWEM for a batch

Input $= \alpha, N, window\_size, num\_skips, lr, \mathbf{w_I} \in V$

▷ *lr is the learning rate for the Gradient Descent*

1: Initialize weights for different layers
   First: $\mathbf{v_{w_I}} \in W_{|V|\times N} \rightarrow \mathcal{N}(0,1)$
   Hidden: $\mathbf{h} \rightarrow \mathbf{v_{w_I}}^T$
   Output: $\mathbf{v^{'}_{w_O}} \in W^{'}_{N\times|V|} \rightarrow \mathcal{N}(0,1)$
2: Create the WCD sets, having words from same taxonomy class $m^{'} \in \mathbf{M}$ in same set
   $WCD \rightarrow \{w_{T,m_{k_p}}, w_{T,m_{k_q}} \in m^{'} \forall(m_k \in \mathbf{M})\}$
3: Create the BCS sets, having words from different taxonomy classes $m_k, m_l \in \mathbf{M}$.
   $BCS \rightarrow \{w_{T,m_{k_r}} \in m_k, w_{T,m_{l_s}} \in m_l \forall(m_k, m_l \in M)\}$
4: Calculate $NCELoss$ using equation 2.3
5: Calculate WCD using equation 2.4
6: Calculate BCS using equation 2.5
7: $CWEMLoss = \alpha \times NCELoss + (1-\alpha) \times \{WCD + BCS\}$
8: $lr\_ex\_decay \rightarrow lr \times decay\_rate^{\{\frac{global\_step}{decay\_step}\}}$
9: Optimizer $\rightarrow Gradient\_Descent$ (minimize $CWEMLoss$ using lr_ex_decay)

we use the hierarchical equipment taxonomy where the equipment is branched into various sub-units which are further branched into maintainable items and parts. In Setting 2, we try to cluster maintenance records that are associated with the same sub-units, maintainable items or parts.

Setting 1: On Basis of Failure Mechanism

Here, our aim is to identify clusters of maintenance records which describe maintenance events caused by similar failure mechanisms. The failure mechanism taxonomy is adapted from [70] and a few sample rows are shown in Table 2.1 . We have five failure mechanisms in the taxonomy as denoted in [70]. The number of terms in the taxonomy are summarized in Table 2.2. We consider uni-gram (single word) tokens for this Setting. Tokens in the taxonomy are processed using steps demonstrated in Figure 2.6a. The lementized tokens are obtained using a third party package in Python. For any tokens that are not accurately converted, we manually add extra tokens which represent the base form of the words (e.g., for the token 'Leakage', we also add 'Leak'). The taxonomy tokens are then combined with each other to form elements of the WCD set which are pairwise combination of tokens from

Table 2.1: A sample of taxonomy obtained from [70]

Failure Mechanism	Subdivision	Description
Mechanical	Leakage	External and internal leakages, either liquids or gases. If the failure mode at equipment unit level is leakage, a more causal-oriented failure descriptor should be used wherever possible
	Vibration	Abnormal vibration. If the failure mode at equipment level is vibration, a more causal-oriented failure descriptor should be used wherever possible
..	..	..

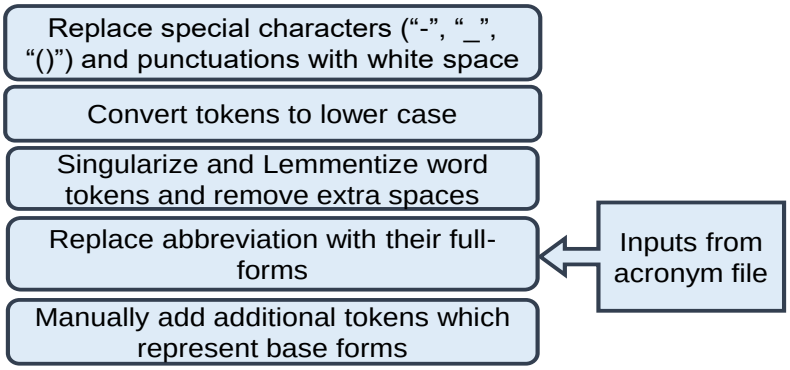
Table 2.2: Number of terms in failure mechanism taxonomy groups

Failure mech.	Mechanical	Material	Hydraulic	Electrical	Control
# of tokens	28	25	10	10	6

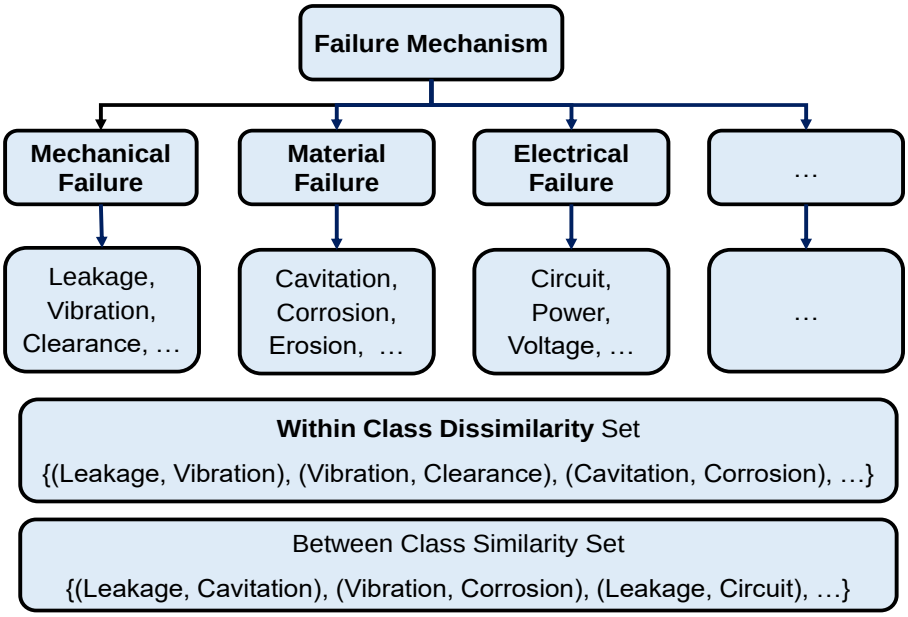
the same class. Similarly, the BCS set is formed by pairwise combination of tokens from the same class. The formation of the sets is illustrated in Figure 2.6b.

Setting 2: On Basis of Mud-Pump Equipment Taxonomy

In this Setting, our aim is to identify the cluster of maintenance records which describe maintenance activities concerned with same sub-units (or constituents components) present in the mud-pump taxonomy. The sample of the mud-pump taxonomy is shown in Table 2.3. The taxonomy is created as a bill of materials (BOM) for the equipment. As the current model incorporates information from a single hierarchy taxonomy, we collapse the multi-hierarchy taxonomy of mud-pump to a single hierarchy by collapsing the maintainable items and parts of each sub-unit into a set which we term as sub-parts. We have five classes in the equipment taxonomy. The number of terms present in each class are shown in Table 2.4. The processing step Figure 2.7a for the taxonomy has an additional step as there are many sub-parts which have multiple words in them and are thus converted appropriately into n-grams before training the model. Word tokens present in multiple sub-branches are removed so that each token can have membership exclusively to a single class. Sets of tokens



(a) Processing failure mechanism taxonomy.



(b) Formation of sets for Setting 1.

Figure 2.6: Setting 1 data preparation.

Table 2.3: A sample of equipment taxonomy for mud-pump

Equipment Unit	Sub-Unit	Maintainable Item	Parts
Mud-Pump	Fluid End	Manifold	Discharge Manifold
			Suction Manifold
		Piston and Liner	Piston

Table 2.4: Number of terms in equipment taxonomy groups

Sub-units	Drive Motor	Fluid End	Motor Cooling	Power End	Tool Control
# of tokens	10	22	14	19	10

required to develop the model are then created in the similar fashion as discussed in Section 2.5.2 and shown in Figure 2.7b.

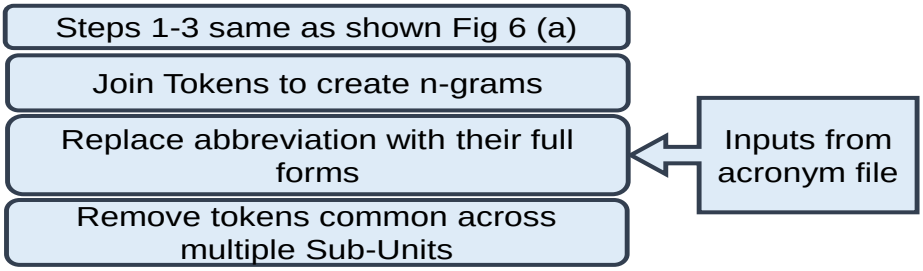
2.5.3 Experimental Setup

We train the word embeddings using the CWEM for each Setting separately and generate separate word embeddings for each Setting. As shown in Algorithm 1, the embedding vector for each word is initialized using a standard normal distribution. To train the algorithm, we choose the word embedding lengths $N = 100$. The context window length is chosen to be 10 ($C = 10$; 5 context/target words to the left and right of the input word) and the batch size is chosen to be 256. We set the hyperparameter $num\_skips = 8$ to specify the number of words to be randomly sampled from the context window while learning embedding. To exponentially decay the learning rate of the gradient descent optimization, a decay step of 200 and a decay rate of 0.875 is selected. We use three competitive models to compare the performance of clustering of our method as described below:

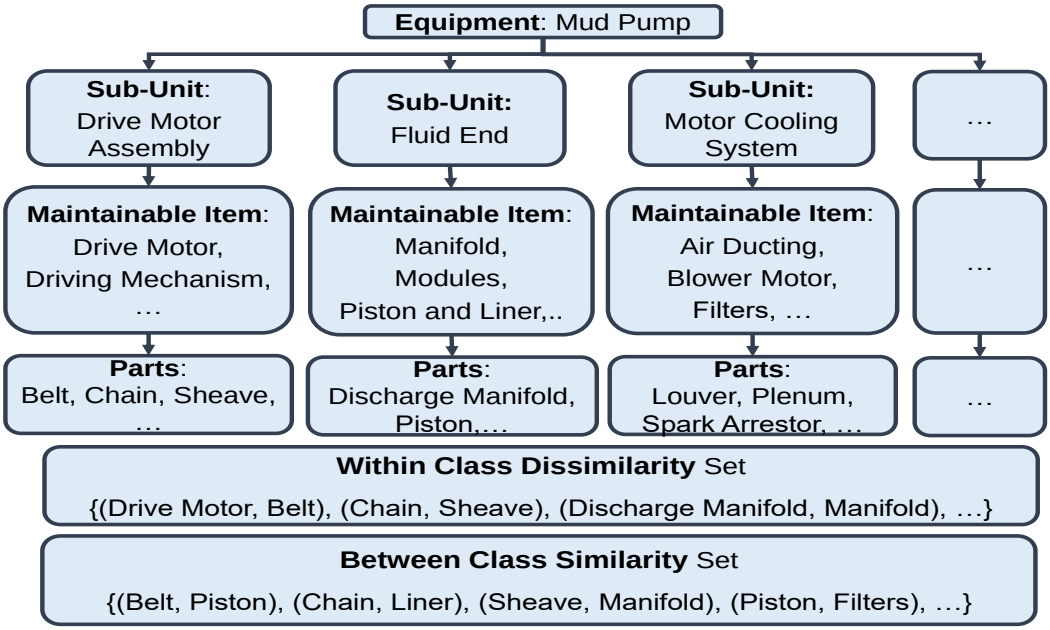
Competing Models

1. **CWEM with $\alpha = 1$/SG model**: The CWEM with α= 1.0 represents the original skip-gram model in our setting.

2. **GoogleNews**: Google‘s set of global word embeddings which contain pre-trained



(a) Processing mud-pump taxonomy.



(b) Formation of Sets for Setting 2.

Figure 2.7: Setting 2 data preparation.

word embeddings developed on several Google news article stored in the GoogleNews-vectors-negative300.bin.gz

3. **Attract-Repel Model**: The Attract-Repel model proposed by [112] uses pre-trained word embeddings and incorporates additional information regarding the pairs of words which are either synonyms or antonyms. Please note that, for Attract-Repel model, elements of WCD set forms synonym set while elements of the BCS set forms antonyms set.

4. **Dict2Vec**: The Dict2Vec model presented in [159] learns embedding for various words using dictionary definitions provided in multiple dictionaries like Cambridge, Oxford, etc., to incorporate the meaning of each word while learning word embeddings. We utilize the pre-trained word embeddings for Dict2Vec with embedding size of 100 and also use Dict2Vec as a base embedding for Attract-Repel Model.

5. **Joint Rep using GloVe**: The joint learning model that we incorporate is proposed in [8], [23]. The authors proposes to use a single step joint learning model which incorporates the contextual information by using the loss function for GloVe [118]. The GloVe loss function is constructed using a term co-occurrence matrix which incorporates terms that co-occur with each other in a maintenance record. The taxonomic knowledge is incorporated by minimizing the distance between the terms that belong to the same taxonomy branch and also co-occur with each other in the record.

Implementation

A clustering experiment is conducted by forming clusters of the selected documents (maintenance records) using the competitive models and CWEM with $\alpha \in \{0.2, 0.35, 0.5, 0.65, 0.8 \text{ and } 1.0\}$. To perform the experiment, documents from the processed corpus are selected for the two different Settings. We select a random sample of $L = 100$ documents for Setting 1. We manually identify the true failure mechanism indicated by each document and mark

it as the true label for the given record. For Setting 2 as well, we select a set of $L = 100$ documents indicating the sub-unit which was repaired for the mud-pump. The sub-parts and sub-units for documents in Setting 2 can be assumed to occur primarily as 'nouns' in the documents, and hence we filter the selected documents to only include 'noun' and 'verb' before performing the experiment in order to remove noisy words. Such Part of Speech ('POS') targeted filtering is not feasible for Setting 1 as failure mechanisms could also occur as adjectives describing the condition of the part requiring maintenance.

Next, to cluster the documents in each dataset, the pairwise distance matrix is generated for maintenance records using word embeddings from each model (competitive and CWEM). The pairwise distance matrix measure is the pairwise distance between documents contained in the dataset and is a square matrix of dimension $(L \times L)$ for each dataset. To generate the pairwise distance matrix, one approach would be to just simply average the word embeddings of all words in the maintenance record and calculate the distance between these averaged embeddings. However, this will be very naive and would not be able to incorporate the essential information in the maintenance records because of the noise present in the dataset. To overcome this limitation, the pairwise distance between documents is measured using the *Words Mover Distance* (WMD) algorithm proposed by [78]. The WMD distance measures the dissimilarity between two text documents as the minimum amount of distance that the embedded words of one document need to "travel" to reach the embedded words of another document. The distance matrix is then supplied as an input to the k-means algorithm in order to determine the cluster label for each document. We use k-means because we have predetermined numbers of clusters for our experiment. Further, the k-means algorithm is highly efficient having comparable performance with other available clustering algorithms [131]. Using the predicted label from k-means and the manually marked label for each document evaluation metrics are generated.

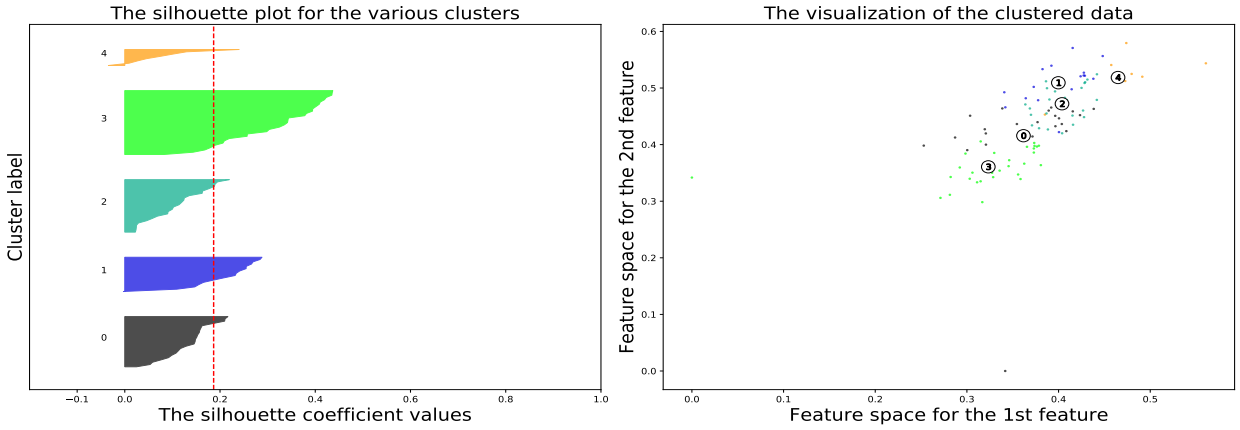
2.5.4 Evaluation Criteria and Metrics

To evaluate the models' performance, we use the Adjusted Ranked Index (ARI) and the Silhouttes score (SSc) as the metric. The ARI is a similarity measure between two clustering schemes which considers all pairs of samples and the count of pairs that are assigned in the same or different clusters in the predicted and true clustering scheme. The ARI is calculated between the predicted label and manually marked label. High ARI would mean that high association exists between the predicted labels and the true labels. The SSc measures the consistency of the clusters formed and indicates how well the element is matched to its own cluster as opposed to the neighboring clusters. The range of the SSc is [-1, 1], and the higher the SSc, the better the efficiency of the clusters formed.

2.5.5 Results

The ARI and SSc scores obtained from the clustering analysis are discussed here. Figure 2.8 is generated during silhouettes analysis for the CWEM with $\alpha = 0.65$, SG model and Attract-Repel (SG) model. The plots on the left panels demonstrate the thickness of the cluster formed and the plots on the right panel indicate the clustered points on a 2-dimensional space. It can be seen from the plots on the right panel the clusters for CWEM with $\alpha = 0.65$ model are well separated as compared to other competitive models. Also, the average SSc (indicated by dashed red line in the left panel) is higher for CWEM with $\alpha = 0.65$ model. The training time for 100 batches of Attract-Repel model with Skip-Gram embeddings is 18 seconds, whereas, it took 0.2 seconds to train 100 batches of CWEM.

The results for both the Settings of all the models are summarized in Table 2.5. It can be observed that for a random sample of 100 documents the ARI and SSc for all CWEM with $\alpha < 1.0$ is better than the competitive (Attract-Repel, Dict2Vec, Glove) and baseline (word2vec, Google News) models. The results for the competitive (Attract-Repel, Attract-Repel (Dict2Vec), Joint Rep using GloVe) model are however better than their corresponding



(a) Silhouettes analysis for CWEM with $\alpha = 0.65$ model

(b) Silhouettes analysis for SG (CWEM with $\alpha = 1.0$) model

(c) Silhouettes analysis for Attract-Repel (SG) Model

Figure 2.8: Comparison of silhouettes analysis for CWEM with $\alpha = 0.65$ model, SG (CWEM with $\alpha = 1.0$) model and Attract-Repel(SG) model

Table 2.5: Comparison of model performance

	Setting 1		**Setting 2**	
Model Name	**ARI**	**SSc**	**ARI**	**SSc**
CWEM $\alpha = 0.20$ scaled	0.211	0.104	0.191	0.136
CWEM $\alpha = 0.35$ scaled	0.295	0.104	0.321	0.136
CWEM $\alpha = 0.50$ scaled	0.336	0.105	0.342	0.139
CWEM $\alpha = 0.65$ scaled	0.338	0.106	0.42	0.187
CWEM $\alpha = 0.80$ scaled	0.356	0.098	0.423	0.181
CWEM $\alpha = 1.0$ scaled/SG	0.028	0.079	0.096	0.093
Attract-Repel (SG)	0.059	0.081	0.17	0.105
Google News	0.052	0.09	0.116	0.091
Attract-Repel (Google News)	0.106	0.087	0.129	0.094
Dict2Vec	0.088	0.048	0.141	0.089
Attract-Repel(Dict2Vec)	0.155	0.064	0.131	0.088
Joint Rep using GloVe	0.073	0.084	0.132	0.122

baseline models. For both the Settings, it can be observed that ARI and SSc is better for $\alpha > 0.5$. The ARI and SSc for Setting 2 are highly distinct for CWEM as compared to the competitive models. This distinction can be attributed to the POS filtering carried out on the documents for Setting 2 which results in a higher frequency of taxonomy terms in the documents as compared to the other terms. The CWEM with $\alpha = 0.65$ performs better in terms of SSc for both the Settings. The Joint Rep using GloVe model does not perform well in the experiments. This can be attributed to the requirement of co-occurring taxonomic terms in the maintenance records. The Attract-Repel (Dict2Vec) model performs pretty well for Setting 1 as taxonomy terms in Setting 1 contains terms like 'leak', 'corrosion' etc., which are generic in nature and have the same contextual meaning in various English dictionaries. However the model performance decreases when a more specific industrial taxonomy is used as can be seen in Setting 2 results. To better understand the convergence of ARI for each model, as well as the consistency of the models, we consider bootstrapping next.

For bootstrapping, we use the same dataset of selected 100 documents from which we create a balanced set of 50 documents where we sample $L = 10$ documents of each class from the initial set. This process is repeated for $B = 100$ iterations, and in each iteration, we calculate the ARI for the sampled set. The mean value of the ARI is calculated to

analyze the performance of each model and is tabulated in table 2.6. By observing the bootstrapped results, we infer that, for documents having high frequency of non-taxonomy tokens (as in the case for maintenance records of Setting 1 and Setting 2), it is better to incorporate semantic knowledge to a higher extent by keeping $0.5 < \alpha < 1.0$ to allow for better clustering. However, when it is intuitive that the number of taxonomy tokens would occur at high frequency in the documents under consideration (as in Setting 2), the model performs well even when taxonomic information is incorporated at a higher extent with values of $0.2 < \alpha < 0.5$. We illustrate the results obtained for bootstrapping in Figure 2.9. It can be seen that the mean of bootstrapped ARI for CWEM with $\alpha = 0.65$ is significantly larger than the mean of the other competitive models.

Table 2.6: Results from bootstrapping

Model Name	**Mean ARI Setting 1**	**Mean ARI Setting 2**
CWEM $\alpha = 0.2$	0.1263	0.2669
CWEM $\alpha = 0.35$	0.1282	0.3689
CWEM $\alpha = 0.5$	0.1268	0.3299
CWEM $\alpha = 0.65$	0.1513	0.3605
CWEM $\alpha = 0.8$	0.1398	0.3560
CWEM $\alpha = 1$ scaled/SG	0.0436	0.1742
Attract-Repel (SG)	0.1000	0.2506
Google News	0.0863	0.1188
Attract-Repel (Google News)	0.1251	0.1330
Dict2Vec	0.0882	0.1405
Attract-Repel (Dict2Vec)	0.1452	0.1509
Joint Rep using GloVe	0.0986	0.1308

2.6 Conclusion and Future Directions

In this chapter, we have proposed a novel distributed representation of textual description available in maintenance records. We use information from two sources (namely semantic information and taxonomic information) to efficiently learn the word distribution, and a weighting parameter governs the learned representation. In terms of identifying clus-

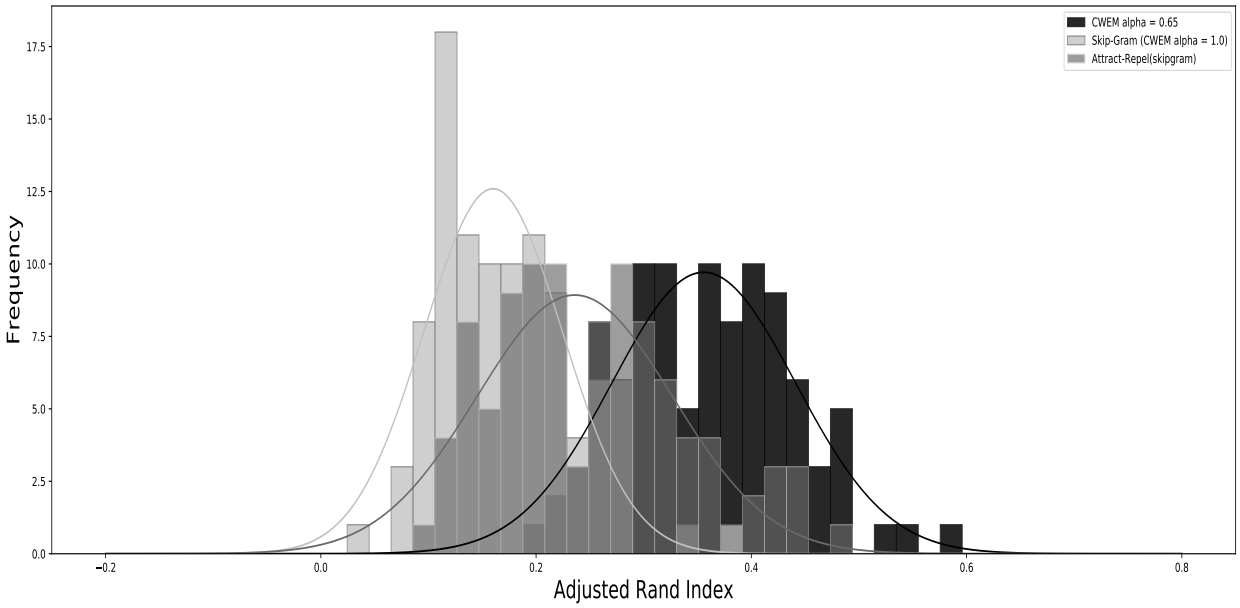


Figure 2.9: Bootstrapping results for CWEM with $\alpha = 0.65$, SG (CWEM with $\alpha = 1.0$) and for Attract-Repel (SG) model

ters, our proposed methodology using CWEM outperforms other models. The model also demonstrates that simultaneous incorporation of taxonomic and contextual/semantic information helps in developing efficient numerical representations for words as compared to the traditional two-step procedure. There are two key observations in this regard: First, the classification of documents depend on the secondary source of information, i.e., the clusters formed will be influenced by the taxonomy which has been used. Second, the parameter α provides a simple way to control the degree of influence exerted by the secondary source of information.

In practice, for a new dataset, the parameter $\alpha = 1$ would yield clusters by using information present only within the documents, thus providing insights about the groups from a general perspective. By gradually decreasing the value of α in the presence of an available taxonomy, the model will identify clusters based on the taxonomy. Therefore, the clusters will be similar based on the provided information. In this manner, the way of combining different sources of information to identify groups would yield similarities or differences with respect to the provided taxonomy. That is, two documents can belong to one cluster with

respect to one taxonomy, while they may belong to different clusters with respect to another taxonomy.

The work in this chapter can be used by OEMs in a variety of applications. For example, the clusters of documents in Setting 1 would represent records from different failure mechanisms, this information can help OEMs to better stratify their event data before performing reliability studies. For example, the authors in [111], analyze clinical notes of patients for unit-matching in survival analysis. In industrial domain a similar approach can be used for identifying stratification in reliability data using the clusters obtained by CWEM in Setting 1. For Setting 2 the clusters represent different components or sub-units of the equipment that required maintenance, thus, clustering them can help OEMs have targeted improvements in equipment design or changes to their warranty policies or spares parts inventory management. For future research, the current model could be extended to incorporate information about the lexical parent-child relations present in multi-hierarchy taxonomies in a straightforward manner without having the dependency on additional resources/supervision.

Chapter 3

Confidently extracting hierarchical taxonomy information from unstructured maintenance records of industrial equipment

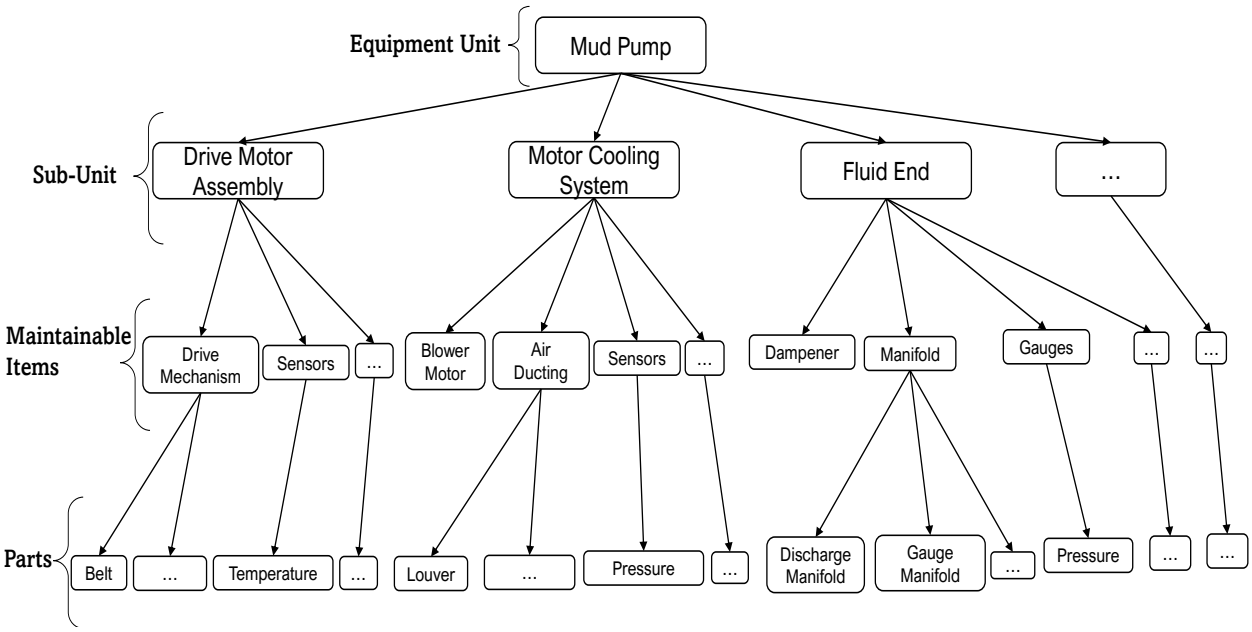
3.1 Focused Abstract

Maintenance records of complex industrial equipment contain a large amount of unstructured data (e.g., technician notes) pertaining to repair actions and associated equipment sub-components, degradation conditions, failure mechanisms, etc. These unstructured data can yield valuable insights to improve the equipment design and maintenance plans, resulting in higher productivity and lower operating costs. Since manual review of information is time consuming, companies make limited use of the maintenance records. To address this opportunity, we propose a taxonomy-guided method for automatically analyzing the unstructured data and inferring critical information, specifically the hierarchy of the equipment's sub-assemblies and constituent parts that malfunctioned or failed during a breakdown event.

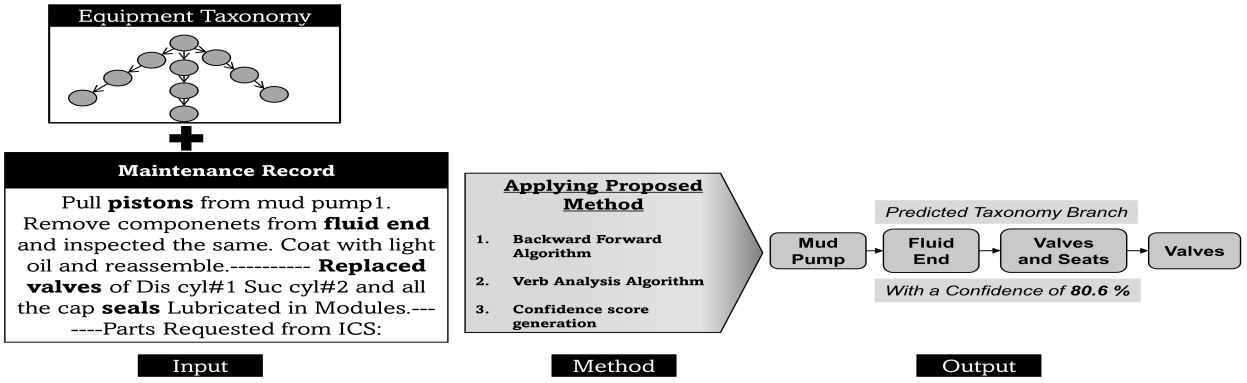
Our method leverages syntactic (related to word frequency) as well as semantic (related to word co-occurrence and their meaning) knowledge. A novel contribution of our work is that we provide a confidence score for the information inferred by our method. Only the maintenance records which receive a low confidence score will require manual review to confirm the automated method's results, thus ensuring minimal use of human resources. We demonstrate the performance of our method using a real-world data set from equipment used in oil rigs.

3.2 Introduction

Every complex industrial equipment can be viewed as a hierarchical system of multiple sub-assemblies and their constituent components which operate synchronously. The relationships between the various sub-assemblies and other sub-components are generally defined in the bill of materials for such a system and are represented by the equipment taxonomy. Each branch in the taxonomy shows the association between components at different levels of the assembly structure of the equipment, as shown in Figure 3.1. In this work, we focus on hierarchical systems having two or more levels of hierarchy in the equipment taxonomy. These levels, from top to bottom, are referred to as '*Equipment*', '*Sub-unit*', '*Maintainable Item*' and '*Part*'. For convenience, the subordinate items at every level of an equipment taxonomy-branch are collectively referred to as '*sub-components*'. A breakdown of any sub-component requires repair action to be taken specifically for that sub-component and this affects the associated equipment taxonomy branch. In Figure 3.1a, an example of a taxonomy-branch is Mud-Pump (*Equipment*) > Drive-Motor-Assembly (*Sub-Unit*) > Drive-Mechanism (*Maintainable Item*) > Belt (*Part*). Ontologies have proven useful for various industry applications such as representation of product assembly structures and product life-cycle management [69]), ([77], [74]). Our focus is specifically on leveraging equipment taxonomy for analysis of maintenance activities within the product life-cycle to infer the taxonomy-branch that malfunctioned.



(a) Sample of OEM equipment taxonomy.



(b) Demonstration of inputs and outputs for the proposed method.

Figure 3.1: Research goal

Technicians document their findings in the form of unstructured free text while inspecting or repairing an equipment. These maintenance records can contain a variety of information including failure mechanism [170], broken sub-components, operating environment etc., that could be used to assist failure diagnosis and prognosis. Accurately determining the sub-components of a taxonomy-branch that failed within an equipment is very valuable to the original equipment manufacturer (OEM) as well as the equipment owner. OEMs use this information to make appropriate equipment design changes that can lead to improved performance and reliability. This information also assists the OEM to update warranty policies

appropriately to suffer minimal loss. Using the information about frequently failing hierarchical taxonomy-branches, equipment owner can focus their maintenance efforts specifically towards these branches. This will help them to improve operation procedures, maintenance practices and spare parts inventory management for the vulnerable taxonomy-branches and would increase the overall equipment uptime and in turn reduce the operating expense. Another essential reason to consider the hierarchical taxonomy-branch is that the same sub-component would occur in different taxonomy-branches and even at different hierarchies. For example a 'seal' is a 'maintainable-item' of 'fluid-end' and also a 'part' for the 'fluid-end → 'piston-and-liner' taxonomy-branch.

While maintenance management systems allow for certain types of information to be documented in a structured format (e.g., by selecting from a list of predefined categories), a common challenge is that such categorical data are often incomplete or inaccurate. [64] states that about 37% of maintenance records contain inaccurate categorical data, such as incorrect attribution of failed sub-components. Further, researchers like [142] propose supervised learning algorithm that rely on manually labeled maintenance record to train a supervised classifier for root cause identification. [146] generate binary labels by assuming the presence of a component in service data as an indicator of its failure, and train a supervised binary multi instance learning classifier to sequentially mark maintenance records as failure events. Similarly, [90] make use of both manually labeled and unlabeled inspection reports to generate semi-supervised conditional random field model for entity recognition. However, in this research we do not have explicit labels describing failed taxonomy-branch associated with a given maintenance record to train such a (semi) supervised classifier. The unavailability of explicitly labeled maintenance records for training a classifier to identify the complete taxonomy-branch of a failed sub-component makes our problem unsupervised in nature. To aid the analysis of unsupervised text data, Ontologies (like bill of material/equipment taxonomy) have proven to be a promising resource that can induct domain specific knowledge [44] while learning unsupervised models. [76] discuss about how Ontologies can help *closed*

loop life-cycle management for products.

While the unstructured data in maintenance records contain valuable information, however, extracting essential information such as the equipment taxonomy-branch associated with unstructured and unlabeled maintenance record is challenging in practice and present several challenges like:

- **C1: Unavailability of labels:-** Maintenance record are not labeled with the associated taxonomy-branch of failed sub-components (unsupervised learning problem)

- **C2: Unstructured text data without template:-** Maintenance records are unstructured and do not follow any fix template as opposed to system logs that have fixed templates Figure 3.2c. The unstructured text data also use synonyms and abbreviations to refer sub-components such as ('swab' → 'piston'), ('variable-frequency-drive' → 'vfd', 'drive'). However, the OEM's equipment taxonomy lacks alternate references of sub-components.

- **C3: Multiple local context of a word:-**Unstructured data require contextual reasoning as the meaning of a specific word depends on the context of its use. For example, 'pump' can refer to the action of pumping or an equipment sub-component. Thus, incorporating contextual or semantic information while analyzing maintenance records is essential [134], [156], [85], [69].

- **C4: Incomplete information in text about failed equipment taxonomy-branch:-** The information contained alone in the maintenance record is usually incomplete and inadequate to determine the equipment taxonomy-branch. As shown in Figure 3.2a, the technician's indicate that 'louvers' need replacement, but, to infer that the associated taxonomy-branch is 'motor-cooling-assembly' → 'air-ducts' → 'louvers' alone from maintenance record is not possible. Thus, using ontologies like equipment taxonomy help identify the complete taxonomy-branch. [74], [69], [161], and [186].

- **C5: Presence of misleading taxonomy tokens:-** While preparing the maintenance record, the technician may document multiple parts which were inspected but did not fail. This infest the maintenance record with noisy tokens that did not actually need repairs. In Figure 3.2b, while 'valves' and 'seals' are mentioned, the seals did not need any repair. Thus, applying conventional Named Entity Recognition methods would be incorrect to extract the complete taxonomy-branch of failed sub-component.

- **C6: Non-unique instances of taxonomy tokens:-** As mentioned above certain sub-components that appear in multiple branches of equipment taxonomy, makes the identification of correct taxonomy-branch difficult. In Figure 3.1a, 'sensors' appear as a *Maintainable Item* for the 'drive-motor-assembly' *Sub-Unit* as well as the 'motor-cooling-system' *Sub-Unit*. Thus, only observing 'sensor' does not allow to infer the specific equipment taxonomy-branch that failed.

Apart from the challenges (C1-C6) mentioned above, the proposed solution should also provide confidence level associated with extracted taxonomy-branch of failed sub-components for each maintenance record. The confidence score associated with the predicted taxonomy-branch helps in deciding whether or not a manual review of the maintenance record is needed to confirm the accurate taxonomy-branch. As discussed below in Section 3.3, the existing research literature lacks an unsupervised framework that can overcome these challenges and fulfill these requirements simultaneously. Thus, to address these research gaps, this paper focuses on automatically analyzing unstructured maintenance records in an unsupervised manner, without labeled data, to determine the complete equipment taxonomy-branch corresponding to each breakdown event by leveraging OEMs equipment taxonomy as shown in Figure 3.1b. The solution methodology proposed handles challenges mentioned in C1-C6. Our methodology incorporates two novel algorithms, namely 1) Backward-Forward and 2) Verb-Analysis, that are based on syntactic (frequentist) rule-based methods in Natural Language Processing (NLP). However, since frequentist methods can be misleading in analyzing maintenance records, semantic techniques like word embeddings are also used to combine

the results of the two frequentist algorithms and generate a single confidence score using non-parametric density curve. The proposed confidence score serves as a metric to classify the maintenance records into ones whose automated solution results can be fully trusted and the ones whose results may need confirmation by manual review. We also discuss the performance of our methodology using a real-world data set.

3.3 Literature Review

NLP methods are applied to meet several industries needs [175]. In Table 3.1, we categorize five categories of methods relevant to our problem and identify the limitations of each existing method in tackling the challenges (C1-C6) mentioned in section 3.2.

The first research category is related to *keyword extraction and named entity recognition (NER) for text summarization* [145]. The second research category focuses on utilizing commercially available *template guided log parser to analyze semi-structured software logs* (generated by the keyword *logs.info()*). The third category of research involves *manual labeling or tagging* maintenance records. This stream of research requires human effort to manually label or tag documents to provide supervision for training and thus is not suitable in context of developing automated unsupervised method for extracting equipment taxonomy-branch from maintenance records. Researchers like [142], [140], [141], [138], [90], and [25] have shown applications of these methods in maintenance domain.

The fourth research category includes *ontology based information extraction*. Ontologies are a promising source for enhancing product life cycle management [74] by resolving the issues of interoperability [156], [85] and [69]. Specifically, in the context of maintenance records [127], [125] and [126] propose to develop and use ontologies for improving reliability models, but their work focuses on extracting roootcause and does not address the issues discussed in section 3.2. Further, [134] extend the equipment ontology by using descriptions of components from equipment taxonomy in maintenance domain, however, their method

uses labeled data which is a critical limitation in adapting their research. The fifth research category focuses on *automatic labeling of topic models* like Latent Dirichlet Allocation [101]. This research stream is further sub divides into two streams, 1) research where similarity between topic labels and topical words documents is measured using similarity measures like Kullback–Leibler divergence [101]; and 2) research where similarity between topic labels and topical words/documents is measured using graph centrality [6], [67]. As graph centrality based models assume topic labels to be represented by a single dominant authoritative graph node, these methods are not suitable for our application because in our application the entire taxonomy-branch is equally authoritative.Methods proposed in [162] make use of a similarity measure called 'C-value' first proposed in [51]. The target of the model proposed in [162] is to extract candidates for labeling topics from the corpus itself. However, as highlighted in challenge C4, industrial maintenance records may not contain complete information about failed taxonomy-branch. Thus, directly using the model proposed by [162] is not feasible for our application and requires adaptations. While methods using similarity scores in topic labeling may provide some help after proper adaptations, these methods do not address the important challenge (C5) of misleading taxonomy terms. We adapt the methods proposed by [101] and [162] for our setting and use these adapted methods as benchmarks for performance comparison with our proposed method in section 3.5.

For practical deployment of the proposed automated method, associating a confidence score for the extracted taxonomy-branch of each maintenance record is essential. Extant methods in the literature make use of labeled data to estimate such confidence values in a supervised setting ([14], [153]). However, in our work, as we do not have any labeled maintenance records, we propose to generate a non-parametric density curve using scores which are derived in a completely unsupervised manner as explained in section 3.4.5. Thus, the confidence score value generated by our model does not depend on labeled data. As described in section 3.5, the only purpose of validation data is to provide recommendation of cutoff limits for confidence values to technicians, these cut-off limits enables technicians

Table 3.1: Limitations of extant methods to tackle the challenges (C1-C6 from section 3.2) in this work

Challenges							
Existing Methods		**C1**	**C2**	**C3**	**C4**	**C5**	**C6**
Keyword Named Entity Recognition for text summary	[28]	✓	✓	✓			
	[98]		✓				
	[164]	✓	✓				
	[116]	✓	✓	✓			
Template Guided Log Parser	[89]	✓					
	[104]	✓		✓			
	[180]	✓					
Manual Labeling or Tagging	[142]		✓	✓			
	[140]	✓	✓				
	[141]		✓	✓			
	[138]		✓	✓			
	[139]		✓	✓			
	[90]		✓	✓	✓		
	[25]		✓				
Ontology based Information Extraction	[127]	✓	✓	✓			
	[125]	✓	✓	✓			
	[126]	✓	✓	✓			
	[134]		✓	✓	✓		
Automatic Labeling of Topic Models	[101]	✓	✓		✓		✓
	[6]	✓	✓	✓			
	[67]	✓	✓	✓			
	[162]	✓	✓		✓		✓

to identify the maintenance records for which they can trust the predicted taxonomy-branch versus the maintenance records that would require manual verification.

Following the limitations of extant methodologies for solving our research challenges, in this paper, we propose a comprehensive unsupervised method (thereby not requiring a labeled dataset) to automatically analyze the unstructured data in maintenance records while tackling the challenges (C1-C6) to determine equipment taxonomy-branch associated with each maintenance record along with a confidence score for each prediction. Next, in section 3.4 we provide a detailed description of our method.

3.4 Proposed Method

Given a set of maintenance records $d \in \mathbf{Doc}$ and the OEM's hierarchical equipment taxonomy (represented by set $\mathbf{T}$) as input data, our method aims to infer the specific taxonomy-branch of failed sub-component ($t \in \mathbf{T}$) corresponding to each maintenance record (d) and provide a confidence score for our predicted result. As shown in Figure 3.3, our method takes as input the raw maintenance record and OEM equipment taxonomy. In the first step, the maintenance record and the taxonomy are processed and initial candidate tokens (along with their initial scores s_p) are extracted using the steps described in section 3.4.1. The initial candidate tokens, which potentially describe the taxonomy-branch associated with the maintenance record, are then fed as input to the proposed Backward-Forward (Bwd-Fwd) and Verb-Analysis algorithm. The Verb-Analysis algorithm also takes as input a set of action-verbs which are identified from the maintenance record as described in section 3.4.3. The output from the Bwd-Fwd algorithm is a subset of taxonomy-branches $\mathbf{BF} \subset \mathbf{T}$ and their corresponding scores $s_{BF}(d,t) \forall t \in \mathbf{BF}$ while the output from the Verb-Analysis algorithm is a subset of taxonomy-branches $\mathbf{VA} \subset \mathbf{T}$ and their corresponding scores $s_{VA}(d,t) \forall t \in \mathbf{VA}$. The scores for each taxonomy-branches generated by both the algorithms respectively are semantically adjusted by using semantic similarity measure ($sim(d,t)$) be-

tween taxonomy-branch $t \in \mathbf{BF}$ &$t \in \mathbf{VA}$ and the maintenance record (d) generated using word embeddings as described in section 3.4.4. The adjusted scores aid in ranking the taxonomy-branches suggested by both the algorithms. The final taxonomy-branch is selected by using *If-Else* scenario that also generates the cumulative score $s_{cum}(d)$ for the final taxonomy-branch which is chosen to be the answer for the given maintenance record ($d \in \mathbf{Doc}$) as shown in section 3.4.5. The final confidence score for each result is generated by fitting a non-parametric kernel density curve $\hat{f}_X(x)$ for the cumulative score.

Before presenting the details of our method, we provide below a brief overview of essential NLP terminology.

- **Word:** the fundamental unit of text ("repair").
- **Token:** single word or a group of words concatenated together to indicate a single entity. Token is uni-gram ('filter'), bi-gram ('drive-motor') or tri-gram ('motor-cooling-system')
- **Vocabulary:** set of all words referred as set **V**
- **Document:** A document (d) is a collection of words (e.g., maintenance record). We denote the set of all documents by $\mathbf{Doc} \ni d$
- **Corpus:** collection of multiple documents

3.4.1 Preprocessing and Preliminary Information Extraction

Maintenance records contain synonyms, abbreviations and acronyms that are absent in the OEM's equipment taxonomy and are often written with poor sentence structure and grammar as can be seen in Figure 3.4a. We will demonstrate the application of our method on this sample maintenance record. It is difficult to apply pattern recognition algorithms for analyzing unstructured maintenance records without any refinement. Pattern recognition methods make use of regular expressions which are sets of characters that help retrieve the

sub-string (sub-components) from the string (maintenance record). To address these data quality issues, the unstructured data in the maintenance records undergo a series of processing steps shown in Figure 3.4b. After processing the records, we get a clean version of maintenance records which only contain lemmatized English words and do not have punctuation, numerals or acronyms. (Note that we do not convert words to their lemmatized form while generating word embeddings based on Parts of Speech (POS) in Section 3.4.5).

After processing the text, the Term Frequency-Inverse Document Frequency (TFIDF) score ([129]) is generated for each individual word (w) in the document which provides the relevance of a word (w) in a given maintenance record. Figure 3.4c shows the TFIDF scores for the words in the document. Next, by concatenating the uni-gram tokens in a maintenance record, n-gram tokens w_D are generated. This creates an exhaustive set '$\mathbf{D} \ni w_D$ ' (Figure 3.4d) of tokens for the maintenance record.

The words in the taxonomy are also processed (Figure 3.5a) to get their lemmatized form. Words representing an individual sub-component are concatenated depending on whether the sub-component is a bi-gram or a tri-gram token. We denote the set of all 'Sub-Unit' tokens in the taxonomy as '$\mathbf{SU_T}$' (e.g., 'Drive-Motor-Assembly', in Figure 3.1a). The set of all 'Maintainable Items' tokens is denoted by '$\mathbf{MI_T}$' (e.g., 'Drive-Mechanism' in Figure 3.1a). The set constituted by tokens which are 'Parts' in the taxonomy is represented as '$\mathbf{PT_T}$' (e.g., 'Belt', in Figure 3.1a). We represent the taxonomy tokens present in the set $\mathbf{SU_T} \cup \mathbf{MI_T} \cup \mathbf{PT_T}$ by w_T.

Candidate sets for taxonomy-branch are generated by measuring the Levenshtein distance ([83]) between the tokens w_T present in sets '$\mathbf{SU_T} \cup \mathbf{MI_T} \cup \mathbf{PT_T}$' and tokens '$w_d \in \mathbf{D}$' respectively for each document $d \in \mathbf{Doc}$. As Levenshtein distance between two terms w_T, w_D (indicated by $Lev(w_T, w_D)$) measures the number of edits required to transform $w_D \rightarrow w_t$ we use the same to incorporate spelling errors made by technicians in maintenance record. All the tokens $w_T \in \mathbf{SU_T}$ that have a Levenshtein distance ≤ 2 with tokens $w_d \in \mathbf{D}$ makes up the candidate set '$\mathbf{SU_C}$'. Similarly, tokens in $w_T \in \mathbf{MI_T}, \mathbf{PT_T}$ with Levenshtein

distance ≤ 2 from tokens $w_D \in \mathbf{D}$ make the candidate sets '$\mathbf{MI_C}$', '$\mathbf{PT_C}$'. Thus, the tokens w_T that have Levenshtein distance ≤ 2 with tokens w_D are present in the candidate sets '$\mathbf{SU_C} \cup \mathbf{MI_C} \cup \mathbf{PT_C}$' and we represent them as w_C. (Please note that tokens w_C are also elements from set '$\mathbf{SU_T} \cup \mathbf{MI_T} \cup \mathbf{PT_T}$'.) We measure the phonetic similarity between the tokens $(w_D,\ w_C\ \forall w_D,\ w_C \mid Lev(w_D, w_C) \leq 2)$ using editex algorithm proposed by [193]. The reason we measure the phonetic similarity is that terms like 'shoe' and 'hose' have the same set of alphabets (letters) and hence a set-based distance measure like Jaccard fails to identify any difference between them, but phonetically they are very distinct as suggested by editex. After forming the candidate sets '$\mathbf{SU_C}$', '$\mathbf{MI_C}$', '$\mathbf{PT_C}$' the TFIDF scores for n-gram tokens in w_C is assigned by averaging the TFIDF scores of individual words in n-gram token $w_D\ \forall w_D,\ w_C \mid Lev(w_D, w_C) \leq 2$ as shown in Eq. 3.1. An initial score, s_p, for each token in the candidate set is generated by multiplying the TFIDF score of the token by a normalizing constant (measuring the phonetic dissimilarity using editex algorithm) as given in Eq. (3.2). The steps to generate the candidate sets and the initial scores are outlined in Figure 3.5b.

Figure 3.5c represents the demonstration of generated candidate sets for the illustrative example in Figure 3.4a. The tokens 'pump' and 'manifold' are identified as candidates of 'Maintainable Item', while the tokens 'piston', 'pressure' and 'hose' are identified as candidates for 'Parts'. Note that in Figure 3.4a, the token 'hose' is not present but is generated due to a mismatch with token 'hole'. However, the initial score s_p for candidate 'hose' is very small as per the calculation in Eq. 3.2. The candidate for Sub-Unit is 'fluid end'. Thus, using Named Entity Recognition techniques, there are multiple taxonomy-branches that can emerge as the most probable equipment taxonomy-branch for this maintenance event.

$$TFIDF(w_C) = \begin{cases} TFIDF(w_D), & \text{if } w_D\ is\ uni-gram \\ \frac{\sum_n TFIDF(w_n)}{n} \forall \{w_n\} \in w_D & \text{if } w_D\ is\ n-gram \end{cases} \tag{3.1}$$

$$s_p(term) = \{\frac{1}{1 + editex(w_D, w_C)}\}^2 \times TFIDF(w_C)$$

$$\forall\ w_C \in \{SU_C, MI_C, PT_C\}\ and\ w_D \in \mathbf{D} \quad (3.2)$$

3.4.2 Backward-Forward (Bwd-Fwd) Algorithm

Our proposed Backward-Forward (Bwd-Fwd) algorithm leverages the OEM's equipment taxonomy hierarchy to infer the specific taxonomy-branch (i.e., 'Equipment-Unit' $\rightarrow$ 'Sub-Unit' $\rightarrow$ 'Maintainable-Item' $\rightarrow$ 'Parts') associated with a given maintenance record $d \in Doc$. Let $l_i,\ i \in \{1, 2, 3, 4\}$ denote the levels in the OEM equipment hierarchy, ranging from 1=Equipment to 4=Part. Let $\mathbf{T}$ denote the set of all the branches in the OEM equipment taxonomy. For an element $e \in \mathbf{SU_C} \cup \mathbf{MI_C} \cup \mathbf{PT_C}$, at a level l_i, the algorithm starts with a backward tracking journey to select (filter) all the taxonomy-branches ($\mathbf{B_i} \subset \mathbf{T}$) which contain the element (e) at level l_i. Then, progressing one step in the backward tracking journey, the algorithm selects those taxonomy-branches ($\mathbf{B_{i-1}} \subset \mathbf{B_i}$) which contain an element ($e^{'} \in \mathbf{SU_C} \cup \mathbf{MI_C} \cup \mathbf{PT_C}$) at l_{i-1} level. This process is continued until the algorithm reaches l_1 level of the taxonomy-branch. For each selected taxonomy-branch, a score is computed by the summation of the initial score s_p for all the elements in the branch that are present in the candidate sets. This score is multiplied by a scaling factor which increases in value with each additional level of the hierarchy through which the candidate branch is retained. Thus, the more levels that a candidate taxonomy-branch is retained, the higher is its score due to the increasing scaling factor. Algorithm **??** provides the pseudo code of the backward progression of the Bwd-Fwd algorithm. The forward step tracking progression of the algorithm is performed similarly, but with the difference that instead of moving from level $l_i \rightarrow l_{i-1}$ the algorithm moves from level $l_i \rightarrow l_{i+1}$. The selected taxonomy-branches left after the i^{th} forward step iterations are represented by a set ($\mathbf{F_i} \subset \mathbf{T}$). The

final candidates for most probable taxonomy-branches for the given maintenance record d are determined by intersecting the branches selected during the backward and forward journeys, and their scores, $s_{BF}(d,t)$ are calculated by taking the maximum of the scores from those branches. We conceptually illustrate the Bwd-Fwd algorithm in Figure 3.6a while the Bwd-Fwd algorithm's output for the example in Figure 3.4a is shown in Figure 3.6b. The Bwd-Fwd algorithm tackles challenge (C4) of missing information in text by leveraging equipment taxonomy. However, the algorithm suffers by the presence of misleading taxonomy tokens in text. As can be seen Figure 3.6b the Bwd-Fwd algorithm predicts the taxonomy-branches with 'fluid-end' (Sub-Unit) $\rightarrow$ 'manifold' (Maintainable Item) as the taxonomy-branch of the failed sub-component. However, this might be incorrect as failure is observed in pistons of fluid-end which appears to be the fourth ranked branch by Bwd-Fwd algorithm. Our novel Verb-Analysis algorithm tackles the issues created by noisy tokens in maintenance records.

3.4.3 Verb Analysis Algorithm

Maintenance records from industrial equipment typically comprise descriptions of and observations from various actions (including inspection, diagnosis, testing, repair and replacement) performed by technicians. Hence, a typical maintenance record (Figure 3.7a) will contain word tokens that are action-verbs (e.g., observed, change, fixed, etc.), along with other word tokens in the vicinity representing the sub-components on which the actions were performed. We make use of this observation and propose a novel Verb-Analysis algorithm that takes advantage of action-verbs to identify the sub-component that lie in the vicinity of the action-verb. Using this approach can help to overcome the challenge of noisy taxonomy tokens (C5) that causes error in inferring the equipment taxonomy-branch of the failed sub-component. The Verb-Analysis algorithm takes as input the tokens from candidate sets '$\mathbf{SU_C}$', '$\mathbf{MI_C}$', '$\mathbf{PT_C}$' along with a holistic set of Action-Verbs $\mathbf{VB}$. We generate the complete set of Action-Verbs $\mathbf{VB}$ by identifying most frequent words that are tagged as 'verb' using Part of Speech (POS) tagger provided in 'nltk' package in python [151] which

Algorithm 2: Backward Part of Backward Forward Algorithm

Input: Sets $\mathbf{T}$, $\mathbf{SU_C}$, $\mathbf{MI_C}$, $\mathbf{P_C}$

/* elements in $\mathbf{T}$ are represented by t_{l_i} where l_i indicate the level of the element t in taxonomy branch */

/* every element in set $\mathbf{SU_C} \cup \mathbf{MI_C} \cup \mathbf{PT_C}$ have a initial score s_p associated with it */

Output: Set $\mathbf{B}$

for $d \in \mathbf{Doc}$ **do**
  for $e \in \mathbf{SU_C} \cup \mathbf{MI_C} \cup \mathbf{PT_C}$ *at* l_i **do**
    $\mathbf{B_i} \subset \mathbf{T} \ni t_{l_i} \leftarrow e$
    $s_b = \exp \times \sum_{i=1}^{4} s_{p_{t_{l_i}}} \forall\ b \in \mathbf{B_i}$
    /* Note here b represents an instance of taxonomy branch in set $\mathbf{B_i}$ */
    $j = 1$;
    while $i \leftarrow 1$ **do**
      for $e^{'} \in \mathbf{SU_C} \cup \mathbf{MI_C} \cup \mathbf{PT_C}$ *at* l_{i-1} **do**
        $\mathbf{B_{i-1}} \subset \mathbf{B_i} \ni t_{l_{i-1}} \leftarrow e^{'}$
        $s_{b'} = \exp^{j} \times s_b \ \forall\ b^{'} \in \mathbf{B_{i-1}}$;
        /* Note here $b^{'}$ represent the same taxonomy branch b after an iteration, as $B_{i-1} \subset B_i$. After iteration, the initial score s_b is modified by $s_{b'}$ */
      $i \leftarrow i - 1$
      $j \leftarrow j + 1$;

/* After obtaing taxonomy branches $\mathbf{B_1}$ from Backward iteration, forward iteration is performed for every element in set $\mathbf{SU_C} \cup \mathbf{MI_C} \cup \mathbf{PT_C}$ to get the filtered set of taxonomy branches $\mathbf{F_N}$ (where N is the maximum number of level of the taxonomy branch) */

/* The final Backward-Forward output is obtained by intersecting the branches present in set $\mathbf{B_1}$ and $\mathbf{F_N}$ to get the final set of taxonomy branches $\mathbf{BF}$ having the score $s_{BF}(d,t)\ \forall\ t \in \mathbf{T}$ */

uses the Penn Treebank tagset [135] from the corpus of maintenance records. From the set of verbs generated we then filter out the verbs that are associated to repair action manually. To ease reproducibility we provide the action-verb set **VB** in supplementary material. The Verb-Analysis algorithm starts by analyzing the maintenance record to identify action-verbs using regular expression provided in Figure 3.7a. The regular expression extracts a maximum of two words (indicated by w) to the left and right of the 'action-verb' along with white spaces (indicated by s) between them. The expression '((w*)s+)0,2' extracts a maximum of

two words in the neighborhood of a given 'action-verb'. This is done using a Part of Speech (POS) tagger. Regular expression is generated to extract phrases (denoted by set $\mathbf{P_{Reg}}$) that include a set of words in the neighborhood of the action-verb. A set, $\mathbf{D_{VA}}$, comprising unigram, bi-gram and tri-gram are generated from the phrases in set $\mathbf{P_{Reg}}$. Tokens belonging to the set $\mathbf{D_{VA}}$ and representing the sub-components ($\mathbf{SU_C} \cup \mathbf{MI_C} \cup \mathbf{PT_C}$) are extracted and are provided a score s_v. The pseudo code for the Verb-Analysis algorithm is outlined in Algorithm 3. The final Verb-Analysis score ($s_{VA}(d,t)$), for a taxonomy-branch $t \in \mathbf{T}$, measures the relevance of the taxonomy-branch to the given maintenance record $d \in \mathbf{Doc}$, and is computed by summing up $s_v(t_i)$, for every element i in the taxonomy-branch. The taxonomy-branches that have non-zero Verb-Analysis scores are compiled to create the set **VA**. Figure 3.7b demonstrates the output obtained by applying the Verb-Analysis algorithm over the example shown in Figure 3.4a.

Algorithm 3: Verb Analysis Algorithm

Input: Sets $\mathbf{T}$, $\mathbf{VB}$, $\mathbf{SU_C}$, $\mathbf{MI_C}$, $\mathbf{P_C}$
Output: Set $\mathbf{VA}$
for $d \in \mathbf{Doc}$ **do**
  $\mathbf{P_{Reg}} \leftarrow$ Phrases from maintenance records using RegEx $\forall\ v \in \mathbf{VB}$
  $\mathbf{D_{VA}} \leftarrow$ uni-gram, bi-gram and tri-gram generated from $\mathbf{P_{Reg}}$
  $s_v(e) = Frequency(e) \times s_p(e)\ \forall\ e \in \mathbf{SU_C} \cup \mathbf{MI_C} \cup \mathbf{PT_C} \cap \mathbf{D_{VA}}$
  `/* here the frequency is calculated as per the set` $\mathbf{D_{VA}}$ `*/`
  $s_{VA}(d,t) = \sum_{\forall e \in t} s_v(e)\ \forall\ t \in \mathbf{T}$

For the example of Figure 3.4a, as can be seen in Figure 3.6b, the taxonomy-branches extracted by the Bwd-Fwd algorithm contain 'manifold' as the Maintainable Item. However, the Verb-Analysis algorithm picks the taxonomy-branch with piston as the top ranked branch because it lies in the vicinity of the action-verbs like 'change', 'failure' as shown in Figure 3.7b. Thus, the ambiguity introduced by noisy taxonomy tokens is mitigated by the Verb-Analysis algorithm.

3.4.4 Semantic Adjustment and Extraction of Top ranked taxonomy-branch for both algorithms

The Backward-Forward algorithm and the Verb-Analysis algorithm described above are two different algorithms to generate scores which measure the relevance an equipment taxonomy-branch to a maintenance record (document). The higher the score for a taxonomy-branch, the higher the likelihood of the taxonomy-branch to be related to the maintenance record. We aim to leverage the scores of both these algorithms to compute a single confidence score.

A common characteristic of both these algorithms is that they utilize frequentist methods in NLP and do not incorporate the contextual knowledge present in the maintenance records. To incorporate the semantic information present in the text, words are represented using embedding vectors. In this paper, two different models for generating word embeddings are studied to determine the effects caused by different methods for incorporating contextual knowledge. The first model used to generate word embeddings is the skip-gram word2vec model proposed by [107]. In this model, word representations are developed by taking into consideration the contextual words present in the neighborhood of a given word. The second word embedding model used to generate the word representation is the model proposed by [173] which generates word embeddings and their POS embeddings by not only considering the neighboring words, but also the POS tags.

After generating word embeddings from the corpus, the distance between a given maintenance record and an equipment taxonomy-branch is measured. To do so, the sentence embedding for the maintenance record ($d \in Doc$) is first created using word embeddings. The sentence embedding is generated by taking the weighted average of the word embeddings present in the document, and then modifying it using PCA/SVD as in [12]. The sentence embedding vector for a document is represented by $\overrightarrow{\mathbf{d_e}}$. Then, embedding vectors for each taxonomy-branch ($\overrightarrow{\mathbf{t_e}}$), is created by averaging the word embeddings of individual words present in the taxonomy-branch. While using the word embeddings, the average

of each word embedding and its corresponding POS tag embedding is taken. For the tokens in the taxonomy, the POS tag is assumed to be *noun*. The cosine similarity between the sentence embedding vector $\overrightarrow{\mathbf{d_e}}$ and the taxonomy embedding vector $\overrightarrow{\mathbf{t_e}}$ measures the similarity ($sim(d,t)$) between the document $d \in \mathbf{Doc}$ and the taxonomy-branch $t \in \mathbf{T}$. The Backward-Forward algorithm's score for the taxonomy-branch ($s_{BF}(d,t)$) and the Verb-Analysis algorithm's score ($s_{VA}(d,t)$) is multiplied by this similarity measure ($sim(d,t)$) to generate the adjusted Backward-Forward ($s_{BF_a}(d,t)$) and Verb-Analysis scores ($s_{VA_a}(d,t)$) for each taxonomy-branch (t) for a given document $d \in Doc$.

Figure 3.8 shows the output obtained by adjusting for semantic similarity for the example in Figure 3.4a. We find that the cosine-similarity ($(sim(d,t))$) is highest for the branch with 'piston_and_liner' as Maintainable Item and 'piston' as Parts. We present the results generated by using word2vec word embeddings in Figure 3.8. The taxonomy-branch scores $s_{BF}(d,t),\ s_{VA}(d,t)$ obtained using the Bwd-Fwd and Verb-Analysis algorithms are multiplied by this semantic similarity $sim(d,t)$ to get the adjusted branch scores $s_{BF_a}(d,t),\ s_{VA_a}(d,t)$ as shown in Figure 3.8.

Using the adjusted scores from both algorithms for each taxonomy-branch ($t \in \mathbf{T}$), each taxonomy-branch is ranked in a descending order of the score, i.e., the order of potential association with the maintenance record. Intuitively, the higher the difference in value between the score of the top ranked branch and the score of the next ranked one, the greater is the confidence associated with the top ranked branch. Thus, the difference in scores between the taxonomy-branch at rank 1 and rank 2 is a significant factor in determining the final confidence score. Also, as we have two different algorithms that measure the relevance of the taxonomy-branch to the maintenance record, the confidence in the inferred taxonomy-branch is higher when the outputs from both the algorithms converge. To take this into consideration, the difference between the adjusted branch scores for rank 1 and rank 2 taxonomy-branch is calculated for both the algorithms ($s^{BF_a}_{diff}(d),\ s^{VA_a}_{diff}(d)$) for each document $d \in \mathbf{Doc}$.

3.4.5 Generation of Confidence Score

To merge the output of the two algorithms, an if-else scenario is considered, where the intuition is that if both algorithms provide the same taxonomy-branch at the first rank, then the algorithms are said to converge and the prediction for the corresponding maintenance record is given a high confidence score. The scores ($s^{BF_a}_{diff}(d)$, $s^{VA_a}_{diff}(d)$) are mathematically adjusted as per the if-else scenario to generate the final cumulative score s_{cum}. However, comparing the scores ($s^{BF_a}_{diff}(d)$ and $s^{VA_a}_{diff}(d)$) directly would be inappropriate without accounting for the variance in them, as these scores are generated from two different algorithms with different variances. Thus, standardization of these scores is necessary and is done by dividing them with their corresponding standard deviations. The Backward-Forward algorithm's standard deviation $\sigma_{b_{BF}}$ is calculated by bootstrapping the scores $s^{BF_a}_{diff}(d)$ for different documents $d \in \mathbf{Doc}$. The same is done to measure the Verb-Analysis algorithm's standard deviation $\sigma_{b_{VA}}$. The standardized algorithm scores $\dot{s^{BF_a}_{diff}}(d)$ and $\dot{s^{VA_a}_{diff}}(d)$, are adjusted as per the if-else scenario as shown in the Figure 3.9a to generate a final cumulative score, s_{cum}, for each document. We demonstrate these steps in Algorithm 4 which provides the pseudo code of the proposed confidence score generation algorithm.

To demonstrate this, the top two taxonomy-branches based on their adjusted branch scores $s_{BF_a}(d,t)$, $s_{VA_a}(d,t)$ from Figure 3.8, are analyzed further to generate the cumulative score $s_{cum}(d)$ for document $d \in \mathbf{Doc}$. The difference between the adjusted branch scores of the top two branches ($s^{BF_a}_{diff}(d)$, $s^{VA_a}_{diff}(d)$), is calculated for both algorithms as shown in Table 3.2. After generating the difference between the top two taxonomy-branches for all maintenance records, the mean ($\overline{s^{BF_a}_{diff}}$, $\overline{s^{VA_a}_{diff}}$) and thus the bootstrapped standard deviation ($\sigma_{b_{BF}}$, $\sigma_{b_{VA}}$) for $s^{BF_a}_{diff}(d)$, $s^{VA_a}_{diff}(d)$ are estimated. Then using this bootstrapped standard deviation, the standardized algorithm scores $\dot{s^{BF_a}_{diff}}(d), \dot{s^{VA_a}_{diff}}(d)$ are generated by dividing $s^{BF_a}_{diff}, s^{VA_a}_{diff}$ with their respective bootstrapped standard deviations ($\sigma_{b_{BF}}$, $\sigma_{b_{VA}}$) as

shown in Table 3.2. As the top branch suggested by both the algorithms are diverging, we follow the if-else scenario (Figure 3.9a) to compare the standardized algorithm scores of both the algorithms. We find out that $s^{\dot{V}A_a}_{diff}(d) > s^{\dot{B}F_a}_{diff}(d)$ and, thus, the top branch suggested by the Verb-Analysis algorithm is the correct answer. The final cumulative score is generated as per the if-else scenario by subtracting the standardized algorithm scores as $s_{cum}(d) = s^{\dot{V}A_a}_{diff}(d) - s^{\dot{B}F_a}_{diff}(d)$. For this example, the top branch suggested by the Verb-Analysis algorithm agrees with the answer provided by the industry expert. Also, we can observe from Figure 3.8 that the semantic similarity of the maintenance record is highest for this taxonomy-branch. Thus, although the Bwd-Fwd algorithm fails to identify the correct taxonomy-branch due to the incorrectly matched token 'manifold', our method successfully determines the correct equipment taxonomy-branch associated with this maintenance record.

After generating the final cumulative score $s_{cum}(d)$ and the final taxonomy-branch for each maintenance record $d \in Doc$, a non-parametric kernel density (Figure 3.9b) is estimated using Eq. (3.3) for the independent and identically distributed scores $s_{cum}(d)$ to generate the confidence score for the predicted taxonomy-branch. There are various choices available for kernel density estimation [31]. We use the Gaussian kernel (Eq. 3.4) and tune the bandwidth hyperparameter (h) using 'kedd' package in R. The bandwidth parameter (h) is essential in smoothing the density plot. Using the optimal kernel density, the cumulative density function (cdf) is generated using cumulative scores for all documents $d \in \mathbf{Doc}$, as shown in Figure 3.9c. Records with confident predictions are the ones that lie on the upper tail of the non-parametric kernel density curve, i.e., the record for which the cdf value is large. We provide an approach, based on a validation dataset (Section 3.5), to decide the cutoff limits for the cumulative density value, above which a record would be treated as having been correctly predicted by our method with high confidence. For the example in Figure 3.4a, where we have shown results with word2vec embedding, the optimized bandwidth parameter $h = 0.713$. Using this parameter and the cumulative score, we generate the cdf to identify the confidence score. As the algorithms do not converge, we get a slightly lower confidence

value of around 68%, corresponding to the cumulative score of $s_{cum} = 0.86$ as shown in Figure 3.9c.

$$\hat{f}_X(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K\frac{(x - x_i)}{h} \tag{3.3}$$

$$K(x) = \frac{\exp(-||x||^2/2)}{\sqrt{2\pi}} \tag{3.4}$$

3.5 Performance Assessment and Discussion

To assess the overall performance of our proposed method we use a total of 2358 maintenance records with a total of 3987 unique vocabulary uni-gram tokens . Large number of maintenance records help us to generate accurate TFIDF values and word embeddings for each word in the vocabulary. In order to benchmark the performance of our proposed method we use a set of 251 manually labeled maintenance records. We benchmark our proposed method with automatic topic labeling methods proposed in [101] and [162] (introduced in section) 3.3. We refer the competing method proposed by [101] and [162] as 'ALMTM' and 'Adapted-TLATR' respectively. For 'ALMTM' model we develop topic model (LDA) for each maintenance record individually and set the number of topics as 1 to identify the single dominant taxonomy-branch that describe each maintenance record. The method proposed by [162] is not directly applicable to our research and thus we adapt it by generating candidate labels using equipment taxonomy to evaluate the 'C-value' between maintenance record and candidate label. For both 'ALMTM' and 'Adapted-TLATR' set **D** is the input to match the n-gram taxonomy tokens. To compare the competing models with our proposed method we evaluate the percentage of manually labeled maintenance records for which the correct taxonomy-branch is identified. For our proposed method, we compare the performance in the context of two types of embeddings discussed earlier, namely word2vec embedding and POS

embedding. We show the comparison in Figure 3.10a where we can see that our proposed method with word2vec embedding outperforms other methods.

Tracing back the obtained results to the challenges (C1-C6) mentioned in section 3.2, we find that all the methods discussed in Figure 3.10a addresses the challenges of *unavailability of labels (C1) and unstructured text (C2).* Also, after adapting the method proposed by [162], all the methods make use of equipment taxonomy to tackle the challenge of incomplete information in maintenance record. Further the challenge posed by presence of non-unique taxonomy tokens (C6) is tackled by the proposed Bwd-Fwd algorithm, while the similarity measure between maintenance record and taxonomy-branches tackle the same in competing methods. However, the reason for superior performance of our proposed method is that Verb-Analysis algorithm handles the challenge of misleading taxonomy tokens and word2vec handles the issue issue created by multiple contexts of words with higher effectiveness.

Next, we demonstrate how the generated confidence score for predicted taxonomy-branch of each maintenance record can help technicians decide whether or not to trust the prediction of proposed method. We wish to highlight that our proposed method until this point (including the generation of confidence score) does not rely on labeled data in any way and is unsupervised. But, to help technicians effectively trust the confidence score values, we determine the cutoff limits for three regimes of the confidence score using labeled/Gold standard data as described below.

First, predictions are generated for the labeled dataset using the proposed workflow. The top branch suggested by the workflow is compared with the Gold standard. The comparison results are then grouped into three groups:

- High Score Regime: The automated method's results that have a confidence score in this regime require no manual review and can be accepted as accurate, i.e., the top ranked taxonomy-branch for a given maintenance record is the correct answer.

- Medium Score Regime: For maintenance records whose results fall in this confidence score regime, the correct answer may or may not correspond to the top ranked taxonomy-

branch but is among the top five predicted taxonomy-branches. The user would only need to review the top five predicted taxonomy-branches and confirm the one that is the correct answer.

- Low Score Regime: Maintenance records in this regime are ones for which the method is likely not successful in predicting the correct answer. These records would need a thorough manual review to determine the taxonomy-branch.

Figure 3.10b summarizes the number of maintenance records that belong to each of three regimes. It can be observed that using the cutoff limits by word2vec (POS) embeddings the technicians can trust predictions from proposed method if the confidence score generated is above 53.75% (51.12%). We also observe that higher number of records are identified to be in High score regime when we use POS embedding (93) as opposed to word2vec embedding (88) for the proposed method. However, the overall accuracy obtained by using word2vec embedding is high, but as most of these correct predictions have low confidence they lie in the Low score regime and would need manual inspection.

From a practical standpoint, as the maintenance records that fall into the Low Score Regime require manual analysis, the lower the percentage of records in this regime, the better is the performance of the automated method. While ideally it is desirable to have all the results to be in the High Score Regime, this is not realistic and our proposed method tries to tackle all the challenges at it best. However, still some maintenance records present really high complexity including high number of misleading taxonomy words and incomplete knowledge of synonymous taxonomy tokens etc. Hence, the confidence scores associated with the results provided by our automated method helps technicians target their effort and thereby minimize the amount of time they need to spend on manually analyzing the maintenance records.

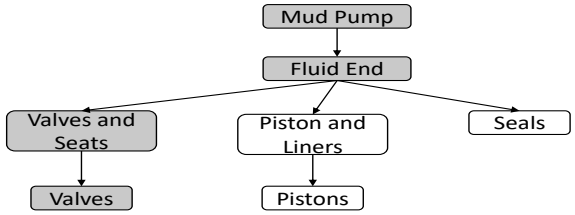
3.6 Summary and Future Work

In this paper, we have proposed a method that uses unsupervised learning to automatically extract the equipment taxonomy-branch pertaining to a maintenance record, and also provide a confidence score for the result. The ability to automatically extract such insights from maintenance records represents a significant time savings, allowing industrial personnel to focus their efforts on analyzing only those results with lower confidence scores. Our proposed method addresses the critical challenges faced in automated analysis of unstructured text data including unavailability of labels, incomplete information in text, multiple contextual usage, presence of misleading and non-unique taxonomy tokens. To tackle these challenges, our method utilizes equipment taxonomy and leverage the use of word embeddings for semantic disambiguation. Our Verb-Analysis algorithm helps overcome the issues posed by misleading taxonomy tokens and our Bwd-Fwd algorithm addresses the challenges due to non-unique taxonomy tokens. Our method also demonstrates a novel way to generate confidence scores for each classification results without making use of labeled data.

In our future research, we will explore how the results from ongoing usage of the proposed automated method can help identify and address gaps in the OEM's current taxonomy as well as enable improvements in the way maintenance records are documented by technicians in the field. A crucial task for future research is to enrich the existing equipment taxonomy (by unidentified synonymous terms) in an automated manner to boost the performance of the proposed method. To achieve this, the maintenance records present in high confidence regime could be used as a source of supervision for training supervised classification models like BERT [167] to tackle the problems like named entity recognition, maintenance record summarization etc. and provide better thesaurus for OEMs to use in future.

Maintenance Record	Failed Taxonomy Branch
The louvres ordered on this job have been ordered to replace the repaired one on mud pump #2. The existing one was repaired but the remaining ones are not in good shape--------- Parts Requested from ICS. ---------- Louvers received and replaced the damaged one in MP 2 & MP 4 Ducts. All others are in good condition. So kept the remaining TWO in workshop for future immediate use.	Mud Pump (Equipment Unit) ↓ Motor Cooling System (Sub-Unit) ↓ Air Ducting (Maintainable Item) ↓ Louver (Part)

(a) Sample of input maintenance records and the associated taxonomy-branch.

Maintenance Record	Taxonomy Branch
Pull **pistons** from mud pump1. Remove componenets from **fluid end** and inspected the same. Coat with light oil and reassemble.---------- **Replaced valves** of Dis cyl#1 Suc cyl#2 and all the cap **seals** Lubricated in Modules.------- Parts Requested from ICS:	

(b) Challenge with uniquely identifying the equipment taxonomy-branch from maintenance record.

Console Software Log

LOG.info("**Received error type**" + error + " **with size in bytes** "+ error.getNumBytes( ) + "**from**" + error.inAddr)

2021-10-22 19:02:20 INFO. error_obj: Received error type err-1213412313213 with size in bytes 620120 from /12.252.91.86

Structured Log

TIMESTAMP	2021-10-22 19:02:20
LEVEL	INFO.
COMPONENT	error_obj
TEMPLATE	**Received error type <...> with size in bytes <...> from /<...>**
VARIABLES	err-1213412313213 ; 620120 ; 12.252.91.86

Industrial Maintenance Records (Logs)

Pump 1 cyl 1 liner 2803hrs Pump 1 cyl 1 piston 552 hrs Replace mud pump 1 cyl 1 piston due fluid passing. Once this was pulled the liner was observed to have some bad scoring. Replaced liner and piston. No other damage to report.

Mud pump # 4 Lube oil filter element replaced magnetic filter removed cleaned linear cooling wash pump stuffing box packing seals removed old and fitted new seals---------Oil Pressure 90/75 Psi Filter change overdue 599 hours

(c) Comparison of software console logs and industrial maintenance records (logs)

Figure 3.2: Illustrations of different types of maintenance records and their associated challenges.

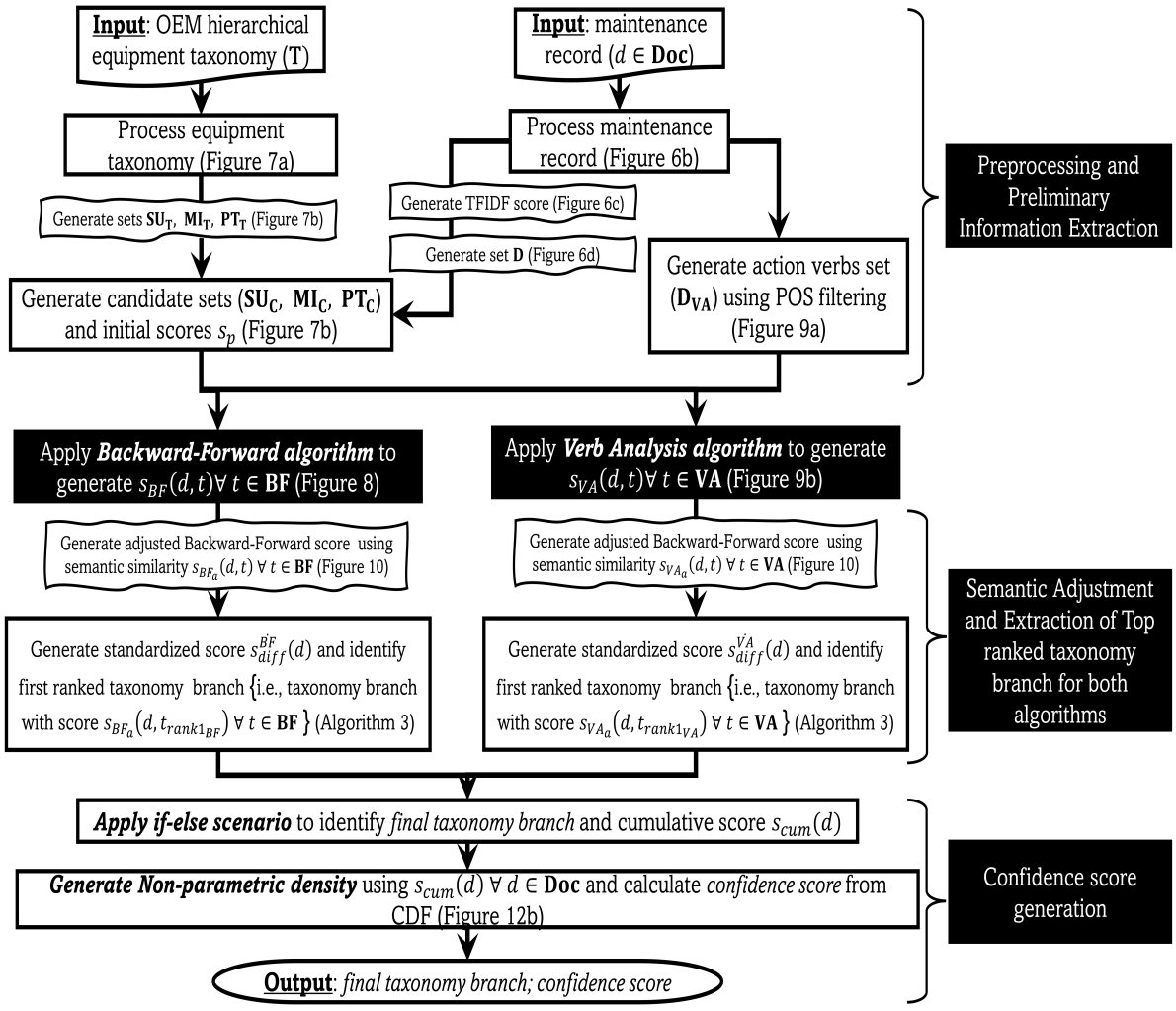
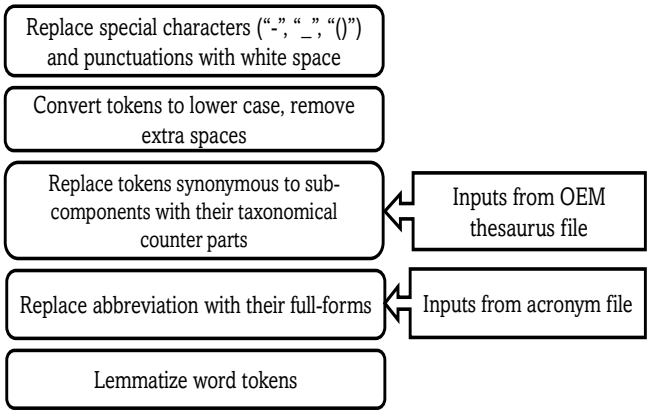


Figure 3.3: Step by step overview of the proposed method.

Root Cause	Event Description	Preventive Action	Immediate Action Description
Drilling with 4200 l/m and 280 bar. Failure of piston rubber. Total hours on piston 294.	While drilling 17 1/2 hole on Z4, driller observed pressure drop on stand pipe manifold. Derrickman reported a wash on #2 fluid end piston.	Regular checks on mud pumps while drilling.	Shut down pump, isolate pump. TOFS. Change piston.

(a) Example of maintenance record $d \in \mathbf{Doc}$.



(b) Processing of maintenance records.

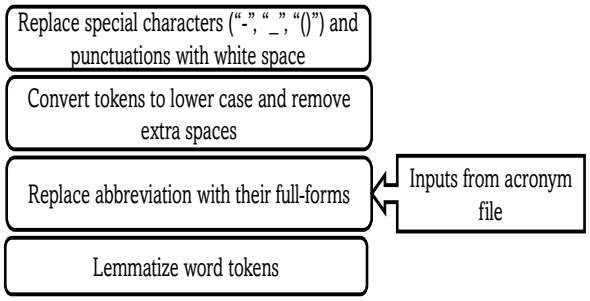
[('with', 0.063), ('while', 0.211), ('we', 0.083), ('wash', 0.099), ('total', 0.156), ('tofs', 0.159), ('to', 0.039), ('time', 0.095), ('the', 0.084), ('stand', 0.147), ('shut', 0.125), ('same', 0.112), ('rubber', 0.142), ('report', 0.109), ('regular', 0.17), ('pump', 0.165), ('pressure', 0.08), ('piston', 0.455), ('pipework', 0.129), ('on', 0.284), ('of', 0.044), ('observe', 0.11), ('mud', 0.11), ('manifold', 0.142), ('isolate', 0.103), ('hour', 0.101), ('hole', 0.233), ('fluid', 0.166), ('failure', 0.065), ('fail', 0.116), ('end', 0.164), ('drop', 0.128), ('driller', 0.136), ('drill', 0.285), ('down', 0.102), ('derrickman', 0.158), ('continue', 0.126), ('circulate', 0.13), ('check', 0.092), ('change', 0.055), ('bar', 0.149), ('at', 0.083), ('and', 0.029)]

(c) Generation of TFIDF score.

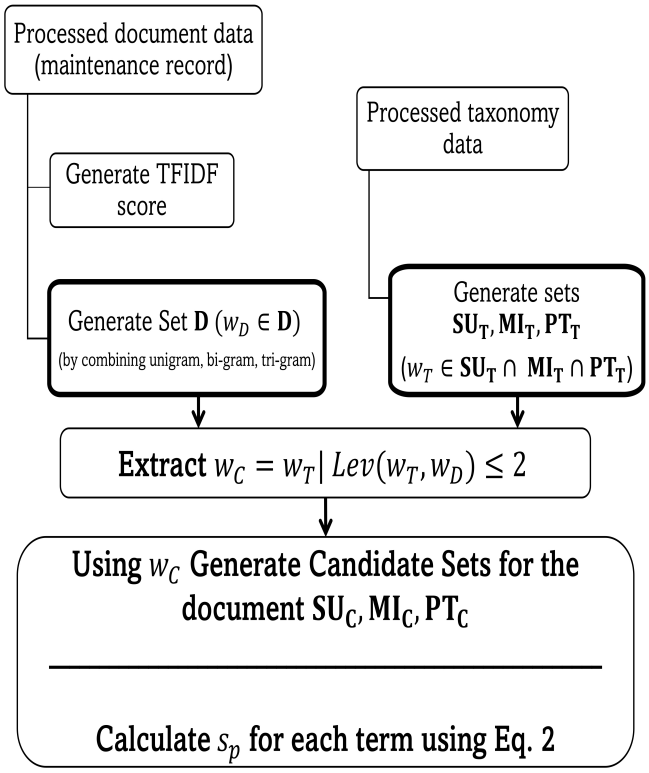
while drill hole on z driller observe pressure drop on stand pipework manifold derrickman report a wash on mud pump fluid end piston at the same time a we continue to circulate on the hole a piston fail on mud pump fluid end piston drill with l m and bar failure of piston rubber total hour on piston regular check on mud pump while drill shut down pump isolate pump tofs change piston while_drill drill_hole hole_on on_z z_driller driller_observe observe_pressure pressure_drop drop_on on_stand stand_pipework pipework_manifold manifold_derrickman derrickman_report report_a a_wash wash_on on_mud mud_pump pump_fluid fluid_end end_piston piston_at at_the the_same same_time time_a a_we we_continue continue_to to_circulate circulate_on on_the the_hole hole_a a_piston piston_fail fail_on on_mud mud_pump pump_fluid fluid_end end_piston piston_drill drill_with with_l l_m m_and and_bar bar_failure failure_of of_piston piston_rubber rubber_total total_hour hour_on on_piston piston_regular regular_check check_on on_mud mud_pump pump_while while_drill drill_shut shut_down down_pump pump_isolate isolate_pump pump_tofs tofs_change change_piston while_drill_hole drill_hole_on hole_on_z on_z_driller z_driller_observe driller_observe_pressure observe_pressure_drop pressure_drop_on drop_on_stand on_stand_pipework stand_pipework_manifold pipework_manifold_derrickman manifold_derrickman_report derrickman_report_a report_a_wash a_wash_on wash_on_mud on_mud_pump mud_pump_fluid pump_fluid_end fluid_end_piston end_piston_at piston_at_the at_the_same the_same_time same_time_a time_a_we a_we_continue we_continue_to continue_to_circulate to_circulate_on circulate_on_the on_the_hole the_hole_a hole_a_piston a_piston_fail piston_fail_on fail_on_mud on_mud_pump mud_pump_fluid pump_fluid_end fluid_end_piston end_piston_drill piston_drill_with drill_with_l with_l_m l_m_and m_and_bar and_bar_failure bar_failure_of failure_of_piston of_piston_rubber piston_rubber_total rubber_total_hour total_hour_on hour_on_piston on_piston_regular piston_regular_check regular_check_on check_on_mud on_mud_pump mud_pump_while pump_while_drill while_drill_shut drill_shut_down shut_down_pump down_pump_isolate pump_isolate_pump isolate_pump_tofs pump_tofs_change tofs_change_piston

(d) Generation of set $\mathbf{D}$.

Figure 3.4: Pre-processing of unstructured maintenance records.



(a) Processing of equipment (Mud-Pump) taxonomy.

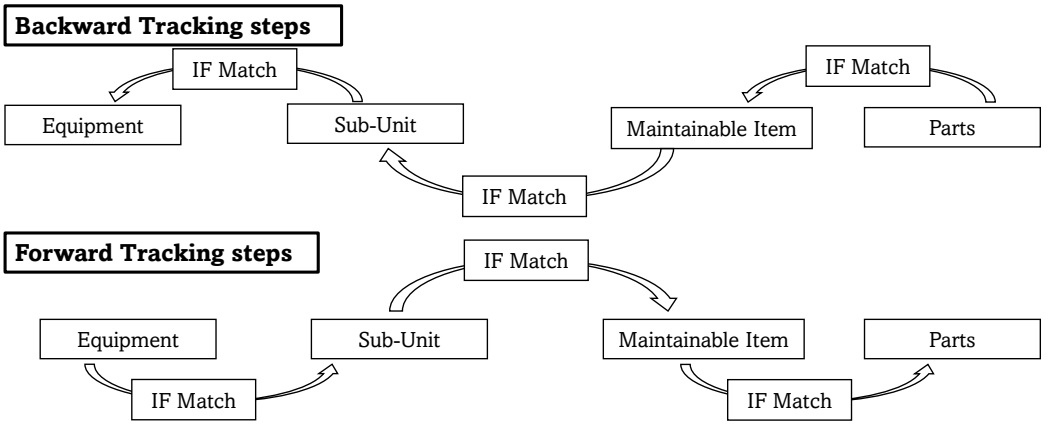


(b) Procedure to generate candidate sets and extract initial scores s_p.

SU_C		MI_C		PT_C	
Candidate	s_p	**Candidate**	s_p	**Candidate**	s_p
fluid_end	0.165	manifold	0.142	hose	0.0086
				piston	0.455
		pump	0.0165	pressure	0.08

(c) Creation of candidate sets $\mathbf{SU_C}$, $\mathbf{MI_C}$, $\mathbf{PT_C}$ for the given example in Figure 3.4a.

Figure 3.5: Initial steps of the proposed method.



(a) Intuition behind Backward-Forward algorithm.

Result after Bwd-Fwd Analysis				
Equipment Unit	**Sub Unit**	**Maintainable Item**	**Parts**	$s_{BF}(d,t)$
mud_pump	**fluid_end**	**manifold**	discharge_manifold	0.835
mud_pump	**fluid_end**	**manifold**	gauge_manifold	0.835
mud_pump	**fluid_end**	**manifold**	suction_manifold	0.835
mud_pump	**fluid_end**	piston_and_liner	**piston**	0.62
mud_pump	**fluid_end**	gauge	**pressure**	0.245
mud_pump	liner_wash	gauge	**pressure**	0.08
mud_pump	liner_wash	sensor	**pressure**	0.08
mud_pump	lubrication_system	gauge	**pressure**	0.08
mud_pump	lubrication_system	sensor	**pressure**	0.08
mud_pump	motor_cooling_system	sensor	**pressure**	0.08
mud_pump	charge_system	**pump**		0.017
mud_pump	liner_wash	**pump**		0.017
mud_pump	lubrication_system	**pump**		0.017
mud_pump	liner_wash	plumbing	hose	0.009
mud_pump	lubrication_system	plumbing	hose	0.009
mud_pump	motor_cooling_system	plumbing	hose	0.009

(b) Output of Bwd-Fwd algorithm for the example.

Figure 3.6: Backward-Forward algorithm.

Regular Expression: r'(((\w*)\s+){0,2})('action-verb')((\s+(\w*)){0,2})'

Drilling with 4200 l/m and 280 bar. Failure of piston rubber. Total hours on piston 294. While drilling 17 1/2 hole on Z4, driller observed pressure drop on stand pipe manifold. Derrickman reported a wash on #2 fluid end piston. Regular checks on mud pumps while drilling. Shut down pump, isolate pump. TOFS. Change piston.

(a) Example of maintenance record with action-verbs and candidate tokens along with regular expression for extracting phrases.

Tokens extracted from Action Verbs

$$\{SU_C \cup MI_C \cup PT_C\} \cap D_{VA}$$

$$\{\text{('piston', 0.91), ('pump', 0.066), ('pressure', 0.08)}\}$$

Result after Verb Analysis				
Equipment Unit	**Sub Unit**	**Maintainable Item**	**Parts**	$s_{VA}(d,t)$
mud_pump	fluid_end	piston_and_liner	**piston**	0.91
mud_pump	fluid_end	gauge	**pressure**	0.08
mud_pump	liner_wash	gauge	**pressure**	0.08
mud_pump	liner_wash	sensor	**pressure**	0.08
mud_pump	lubrication_system	gauge	**pressure**	0.08
mud_pump	lubrication_system	sensor	**pressure**	0.08
mud_pump	motor_cooling_system	sensor	**pressure**	0.08
mud_pump	charge_system	**pump**		0.066
mud_pump	liner_wash	**pump**		0.066
mud_pump	lubrication_system	**pump**		0.066

(b) Output of Verb-Analysis algorithm for the example.

Figure 3.7: Verb-Analysis algorithm.

$$s_{BF_a} = s_{BF} \times sim(d,t)$$

Result after Bwd-Fwd					
Equipment Unit	**Sub-Unit**	**Maintainable Item**	**Parts**	$s_{BF_a}(t)$	$sim(d,t)$
mud_pump	fluid_end	manifold	gauge_manifold	0.624	0.747
mud_pump	fluid_end	manifold	discharge_manifold	0.621	0.743
mud_pump	fluid_end	manifold	suction_manifold	0.611	0.732
mud_pump	fluid_end	piston_and_liner	piston	0.498	**0.803**
mud_pump	fluid_end	gauge	pressure	0.183	0.745
mud_pump	motor_cooling_system	sensor	pressure	0.057	0.714
mud_pump	liner_wash	sensor	pressure	0.055	0.692
mud_pump	liner_wash	gauge	pressure	0.054	0.676
mud_pump	lubrication_system	sensor	pressure	0.049	0.613
mud_pump	lubrication_system	gauge	pressure	0.048	0.598
mud_pump	liner_wash	pump		0.013	0.753
mud_pump	charge_system	pump		0.012	0.727
mud_pump	lubrication_system	pump		0.012	0.705
mud_pump	motor_cooling_system	plumbing	hose	0.006	0.706
mud_pump	liner_wash	plumbing	hose	0.005	0.601
mud_pump	lubrication_system	plumbing	hose	0.005	0.595

(a) Bwd-Fwd Algorithm

$$s_{VA_a} = s_{VA} \times sim(d,t)$$

Result after Verb Analysis					
Equipment Unit	**Sub-Unit**	**Maintainable Item**	**Parts**	$s_{VA_a}(t)$	$sim(d,t)$
mud_pump	fluid_end	piston_and_liner	piston	0.730	**0.803**
mud_pump	fluid_end	gauge	pressure	0.060	0.745
mud_pump	motor_cooling_system	sensor	pressure	0.057	0.714
mud_pump	liner_wash	sensor	pressure	0.055	0.692
mud_pump	liner_wash	gauge	pressure	0.054	0.676
mud_pump	liner_wash	pump		0.050	0.753
mud_pump	lubrication_system	sensor	pressure	0.049	0.613
mud_pump	charge_system	pump		0.048	0.727
mud_pump	lubrication_system	gauge	pressure	0.048	0.598
mud_pump	lubrication_system	pump		0.047	0.705

(b) Verb-Analysis Algorithm.

Figure 3.8: Adjusting scores of both algorithms by semantic similarities.

Algorithm 4: Confidence score generation

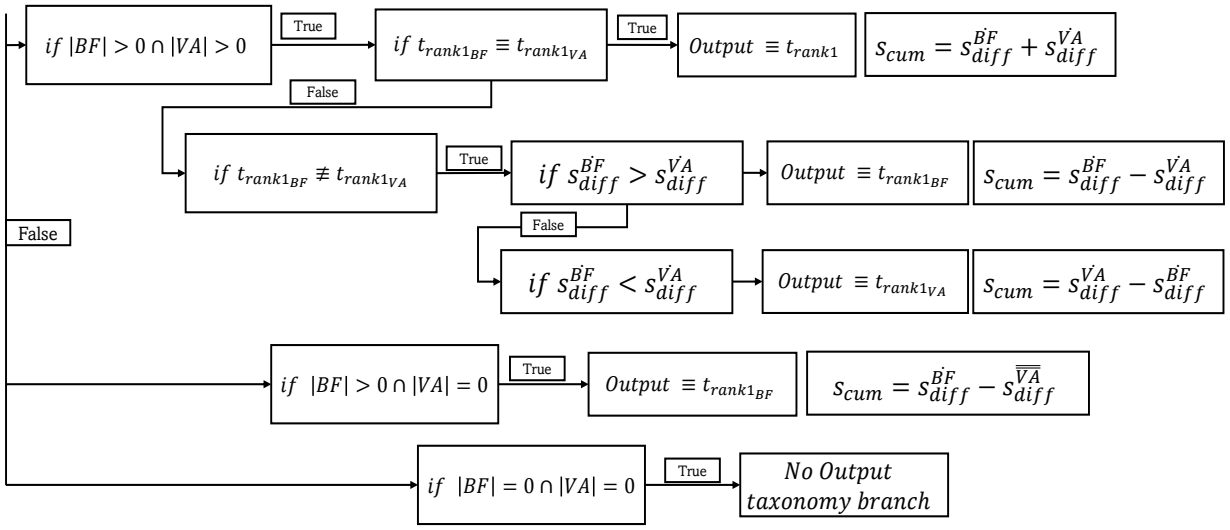
Input: For all documents $d \in$ **Doc** we input
Set **BF** *with* $s_{BF}(t)\ \forall\ t \in$ **BF**, *Set* **VA** *with* $s_{VA}(t)\ \forall\ t \in$ **VA**,
sentence embedding $\overrightarrow{\mathbf{d_e}}$, *taxonomy branch embedding* $\overrightarrow{\mathbf{t_e}}\ \forall\ t \in \mathbf{T}$

Output: Cumulative Score for the document/maintenance record s_{cum}

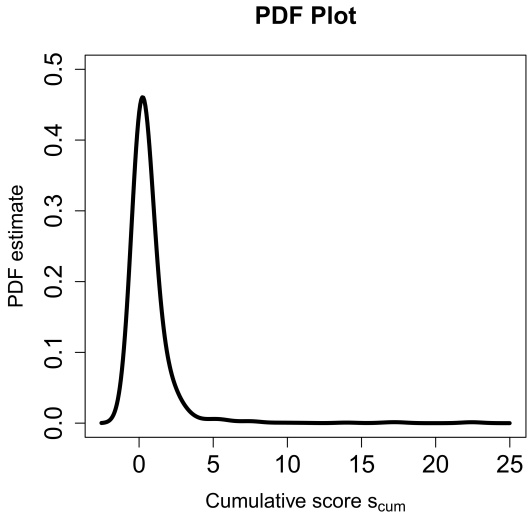
for *document* $d \in$ **Doc do**
  for *taxonomy branch* $t \in$ **BF** *or* **VA do**
    $sep(d,t) \leftarrow 1 - \frac{\overrightarrow{\mathbf{d_e}} \cdot \overrightarrow{\mathbf{t_e}}}{||\overrightarrow{\mathbf{d_e}}||\ ||\overrightarrow{\mathbf{t_e}}||}$
    $s_{BF_a}(d,t) = \frac{s_{BF}(d,t)}{sep(d,t)}$
    $s_{VA_a}(d,t) = \frac{s_{VA}(d,t)}{sep(d,t)}$
  ;
  $s_{diff}^{BF_a}(d) = s_{BF_a}(d, t_{rank_1}) - s_{BF_a}(d, t_{rank_2})$
  $s_{diff}^{VA_a}(d) = s_{VA_a}(d, t_{rank_1}) - s_{VA_a}(d, t_{rank_2})$
;
Bootstrap n documents $d_i \in$ **Doc** and their corresponding $s_{diff}^{BF_a}(d_i)$ *and* $s_{diff}^{VA_a}(d_i)$
`/* Calculate Bootsrapped Means for both algorithms */`
$\overline{\overline{s_{diff}^{BF_a}}} = \frac{\sum_{i=1}^{n} s_{diff}^{BF_a}(d_i)}{n}$
$\overline{\overline{s_{diff}^{VA_a}}} = \frac{\sum_{i=1}^{n} s_{diff}^{VA_a}(d_i)}{n}$
`/* Calculate Bootsrapped Standard Deviation for both algorithms */`
$\sigma_{b_{BF}} = \frac{\sum_{i=1}^{n} (s_{diff}^{BF_a}(d_i) - \overline{\overline{s_{diff}^{BF_a}}})^2}{n-1}$
$\sigma_{b_{VA}} = \frac{\sum_{i=1}^{n} (s_{diff}^{VA_a}(d_i) - \overline{\overline{s_{diff}^{VA_a}}})^2}{n-1}$
`/* Geneate final standardized score for each document` $d \in$ **Doc** `*/`
for *documents* $d \in$ **Doc do**
  $\dot{s_{diff}^{BF_a}}(d) = \frac{s_{diff}^{BF_a}}{\sigma_{b_{BF}}}$
  $\dot{s_{diff}^{VA_a}}(d) = \frac{s_{diff}^{VA_a}}{\sigma_{b_{VA}}}$
  $s_{cum}(d) \leftarrow if - else - scenario\ Fig\ 3.7\ (\dot{s_{diff}^{BF_a}}(d), \dot{s_{diff}^{VA_a}}(d))$
  Final Taxonomy Branch $\leftarrow if - else - scenario\ Fig\ 3.7\ (\dot{s_{diff}^{BF_a}}(d), \dot{s_{diff}^{VA_a}}(d))$
;

Table 3.2: Merging the results of both the algorithms to generate the cumulative score $s_{cum}(d)$ for the maintenance record (document d). The taxonomy-branch in bold denotes failed taxonomy-branch

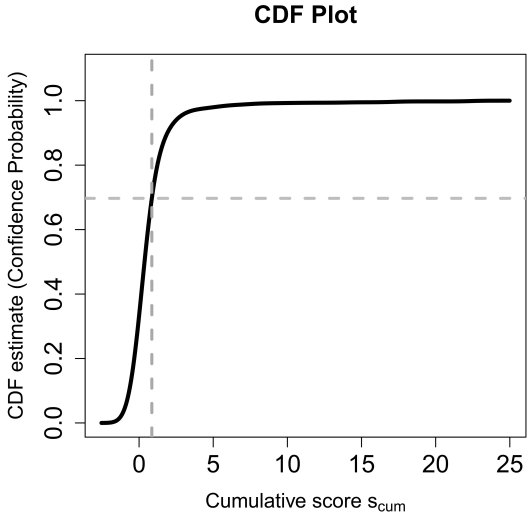
Taxonomy Branch Rank Bwd-Fwd	Taxonomy Branch Bwd-Fwd	$s_{BF_a}(d,t)$	$s_{diff}^{BF_a}(d)$	$\sigma_{b_{BF}}$	$\dot{s_{diff}^{BF_a}}(d)$	$s_{cum}(d)$
1	mud_pump, fluid_end, manifold, gauge_manifold	0.624	0.003	0.367	0.009	**0.86**
2	mud_pump, fluid_end, manifold, discharge_manifold	0.621				
Taxonomy Branch Rank VA	Taxonomy Branch VA	$s_{VA_a}(d,t)$	$s_{diff}^{VA_a}(d)$	$\sigma_{b_{VA}}$	$\dot{s_{diff}^{VA_a}}(d)$	
1	**mud_pump, fluid_end, piston_and_liner, piston**	**0.73**	**0.67**	**0.768**	**0.872**	
2	mud_pump, fluid_end, gauge, pressure	0.06				



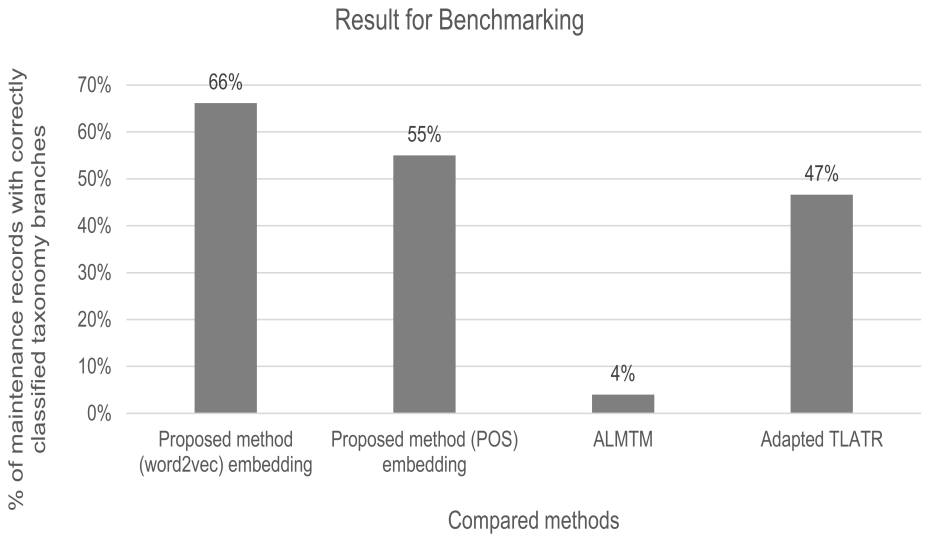
(a) If-else scenario for assessing convergence of two algorithms and generating final cumulative score s_{cum}.



(b) Gaussian kernel density function using cumulative scores of all documents.

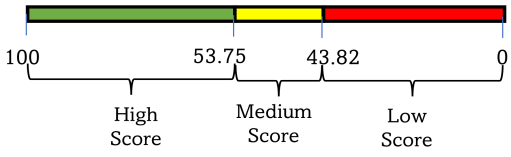


(c) Confidence score generated for the given



(a) Percentage of correctly classified maintenance records using proposed framework (with word2vec and POS embedding) and adapted competitive topic labeling model.

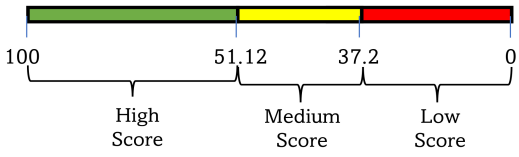
Results using Word2Vec embedding



Predicted # of records		
In High Score Regime	In Medium Score Regime	In Low Score Regime
88	22	141

Actual # of records		
With correct answer at top	With correct answer in top 2-5	With no correct answer
166	47	38

Results using POS embedding



Predicted # of records		
In High Score Regime	In Medium Score Regime	In Low Score Regime
93	40	118

Actual # of records		
With correct answer at top	With correct answer in top 2-5	With no correct answer
138	73	40

(b) Practical application of proposed method and cutoff limits for confidence score to identify maintenance records in three regimes.

Figure 3.10: Final Results over gold standard dataset.

Chapter 4

A Unsupervised Multi-class Ensemble Classifier for Identifying Equipment Failure Mechanisms from Maintenance Records

4.1 Focused Abstract

Unstructured data in equipment maintenance records contain valuable information regarding failures. The ability to classify failure incidents into contributing failure mechanisms can help in improving equipment design and maintenance plans to achieve higher equipment uptime. Supervised learning approaches for automated extraction of failure mechanisms from unstructured data are impractical due to the manual labeling effort. Further, due to the complexities inherent in unstructured data, it can be beneficial to utilize multiple base classifier algorithms for analyzing the maintenance records from different perspectives. In this chapter, we propose a novel unsupervised multi-class ensemble classifier (UMEC) model to extract failure mechanisms from unstructured maintenance records by leveraging contin-

uous scores generated by multiple base classifiers. The model decomposes the multi-class problems into multiple binary classifiers using error-correcting output codes (ECOC). We propose a novel way to encode multi-class problems to binary class by using *maximum* as an order statistic to reduce multi-class scores to binary-classes. We also address the issue of unbalanced datasets in unsupervised classification. We study the influence of different types of noise structure (including feature noise and class mislabeling noise) over the classifiers and demonstrate the effectiveness of our approach using simulation and real world industrial data.

4.2 Introduction

Industrial equipment, such as mud pumps used in the oil and gas exploration industry, are large and complex systems. Equipment failures and associated downtime in production can result in significant financial losses. Identifying the failure mechanism underlying each failure incident can help in improving equipment design and maintenance plans to achieve higher equipment uptime. ISO has defined five primary failure mechanisms listed in (Table 4.1) for oil rig equipment [70]. The unstructured textual data (or notes) that technicians document in maintenance records can be valuable for inferring the failure mechanism. For instance, the maintenance record in Figure 4.1a notes that the equipment failure occurred due to a leaking valve, and hence was a *Mechanical Failure*. Since the effort for manual analysis of maintenance records is prohibitive, there is a need for a machine learning enabled decision system that can automatically extract failure mechanisms from the maintenance records.

Table 4.1: Failure mechanisms from [70]

Failure Mechanism				
Mechanical Failure (Leakage, Vibration, Alignment $\cdots$)	Material Failure (Corrosion, erosion, wear, $\cdots$)	Control Failure (No/Faulty signal, Calibration,$\cdots$)	Electrical Failure (Short Circuit, Faulty power, $\cdots$)	Hydraulic Failure (Blockage, plugged, contamination, $\cdots$)

Extracting information from raw unstructured maintenance records is met with many challenges. First, *unsupervised data:* Manually labeling maintenance records to create a training data is impractical. Second, *single source of domain knowledge could be inadequate:* For example an unsupervised classifier that identify failure mechanism by matching words present in Table 4.1 and maintenance record would fail in cases where such words are not mentioned as shown in Figure 4.1b. Third, *simultaneous presence of misleading tokens:* For example an unsupervised classifier described previously would also suffer if there are high number of contradicting tokens present in maintenance record as shown in Figure 4.1c, where *breakdown*, *wear*, *power imbalance* all represent different failure mechanisms. Such an unsupervised classifier leveraging *words as features* would thus have high feature noise [52] in them. To tackle these challenges we can develop another classifier based on a different source of domain knowledge. For instance, an unsupervised classifier that identify failure mechanism based on *parts that failed* can help to tackle the second challenge of inadequacy in domain knowledge. As instance, *inverter* in Figure 4.1b indicates that the failure mechanism should be electrical. However, such a classifier would face the fourth challenge of *ambiguity in domain knowledge:* For example, *motor* in Figure 4.1c are equally prone to *Electrical* and *Mechanical* failures. This domain ambiguity will cause the classifier to suffer *mislabeling noise (label noise)* [52]. Fifth challenge is that the *context of words change their meaning:* For example, *alarm* and *fuse* in Figure 4.1b have very different meaning based on their usage as *noun* or *verb*.

To tackle the above mentioned challenges multiple base classifiers that use different facets of domain knowledge for analyzing maintenance records can help. However, it is possible that results from these different base classifiers do not converge. Thus, to optimally ensemble results of multiple base classifiers becomes an important challenge. Spectral methods have been used by researchers to ensemble classifiers for unsupervised learning. [117] propose an ensemble method for binary classification using discrete class labels from base algorithms, while [189] extend it to a multi-class ensemble case. [2] present an ensemble learning model

Maintenance Record	Failure Mechanism
Leak on booster valve. - To be investigated when operation allows valve to be dismantled. Trouble getting pressure on MP 3. RMS job made to repair valve. Checked that the valve are on maintenance plan ok. Stop and restart pump. vent supercharge. Found leaking valve on MP'3 to booster valve	**Mechanical Failure**

(a) Maintenance record to demonstrate equipment undergoing Mechanical Failure.

Maintenance Record	Failure Mechanism
Send Inverter to ABB for repair and return this inverter failed from MP #2 B motor. when common alarm given drive was inspected and both line side fuses had blown MP was not in use at that time which means the inverter had failed this was replaced with spare unit and system is operation.. inverter to be sent to ABB singapore for repair and return please see below from ABB and attached. Please advise the AWB once you have shipped the item. This inverter is back on the rig and in stored in the DSGR	**Electrical Failure**

(b) Maintenance Record with no direct words indicating underlying Failure Mechanism.

Maintenance Record	Failure Mechanism
TROUBLESHOOT ABNORMAL TORQUE AND POWER READINGS. *Breakdown* Job, Normal expected wear tear from operating equipment designed MUD PUMP. MEASURED MOTOR CURRENT VALUES AND COMPARED WITH VFD CURRENT READINGS VFD. Iact Motor Phase Measurement Per Phase VFD. **POWER IMBALANCE BETWEEN MOTOR**. Checked belt *alignment*. Turn adjustment screw made alignment motor perfect using wire clamped from sheave other Belt tension checked psi aligned earlier.	**Electrical Failure**

(c) Maintenance Record with misleading words conflicting with true Failure Mechanism.

Figure 4.1: Different types of unstructured maintenance records, their corresponding failure mechanisms and issues.

called SUMMA for binary classification and demonstrate that using continuous class scores, instead of discrete class labels, helps retain the comparative inference of base algorithms. However, *ensemble learning using continuous scores of base algorithms for multi-class classification remains an open research problem.* Methods that decompose the multi-class problem into multiple binary-class problems includes schemes like the One-Versus-All (OVA), the Error-Correcting-Output-Codes (ECOC) explained in section (4.4.2). But, these schemes have been studied only in supervised learning settings [7]. Further, extant multi-class to multiple-binary class decomposition methods use *mean* of the scores in binary groups as decomposition (reduction) statistics[2]. But, the usefulness of other order statistics remains to be studied. Apart from this, issues like unbalanced dataset, feature noise and mislabeling

noise significantly increase the complexity to ensemble results from base classifiers.

Thus, in this chapter, we present a solution framework for multi-class classification problems using unsupervised ensemble learning with continuous scores from multiple base classifiers while also tackling issues like class imbalance and feature/label noise in the dataset. While our approach for multi-class classification is inspired by extant research on binary-class classification, we make the following key contributions:

- Development of an Unsupervised Multi-class Ensemble Classification model (UMEC) which leverages continuous scores of underlying base classifiers.
- Novel application of encoding schemes such as ECOC to decompose (encode) *unsupervised* multi-class classification problem into multiple binary-class classifications.
- Novel use of *maximum* as an order statistic to encode the multi-class scores into multiple binary-classes.
- Novel *decoding scheme* to tackle issues associated with class imbalance in an unsupervised setting and empirical study of different noises over ensemble classifiers.

We demonstrate the effectiveness of our model through extensive simulation studies and by application on real-world data related to industrial equipment on oil rigs. The remainder of the chapter is organized as follows. In Section 4.3, we review the gap in the existing literature. Section 4.4 describes the structure of the base classifiers and our proposed unsupervised ensemble learning model and our approach to addressing the issues of class imbalance and noise. In Section 4.5, we provide an extensive simulation study to assess the performance of our method under various scenarios. We demonstrate the application of our model on real-world case study data in Section 4.6. The chapter concludes with a summary.

4.3 Literature Review

Research on ensemble classifiers leverage the results of base classifiers to achieve prediction capabilities better than the best base classifier. In supervised settings, classifiers aim to optimize the overall area under the receiver operating curves (AUC) of the ensemble [15], while in semi-supervised setting, labeled data are used to improve the worst case performance of the ensemble classifiers [100]. However, in a fully unsupervised setting, the challenge lies in generating estimates for base classifier accuracy (without training data) that can be used as weights for the ensemble.

Research on inferring mixture of product distributions for latent variable models has drawn significant interest [53]. Foundational theory for using spectral methods (especially tensor decomposition methods for moment matching) to achieve this is presented in [10]. Utilizing this idea, [117] demonstrate the use of spectral methods for unsupervised binary classification. They show that spectral decomposition of the covariance matrix (second-order moment) of predicted binary labels can help estimate the ensemble's balance accuracy (weights). Their work was further enriched by [71] who showed theoretically that, relying only on second-order moments from class labels (covariance matrix) is insufficient for solving a multi-class classification problem such as ours.

In multi-class problem settings, [189] solved the problem of estimating the accuracy of base classifiers by using spectral decomposition of second and third-order sample moments. The researchers here refine their initial estimates of base classifier accuracy using EM algorithms which, however, is known to suffer from local optima. This method was refined by [160] who also utilized moment matching techniques and leveraged unknown class prior probability. However, all these methods rely on discrete class labels for generating unsupervised multi-class ensemble classifiers and do not leverage the discriminatory information contained in continuous scores of base classifiers. Further, although the model proposed by [160] tackles the issue of class imbalance by using Maximum A Posterior (MAP) estimates

(rather than Maximum Likelihood (ML) estimates), there is no existing work that studies the impact of different noise structures on the classification problem [79] in multi-class unsupervised settings. In this chapter, we propose to use a Bernoulli noise model inspired by the uniform label noise model of [82]) to model label noise and a white noise" process to model feature noise for empirically study.

[2] propose to use moments generated by ranks of binary classifiers scores rather than discrete labels. Rank statistics preserve the discrimination of continuous scores, as asymptotically ranks are highly correlated with the underlying variate [154], while avoiding any assumption of distribution[54]. However, extending the idea of ranking sample scores from binary to the multi-class case is not straightforward. To overcome this challenge and to leverage the benefits of continuous class scores, we propose to decompose the original multi-class problems into multiple binary-class problems. We study the OVA [128] and ECOC [41], [91], [49] schemes for multi-class to binary-class decomposition in detail in section 4.4.2. Authors like [130], [17] have proposed theoretical extensions for applying OVA and ECOC schemes. ECOC and OVA schemes have been studied and compared, in supervised [7], [195] and partially supervised settings [88], [150], where ECOC usually outperforms OVA scheme. However, no ECOC method exists for the case of complete unsupervised classification. Apart from this a novel statistical aspect considered in the chapter is the proposition to use maximum of base algorithm scores in decomposing the multi-class problems to multiple binary class case. Max linear combinations have shown significant improvements in pattern analysis literature [60], however its application for unsupervised classification still remains unstudied.

Thus, in this chapter our proposed UMEC model leverages the ECOC scheme using continuous scores of base classifiers as inputs and explores the use of order statistic other than *mean* to decompose them into binary class scores. The proposed model also tackles the issue of class imbalance. We compare our model with state-of-the-art models [189] [160], while studying the impact of different noise structure empirically and demonstrate the effectiveness of using continuous scores and other order statistics in generating an optimal

unsupervised multi-class ensemble classifier.

4.4 Model Development

Figure 4.2 shows the framework for our unsupervised multi-class ensemble classification (UMEC) model. It comprises of four steps. The first step is *generating unsupervised base classifiers*, where we generate mutually independent base classifiers, either by leveraging distinct facets of domain knowledge or by using different design principles as discussed in Section 4.4.1 and 4.6. In the second step we *encode multi-class problem to multiple sub-binary class problems* by generating binary groupings (indicated by +1 and -1) using a novel reduction statistics that decompose multi-class continuous scores into multiple binary classes (section 4.4.2). In the third step we perform *spectral decomposition of second and third order moments for each sub-binary problem.* The moments are generated using the ranks of difference between the reduction statistics for each base algorithm (section 4.4.3). Finally, in the fourth step we *decode the outputs of spectral decomposition to predict multi-class labels* where we propose a novel algorithm to tackle class imbalance (section 4.4.4).

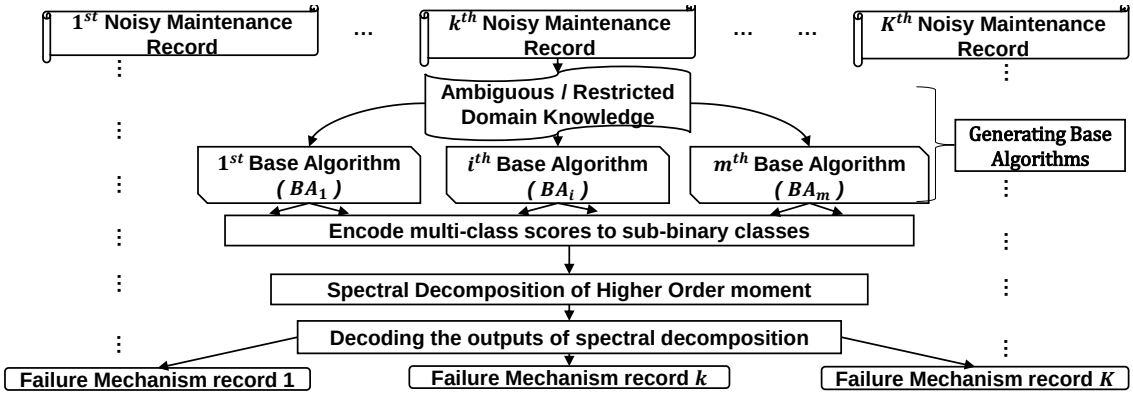


Figure 4.2: Structure of Unsupervised multi-class ensemble classification (UMEC) model

4.4.1 Generating unsupervised base classifiers

We provide a brief description of the base algorithms used for maintenance record classification based on different facets of domain knowledge. We denote the classification problem to be a C class classification problem with $c \in \{1, \cdots, C\} \equiv \mathbf{C}$. We assume that the features of each base algorithms are noisy and denoted by $\tilde{x_i} \in \tilde{\mathbf{x}}_\mathbf{i}$, and due to incorrect-labeling sometimes the true class label $c = c^t$ could be incorrectly observed as $\tilde{c} = c^n$. Thus, the classification problem for each base algorithm is given by equation 4.1.

$$p(\tilde{c} = c^n|\tilde{\mathbf{x}}_\mathbf{i}) = \sum_{c^t=1}^{c^t=C} p(\tilde{c} = c^n|c = c^t)p(c = c^t|\tilde{\mathbf{x}}) \tag{4.1}$$

We have M such base algorithms denoted by $i, j \in \{1, \cdots M\} \equiv \mathbf{M}$ for classifying maintenance records $k \in \{1, \cdots K\} \equiv \mathbf{K}$. We represent the individual score of any class (c) of k^{th} maintenance record generated by i^{th} base algorithm as $x^c_{i,k}$. We consider this score to be generated from the probability density function represented as $f^c_i(x)$ and its k^{th} realization for k^{th} maintenance record as $f^c_{i,k}(x)$. Let $\mathbf{X}^\mathbf{c}_\mathbf{i}$ denote the set of all scores $x^c_{i,k} \sim f^c_i(x)\ \forall\ k \in \mathbf{K}$. The predicted class of k^{th} maintenance by i^{th} base classifier is decided by the rule $\hat{c}^i_k = argmax_{c\ \in\ \mathbf{C}}\{f^c_{i,k}(x)\}$. Thus, for each base algorithm, the predicted class is the class the with maximum score.

4.4.2 Encoding multi-class problem to multiple sub-binary class

In order to ensemble the outcomes of the base classifiers the multi-class classification problem is decomposed into to multiple sub-binary problems have been a long-advised method to solve multi-class classification problem, using strategies like One-Versus-All (OVA) and Error-Correcting Output Code (ECOC) [7]. We show the encoding matrices for five failure mechanism classes using OVA and ECOC decomposing schemes in Figure 4.3. Each decomposing schemes is represented as a matrix with entries $\in \{-1, 1\}$. Each *column* of the matrix is used as a guide to generate binary classifiers. Here, a class corresponding to "-1"

indicates that the class is placed in the negative group of the binary decomposition while "+1" implies that the class is in the positive group. *Please note that we do not study AVA for our application because the outputs of the spectral decomposition scheme used in this chapter are binary, and we do not have a-prior information of class labels in unsupervised classification. Thus, we cannot identify the records that must be considered as left out in the AVA comparison scheme.*

In the ECOC scheme, a class is represented by the binary bits vector corresponding to the row of the ECOC matrix $\boldsymbol{E}$. Here the scores in the binary classes '+1' and '-1' are reduced using appropriate reduction statistics and are compared with each other. The basic intuition behind ECOC scheme is that error propagating per bit of the class vector is minimized continually as more binary classifiers are added. To ensure high separation between classes, the Hamming distances between rows (and thus, between classes) of the ECOC matrix $\boldsymbol{E}$ must be large. The Hamming distance between two strings of symbols (+1, -1 for rows of ECOC matrix) is a measure of the number of positions at which the corresponding symbols differ. The higher the distance between rows, the better the efficiency of ECOC scheme in separating classes. To generate cost-effective ECOC columns constraints are implied in the selection of ECOC columns [195]. In our setting, however, running a binary classifier is not costly. Thus, we sample all ECOC columns with no-duplication constraints. We also ensure that no two columns in the ECOC matrix are exact negatives of each other (complementing each other) [41]. Also, it is observed that as the pairwise distance between columns decreases marginally on account of the addition of new binary columns in ECOC, the distance between rows increases significantly, thus providing higher class separation. Algorithm 5 outlines our approach for decomposing and encoding multi-class scores from the base algorithms into binary groups of positive and negative class. Figure 4.3b provides a graphical illustration of various steps involved in algorithm 5.

The usual choice suggested for the reduction operator Θ is the *mean* function. However, in Proposition 1, we propose the use of order statistics which are a function *maximums* for

Algorithm 5: Decomposing and encoding multi-class scores to binary classes

Input: $\mathbf{M}, \mathbf{C}, \mathbf{K}, \mathbf{X_i^c}, \boldsymbol{E}$

Output: $\mathbf{R_i^u}$ Set of sorted ranks for base algorithm i corresponding to binary classifier of $u \in \boldsymbol{E}$

for $u \in \boldsymbol{E}$ **do**
- Partition $\mathbf{C} \in \mathbf{C_u^P}, \mathbf{C_u^N}$ /* $\blacktriangleright$ $\mathbf{C_u^P} \cup \mathbf{C_u^N} \equiv \mathbf{C}$ & $\mathbf{C_u^P} \cap \mathbf{C_u^N} \equiv \emptyset$ */
- /* where $\mathbf{C_u^P} \ni \{c^{u_p}\} \mid u_p \in u$ & $u_p = +1$ are classes partitioned into +1 group & $\mathbf{C_u^N} \ni \{c^{u_n}\} \mid u_n \in u$ & $u_n = -1$; are classes partitioned into -1 group by ECOC column (u) */
- **for** $i \in \mathbf{M}$ **do**
  - Initialize $\delta_\mathbf{i} \equiv \emptyset$ & $\mathbf{R_i^u} \equiv \emptyset$
  - /* where $\delta_\mathbf{i}$ is the set of difference statistics for each algorithm i and $\mathbf{R_i^u}$ is the set of ranks for difference statistics $\delta_\mathbf{i}$ */
  - **for** $k \in \mathbf{K}$ **do**
    - Let $\mathbf{X_{i,k}^{C_u^P}} \ni x_{i,k}^c \; \forall \; c \in \mathbf{C_u^P}$
    - Let $\mathbf{X_{i,k}^{C_u^N}} \ni x_{i,k}^c \; \forall \; c \in \mathbf{C_u^N}$
    - $t_{i,k}^{C_u^P} \rightarrow \Theta(\mathbf{X_{i,k}^{C_u^P}})$
    - $t_{i,k}^{C_u^N} \rightarrow \Theta(\mathbf{X_{i,k}^{C_u^N}})$
    - /* where $t_{i,k}^{C_u^P}$ & $t_{i,k}^{C_u^N}$ are the reduction statistics for the class scores in +1 & -1 groups respectively & Θ is the reduction operator */
    - $\delta_{i,k}^u = t_{i,k}^{C_u^P} - t_{i,k}^{C_u^N}$
    - $\delta_\mathbf{i}^\mathbf{u} \rightarrow \delta_\mathbf{i}^\mathbf{u} \cup \{\delta_{i,k}^u\}$
  - $\mathbf{R_i^u} \rightarrow \ll \delta_\mathbf{i}^\mathbf{u} \gg$ /* sorting $\delta_\mathbf{i}^\mathbf{u}$ to generate ranks $r_{i,k}^u \in \mathbf{R_i^u} \; \forall \; k \in \mathbf{K}$ */

the reduction operator. As highlighted in [60], max statistics introduces non-linearity and are non-smooth in nature, thus making the use of traditional estimation methods challenging. A proof by contradiction is provided in the supplementary file (section 1), where we also show that the Kullback-Leibler (K-L) divergence and Jensen-Shannon (J-S) divergence of scores in positive and negative groups generated using *maximum* as reduction statistic are empirically better than using *mean* as reduction statistic.

Proposition 1 *Suppose a classification process is such that the scores of each class $c \in \mathbf{C}$ are distributed according to the pdf $f^c(x)$. At a given instant k, the classifier predicts the class to be $\hat{c}$ such that $\hat{c} = argmax_{\{c \in \mathbf{C}\}}(f_k^c(x))$. Let $t_k^{CP}, t_k C^N$ be the reduction (decomposition) statistics to decompose scores of sample k of a multi-class classifier into binary groups (of*

positive $\mathbf{C^P} \subset \mathbf{C}$ *and negative classes* $\mathbf{C^N} \subset \mathbf{C}$*;* $\mathbf{C^P} \cup \mathbf{C^N} \equiv \mathbf{C}$*) for multiple comparisons. Then, a reduction (decomposition) statistic, which is a function of* ***maximums*** *of scores in each binary group* ($t_k^{C^P} = \zeta(max(\mathbf{X_k^{C^P}})), t_k^{C^N} = \zeta(max(\mathbf{X_k^{C^N}}))$*), performs better than any other reduction (decomposition) statistic.*

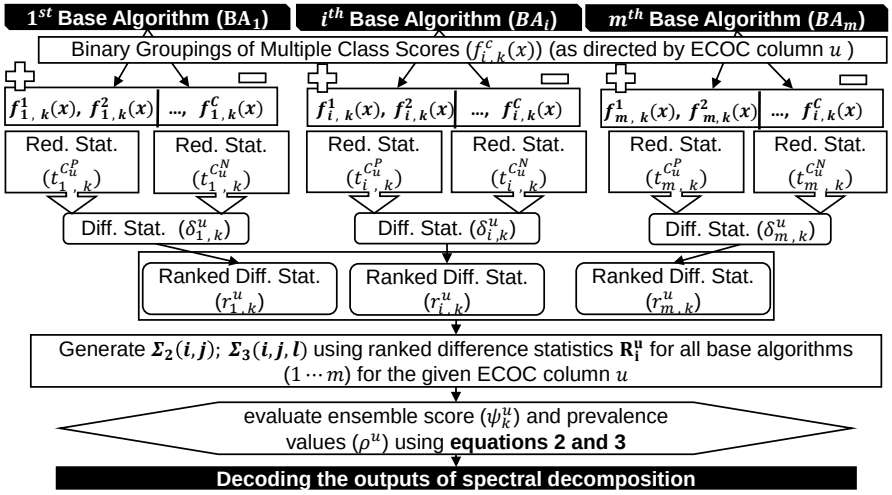
ECOC-Scheme

Material Failure	1	-1	-1	-1	-1	-1	1	1	1	-1	1	-1	1	1	1
Mechanical Failure	-1	1	-1	-1	-1	1	-1	-1	1	1	1	1	1	-1	1
Hydraulic Failure	-1	-1	1	-1	-1	-1	1	1	1	1	-1	1	-1	-1	-1
Control Failure	-1	-1	-1	1	-1	1	-1	1	-1	-1	-1	1	-1	1	1
Electrical Failure	-1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	-1	1	-1	-1

OVA-Scheme

	Material Failure	Mechanical Failure	Hydraulic Failure	Control Failure	Electrical Failure
Material Failure	1	-1	-1	-1	-1
Mechanical Failure	-1	1	-1	-1	-1
Hydraulic Failure	-1	-1	1	-1	-1
Control Failure	-1	-1	-1	1	-1
Electrical Failure	-1	-1	-1	-1	1

(a) Different encoding strategies for multi-class to binary class decomposition.



(b) Steps to encode multi-class scores of base algorithms to multiple binary groups using ECOC scheme and determination of ensemble score ψ_k^u.

Figure 4.3: Demonstration of Encoding multi-class classification problem to multiple sub-binary class classification problem

4.4.3 Spectral decomposition of higher order moments

The rank statistic $\mathbf{R_i^u}$ (in 5) obtained for each base algorithm $i \in \mathbf{M}$ for a given binary classifier $u \in \boldsymbol{E}$ are used to generate empirical second order covariance matrix $\hat{\Sigma}_2^u(i,j)$ and third order covariance tensor $\hat{\Sigma}_3^u(i,j,l)$ of ranks predicted by base classifiers $i,j,l \in \mathbf{M}$. The performance of i^{th} binary base classifier can be measured as $\Delta_i^u = \mathbb{E}[\mathbf{R_i^u}|c=-1] - \mathbb{E}[\mathbf{R_i^u}|c=1]$ where $\mathbb{E}[\mathbf{R_i^u}|c=j]$ for j=-1,1 represents the average rank given the respective class for the

i^{th} base classifier and u^{th} ECOC binary groups. For q^{th} order conditionally independent classifiers, it has been shown in [2] that their q^{th} order covariance tensor, Σ_q^u, (which is defined as $\Sigma_q^u(1,\cdots,q) := \mathbb{E}[(\mathbf{R_1^u} - \mathbb{E}[\mathbf{R_1^u}])\cdots(\mathbf{R_q^u} - \mathbb{E}[\mathbf{R_q^u}])]$) follows the equality $\Sigma_q^u(1,\cdots,q) = H(\rho^u)((\rho^u)^{q-1} - (\rho^u - 1)^{q-1})\prod_{i=1}^{q}\Delta_i^u$, where $H(\rho^u) = \rho^u(1-\rho^u)$ and ρ^u denotes the prevalence of the positive class. Rank one decomposition of covariance matrix $\Sigma_2^u(i,j)$ of rank statistics ($r_{i,k}^u \in \mathbf{R_i^u}\ \forall\ k \in \mathbf{K}$) is estimated as $\Sigma_2^u(i,j) = \kappa_c^u \vec{\nu^u}\vec{\nu^u}^T - diag(\kappa_c^u \vec{\nu^u}\vec{\nu^u}^T) + \frac{N^2-1}{12}I$ using iterative Singular Value Decomposition (SVD) of rank one matrix generated by off-diagonal element of $\Sigma_2^u(i,j)$ (appendix (section 4.9)). Here $\kappa_c^u \vec{\nu^u}\vec{\nu^u}^T$ is a rank one matrix with $\kappa_c^u = H(\rho^u)||\Delta^u||_2^2$ and $\nu_i^u = \frac{\Delta_i^u}{\sqrt{\Sigma^u{}_{j=1}^{M}\Delta^u{}_j^2}}$ is a unit norm-vector which represents the normalized accuracy of each base classifier and are used as weights of each base algorithms in final *estimated ensemble score* (ψ_k^u) for each maintenance record $k \in \mathbf{K}$ and binary classifier $u \in \mathbf{E}$ as given in equation 4.2.

$$\psi_k^u = -\sum_{i=1}^{M}(\hat{\nu}_i^u r_{i,k}^u) \tag{4.2}$$

In order to estimate the prevalence (ρ^u) of positive class the third-order covariance tensor $\Sigma_3(i,j,l)$ is generated using the rank statistics of the base algorithms which follows the equality $\Sigma_3^u(i,j,l) = H(\rho^u)(2\rho^u - 1)\Delta_i^u\Delta_j^u\Delta_l^u$ for each base algorithm $i,j,l \in \mathbf{M}$. The covariance tensor is off-diagonal rank-one and follows the decomposition given by $\Sigma_3^u(i,j,l)^u = \kappa_t^u \vec{\alpha}^u \bigotimes \vec{\alpha}^u \bigotimes \vec{\alpha}^u$. The value of κ_t^u & $\vec{\alpha}^u$ is obtained by iterative algorithm using Singular Value Tensor Decomposition (appendix (section 4.9)) of $\Sigma_3^u(i,j,l)$. Using estimated κ_c^u and κ_t^u class prevalence (ρ^u) is estimated as shown in equation 4.3.

$$\rho^u = \tfrac{1}{2} \pm \tfrac{1}{2}\sqrt{\tfrac{\tau^u}{\tau^u+4}},\ \text{where}\ \tau^u = (\kappa_t^u)^2/(\kappa_c^u)^3 \tag{4.3}$$

4.4.4 Decoding the outputs of spectral decomposition

The ensemble scores (equation 4.2) and predicted class labels generated using spectral decomposition in Section 4.4.3, for each binary comparison $u \in \mathbf{E}$ are then decoded to generate final predicted multi-class label. Traditionally in the decoding process, hard labels predicted by each binary comparison ($u \in \mathbf{E}$) for every sample $k \in \mathbf{K}$ are stored to generate a binary vector of prediction $\vec{p_k}$. We calculate the distance between the failure mechanism class vector $\vec{w_c}$ (given by rows of ECOC matrix $\boldsymbol{E}$) and the prediction vector $\vec{p_k}$, for a given sample k. The class closest to sample k is the predicted class for the sample k. Traditionally Hamming distance is proposed as a distance measure between the vectors $\vec{p_k}$ and $\vec{w_c}$, however we use the distance measure (ς) inspired from [7] (equation 4.4), which relies on calculating the dot product between $\vec{p_k}$ and $\vec{w_c}$.

$$\varsigma = (|\vec{w_c}| - \vec{w_c} \odot \vec{p_k})/2 \tag{4.4}$$

Rather than using the hard binary labels for the decoding process, we propose to use the ensemble-scores ($\psi_k^u \in \mathbf{\Psi^u}$) of equation 4.2 when the dataset is known to be approximately balanced, and use prevalence normalized ensemble scores ($\psi_{\rho,k}^u \in \mathbf{\Psi_\rho^u}$) when the dataset is known to be imbalanced. It is quite common for industrial datasets (such as maintenance records) to have a high level of imbalance. For the Mud-Pump equipment studied in this chapter, *Material Failure* mechanism class is much more prevalent than other failure mechanisms. In our Algorithm 6 outlined below, we address the challenge of class imbalance by normalizing the ensemble scores with the estimates of the binary classifier prevalence values (ρ) predicted using equation 4.3. We describe the decoding process for a maintenance record using different types of prediction vectors. The prediction vectors $\vec{p_{k,d}}, d \in \{\ell_k,\ \ell_{\rho,k},\ \psi_{\rho,k}\}$ correspond to different binary labels and scores generated using algorithm 6 as shown in Figure 4.4. The advantage of using continuous scores as opposed to discrete labels is that the former always produces distinct values whereas the latter may result in non-unique distances

for multiple classes. For example, in Figure 4.4 we can see that distance (ς) using $\vec{w_c}$ and $p_{k,\vec{\ell}_{\rho,k}}$ or $p_{k,\vec{\ell}_k}$ generated for binary labels have minimum values for multiple classes. Using vector of continuous scores $p_{k,\vec{\psi}_{\rho,k}}$ resolves this ambiguity, while normalizing the predicted labels by class prevalence helps to overcome class imbalance.

Material	-1	-1	1	1	-1	1	-1	-1	1	1	1	1	-1	-1	-1
Mechanical	-1	1	-1	1	-1	1	-1	1	1	1	-1	-1	1	1	1
Hydraulic	1	-1	-1	-1	-1	1	-1	1	-1	-1	1	-1	-1	-1	1
Control	-1	-1	-1	-1	-1	-1	1	-1	1	-1	1	1	1	1	1
Electrical	-1	-1	-1	-1	1	-1	-1	-1	-1	1	-1	-1	1	-1	-1
p_{k,ℓ_k}	1	1	-1	-1	1	-1	1	1	-1	-1	-1	-1	1	1	1
$p_{k,\ell_{\rho,k}}$	-1	1	-1	-1	1	-1	-1	1	-1	1	-1	-1	1	1	1
$p_{k,\psi_{\rho,k}}$	-0.07	0.14	-0.62	-0.61	0.92	-0.83	-0.07	0.17	-0.65	0.07	-0.63	-0.64	0.59	0.1	0.14

ℓ_k	$\ell_{\rho,k}$	$\psi_{\rho,k}$
15	12	10.41
7	4	7.38
7	8	7.73
7	8	7.67
7	4	4.93

Generating Distance between $p_{k,d}$ and w_c

Figure 4.4: Results using different decoding vector $\vec{p_{k,d}}, d \in \{\ell,\ \ell_\rho,\ \psi_\rho\}$

4.5 Simulation Experiments

We evaluate the performance of the proposed UMEC model using empirical studies. In order to conduct simulation studies, we simulate the three base algorithms which follow the structure as given in equation 4.1. We simulate scores for five classes for each of the three base algorithms. To generate simulation data, we fix the prevalence of each class and generate ground truth labels as per the fixed prevalence. In order to generalize behavior of the proposed UMEC model, we model the scores $f_i^c(x)$ for each class $c \in \mathbf{C}$ and base algorithm $i \in \mathbf{M}$ using multi-variate Gaussian (MVN), exponential (exp), and Gamma distributions separately. We begin by generating a set of sample scores for all classes of each base algorithm. We select only those set of scores for which the class with the maximum score corresponds to the simulated ground truth class label. This way of sampling necessarily preserves class prevalence in simulated samples. In order to simulate feature noise to generate distorted features $\tilde{x_i^c}$, we assume white noise to be added over the simulated feature scores

Algorithm 6: Prevalence normalization for decoding process

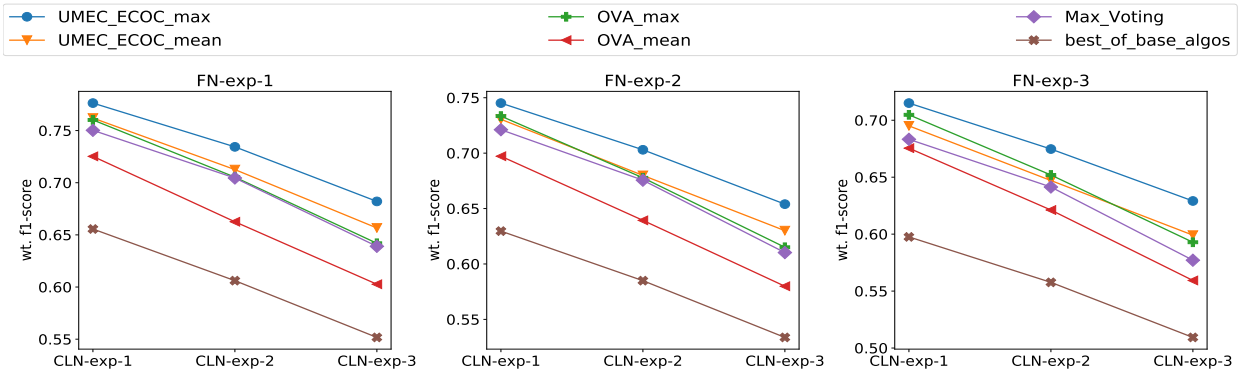
Input: $\mathbf{R_i^u}$ Set of sorted ranks for base algorithm i
Output: $\mathcal{L}^{\mathbf{u}}, \mathbf{\Psi^u}, \mathcal{L}_\rho^{\mathbf{u}}, \mathbf{\Psi}_\rho^{\mathbf{u}}$

for $u \in \boldsymbol{E}$ **do**
  Initialize $\mathbf{\Psi}_\rho^{\mathbf{u}}; \mathcal{L}_\rho^{\mathbf{u}}, \mathbf{\Gamma_p}; \mathbf{\Gamma_n}; \mathbf{O_p}; \mathbf{O_n} \rightarrow \emptyset$
  $\mathbf{R^u} = \mathbf{R^u} \cup \mathbf{R_i^u} \;\; \forall \;\; i \in \mathbf{M}$
  $\mathbf{\Psi^u} \rightarrow$ Apply equation 4.2 on $\Sigma_2(i,j)\ of\ \mathbf{R^u}$
  if $\Psi_k^u \geq 0$ **then return** $\ell_k^u = 1$;
  else return $\ell_k^u = -1$;
  $\mathcal{L}^{\mathbf{u}} \rightarrow \mathcal{L}^{\mathbf{u}} \cup \{\ell_k^u\} \forall k \in \mathbf{K}$
  $\rho^u \rightarrow$ Apply equation 4.3 using $\Sigma_2(i,j), \Sigma_3(i,j,l)\ of\ \mathbf{R^u}$
  $\mathbf{\Phi^u} \rightarrow sort(\mathbf{\Psi^u})$
  $n_p = \lceil K \times \rho \rceil; n_n = \lceil K \times (1-\rho) \rceil;$
  for $\tilde{k} \in \mathbf{K}$ **do**
    if $\tilde{k} \leq n_n$ **then**
      $\mathcal{L}_\rho^{\mathbf{u}} \rightarrow \mathcal{L}_\rho^{\mathbf{u}} \cup \{-1\}; \mathbf{\Gamma_n} \rightarrow \mathbf{\Gamma_n} \cup \mathbf{\Phi^u}[\tilde{k}]; \;\; \mathbf{O_n} \rightarrow \mathbf{O_n} \cup \tilde{k}$
    else
      $\mathcal{L}_\rho^{\mathbf{u}} \rightarrow \mathcal{L}_\rho^{\mathbf{u}} \cup \{1\}; \mathbf{\Gamma_p} \rightarrow \mathbf{\Gamma_p} \cup \mathbf{\Phi^u}[\tilde{k}]; \;\; \mathbf{O_p} \rightarrow \mathbf{O_p} \cup \tilde{k}$
  $\widehat{\mathbf{\Gamma_n}} \rightarrow$ MinMaxScaler($\mathbf{\Gamma_n}$);
  $\widehat{\mathbf{\Gamma_p}} \rightarrow$ MinMaxScaler($\mathbf{\Gamma_p}$);
  for $\tilde{k} \in \mathbf{K}$ **do**
    if $\tilde{k} \in \mathbf{O_n}$ **then**
      $\mathbf{\Psi}_\rho^{\mathbf{u}} \cup \widehat{\mathbf{\Gamma_N}}[\tilde{k}] \;\; | \;\; \tilde{k} \in \mathbf{O_n}[\tilde{k}];$
    else
      $\tilde{k} \in \mathbf{O_p}$
    $\mathbf{\Psi}_\rho^{\mathbf{u}} \cup \widehat{\mathbf{\Gamma_p}}[\tilde{k}] \;\; | \;\; \tilde{k} \in \mathbf{O_p}[\tilde{k}]$

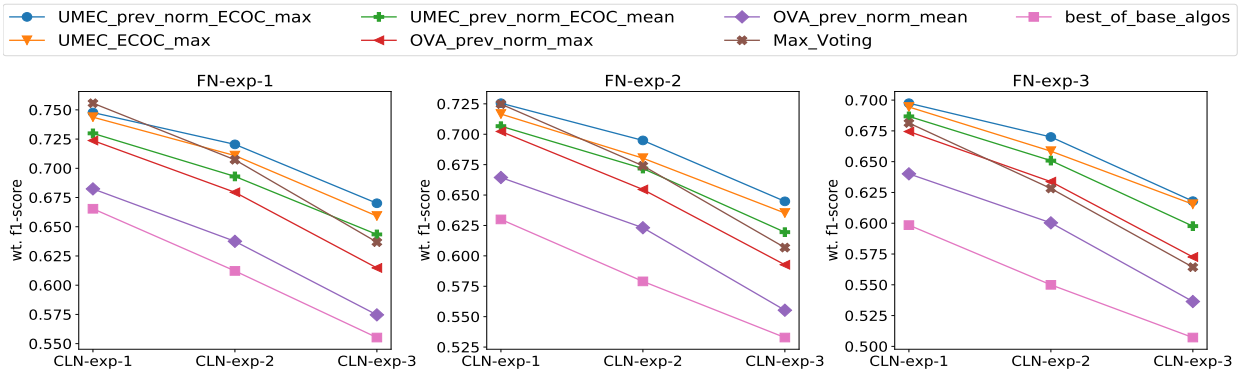
($x_i^c \sim f_i^c(x)$) giving us noisy features as $\tilde{x_i^c} = x_i^c + p_1 \times \mathcal{N}(0, \sigma^2)$, where p_1 is the proportion (power) of the white noise to be added. We adjust p_1 depending on the type of distribution selected to generate each class score. To simulate label noise, we use a Bernoulli noise structure. We randomly sample p_2 proportion of samples from the simulated dataset and assign wrong labels to them by swapping the scores. The swapping of scores results in a change in class label corresponding to maximum score, thus mislabeling the sample. We generate 2500 samples for the three base algorithms for balanced and unbalanced cases. For all distribution ($f_i^c(x) \sim \mathcal{D} \; \forall \; \{\mathcal{D}_f\} \in$\{exp, MVN, Gamma\}) we set the prevalence value to be 0.2 for balanced case and prevalence value to be $\{0.17, 0.18, 0.39, 0.10, 0.16\}$ for unbalanced case. We generate 9 simulation sets by varying feature noise at three levels {"FN-$\mathcal{D}_f$-1", "FN-$\mathcal{D}_f$-2", "FN-$\mathcal{D}_f$-3"}. Corresponding to each feature noise level, three levels of mislabel noise are simulated {"CLN-$\mathcal{D}_f$-1", "CLN-$\mathcal{D}_f$-2", "CLN-$\mathcal{D}_f$-3"}. The value of parameters used to generate simulated samples are given in appendix (section 4.10). We discuss the results for simulation runs of scores distributed as $\mathcal{D} \in \{MVN\}$ below, while the results for exponential and Gamma distributed scores are discussed in appendix section 4.10.1 and section 4.10.3 respectively).

In Figures 4.5, 4.6 and 4.7c our proposed method is mentioned as *UMEC_ECOC* when decoding is done using $\mathbf{\Psi^u}$ and as *UMEC_prev_norm_ECOC* when decoding is done using $\mathbf{\Psi^u_\rho}$. The reduction statistic (*max* or *mean*) used in encoding are subsequently mentioned. We compare our method with *Max-Voting* scheme along with the best performing base algorithm (*best_of_base_algos*). The results for balanced and unbalanced case for exponential distribution are shown in Figure 4.5. In Figure 4.5a, which consists of balance datasets it can be seen that our proposed method (*UMEC*) with *ECOC* encoding scheme and using $\mathbf{\Psi^u}$ for decoding produces best results. Also, it is worth noting that when the reduction order statistics is *max* we get best results as indicated by proposition 1. It can also be inferred that increasing feature and class label noise both degrades the performance of all the algorithms. For unbalanced dataset generated using exponential distribution as shown in Figure 4.5b,

we see that decoding done using prevalence normalization $\mathbf{\Psi}^{\mathbf{u}}_{\rho}$ provides the best results for *UMEC* model with *ECOC* encoding scheme in most cases.



(a) Simulation results for exp distributed class scores (balanced case).



(b) Simulation results for exp distributed class scores (unbalanced case)

Figure 4.5: Simulation results for exp distributed class scores

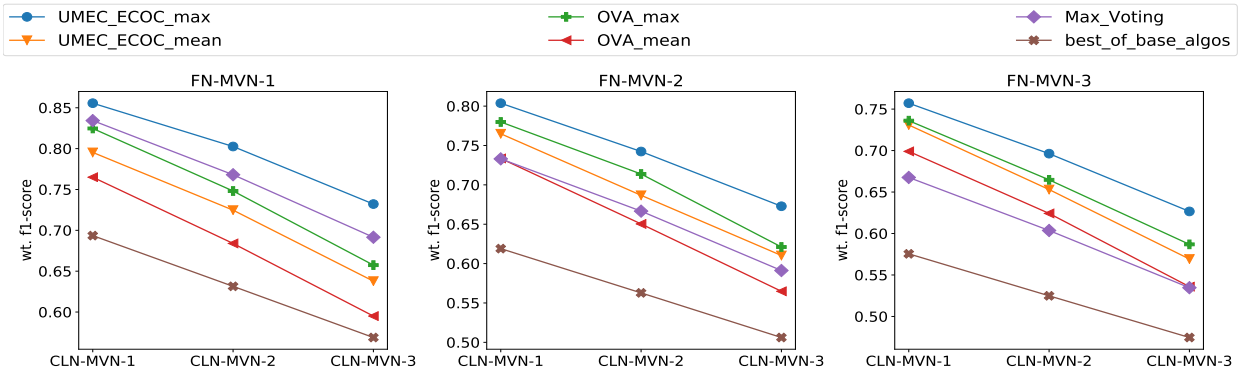
Figure 4.6 provides similar results as Figure 4.5, however, here the underlying distribution for score generation is MVN. Figure 4.6a again shows that for balanced case our proposed *UMEC* method with *ECOC* encoding scheme and $\mathbf{\Psi}^{\mathbf{u}}$ decoding scheme produces best results. For unbalanced case (shown in Figure 4.6b), we also compare the performance of state-of-the-art models from [160] (referred to as Blind_Ensemble_ML, Blind_Ensemble_MAP) and from [189] (referred as EM_Spectral and EM_Max_Voting) and also increase the level of feature noise (as described in section 3.4 of supplementary file) for drawing conclusive inference. It can be observed that the state-of-the-art methods preform better than our proposed method when the feature noise level is low, however, as the feature noise level increases our proposed method beats other state-of-the-art models. Also the toughest competition is given

by the models proposed in [160] where MAP estimates does a better job than ML estimates (presumably due to unbalance nature of the simulated data). For UMEC model too, we achive best results by using prevalence normalization $\mathbf{\Psi}_{\rho}^{\mathbf{u}}$ for decoding. For unsupervised maintenance record classification the feature noise tend to arise due to noisy words present in the maintenance records as shown in Figure 4.1b. In many text classification problems such feature noise tend to occur more often. As can be seen from Figure 4.6b, that when the feature noise is substantial (which mostly will be the case for real world maintenance records) using the proposed UMEC model with $\mathbf{\Psi}_{\rho}^{\mathbf{u}}$ scores for decoding would help industry practitioner achieve better accuracy.

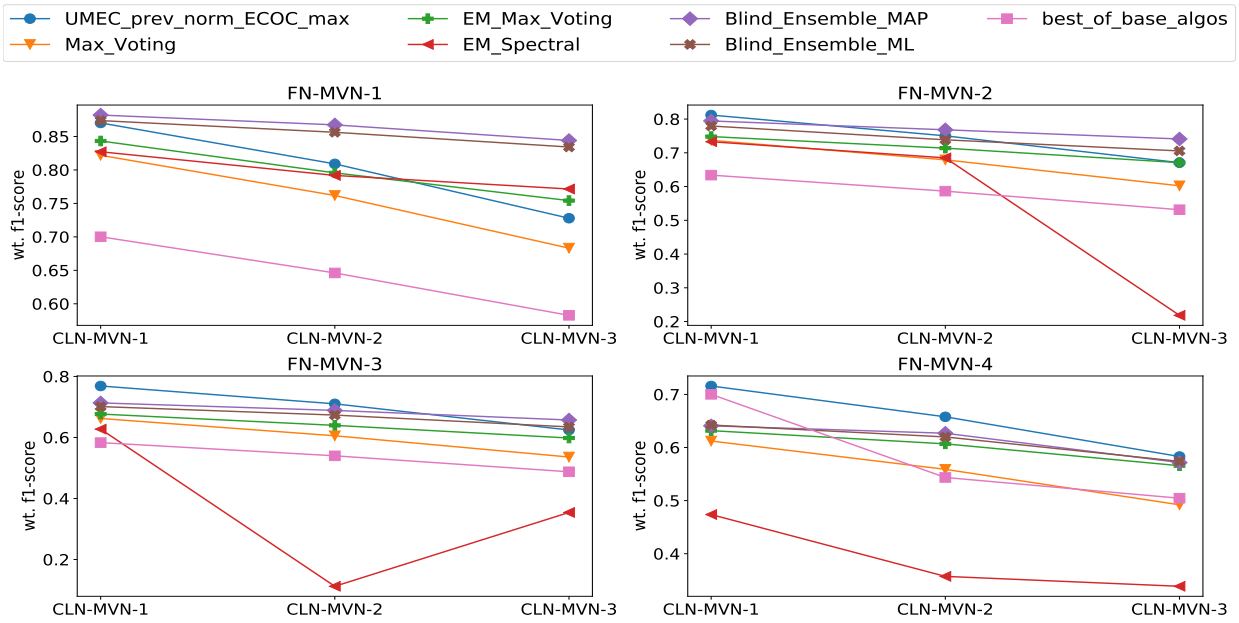
For distributions dependent on location parameters like MVN, the location parameters of the underlying distributions could impact the prevalence of a class. This is because the structure of the base algorithms is such that the class with maximum score (x_i^c) is the winner class. Thus, the number of times the particular class would have a maximum score (compared to other classes) will depend on how far the particular class' location parameter is from other classes. In order to check such a case, we also simulate an unbalanced multivariate Gaussian dataset with the same scale parameter as in Figure 4.6b, however, location parameters depend on class prevalence values as demonstrated in supplementary materials. The results for unbalanced dataset for MVN distribution with varying location parameter show a similar and a more concrete trend as of Figure 4.6b. We provide further discussion and report the results in appendix section 4.10.4.

4.6 Case Study

In this chapter, we employ three base classifier algorithms. We term the first base algorithm as *token matching algorithm*. It uses a list of tokens for failure mechanisms defined by the ISO standard (Table 4.1) and subject matter experts. Term-Frequency Inverse Document Frequency (TFIDF) scores are generated for each word in the maintenance records



(a) Simulation results for MVN distributed class scores (balanced case).



(b) Simulation results for MVN distributed class scores (unbalanced case)

Figure 4.6: Simulation results for MVN distributed class scores

[129]. The TFIDF scores indicate a word's prominence in a given maintenance record. Finally, we calculate the score of each failure mechanism class for a given maintenance record by summing the TFIDF scores of tokens associated with the failure mechanism class. The predicted failure mechanism for a maintenance record is the class with the highest summed TFIDF score.

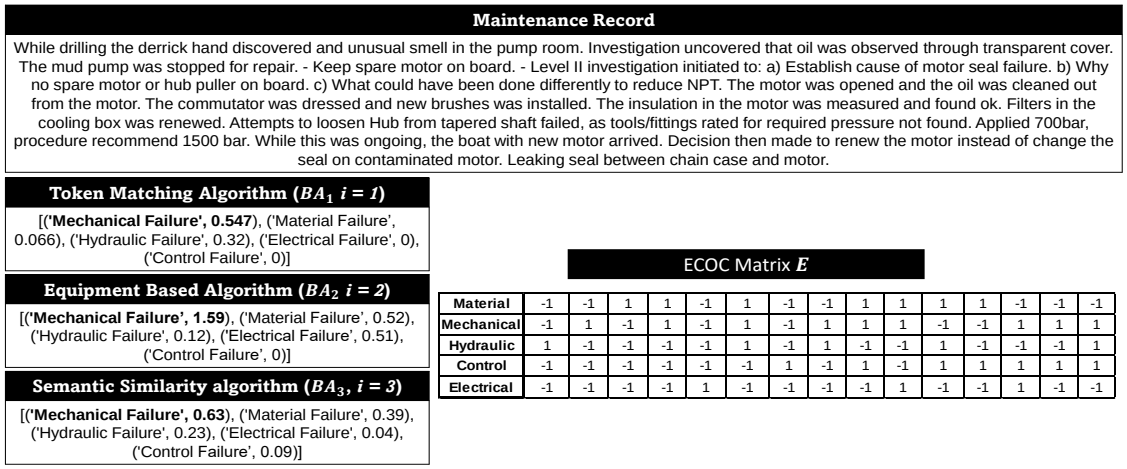
We call the second base algorithm as *equipment based algorithm*. This algorithm begins by identifying tokens or words that indicate the part/component that failed in the equipment. Then, using domain knowledge, the prominence of each failure mechanism is identified for

all the identified parts/components from the maintenance records. The score of each failure mechanism class is the weighted sum of the TFIDF scores of the identified parts/components. The prominence (order) of the failure mechanism for the concerned part/component gives us the weights for that failure mechanism class.

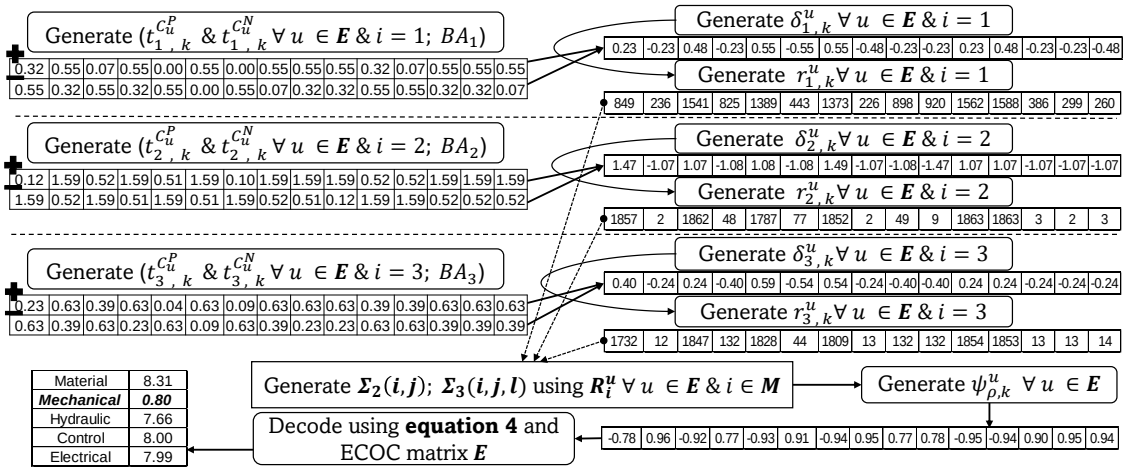
The third base algorithm is called *semantic similarity based algorithm.* This algorithm is based on a different design mechanism, and thus satisfies mutual independence. Here, we first generate word embeddings using CWEM model proposed in [20]. These embeddings help to generate a mathematical representation for the maintenance record by using sentence embedding [12]. We also generate embeddings for each failure mechanism class by averaging the word embeddings for tokens present in Table 4.1. The class with the highest similarity between the generated sentence embedding and the each class embedding is the failure mechanism for the given maintenance record. We demonstrate the application of all three algorithms for a maintenance record in Figure 4.7a.

Next as shown in Figure 4.7b, we generate reduction statistics $t_{i,k}^{C_u^P}$ & $t_{i,k}^{C_u^N}$ from scores predicted by each base algorithm i, for all binary partitions directed by each column u of ECOC matrix $\boldsymbol{E}$. Using the reduction statistics for positive $t_{i,k}^{C_u^P}$ and negative $t_{i,k}^{C_u^N}$ classes the difference score $\delta_{i,k}^u$ is generated for each base algorithm i for all ECOC column $u \in \boldsymbol{E}$. Following this ranks $(r_{i,k}^u)$ of difference score $\delta_{i,k}^u$ for all maintenance records $k \in \mathbf{K}$ are generated for each algorithm $i \in \mathbf{M}$ for each ECOC column $u \in \boldsymbol{E}$. The rank statistics $\mathbf{R_i^u}$ for each algorithm (i) and ECOC column (u) are used to generate higher order moments $(\hat{\Sigma}_2^u(i,j)$ and $\hat{\Sigma}_3^u(i,j,l))$ which are decomposed using spectral methods to generate ensemble scores $\psi_{\rho,k}^u$ using algorithm 6. The vector of scores $p_{k,\vec{\psi}_{\rho,k}}$ are used along with rows of ECOC matrix $\boldsymbol{E}$ to generate distance measure using equation 4.4. The class corresponding with minimum score is the predicted class.

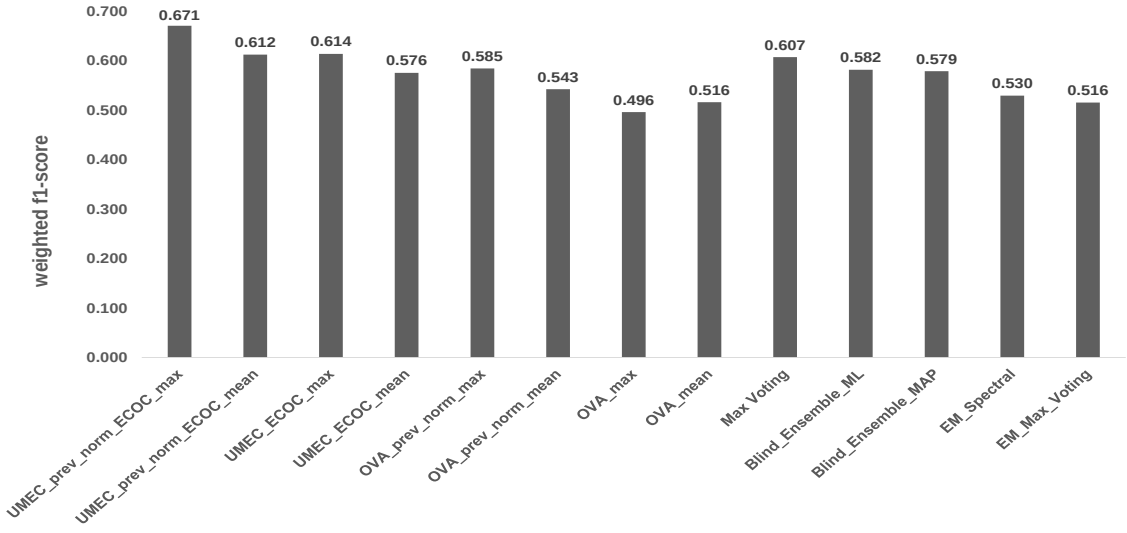
We evaluate the proposed model using a real-world industrial dataset comprising maintenance records from a mudpump used in oil rigs. For the performance evaluation, we utilized 93 maintenance records for which the correct failure mechanisms were determined and la-



(a) Generation of base algorithms and ECOC matrix $\boldsymbol{E}$ used



(b) Encoding Multi-Class scores to Multiple Binary Classes



(c) Weighted f1-score of applying different models over industrial dataset

Figure 4.7: Step by step application of UMEC model and final result over industrial dataset

beled via manual review by industry subject matter expert. Figure 4.7c shows the results of using different models over the labeled dataset. In this figure we report the weighted f1-scores for the multi class classification problem on the y-axis with bars representing f1-score for each model. From figure 4.7c it is evident that using our proposed UMEC model with ECOC encoding scheme with *max* reduction statistic and $\mathbf{\Psi}_{\rho}^{\mathbf{u}}$ scores for decoding with prevalence normalization (*UMEC_prev_norm_ECOC_max*) achieves highest accuracy than all other methods including the stat-of-the-art models. Our proposed method achieves weighted f1-score of 0.671 (*UMEC_prev_norm_ECOC_max*) while other state-of-the-art methods achieves the f1 scores of 0.582 (Blind_Ensemble_ML), 0.579 (Blind_Ensemble_MAP), 0.53 (EM_Spectral), 0.516 (EM_Max_Voting) and 0.607 (Max-Voting) on the industrial dataset. The low performance of state-of-the-art models can be attributed to the influence of feature noise. Also using prevalence normalized scores with in decoding step helps us achieve higher accuracy for the UMEC model.

4.7 Summary

In this chapter, we propose a novel method for unsupervised multi-class ensemble (UMEC) learning using ECOC matrix that solves a multi-class classification problem by encoding them into multiple binary problems. Our method addresses the issue of class prevalence prominent with methods dependent on ML estimations. We also propose to use *maximum* instead of *mean* as a statistic to reduce the multiple class scores into binary groups. Our simulation results show the strength of our model in handling different sources of noise prominent in classification problems. The study also affirms that using continuous scores output by the base classifiers is much more accurate than relying on discrete labels. Our research demonstrates that unsupervised multi-class classification problems can be solved efficiently with encoding methods like ECOC using continuous scores from multiple base classifiers. Our proposed method helps in determining the labels for the classification problem instead of

finding cluster means prevalent in the literature for learning product distributions. Further research could extend by generating unsupervised mutli-class classifier using a single objective function to infer multi-class labels using continuous scores.

4.8 Appendix A: Proposition Proof

The usual choice suggested for the reduction operator Θ is the mean (average) function. Instead, in Proposition 2, we propose the use of other order statistics for the reduction operator.

Proposition 2 *Suppose a classification process is such that the scores of each class $c \in \mathbf{C}$ are distributed according to the pdf $f^c(x)$. At a given instant k, the classifier predicts the class to be $\hat{c}$ such that $\hat{c} = argmax_{\{c \in \mathbf{C}\}}(f_k^c(x))$. Let t_k^{Pos}, t_k^{Neg} be the reduction (decomposition) statistics to decompose scores of sample k of a multi-class classifier into binary groups (of positive $\mathbf{C^{Pos}} \subset \mathbf{C}$ and negative classes $\mathbf{C^{Neg}} \subset \mathbf{C}$; $\mathbf{C^{Pos}} \cup \mathbf{C^{Neg}} \equiv \mathbf{C}$) for multiple comparisons. Then, a reduction (decomposition) statistic, which is a function of **maximums** of scores in each binary group ($t_k^{Pos} = g(max(x_k^{\mathbf{C^{Pos}}})), t_k^{Neg} = g(max(x_k^{\mathbf{C^{Neg}}}))$), performs better than any other reduction (decomposition) statistic. Note that the mean of binary groups is also a linear combination of **maximums** of each group.*

4.8.1 Appendix A.1: Proof

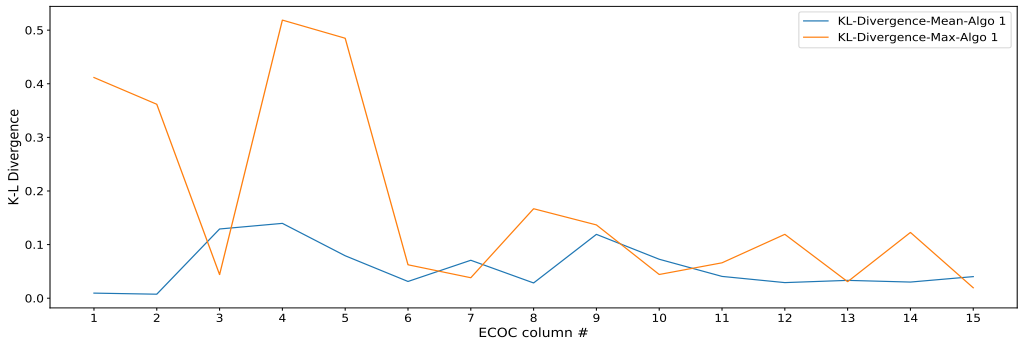
In order to prove the above proposition, we first revisit the structure of the base algorithms provided in Section 3.1 in the manuscript. Each base algorithm is designed in such a way that the class with the maximum score is the winner class for that base algorithm. Let us consider any relevant dichotomization/ binarization scheme applied over multi-class scores to encode them into binary class. Let c_{max} be the class with the maximum score in $\mathbf{C}$ multi-classes. Let $\mathbf{C^{Pos}}$, $\mathbf{C^{Neg}}$ denote the sets of classes contained positive and negative group of binary comparison formed after encoding multiple classes. Thus, $\mathbf{C^{Pos}} \cup \mathbf{C^{Neg}} \equiv \mathbf{C}$.

We want to determine reduction statistics t_k^{Pos}, t_k^{Neg} that could serve the best to represent the original multi-class classification problem in the encoded binary comparison.

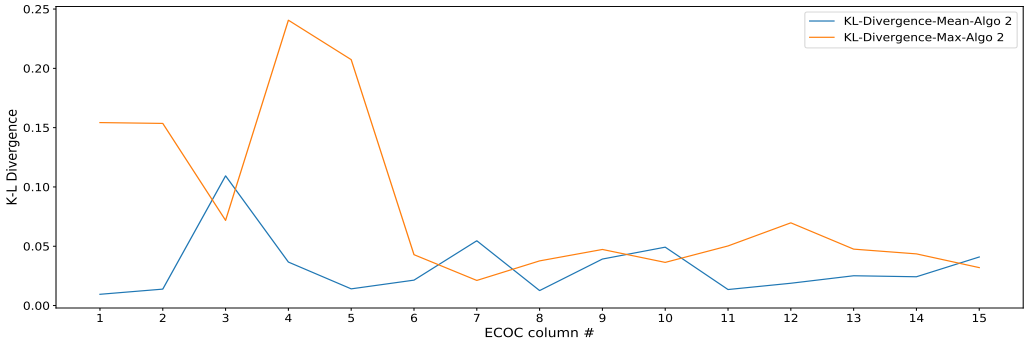
Let o_k^{Pos}, o_k^{Neg} be order statistics of scores in both sets $x_k^{\mathbf{C_k^{Pos}}}$, $x_k^{\mathbf{C_k^{Neg}}}$ other than *maximum*. The subscript k ensures that we are considering a single maintenance record. Let $c_k^0 = argmax(x_k^{\mathbf{C_k}})$ be the true failure mechanism class for k^{th} maintenance record. For efficient binary partitioning of the multi-class structure into binary classes, it is important that the representative score derived by using scores of either of the binary classes ($\mathbf{C_k^{Pos}}$, $\mathbf{C_k^{Neg}}$) is a function of the score of the true class c_k^0. If the score of the true class c_k^0 does not appear in the encoded scores of the binary classes then the following unsupervised classification will not be able to draw correct inference from the binary partitioning. Next, we consider the following two cases:

Let $c_k^0 \in \mathbf{C_k^{Pos}}$, thus the true class lies in the set of positive classes (as guided by the ECOC matrix). Now, any order statistic o_k^{Pos} other than the *maximum* cannot represent c_k^0 because by structure of the base algorithm, $c_k^0 = argmax_c(x_k^{\mathbf{C_k}})$, also $\mathbf{C_k^{Pos}} \subset \mathbf{C_k} \implies x_k^{\mathbf{C_k^{Pos}}} \subset x_k^{\mathbf{C_k}}$. Thus, if $c_k^0 \in \mathbf{C_k^{Pos}}$, then the score associated to class c_k^0 should be the maximum amongst the other scores in $\mathbf{C_k^{Pos}} \implies c_k^0 = argmax_c(x_k^{\mathbf{C_k^{Pos}}})$. For the second case, when $c_k^0 \in \mathbf{C_k^{Neg}}$, we can similarly infer that $c_k^0 = argmax_c(x_k^{\mathbf{C_k}}) = argmax_c(x_k^{\mathbf{C_k^{Neg}}})$ if $c_k^0 \in \mathbf{C_k^{Neg}}$ *and as* $\mathbf{C_k^{Neg}} \subset \mathbf{C_k}$ thereby implying that any order statistics o_k^{Pos} other than the *maximum* cannot represent c_k^0. Thus, for any representative score (reduction statistic) of the binary class, the true class scores $x_k^{c_k^0}$ can appear only when the reduction statistic is a function of the *maximum*.

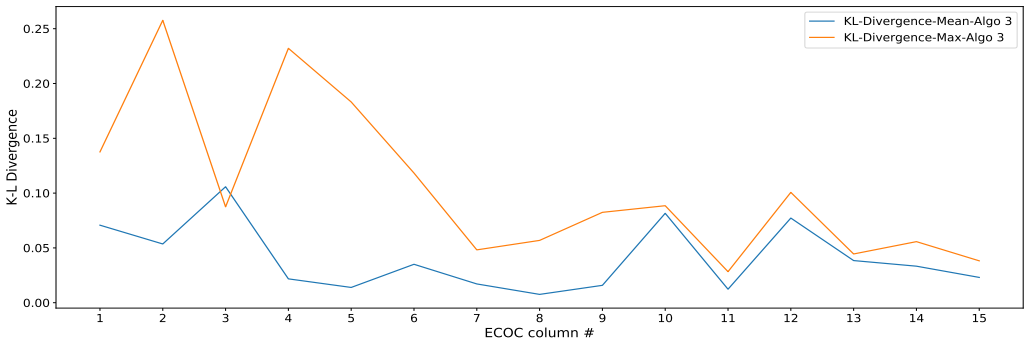
Further, we also want that the reduction statistics of the binary classes are such that the separation between the distribution formed by using the reduction statistics for both classes are well separated from each other. Thus, we further study the Kullback-Leibler divergence and the Jensen-Shannon divergence between the distributions of the reduction statistics of the positive class and the negative class. By comparing the divergence when the reduction statistics is *maximum* versus when it is *mean*. We find that a larger divergence is achieved using *maximum* in most of the cases as opposed to using *mean* in Figure 4.8 and 4.9.



(a) K-L Divergence for base algorithm 1.

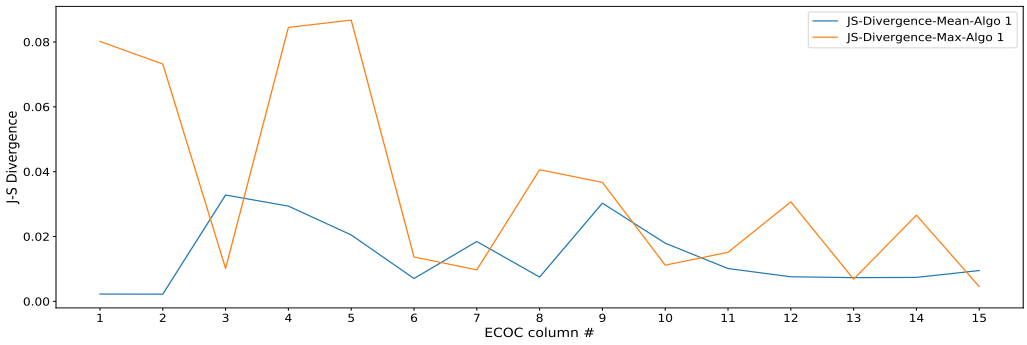


(b) K-L Divergence for base algorithm 2.

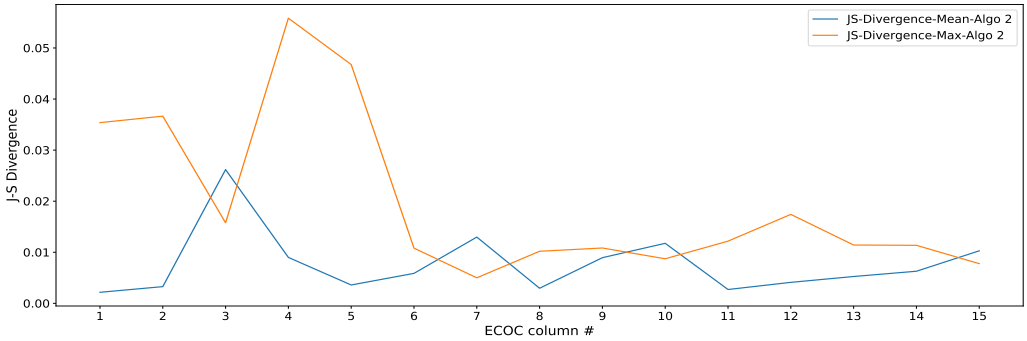


(c) K-L Divergence for base algorithm 3.

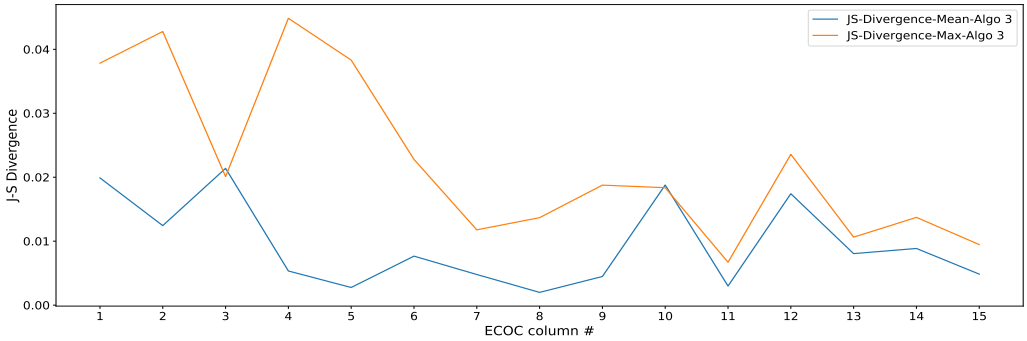
Figure 4.8: Comparison of K-L Divergence between positive and negative class scores for different base algorithms using *max* and *mean* reduction statistics.



(a) J-S Divergence for base algorithm 1.



(b) J-S Divergence for base algorithm 2.



(c) J-S Divergence for base algorithm 3.

Figure 4.9: Comparison of J-S Divergence between positive and negative class scores for different base algorithms using *max* and *mean* reduction statistics.

4.9 Appendix B: Iterative algorithms for rank 1 decomposition

Rank-1 decomposition of empirical covariance matrix $\hat{\Sigma}_2^u$ of rank statistics ($r_{i,k}^u \in \mathbf{R_i^u} \; \forall \; k \in \mathbf{K}$) is determined using Algorithm 7 to achieve the estimates of normalized accuracy ($\vec{\nu_i^u}$) for each base classifier $i \in \mathbf{M}$.

Algorithm 7: Rank-1 decomposition of second order moments using Singular Value Decomposition (SVD)

Input: empirical second order moments of ranks $\hat{\Sigma}_2^u$
Output: $\kappa_c^u, \; \vec{\nu^u}$

Let $\epsilon > 0, \; s = 1$, Initialize $\kappa^u(0) = \nu_o^u(0) = 0$
$\kappa^u(1), \; \nu_o^u(1) = \text{SVD}(\hat{\Sigma}_2^u)$
`/* where SVD represents singular value decomposition while` $\kappa^u(1), \; \vec{\nu_o^u}(1)$ `corresponds to the largest singular value and corresponding singular vector of` $\hat{\Sigma}_2^u$ `*/`
while $\kappa^u(s) - \kappa^u(s-1) > \epsilon$ **do**
    $\bigtriangledown(s) = \hat{\Sigma}_2^u - \text{diag}(\hat{\Sigma}_2^u) + \text{diag}(\kappa^u(s)\vec{\nu_o^u}(s)\vec{\nu_o^u}(s)^T)$
    $s = s + 1$
    $\kappa^u(s), \; \vec{\nu_o^u}(s) = \text{SVD}(\bigtriangledown(s))$
$\kappa_c^u = \kappa^u(s), \vec{\nu^u} = \vec{\nu_o^u}(s)$

Rank-1 decomposition of empirical covariance tensor $\hat{\Sigma}_3^u$ is determined using Algorithm 8 which helps to achieve the prevalence value ρ.

4.10 Appendix C: Simulation Parameters

4.10.1 For exponential distribution

For class scores generated from an exponential distribution, we set the scale parameter for each class of first base algorithm as 0.91, for second base algorithm as 0.83, and the third base algorithm as 1.11. We generate nine simulation sets by varying feature noise at three levels of {'FN-exp-1':$p_1 = 0.4$, 'FN-exp-2':$p_1 = 0.5$, 'FN-exp-3':$p_1 = 0.6$} and corresponding

Algorithm 8: Rank-1 decomposition of second order moments using Tensor Singular Value Decomposition (tSVD)

Input: empirical third order moments of ranks $\hat{\Sigma}_3^u$
Output: $\kappa_t^u,\ \vec{\alpha^u}$

Let $\epsilon > 0,\ s = 1$, Initialize $\kappa^u(0) = \alpha_o^u(0) = 0$
$\kappa^u(1),\ \alpha_o^u(1) = \text{tSVD}(\hat{\Sigma}_3^u)$
`/* where tSVD represents tensor singular value decomposition while` $\kappa^u(1),\ \vec{\alpha_o^u}(1)$ `corresponds to the largest singular value and corresponding singular tensor of` $\hat{\Sigma}_3^u$ `*/`
while $\kappa^u(s) - \kappa^u(s-1) > \epsilon$ **do**
  $\bigtriangledown(s) = \hat{\Sigma}_3^u - \text{diag}(\hat{\Sigma}_3^u) + \text{diag}(\kappa^u(s)\vec{\alpha_o^u}(s) \bigotimes \vec{\alpha_o^u}(s)^T)$
  $s = s + 1$
  $\kappa^u(s),\ \vec{\alpha_o^u}(s) = \text{tSVD}(\bigtriangledown(s))$
$\kappa_t^u = \kappa^u(s), \vec{\alpha^u} = \vec{\alpha_o^u}(s)$

to each feature noise level, three levels of mislabel noise are simulated {'CLN-exp-1':$p_2 = 0.2$, 'CLN-ecexp-2':$p_2 = 0.275$, 'CLN-exp-3':$p_2 = 0.35$}. The results for balanced and unbalanced cases for exponential distribution are shown in Figure 4.5.

4.10.2 For MVN distribution

For class scores generated from multivariate-normal distribution, we set the location parameter for each class of first base algorithm as 1.0, second base algorithm as 0.7, and third base algorithm as 0.3. The scale parameter for each algorithm is a 5×5 positive-semi-definite covariance matrix, which we randomly sample as:

$$S_{MVN_1} = \begin{pmatrix} 1.267 & 0.722 & 0.714 & 1.522 & 1.128 \\ 0.722 & 1.496 & 0.543 & 1.075 & 1.505 \\ 0.714 & 0.543 & 0.933 & 1.223 & 0.808 \\ 1.522 & 1.075 & 1.223 & 2.387 & 1.648 \\ 1.128 & 1.505 & 0.808 & 1.648 & 1.882 \end{pmatrix};$$

$$S_{MVN_2} = \begin{pmatrix} 2.070 & 1.570 & 2.371 & 1.204 & 2.011 \\ 1.570 & 2.127 & 2.164 & 1.479 & 1.688 \\ 2.371 & 2.164 & 3.385 & 1.938 & 2.591 \\ 1.204 & 1.479 & 1.938 & 1.594 & 1.206 \\ 2.011 & 1.688 & 2.591 & 1.206 & 2.251 \end{pmatrix}$$

and $S_{MVN_3} = \begin{pmatrix} 0.531 & 0.834 & 1.085 & 0.633 & 0.696 \\ 0.834 & 1.802 & 1.809 & 1.410 & 1.064 \\ 1.085 & 1.809 & 2.398 & 1.446 & 1.525 \\ 0.633 & 1.410 & 1.446 & 1.236 & 0.923 \\ 0.696 & 1.064 & 1.525 & 0.923 & 1.070 \end{pmatrix}$.

For balanced dataset, we generate nine simulation sets by varying feature noise at three levels of {'FN-MVN-1':$p_1 = 0.1$, 'FN-MVN-2':$p_1 = 0.185$, 'FN-MVN-3':$p_1 = 0.25$} and corresponding to each feature noise level, three levels of mislabel noise are simulated {'CLN-MVN-1':$p_2 = 0.2$, 'CLN-MVN-2':$p_2 = 0.275$, 'CLN-MVN-3':$p_2 = 0.35$}. For unbalanced dataset we generate 12 simulation sets by varying feature noise at three levels of {'FN-MVN-1':$p_1 = 0.1$, 'FN-MVN-2':$p_1 = 0.185$, 'FN-MVN-3':$p_1 = 0.25$, 'FN-MVN-4':$p_1 = 0.315$} and corresponding to each feature noise level, three levels of mislabel noise are simulated {'CLN-MVN-1':$p_2 = 0.2$, 'CLN-MVN-2':$p_2 = 0.275$, 'CLN-MVN-3':$p_2 = 0.35$}. The results for balanced and unbalanced case for multivariate-Gaussian distribution are shown in Figure 4.6.

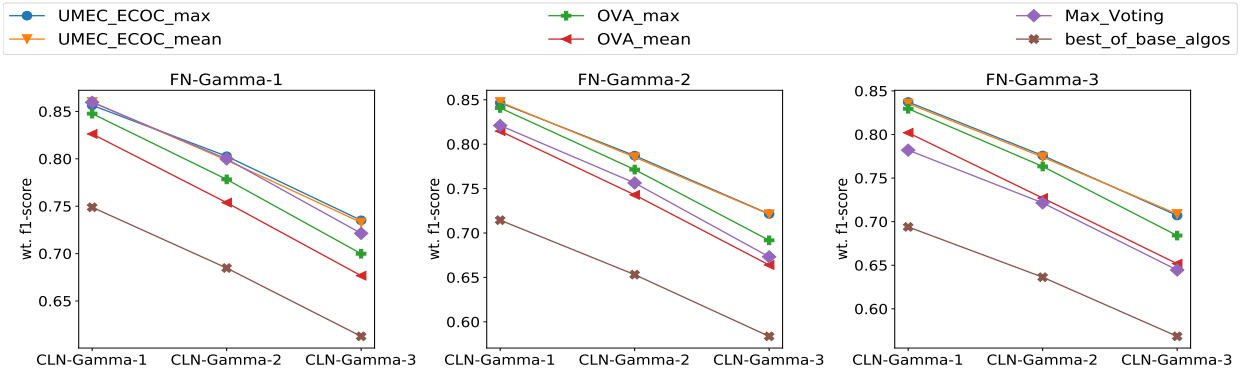
4.10.3 For Gamma distribution

For class scores generated using Gamma distribution, we set the shape parameter for each class of first base algorithm as 1.0, second base algorithm as 0.7, and third base algorithm as 0.3. The scale parameter for each of the five classes in the three algorithms are

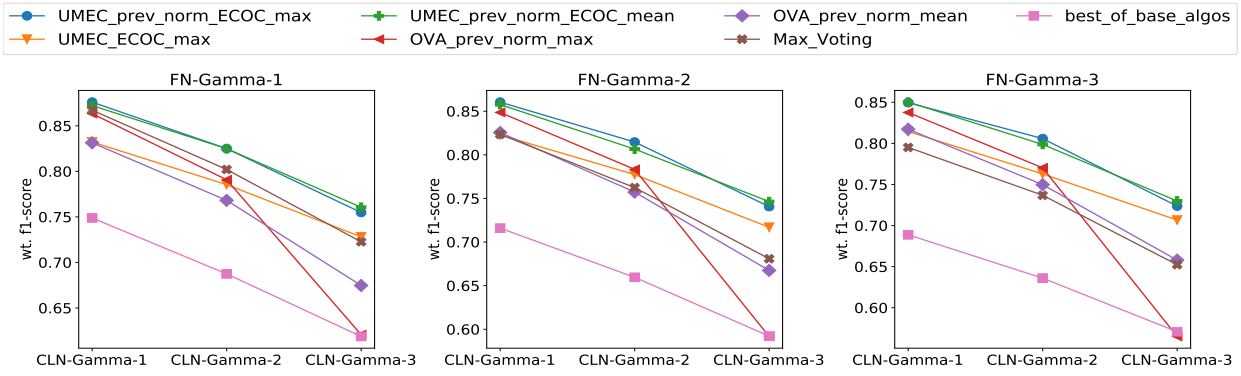
$S_{Gamma_1} = \begin{pmatrix} 0.549 & 0.715 & 0.603 & 0.545 & 0.424 \end{pmatrix}$;

$S_{Gamma_2} = \begin{pmatrix} 0.872 & 0.052 & 1.099 & 0.871 & 0.841 \end{pmatrix}$ and

$S_{Gamma_3} = \begin{pmatrix} 1.826 & 2.062 & 1.436 & 1.766 & 2.339 \end{pmatrix}$. Similar to the Gaussian case, we generate nine simulation sets by varying the feature noise at three levels of {'FN-Gamma-1':$p_1 = 0.1$, 'FN-Gamma-2':$p_1 = 0.185$, 'FN-Gamma-3':$p_1 = 0.25$} and corresponding to each feature noise level, three levels of mislabel noise are simulated {'CLN-Gamma-1':$p_2 = 0.2$, 'CLN-Gamma-2':$p_2 = 0.275$, 'CLN-Gamma-3':$p_2 = 0.35$}. The results for balanced and unbalanced case for multivariate-Gaussian distribution are shown in Figure 10 of the manuscript. The simulation results for Gamma distribution is shown in Figure 4.10. The results when class scores follow Gamma distributions demonstrate that using *max* or *mean* as reduction statistics do not generate significant difference in performance.



(a) Simulation results for Gamma distributed class scores (balanced case).



(b) Simulation results for Gamma distributed class scores for (unbalanced case)

Figure 4.10: Simulation results for Gamma distributed class scores

4.10.4 For prevalence dependent MVN distribution

The location parameters for simulating unbalanced data set with prevalence dependent on location parameters of each class for the three base algorithms are given by:

$\mu_{MVN_1} = \begin{pmatrix} 0.857 & 0.898 & 1.966 & 0.503 & 0.776 \end{pmatrix}$;

$\mu_{MVN_2} = \begin{pmatrix} 0.600 & 0.629 & 1.376 & 0.352 & 0.543 \end{pmatrix}$

and $\mu_{MVN_3} = \begin{pmatrix} 0.257 & 0.269 & 0.590 & 0.151 & 0.233 \end{pmatrix}$.

The scale parameter for each algorithm is a 5×5 positive-semi-definite covariance matrix, which we randomly sample as

$$S_{MVN_1} = \begin{pmatrix} 1.267 & 0.722 & 0.714 & 1.522 & 1.128 \\ 0.722 & 1.496 & 0.543 & 1.075 & 1.505 \\ 0.714 & 0.543 & 0.933 & 1.223 & 0.808 \\ 1.522 & 1.075 & 1.223 & 2.387 & 1.648 \\ 1.128 & 1.505 & 0.808 & 1.648 & 1.882 \end{pmatrix}$$

$$S_{MVN_2} = \begin{pmatrix} 2.070 & 1.570 & 2.371 & 1.204 & 2.011 \\ 1.570 & 2.127 & 2.164 & 1.479 & 1.688 \\ 2.371 & 2.164 & 3.385 & 1.938 & 2.591 \\ 1.204 & 1.479 & 1.938 & 1.594 & 1.206 \\ 2.011 & 1.688 & 2.591 & 1.206 & 2.251 \end{pmatrix}$$

and

$$S_{MVN_3} = \begin{pmatrix} 0.531 & 0.834 & 1.085 & 0.633 & 0.696 \\ 0.834 & 1.802 & 1.809 & 1.410 & 1.064 \\ 1.085 & 1.809 & 2.398 & 1.446 & 1.525 \\ 0.633 & 1.410 & 1.446 & 1.236 & 0.923 \\ 0.696 & 1.064 & 1.525 & 0.923 & 1.070 \end{pmatrix}.$$

We generate twelve simulation sets by varying feature noise at three levels of {'FN-MVN-1':$p_1 = 0.1$, 'FN-MVN-2':$p_1 = 0.185$, 'FN-MVN-3':$p_1 = 0.25$, 'FN-MVN-4':$p_1 = 0.315$} and corresponding to each feature noise level, three levels of mislabel noise are simulated {'CLN-MVN-1':$p_2 = 0.2$, 'CLN-MVN-2':$p_2 =$

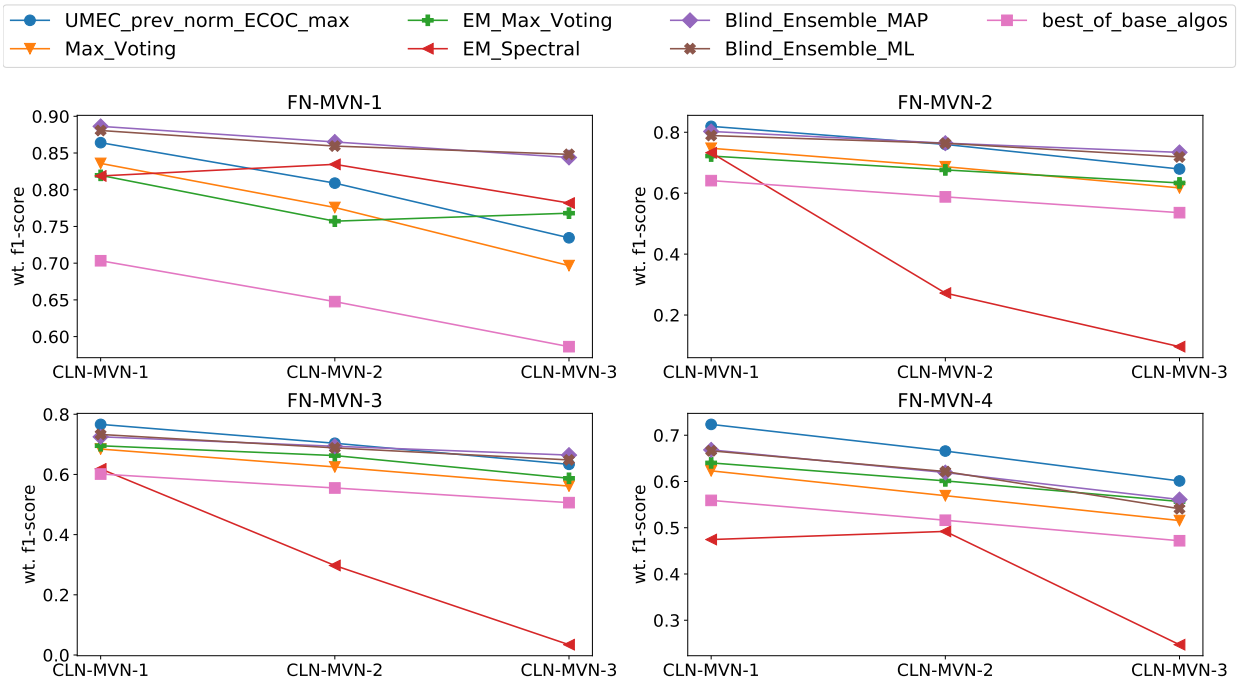


Figure 4.11: Simulation results for MVN with prevalence dependent location parameters

0.275, 'CLN-MVN-3':$p_2 = 0.35$}. The results for un-balanced case of MVN with prevalence depending on location parameter are shown in Figure 4.11. In this figure we include other state-of-the-art models from [160] (referred to as Blind_Ensemble_ML, Blind_Ensemble_MAP) and from [189] (referred as EM_Spectral and EM_Max_Voting). We get similar results as in Figue 9b of manuscript where it can be seen that the state-of-the-art methods perform better than our proposed method when the feature noise level is low, however, as the feature noise level increases our proposed method beats the state-of-the-art models. Since real-world maintenance records are commonly subject to a high level of feature noise, the proposed UMEC method is anticipated to be more accurate in practice relative to existing methods.

Chapter 5

Identifying equipment health status from maintenance records using Lexicon based Unsupervised Sentiment Analysis Adjusted for Negation (LUSAA-N)

5.1 Focused Abstract

During periodic inspection and repair of industrial equipment, technicians create maintenance records in the form of unstructured text to document valuable information regarding the equipment condition, operating environment, failed components, failure mechanism, etc. Inferring the equipment health status from these records can help in planning future maintenance actions to avoid expensive machine failures. However, automatic inference of the equipment health status is non-trivial due to various challenges inherent in analyzing unstructured text such as domain-specific vocabulary and negation words. Further, manual labeling of the records for supervised classification is arduous and impractical. Thus, in this chapter, we present a model to automatically extract the equipment health status from main-

tenance records in a completely unsupervised manner. We show that using general English sentiment lexicons is highly inefficient for analyzing maintenance records, and thus propose a method that leverages sentiment lexicons curated for the maintenance domain and addresses the central issue of negation. We demonstrate the effectiveness of our proposed model over extant research using data from real-life maintenance records from oil rig equipment.

5.2 Introduction

Effective maintenance of equipment is critical to ensure industrial productivity and safety. Equipment breakdown often results in a high loss of production time, which could be costly depending on the availability of appropriate inventory/mitigating measures. Technicians usually schedule periodic inspections to check equipment functionality to avoid such untimely interruptions. Apart from this, a thorough inspection is conducted if equipment malfunctions, and consequently, appropriate maintenance actions are taken. The observations of technicians during such inspections are documented in maintenance records in the form of unstructured free text. The maintenance records thus consist of a wealth of information about equipment health status, its operating environment and control limits, its vulnerable parts/components, their failure mechanisms, etc. Thus, valuable inference about equipment health status could be retrieved from such unstructured maintenance records. Figure 5.1 shows examples of different maintenance records collected either during periodic inspections or breakdown events and the technicians' sentiments about equipment's health status. Efficiently extracting the sentiment regarding equipment health status could help develop better reliability models and schedule better spare part inventory management. For example, [11] introduces the framework of vectorizing an annotated maintenance record into numerical values representing costs, sentiment and association where maintenance records are annotation done by operators.

Though maintenance records consist wealth of information about industrial equipment,

Maintenance Record	Sentiment \ Equipment Health
Observed unusual noise coming from fluid end. Isolated pump and inspected fluid end.. MP#1 inspection. Breakdown Job	Negative (Equipment in poor health state)
the impeller has been arrived and received from the store and is installed on mud pump #1 check the notes of Job # 165253 this job can be closed ----------	Positive (Equipment is repaired and is as good as new)
Visually inspected the BX titan valves for general condition. Inspected the exterior of valve for mud leakage. Found ok. Checked the pneumatic supply pressure setted to 115 psi. Checked the air hose connections and the pneumatic line connection for tightness and leaks. Found ok..	Neutral (Equipment is been running and seems to have no problem)

Figure 5.1: Illustration of maintenance records and sentiment about equipment health status

automatically inferring the sentiment regarding equipment health status is not an easy task. Further, sentiment analysis is a significant area in the natural language processing (NLP) literature [188]. However, traditionally sentiment classifiers are trained using sentiment labels available either as annotations, ratings, or derived from emoticons. However, no such training data is available for industrial use, and manually annotating such unstructured maintenance records takes a tremendous manual effort. Recent development in transfer learning models leveraging pre-trained language models is highly resource-intensive [40] for training. A cheap alternative that is usually advised is to fine-tune the pre-trained models like BERT [166] using out of domain data as these models are supposed to be highly generalized for domain adaptation. However, this alternative suffers serious inaccuracies when domain data is highly customized, as shown in section 5.5.

Unsupervised sentiment analysis [50] helps to overcome these challenges and provides a resource-efficient alternative to model equipment health status from maintenance records. These methods rely on sentiment *lexicons*, which are words/phrases that express sentiments and are usually assigned a polarity score for different sentiments. General purpose lexicon lists such as SentiWordNet [13] are freely available for applications. [57] propose to use (Twitter Specific Lexicon Set) TSLS that maps lexicons to seven aspects suitable for twitter sentiment analysis which are scored by manual annotators. However, such resources are of rare use for industrial maintenance records. As can be seen in Figure 5.2, if the word *replaced* is used as an infinitive verb, it indicates that some component within the equipment has failed

and needs to be replaced, thus indicating that equipment is in poor health. However, when the word *replaced* is used as a past tense verb, it means that the action of replacing a bad component has been completed, and now the system is back to good health. Thus, the existing English lexicons sets are insufficient to conduct unsupervised sentiment analysis in the industrial maintenance domain.

Maintenance Record	Sentiments/ Health status
Parts Requested from ICS:. Rupture disc on MP4 is ***to be replaced***. Breakdown Job	Negative (Equipment in poor health state)
Replaced module with South west discharge Module. All componets on module showed ecsessive corrosion, ***replaced*** Discharge and installed new cap screws for discharge spacer.----------	Positive (Equipment is repaired and is as good as new)

Figure 5.2: Illustration of maintenance records with domain specific lexicons

Apart from domain-specific vocabulary, the text data also present other important challenges like the presence of *negation* words. Handling negation has always been an important task for effective sentiment analysis [177] as the presence of a negation word could completely alter the sentiment indicated by a lexicon. For example, consider the maintenance records shown in Figure 5.3. The presence of words or phrases like *leak*, *corrosion*, *abnormal vibration* indicate that the machine is in poor state and may soon need repair. However, due to the presence of the 'negation cue' word 'no,' the sentiment completely changes. This change in sentiment is due to the presence of words like *leak* inside the scope of the negation cue word. The traditional lexicon-based sentiment analysis methods rely only on the count of the different sentiment lexicons and assume equal predictivity of each lexicon while evaluating sentiment. [46] demonstrates that sentiment lexicons are not equally predictive. The predictiveness of lexicons depends on the co-occurrence of lexicons of opposite polarity. However, [46] do not tackle the important issue of negation.

To address the above mentioned challenges we propose a new model called Lexicon based Unsupervised Sentiment Analysis Adjusted for Negation ($LUSAA\text{-}N$). Essentially the highlights of this work are:

1. We manually extract and provide novel seed lexicons to identify operators sentiment

Maintenance Record
Checked all fluid end *no leaks*. Was performed visually inspect suction and discharge manifolds *no corrosion* cracks washouts damage .Checked discharge and modules *no founded* any movement.. SPM01-FEND-Checks.
1 the motor was in good condition 2 there was *no abnormal noise* 3 *no abnormal vibration* 5 was working proper 6 the motor blower was in good condition 7 was cleaned.

Figure 5.3: Illustration of maintenance records with negation words

towards industrial equipment health status as described in maintenance records

2. We provide a novel framework to bootstrap sentiment lexicons using raw maintenance records and seed lexicons by fine tuning pre-trained BERT model.

3. Our model formulation uses information about words in negation scope to optimize sentiment lexicon predictivity while generating an overall sentiment score of the maintenance record.

4. The proposed model does not require any external lexical resources or intensive computing resources and could be easily trained on CPU machines rather than GPU or TPU.

This chapter is organized as follows. Section 5.3 provides a literature review of unsupervised sentiment analysis methods and addresses their limitations. Section 5.4 outlines the proposed framework and details the *LUSAA-N* model. In Section 5.5 we discuss the application of the proposed model on real industrial data from oil rigs and compare our method with existing models. Finally, section 5.6 provides the summary of the work and discusses the future research areas for leveraging sentiment analysis in the maintenance domain.

5.3 Literature Review

Identifying anomalies is an essential task of reliability engineering. [181] cluster quality related text data to provide optimal solutions for given quality/repair problems using *problem cluster-solution toolbox* model. Apart from this, application of NLP methods in classifying

console/system logs (Syslogs) has been studied extensively [63] to aid reliability engineering of software. However, these methods are either based on the assumption that different types of console logs (normal and anomalous) have fixed templates [45], [103] that differ from each other or require some training data for the downstream task of classification [102], [191]. Apart from the software domain, authors in [48] propose to identify hazardous events on construction sites by training NLP models like BERT on labeled safety reports. However, maintenance records generated while inspecting or during a breakdown event do not follow any specific template, and neither have training data available. Hence, identifying the health status of equipment is entirely unsupervised and presents challenges of high variability in text.

In unsupervised sentiment analysis, researchers utilize three major models to model data lacking sentiment labels. The first model extracts latent topics that are assumed to generate words in the text using Latent Dirichlet Allocation (LDA) [21]. [86] propose to use LDA for the joint topic (aspect) and sentiment modeling and show that when there is a single aspect, the models reduce to latent sentiment analysis model (LSA). [73] deploy LDA to classify case studies in construction projects. The second modeling approach is based on transfer learning where sentiment labels are not available in the target domain, and labeled source domain data is used. The model tries to minimize the distance between target and source domain data while predicting labels for source data. Pre-trained language models like BERT are used extensively for the same [179]. However, they are heavily dependent on expensive computing resources like GPU. The third type of model makes use of pre-existing sentiment lexicon and infers sentence sentiment based on lexicon polarity [42], [157]. [57] propose to generate transferable features by mapping manually scored sentiment lexicons to seven aspects that can be used by YAC2 clustering algorithm. However, the transferable domains of application for which lexicons are generated are of similar nature (for example analyzing sentiments in tweets for Starbucks versus Verizon). A pre-trained lexicon-based sentiment analysis model *VADER* is proposed by [68] which makes use of publicly available SentiWordnet lexicons

[13]. However, as discussed in section 5.2 such general purpose lexicons are not suitable for industrial maintenance sentiment analysis. Table 5.1 summarizes various semi-supervised and unsupervised methods used for sentiment analysis.

For effective application of lexicon-based unsupervised sentiment analysis, the two most important considerations that need to be taken care of are the domain relevance of sentiment lexicons and the influence of negation cues. Sentiment lexicons originated in [152], following which creating or extending lexicons received significant interest [163]. [61] review various methods to expand domain-specific sentiment lexicons and propose a weighted lexical graph generated by domain-specific word embeddings to propagate sentiment labels. However, these methods depend on the co-occurrence of sentiment lexicons for expansion but as shown in section 5.2 lexicons in maintenance records may not necessarily co-occur and also have additional dependencies on the grammar/tense of the sentence. [120] propose to use part of speech (POS) tags to extend sentiment lexicons. However, their method places hard constraints on POS tags of sentiment words and works for unigrams. As we have the labels of seed lexicons, word embeddings trained on domain data and having attention from POS tags can help bootstrap other lexicons from the corpus. [174] propose embeddings with POS guided attention. However, models like BERT [166] not only summarize POS attention but also take the context of words into consideration while generating efficient contextualized embeddings. Hence, we propose a framework to bootstrap sentiment lexicons by fine-tuning BERT on seed lexicons.

The second challenge of handling negation is to identify the scope of the negation cue (as shown in Figure 5.3) and to check if the sentiment lexicon belongs to the scope. Negation scope detection is an extensively studied topic. [72] provide an overview of several labeled corpus to aid negation detection. [115] proposed deep semantic parsing to extract negation scope on Linux systems, which met with application issues while replicating on a Windows Subsystem for Linux. Authors in [36] used the conditional random field method to predict negation scope, while [194] used the word-topic graph model to identify the same. As

Table 5.1: Comparison of different research streams for Sentiment Analysis

Classification Types	**Model Types**	**References**	**Limitation**
Semi-Supervised	Template Based Methods	[45], [103]	Complete Unstructured Data
	Partial Labeled Data	[102], [191], [48]	No Labeled Data
Unsupervised	Topic Models	[86], [73]	1) Does not handle negation; 2) Existing Domain Lexicons not suitable for Maintenance
	Transfer Learning Model	[179]	
	Lexicon Based Model	[42], [157], [57], [68], [163], [46]	

methods to identify negation scope is not a novel contribution of this chapter, we use an open-source model termed Negtool [47] to identify negation scope in this research work.

The bootstrapped domain-specific lexicons, identified negation scopes, and maintenance records become inputs to our proposed *LUSAA-N* model. The *LUSAA-N* uses these inputs to create discriminatory scores between each sentiment class, providing us with the final sentiment class labels as described in section 5.4.

5.4 Proposed framework and *LUSAA-N* model

This section presents our framework (see Figure 5.4) to extract equipment health status from maintenance records. We begin by manually generating seed sentiment lexicons for industrial equipment health status analysis. The generated seed lexicon set is used with raw maintenance records to bootstrap sentiment lexicons, as demonstrated in section 5.4.1. We then replaced the bootstrapped lexicons in the raw maintenance records to generate processed maintenance records, as shown in Figure 5.5. We use the processed maintenance records to extract the negation scope as explained in section 5.4.2. The sentiment lexicons, the processed maintenance records, and the extracted negation scope become the inputs to the *LUSAA-N* model as explained in section 5.4.2. The scores output by the *LUSAA-N*

model is then classified to generate sentiment labels for the equipment health status as shown in section 5.4.3.

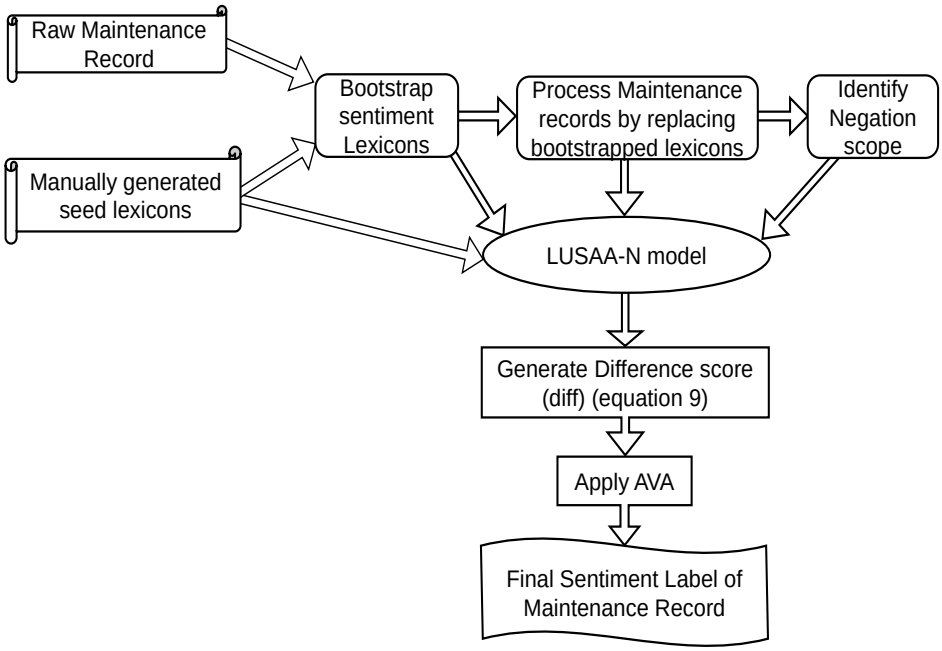


Figure 5.4: Framework for extracting equipment health status from maintenance records

5.4.1 Bootstrapping domain specific lexicon

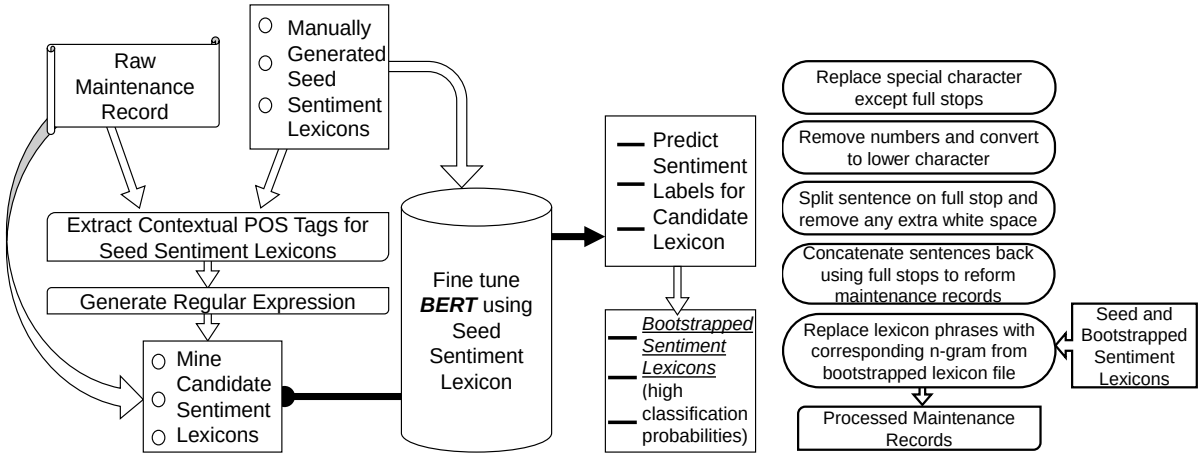
Raw maintenance records are analyzed to generate a seed lexicon set indicating equipment health status. This work considers that equipment could lie in three health states. The first is the *poor or degraded* state, which in sentiment analysis terms would be a 'negative state.' We generate the seed lexicons for the negative state. These lexicons represent a problem or a failure mechanism in the equipment, for example, terms and phrases like 'leak,' 'eroded,' 'needs repair,' 'to be replaced,' etc. The second state that we consider is the *working fine* or 'neutral' state. The equipment is said to be in a 'neutral' state if it is observed that everything is working as expected during the inspection. Examples of seed lexicon for neutral states would include 'all ok,' 'under control,' 'looks good,' etc. The third state that is the *as good as new* state or 'positive' state. After a repair action is taken on the equipment, we assume that it is back in its original state and would now function at its best capability. The seed

lexicons for the positive state would include terms like 'was repaired,' 'ready to use, 'is fixed,' 'as good as new,' etc. The complete list of seed sentiment lexicons for industrial sentiment analysis is provided in Appendix 5.9.

After generating seed lexicons, we extract the contextual POS tags for each sentiment lexicon using 'nltk' library in python. The context is provided as per the use of seed lexicons in maintenance records. The extracted contextual POS tags help to form a regular expression pattern. We use the RegEx pattern to mine words or phrases from maintenance records that form the candidates of being sentiment lexicons. As fine-tuning in BERT [166] needs limited data and is assumed to provide decent accuracy for text classification, we use the base-BERT model and fine-tune it with seed lexicons to classify candidate lexicons. The seed lexicons provide training examples to fine-tune BERT. As BERT is assumed to provide high contextual embeddings, the information about POS tags and other semantic contexts is already incorporated into the base BERT model. The extracted candidate phrases are classified into three sentiment classes using fine-tuned BERT. The complete workflow to classify new sentiment lexicons is illustrated in Figure 5.5a. As BERT outputs a soft-max score corresponding to each class for a given lexicon, we select only lexicons having a soft-max score corresponding to any sentiment class larger than a chosen threshold for that sentiment class as the bootstrapped sentiment lexicon. Finally, we process the raw maintenance records with steps shown in Figure 5.5b where we replace the seed and bootstrapped sentiment lexicon phrases with their appropriate n-grams in the raw maintenance records.

5.4.2 Extracting Negation scope & Lexicon based Unsupervised Sentiment Analysis Adjusted for Negation (*LUSAA-N*) model

We use the processed maintenance records to extract words that lie in the negation scope of a given negation 'cue' using the procedure shown in Figure 5.6. We use an open-source tool called Negtool proposed by [47]. Specifically for negation scope extraction, we first convert the processed maintenance records to CONLL-X format using [26]. For converting raw



(a) Workflow to Bootstrap Sentiment Lexicons from Seed Sentiment Lexicons

(b) Processing of maintenance records with Bootstraped Sentiment Lexicons

Figure 5.5: Bootstrap domain specific sentiment lexicon and process maintenance records

maintenance records into CONLL-X format, we use the Core-NLP toolkit in python proposed by [96]. After having the CONLL-X format for the maintenance records, we predict the negation cue using SVM-based binary classifiers. We then predict the negation scope using the maximum-margin CRF model for each predicted negation cue. We use the frequency of sentiment lexicons present in the extracted negation scope to model negation in the proposed *LUSAA-N* model as shown next.

Figure 5.6: Extracting Negation scope from processed maintenance records

The proposed *LUSAA-N* model takes the count vector of words present in the processed maintenance records (indicated as x) along with the count vector of words present in the negation scope (indicated as $\tilde{x}$) as inputs. We assume the generative process for the document to follow the Dirichlet Compound Multinomial (DCM) model proposed by [94]. The DCM model is a hierarchical model where a sample is first drawn from a Dirichlet distribution to get parameters of the Multinomial distribution. Words are drawn for the document based on the Multinomial distribution. To handle negation, the *LUSAA-N* model assumes words

in negation scope to follow the same Multinomial distribution as directed by the Dirichlet sample. However, as these words are negated to diminish their effect, their count $\tilde{x}$ is subtracted from the original count vector x. The generation model for the words (w) having a frequency of x_w in a given maintenance record and a frequency of $\tilde{x_w}$ in the negation scope of the maintenance record in equation 5.1. In equation 5.1 ξ denotes multinomial probabilities and α_w denotes Dirichlet parameter with$w$ components.

$$
\begin{aligned}
p(x,\tilde{x}|\alpha_w) = & \int_{\xi} P_{multinomial}(x,\tilde{x}|\xi) P_{Dirichlet}(\xi|\alpha_w) d\xi \\
= & \int_{\xi} \frac{n!}{\prod_{w=1}^{W}(x_w - \tilde{x_w})!} \left(\prod_{w=1}^{W} \xi^{x_w - \tilde{x_w}} \right) \frac{\Gamma\left(\sum_{w=1}^{W} \alpha_w\right)}{\prod_{w=1}^{W} \Gamma(\alpha_w)} \prod_{w=1}^{W} \xi_w^{\alpha_w - 1} d\xi \\
= & \frac{n!}{\prod_{w=1}^{W}(x_w - \tilde{x_w})!} \frac{\Gamma\left(\sum_{w=1}^{W} \alpha_w\right)}{\prod_{w=1}^{W} \Gamma(\alpha_w)} \int_{\xi} \prod_{w=1}^{W} \xi_w^{x_w - \tilde{x_w} + \alpha_w - 1} d\xi \\
p(x,\tilde{x}|\alpha_w) = & \frac{n!}{\prod_{w=1}^{W}(x_w - \tilde{x_w})!} \frac{\Gamma\left(\sum_{w=1}^{W} \alpha_w\right)}{\Gamma\left(\sum_{w=1}^{W}(x_w - \tilde{x_w}) + \alpha_w\right)} \prod_{w=1}^{W} \frac{\Gamma(x_w - \tilde{x_w} + \alpha_w)}{\Gamma \alpha_w}
\end{aligned}
\tag{5.1}
$$

The classifier rule used by the *LUSAA-N* model is the binary Bayes classifier rule. The novelty of the *LUSAA-N* model is that it also takes the influence of negated words while generating the classifier score. Although the sentiment classification of industrial maintenance records is ternary, we choose the classification rule to be binary and then apply ternary to binary decomposition technique as demonstrated in section 5.4.3. The binary classifier rule is chosen because of its simple interpretation and computational benefits, as explained below. To generate classifier using DCM model the Dirichlet parameter α_w is assumed to be dependent on sentiment label $s \in \mathbf{S}$ where $\mathbf{S} \ni \{s, \neg s\}$. Equation 5.2 shows the likelihood of a given maintenance record under the influence of label s. Here $K(x)$ is a constant term that counts the frequency of words (without any influence on their predictive ability). In 5.7 we show that the classifier decision rule is independent of $K(x)$, N_t is the length of the complete maintenance record, and N_{ns} is the length of the negation scope. Γ denotes the

gamma function, where $\Gamma(x) = x!$.

$$p(x, \tilde{x}|\alpha_{w,s}) = \frac{n!}{K(x)} \frac{\Gamma\left(\sum_{w=1}^{W} \alpha_{w,s}\right)}{\Gamma\left(N_t - N_{ns} + \sum_{w=1}^{W} \alpha_{w,s}\right)} \prod_{w=1}^{W} \frac{\Gamma(x_w - \tilde{x_w} + \alpha_{w,s})}{\Gamma \alpha_{w,s}} \tag{5.2}$$

Let $P(W = w|S = s) = \beta_{w,s}$ denote the probability that a single word token w belongs to a class of sentiment s. Following [46], we assume that any word would have a baseline ν_w. If a given word w is a lexicon for a sentiment s, then its predictiveness parameter is $1 + \gamma_w$ if the document have sentiment s otherwise if the document has sentiment label $\neg s$ then its predictiveness parameter is $1 - \gamma_w$. Using this, the probability ($\beta_{w,s}$) of word w depending on whether its a lexicon of a given sentiment class s of a maintenance record can be written as shown in equation 5.3. The binary classes are assumed to have equal prior probabilities. The Bayes classifier decision rule in terms of likelihood is given by $\log P(x, \tilde{x}|S = s) \gtrless \log P(x, \tilde{x}|S = \neg s)$ in favor of sentiment class $s(>)$ or $\neg s(<)$ respectively. The Dirichlet parameter ($\alpha_{w,s}$) is assumed to be dependent on the probability of word = $\beta_{w,s}$ by the relation $\alpha_{w,s} = \delta\beta_{w,s}$, where δ is the concentration parameter for the DCM distribution. As δ grows, the prior on multinomial probabilities ξ gets tightly linked to β_w, thus reducing the DCM model to the multinomial model.

$$\beta_{w,s} = \begin{cases} (1 + \gamma_w)\nu_w, & \text{if } \textit{document sentiment label} = s \textit{ and } w \textit{ is a lexicon of sentiment } s \\ (1 - \gamma_w)\nu_w, & \text{if } \textit{document sentiment label} = s \textit{ but } w \textit{ is a lexicon of sentiment } \neg s \\ \nu_w, & \text{if } \textit{document sentiment label} = s \textit{ but } w \textit{ is not a lexicon} \end{cases} \tag{5.3}$$

The final decision rule is derived in 5.7 using equation 5.2, 5.3 and is given by equation 5.4. The predictiveness parameter γ_w for each lexicon is derived by penalizing the lexicons that co-occur with lexicons of opposite sentiments after adjusting for negation. The co-occurrence count of lexicon w_s with all opposite lexicons $w_{\neg s}$ adjusted for their presence in negation scope

is given in equation 5.5. For a given maintenance record, we derive the expected product of count of word pairs $(w_s, w_{\neg s})$ adjusted for negation to be given by equation 5.6 as derived in 5.8. Here, N_t and N_{ns} have the same meaning as defined earlier. Even after adjusting for negation, we find that the expected product of count of word pairs $(w_s, w_{\neg s})$ is proportional to the product of probabilities $\beta_{w_s} \times \beta_{w_{\neg s}}$ (where we omit the first index to avoid redundancy of notation). The proportional dependence on product of probabilities $\beta_{w_s} \times \beta_{w_{\neg s}}$ is a similar finding as [46], however, the proportionality constant is now dependent not only on the length of maintenance record (N_t) but also on the length of negation scope (N_s) after adjusting for negation. To optimize the predictiveness parameters $\gamma_{w,s},\ \gamma_{w,\neg s}$, we minimize the squared error between the expected values of counts $\mathbb{E}[co - counts_{w,s}]$ (equation 5.7) and their true value (equation 5.5) as given in equation 5.8. We use an alternating direction method of multipliers (ADMM) optimization scheme for binary blocks to learn the predictiveness parameter $\gamma_{w,s},\ \gamma_{w,\neg s}$ for words w_s and $w_{\neg s}$ [1]. Apart from the ease of calculation and inference, another reason to rely on binary optimization is that convergence of ADMM for simple extensions of binary-block to a multi-block convex minimization problem is not easily guaranteed [29]. After optimizing the predictiveness parameter, the difference of scores for each maintenance record (equation 5.9) is calculated using the Bayes classifier decision rule of equation 5.4. The difference score is generated for all binary comparisons as dictated by the ternary classifications scheme (shown in section 5.4.3) which are finally used to generate class labels as explained in the next section.

$$\begin{aligned} &\sum_{w \in s} \log \left(\frac{\Gamma(x_w - \tilde{x_w} + \delta(1+\gamma_w)\nu_w)\Gamma(\delta(1-\gamma_w)\nu_w)}{\Gamma(\delta(1+\gamma_w)\nu_w)\Gamma(x_w - \tilde{x_w} + \delta(1-\gamma_w)\nu_w)} \right) \\ \gtrless &\sum_{w \in \neg s} \log \left(\frac{\Gamma(x_w - \tilde{x_w} + \delta(1+\gamma_w)\nu_w)\Gamma(\delta(1-\gamma_w)\nu_w)}{\Gamma(\delta(1+\gamma_w)\nu_w)\Gamma(x_w - \tilde{x_w} + \delta(1-\gamma_w)\nu_w)} \right) \end{aligned} \tag{5.4}$$

[1] We use an optimization scheme which is similar in application to [46]

$$co - count_{w_s} = \sum_{d=1}^{D} \sum_{w_{\neg s} \in \neg \mathbf{S}} (x_{w,s}^{(d)} - \tilde{x_{w,s}}^{(d)})(x_{w,\neg s}^{(d)} - \tilde{x_{w,\neg s}}^{(d)}) \tag{5.5}$$

$$\mathbb{E}[(x_{w,s}^{(d)} - \tilde{x_{w,s}}^{(d)})(x_{w,\neg s}^{(d)} - \tilde{x_{w,\neg s}}^{(d)})] = [(N_t - N_{ns})(N_t - N_{ns} + 1)]\beta_{w,s}\beta_{w,\neg s} \tag{5.6}$$

$$\mathbb{E}[co - counts_{w_s}] = \sum_{d=1}^{D}[(N_t - N_{ns})(N_t - N_{ns} + 1)]\nu_{w,s} \sum_{w \in \neg \mathbf{S}} \nu_{w,\neg s}[1 - \gamma_{w,s}\gamma_{w,\neg s}] \tag{5.7}$$

$$\min_{\gamma_{w,s}\gamma_{w,\neg s}} \frac{1}{2}\sum_{w \in \mathbf{S}}(co - counts_{w_s} - \mathbb{E}[co - counts_{w_s}]) + \frac{1}{2}\sum_{w \in \neg \mathbf{S}}(co - counts_{w_{\neg s}} - \mathbb{E}[co - counts_{w_{\neg s}}])$$
$$subject\ to \sum_{w \in \mathbf{S}} \beta_{w,s} = \sum_{w \in \neg \mathbf{S}} \beta_{w,\neg s} = 1;\ \forall_{w \in \mathbf{S}}\ 0 \leq \gamma_{w,s} < 1;\ \forall_{w \in \neg \mathbf{S}}\ 0 \leq \gamma_{w,\neg s} < 1; \tag{5.8}$$

$$\begin{aligned} diff = &\sum_{w \in s} \log\left(\frac{\Gamma(x_w - \tilde{x_w} + \delta(1+\gamma_w)\nu_w)\Gamma(\delta(1-\gamma_w)\nu_w)}{\Gamma(\delta(1+\gamma_w)\nu_w)\Gamma(x_w - \tilde{x_w} + \delta(1-\gamma_w)\nu_w)}\right) \\ &- \sum_{w \in \neg s} \log\left(\frac{\Gamma(x_w - \tilde{x_w} + \delta(1+\gamma_w)\nu_w)\Gamma(\delta(1-\gamma_w)\nu_w)}{\Gamma(\delta(1+\gamma_w)\nu_w)\Gamma(x_w - \tilde{x_w} + \delta(1-\gamma_w)\nu_w)}\right) \end{aligned} \tag{5.9}$$

5.4.3 Extracting sentiment labels

We use the difference score achieved from equation 5.9 in an All-Versus-All (AVA) decomposition scheme, which decomposes ternary class classification to the binary class case. As the name suggests in the AVA scheme, a binary classifier is generated for each possible pair of class comparisons. For example, if there are a total of M classes in a multi-class classification problem, then the number of pairs for comparison becomes ${}^{M}C_2$. The AVA scheme is represented by a matrix $\boldsymbol{E}$ whose columns (denoted by $j \in \mathbf{j}$) give a binary classifier while the rows (denoted by $i \in \mathbf{i}$) represent the embedding vector of class labels. We represent the

class embedding vector as $\vec{i_m}$. In this chapter, we represent the AVA scheme by a 3×3 matrix as shown in Figure 5.7. We generate a difference score corresponding to each base binary classifier using equation 5.9. This forms a one-dimensional array ($\vec{diff^d}$) (of $M = 3$) elements of difference score corresponding to each binary classifier for each maintenance record d. We evaluate the dot product between the one-dimensional array of the difference-score for a maintenance record d and the class embedding vector for each of the three classes and denote it as v_m^d. We generate a distance-score u_m^d measuring the distance between the array of score $\vec{diff^d}$ and the class embedding vector (represented by the row of the AVA matrix) by following the distance measure suggested in [7]. Mathematically $v_m^d = \vec{i_m} \cdot \vec{diff^d}$ and $u_m^d = |\vec{i_m}| - \frac{v_m^d}{2}$. We identify the class with which the score u_m^d is minimum ($argmin_m\{u_m^d\}$) to be the closest to the one-dimensional array of the score, and thus the maintenance record is assigned the corresponding class label (which has the minimum dot product). We demonstrate the application of the proposed AVA scheme for a maintenance record d in Figure 5.7. In Figure 5.7 the minimum distance-score u_m^d is obtained for the positive class (2.825). Thus, maintenance record d is given a positive class label. The steps involved in applying LUSAA-N model are summarized in algorithm 9.

Classifier/ Class (m)	C(1): Pos vs Neg	C(2): Ntr vs Pos	C(3): Neg Vs Ntr	Array of diff score ($\vec{diff^d}$)	Dot – Product (v_m^d)	Final – Score (u_m^d)
Positive	1	-1	0	[0.23, -0.12, -0.09]	= 0.35	3-0.35/2 = 2.825
Negative	-1	0	1	[0.23, -0.12, -0.09]	= -0.32	3-(-0.32/2) = 3.16
Neutral	0	1	-1	[0.23, -0.12, -0.09]	= -0.03	3-(-0.03/2) = 3.015

Figure 5.7: Illustration of AVA scheme to identify sentiment label for a maintenance record $d \in \mathbf{D}$

Algorithm 9: Steps to learn LUSAA-N parameters

Input: $x_{w_{pos}}, x_{w_{neu}}, x_{w_{neg}}, x_{\tilde{w_{pos}}}, x_{\tilde{w_{neu}}}, x_{\tilde{w_{neg}}}, \boldsymbol{E}$ *documents* $d \in \mathbf{D}$

/* $w_{pos}, w_{neu}, w_{neg}$ denote the lexicons belonging to positive, neutral and negative sentiment classes */

Output: $\vec{diff}^d$ vector of difference scores of each binary classifier for each maintenance record d.

Initialize $\vec{diff}^d \rightarrow \emptyset$

for $j \in \boldsymbol{E}$ **do**

  Let $s \rightarrow$ sentiment class with entry $+1$ in j & $\neg s \rightarrow$ sentiment class with entry -1 in j

  Initialize $\nu_{w,s}, \nu_{w,\neg s} \rightarrow$ with word frequency $\forall d \in \mathbf{D}$

  Generate $co - counts_{w_s}$ and expected co-counts $\mathbb{E}[co - counts_{w_s}]$ using equation 5.5 and 5.7

  Optimize $\gamma_{w,s}, \gamma_{w,\neg s}$ by optimizing equation 5.8

  for $d \in \mathbf{D}$ **do**

    Calculate $diff^d(j)$ using equation 5.9

$\vec{diff}^d \rightarrow < diff^d(j) > \forall j \in \mathbf{j}$

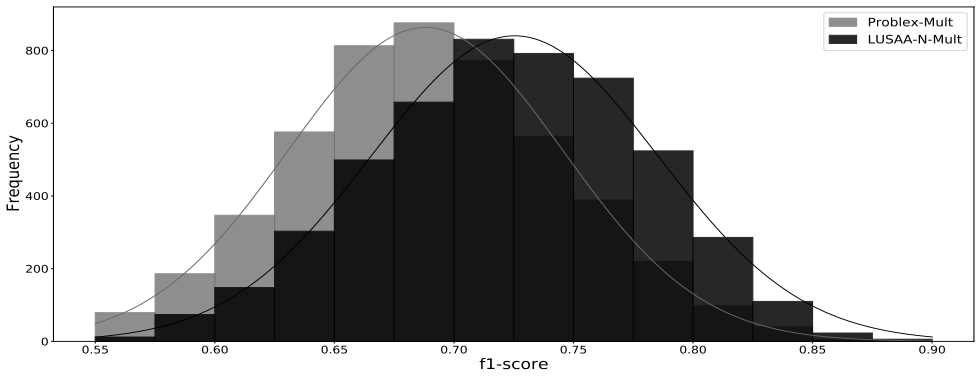
5.5 Case Study

We evaluate the proposed model using real-world maintenance records from oil rig equipment. We select data for a single equipment (Mud-Pump) for this case study with 4893 maintenance records. The total vocabulary size for the maintenance record is 5693. As mentioned earlier, our work assumes three degradation states for the equipment, which are mapped to three sentiments of Positive, Negative, and Neutral as described in section5.4.1. Our proposed *LUSAA-N* model has two variants depending on the concentration parameter δ values. When δ is very large, the model reduces to a Multinomial model, and we denote the same as *LUSAA-N-Mult*; otherwise, the model is a DCM model, and we denote the same as *LUSAA-N-DCM*[2]. The first comparison is with the baseline *ProbLex-Mult* and *ProbLex-DCM* model proposed by [46] which do not consider negation. To test the efficiency of adjusting for negation, we generate a manually labeled dataset of 153 maintenance records which is almost balanced with 52 records of positive sentiment, 51 of neutral sentiment,

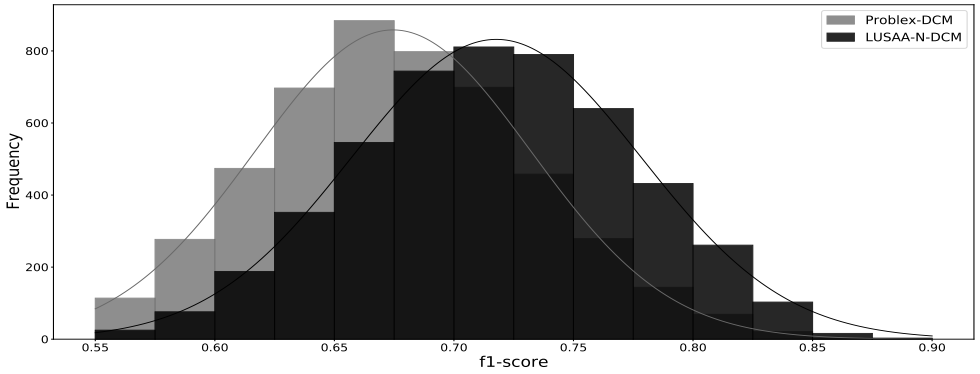
[2]The code would be uploaded at GitHub, post double blinded peer-review completion

and 50 of negative sentiment. We apply the proposed *LUSAA-N* and the baseline *Problex* model proposed by [46] with their two variants of multinomial and DCM distribution and generate final labels using the AVA decomposition scheme. We can use a metric relevant to classification such as weighted 'f1-score' to evaluate the models as we have class labels for predictions. We do not simply generate the weighted f1-score of the 153 labeled records but instead make the difference statistically apparent by bootstrapping 30 percent of the labeled records 5000 times and generating an f1-score for the bootstrapped samples for each of the four models. We keep the proportion of labels in the bootstrapped sample the same as in the original dataset. In Figure 5.8 we demonstrate the bootstrapped f1-score for each model where we try to present the benefits of adjusting for negation terms. As can be seen from Figure 5.8 the mean of the bootstrapped f1-score for *LUSAA-N* model (proposed by us) dominates the mean of the bootstrapped f1-score for the *Problex* model proposed by [46] for both variants of multinomial (Figure 5.8a) and DCM (Figure 5.8b) distributions.

Further, to demonstrate the effectiveness of the generated lexicons, we compare the proposed model with an open-source sentiment analysis model *VADER* [68]. To analyze the generalization of *BERT* [166] over other domains, we *fine-tune* it on the IMDB dataset of three sentiment labels to predict the health status of the industrial equipment. To benchmark the effectiveness of the proposed *LUSAA-N*s model, we compare the proposed model with the Point-wise Mutual Information (*PMI*) of [163]. We further compare the model with the Latent Sentiment Model (*LSM*) as proposed in [86] with sentiment priors initialized by bootstrapped lexicons for the base LDA model. We also consider the TSWE model proposed in [55], which uses word embeddings trained on maintenance records by the celebrated word2vec model of [106]. The TSWE model proposed by [55] extends the framework of [113], which was initially generated to improve topic models (LDA) by incorporating information from word embeddings. However, as we only have a single topic (Mud-Pump) throughout the case study, we consider the number of topics to be 1, and thus we name the implementation as (*SWE*). Further, as the *base-line topic models do not provide class labels* but instead help



(a) Bootstrapped f1-score comparison of LUSAA-N-Mult model with baseline Problex-Mult model (by [46])



(b) Bootstrapped f1-score comparison of LUSAA-N-DCM model with baseline Problex-DCM model (by [46])

Figure 5.8: Bootstrapped f1-score comparison for LUSAA-N and Problex models

to achieve clusters, we use Adjusted Rand Index (ARI) [66] as a metric to judge the effectiveness of each model for benchmarking. ARI automatically identifies the maximum possible overlap between two clusters, and hence the need to identify correct labels is avoided. The unavailability of class labels can also be considered as a potential shortcoming of methods that only identify topics using the topic modeling (mixture distribution) framework. The application of an encoding strategy proposed in *LUSAA-N* helps achieve the class labels. We plot the ARI for all the models in Figure 5.9. From Figure 5.9, we can easily infer that both variants of *LUSAA-N* model outperform other competitive models. Specifically, we can see that applying an open-source tool like *VADER* is highly inefficient, and also the adaptability of *fine-tunned BERT* on any publicly available labeled dataset gives bad results. Thus, to identify the equipment health status from maintenance records, domain curated seed lexi-

cons are needed, and custom models like *LUSAA-N* that can handle negation provide the best accuracy.

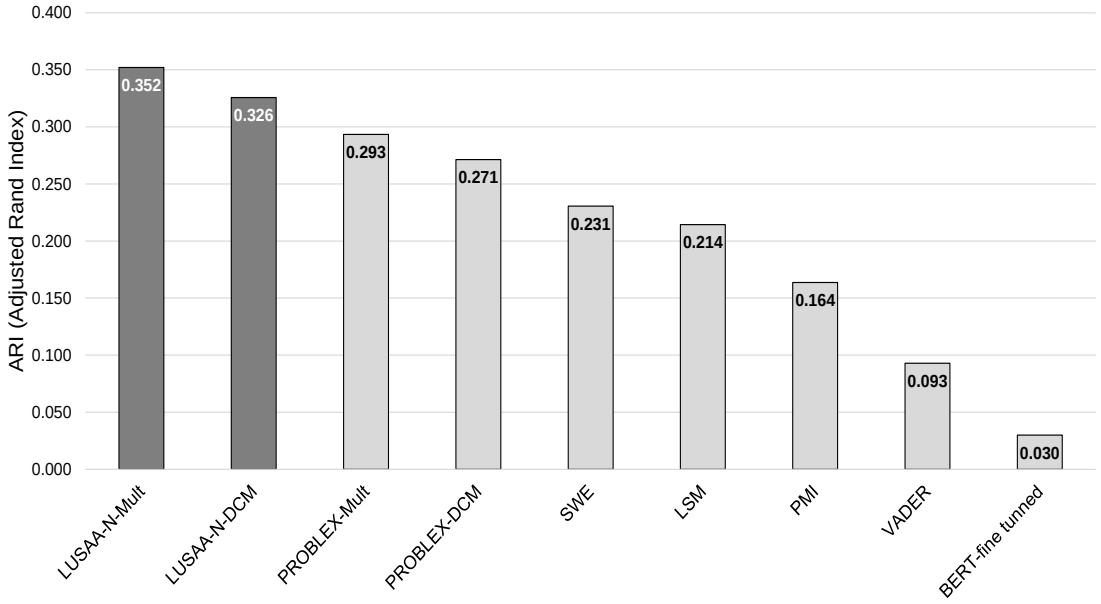


Figure 5.9: Comparison of Adjusted Rand Index (ARI) for all models

5.6 Future Work and Summary

In the future, we would like to extend our proposed work to conduct fine-grained sentiment analysis to identify health status of each sub-component of an hierarchical equipment under inspection. While research on fine grained sentiment analysis exists (e.g., [178], in the context of the hospitality industry), it has not yet been applied to analyze maintenance records and determine the sentiment of maintenance technicians regarding the condition of sub-components in industrial equipment. It will also be interesting to investigate multi-task models that model transfer learning and negation simultaneously. Further, the sentiment labels generated from the proposed *LUSAA-N* model could be used as additional information to generate stratified reliability models. Apart from this, the knowledge about equipment health status could be jointly used for maintenance decision making as proposed in [121].

In this chapter, we propose a novel Lexicon-based Unsupervised Sentiment Analysis Adjusted for Negation ($LUSAA$-N) model to identify equipment health status from unstructured, unlabeled maintenance records without any requirement of expensive computational resources. By testing the performance of the proposed model over a manually labeled dataset, we could identify that the proposed model outperforms the existing model. We also demonstrate the usefulness of having seed lexicons for industrial maintenance records and illustrate a workflow for bootstrapping domain-specific lexicons from a set of seed lexicons. Our results in the case study of section 5.5 show that using general English lexicons is ineffective in extracting equipment health status from maintenance record. Our work shows the importance of modeling negation in accurately identifying equipment health status from maintenance records. We also find that domain generalization of base pre-trained models like BERT is poor for transfer learning applications by fine-tuning when the target and source domains are highly separated.

5.7 Appendix A: Derivation of Binary Bayes Decision Rule

The binary Bayes classifier decision rule in terms of likelihood is given by $\log P(x, \tilde{x}|S = s) \gtrless \log P(x, \tilde{x}|S = \neg s)$ in favor of sentiment class $s(>)$ or $\neg s(<)$ respectively. Using the expression of Likelihood from equation 5.2 and taking *log* on both sides, the binary Bayes

classifier decision rule can be expanded as shown in equation 5.10

$$\log\left(\frac{n!}{K(x)}\right)+\log\left(\frac{\Gamma(\sum_{w=1}^{W}\alpha_{w,s})}{\Gamma(N_t-N_{ns}+\sum_{w=1}^{W}\alpha_{w,s})}\right)+\sum_{w=1}^{W}\log\left(\frac{\Gamma(x_w-\tilde{x_w}+\alpha_{w,s})}{\alpha_{w,s}}\right)\gtrless$$
$$\log\left(\frac{n!}{K(x)}\right)+\log\left(\frac{\Gamma(\sum_{w=1}^{W}\alpha_{w,\neg s})}{\Gamma(N_t-N_{ns}+\sum_{w=1}^{W}\alpha_{w,\neg s})}\right)+\sum_{w=1}^{W}\log\left(\frac{\Gamma(x_w-\tilde{x_w}+\alpha_{w,\neg s})}{\alpha_{w,\neg s}}\right)$$
$$\textit{Cancelling the } \log\left(\frac{n!}{K(x)}\right) \textit{ term.} \tag{5.10}$$

Also, using inequality, $\Gamma(y+1)=(y)\Gamma(y)$ *we can write*

$$\Gamma\left(N_t-N_{ns}+\sum_{w=1}^{W}\alpha_{w,\neg s}\right)=(N_t-N_{ns})!\Gamma\left(\sum_{w=1}^{W}\alpha_{w,\neg s}\right)$$

Substituting this in the above rule becomes

$$\log\left(\frac{\Gamma(\sum_{w=1}^{W}\alpha_{w,s})}{(N_t-N_{ns})!\Gamma(\sum_{w=1}^{W}\alpha_{w,s})}\right)+\sum_{w=1}^{W}\log\left(\frac{\Gamma(x_w-\tilde{x_w}+\alpha_{w,s})}{\alpha_{w,s}}\right)\gtrless$$
$$\log\left(\frac{\Gamma(\sum_{w=1}^{W}\alpha_{w,\neg s})}{(N_t-N_{ns})!\Gamma(\sum_{w=1}^{W}\alpha_{w,\neg s})}\right)+\sum_{w=1}^{W}\log\left(\frac{\Gamma(x_w-\tilde{x_w}+\alpha_{w,\neg s})}{\alpha_{w,\neg s}}\right) \tag{5.11}$$

The first terms on both right hand side and left hand side cancels out in equation 5.11 reducing the decision rule to equation 5.12.

$$\sum_{w=1}^{W}\log\left(\frac{\Gamma(x_w-\tilde{x_w}+\alpha_{w,s})}{\alpha_{w,s}}\right)\gtrless\sum_{w=1}^{W}\log\left(\frac{\Gamma(x_w-\tilde{x_w}+\alpha_{w,\neg s})}{\alpha_{w,\neg s}}\right) \tag{5.12}$$

The first term in equation 5.12 gives the score for the true underlying sentiment class to correspond to s, while second term gives the score of the true underlying sentiment class corresponding to $\neg s$. Thus, for a given maintenance record if in reality the true class is s (corresponding to first term of equation 5.12), then lexicons of type $\neg s$ in the document would have reduced predictiveness given by $(1-\gamma_{w,\neg s})\nu_{w,\neg s}$. But if the true underlying sentiment class is $\neg s$ (corresponding to second term of equation 5.12), then lexicons of type s would instead have reduced predictiveness given by $(1-\gamma_{w,s})\nu_{w,s}$. Also using the fact that $\alpha_w=\delta\beta_w$ the decision rule of equation 5.12 in terms of lexicons can be written as shown in

equation 5.13.

$$\sum_{w\in s}\log\left(\frac{\Gamma(x_{w,s}-\tilde{x_{w,s}}+\delta(1+\gamma_{w,s})\nu_{w,s})}{\delta(1+\gamma_{w,s})\nu_{w,s}}\right)+\sum_{w\in\neg s}\log\left(\frac{\Gamma(x_{w,\neg s}-\tilde{x_{w,\neg s}}+\delta(1-\gamma_{w,\neg s})\nu_{w,\neg s})}{\delta(1-\gamma_{w,\neg s})\nu_{w,\neg s}}\right)$$
$$\geqslant$$
$$\sum_{w\in\neg s}\log\left(\frac{\Gamma(x_{w,\neg s}-\tilde{x_{w,\neg s}}+\delta(1+\gamma_{w,\neg s})\nu_{w,\neg s})}{\delta(1+\gamma_{w,\neg s})\nu_{w,\neg s}}\right)+\sum_{w\in s}\log\left(\frac{\Gamma(x_{w,s}-\tilde{x_{w,s}}+\delta(1-\gamma_{w,s})\nu_{w,s})}{\delta(1-\gamma_{w,s})\nu_{w,s}}\right) \tag{5.13}$$

Arranging lexicons of same types to the same side of the inequality the decision rule corresponds to equation 5.14

$$\sum_{w\in s}\log\left(\frac{\Gamma(x_{w,s}-\tilde{x_{w,s}}+\delta(1+\gamma_{w,s})\nu_{w,s})}{\delta(1+\gamma_{w,s})\nu_{w,s}}\right)-\sum_{w\in s}\log\left(\frac{\Gamma(x_{w,s}-\tilde{x_{w,s}}+\delta(1-\gamma_{w,s})\nu_{w,s})}{\delta(1-\gamma_{w,s})\nu_{w,s}}\right)$$
$$\geqslant$$
$$\sum_{w\in\neg s}\log\left(\frac{\Gamma(x_{w,\neg s}-\tilde{x_{w,\neg s}}+\delta(1+\gamma_{w,\neg s})\nu_{w,\neg s})}{\delta(1+\gamma_{w,\neg s})\nu_{w,\neg s}}\right)-\sum_{w\in\neg s}\log\left(\frac{\Gamma(x_{w,\neg s}-\tilde{x_{w,\neg s}}+\delta(1-\gamma_{w,\neg s})\nu_{w,\neg s})}{\delta(1-\gamma_{w,\neg s})\nu_{w,\neg s}}\right) \tag{5.14}$$

Combing the log terms together in equation 5.13 give us the decision rule of equation 5.4, thus concluding the proof.

5.8 Appendix B: Derivation of Expected Co-counts

We first derive the expected *product* of counts for a given word pair $(w_s, w_{\neg s})$. Using the definition of co-occurrence count of lexicon w_s with all opposite lexicons $w_{\neg s}$ adjusted for negation is give by equation 5.5. We assume that lexicons $x_{w,s}$ of sentiment type s in a maintenance record d of length N_t follow Multinomial$(N_t, \beta_{w,s})$ distribution, while lexicons $x_{w,\neg s}$ of sentiment type $\neg s$ in a maintenance record d of length N_t follow Multinomial$(N_t, \beta_{w,\neg s})$

distribution. It is reasonable to assume that lexicons occur with the same probability in negation scope. Thus, lexicons $x_{w,s}^{\sim}$ of sentiment type s follow Multinomial$(N_{ns}, \beta_{w,s})$ distribution in negation scope of length N_{ns}, while lexicons $x_{w,\neg s}^{\sim}$ of sentiment type $\neg s$ follow Multinomial$(N_{ns}, \beta_{w,\neg s})$ distribution in negation scope of length N_{ns}. Before proceeding to the derivation of equation 5.6 we would like to give the equations for first and second order moments to be used in the proof. Under the distributional assumptions the moments are given in equation 5.15. The covariance of cross terms like $cov\{x_{w,s}, x_{w,\neg s}^{\sim}\}$ in equation 5.15 is obtained by realizing the fact that covariance between multinomials defined on sets of length N_t and N_{ns} where $N_{ns} \subset N_t$ would be non-zero only for the set N_{ns}, further by assumption the probability of occurrence are same in both the sets N_t and N_{ns}.

$$
\begin{aligned}
&\mathbb{E}(x_{w,s}) = N_t\beta_{w,s}, \mathbb{E}(x_{w,\neg s}) = N_t\beta_{w,\neg s} \\
&\mathbb{E}(x_{w,s}^{\sim}) = N_{ns}\beta_{w,s}, \mathbb{E}(x_{w,\neg s}^{\sim}) = N_{ns}\beta_{w,\neg s} \\
&cov\{x_{w,s}, x_{w,\neg s}\} = N_t\beta_{w,s}\beta_{w,\neg s} \\
&cov\{x_{w,s}^{\sim}, x_{w,\neg s}^{\sim}\} = N_{ns}\beta_{w,s}\beta_{w,\neg s} \\
&cov\{x_{w,s}^{\sim}, x_{w,\neg s}\} = N_{ns}\beta_{w,s}\beta_{w,\neg s} \\
&cov\{x_{w,s}, x_{w,\neg s}^{\sim}\} = N_{ns}\beta_{w,s}\beta_{w,\neg s}
\end{aligned}
\tag{5.15}
$$

We proceed with the derivation subsequently in equation 5.16.

$$
\begin{aligned}
\mathbb{E}[(x_{w,s} - x_{\tilde{w},s})(x_{w,\neg s} - x_{\tilde{w},\neg s})] = & \, cov[(x_{w,s} - x_{\tilde{w},s})(x_{w,\neg s} - x_{\tilde{w},\neg s})] \\
& + \mathbb{E}(x_{w,s} - x_{\tilde{w},s})\mathbb{E}(x_{w,\neg s} - x_{\tilde{w},\neg s}) \\
= & cov[(x_{w,s}, x_{w,\neg s})] - cov[(x_{w,s}, x_{\tilde{w},\neg s})] - cov[(x_{\tilde{w},s}, x_{w,\neg s})] + \\
& cov[(x_{\tilde{w},s}, x_{\tilde{w},s})] \\
& + [\mathbb{E}(x_{w,s}) - \mathbb{E}(x_{\tilde{w},s})][\mathbb{E}(x_{w,\neg s}) - \mathbb{E}(x_{\tilde{w},\neg s})] \\
= & cov[(x_{w,s}, x_{w,\neg s})] - cov[(x_{w,s}, x_{\tilde{w},\neg s})] - cov[(x_{\tilde{w},s}, x_{w,\neg s})] + \\
& cov[(x_{\tilde{w},s}, x_{\tilde{w},s})] + \mathbb{E}(x_{w,s})\mathbb{E}(x_{w,\neg s}) \\
& - \mathbb{E}(x_{\tilde{w},s})\mathbb{E}(x_{w,\neg s}) - \mathbb{E}(x_{w,s})\mathbb{E}(x_{\tilde{w},\neg s}) + \mathbb{E}(x_{\tilde{w},s})\mathbb{E}(x_{\tilde{w},\neg s}) \\
= & N_t\beta_{w,s}\beta_{w,\neg s} - N_{ns}\beta_{w,s}\beta_{w,\neg s} - N_{ns}\beta_{w,s}\beta_{w,\neg s} + N_{ns}\beta_{w,s}\beta_{w,\neg s} \\
& + N_t^2\beta_{w,s}\beta_{w,\neg s} - N_{ns}\beta_{w,s}N_t\beta_{w,\neg s} - N_t\beta_{w,s}N_{ns}\beta_{w,\neg s} + \\
& N_{ns}^2\beta_{w,s}\beta_{w,\neg s} \\
= & [N_t^2 - 2N_tN_{ns} + N_{ns}^2 + N_t - N_{ns}]\beta_{w,s}\beta_{w,\neg s} \\
= & [(N_t - N_{ns})(N_t - N_{ns} + 1)]\beta_{w,s}\beta_{w,\neg s}
\end{aligned}
\tag{5.16}
$$

Further using equation 5.16 and 5.3 we can write.

$$\begin{aligned}
\mathbb{E}[(x_{w,s} - \tilde{x_{w,s}})(x_{w,\neg s} - \tilde{x_{w,\neg s}})|S = s] &= [(N_t - N_{ns})(N_t - N_{ns} + 1)]* \\
&\quad \{\nu_{w,s}(1 + \gamma_{w,s})\nu_{w,\neg s}(1 - \gamma_{w,\neg s})\} \\
&= [(N_t - N_{ns})(N_t - N_{ns} + 1)]\nu_{w,s}\nu_{w,\neg s}[1 + \\
&\quad \gamma_{w,s} - \gamma_{w,\neg s} - \gamma_{w,s}\gamma_{w,\neg s}] \\
\mathbb{E}[(x_{w,s} - \tilde{x_{w,s}})(x_{w,\neg s} - \tilde{x_{w,\neg s}})|S = \neg s] &= [(N_t - N_{ns})(N_t - N_{ns} + 1)]* \\
&\quad \{\nu_{w,s}(1 - \gamma_{w,s})\nu_{w,\neg s}(1 + \gamma_{w,\neg s})\} \\
&= [(N_t - N_{ns})(N_t - N_{ns} + 1)]\nu_{w,s}\nu_{w,\neg s} \\
&\quad [1 - \gamma_{w,s} + \gamma_{w,\neg s} - \gamma_{w,s}\gamma_{w,\neg s}]
\end{aligned} \tag{5.17}$$

Thus, the expected *product* of counts for a given word pair $(w_s, w_{\neg s})$ can be given as.

$$\begin{aligned}
\mathbb{E}[(x_{w,s} - \tilde{x_{w,s}})(x_{w,\neg s} - \tilde{x_{w,\neg s}})] &= P(S = s)\mathbb{E}[(x_{w,s} - \tilde{x_{w,s}})(x_{w,\neg s} - \tilde{x_{w,\neg s}})|S = s] \\
&\quad + P(S = \neg s)\mathbb{E}[(x_{w,s} - \tilde{x_{w,s}})(x_{w,\neg s} - \tilde{x_{w,\neg s}})|S = \neg s] \\
&= [(N_t - N_{ns})(N_t - N_{ns} + 1)]\nu_{w,s}\nu_{w,\neg s}[1 - \gamma_{w,s}\gamma_{w,\neg s}]
\end{aligned} \tag{5.18}$$

Next we derive the expected co-counts of a word w_s occurring with a cross lexicon word pair $w_{\neg s}$ by summing over all words $w_{\neg s} \in \neg\mathbf{S}$ over all documents $\mathbf{D}$.

$$\begin{aligned}
\mathbb{E}[co - counts_{w,s}] &= \sum_{d=1}^{D} \sum_{w \in \neg\mathbf{S}} \mathbb{E}[(x_{w,s}^{d} - \tilde{x_{w,s}^{d}})(x_{w,\neg s} - \tilde{x_{w,\neg s}^{d}})] \\
&= \sum_{d=1}^{D} [(N_t - N_{ns})(N_t - N_{ns} + 1)]\nu_{w,s} \sum_{w \in \neg s} \nu_{w,\neg s}[1 - \gamma_{w,s}\gamma_{w,\neg s}]
\end{aligned} \tag{5.19}$$

5.9 Appendix C: Seed Industrial Sentiment Lexicon

Positive Seed Lexicon (Equipment restored to healthy (as good as new) condition): replaced; installed; was repaired; was replaced; back in service; was reset; ready to use; ready for use; deisolated; changed on; is being closed ; is replaced; is closed; is changed; was replaced; was installed; was closed; is installed; is fixed; was fixed; is up; completed; has been replaced; has arrived; have arrived; have been replaced; have been repaired; has been repaired; have been installed; has been installed; has been completed; have been completed; job closed; job completed; module installed; parts received; parts replaced; is changed; is done; is fitted; is returned; is working; is rested; properly completed; is operational now; back in operation; is finished; was finished; after replacement; after change; corrected; correctly changed; correctly completed; correctly removed; currently installed; currently in use; is activated; assembled; is updated; just completed; restored; auto canceled; request closed; damage repaired; removed; rectified; reconstructed; adjusted; new part; as good as new; reinstate; renewed; rebooted; closing the job; completed the cm; back to normal; removed old installed new; replaced and tested ok; changed with new.

Negative Seed Lexicon (Equipment in Bad health condition): to be ordered; to be fitted; out of service; needs repair; unusual; stopped; work in progress; was contaminated; triggered; isolated; washed out ; found leak ; to be fixed; is down; abnormal corrosion; broken; found corroded; damaged; excessive damage; excessive scuffing; found corrosion; is frozen; found leaks; is leaking; severe corrosion; significant damage; wear damage; worn; deficiencies found; deterioration found; discrepancies found; issues found; job created; job opened; failed; is damaged; is shut; is stuck; abnormalities found; eroded; corroded; out of operation; not in operation; corrective maintenance ; a cm; is broken; is defective; disassembled; to be repaired; to be replaced; to be changed; to be issued; to be performed; to be removed; to be sent; will be opened; will be changed; will be fixed; will be installed; to be

installed; seized working; malfunctioning; faulted; above normal; to be rectified; heavy fault; distort; pop off; catastrophe; above recommended; loose connection; not working fine; are not functioning; missing; to repair; need to fit; still remained failed; not come back up; to perform.

Neutral Seed Lexicon (Equipment working fine): confirmed; small discrepancy; good test; checked; certified; tested; cleaned; found fit; good condition; set correctly; looks great; all ok; completed cleaning; examined; inspected; is fine; is good; to be tested; have been tested; in good condition; in good order; is checked; is ok; is acceptable; is in accordance; all good; all acceptable; all in good condition; all in good order; below excessive wear; below damage; is inspected; found ok; found good; in accordance; all certified; parts cleaned; minor repairs; preventive maintenance; a pm; tightened; already calibrated; monitored; confirmed good; correctly checked; correctly cleaned; correctly ensured; correctly examined; is examined; firmly tightened; function tested; have confirmed; in acceptable condition; in acceptable operating; in correct position; looks satisfactory; looks ok; below normal; is tightened; is cleaned; cleaned dirt; cleaned mud; visually inspect; no issues found; free of ; all working well; are functioning; verified; completed the pm; secured; closed with no action; are tight; found to be good ; is working properly; is transmitted correctly; found dust; are operational; functions worked properly; working fine.

Chapter 6

Condition Based Maintenance by joint modeling of continuous sensor signal and discrete maintenance events using Action Specific-Input Output Hidden Markov Model (AS-IOHMM)

6.1 Focused Abstract

Costly downtime of complex industrial equipment is a significant concern for companies. To prevent downtime, technicians conduct periodic inspections and manual repairs, documenting their findings as unstructured free text in maintenance records. These records contain valuable information, including direct observations of the equipment's health status. Further, companies also gather real-time sensory data to indirectly assess the equipment's health. However, current prognosis techniques are limited in their ability to model the system dynamics using both direct and indirect state observations, as well as the effects of

different maintenance actions. This research introduces a novel joint modeling approach called Action-Specific Input-Output Hidden Markov Model (AS-IOHMM), which integrates real-time sensor signals and discrete state information from manual interventions to enable holistic condition-based maintenance. This approach enhances the modeling accuracy of system dynamics, and overcomes challenges commonly faced in learning transition probability matrices for prognosis. We demonstrate the effectiveness of AS-IOHMM through a numerical case study and validate its performance using real-world industry data.

6.2 Introduction

Modeling system dynamics is crucial for reliability studies and developing optimal maintenance and warranty policies. System failures result in costly downtime for equipment owners, depending on the fault magnitude and spare parts availability. System dynamics modeling helps estimate remaining useful life, infer current health status, and conduct prognostic studies in industrial maintenance [59] and healthcare [56]. Periodic manual inspections remain a significant approach for gaining insights into system health and degradation [109], although they provide sparse and discontinuous information. Condition Based Maintenance (CBM) systems complement manual inspections by gathering real-time data from multiple sensors to indirectly infer the latent system health status.

Statistical analysis of sensory data to assess equipment condition is a prominent area of research in reliability studies [143]. The typical workflow involves obtaining a degradation signal (e.g., car battery resistance [149]) that is assumed to monotonically increase and is modeled to infer the latent system health state under certain assumptions. When the degradation signal crosses a predefined threshold, it implies that the system is approaching failure, prompting maintenance actions to prevent equipment breakdown. This type of failure is referred to as soft failures [149]. However, these models are inefficient for cases where equipment suddenly stops working due to catastrophic failures, known as hard failures [149].

Hard failures are typically modeled using the proportional hazard model [87] to predict equipment failure by estimating its hazard or failure rate. However, standalone hazard models do not consider the additional information provided by CBM signals. To address this, researchers have developed two-step methods that integrate Degradation and Hazard models [149], [192], [65], [99]. In some cases, variational inference algorithms have been deployed to estimate model parameters by simultaneously learning from time-series degradation data and hard failure event data [184]. It is worth noting that existing joint models only consider the simultaneous modeling of degradation signals with hard failure data, where equipment ceases to work.

In practical scenarios, equipment owners rely on both CBM signals and periodic manual inspections to assess system health. Additionally, maintenance actions, such as minor preventive repairs (e.g., greasing or tightening bolts) or major corrective repairs (e.g., part replacement), are performed, which not only improve the system's current health but also impact the trajectory of CBM signals. For example, a pump with loose bolts produces larger vibrations than one with properly tightened bolts. Therefore, it is essential to consider the impact of the periodic maintenance actions while modeling system dynamics. Moreover, technicians performing maintenance actions directly observe the system state [109]. Their observations during inspections are recorded in maintenance records as unstructured free text, containing valuable information about equipment health, operating conditions, vulnerable parts, failure mechanisms, etc. Extracting meaningful insights about equipment health from these records can be achieved through manual review or natural language processing techniques like sentiment analysis. By combining observations from manual inspections and CBM signals, we can obtain both direct and indirect information about the system state.

Maintenance records can provide valuable insights for prognostic studies. Figure 6.1 contains maintenance records from oil and gas exploration equipment showing the actions performed by technicians and their outcomes. In this study, four different actions are considered: equipment inspection, preventive maintenance, corrective maintenance, and reliance

on CBM signals for system state information. In this chapter, preventive and corrective maintenance activities are collectively referred to as repair actions. By analyzing the maintenance record, either manually or using NLP sentiment analysis, the system's state after the completion of an action can be inferred. If technicians only inspect the equipment, the inferred state corresponds to the system's state at the time of inspection, as assumed in [109]. However, if technicians perform any maintenance action, the inferred state from the maintenance record depends on the job completion status and the type of repair (perfect/imperfect) conducted.

Date	Maintenance Records	Action Taken	Infered State after completing Action as per Maintenance Records
9/30/2017 0:19	**Replaced** broken wire with new. **Motor returned back to service in good condition**.	Corrective Maintenance (Replace)	Positive
11/18/2017 20:42	**Greased** motor with a maxium of 4 shots of grease. Motor operating **normally** now.	Preventive Maintenance (Grease)	Neutral
4/23/2018 22:17	**Changed** out nuts due to teflon being worn in other nuts. **Motor operations** still down**.**	Corrective Maintenance (Replace)	Negative
8/30/2018 21:03	**Changed** motor bearing with new ones. Motor operating smoothly without any issues**.**	Corrective Maintenance (Replace)	Negative
12/7/2018 0:50	Blower motors in **good condition**. Hoses and connections are in **good condition**. Grounds were secured in **good condition**	Inspection	Positive
01/20/2019 10:02	Job **was not completed** and **is being voided** due to incorrect part coding. **Parts were cancelled and not replaced** and **will be requested again with the new job**.	Corrective Maintenance (Replace)	Negative
1/30/2019 3:53	**Inspected** the motors and **found loose wires** from the mud side.	Inspection	Negative
3/11/2019 4:34	Inspection of motor **while running.** Checking for abnormal vibration or bearing noise cleanliness glands and cabling and mounting. **No issues found at time of inspection.**	Inspection	Neutral

Figure 6.1: Illustration of maintenance events with inferred states from maintenance records

Unlike existing literature, this research emphasizes the significance of the states inferred from the maintenance record and the knowledge of technicians' actions in modeling system dynamics. Incorporating this information can enhance the precision of modeling CBM sensory signals. To address this, we propose a novel Action-Specific Input-Output Hidden Markov Model (AS-IOHMM). The AS-IOHMM captures the unobserved dynamics of a system monitored through CBM sensory signals and subjected periodically to manual maintenance actions. The model and algorithm proposed in this chapter have several novel

capabilities:

1. Our model considers discrete actions performed on the system as inputs, generating a corresponding transition matrix for each action. This enables us to capture the difference in system dynamics based on each action.

2. The information obtained from inspections or repair actions improves the precision of modeling CBM sensory signals.

3. Our algorithm enables simultaneous modeling of missing discrete variables, derived from manual actions, and continuous CBM sensory signals, providing a novel approach for jointly modeling discrete and continuous variables.

4. Our model employs the Expectation-Maximization algorithm, leveraging numerical optimization for learning model parameters. We provide new routines for optimal parameter initialization, crucial for convergence of the numerical optimizer using hyperparameter tuning framework.

5. Once trained, the model has practical applications in developing model-based decision-making frameworks.

The chapter is structured as follows: Section 6.3 reviews existing literature on HMM and related topics and identifies gaps. Section 6.4 describes our proposed model and relevant algorithms. In Section 6.5, we evaluate the efficiency of our model using a simulated dataset, and in Section 6.6, we apply our model to real-world equipment data from the oil and gas industry. Finally, in Section 6.7, we conclude our work and outline future directions.

6.3 Literature Review

Our study aims to model the latent dynamics of a system that is partially observed using a CBM sensory signal and is subjected to different actions that impact the system's dynamics and help to directly observe the system state. This modeling approach helps increase

precision in capturing the latent dynamics of CBM sensory signals. The modeled dynamics would aid in prognostic studies by predicting future states and assist in developing optimal maintenance and inspection policies. When labels for system states are available, developing discriminative models for time series segmentation using Conditional Random Fields (CRF) [1] may help achieve the initial prognostic task. However, in industrial settings, the latent states are only revealed when a manual action is performed on the system. Therefore, to accurately model the latent dynamics, generative models such as Hidden Markov Models (HMM) [122] are considered superior to CRF since they do not require supervised training with labeled data [155]. [58] performed time series segmentation using Bernoulli emissions in a regular HMM, which segments a time series signal into the onset or offset of an activity. In our case, the emissions involve mixed variables, including continuous CBM sensory signals and discrete state emissions variables generated from manual maintenance actions.

Hidden Markov Models and their variants have long been used in reliability studies to model degradation signals [27]. To represent the health status of equipment monitored synchronously by multiple sensors, [43] relaxed the one-step assumption of HMMs by employing Hidden Semi-Markov Models. Additionally, [108] utilized a flexible stochastic process known as the nonhomogeneous continuous-time hidden semi-Markov process (NHCTHSMP) to aid in multi-state degradation modeling at the component/device level. Another approach presented by [187] involved an Expectation-Maximization method to model binary emissions with missing values. Furthermore, [32] proposed an HMM with auto-correlated observations (HMM-AO) where the observations depend on the corresponding hidden system state and the previous observations. In [147], the authors employed the Kalman filter to model continuous degradation signals and Logistic regression to model the binary failure process as a two-step joint model. Using Radial Basis Kernels, [182] proposed an HSMM with a Mixture of Kernels (MoK-HSMM) that combines condition monitoring data through a linear combination of RBF kernels. More recently, [38] proposed joint modeling of CBM sensory signals and failure event data using HMM for a system that continually degrades through a series of

hidden states. Although these extensions have significantly advanced degradation modeling, they do not consider the impact of varying actions taken during system operation. Moreover, these models are based on the traditional assumption that the degradation state of the equipment is always indirectly observed via CBM sensory signals. In contrast, our work not only considers the indirect observation of the system health status via CBM sensory signals but also consider the influence of varying discrete actions on system dynamics, which can help restore system health and provide direct insight about its health status.

To model the effect of external inputs/factors on system dynamics and emissions, [18] proposed an Input-Output Hidden Markov Model (IOHMM). However, their formulation does not distinguish between continuous and discrete input signals and provides a recurrent neural network-based structure. While leveraging the structure of IOHMM, our proposed model distinguishes itself by modeling the effect of discrete manual actions (such as preventive or corrective maintenance) on system dynamics through the generation of different transition matrices for each discrete action, rather than using a simple multinomial-link function between the discrete state space and continuous input sequences [34].

[9] introduced the very first mixed HMM to model the influence of random effects on lesion counts in multiple sclerosis patients. Their work utilizes a linear link function to model each patient's shape (mean) parameter with their random effects. In contrast, our work hypothesizes that, for a single system, the discrete actions taken by technicians can help model the scale and, consequently, the precision parameter of CBM signals more effectively. Recently, [137] proposed the use of a personalized IOHMM that combines the functionalities of Mixed HMM with IOHMM. However, their work assumes a standard transition matrix and does not consider the impact of medication (an input in their model) on the transition matrix. Furthermore, they do not address the case of jointly modeling continuous CBM signals and missing discrete action outcomes as emissions for IOHMM.

In equipment degradation modeling, [37] proposed the use of a Partially hidden Markov chain linear auto-regressive model, which models partial direct observations of states as

annotations using the partial HMM framework (proposed by [114], [136]). However, the authors of this work only utilize partial state information in the form of semi-supervision during HMM training and do not consider the effect of state knowledge in increasing the precision of the emission model for the CBM signal. Additionally, they do not account for the impact of manual actions performed by technicians on system state transitions. [33] trained a series of multiple operation-specific HMMs, where a separate HMM is trained for each operational condition (such as the thickness of semiconductor film deposition), corresponding to different emissions for each operation. The operation-specific HMMs assume the same hidden states representing degradation states. Lastly, [148] proposed a hybrid state-space model that captures the dependency of sensor signals on latent degradation states and the operating environment through Markovian assumptions using a feed-forward neural network, along with the binary failure process. However, their work considers a continuous operating environment, in contrast to the discrete manual actions considered in our work. Furthermore, the emission signal in their model solely focuses on the binary working status of the equipment (operational/failed), rather than the true degradation state revealed by manual actions.

6.4 Proposed Model

The proposed Action Specific Input-Output Hidden Markov Model (AS-IOHMM) models the emission signals by not only conditioning them on the latent degradation dynamics of the system but also the actions taken by technicians on the field for equipment. Further, the model also allows the transitions between different latent degradation states to depend on the actions taken by industrial technicians in the field. The AS-IOHMM (Figure 6.2) is composed of three elements, a conditional emission model (section 6.4.1) for the observed variable (which in our case are the CBM signals and inferred states observations revealed by manual actions), a transition model (section 6.4.1) for learning the hidden or latent state dynamics (which in our case would also depend on manual actions taken) and finally the learning

algorithm to learn the model parameters (section 6.4.2). Figure 6.2 shows how the proposed AS-IOHMM models the latent degradation state dynamics of the system. **S** represents the set of all discrete degradation states s that system could lie in. The set of actions available to the technicians at each time is represented by **U**, while every action taken by technicians is represented by u. The CBM signal, which provides indirect observation of the system state, is represented by a random variable **Y** whose individual realizations are represented by y. The random variable **X** is the discrete categorical variable that corresponds to the degradation state revealed by the action taken by the technician. Note that the variable **X** contains missing values (indicated in white in Figure 6.2) whenever the technician chooses to do nothing at all and relies on CBM signals for inferring latent states. **T** represent the set of all time steps t for which the system is operated. Our model is based on the single-step Markovian assumption, where the state of the system at any time t depends on the state at a previous time $t-1$ and the action taken at the previous time step $t-1$. The emissions $y,\ x$ depend on the state at any time t and the action taken at either step t or $t-1$, which we discuss further in section 6.4.1.

We introduce the list of notations that would be helpful for defining the model and its learning algorithm.

$t \in \mathbf{T} =$ instance of time;

$s_t \in \mathbf{S} =$ instance of latent degradation state at time t;

$u_t \in \mathbf{U} =$ instance of action taken at time t;

$x_t \in \mathbf{X} =$ instance of discrete state emission variable at time t;

$y_t \in \mathbf{Y} =$ instance of continuous variable for CBM signals;

$\pi \in \Pi =$ initial probabilities of states

$a_{i,j}(u) = p(s_t = j | s_{t-1} = i, u_{t-1}) \in \boldsymbol{A(i,j,u)}$ Transition matrix denoting transition probabilities of latent degradation states;

$\alpha_t(i) = p(x_1, y_1, x_2, y_2, \cdots, x_t, y_t, s_t = i | \mathbf{U}_1^t, \Theta)$ forward probability;

$\beta_t(j) = p(x_{t+1}, y_{t+1}, x_{t+2}, y_{t+2}, \cdots, x_T, y_T | s_t = j, \mathbf{U}_{t+1}^T, \Theta)$ backward probability;

$\gamma_t(i) = p(s_t = i | \mathbf{U}_1^T, \mathbf{X}_1^T, \mathbf{Y}_1^T, \Theta)$ probability of a degradation state at time t ;

$\xi_t(i,j) = p(s_t = i, s_{t+1} = j | \mathbf{U}_1^T, \mathbf{X}_1^T, \mathbf{Y}_1^T, \Theta)$ joint probability of consecutive states;

$\mu_s =$ mean parameter of CBM signals pdf depending on state s;

$\nu_s =$ precision parameter of CBM signals pdf depending on state s;

$\nu_u =$ precision parameter of CBM signals pdf depending on action u;

$\Theta = \{\eta_{s_t}, \eta_{u_t}, \mu_{s_t}, a_{i,j}(u)\}$ set of model parameters;

$\Omega =$ set of hyper-parameters used by Trust region algorithm and initial transition probabilities;

6.4.1 Conditional Emission and Transition Model

For AS-IOHMM we have two different emission variables, the continuous CBM signal y_t and the discrete state emission signal x_t. The emissions in AS-IOHMM depend not only on the latent degradation states but also on the action taken. In this work, we consider that technicians can take four actions. The technicians can do nothing and only observe the CBM signal (referred to as $u_t =$'0'), they can inspect the system (referred to as $u_t =$'1'), they can choose to preventively maintain the system by doing minor repairs (referred as $u_t =$'2') and finally, if the technicians can correctively maintain the system by doing a major repair (referred as $u_t =$'3'). Depending on the actions taken, the conditional dependence of the emission variables change. If the technicians choose to either do nothing or inspect the equipment, then conditioned of state and actions at time t emission variables (y_t, x_t) are independent of each other and their histories, i.e., $p(y_t | y_{1\cdots(t-1)}, x_{1\cdots(t-1)}, u_t, s_t) = p(y_t | u_t, s_t)$. Further, if $u_t = 1$, then $x_t = s_t$, i.e., the inspection at any time reveals the state at that time. Where else, if the technicians perform any repair action ($u_{t-1} = 1$ *or* $u_{t-1} = 2$), then the state information is revealed at the next time step, i.e., s_t is available and thus, conditional on given state and action taken at the previous time step the emissions are independent of

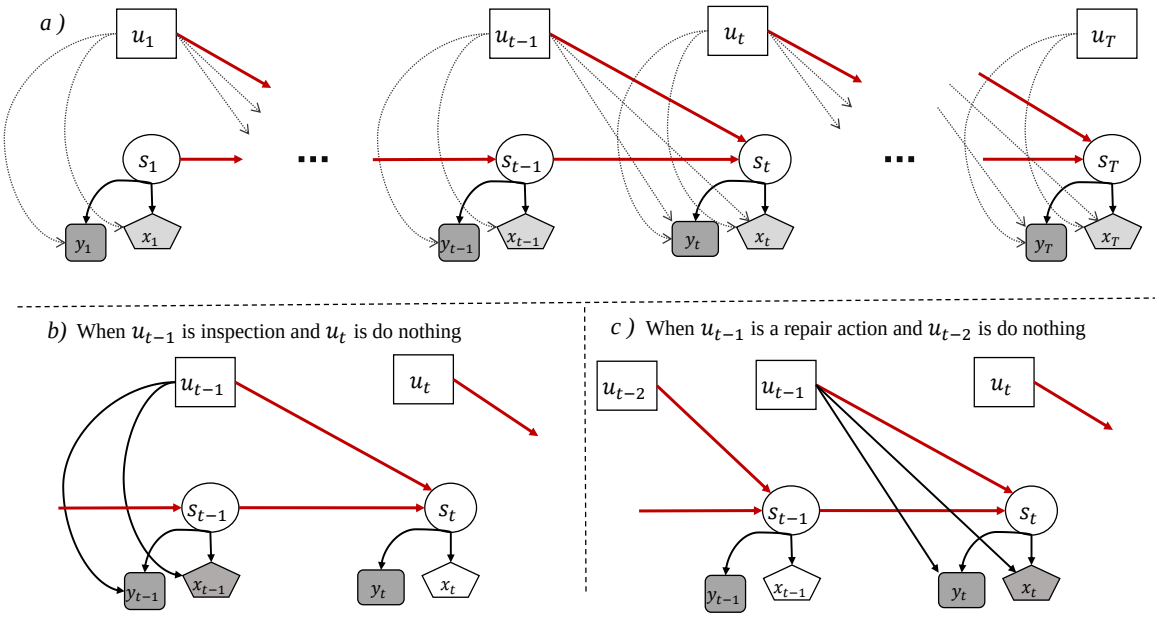


Figure 6.2: Proposed Action Specific-Input Output Hidden Markov Model (AS-IOHMM). The top part a represent the overall dynamics. The type of action taken by technician at each time step influence the emission models differently as shown in parts b and c.

each other, i.e., $p(y_t|y_{1\cdots(t-1)}, x_{1\cdots(t-1)}, u_{t-1}, s_t) = p(y_t|u_{t-1}, s_t)$. Also, if $u_{t-1} = 2$ *or* 3 then $x_t = s_t$.

We model the CBM signals using state-dependent Normal distributions. However, rather than specifying the Normal distribution in terms of variance, we use precision to model the scale parameter of the CBM signal. We parameterize the CBM signal emitting from a state s to have a mean μ_s and a precision associated with the state as ν_s. We define the precision associated with a state s as the inverse of the variance for normal distribution [158]. As precision is directly related to the spread of the values around the mean of the normal distribution, it is reasonable to assume that whenever we have the availability of the state information due to inspection or after a repair action, the emitted CBM signal value y_t should be close to the truth of the system at time t. Thus, we can say that our precision toward estimating the mean should increase if our actions reveal the system's state. Thus, depending on the action taken, we add the precision associated with action ν_u to the state

precision parameter. Particularly, when $u_t = `1' or u_{t-1} = `2'/`3'$ then the precision for the sample becomes $\nu_s + \nu_u$. Under these considerations, the probability distribution function of a uni-variate CBM signal is given by equation 6.1.

$$\begin{aligned} p(y_t|s_t, u_{t-1} = 0 \ or \ 1, u_t = 0) &= \frac{1}{\sqrt{2\pi}}\{\sqrt{\nu_s}\}\exp\{-1/2(y_t - \mu_s)^2\nu_s\} \\ p(y_t|s_t, u_t = 1) &= \frac{1}{\sqrt{2\pi}}\{\sqrt{\nu_s + \nu_u}\}\exp\{-1/2(y_t - \mu_s)^2(\nu_s + \nu_u)\} \\ p(y_t|s_t, u_{t-1} = 2 \ or \ 3) &= \frac{1}{\sqrt{2\pi}}\{\sqrt{\nu_s + \nu_u}\}\exp\{-1/2(y_t - \mu_s)^2(\nu_s + \nu_u)\} \end{aligned} \tag{6.1}$$

The discrete state emission variable x_t is defined on the domain of the state variable s_t and takes the value of state s_t whenever $u_t = `1'$ *or* $u_{t-1} = `2'/`3'$. Thus, its probability mass function corresponds to that of a Dirac delta function whose value is 1 whenever $x_t = s_t$ if $u_t = `1' or u_{t-1} = `2'/`3'$ and it is zero for all other $x_t \neq s_t$ as shown in equation 6.2. Whenever, the action is $u_{t-1} = 0$ *or* 1 *and* $u_t = 0$, the p.m.f of x_t is a uniformly distributed over $0 - |\mathbf{S}|$ at all discrete s_t. Finally, for the conditional transition probabilities, as state s_t at time t would depend upon the state s_{t-1} and action u_{t-1} at time $t-1$ the conditional transition probabilities are denoted as $p(s_t = j|s_{t-1} = i, u_{t-1})$. We note that the transition probability matrix $\boldsymbol{A(i, j, u)}$ in our case is a 3-dimensional array, where we have a separate transition matrix corresponding to each action $u \in \mathbf{U}$. As our actions are discrete, we do not need to specify any parametric model for conditional transition probabilities and directly

obtain them as statistics resulting from a counting process, as explained in section 6.4.2.

$$
\begin{aligned}
p(x_t|s_t, u_t = 1) &= \begin{cases} 1, & \text{if } x_t = s_t \\ 0, & \text{otherwise} \end{cases} \\
p(x_t|s_t, u_{t-1} = 2 \; or \; 3) &= \begin{cases} 1, & \text{if } x_t = s_t \\ 0, & \text{otherwise} \end{cases} \\
p(x_t|s_t, u_{t-1} = 0 \; or \; 1, u_t = 0) &= 1/|\mathbf{S}| \; \forall x_t = s_t
\end{aligned} \tag{6.2}
$$

6.4.2 Learning Algorithm for AS-IOHMM

In this section we describe the learning algorithm for the parameters of AS-IOHMM. The complete likelihood of AS-IOHMM, depends not only on the observed emission data $(\mathbf{X}_1^T, \mathbf{Y}_1^T)$ but also on the unobserved latent degradation state $(\mathbf{S})$ conditioned on the actions $(\mathbf{U}_1^T)$ taken by technicians at each time step. Thus maximizing the complete likelihood $\mathcal{L}_c(\Theta; \mathbf{X}_1^T, \mathbf{Y}_1^T, \mathbf{S}_1^T, \mathbf{U}_1^T)$ is troublesome due to the unobserved latent degradation states $(\mathbf{S}_1^T \in \mathcal{S})$. To overcome this issue, [16] proposed to maximize the auxiliary likelihood function $\mathcal{Q}(\Theta; \hat{\Theta}) = \mathbb{E}_{\mathcal{S}}[\mathcal{L}_c(\Theta; \mathbf{X}_1^T, \mathbf{Y}_1^T, \mathbf{S}, \mathbf{U}_1^T)|\mathbf{X}_1^T, \mathbf{Y}_1^T, \mathbf{U}_1^T, \hat{\Theta}]$ iteratively using Expectation-Maximization (EM) algorithm.

For a single system (equipment/unit), that is operated till time T, the complete data likelihood can be written as shown in equation 6.3. We decompose the complete data likelihood, by taking into considerations that condition on the given state and the type of action taken at time t or $t-1$ the emission variables are independent of each other. We also assume that the state variables follow one-step markov property, where state at time t depends on the state at time $t-1$ and the action taken at time $t-1$. In equation 6.3 we also note that $\mathcal{I}(i,t)$ are indicator variables that takes a value of 1 when $s_t = i$ otherwise its zero. Further

we also have, $\mathbb{E}_{\mathcal{S}}[\mathcal{I}(i,t)|\mathbf{X}_1^T, \mathbf{Y}_1^T, \mathbf{U}_1^T, \hat{\Theta}] = p(x_t = i|\mathbf{X}_1^T, \mathbf{Y}_1^T, \mathbf{U}_1^T, \hat{\Theta})$ as used in equation 6.4.

$$
\begin{aligned}
\mathcal{L}_c(\Theta; \mathbf{X}_1^T, \mathbf{Y}_1^T, \mathbf{S}_1^T, \mathbf{U}_1^T) &= p(\mathbf{X}_1^T, \mathbf{Y}_1^T, \mathbf{S}_1^T|\mathbf{U}_1^T; \Theta) \\
&= \prod_{t=1}^{T} p(x_t, y_t, s_t|s_{t-1}, u_{t-1}^t; \Theta) \\
&= \prod_{t=1}^{T} p(x_t|s_t, u_{t-1}^t; \Theta) p(y_t|s_t, u_{t-1}^t; \Theta) p(s_t|s_{t-1}, u_{t-1}; \Theta) \\
&= \prod_{t=1}^{T} \prod_{j=1}^{|\mathbf{S}|} [p(x_t|s_t = j, u_{t-1}^t; \Theta) p(y_t|s_t = j, u_{t-1}^t; \Theta)]^{\mathcal{I}(j,t)} \cdot \\
&\quad \prod_{i=1}^{|\mathbf{S}|} [p(s_t = j|s_{t-1} = i, u_{t-1}; \Theta)]^{\mathcal{I}(j,t), \mathcal{I}(i,t-1)} \\
log(L_c(\Theta; \mathbf{X}_1^T, \mathbf{Y}_1^T, \mathbf{S}_1^T, \mathbf{U}_1^T)) &= \sum_{t=1}^{T} \sum_{j=1}^{|\mathbf{S}|} \mathcal{I}(j,t)(log(p(x_t|s_t = j, u_{t-1}^t; \Theta)) + log(p(y_t|s_t = j, u_{t-1}^t; \Theta))) \\
&\quad + \sum_{i=1}^{|\mathbf{S}|} \mathcal{I}(j,t), \mathcal{I}(i,t-1) log(p(s_t = j|s_{t-1} = i, u_{t-1}; \Theta))
\end{aligned}
\tag{6.3}
$$

Next we describe the steps for the iterative Expectation-Maximization algorithm of [16]. The EM-algorithm starts by computing the expected value of $log(L_c(\Theta; \mathbf{X}_1^T, \mathbf{Y}_1^T, \mathbf{S}_1^T, \mathbf{U}_1^T))$ over the latent degradation states $\mathcal{S}$ given the observed data $\mathbf{X}_1^T, \mathbf{Y}_1^T, \mathbf{U}_1^T$ and an initial guess of model parameters $\hat{\Theta}$ as shown in equation 6.4. This Expected value helps us evaluate the auxiliary likelihood $\mathcal{Q}(\Theta; \hat{\Theta})$ for an initial guess of parameters. This is referred to as the E-Step. Using the notations defined in section 6.4 the Expected value of the complete log-

likelihood over latent states for the AS-IOHMM can be written as shown in equation 6.4.

$$
\begin{aligned}
\mathcal{Q}(\Theta;\hat{\Theta}) = \; & \mathbb{E}_{\mathcal{S}}[\mathcal{L}_c(\Theta;\mathbf{X}_1^T,\mathbf{Y}_1^T,\mathbf{S},\mathbf{U}_1^T)|\mathbf{X}_1^T,\mathbf{Y}_1^T,\mathbf{U}_1^T,\hat{\Theta}] \\
= & \sum_{t=1}^{T}\sum_{j=1}^{|\mathbf{S}|}\mathbb{E}_{\mathcal{S}}[\mathcal{I}(j,t)|\mathbf{X}_1^T,\mathbf{Y}_1^T,\mathbf{U}_1^T,\hat{\Theta}](log(p(x_t|s_t=j,u_{t-1}^t;\hat{\Theta}))+log(p(y_t|s_t=j,u_{t-1}^t;\hat{\Theta}))) \\
& +\sum_{i=1}^{|\mathbf{S}|}\mathbb{E}_{\mathcal{S}}[\mathcal{I}(j,t),\mathcal{I}(i,t-1)|\mathbf{X}_1^T,\mathbf{Y}_1^T,\mathbf{U}_1^T,\hat{\Theta}]log(p(s_t=j|s_{t-1}=i,u_{t-1};\hat{\Theta})) \\
= & \sum_{t=1}^{T}\sum_{j=1}^{|\mathbf{S}|}p(s_t=j|\mathbf{X}_1^T,\mathbf{Y}_1^T,\mathbf{U}_1^T,\hat{\Theta})(log(p(x_t|s_t=j,u_{t-1}^t;\hat{\Theta}))+log(p(y_t|s_t=j,u_{t-1}^t;\hat{\Theta}))) \\
& +\sum_{i=1}^{|\mathbf{S}|}p(s_t=j,s_{t-1}=i|\mathbf{X}_1^T,\mathbf{Y}_1^T,\mathbf{U}_1^T,\hat{\Theta})log(p(s_t=j|s_{t-1}=i,u_{t-1};\hat{\Theta})) \\
= & \sum_{t=1}^{T}\sum_{j=1}^{|\mathbf{S}|}\hat{\gamma_t(j)}(log(p(x_t|s_t=j,u_{t-1}^t;\hat{\Theta}))+log(p(y_t|s_t=j,u_{t-1}^t;\hat{\Theta}))) \\
& +\sum_{i=1}^{|\mathbf{S}|}\hat{\xi_{t-1}(i,j)}log(p(s_t=j|s_{t-1}=i,u_{t-1};\hat{\Theta}))
\end{aligned}
\tag{6.4}
$$

Note that in equation 6.4 the expected values depend on the conditional emission densities, which are evaluated from equations 6.1 and 6.2 by using initial guess values of the distribution parameters $\hat{\Theta}$. Note that the variable x is deterministic and free of any parameter. The term $p(s_t = j|s_{t-1} = i, u_{t-1};\hat{\Theta})$ represents an initial transition probability matrix and could be denoted as $\hat{a_{i,j}(u)}$. We propose a hyper-parameter tunning framework to tune the best values of the initial transition probabilities in section 6.5. The value of the variables $\hat{\gamma_t(j)}$, $\hat{\xi_{t-1}(i,j)}$ are calculated using the forward ($\alpha_t(i)$) and backward variables ($\beta_{t+1}(j)$). The recursive evaluation of the forward and the backward variables are shown in equation 6.5 and 6.6 respectively. Please note that we have used appropriate conditional independence

assumptions in factorizing the probabilities values in equations 6.5 and 6.6.

$$
\begin{aligned}
\textit{Initialize } \alpha_1(i) &= p(x_1|s_1 = i, u_1, \hat{\Theta})p(y_1|s_1 = i, u_1, \hat{\Theta})\pi(i) \\
\alpha_t(i) &= p(\mathbf{X}_2^t, \mathbf{Y}_2^t, s_t = i|\mathbf{U}_2^t, \hat{\Theta}) \\
&= \sum_{k=1}^{|\mathbf{S}|} p(\mathbf{X}_2^t, \mathbf{Y}_2^t, s_t = i, s_{t-1} = k|\mathbf{U}_2^t, \hat{\Theta}) \\
&= \sum_{k}^{|\mathbf{S}|} p(x_t, y_t|\mathbf{X}_2^{t-1}, \mathbf{Y}_2^{t-1}, s_t = i, s_{t-1} = k, \mathbf{U}_2^t, \hat{\Theta}) \\
&\quad \cdot p(s_t = i|s_{t-1} = k, \mathbf{X}_2^{t-1}, \mathbf{Y}_2^{t-1}, \mathbf{U}_2^{t-1}, \hat{\Theta}) \\
&\quad \cdot p(\mathbf{X}_2^{t-1}, \mathbf{Y}_2^{t-1}, s_{t-1} = k|\mathbf{U}_2^{t-1}, \hat{\Theta}) \\
&= p(x_t|s_t = i, u_{t-1}^t, \hat{\Theta})p(y_t|s_t = i, u_{t-1}^t, \hat{\Theta}) \sum_{k}^{|\mathbf{S}|} a_{k,i}(\hat{u}_{t-1})\alpha_{t-1}(k)
\end{aligned}
\tag{6.5}
$$

$$
\begin{aligned}
\textit{Initialize } \beta_T(j) &= 1 \\
\beta_{t+1}(j) &= p(\mathbf{X}_{t+2}^T, \mathbf{Y}_{t+2}^T|s_{t+2} = j, \mathbf{U}_{t+1}^T, \hat{\Theta}) \\
&= \sum_{k=1}^{|\mathbf{S}|} p(\mathbf{X}_{t+2}^T, \mathbf{Y}_{t+2}^T, s_{t+3} = k|s_{t+2} = j, \mathbf{U}_{t+1}^T, \hat{\Theta}) \\
&= \sum_{k}^{|\mathbf{S}|} p(x_{t+2}, y_{t+2}|\mathbf{X}_{t+3}^T, \mathbf{Y}_{t+3}^T, s_{t+2} = j, s_{t+3} = k, \mathbf{U}_{t+1}^T, \hat{\Theta}) \\
&\quad \cdot p(s_{t+3} = k|s_{t+2} = j, \mathbf{X}_{t+3}^T, \mathbf{Y}_{t+3}^T, \mathbf{U}_{t+1}^T, \hat{\Theta}) \\
&\quad \cdot p(\mathbf{X}_{t+3}^T, \mathbf{Y}_{t+3}^T|s_{t+3} = k, \mathbf{U}_{t+2}^T, \hat{\Theta}) \\
&= \sum_{k}^{|\mathbf{S}|} p(x_{t+2}|s_{t+2} = j, \mathbf{U}_{t+1}^{t+2}, \hat{\Theta})p(y_{t+2}|s_{t+2} = j, \mathbf{U}_{t+1}^{t+2}, \hat{\Theta})p(s_{t+3} = k|s_{t+2} = j, u_{t+2}, \hat{\Theta}) \\
&\quad \cdot p(\mathbf{X}_{t+3}^T, \mathbf{Y}_{t+3}^T|s_{t+3} = k, \mathbf{U}_{t+2}^T, \hat{\Theta}) \\
&= \sum_{k}^{|\mathbf{S}|} p(x_{t+2}|s_{t+2} = j, \mathbf{U}_{t+1}^{t+2}, \hat{\Theta})p(y_{t+2}|s_{t+2} = j, \mathbf{U}_{t+1}^{t+2}, \hat{\Theta})a_{j,k}(\hat{u}_{t+2})\beta_{t+2}(k)
\end{aligned}
\tag{6.6}
$$

We note that the forward variable help us to get the complete likelihood for the observed data when summed over all the possible states. i.e., $\mathcal{L}_o(\Theta; \mathbf{X}_1^T, \mathbf{Y}_1^T, \mathbf{U}_1^T) = p(\mathbf{X}_1^T, \mathbf{Y}_1^T, |\mathbf{U}_1^T, \hat{\Theta}) = \sum_{i=1}^{|\mathbf{S}|} p(\mathbf{X}_1^T, \mathbf{Y}_1^T, s_T = i|\mathbf{U}_1^T, \hat{\Theta}) = \sum_{i=1}^{|\mathbf{S}|} \alpha_T(i)$. The steps to evaluate $\hat{\gamma_t(j)}$, $\hat{\xi_{t-1}(i,j)}$ are shown in equations 6.7 and 6.8 respectively.

$$
\begin{aligned}
\xi_{t-1}(i,j) &= p(s_t = j, s_{t-1} = i|\mathbf{X}_1^T, \mathbf{Y}_1^T, \mathbf{U}_1^T, \hat{\Theta}) \\
&= p(s_t = j, s_{t-1} = i, \mathbf{X}_1^T, \mathbf{Y}_1^T|\mathbf{U}_1^T, \hat{\Theta})/p(\mathbf{X}_1^T, \mathbf{Y}_1^T|\mathbf{U}_1^T, \hat{\Theta}) \\
&= \{p(\mathbf{X}_{t+1}^T, \mathbf{Y}_{t+1}^T|s_t = j, s_{t-1} = i, \mathbf{X}_1^t, \mathbf{Y}_1^t, \mathbf{U}_1^T, \hat{\Theta}) \\
&\quad \cdot p(x_t|s_t = j, s_{t-1} = i, \mathbf{X}_1^{t-1}, \mathbf{Y}_1^{t-1}, \mathbf{U}_1^T, \hat{\Theta}) \\
&\quad \cdot p(y_t|s_t = j, s_{t-1} = i, \mathbf{X}_1^{t-1}, \mathbf{Y}_1^{t-1}, \mathbf{U}_1^T, \hat{\Theta}) \\
&\quad \cdot p(s_t = j|s_{t-1} = i, \mathbf{X}_1^{t-1}, \mathbf{Y}_1^{t-1}, \mathbf{U}_1^T, \hat{\Theta}) \cdot p(\mathbf{X}_1^{t-1}, \mathbf{Y}_1^{t-1}, s_{t-1} = i|\mathbf{U}_1^T, \hat{\Theta})\}/\mathcal{L}_o \\
&= \{\alpha_{t-1}(i)a_{i,j}(u_{t-1})p(x_t|s_t = j, u_{t-1}^t, , \hat{\Theta})p(y_t|s_t = j, u_{t-1}^t, , \hat{\Theta})\beta_t(j)\}/\mathcal{L}_o
\end{aligned}
\tag{6.7}
$$

$$
\begin{aligned}
\gamma_t(i) &= p(s_t = i|\mathbf{X}_1^T, \mathbf{Y}_1^T, \mathbf{U}_1^T, \hat{\Theta}) \\
&= p(s_t = i, \mathbf{X}_1^T, \mathbf{Y}_1^T|\mathbf{U}_1^T, \hat{\Theta})/p(\mathbf{X}_1^T, \mathbf{Y}_1^T|\mathbf{U}_1^T, \hat{\Theta}) \\
&= p(\mathbf{X}_{t+1}^T, \mathbf{Y}_{t+1}^T|s_t = i, \mathbf{X}_1^t, \mathbf{Y}_1^t, \mathbf{U}_1^T, \hat{\Theta})p(\mathbf{X}_1^t, \mathbf{Y}_1^t, s_t = i|\mathbf{U}_1^T, \hat{\Theta})/\mathcal{L}_o \\
&= p(\mathbf{X}_{t+1}^T, \mathbf{Y}_{t+1}^T|s_t = i, \mathbf{U}_t^T, \hat{\Theta})p(\mathbf{X}_1^t, \mathbf{Y}_1^t, s_t = i|\mathbf{U}_1^t, \hat{\Theta})/\mathcal{L}_o \\
&= \beta_t(i)\alpha_t(i)/\mathcal{L}_o
\end{aligned}
\tag{6.8}
$$

After evaluating the expected value of the Likelihood overall latent states (i.e., after evaluating the auxiliary function $mathcalQ(\Theta; \hat{\Theta})$) we have an estimate for $\hat{\alpha_t(i)}$, $\hat{\beta_t(j)}$, $\hat{\epsilon_{t-1}(i,j)}$ & $\hat{\gamma_t(i)}$. These estimates are then used in the Maximization step (M-step) of the EM algorithm to improve the value of the initial parameters. For AS-IOHMM, we need to optimize the parameters for the distribution of the continuous CBM signal y along with the estimates of the action-specific transition probability matrix $a_{i,j}(u) = p(s_t = j|s_{t-1} = i, u_{t-1}) \in \boldsymbol{A(i,j,u)}$.

From 6.4 the portion of the auxiliary function associated with the continuous CBM signal y that is to be maximized is given in equation 6.9, which we refer to as continuous signal likelihood $\mathcal{L}_y(\mathbf{Y}|\Theta')$ (where $\Theta' = \{\mu_j, \nu_j, \nu_u\} \subset \Theta \forall j \in \mathbf{S}$). To evaluate this value we introduce an Indicator function $\mathcal{I}(u_{t,t-1})$ which is set to 1 whenever $u_t =$ '1' or $u_{t-1} =$ '2'/ '3'. The usual steps to maximize the model's parameters involve differentiating the continuous signal likelihood w.r.t each parameter and then finding its root. However, as the continuous signal likelihood ($\mathcal{L}_y(\mathbf{Y})$) changes with each time step depending on the value of action u_{t-1}^t we do not have a close form solution for the optimal $\mathcal{L}_y(\mathbf{Y})$ parameters. Thus, we rely on Trust-Region-Algorithms [35] to numerically maximize the continuous signal likelihood and solve for each model parameter.

$$
\begin{aligned}
\mathcal{L}_y(\mathbf{Y}|\hat{\Theta}') &= \sum_{t=1}^{T}\sum_{j=1}^{|\mathbf{S}|} \hat{\gamma_t}(j)(log(p(y_t|s_t = j, u_{t-1}^t; \hat{\Theta}')) \\
&= \sum_{t=1}^{T}\sum_{j=1}^{|\mathbf{S}|} \hat{\gamma_t}(j) log\Big(\frac{1}{\sqrt{2\pi}}\{\sqrt{\nu_j + \mathcal{I}(u_{t,t-1})\nu_u}\}\exp\{-1/2(y_t - \mu_j)^2(\nu_j + \mathcal{I}(u_{t,t-1})\nu_u)\}\Big) \\
&= \sum_{t=1}^{T}\sum_{j=1}^{|\mathbf{S}|} \hat{\gamma_t}(j)\Big[-\frac{log(2\pi)}{2} + \frac{log(\nu_j + \mathcal{I}(u_{t,t-1})\nu_u)}{2} - \frac{(y_t - \mu_s)^2(\nu_s + \mathcal{I}(u_{t,t-1})\nu_u)}{2}\Big]
\end{aligned}
\tag{6.9}
$$

The numerical optimizer works more efficiently when they are provided with gradients of the likelihood functions w.r.t the parameters of optimizations. We compute the gradients for the continuous signal likelihood ($\mathcal{L}_y(\mathbf{Y}|\Theta')$) w.r.t each parameter in equation 6.10. It is reasonable to assume that the precision obtained in estimating a state when a technician judges the system state in person is higher than the precision obtained while inferring the states with the help of CB signals. Thus we have inequality constraints such that the precision parameter for action is greater than the precision parameter for each state $\nu_u > \nu_j \forall j \in \mathbf{S}$. The constraint optimization problem also helps to make the model identifiable by separating the precision parameters for action ν_u from the precision parameters of the states $\nu_j \forall j \in \mathbf{S}$. It is important to note that the Trust Region algorithms have different hyper-parameters (such

as trust radius, constraint penalty, barrier parameter, etc., $\in \Omega$) that are to be optimized for optimal convergence. In section 6.4.3, we visit the hyper-parameter tunning framework proposed in this work.

$$\begin{aligned}
\frac{\partial \mathcal{L}_y(\mathbf{Y}|\hat{\Theta}')}{\partial \mu_j} &= \sum_{t=1}^{T}\sum_{j=1}^{|\mathbf{S}|}\hat{\gamma_t(j)}\Big[-(y_t-\mu_s)(\nu_s+\mathcal{I}(u_{t,t-1})\nu_u)\Big] \\
\frac{\partial \mathcal{L}_y(\mathbf{Y}|\hat{\Theta}')}{\partial \nu_j} &= \sum_{t=1}^{T}\sum_{j=1}^{|\mathbf{S}|}\hat{\gamma_t(j)}\Big[-\frac{1}{2((\nu_j+\mathcal{I}(u_{t,t-1})\nu_u))}+\frac{(y_t-\mu_s)^2}{2}\Big] \\
\frac{\partial \mathcal{L}_y(\mathbf{Y}|\hat{\Theta}')}{\partial \nu_u} &= \sum_{t=1}^{T}\sum_{j=1}^{|\mathbf{S}|}\hat{\gamma_t(j)}\Big[-\frac{1}{2((\nu_j+\mathcal{I}(u_{t,t-1})\nu_u))}+\frac{(y_t-\mu_s)^2}{2}\Big]
\end{aligned} \tag{6.10}$$

The model parameters that are yet to be optimized are the conditional transition probabilities $a_{i,j}(u) = p(s_t = j|s_{t-1} = i, u_{t-1}) \in \boldsymbol{A(i,j,u)}$ and the initial probabilities π. We evaluate the probability of transition made from state i at time $t-1$ to state j at time t after taking an action u_{t-1} is given by evaluating the expected number of transitions made from $i \rightarrow j$ under action u_{t-1} and divide it by the expected number of transitions made from state i under action u_{t-1}. The variable $\xi_t(i,j)$ gives us the joint probability of system being in states $s_t = i,\ s_{t+1} = j$, while the variable $\gamma_t(i)$ gives us the probability of state i at time t. We let $\mathcal{I}(u_t = u)$ denote the indicator for taking action u at time t. Using this the transition probabilities corresponding to each action $u \in \mathbf{U}$ is given by equation 6.11. The initial probabilities are given as $\pi(i) = \gamma_1(i)$. To summarize, algorithm 10 gives the complete algorithm for learning parameters of the AS-IOHMM.

$$a_{i,j}(u) = \frac{\prod_{t=1}^{T}\{\mathcal{I}(u_t = u)\xi_t(i,j)\}}{\prod_{t=1}^{T}\{\mathcal{I}(u_t = u)\gamma_t(i)\}} \tag{6.11}$$

6.4.3 Optimize hyper-parameters for AS-IOHMM

We now describe the framework to optimize the hyper-parameters (Ω) essential for ensuring the convergence of the model. We note that the hyper-parameters of the Trust-Region

Algorithm 10: Expectation-Maximization Algorithm for AS-IOHMM

Input: $\mathbf{X_1^T}, \mathbf{Y_1^T}, \mathbf{U_1^T}, \hat{\pi(i)}, \hat{a_{i,j}}(u), \hat{\mu_j}, \hat{\nu_j}, \hat{\nu_u} \forall i, j \in \mathbf{S}$, $niter$, tol

`/* parameters with ^ represent the initial guess of the parameters */`

Output: $\bar{a_{i,j}}(u), \bar{pi(i)}, \bar{\mu_j}, \bar{\nu_j}, \bar{\nu_u} \forall i, j \in \mathbf{S}$ optimized parameters $\bar{\Theta}$

$\mathcal{L}'_o \leftarrow 0$
for $it \in 1 \rightarrow niter$ **do**
  Perform E-Step;
  for $t \in 1 \rightarrow \mathbf{T}$ **do**
    $\hat{\alpha_t(i)} \leftarrow$ from equation (6.5);
    $\hat{\beta_t(j)} \leftarrow$ from equation (6.6);
    $\hat{\xi_{t-1}(i,j)} \leftarrow$ from equation (6.7);
    $\hat{\gamma_t(i)} \leftarrow$ from equation (6.8);
  $\mathcal{L}_o \leftarrow \sum_{i=1}^{|\mathbf{S}|} \alpha_T i$
  Perform M-Step;
  $\mu_j, \nu_j, \nu_u \leftarrow argmax_{\mu_j,\nu_j,\nu_u}(\mathcal{L}_y(\mathbf{Y}|\hat{\Theta}))$ subject to $\nu_u > \nu_j \forall j \in \mathbf{S}$
  $a_{i,j}(u) \leftarrow$ from equation (6.11);
  $\pi(i) \leftarrow \hat{\gamma_0(i)}$
  $\Delta \leftarrow |\mathcal{L}'_o - \mathcal{L}_o|$
  $\mathcal{L}'_o \leftarrow \mathcal{L}_o$
  if $\Delta < tol$ **then**
    continue for loop
  else
    Break for loop

algorithm affect the convergence of the continuous signal likelihood $\mathcal{L}_y(\mathbf{Y}|\Theta')$. Further, to begin the EM algorithm, an initial guess of the transition probabilities $\hat{a_{i,j}}(u)$ is also needed, towards which researchers usually spend some considerable effort [38]. However, in this work, we cast initial transition probabilities as hyper-parameters optimized simultaneously along with the hyper-parameters of the Trust-Region algorithm. The method is applicable not only for the cases where we have a long sequence of single units but also for the case where we have short sequences of multiple identical units to train the AS-IOHMM. Please note that all the equations and algorithms described in section 6.4 are easily extended to the case of having multiple identical units by simply summing over the data of different units. For example, the equation 6.3 with multiple units takes the form as shown in equation 6.12.

Similar updates could be made for equations 6.8, 6.9, and 6.10 to train the parameters using all the available sequences.

$$\mathcal{Q}(\Theta;\hat{\Theta}) = \sum_{n=1}^{N}\sum_{t=1}^{T}\sum_{j=1}^{|\mathbf{S}|}\hat{\gamma_t^n}(j)(log(p(x_t^n|s_t=j,(u_{t-1}^t)^n;\hat{\Theta})) + log(p(y_t^n|s_t=j,(u_{t-1}^t)^n;\hat{\Theta})))$$
$$+\sum_{i=1}^{|\mathbf{S}|}\hat{\xi_{t-1}^n}(i,j)log(p(s_t=j|s_{t-1}=i,u_{t-1}^n;\hat{\Theta})) \tag{6.12}$$

For the hyper-parameter tunning framework, we use the fact that maintenance actions like inspection or repair help us realize the true system states. Thus, we could use the Viterbi-Algorithm [168], which helps to find us the optimal sequence of states $\bar{S_1^T}$ that maximize the joint probability of the observed data and the latent states conditioned on the actions-taken $p(\mathbf{X_1^T},\mathbf{Y_1^T},\mathbf{S_1^T}|\mathbf{U_1^T})$. To train the hyper-parameters for AS-IOHMM, we use the usual Machine Learning pipeline of splitting the data into Training-Validation and Test parts. The Test part is not needed for tunning hyper-parameters but can be used to identify the model's overall accuracy. To begin tunning the hyper-parameters, we sample the hyper-parameters using the Tree-structured Parzen Estimator model [19], which sample hyper-parameters sequentially. We then train the AS-IOHMM on the training data using the sampled hyper-parameters and predict the optimal state sequence for the validation part. Using the states revealed by the technicians during maintenance actions as ground truth, we evaluate the accuracy of the optimal states predicted by AS-IOHMM for the validation data under the sampled hyper-parameters. The TPE sampler then samples other hyper-parameter sets and identifies the ones with the highest Expected Improvement.

The Viterbi-Algorithm uses a two-dimensional array $\delta_t(i)$ of size $[T,S]$ to store the highest probability of the best path that ends in state i at time t. It also consists of a pointer variable $psi_t(i)$, which stores the states traversed by that best path to reach state i at time t. This pointer variable $\psi_t(i)$ is then backtracked by starting from the state with the highest value

of $\delta_T(i)$ to evaluate the optimal state sequences. The variable $\delta_t(i)$ is similar to the forward variable and is evaluated sequentially. Hence, if we have a long sequence of single units available to us, we have to split it into Training, Validation, and Test parts sequentially, where the first part is used for training the model, the second part is used for validation, and the last part is for testing the model. If, however, we have multiple sequences from N-identical systems, we use the N-1 approach to generate training and validation data, i.e., the first N-1 sequences form the training set, we also allow some initial samples of the N_{th} sequence to be in the training set and the remaining samples from the validation set which are then followed by the Test set sequentially. The hyper-parameter tunning framework used in this work is given in algorithm 11 for the case when we have N short sequences from N identical systems.

Algorithm 11: Hyper-Parameter Tunning Algorithm for AS-IOHMM

Input: $[\mathbf{X_1^T}]_1^N, [\mathbf{Y_1^T}]_1^N, [\mathbf{U_1^T}]_1^N,\ iter$

Output: $\bar{\omega}$ optimized hyper-parameters

$Train \leftarrow [[[\mathbf{X_1^T}]_1^{N-1}, \mathbf{X_1^{t_1}}(N)], [[\mathbf{Y_1^T}]_1^{N-1}, \mathbf{Y_1^{t_1}}(N)], [[\mathbf{U_1^T}]_1^{N-1}, \mathbf{U_1^{t_1}}(N)]]$;

$Valid \leftarrow [\mathbf{X_1^{t_2}}(N), \mathbf{Y_1^{t_2}}(N), \mathbf{U_1^{t_2}}(N)]$;

/* $t_1 < t_2$, $t_2 - t_1$ represent the validation data samples */

;

$Test \leftarrow [\mathbf{X_1^T}(N), \mathbf{Y_1^T}(N), \mathbf{U_1^T}(N)]$;

for $it \in 1 \rightarrow iter$ **do**

  Sample $\omega_{it} \in \Omega$;

  $\bar{\Theta} \leftarrow$EM-Algorithm10$(Train, \omega_{it})$;

  $\tilde{\mathbf{S_1^{t_2}}}(N) \leftarrow Viterbi - Algorithm(Valid, \bar{\Theta})$;

  /* Where $\tilde{\mathbf{S_1^{t_2}}}(N)$ represent the Optimal state sequence predicted by the Viterbi-Algorithm using optimized model parameters $\bar{\Theta}$ from the trained AS-IOHMM and the validation dataset */

  Let $\mathbf{X_1^{*t_2}}(N) \subset \mathbf{X_1^{t_2}}(N)\ \forall\ t \in \{1, t_2\}\ \ni\ x_t(N) \neq missing$;

  Define $\tilde{\mathbf{S_1^{*t_2}}}(N) \subset \tilde{\mathbf{S_1^{t_2}}}(N) \forall\ t \in \{1, t_2\}\ \ni\ x_t(N) \neq missing$;

  Validation-Accuracy $= acc_{it} = \frac{\sum \mathcal{I}(x_t^*(N) = \tilde{s_t^*})}{|\mathbf{X_1^{*t_2}}(N)|} \forall\ t \in \{1, t_2\}\ \ni\ x_t(N) \neq missing$;

  $\mathbf{Acc} \cup acc_{it};\ \Omega \cup \omega_{it}$;

$\bar{\omega} \leftarrow \Omega(argmax(acc_{it}))$

6.5 Numerical Study

In this section, we simulate CBM signals using the generative model framework explained in section 6.4. We describe the effects of different components involved in the AS-IOHMM model and demonstrate their usefulness in two different simulation scenarios. We compare the result of using the full AS-IOHMM (proposed) model with three cases. In the first case, while training the model, we do not have the information about states from maintenance actions (i.e., variable $\mathbf{X}$ is not present). Thus we do not have increased precision, and the extra knowledge about states is lost while we train the model. We refer to this case as a No state information (NSI) case in the following results and graphs. In the second case, we even remove the information about different actions taken, thus restricting the transition probabilities to depend on actions taken, reducing the model to simple HMM with a 2-Dimensional transition probability matrix. This case is a No state and action information (NSAI) case. We also benchmark our proposed model against the Gaussian Hidden Markov Model (GHMM) (the assumptions of which are similar to the NSAI case, but emissions are modeled Gaussian distribution with variance as a scale parameter) and Gaussian Mixture Hidden Markov model (GMHMM) [110], [97]. We also wish to indicate that other Discriminative supervised learning models need labels for training. However, the number of maintenance actions taken on the field will be sparse. Thus discriminative supervised models cannot be used to model the system's dynamics considered in this chapter. Further, these models do not provide any insights about state transition, thus limiting the applicability of the proposed model to any kind of Decision Making.

To simulate data, we choose the transition probability matrix (TPM) corresponding to the four actions of Do-Nothing:0, Inspection:1, Preventive Maintenance:2, and Corrective Maintenance:3, as shown in the array 6.13. The TPM for Do-Nothing and Inspection is upper-triangular to represent the standard Failure Process TPM [38]. For emission signals, we simulate two different scenarios to get some insights into the performance of AS-IOHMM.

In the first Scenario (Scn:1), we allow the parameters of the continuous signal to be well separated. We simulate data for three latent states considering good:1, neutral:2 and bad:3 states of the system $\mu_1 = 3, \mu_2 = 6, \mu_3 = 9, \nu_1 = 0.55, \nu_2 = 0.85, \nu_3 = 1.05, \nu_u = 1.25$ and assume the initial probabilities to be $\pi(1) = 0.8, \pi(2) = 0.1, \pi(3) = 0.1$. For the second Scenario (Scn:2), the mean of CBM signals corresponding to each state are not well separated. Also, the precision associated with each state is low to allow the wider spread of the signals. We keep the parameters for the CBM signal for the second Scenario as $\mu_1 = 85, \mu_2 = 92, \mu_3 = 98, \nu_1 = 0.055, \nu_2 = 0.065, \nu_3 = 0.04, \nu_u = 0.08$. For each Scenario, we first simulate latent states and actions. We assume the actions taken by a technician follow a random policy such that if the state at a given time step is 'good:1', then technicians sample an action from the probability vector $\{0 : 0.7,\ 1 : 0.1,\ 2 : 0.1,\ 3 : 0.1\}$ (where the first number represent the type of action chosen). If the state is 'neutral:2' then the probability of actions are $\{0 : 0.3,\ 1 : 0.3,\ 2 : 0.3,\ 3 : 0.1\}$, wherelse if the state is 'bad:3' then the probability vectors looks like $\{0 : 0.2,\ 1 : 0.15,\ 2 : 0.25,\ 3 : 0.4\}$. Based on the previous states and action taken, the next state is sampled randomly from the TPM 6.13.

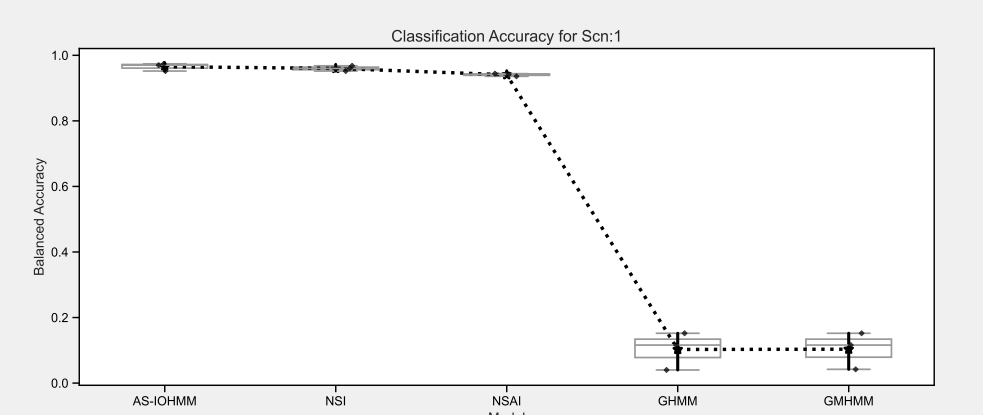
$$\begin{bmatrix} \textbf{Do-Nothing} & \textbf{Inspection} & \textbf{Preventive Mnt.} & \textbf{Corrective Mnt.} \\ \begin{bmatrix} 0.3 & 0.5 & 0.2 \\ 0 & 0.45 & 0.55 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0.3 & 0.5 & 0.2 \\ 0 & 0.45 & 0.55 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0.8 & 0.15 & 0.05 \\ 0.6 & 0.3 & 0.1 \\ 0.3 & 0.4 & 0.3 \end{bmatrix} & \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.7 & 0.2 & 0.1 \\ 0.6 & 0.25 & 0.15 \end{bmatrix} \end{bmatrix} \tag{6.13}$$

Using the simulated states and actions, continuous signals y and discrete x state indicators are sampled following equations 6.1 and 6.2. For each Scenario, we simulate two sequences of lengths 1200 and 1000. For both Scenarios, we create the training dataset using the complete first sequence (of length 1200). We split the second sequence of length 1000 as $\{250 - 250 - 500\}$, using the first 250 samples for training along with the first sequence. We use the successive 250 samples from the second sequence for our validation dataset and keep the remaining 500 samples for the test

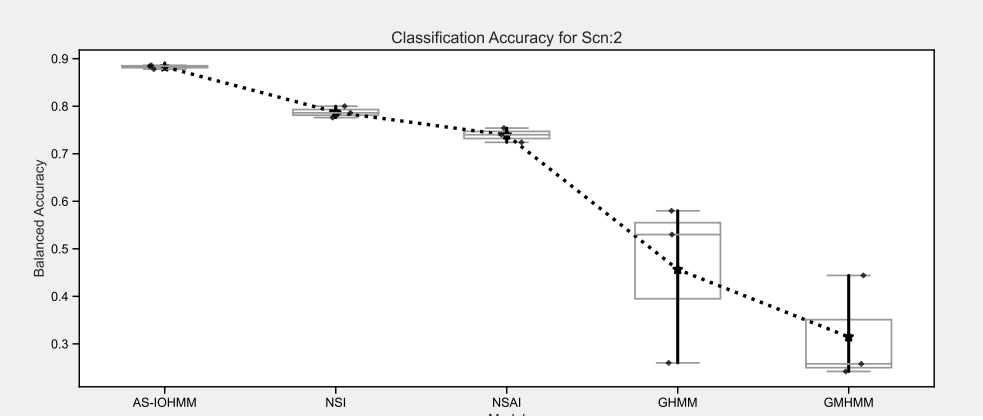
set. We use the TPE sampler from Optuna [3] to tune the hyper-parameters using the framework demonstrated in algorithm 11. After optimizing the hyper-parameters, the final model is trained using them, which is then used to generate optimal state sequences using Viterbi-Algorithm for the complete second sequence. For simulated data, as we have the actual state values for all the samples in the Test set, we use them along with the predicted optimal state sequence to calculate the balanced classification accuracy for the model. Please note that in reality, we would not have the state value for all the data in the Test set, and we only need to rely on the samples where the discrete state variable x_t is not missing for model evaluation.

We show the classification accuracy for both Scenarios for all five model types in Figure 6.3. We repeat the entire numerical study thrice for both Scenarios by generating different samples under the same distributional assumptions. The Box-Plot generated in Figure 6.3 shows the variability in the prediction made by all the models. We find that for Scenario:1, where the continuous signals for each state are well separated from each other, all the variants of the proposed AS-IOHMM model produce really good results with a mean accuracy of the full AS-IOHMM being 0.965, that of NSI is 0.96 while that of the NASI being 0.94. Although the best accuracy is achieved for the full AS-IOHMM model, we could still conclude that there is not much effect of removing the additional inspection information and the action condition transition probabilities when we perform the classification task for signals that are well separated from each other conditioned on their latent states. However, when we look at the mean accuracy of the models in scenario 2, where continuous signals are not well separated from each other, we find the utility of each component of the AS-IOHMM model. In this case, the AS-IOHMM model's mean accuracy is 0.88. However, as soon as we remove the information provided by the Discrete state emission variable x, the accuracy drops to 0.79 for the NSI model. Further, when we remove the conditional action-specific transition probability assumption, in that case, the mean accuracy drops to 0.739 for the NSAI model emphasizing the utility of the various components of the proposed AS-IOHMM. The existing GHMM and MGHMM models' accuracy is pretty low for both simulation Scenarios.

We show the Transition Probability matrix learned by the best of three performing AS-IOHMM models for both simulation Scenarios. Although both TPMs look good for both Scenarios, we observe that the TPM for scenario 1 is in much better agreement with the original TPM. This is



(a) Comparison of Balanced Accuracy for Scenario 1 over Test Data



(b) Comparison of Balanced Accuracy for Scenario 2 over Test Data

Figure 6.3: Balanced Accuracy over Test Data predicted by each model for both simulation scenarios

because the emission signals for Scenario:1 were separated in means from that of Scenario:2. The model parameters for the continuous model density for scenario 1 for the best of three AS-IOHMM are $\mu_1 = 3.02, \mu_2 = 6.02, \mu_3 = 9.02, \nu_1 = 0.568, \nu_2 = 0.512, \nu_3 = 1.053, \nu_u = 1.44$ which are very close to the actual values. Similarly, the model parameters for the continuous model density for scenario 2 for the best of three AS-IOHMM are $\mu_1 = 85.39, \mu_2 = 92.0, \mu_3 = 97.82, \nu_1 = 0.0565, \nu_2 = 0.0417, \nu_3 = 0.0386, \nu_u = 0.0865$ which again are very close to the actual values. The closeness of the learned parameters with the actual values demonstrates the effectiveness of the AS-IOHMM framework along with the prowess of the hyper-parameter tuning algorithm 11.

$$
\begin{bmatrix}
Scen & \textbf{Do-Nothing} & \textbf{Inspection} & \textbf{Preventive Mnt.} & \textbf{Corrective Mnt.} \\
Scn:1 & \begin{bmatrix} 0.30 & 0.54 & 0.16 \\ 0 & 0.48 & 0.52 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0.31 & 0.51 & 0.17 \\ 0 & 0.5 & 0.5 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0.9 & 0.074 & 0.03 \\ 0.604 & 0.294 & 0.102 \\ 0.359 & 0.375 & 0.264 \end{bmatrix} & \begin{bmatrix} 0.905 & 0.020 & 0.075 \\ 0.698 & 0.224 & 0.076 \\ 0.665 & 0.197 & 0.138 \end{bmatrix} \\
Scn:2 & \begin{bmatrix} 0.24 & 0.56 & 0.2 \\ 0 & 0.56 & 0.44 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0.39 & 0.54 & 0.07 \\ 0 & 0.43 & 0.57 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0.861 & 0.138 & 0.00 \\ 0.545 & 0.297 & 0.158 \\ 0.313 & 0.32 & 0.368 \end{bmatrix} & \begin{bmatrix} 0.882 & 0.0 & 0.118 \\ 0.684 & 0.228 & 0.087 \\ 0.641 & 0.231 & 0.128 \end{bmatrix}
\end{bmatrix} \tag{6.14}
$$

6.6 Industry Case Study

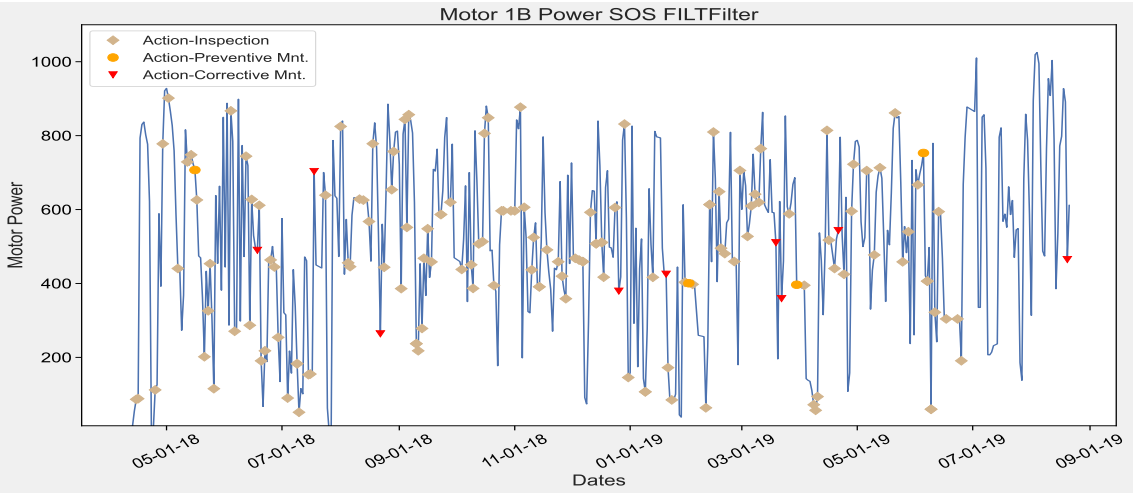
In this section, we discuss applying the proposed model to industrial data. We have data from two different Duplex Mud-Pumps from an Oil Rig. Each Mud-Pump has two identical motors installed on them, and power extracted by each mud-pump motor is recorded as a CBM signal. Thus we have data from two identical units (motor 1A/1B and 2A/2B) of two different Mud-Pumps MP-1 and MP-2. The descriptive statistics of the CBM signal recorded for each motor are shown in Table 6.2. The descriptive statistics of CBM signal for motors in a given Mud-Pump are very close to each other. Further, both the motors of a given Mud-Pump operate in the same environment; thus, we consider motors from the same Mud-Pump identical. Next, we apply some pre-processing steps to remove the noise from the CBM signal. We only select those segments of signals during which motors were on. Then for each segment, we first compute the power spectral density of the CBM signal using Welch Method [176] to identify the frequency component that contains the maximum signal power. We then use this frequency to obtain third-order Butterworth Filter coefficients used in a forward-backward digital filter [123]. We then merge the processed segments back to form the single sequence and then down-sample it to merge it with the events file. The event file contains information about maintenance actions and the generated maintenance records, which help us identify the motor's health state directly.

Table 6.2: Descriptive Statistics of CBM signal recorded for Motors A and B of Mud Pump 1 and Mud Pump 2

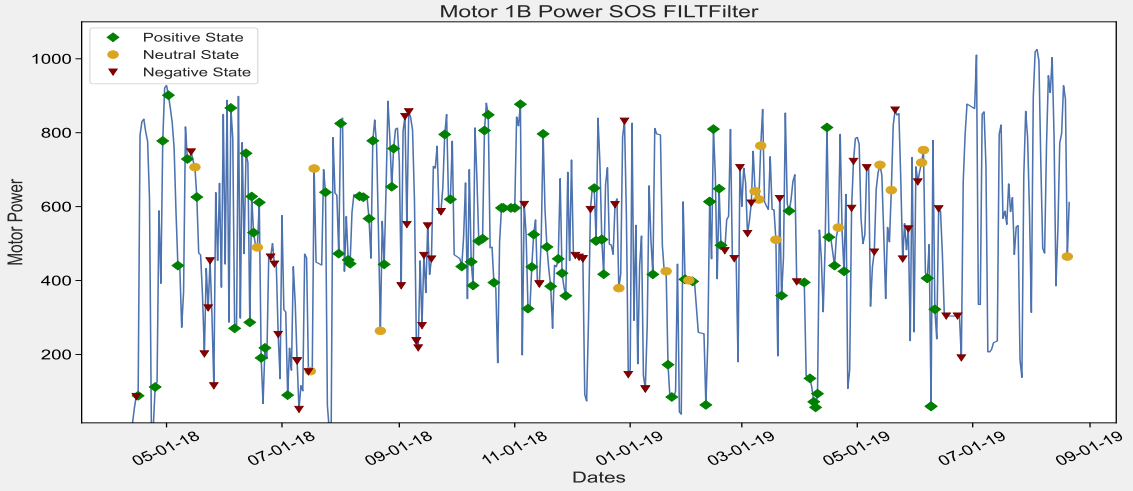
Statistic	Mtr-1A Power[kW]	Mtr-1B Power[kW]	Motor-2A Drive Power[kW]	Motor-2B Drive Power[kW]
Mean	686.58	683.98	260.37	258.09
Std. Dev	246.431	246.599	161.43	159.45
Minimum	-30.21	-42.66	-69.09	-76.16
1st Quartile	537.05	535.58	105.60	105.21
2nd Quartile	765.75	763.18	258.70	257.13
3rd Quartile	874.15	871.58	414.15	410.62
Maximum	1054.69	1053.41	644.97	641.05

Figure 6.4 shows the final merged data for motor 1B of MP-1. As seen from Figure 6.4, the number of events where inspections are made is very high compared to preventive or corrective maintenance. Similar data is obtained for motor 1A, from Mud Pump (MP)-1 along with motor 2A, 2B of MP-2. To demonstrate the capabilities of the proposed model, we repeat a similar exercise as done in section 6.5. We generate two scenarios of prediction results for each model corresponding to each Mud Pump, where we train the models on the data of Motors indexed by A's and divide the data of motors indexed by B's into validation and testing parts. For Scenario-1, we train the models using complete data for motor-1A, validate the models using the first half of data for motor 1B and evaluate accuracy by predicting the optimal state sequence for the remaining data of motor-1B. While generating the classification accuracy, we use only those test data points for which the states are inferred by technicians as ground truth values.

To begin training the model, we initialize the parameters for the continuous signal likelihood 6.9 using the sample mean and precision for each inferred state. The accuracy achieved by modeling the motors for both the Mud-Pumps is shown in Figure 6.5. For both the Mud-Pumps motors, we observe that the proposed AS-IOHMM model achieves the best accuracy, and the accuracy reduces as we remove information about inferred states and actions. The accuracy of the Gaussian HMM and Mixture of Gaussian HMM is poor compared to the proposed model. An important thing to note is that the consideration of increased precision in modeling the continuous signal density via state information received by manual actions has essential effects in impacting the high accuracy of the proposed AS-IOHMM. Further, we see a similar trend as of section 6.5 that CBM



(a) Motor power plot with different actions taken by technicians



(b) Motor power plot with different states inferred by technicians

Figure 6.4: Plotting CBM signals with different actions and states

signals of motors of MP-2 are much better separable than the CBM signals of MP-1, and hence we observe an increased accuracy in predictions made by AS-IOHMM for MP-2 over MP-1. We also show the transition probability matrix (TPM) learned by the proposed AS-IOHMM model for the Mud-Pumps motors in 6.15. The extreme probabilities of the TPM in 6.15 result from sparse data points collected for a particular kind of action. Further, we observe that for actions where the technicians make no manual intervention, the motor continuously degrades from a good to a bad health state which is the trend for a general failure transition matrix [38]. The TPM corresponding to repair actions mainly follows a lower triangular trend, which means that a repair mostly helps revive the system's health. However, sometimes we do observe a high transition

probability of moving from a good to a bad health state under maintenance action, suggesting that the repairs done sometimes are not perfect. We conclude this section by noting the continuous density parameters learned by the AS-IOHMM model for the motors of both the Mud-Pumps are motor-1:- $\{\mu_{s_1} : 450.9, \mu_{s_2} : 453.2, \mu_{s_3} : 690.1, \nu_{s_1} : 0.0045, \nu_{s_2} : 0.00502, \nu_{s_3} : 0.0082, \nu_u : 0.1\}$; motor-2:- $\{\mu_{s_1} : 105.5, \mu_{s_2} : 129.87, \mu_{s_3} : 351.2, \nu_{s_1} : 0.0102, \nu_{s_2} : 0.0086, \nu_{s_3} : 0.0083, \nu_u : 0.051\}$.

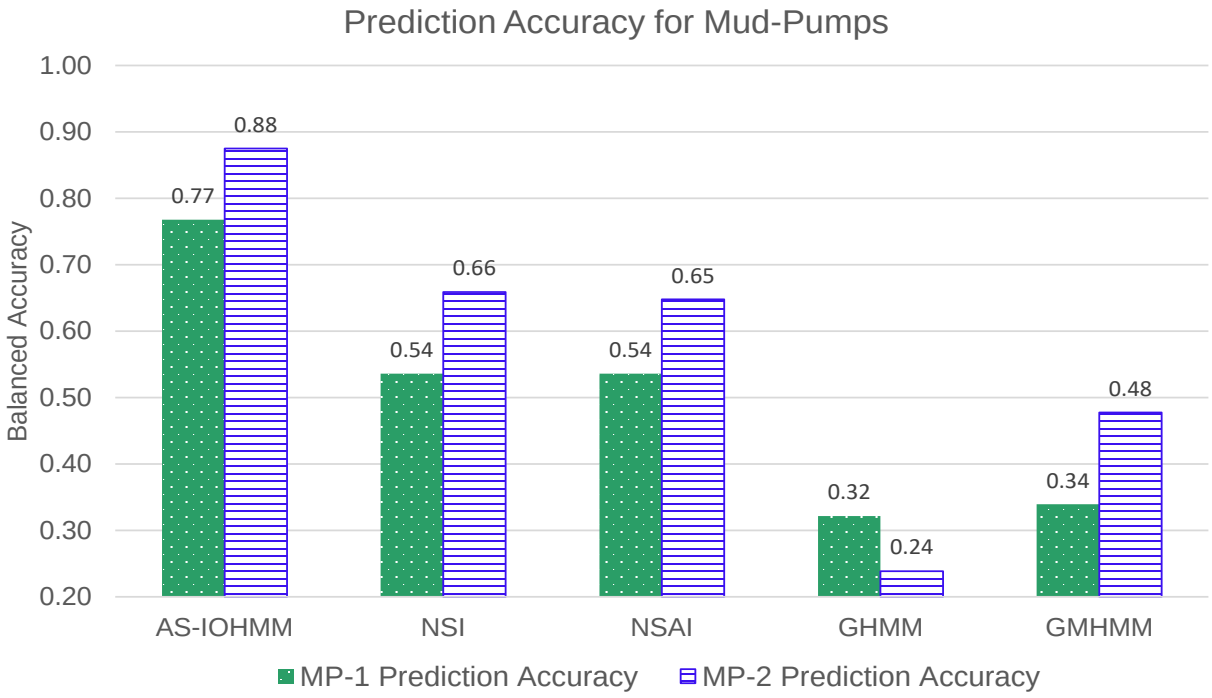


Figure 6.5: Classification accuracy of both Mud-Pump motors by using all the models

.

$$\begin{bmatrix} MP & \textbf{Do-Nothing} & \textbf{Inspection} & \textbf{Preventive Mnt.} & \textbf{Corrective Mnt.} \\ MP1 & \begin{bmatrix} 0.925 & 0.057 & 0.018 \\ 0 & 0.942 & 0.058 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0.99 & 0.01 & 0 \\ 0 & 0.9 & 0.1 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0.67 & 0.33 & 0.0 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0.67 & 0.33 & 0 \\ 0.75 & 0.125 & 0.125 \end{bmatrix} \\ MP2 & \begin{bmatrix} 0.958 & 0.03 & 0.012 \\ 0 & 0.98 & 0.02 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0.951 & 0.037 & 0.012 \\ 0 & 0.913 & 0.087 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0.8 & 0.2 & 0 \\ 0.2 & 0.4 & 0.4 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0.45 & 0.19 & 0.36 \end{bmatrix} \end{bmatrix} \tag{6.15}$$

6.7 Future Work and Conclusion

This work demonstrates the importance of incorporating information from maintenance records and CBM sensory signals to aid prognosis of equipment health. The joint modeling approach presented in this work introduces a novel way of separately identifying the transitions performed by a system under different actions, which are typically averaged out in current HMM models. Thus, we can distinguish the transition dynamics followed by the system under each action taken. Through numerical studies and application to real-world industry data, we demonstrate that incorporating information gained from manual maintenance actions helps increase the precision of the continuous-time model for inferring the system states. We propose a hyperparameter tuning framework to optimize the hyperparameters of the numerical solvers and solve the issue of selecting the initial transition probabilities which is crucial for model convergence.

The proposed model achieves the highest accuracy among existing models in numerical studies, emphasizing the significance of each model component. One of the key advantages of the current model is its simplicity in developing model-based approximate dynamic programs, which can support decision-makers in optimally scheduling maintenance and inspection activities. A natural extension of the current work involves extending the discrete-time HMM structure to a continuous time Markov chains. Furthermore, incorporating the Semi-Markov structure in the current framework could enhance the utility of the proposed model.

Chapter 7

Summary and Future Work

This thesis demonstrates how unstructured text data helps improve the failure diagnosis of complex equipment and identify their health status. Failure diagnosis itself provides valuable insights for Original Equipment Manufacturers (OEMs) and equipment owners, enabling them to enhance machine designs, warranty policies, spare part inventory management, and more. Specifically, the research conducted in chapters 2 to 4 focuses on extracting in-depth knowledge about failures occurring in industrial equipment, while chapter 5 introduces a model for identifying equipment health status. The key input in these chapters is the unstructured maintenance records generated by technicians during inspections and repairs of industrial equipment. The proposed unsupervised methods in handling these voluminous unstructured textual data address various challenges, including semantic ambiguity, domain-specific vocabulary, and negation of document sentiments, without relying on labeled data. In Chapter 6, we demonstrate the fusion of information extracted from unstructured maintenance records with Condition-Based Maintenance (CBM) sensory signals by proposing a novel joint modeling approach called AS-IOHMM. This approach integrates CBM sensory signals with direct state observations inferred from manual interventions, enabling separate modeling of system dynamics for each maintenance action undertaken by technicians. Figure 7.1 illustrates the contributions made to date and outlines future work.

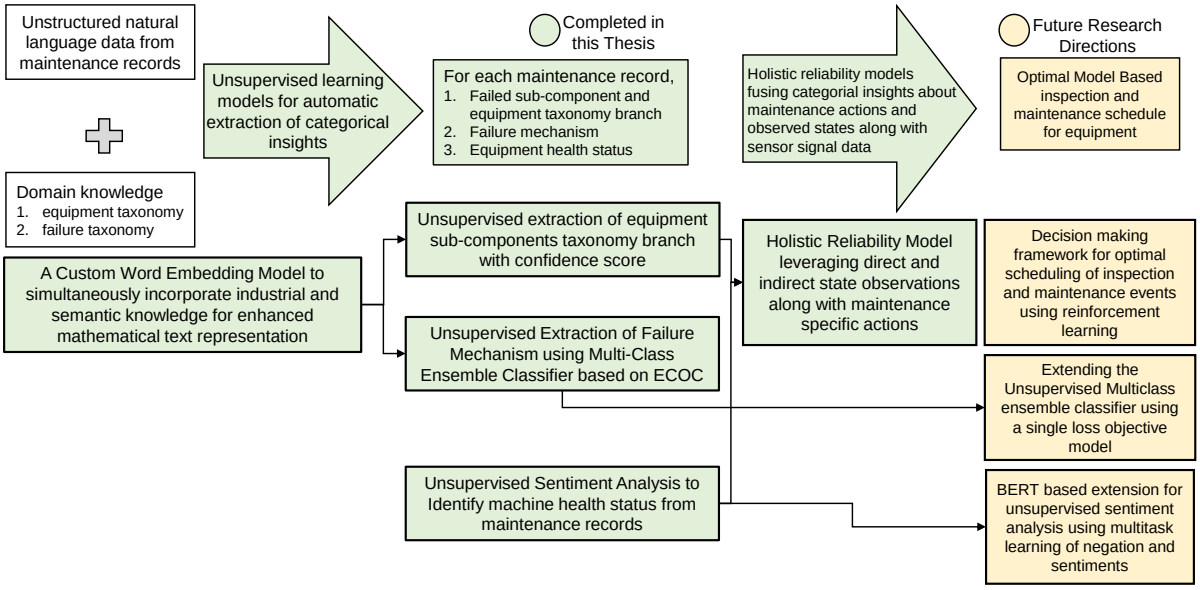


Figure 7.1: Research to date and future work

7.1 Research work to date and summary of contributions

The contributions of the research described in chapter 2 - chapter 6 can be summarized as follows:

1. Generation of custom word embeddings for industrial maintenance records:
   In **Chapter-2**, we proposed a novel custom word embedding model to generate numerically distributed representation of maintenance records by combining information from two sources namely semantic information and taxonomic information. Our method combines the skip-gram model with standard industrial taxonomy to efficiently learn the word distribution, by a weighting parameter α that control the degree of influence exerted by semantic and taxonomic source of information. The CWEM word embeddings outperforms embeddings generated using other existing models for clustering industrial records. By simultaneously incorporating the taxonomic and semantic information, there is no need for contextually associated words to co-occur in a document (as needed by other existing models). The single step learning of model parameters reduces the number of hyper-parameters and the model is learnt without any supervision or third party dependencies.

2. Extracting complete taxonomy branch of failed sub-components of hierarchical equipment

with confidence score:

In **Chapter-3**, we tackle an important aspect of failure diagnosis by developing a comprehensive methodology to automatically identify the equipment taxonomy branch related to a maintenance record documented while inspecting an equipment breakdown. The methodology proposed leverages OEM's equipment taxonomy. We devise two algorithms, namely 1) Backward-Forward and 2) Verb Analysis, that are based on syntactic (frequentist) rule-based methods in NLP. To tackle the limitations of frequentist methods, semantic information is incorporated by leveraging word embeddings to generate a non-parametric density curve that provides confidence score for the extracted taxonomy branch.

3. Identifying failure mechanism from unstructured maintenance records using unsupervised multi-class ensemble classification model:

   In **Chapter-4**, we present a solution framework for unsupervised multi-class ensemble classification problems using continuous scores from multiple base classifiers. We also study the issues of class imbalance and feature/label noise in the dataset that effect the classifiers performance. To develop the model, we incorporate the use of muti-class to binary class decomposition techniques like OVA and ECOC and illustrate the effectiveness of order statistics like maximums in ensuring optimal results. We provide an extensive empirical study and demonstrate the effectiveness of the proposed method over real world case study data.

4. Determining equipment health status from maintenance records using lexicon based unsupervised sentiment analysis model:

   In **Chapter-5**, we present a framework to estimate equipment health status from unstructured maintenanace records. We illustrate that the existing transfer learning models like BERT fail to adapt to domains which are highly distinct from each other and thus there is a need to use lexicon based methods for unsupervised estimation of equipment health status. The existing lexicons used in sentiment analysis are not suitable for industrial data and hence we propose a list of new seed lexicons suitable for industrial use. We also address the major issue of negation found extensively in industrial records and demonstrate the superiority of our proposed method over other topic models existing in literature.

5. Holistic Condition Based Maintenance Model by jointly modeling CBM sensory signal and direct state observations for system subjected to different maintenance actions:
   In **Chapter-6**, we present a novel joint modeling approach that facilitates the modeling of continuous sensor signals alongside a discrete state information variable with missing values. This occurs during manual actions performed for a system that is jointly maintained using Condition-Based Maintenance (CBM) and manual interventions. The proposed model enables the representation of changes in system dynamics resulting from different actions performed on it while it is operational. To estimate the model parameters, we employ the Expectation-Maximization (EM) algorithm, leveraging a numerical optimizer to maximize the likelihood of the model parameters. Since the accuracy of the numerical optimizer and EM algorithm relies on the precision of hyperparameters and the initialization of model parameters, which are unavailable for a new problem, we propose routines that allow users to search for optimal hyperparameters and initialize model parameters by framing the problem as a hyperparameter tuning framework.

7.2 Future work

This thesis demonstrates the utility of unstructured natural language data not only in failure diagnosis but also in improving reliability models for studying changing dynamics of a system subjected under different manual actions. Although, an unsupervised lexicon based method to extract information about equipment health status is provided in chapter 5, the complete power of transfer learning models like BERT still remains to be tested. Transfer learning models like BERT are claimed to be highly generalized for domain adaptation, however, as shown in chapter 5 this is not the case. Methods to increase the generalization of BERT for domain adaptation include pre-training on the target domain data. Apart from this, multi task fine-tuning of BERT for negation and sentiments on source data may also help increase the current accuracy of the BERT model achieved in chapter 5. Thus, as future research direction I plan to explore a new multi-task learning model using transformer models like BERT pre-trained on target domain data for identifying equipment health status. Further, the UMEC model proposed in chapter 4 can also potentially be extended by proposing a single objective classifier using non-parametric density

representation for scores of different classifiers.

As described in Chapter 6, industries rely on a dual maintenance strategy in which they deploy both Condition Based Maintenance (CBM) sensory signals to perceive equipment state and schedule calendar-based manual maintenance activities for inspecting and repairing the equipment. Manual inspections help technicians gain direct observations about the system's health condition; however, they incur additional costs due to manual labor. On the other hand, CBM monitoring is relatively inexpensive but provides indirect inference about equipment health conditions. Other manual maintenance actions, such as preventive and corrective maintenance, also come with high costs but are unavoidable in many instances. Therefore, there is a need to develop a decision-making framework to optimize the schedules of various manual maintenance actions, including inspection, preventive maintenance, and corrective maintenance, in order to minimize long-term expenditures.

We plan to develop a decision-making framework using the Approximate Policy Iteration (API) algorithm for the Model-Based Partially Observed Markov Decision Process (POMDP). Chapter 6 of this thesis provides us with the environment model, justifying the use of the model-based policy iteration algorithm. The API algorithm will parameterize the value function to take the belief state as input and generate the estimated value for each action in the current belief state as output. The structure of the model described in Chapter 6 will assist in selecting the optimal parametric value function.

The approximate PI algorithm will improve and update the policy (initialized by parameters) using policy gradient methods to optimize the parameters for minimal cost. The updated policy, along with the environmental model from Chapter 6, will help simulate trajectories of state-action pairs and their rewards for various initial beliefs. This simulation will aid in evaluating the policy by averaging the reward of the entire trajectory using Monte Carlo methods.

The API algorithm will iterate between the policy improvement and policy evaluation steps, incorporating the environmental model. In each iteration, the policy parameters are updated using policy gradient methods based on the estimated values. After updating the policy, Monte Carlo simulation is performed using the environment model to evaluate the updated policy by estimating its value. The iterations continue until the policy converges to an optimal or near-optimal policy.

Bibliography

[1] Roy Adams and Ben Marlin. "Learning Time Series Detection Models from Temporally Imprecise Labels". In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Ed. by Aarti Singh and Jerry Zhu. Vol. 54. Proceedings of Machine Learning Research. PMLR, 2017, pp. 157–165. URL: https://proceedings.mlr.press/v54/adams17a.html.

[2] Mehmet Eren Ahsen, Robert M Vogel, and Gustavo A Stolovitzky. "Unsupervised Evaluation and Weighted Aggregation of Ranked Classification Predictions". In: *Journal of Machine Learning Research* 20.166 (2019), pp. 1–40. URL: http://jmlr.org/papers/v20/18-094.html.

[3] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019.

[4] Mehwish Alam, Russa Biswas, Yiyi Chen, Danilo Dessì, Genet Asefa Gesese, Fabian Hoppe, and Harald Sack. "HierClasSArt: Knowledge-Aware Hierarchical Classification of Scholarly Articles". In: *Companion Proceedings of the Web Conference*. 2021.

[5] Mohammed Alkahtani, Alok Choudhary, Arijit De, and Jennifer Anne Harding. "A decision support system based on ontology and data mining to improve design using warranty data". In: *Computers & Industrial Engineering* 128 (2019), pp. 1027–1039.

[6] Mehdi Allahyari, Krys J Kochut, and Maciej Janik. "Ontology-based text classification into dynamically defined topics". In: *2014 IEEE international conference on semantic computing*. IEEE. 2014, pp. 273–278.

[7] Erin L Allwein, Robert E Schapire, and Yoram Singer. "Reducing multiclass to binary: A unifying approach for margin classifiers". In: *Journal of machine learning research* 1.Dec (2000), pp. 113–141.

[8] Mohammed Alsuhaibani, Danushka Bollegala, Takanori Maehara, and Ken-ichi Kawarabayashi. "Jointly learning word embeddings using a corpus and a knowledge base". In: *PloS one* 13.3 (2018), e0193094.

[9] Rachel MacKay Altman. "Mixed Hidden Markov Models". In: *Journal of the American Statistical Association* 102.477 (2007), pp. 201–210. DOI: 10.1198/016214506000001086. eprint: https://doi.org/10.1198/016214506000001086. URL: https://doi.org/10.1198/016214506000001086.

[10] Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. "Tensor decompositions for learning latent variable models". In: *Journal of machine learning research* 15 (2014), pp. 2773–2832.

[11] Fazel Ansari. "Cost-based text understanding to improve maintenance knowledge intelligence in manufacturing enterprises". In: *Computers & Industrial Engineering* 141 (2020), p. 106319. ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2020.106319. URL: https://www.sciencedirect.com/science/article/pii/S036083522030053X.

[12] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. "A simple but tough-to-beat baseline for sentence embeddings". In: (2016). URL: https://github.com/talmago/simple-but-tough-to-beat-examples.

[13] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining". In: *Pro-*

ceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10). Valletta, Malta: European Language Resources Association (ELRA), May 2010. URL: http://www.lrec-conf.org/proceedings/lrec2010/pdf/769_Paper.pdf.

[14] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Muller. "How to explain individual classification decisions". In: *The Journal of Machine Learning Research* 11 (2010), pp. 1803–1831.

[15] Marco Barreno, Alvaro Cardenas, and J Doug Tygar. "Optimal ROC curve for a combination of classifiers". In: *Advances in Neural Information Processing Systems* 20 (2007).

[16] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains". In: *The Annals of Mathematical Statistics* 41.1 (1970), pp. 164–171. ISSN: 00034851. URL: http://www.jstor.org/stable/2239727 (visited on 05/05/2023).

[17] Miguel Ángel Bautista Martin, Oriol Pujol, Fernando De la Torre, and Sergio Escalera. "Error-Correcting Factorization". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.10 (2018), pp. 2388–2401. DOI: 10.1109/TPAMI.2017.2763146.

[18] Y. Bengio and P. Frasconi. "Input-output HMMs for sequence processing". In: *IEEE Transactions on Neural Networks* 7.5 (1996), pp. 1231–1249. DOI: 10.1109/72.536317.

[19] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. "Algorithms for Hyper-Parameter Optimization". In: *Advances in Neural Information Processing Systems*. Ed. by J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger. Vol. 24. Curran Associates, Inc., 2011. URL: https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf.

[20] Abhijeet Sandeep Bhardwaj, Akash Deep, Dharmaraj Veeramani, and Shiyu Zhou. "A Custom Word Embedding Model for Clustering of Maintenance Records". In: *IEEE Transactions on Industrial Informatics* 18.2 (2021), pp. 816–826.

[21] David Blei, Andrew Ng, and Michael Jordan. "Latent Dirichlet Allocation". In: *Advances in Neural Information Processing Systems*. Ed. by T. Dietterich, S. Becker, and Z. Ghahramani. Vol. 14. MIT Press, 2001. URL: https://proceedings.neurips.cc/paper/2001/file/296472c9542ad4d4788d543508116cbc-Paper.pdf.

[22] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet allocation". In: *Journal of Machine Learning research* 3.4-5 (2003), pp. 993–1022. ISSN: 15324435. DOI: 10.1016/b978-0-12-411519-4.00006-9.

[23] Danushka Bollegala, Mohammed Alsuhaibani, Takanori Maehara, and Ken-ichi Kawarabayashi. "Joint word representation learning using a corpus and a semantic lexicon". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 30. 1. 2016.

[24] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. "Translating embeddings for modeling multi-relational data". In: *Neural Information Processing Systems (NIPS)*. 2013, pp. 1–9.

[25] Michael P Brundage, Thurston Sexton, Melinda Hodkiewicz, Alden Dima, and Sarah Lukens. "Technical language processing: Unlocking maintenance knowledge". In: *Manufacturing Letters* 27 (2021), pp. 42–46.

[26] Sabine Buchholz and Erwin Marsi. "CoNLL-X Shared Task on Multilingual Dependency Parsing". In: *Proceedings of the Tenth Conference on Computational Natural Language Learning*. CoNLL-X '06. New York City, New York: Association for Computational Linguistics, 2006, 149–164. DOI: 10.5555/1596276.1596305.

[27] Carey Bunks, Dan McCarthy, and Tarik Al-Ani. "Condition-based maintenance of machines using Hidden Markov Models". In: *Mechanical Systems and Signal Processing* 14 (July 2000), pp. 597–612. DOI: 10.1006/mssp.2000.1309.

[28] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alipio Jorge, Célia Nunes, and Adam Jatowt. "YAKE! Keyword extraction from single documents using multiple local features". In: *Information Sciences* 509 (2020), pp. 257–289.

[29] Caihua Chen, Bingsheng He, Yinyu Ye, and Xiaoming Yuan. "The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent". In: *Mathematical Programming* 155.1 (2016), pp. 57–79. DOI: https://doi.org/10.1007/s10107-014-0826-5.

[30] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. "Neural natural language inference models enhanced with external knowledge". In: *arXiv preprint arXiv:1711.04289* (2018).

[31] Yen-Chi Chen. "A tutorial on kernel density estimation and recent advances". In: *Biostatistics & Epidemiology* 1.1 (2017), pp. 161–187.

[32] Zhen Chen, Yaping Li, Tangbin Xia, and Ershun Pan. "Hidden Markov model with auto-correlated observations for remaining useful life prediction and optimal maintenance policy". In: *Reliability Engineering & System Safety* 184 (2019). Impact of Prognostics and Health Management in Systems Reliability and Maintenance Planning, pp. 123–136. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2017.09.002. URL: https://www.sciencedirect.com/science/article/pii/S0951832017301710.

[33] Michael E. Cholette and Dragan Djurdjanovic. "Degradation modeling and monitoring of machines using operation-specific hidden Markov models". In: *IIE Transactions* 46.10 (2014), pp. 1107–1123. DOI: 10.1080/0740817X.2014.905734. eprint: https://doi.org/10.1080/0740817X.2014.905734. URL: https://doi.org/10.1080/0740817X.2014.905734.

[34] Marina Cidota and Monica Dumitrescu. "A multinomial - Hidden Markov model for communication systems influenced by external factors". In: *2012 7th IEEE Interna-*

tional Symposium on Applied Computational Intelligence and Informatics (SACI). 2012, pp. 235–240. DOI: 10.1109/SACI.2012.6250008.

[35] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust Region Methods*. Society for Industrial and Applied Mathematics, 2000. DOI: 10.1137/1.9780898719857. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9780898719857. URL: https://epubs.siam.org/doi/abs/10.1137/1.9780898719857.

[36] Isaac Councill, Ryan McDonald, and Leonid Velikovich. "What's great and what's not: learning to classify the scope of negation for improved sentiment analysis". In: *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*. Uppsala, Sweden: University of Antwerp, July 2010, pp. 51–59. URL: https://aclanthology.org/W10-3110.

[37] Fatoumata Dama and Christine Sinoquet. "Prediction and Inference in a Partially Hidden Markov-switching Framework with Autoregression. Application to Machinery Health Diagnosis". In: *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*. 2021, pp. 1–9. DOI: 10.1109/ICTAI52525.2021.00009.

[38] Akash Deep, Shiyu Zhou, Dharmaraj Veeramani, and Yong Chen. "HMM-Based Joint Modeling of Condition Monitoring Signals and Failure Event Data for Prognosis". In: *IEEE Transactions on Reliability* (2022), pp. 1–11. DOI: 10.1109/TR.2022.3193353.

[39] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[40] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June

2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

[41] Thomas G Dietterich and Ghulum Bakiri. "Solving multiclass learning problems via error-correcting output codes". In: *Journal of artificial intelligence research* 2 (1994), pp. 263–286.

[42] Xiaowen Ding, Bing Liu, and Philip S. Yu. "A Holistic Lexicon-Based Approach to Opinion Mining". In: *Proceedings of the 2008 International Conference on Web Search and Data Mining*. WSDM '08. Palo Alto, California, USA: Association for Computing Machinery, 2008, 231–240. ISBN: 9781595939272. DOI: 10.1145/1341531.1341561. URL: https://doi.org/10.1145/1341531.1341561.

[43] Ming Dong and David He. "Hidden semi-Markov model-based methodology for multi-sensor equipment health diagnosis and prognosis". In: *European Journal of Operational Research* 178.3 (2007), pp. 858–878. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2006.01.041. URL: https://www.sciencedirect.com/science/article/pii/S0377221706001421.

[44] Dejing Dou, Hao Wang, and Haishan Liu. "Semantic data mining: A survey of ontology-based approaches". In: *Proceedings of the 2015 IEEE 9th international conference on semantic computing (IEEE ICSC 2015)*. IEEE. 2015, pp. 244–251.

[45] Min Du, Feifei Li, Guineng Zheng, and Vivek Srikumar. "Deeplog: Anomaly detection and diagnosis from system logs through deep learning". In: *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*. 2017, pp. 1285–1298.

[46] Jacob Eisenstein. "Unsupervised Learning for Lexicon-Based Classification". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI'17. San Francisco, California, USA: AAAI Press, 2017, 3188–3194. DOI: https://dl.acm.org/doi/10.5555/3298023.3298032.

[47] Martine Enger, Erik Velldal, and Lilja Øvrelid. "An open-source tool for negation detection: a maximum-margin approach". In: *Proceedings of the Workshop Computational Semantics Beyond Events and Roles*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 64–69. DOI: 10.18653/v1/W17-1810. URL: https://aclanthology.org/W17-1810.

[48] Weili Fang, Hanbin Luo, Shuangjie Xu, Peter ED Love, Zhenchuan Lu, and Cheng Ye. "Automated text classification of near-misses from safety reports: An improved deep learning approach". In: *Advanced Engineering Informatics* 44 (2020), p. 101060.

[49] Kai-Jie Feng, Sze-Teng Liong, and Kun-Hong Liu. "The design of variable-length coding matrix for improving error correcting output codes". In: *Information Sciences* 534 (2020), pp. 192–217. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2020.04.021. URL: https://www.sciencedirect.com/science/article/pii/S0020025520303170.

[50] Milagros Fernández-Gavilanes, Tamara Álvarez-López, Jonathan Juncal-Martínez, Enrique Costa-Montenegro, and Francisco Javier González-Castaño. "Unsupervised method for sentiment analysis in online texts". In: *Expert Systems with Applications* 58 (2016), pp. 57–75. DOI: https://doi.org/10.1016/j.eswa.2016.03.031.

[51] Katerina T Frantzi and Sophia Ananiadou. "Automatic term recognition using contextual cues". In: *In Proceedings of 3rd DELOS Workshop*. Citeseer. 1997.

[52] Benoît Frénay and Michel Verleysen. "Classification in the presence of label noise: a survey". In: *IEEE transactions on neural networks and learning systems* 25.5 (2013), pp. 845–869.

[53] Yoav Freund and Yishay Mansour. "Estimating a mixture of two product distributions". In: *Proceedings of the twelfth annual conference on Computational learning theory*. 1999, pp. 53–62.

[54] Milton Friedman. "The use of ranks to avoid the assumption of normality implicit in the analysis of variance". In: *Journal of the american statistical association* 32.200 (1937), pp. 675–701.

[55] Xianghua Fu, Haiying Wu, and Laizhong Cui. "Topic Sentiment Joint Model with Word Embeddings." In: *DMNLP@ PKDD/ECML*. Citeseer. 2016, pp. 41–48.

[56] Joseph Futoma, Sanjay Hariharan, and Katherine Heller. "Learning to detect sepsis with a multitask Gaussian process RNN classifier". In: *International conference on machine learning*. PMLR. 2017, pp. 1174–1182.

[57] M. Ghiassi, Sean Lee, and Swati Ramesh Gaikwad. "Sentiment analysis and spam filtering using the YAC2 clustering algorithm with transferability". In: *Computers & Industrial Engineering* 165 (2022), p. 107959. ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2022.107959. URL: https://www.sciencedirect.com/science/article/pii/S0360835222000298.

[58] Xinze Guan, Raviv Raich, and Weng-Keen Wong. "Efficient Multi-Instance Learning for Activity Recognition from Time Series Data Using an Auto-Regressive Hidden Markov Model". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 2016, pp. 2330–2339. URL: https://proceedings.mlr.press/v48/guan16.html.

[59] Jian Guo, Zhaojun Li, and Meiyan Li. "A Review on Prognostics Methods for Engineering Systems". In: *IEEE Transactions on Reliability* 69.3 (2020), pp. 1110–1129. DOI: 10.1109/TR.2019.2957965.

[60] Ruosi Guo, Chunming Zhang, and Zhengjun Zhang. "Maximum Independent Component Analysis with Application to EEG Data". In: *Statistical Science* 35.1 (2020), pp. 145 –157. DOI: 10.1214/19-STS763.

[61] William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. "Inducing Domain-Specific Sentiment Lexicons from Unlabeled Corpora". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 595–605. DOI: 10.18653/v1/D16-1057. URL: https://aclanthology.org/D16-1057.

[62] Zellig S Harris. "Distributional structure". In: *Word* 10.2-3 (1954), pp. 146–162.

[63] Shilin He, Pinjia He, Zhuangbin Chen, Tianyi Yang, Yuxin Su, and Michael R Lyu. "A survey on automated log analysis for reliability engineering". In: *ACM Computing Surveys (CSUR)* 54.6 (2021), pp. 1–37.

[64] Melinda Hodkiewicz and Mark Tien-Wei Ho. "Cleaning historical maintenance work order data for reliability analysis". In: *Journal of Quality in Maintenance Engineering* (2016).

[65] Jiawen Hu, Qiuzhuang Sun, Zhi-Sheng Ye, and Qiang Zhou. "Joint Modeling of Degradation and Lifetime Data for RUL Prediction of Deteriorating Products". In: *IEEE Transactions on Industrial Informatics* 17.7 (2021), pp. 4521–4531. DOI: 10.1109/TII.2020.3021054.

[66] Lawrence Hubert and Phipps Arabie. "Comparing partitions". In: *Journal of classification* 2.1 (1985), pp. 193–218. DOI: https://doi.org/10.1007/BF01908075.

[67] Ioana Hulpus, Conor Hayes, Marcel Karnstedt, and Derek Greene. "Unsupervised graph-based topic labelling using dbpedia". In: *Proceedings of the sixth ACM international conference on Web search and data mining*. 2013, pp. 465–474.

[68] C. Hutto and Eric Gilbert. "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text". In: *Proceedings of the International AAAI Conference on Web and Social Media* 8.1 (2014), pp. 216–225. URL: https://ojs.aaai.org/index.php/ICWSM/article/view/14550.

[69] Muhammad Imran and Robert IM Young. "Reference ontologies for interoperability across multiple assembly systems". In: *International Journal of Production Research* 54.18 (2016), pp. 5381–5403.

[70] BS ISO. "14224,"Petroleum, petrochemicals and natural gas industries: collection and exchange of reliability and maintenance data for equipment"". In: *British Standards Institution, UK* (2016), pp. 1–78.

[71] Ariel Jaffe, Boaz Nadler, and Yuval Kluger. "Estimating the accuracies of multiple classifiers without labeled data". In: *Artificial Intelligence and Statistics*. PMLR. 2015, pp. 407–415.

[72] Salud María Jiménez-Zafra, Roser Morante, María Teresa Martín-Valdivia, and L. Alfonso Ureña-López. "Corpora Annotated with Negation: An Overview". In: *Computational Linguistics* 46.1 (Mar. 2020), pp. 1–52. ISSN: 0891-2017. DOI: 10.1162/coli_a_00371. eprint: https://direct.mit.edu/coli/article-pdf/46/1/1/1847769/coli_a_00371.pdf. URL: https://doi.org/10.1162/coli_a_00371.

[73] Namcheol Jung and Ghang Lee. "Automated Classification of Building Information Modeling (BIM) Case Studies by BIM Use Based on Natural Language Processing (NLP) and Unsupervised Learning". In: *Adv. Eng. Inform.* 41.C (2019). ISSN: 1474-0346. DOI: 10.1016/j.aei.2019.04.007. URL: https://doi.org/10.1016/j.aei.2019.04.007.

[74] Soumaya El Kadiri and Dimitris Kiritsis. "Ontologies in the context of product lifecycle management: state of the art literature review". In: *International Journal of Production Research* 53.18 (2015), pp. 5657–5668. DOI: 10.1080/00207543.2015.1052155. eprint: https://doi.org/10.1080/00207543.2015.1052155. URL: https://doi.org/10.1080/00207543.2015.1052155.

[75] Elham Khabiri, Wesley M Gifford, Bhanukiran Vinzamuri, Dhaval Patel, and Pietro Mazzoleni. "Industry Specific Word Embedding and its Application in Log Classifica-

tion". In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2019, pp. 2713–2721.

[76] Dimitris Kiritsis. "Semantic technologies for engineering asset life cycle management". In: *International Journal of Production Research* 51.23-24 (2013), pp. 7345–7371.

[77] Sylvere Krima, Raphael Barbau, Xenia Fiorentini, Rachuri Sudarsan, and Ram D Sriram. "Ontostep: OWL-DL ontology for step". In: *National Institute of Standards and Technology, NISTIR* 7561 (2009).

[78] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. "From word embeddings to document distances". In: *International conference on machine learning*. 2015, pp. 957–966.

[79] Peter A Lachenbruch. "Note on initial misclassification effects on the quadratic discriminant function". In: *Technometrics* 21.1 (1979), pp. 129–132.

[80] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. "Albert: A lite bert for self-supervised learning of language representations". In: *arXiv preprint arXiv:1909.11942* (2019).

[81] Thomas K Landauer and Susan T Dumais. "A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge". In: *Psychological Review* 104.2 (1997), pp. 211–240.

[82] Seokho Lee and Hyelim Jung. "Individual Transition Label Noise Logistic Regression in Binary Classification for Incorrectly Labeled Data". In: *Technometrics* (2021), pp. 1–12.

[83] Vladimir I Levenshtein. "Binary codes capable of correcting deletions, insertions, and reversals". In: *Soviet physics doklady*. Vol. 10. 8. Soviet Union. 1966, pp. 707–710.

[84] Ronghan Li, Zejun Jiang, Lifang Wang, Xinyu Lu, Meng Zhao, and Daqing Chen. "Enhancing Transformer-based language models with commonsense representations for knowledge-driven machine comprehension". In: *Knowledge-Based Systems* (2021), p. 106936.

[85] Yongxin Liao, Mario Lezoche, Hervé Panetto, and Nacer Boudjlida. "Semantic annotations for semantic interoperability in a product lifecycle management context". In: *International Journal of Production Research* 54.18 (2016), pp. 5534–5553.

[86] Chenghua Lin, Yulan He, and Richard Everson. "A Comparative Study of Bayesian Models for Unsupervised Sentiment Detection". In: *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*. Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 144–152. URL: https://aclanthology.org/W10-2918.

[87] D. Y. Lin and L. J. Wei. "The Robust Inference for the Cox Proportional Hazards Model". In: *Journal of the American Statistical Association* 84.408 (1989), pp. 1074–1078. DOI: 10.1080/01621459.1989.10478874. eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.1989.10478874. URL: https://www.tandfonline.com/doi/abs/10.1080/01621459.1989.10478874.

[88] Guangyi Lin, Kunhong Liu, Beizhan Wang, and Xiaoyan Zhang. "Partial label learning based on label distributions and error-correcting output codes". In: *Soft Computing* 25.2 (2021), pp. 1049–1064.

[89] Qingwei Lin, Hongyu Zhang, Jian-Guang Lou, Yu Zhang, and Xuewei Chen. "Log clustering based problem identification for online service systems". In: *2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)*. IEEE. 2016, pp. 102–111.

[90] Kaijian Liu and Nora El-Gohary. "Ontology-based semi-supervised conditional random fields for automated information extraction from bridge inspection reports". In: *Automation in construction* 81 (2017), pp. 313–327.

[91] Kun-Hong Liu, Zhi-Hao Zeng, and Vincent To Yee Ng. "A Hierarchical Ensemble of ECOC for cancer classification based on multi-class microarray data". In: *Information Sciences* 349-350 (2016), pp. 102–118. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2016.02.028. URL: https://www.sciencedirect.com/science/article/pii/S0020025516300937.

[92] Anh Tuan Luu, Yi Tay, Siu Cheung Hui, and See Kiong Ng. "Learning term embeddings for taxonomic relation identification using dynamic weighting neural network". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2016, pp. 403–413.

[93] Yinglong Ma, Jingpeng Zhao, and Beihong Jin. "A Hierarchical Fine-Tuning Approach Based on Joint Embedding of Words and Parent Categories for Hierarchical Multi-label Text Classification". In: *International Conference on Artificial Neural Networks*. Springer. 2020, pp. 746–757.

[94] Rasmus E. Madsen, David Kauchak, and Charles Elkan. "Modeling Word Burstiness Using the Dirichlet Distribution". In: *Proceedings of the 22nd International Conference on Machine Learning*. ICML '05. Bonn, Germany: Association for Computing Machinery, 2005, 545–552. ISBN: 1595931805. DOI: 10.1145/1102351.1102420. URL: https://doi.org/10.1145/1102351.1102420.

[95] Christopher Malon. "Overcoming Poor Word Embeddings with Word Definitions". In: *arXiv preprint arXiv:2103.03842* (2021).

[96] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. "The Stanford CoreNLP Natural Language Processing Toolkit". In: *Proceedings of 52nd Annual Meeting of the Association for Computational Lin-*

guistics: System Demonstrations. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 55–60. DOI: 10.3115/v1/P14-5010. URL: https://aclanthology.org/P14-5010.

[97] Gunasekaran Manogaran, V Vijayakumar, R Varatharajan, Priyan Malarvizhi Kumar, Revathi Sundarasekar, and Ching-Hsien Hsu. "Machine learning based big data processing framework for cancer diagnosis using hidden Markov model and GM clustering". In: *Wireless personal communications* 102 (2018), pp. 2099–2116.

[98] Michał Marcińczuk. "Automatic construction of complex features in conditional random fields for named entities recognition". In: *Proceedings of the International Conference Recent Advances in Natural Language Processing*. 2015, pp. 413–419.

[99] Katya Mauff, Ewout Steyerberg, Isabella Kardys, Eric Boersma, and Dimitris Rizopoulos. "Joint models with multiple longitudinal outcomes and a time-to-event outcome: a corrected two-stage approach". In: *Statistics and Computing* 30 (2020), pp. 999–1014.

[100] Alessio Mazzetto, Cyrus Cousins, Dylan Sam, Stephen H Bach, and Eli Upfal. "Adversarial Multi Class Learning under Weak Supervision with Performance Guarantees". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 7534–7543.

[101] Qiaozhu Mei, Xuehua Shen, and ChengXiang Zhai. "Automatic labeling of multinomial topic models". In: *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2007, pp. 490–499.

[102] Weibin Meng, Ying Liu, Yuheng Huang, Shenglin Zhang, Federico Zaiter, Bingjin Chen, and Dan Pei. "A semantic-aware representation framework for online log analysis". In: *2020 29th International Conference on Computer Communications and Networks (ICCCN)*. IEEE. 2020, pp. 1–7.

[103] Weibin Meng, Ying Liu, Yichen Zhu, Shenglin Zhang, Dan Pei, Yuqing Liu, Yihao Chen, Ruizhi Zhang, Shimin Tao, Pei Sun, et al. "LogAnomaly: Unsupervised de-

tection of sequential and quantitative anomalies in unstructured logs." In: *IJCAI*. Vol. 19. 7. 2019, pp. 4739–4745.

[104] Weibin Meng, Federico Zaiter, Yuheng Huang, Ying Liu, Shenglin Zhang, Yuzhe Zhang, Yichen Zhu, Tianke Zhang, En Wang, Zuomin Ren, et al. "Summarizing Unstructured Logs in Online Services". In: *arXiv preprint arXiv:2012.08938* (2020).

[105] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space". In: *1st International Conference on Learning Representation (ICLR 2013)* (2013), pp. 1–12. arXiv: 1301.3781.

[106] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013). URL: https://arxiv.org/abs/1301.3781.

[107] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality". In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. Curran Associates Inc., 2013, pp. 3111–3119.

[108] Ramin Moghaddass and Ming J. Zuo. "An integrated framework for online diagnostic and prognostic health monitoring using a multistate deterioration process". In: *Reliability Engineering & System Safety* 124 (2014), pp. 92–104. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2013.11.006. URL: https://www.sciencedirect.com/science/article/pii/S0951832013003037.

[109] Ramin Moghaddass, Ming J Zuo, Yu Liu, and Hong-zhong Huang. "Predictive analytics using a nonhomogeneous semi-Markov model and inspection data". In: *IIE transactions* 47.5 (2015), pp. 505–520.

[110] Bhavya Mor, Sunita Garhwal, and Ajay Kumar. "A systematic review of hidden Markov models and their applications". In: *Archives of computational methods in engineering* 28 (2021), pp. 1429–1448.

[111] Reagan Mozer, Luke Miratrix, Aaron Russell Kaufman, and L Jason Anastasopoulos. "Matching with text data: An experimental evaluation of methods for matching documents and of measuring match quality". In: *Political Analysis* 28.4 (2020), pp. 445–468.

[112] Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. "Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints". In: *Transactions of the association for Computational Linguistics* 5 (2017), pp. 309–324.

[113] Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. "Improving Topic Models with Latent Feature Word Representations". In: *Transactions of the Association for Computational Linguistics* 3 (2015), pp. 299–313. DOI: 10.1162/tacl_a_00140. URL: https://aclanthology.org/Q15-1022.

[114] Huseyin Ozkan, Arda Akman, and Suleyman S Kozat. "A Novel Training Algorithm for HMMs with Partial and Noisy Access to the States". In: *arXiv preprint arXiv:1203.4597* (2012).

[115] Woodley Packard, Emily M. Bender, Jonathon Read, Stephan Oepen, and Rebecca Dridan. "Simple Negation Scope Resolution through Deep Parsing: A Semantic Solution to a Semantic Problem". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 69–78. DOI: 10.3115/v1/P14-1007. URL: https://aclanthology.org/P14-1007.

[116] Eirini Papagiannopoulou and Grigorios Tsoumakas. "Local word vectors guiding keyphrase extraction". In: *Information Processing & Management* 54.6 (2018), pp. 888–902.

[117] Fabio Parisi, Francesco Strino, Boaz Nadler, and Yuval Kluger. "Ranking and combining multiple predictors without labeled data". In: *Proceedings of the National Academy of Sciences* 111.4 (2014), pp. 1253–1258.

[118] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[119] Matthew E Peters, Mark Neumann, Robert L Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. "Knowledge enhanced contextual word representations". In: *arXiv preprint arXiv:1909.04164* (2019).

[120] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. "Expanding Domain Sentiment Lexicon through Double Propagation". In: IJCAI'09. Pasadena, California, USA: Morgan Kaufmann Publishers Inc., 2009, 1199–1204. DOI: 10.5555/1661445.1661637.

[121] Siqi Qiu, Xinguo Ming, Mohamed Sallak, and Jialiang Lu. "Joint optimization of production and condition-based maintenance scheduling for make-to-order manufacturing systems". In: *Computers & Industrial Engineering* 162 (2021), p. 107753. ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2021.107753. URL: https://www.sciencedirect.com/science/article/pii/S0360835221006574.

[122] L.R. Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition". In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286. DOI: 10.1109/5.18626.

[123] Charles M. Rader and Leland B. Jackson. "Approximating Noncausal IIR Digital Filters Having Arbitrary Poles, Including New Hilbert Transformer Designs, Via Forward/Backward Block Recursion". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 53.12 (2006), pp. 2779–2787. DOI: 10.1109/TCSI.2006.883877.

[124] K Rajbabu, Harshavardhan Srinivas, and S Sudha. "Industrial information extraction through multi-phase classification using ontology for unstructured documents". In: *Computers in Industry* 100 (2018), pp. 137–147.

[125] Dnyanesh Rajpathak, Rahul Chougule, and Pulak Bandyopadhyay. "A domain-specific decision support system for knowledge discovery using association and text mining". In: *Knowledge and information systems* 31.3 (2012), pp. 405–432.

[126] Dnyanesh Rajpathak and Soumen De. "A data-and ontology-driven text mining-based construction of reliability model to analyze and predict component failures". In: *Knowledge and Information Systems* 46.1 (2016), pp. 87–113.

[127] Dnyanesh G Rajpathak. "An ontology based text mining system for knowledge discovery from the diagnosis data in the automotive domain". In: *Computers in Industry* 64.5 (2013), pp. 565–580.

[128] Ryan Rifkin and Aldebaro Klautau. "In defense of one-vs-all classification". In: *The Journal of Machine Learning Research* 5 (2004), pp. 101–141.

[129] Stephen Robertson. "Understanding inverse document frequency: on theoretical arguments for IDF". In: *Journal of documentation* 60 (2004), pp. 503–520. DOI: 10.1108/00220410410560582.

[130] Anderson Rocha and Siome Klein Goldenstein. "Multiclass From Binary: Expanding One-Versus-All, One-Versus-One and ECOC-Based Approaches". In: *IEEE Transactions on Neural Networks and Learning Systems* 25.2 (2014), pp. 289–302. DOI: 10.1109/TNNLS.2013.2274735.

[131] Mayra Z Rodriguez, Cesar H Comin, Dalcimar Casanova, Odemir M Bruno, Diego R Amancio, Luciano da F Costa, and Francisco A Rodrigues. "Clustering algorithms: A comparative approach". In: *PloS one* 14.1 (2019), e0210236.

[132] Xin Rong. "word2vec Parameter Learning Explained". In: *ArXiv* abs/1411.2738 (2014).

[133] Arpita Roy, Youngja Park, and Shimei Pan. "Predicting malware attributes from cybersecurity texts". In: *UMBC Student Collection* (2019).

[134] Kittipong Saetia, Sarah Lukens, X Hu, and H Pijcke. "Data-driven approach to equipment taxonomy classification". In: *Proceedings of the PHM Society Conference*. 2019.

[135] Beatrice Santorini. "Part-of-speech tagging guidelines for the Penn Treebank Project". In: (1990).

[136] Tobias Scheffer, Christian Decomain, and Stefan Wrobel. "Active Hidden Markov Models for Information Extraction". In: *Advances in Intelligent Data Analysis*. Ed. by Frank Hoffmann, David J. Hand, Niall Adams, Douglas Fisher, and Gabriela Guimaraes. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 309–318. ISBN: 978-3-540-44816-7.

[137] Kristen A. Severson, Lana M. Chahine, Luba Smolensky, Kenney Ng, Jianying Hu, and Soumya Ghosh. "Personalized Input-Output Hidden Markov Models for Disease Progression Modeling". In: *Proceedings of the 5th Machine Learning for Healthcare Conference*. Ed. by Finale Doshi-Velez, Jim Fackler, Ken Jung, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens. Vol. 126. Proceedings of Machine Learning Research. PMLR, 2020, pp. 309–330. URL: https://proceedings.mlr.press/v126/severson20a.html.

[138] Thurston Sexton, Michael P Brundage, Michael Hoffman, and Katherine C Morris. "Hybrid datafication of maintenance logs from ai-assisted human tags". In: *2017 ieee international conference on big data (big data)*. IEEE. 2017, pp. 1769–1777.

[139] Thurston Sexton and Mark Fuge. "Organizing Tagged Knowledge: Similarity Measures and Semantic Fluency in Structure Mining". In: *Journal of Mechanical Design* 142.3 (2020).

[140] Thurston Sexton, Melinda Hodkiewicz, Michael P Brundage, and Thomas Smoker. "Benchmarking for keyword extraction methodologies in maintenance work orders". In: *PHM society conference*. Vol. 10. 2018.

[141] Thurston B Sexton and Michael P Brundage. "Nestor: A Tool for Natural Language Annotation of Short Texts". In: *J. Res. NIST* 124 (2019).

[142] Michael Sharp, Thurston Sexton, and Michael P Brundage. "Toward semi-autonomous information". In: *IFIP International Conference on Advances in Production Management Systems*. Springer. 2017, pp. 425–432.

[143] Xiao-Sheng Si, Wenbin Wang, Chang-Hua Hu, and Dong-Hua Zhou. "Remaining useful life estimation – A review on the statistical data driven approaches". In: *European Journal of Operational Research* 213.1 (2011), pp. 1–14. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2010.11.018. URL: https://www.sciencedirect.com/science/article/pii/S0377221710007903.

[144] Suzanna Sia, Ayush Dalmia, and Sabrina J Mielke. "Tired of Topic Models? Clusters of Pretrained Word Embeddings Make for Fast and Good Topics too!" In: *arXiv preprint arXiv:2004.14914* (2020).

[145] Tipawan Silwattananusarn and Kulthida Tuamsuk. "Data mining and its applications for knowledge management: a literature review from 2007 to 2012". In: *arXiv preprint arXiv:1210.2872* (2012).

[146] Ruben Sipos, Dmitriy Fradkin, Fabian Moerchen, and Zhuang Wang. "Log-based predictive maintenance". In: *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining*. 2014, pp. 1867–1876.

[147] E. Skordilis and R. Moghaddass. "A condition monitoring approach for real-time monitoring of degrading systems using Kalman filter and logistic regression". In: *International Journal of Production Research* 55.19 (2017), pp. 5579–5596. DOI: 10.1080/00207543.2017.1308573. eprint: https://doi.org/10.1080/00207543.2017.1308573. URL: https://doi.org/10.1080/00207543.2017.1308573.

[148] Erotokritos Skordilis and Ramin Moghaddass. "A Double Hybrid State-Space Model for Real-Time Sensor-Driven Monitoring of Deteriorating Systems". In: *IEEE Transactions on Automation Science and Engineering* 17.1 (2020), pp. 72–87. DOI: 10.1109/TASE.2019.2921285.

[149] Junbo Son, Qiang Zhou, Shiyu Zhou, Xiaofeng Mao, and Mutasim Salman. "Evaluation and Comparison of Mixed Effects Model Based Prognosis for Hard Failure". In: *IEEE Transactions on Reliability* 62.2 (2013), pp. 379–394. DOI: 10.1109/TR.2013.2259205.

[150] Moxian Song, Hongyan Li, Chenxi Sun, Derun Cai, and Shenda Hong. "Dlsa: Semi-supervised partial label learning via dependence-maximized label set assignment". In: *Information Sciences* 609 (2022), pp. 1169–1180. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2022.07.114. URL: https://www.sciencedirect.com/science/article/pii/S0020025522008039.

[151] Bird Steven, Klein Ewan, and Loper Edward. "Natural language processing with python, analyzing text with the natural language toolkit". In: *Language Resources and Evaluation* 44.4 (2010), pp. 421–424.

[152] Philip J Stone, Dexter C Dunphy, and Marshall S Smith. "The general inquirer: A computer approach to content analysis." In: (1966).

[153] Erik Strumbelj and Igor Kononenko. "An efficient explanation of individual classifications using game theory". In: *The Journal of Machine Learning Research* 11 (2010), pp. 1–18.

[154] Alan Stuart. "The correlation between variate-values and ranks in samples from a continuous distribution". In: *British Journal of Statistical Psychology* 7.1 (1954), pp. 37–44.

[155] Charles Sutton, Andrew McCallum, et al. "An introduction to conditional random fields". In: *Foundations and Trends® in Machine Learning* 4.4 (2012), pp. 267–373.

[156] Anderson Luis Szejka, Osiris Canciglieri Jr, Hervé Panetto, Eduardo Rocha Loures, and Alexis Aubry. "Semantic interoperability for an integrated product development process: a systematic literature review". In: *International Journal of Production Research* 55.22 (2017), pp. 6691–6709.

[157] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. "Lexicon-Based Methods for Sentiment Analysis". In: *Computational Linguistics* 37.2 (June 2011), pp. 267–307. ISSN: 0891-2017. DOI: 10.1162/COLI_a_00049. eprint: https://direct.mit.edu/coli/article-pdf/37/2/267/1798865/coli_a_00049.pdf. URL: https://doi.org/10.1162/COLI_a_00049.

[158] Andy Taylor, Benjamin Joachimi, and Thomas Kitching. "Putting the precision in precision cosmology: How accurate should your data covariance matrix be?" In: *Monthly Notices of the Royal Astronomical Society* 432.3 (May 2013), pp. 1928–1946. ISSN: 0035-8711. DOI: 10.1093/mnras/stt270. eprint: https://academic.oup.com/mnras/article-pdf/432/3/1928/12612680/stt270.pdf. URL: https://doi.org/10.1093/mnras/stt270.

[159] Julien Tissier, Christophe Gravier, and Amaury Habrard. "Dict2vec: Learning word embeddings using lexical dictionaries". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2017, pp. 254–263.

[160] Panagiotis A Traganitis, Alba Pages-Zamora, and Georgios B Giannakis. "Blind multiclass ensemble classification". In: *IEEE Transactions on Signal Processing* 66.18 (2018), pp. 4737–4752.

[161] Amy J. C. Trappey, Charles V. Trappey, Tzu-An Chiang, and Yi-Hsuan Huang. "Ontology-based neural network for patent knowledge management in design collaboration". In: *International Journal of Production Research* 51.7 (2013), pp. 1992–2005. DOI: 10.1080/00207543.2012.701775. eprint: https://doi.org/10.1080/00207543.2012.701775. URL: https://doi.org/10.1080/00207543.2012.701775.

[162] Ciprian-Octavian Truica and Elena-Simona Apostol. "TLATR: Automatic Topic Labeling Using Automatic (Domain-Specific) Term Recognition". In: *IEEE Access* 9 (2021), pp. 76624–76641.

[163] Peter D. Turney. "Thumbs up or Thumbs down? Semantic Orientation Applied to Unsupervised Classification of Reviews". In: ACL '02. Philadelphia, Pennsylvania: Association for Computational Linguistics, 2002, 417–424. DOI: 10.3115/1073083.1073153. URL: https://doi.org/10.3115/1073083.1073153.

[164] Nadeem Ur-Rahman and Jennifer A Harding. "Textual data mining for industrial knowledge management and text classification: A business oriented approach". In: *Expert Systems with Applications* 39.5 (2012), pp. 4729–4739.

[165] Juan Pablo Usuga Cadavid, Bernard Grabot, Samir Lamouri, Robert Pellerin, and Arnaud Fortin. "Valuing free-form text data from maintenance logs through transfer learning with CamemBERT". In: *Enterprise Information Systems* (2020), pp. 1–29.

[166] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[167] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *arXiv preprint arXiv:1706.03762* (2017).

[168] A. Viterbi. "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm". In: *IEEE Transactions on Information Theory* 13.2 (1967), pp. 260–269. DOI: 10.1109/TIT.1967.1054010.

[169] Ivan Vulić and Nikola Mrkšić. "Specialising word vectors for lexical entailment". In: *arXiv preprint arXiv:1710.06371* (2017).

[170] Jiafu Wan, Shenglong Tang, Di Li, Shiyong Wang, Chengliang Liu, Haider Abbas, and Athanasios V Vasilakos. "A manufacturing big data solution for active preventive

maintenance". In: *IEEE Transactions on Industrial Informatics* 13.4 (2017), pp. 2039–2047.

[171] Feng Wang, Tianhua Xu, Tao Tang, MengChu Zhou, and Haifeng Wang. "Bilevel feature extraction-based text mining for fault diagnosis of railway systems". In: *IEEE transactions on intelligent transportation systems* 18.1 (2016), pp. 49–58.

[172] Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Cuihong Cao, Daxin Jiang, Ming Zhou, et al. "K-adapter: Infusing knowledge into pre-trained models with adapters". In: *arXiv preprint arXiv:2002.01808* (2020).

[173] Shaonan Wang, Jiajun Zhang, and Chengqing Zong. "Learning sentence representation with guidance of human attention". In: *arXiv preprint arXiv:1609.09189* (2016).

[174] Shaonan Wang, Jiajun Zhang, and Chengqing Zong. *Learning Sentence Representation with Guidance of Human Attention*. 2017. arXiv: 1609.09189 [cs.CL].

[175] Yue Wang, Xiang Li, Linda L Zhang, and Daniel Mo. "Configuring products with natural language: a simple yet effective approach based on text embeddings and multilayer perceptron". In: *International Journal of Production Research* (2021), pp. 1–13.

[176] P. Welch. "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms". In: *IEEE Transactions on Audio and Electroacoustics* 15.2 (1967), pp. 70–73. DOI: 10.1109/TAU.1967.1161901.

[177] Michael Wiegand, Alexandra Balahur, Benjamin Roth, Dietrich Klakow, and Andrés Montoyo. "A survey on the role of negation in sentiment analysis". In: *Proceedings of the workshop on negation and speculation in natural language processing*. 2010, pp. 60–68.

[178] Yan Xiao, Congdong Li, Matthias Thürer, Yide Liu, and Ting Qu. "Towards Lean Automation: Fine-Grained sentiment analysis for customer value identification". In: *Computers & Industrial Engineering* 169 (2022), p. 108186. ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2022.108186. URL: https://www.sciencedirect.com/science/article/pii/S036083522200256X.

[179] Hu Xu, Bing Liu, Lei Shu, and Philip Yu. "DomBERT: Domain-oriented Language Model for Aspect-based Sentiment Analysis". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 1725–1731. DOI: 10.18653/v1/2020.findings-emnlp.156. URL: https://aclanthology.org/2020.findings-emnlp.156.

[180] Wei Xu, Ling Huang, Armando Fox, David Patterson, and Michael Jordan. "Online system problem detection by mining patterns of console logs". In: *2009 Ninth IEEE International Conference on Data Mining*. IEEE. 2009, pp. 588–597.

[181] Zhaoguang Xu and Yanzhong Dang. "Solution knowledge mining and recommendation for quality problem-solving". In: *Computers & Industrial Engineering* 159 (2021), p. 107313. ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2021.107313. URL: https://www.sciencedirect.com/science/article/pii/S0360835221002175.

[182] Tianji Yang, Zeyu Zheng, and Liang Qi. "A method for degradation prediction based on Hidden semi-Markov models with mixture of Kernels". In: *Computers in Industry* 122 (2020), p. 103295. ISSN: 0166-3615. DOI: https://doi.org/10.1016/j.compind.2020.103295. URL: https://www.sciencedirect.com/science/article/pii/S0166361520305297.

[183] Zheng Yu, Haixun Wang, Xuemin Lin, and Min Wang. "Learning Term Embeddings for Hypernymy Identification". In: *Proceedings of the 24th International Conference*

on Artificial Intelligence. IJCAI'15. Buenos Aires, Argentina: AAAI Press, 2015, 1390–1397. ISBN: 9781577357384.

[184] Xubo Yue and Raed Al Kontar. "Joint Models for Event Prediction From Time Series and Survival Data". In: *Technometrics* 63.4 (2021), pp. 477–486. DOI: 10.1080/00401706.2020.1832582. eprint: https://doi.org/10.1080/00401706.2020.1832582. URL: https://doi.org/10.1080/00401706.2020.1832582.

[185] Chao Zhang, Fangbo Tao, Xiusi Chen, Jiaming Shen, Meng Jiang, Brian Sadler, Michelle Vanni, and Jiawei Han. "TaxoGen: Unsupervised Topic Taxonomy Construction by Adaptive Term Embedding and Clustering". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '18. London, United Kingdom: Association for Computing Machinery, 2018, 2701–2709. ISBN: 9781450355520. DOI: 10.1145/3219819.3220064.

[186] Heng Zhang, Utpal Roy, and Yung-Tsun Tina Lee. "Enriching analytics models with domain knowledge for smart manufacturing data analysis". In: *International Journal of Production Research* 58.20 (2020), pp. 6399–6415. DOI: 10.1080/00207543.2019.1680895. eprint: https://doi.org/10.1080/00207543.2019.1680895. URL: https://doi.org/10.1080/00207543.2019.1680895.

[187] Kangkang Zhang, Ruben Gonzalez, Biao Huang, and Guoli Ji. "Expectation–Maximization Approach to Fault Diagnosis With Missing Data". In: *IEEE Transactions on Industrial Electronics* 62.2 (2015), pp. 1231–1240. DOI: 10.1109/TIE.2014.2336635.

[188] Lei Zhang, Shuai Wang, and Bing Liu. "Deep learning for sentiment analysis: A survey". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8.4 (2018), e1253. DOI: 10.1002/widm.1253.

[189] Yuchen Zhang, Xi Chen, Dengyong Zhou, and Michael I Jordan. "Spectral methods meet EM: A provably optimal algorithm for crowdsourcing". In: *Advances in neural information processing systems* 27 (2014).

[190] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. "ERNIE: Enhanced language representation with informative entities". In: *arXiv preprint arXiv:1905.07129* (2019).

[191] Zhenfei Zhao, Weina Niu, Xiaosong Zhang, Runzi Zhang, Zhenqi Yu, and Cheng Huang. "Trine: Syslog anomaly detection with three transformer encoders in one generative adversarial network". In: *Applied Intelligence* (2021), pp. 1–10.

[192] Qiang Zhou, Junbo Son, Shiyu Zhou, Xiaofeng Mao, and Mutasim Salman. "Remaining useful life prediction of individual units subject to hard failure". In: *IIE Transactions* 46.10 (2014), pp. 1017–1030. DOI: 10.1080/0740817X.2013.876126. eprint: https://doi.org/10.1080/0740817X.2013.876126. URL: https://doi.org/10.1080/0740817X.2013.876126.

[193] Justin Zobel and Philip Dart. "Phonetic string matching: Lessons from information retrieval". In: *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*. 1996, pp. 166–172.

[194] Bowei Zou, Guodong Zhou, and Qiaoming Zhu. "Unsupervised Negation Focus Identification with Word-Topic Graph Model". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 1632–1636. DOI: 10.18653/v1/D15-1187. URL: https://aclanthology.org/D15-1187.

[195] Jia-Yu Zou, Meng-Xin Sun, Kun-Hong Liu, and Qing-Qiang Wu. "The design of dynamic ensemble selection strategy for the error-correcting output codes family". In: *Information Sciences* 571 (2021), pp. 1–23.