University of
Strathclyde
Engineering

Department of Mechanical and Aerospace Engineering

Analysis of SAP work order data by turbine technology type for onshore wind

Author: Erik Salo

Supervisors:

Dr David McMillan
Dr Paul Tuohy

A thesis submitted in partial fulfilment for the requirement of the degree

Master of Science

Sustainable Engineering: Renewable Energy Systems and the Environment

2017

Copyright Declaration

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Signed: Erik Salo   Date: 25th August 2017

Abstract

This thesis is a contribution to the field of wind turbine maintenance management. The first chapters provide a review of wind turbine maintenance management, in particular the motivations for reliability-centred maintenance. The data requirement of this maintenance approach is considered, and the role of work order free text data as an information source are highlighted. Methods from the field of text mining are however required to extract this information in an actionable format, and an overview of the most relevant text mining approaches is given in the final chapter of the literature review.

The main output of this work is a supervised text mining algorithm for structuring maintenance data that is recorded as free text work orders. The method is applied on two datasets of SAP work orders from major onshore wind farms in Scotland. Common issues found in the raw data are highlighted and data cleaning rule sets are developed to overcome these issues. A lexicon of domain terminology is developed that can be used on these datasets as well as extender for wider use. The methodology is developed in Matlab and consists of nine modules for data cleaning, vectorisation, transformations, supervised prediction of missing values. The outputs are given both as a two-level Pareto chart and frequency tables that allows their use in maintenance decision-making. Results are analysed in terms of algorithm performance and validated against the research aims.

Improvements are also suggested to reduce supervision requirement, raise accuracy, and make the approach more universal in terms of turbine models and terminology. Finally, the economic benefits of automated work order mining, and potential ways to increase its industrial appeal are discussed.

Acknowledgements

First of all, I wish to express my gratitude to my supervisor Dr David McMillan for that in full knowledge of the complexity of the project at hand, he trusted me with its fulfilment. I hope the outcome is worthy of his confidence. It would not have been reached without David's support, unstoppable attitude and inspirational speeches regarding the glory that awaits those who complete the endeavours of work order mining, which with this thesis I have only just begun.

Equal regards go to my second supervisor Dr Paul Tuohy for his exceptional kindness and understanding throughout my studies, and for allowing me freely to wander off the beaten track of the Masters programme in pursuit of challenges such as text mining.

I would like to thank Professor Richard Connor for the illuminating conversation and the most valuable hints in text mining, without which my approach would still be little more than medieval.

To my best friends Simmo, Karl, Tom and Marko, I owe particular regards this time for surprising me with capabilities I never knew they possessed through all these years, and guiding me through a thoroughly unexpected crash course in text analytics.

For an entirely non-academic, but all the more special kind of support, I would like to thank young master Julius Hermann. Your (nearly...) constant smiles and cuddles were always there when most missed after a hard day's work, while the rest of the joys of parenting provided an excellent distraction at times when one was needed.

And finally, my dearest Kristiina – throughout this endeavour, you were always a sight for my screen-sore eyes and inspiration for a mind occupied by the woes of misbehaving algorithms. Your lectures of flowery meadows and fairy tales from distant lands made the most delightfully irrelevant diversion I could wish for. I feel utterly grateful to share with you the journey in which this thesis will soon be yet another fondly remembered chapter.

Table of Contents

List of tables

1. Introduction

1.1. Background and motivations

Renewable energy generators have so far been built, and continue to operate, largely with support from some form of financial subsidy or incentive. However, the energy market is changing - subsidies are being withdrawn while the cost of energy from renewable sources is beginning to compete with that of conventional generators. Wind energy in particular has shown the closest energy price competition, as the technology is maturing, becoming cheaper and more reliable (1). This encourages new wind farm developments and provides a level of certainty for a future without subsidies.

Existing wind farms, on the other hand, need to find different means to also remain competitive on the changing energy market. For onshore wind turbines, around 12% of the lifetime cost of energy is from O&M costs (2). Offshore this can raise up to 33%. Minimising OPEX by O&M planning and financial risk management can offer significant reductions in the overall cost of energy production. Risk management is also gaining importance for operators due to an increasing number of assets exceeding their warranty period (3).

Accurate reliability information is essential for maintenance planning, risk management and insurance purposes. For a rapidly evolving technology like wind turbines, long-term operating experience and reliability records, other than those provided by the OEM, are mostly not available (4). Instead, reliability information can be obtained statistically from the maintenance records of existing wind farms. Maintenance data analysis is one of the three key areas identified in (3) to inform decisions that lead to OPEX reduction.

Work order (WO) records are a particularly underexploited information resource in the wind industry (3–5) as well as other industries (6). Analysing historical work orders can provide unique site- and asset-level statistical information, such as the frequency of different maintenance procedures, the spare parts needed, the number and qualification of technicians required (4). Using this data, the operator can then estimate component failure rates, prioritise and schedule maintenance, estimate man-hours required, plan spare parts logistics – in effect reducing OPEX (1,3,5).

However, the inconsistent, semi-structured nature of free text data complicates the data analysis process significantly (7). Standard quantitative methods cannot be applied to such data directly; the necessary information needs to be extracted and structured first (8–10). Operators often consider this process too costly or time-consuming, which leaves potentially valuable information unused (6).

As a result, there are industry-wide efforts to standardise and structure the collection of maintenance and reliability data (1,7,11). A coal mine in Canada has encountered similar problems, and admitted to a poor understanding of what should be included in WO descriptions and how, if at all, that information would be used (7). In this case, a reliability analysis system was used on historical SAP work order data to improve maintenance procedures, followed by successful adaptation of a standardised maintenance management language.

Similarly, many existing wind farms were not built with data analysis considerations - their records can be noisy and may require extensive restructuring before analysis. For example, a major Scottish onshore wind farm operator attempted manual sorting on a SAP work order dataset from a single wind farm with a view to use the results in maintenance planning (12). The process took an estimated two weeks of specialist working time, which was considered too expensive and inconvenient. The company has since left the data unused.

This clearly identifies a demand for an automated solution for processing work order data. Computational methods for extracting information from semi-structured sources for statistical analysis come under the general terms of data or text mining (8,9). The need for these methods in the wind energy sector has been highlighted (3), however specific literature on mining work order data is scarce. Much of existing research, including (7) has been done on a commercial basis, where details of the methods are not disclosed.

The benefits of research into mining work order free texts are numerous. On the one hand, it is a way to lower operator workloads, optimise maintenance decisions and reduce OPEX in existing wind farms. On the other hand, it is an opportunity to build on the knowledge from current systems, leading to increased intelligence and standardisation in the industry overall (13).

1.2. Aims and objectives

The present report is a contribution towards standardising the wind farm work order management system, so that the reliability information it contains can be easily exploited in O&M planning. For standardising the work order system of the future, an insight is needed into the system in its present form, as well as the maintenance demand of wind turbine assets (12).

The primary aim of this work is to explore the opportunities for mining work order free text data, to estimate the maintenance demand of different components.

A second aim is to use the method development stage as a pilot study, to prepare a knowledge base of data specifications and suitable methods for further research using advanced methodologies.

The overall approach to these aims was narrowed into the following objectives:

1. Establish the state of the art in work order mining by a literature review:
   1.1. demands of the industry – input data sources, desired outputs
   1.2. existing efforts to mine work order data
   1.3. methods and solutions
2. Describe the data type in detail based on a training dataset of SAP work orders, and highlight issues in the text mining context.
3. Develop a robust algorithm for mining SAP work order data
4. Apply the algorithm on a test dataset and analyse the frequency of maintenance tasks.
5. Propose:
   5.1. Improvements for WO data collection, including standard list of tasks.
   5.2. Improvements for the text mining methodology.

1.3. Materials

The practical work in this project was based on two SAP work order datasets from modern megawatt-scale turbines, located in major onshore wind farms in Scotland. Both wind farms were operated by the same company. Additional information included the wind farm operator’s asset portfolio.

1.4. Scope

Due to the diverse nature of available data across the industry, it is difficult to develop a universal work order mining application. This project focused on the SAP maintenance management system, which is widely used for assigning and storing work orders.

In terms of text mining methods, the scope is limited to those applications which are considered relevant in the work order analysis context. There are other useful functionalities, such as automated translation, web analytics, sentiment and polarity analysis (8), which were excluded from this work.

In terms of technological coverage, method development was limited to analysing work orders that regard turbine assemblies themselves. Other functional locations within a wind farm, such as turbine transformers, substations, array cables, or roads, were not within the scope. However, many of the terms used in this study are also applicable in those fields, and the lexicon can be expanded for full coverage.

Based on the dataset that was made available for this project, the focus is on onshore turbine maintenance. The efforts required for each maintenance task can be vastly different onshore vs offshore. For example, a manual restart onshore is a simple task, but offshore requires a vessel to be sent out (13). Accessing a turbine offshore requires a stricter range of weather conditions to be met. There can also be structural and technological differences between onshore and offshore turbines, as well as different failure modes due to operating conditions (14). While there are no inherent limitations to using this approach for analysing offshore records, that may require adjustments which are not within the scope of the present project.

2. Role of data in wind turbine management

Focusing on Objective 1, this chapter presents a literature review of wind turbine maintenance management practices. It focuses on reliability-centred maintenance (RCM), which was identified as the maintenance strategy that could benefit most from work order analysis. A summary of general principles of RCM is followed by a description of fundamental decision-making tools, the data requirement of these tools, and finally possible sources of that data, including work orders.

2.1. Reliability-centred maintenance

Letcher (1) presents an excellent overview of the challenges in wind turbine maintenance: "The wind turbine application is uniquely featured by a stochastic duty cycle, which is similar to automotive applications, and an expected long asset lifetime, which is more similar to aerospace applications. It is also characterized by difficult access, remote and regional resources, strained supply chains, and new functional requirements".

After safety, the principal motivation for asset management in a cost-driven energy market is O&M cost reduction (2). Due to the challenges outlined above, state-of-the-art wind turbines are optimised using complex O&M models. These may involve aspects such as prediction of the stochastic wind resource itself; behaviour of the wind turbines in given conditions; the possibility to carry out maintenance in given conditions, and reliability of turbine components (4,15). Maintenance activities are normally managed as part of this wider O&M strategy, through a computerised maintenance management system (CMMS).

2.1.1. Preventive and predictive maintenance

It is generally acknowledged (although not always implemented in practice in onshore wind), that preventive maintenance should be preferred to corrective (2). The higher the consequence of a failure, the more efforts should be invested in preventing it. For example, major assemblies such as generators have a high capital cost and their long lead times cause a high loss of profit in case of unexpected failure. Consequently an operator should implement a condition monitoring strategy, so that the spares could be ordered in advance, replaced before they fail, and the loss of profit thereby avoided. Some of the main arguments concerning predictive and preventive maintenance are presented below:

- Availability. It is desirable for a wind turbine to be in a fully functional state at all times (including when idle due to low wind, curtailment or other operating decision). Insufficient maintenance can cause unexpected failures and reduce availability.
- Spare part logistics. Lead times are associated with ordering most components; on the other hand, keeping a stock requires an upfront investment as well as storage facilities.
- Logistics and cost of equipment such as cranes can cause delays if not planned in advance (2).
- Weather windows (downtime for maintenance should be planned so that it corresponds to low wind, both for minimising lost yield, and due to the strict weather limitations of most maintenance operations.)
- Economies of scale (i.e. timing a replacement so that the same assembly is replaced in several turbines simultaneously, and resolved under one contract (2)).

Ideally a component should be replaced immediately before failure, so as to maximise its useful life, but not experience the uncertain consequences of failure. Higher precision in anticipating failures and timing of maintenance can increase O&M savings. This is shifting operators towards evidence-based decision-making (6,16).

The first approach is condition-based maintenance (CBM), implemented using a condition monitoring system (CMS). A modern wind turbine may have up to 500 channels for data transfer, including controls and sensors (temperature, oil flow, position, voltage, vibration, etc.). Sensor data is collected and processed at a high frequency (on the order of 10 kHz), as phenomena such as vibration occur in that frequency range. However, it is not practical to record the full data stream at this frequency. Instead, the inputs are averaged over a certain period. For condition monitoring, the suitable averaging frequency is up to 50 Hz. Accordingly the CMS can make first conclusions about asset health, and detect anomalies, on that timescale. Yet most major mechanical failures develop over longer periods (weeks, months) and their progress can be followed using longer-term averaging of signals (2).

SCADA (Supervisory Control and Data Acquisition) is a type of control system architecture that is widely used as the interface between a wind farm operator and turbine assets. Information about an asset's condition and operation is collected usually at a 10-minute interval (30 minutes on older systems). SCADA allows the operator to analyse data in more depth, and make operating decisions on a timescale that is manageable for a person. Outputs from SCADA can be comprehensively analysed for correlation between parameters and long-term changes in asset behaviour (2). That makes it an important tool in condition-based predictive maintenance.

Conversely, reliability-centred maintenance (RCM) attempts to predict component failures from previously gathered downtime and failure data (2). It is becoming a mainstream approach to wind turbine maintenance (6).

Although conceptually different, the two approaches are generally used together in maintenance management. CBM provides a more immediate means of predicting failures and reacting to anomalies, while RCM enables long-term strategic decisions (2).

2.1.1. Downtime and loss of profit

The consequence of a failure and the priority of remedial actions is not determined solely by the reliability or failure rate of a given component (17). Each failure and failure mode has a different consequence in downtime and loss of profit, labour intensity, provision of spare parts, and ultimately O&M cost (2).

Redundancy is very limited in wind turbines - most component failures result in downtime, which directly causes loss of generation, and loss of profit. Those failures that cause a longer downtime also cause a higher loss of profit, and from this simple perspective, should be a higher priority to repair. According to (2), the proportion of repair time of a sub-assembly should be equal to the proportion of downtime it causes.

However, the exact downtime due to a failure is difficult to predict. A simpler approach is to rank failure events in classes by the length of downtime they cause. Table 1 compares two such scales. The approach by Rademakers et al. (17) is motivated by the labour demands of a task, but the total downtime may be different due to other factors (labour, logistics, expected delays due to necessary weather conditions, etc). Tavner's approach (2) is technically less discriminative, so that those factors can be combined for a more accurate final expectation. An accurate estimation still depends on reliable historical data for each of the factors.

Table 1. Comparison of downtime event priority scales

Downtime	Tavner (2)	Rademakers et al. (17)
Longest	Category 4: Major replacement	Category 1: Replacement using external crane
	Category 3: Major repair	Category 2: Replacement using internal crane
	Category 2: Minor repair	Category 3: Replacement of small parts
Shortest	Category 1: Manual restart	Category 4: Inspection and repair

2.1.1. Pareto and ABC analysis

The Pareto analysis is based on the "law of the vital few" – the observation that most downtime is caused by a small variety of significant events. Usually around 80% of downtime is caused by 20% of events. Those maintenance events that form the 20% can be considered the most resource-intensive - these require most attention and potentially include the most savings (18). According to Tavner (2), recent work by Faulstich et al. has found that for onshore wind turbines, 25% of events cause 95% of the downtime. This is mainly attributed to major replacements such as blades, generators, drive trains. These observations are used for prioritising preventive maintenance based on O&M cost according to two standard methods: Pareto and ABC analysis.

If the number of failures of a component and downtime per failure are known, then lost profit due to failures of that component during the given period can be estimated. Each failure also adds a certain repair cost comprised of equipment, labour and spare parts. The sum of lost profit and all repair costs is roughly the total cost consequence arising from the particular component in that period. If a cost is calculated in this way for each component of interest, and the values are ordered, then an empirical distribution of costs is formed. By integration (or adding the costs in categories), the categories are found that make up 80% of the costs – according to the assumption, these should form 20% of the total number of events.

In a maintenance planning context, ABC analysis divides a Pareto distribution into three categories. Thereby, the events that contribute the top 80% of consequences form category A, the next 30% form category B, and the remainder are in category C. It is convenient to represent the consequences on a Pareto plot – a bar chart with the bars ordered by value, from which the relative contributions of each category can easily be compared. The strength of the method is that it is straightforward - based on a simple model with a reasonable data requirement. For these reasons, the method is widely used as a link between reliability and engineering data, and business management.

A similar analysis is possible based on another measure of consequence as mentioned in the previous section. For example, Wilkinson et al. (13) used Pareto charts for plotting failure rate and downtime data.

2.1.1. Bathtub curve

The bathtub curve is a widely used model of failure rates over component life span. It gives a prediction of estimated failure rates for a type of component based on knowledge of previous failures. A bathtub curve is generally divided into three stages of component life (Figure 1, where λ stands for failure intensity function or hazard function (which can be simplified as failure rate), and β is the rate of change of failure rate) (2,19). It begins with a "break-in" period of high, but decreasing failure rates; followed by constant failure rate; and "end-of-life" with an increasing failure rate over time. Break-in failures cannot be avoided with maintenance decisions, except for installing used components (which is rarely done except for major replacements, such as generators (12)). Yet there is considerable value in an accurate bathtub curve as a tool to predict failures towards the end of a component's life. The operator can choose to intensify monitoring of certain parts that are nearing their expected end of life, or to predictively replace such parts (2).

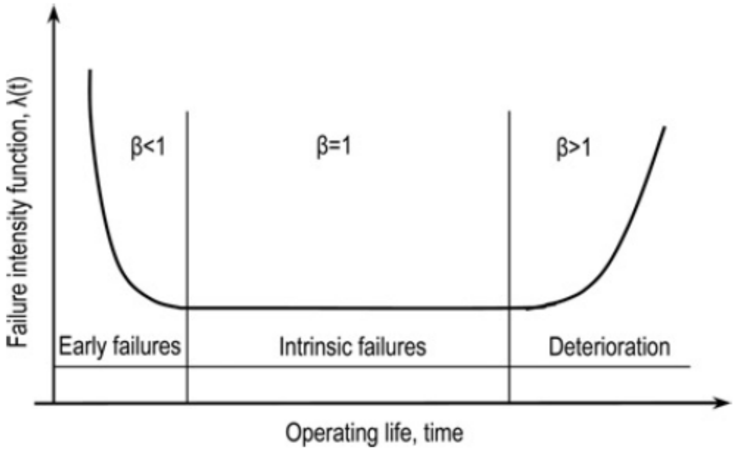


Figure 1. A generalised bathtub curve, adapted from (19)

However, it has been indicated (4) that for offshore turbines, there are many frequently failing systems that do not follow the bathtub curve, and so the entire turbine also might not follow it. The bathtub curve is a model, and needs verification to be reliably used. Constructing a reliable failure rate model such as the bathtub curve is an application where accurate component-specific failure data is essential. Moreover, the data needs to be a time-series, where both the times when a failure occurs, and the failed component's age at that time, are known (2).

2.2. Work order data

Various efforts have been made to create informative databases of wind turbine failure data and allow accurate estimation of failure rates and downtimes of different components (4,13,15). However, these databases are a generic source of data; more value could be obtained from data that has been gathered from the specific wind farm for which maintenance planning is done (3).

On-site data is kept in CMMS systems of various kinds. Differences include component identification standards (6,11,20), database structure, amount of available data (3), use of terminology and abbreviations (6,7), extent of standardised maintenance procedures.

SAP is among the most widely used asset management platforms. It is used for keeping maintenance logs, part orders, work orders. It is a link between the SCADA system, the decision-maker (operator), and technicians (12). A work order is created for every wind turbine maintenance task, to which the person responsible adds a free-text description. Over the years of operation, this generates hundreds of descriptions per turbine, reaching the order of 10^5 per wind farm, which forms a valuable, but underexploited source of information (3).

Work orders are inherently a filtered source of information in terms of significance of the maintenance they represent. All WOs are cases where manual operator intervention was necessary and therefore some resources were consumed, including labour and spare parts. This also corresponds to one of the two criteria that were used to select data for the Reliawind analysis (13). The other – the resulting downtime exceeding one hour – cannot be concluded from the WO data directly.

Work order free texts include particular types of information (such as failure mode, system, subcomponent, maintenance activity) (6). The data type is invariably nominal, and only categorical data analysis methods can be used. This allows few direct inferences. However, value could be added by combining this information with downtime, cost or time series data (21), which in the present context could include a Pareto or ABC analysis or a bathtub failure rate model.

2.2.1. Reference designation systems

Various conventions are used to nominate and refer to turbine components being addressed by a work order. Reference Designation System for Power Plants (RDS-PP) is the international standard that replaced KKS, although around 90% of code letters are the same in both (20). It has also been adapted for wind turbines (11).

In RDS-PP, as well as its predecessor KKS (which was the standard at the time when the wind farms analysed in this work were installed (12)), a component can be described by its of location, functionality, and construction. Each of the aspects form part of component codes. Hierarchically, a wind turbine is classified into Main system (the wind turbine itself); System (such as yaw system); Subsystem (yaw drive); Function (e.g. Drive); Product (e.g. Motor), and components (11).

The ways in which the relationships between wind turbine components are described are not limited to reference designation systems. Especially as the level of detail increases to the product or component level, entities may be present that are not coded under a particular RDS. The KKS system in the wind farms in this study, for example, only covers the hierarchy up to the subsystem level (13 systems with a total of 29 subsystems) (22). Analysis of maintenance records by Wilson (15) was based on the RDS-PP, but also covered only the systems and subsystems. Manufacturers may have more detailed referencing hierarchies in place, but the component codes there tend to deviate from widely used RDS codes.

If work order data is to be used for failure rate estimation, it is important to consider assembly-component-subcomponent relationships in determining their life span. When a major assembly is replaced, the new assembly is usually installed as a whole with new subcomponents, which means premature end of service for subcomponents of the old assembly, although these were still functional (6). Major assemblies have ID codes to simplify tracking, but many components do not (12).

2.2.2. Issues with data

Data quality has been highlighted as the first concern when deciding to use historical work orders for reliability analysis (7). Users often struggle to adopt the CMMS in full extent, resulting in a hybrid system where much of the knowledge is held in the minds of the staff and not systematically stored in data (23). Such “tribal knowledge” of technicians and paper work orders were mentioned as a very ineffective means of WO management, yet very common in the USA (16).

Difference in functional location hierarchies poses a problem for analysis, especially when several organisations are involved (6). The lack of a universal, comprehensive wind turbine specific referencing system has been recognised before (15). Furthermore, the Reliawind project proposed a standard wind turbine taxonomy (13).

An example from the mining sector is presented in (6), where four of the seven companies involved only specified functional location codes down to asset level, while a detailed code hierarchy was available to specify down to the actual component maintained. For those companies where more detailed codes were used, up to 37% of WOs for a particular asset type were recorded without component-level codes. This proportion of falsely identified work orders means that functional location codes alone are not a reliable source of information for estimating the distribution of maintenance activities. More specific data could be found from free text to associate WOs with their actual functional locations.

3. Data and text mining

This chapter continues the literature review for Objective 1. It uses the previous conclusions about the information that is available in WO free text fields, and explains a selection of methods that enable the information to be extracted. The list is not fixed, nor complete: within each stage, different methods may be chosen and combined; steps may be added or removed depending on raw data and output requirements.

3.1. Purpose of mining work orders

Work order data mining, text mining, and data mining of free texts, with notable variation in the use of these terms, has been suggested and used as a source of valuable reliability information in various cases:

1. Data mining techniques based on word associations have been used to extract reliability information from WOs in a coal plant (7).
2. Mining mixed data sources, including WOs, can help associate failures, causes, repair time and downtime (21).
3. Hodkiewicz et al. (6) mined WO free texts from heavy machinery maintenance, to structure missing and problematic data. They found the approach to offer significant advantages in maintenance planning, but indicated that progress in the field is limited because the literature seldom discusses the methods in detail.
4. McMillan et al. (3) highlight WO mining as one of the key areas of OPEX reduction in the wind energy sector.

Text mining deals with the analysis of text documents. It builds on the principal functionalities of data mining; the main distinction is that while data mining deals with structured data, text is considered semi- or unstructured information, and is thereby more difficult to process. The added capabilities of text mining will be covered below, including recognising natural language elements, semantic meaning and grammatical structure. The purpose of text mining is to generate actionable outputs – structured, analysable using computers, or more easily comprehensible for the user than the original dataset (24).

A dataset on which text mining is performed is known as a document collection. Each individual body of text that is analysed is known as a document, the size of which can range from a web site or a book to single post on social media or a chapter or sentence within a book (25). However, the analysis itself usually takes place within documents - the basic unit in most text mining systems is a character string (such as a word, phrase, or other similar entity), usually known as a token. Documents are represented as collections/vectors of tokens. In this work, a WO dataset from one wind farm is considered a document collection, each WO free text is referred to as a document, and the term "token" is used for words or their numeric representations (9).

3.2. Data cleaning

One of the most basic necessities in text mining is data cleaning. It involves the removal of excessive language elements such as punctuation or uninformative words. Processes for correcting spelling and other errors are also included. However, not all of these steps are performed by a single method or algorithm, and they can be distributed over the course of the data mining process (9,26).

Hodkiewicz et al. (6) used a rule-based approach for data cleaning, also known as data transformation. With this approach, a set of rules are usually sequentially applied to the data within an algorithm. An indication of the complexity and importance of data cleaning is that 407 different rules were used, but conflicts still remained. In a rule-based approach, not all necessary transformations can be anticipated and covered by specific rules; a manual verification step may still be necessary in the end of the cleaning process (6).

If any analysis of existing data is undertaken, the results should be used as lessons for improvement, especially in the data cleaning stage. Cleaning brings all relevant data quality issues into focus. A rule-based approach makes it easy to identify their causes (rule-by-rule) and avoid the same issues in the future; otherwise, data cleaning will remain a constant work (6). Any conclusions about data quality issues should be acted upon – either by somehow compensating for problematic data, or by leaving it out and taking the resulting errors into account.

In the English language, the most frequent words are function words such as articles, or stop words, which carry little meaning. These are usually removed in the cleaning stage, both as a means of dimensionality reduction and to reduce noise that may affect later methods. Additional selective procedures can be undertaken to identify and remove less important words and increase the concentration of informative ones. Some classifiers, for example, remove 90-99% of text features without degradation in performance (25).

Some applications, on the other hand, rely on stop words as delimiters that separate content words and semantic classes or help to distinguish terms. The methods that follow need to be considered before removing stop words.

3.3. Tokenisation

Tokenisation involves dividing the documents into the basic units – words or terms, collectively known as tokens (8). Geometrical methods depend on a vector representation of each document - tokens need to be translated to numeric values so that vector calculations can be performed. This can be done by coding – assigning each term to a numeric token – or by a binary representation such as those in section 3.6 (8,9). Depending on data quality issues present, tokenisation is often required already to perform some of the tasks of data cleaning – an example is stop word removal. For syntactic and semantic analysis of documents, tokens can be associated with several layers of information, including part-of-speech tags, categories or association trees. The assignment of such properties partly overlaps with the lexicalisation stage described later in section 3.5.

3.4. Collocations and term extraction

Majority of technical terms found in dictionaries are compounds of more than one word (27). When mining for lexical terms, it is important to capture the term as a whole entity, because omission of any of its component words would usually refer to a different entity.

According to Evert (28), a co-occurrence is the presence of words in a text within a certain distance of each other – a statistical term without any semantic background. On the other hand, he defines: "A collocation is a word combination whose semantic and/or syntactic properties cannot be fully predicted from those of its components, and which therefore has to be listed in a lexicon."

This inevitably involves careful manual work. Any two adjacent words in a text form a co-occurrence – it is not reasonable to assess and lexicalise or discard each one manually. Also the frequency of word pairs alone is not sufficient to determine that two co-occurring words are associated. For common words, this might only be by chance. Instead, various association scoring techniques are used to reduce manual work by ranking co-occurrences, high potential ones first, or removing those that are statistically not meaningful (28). This falls among the applications of automatic term recognition (ATR).

3.5. Lexicalisation

Lexicalisation of terms or tokens is a way to involve background, or domain, information in the text mining process. In text mining, domains are areas of interest that form the scope of individual mining applications (25). Some text mining approaches have been developed for universal (cross-domain) capability and depend on a wide array of information (dictionaries, thesauruses, word almanacs, standard text corpuses), which is used as a background to process the task at hand. These include powerful algorithms used for knowledge discovery within large databases, search engines, automatic translation, etc.

On the other hand, a narrow domain allows to create and manage domain-specific knowledge bases such as lexicons, taxonomies and attribute relationship rules with relative ease (25). If highly specific technical terms are present, then some lexicalisation might be inevitable for effective information extraction or classification, as shown in the previous section.

“Determining what activity has taken place or failure mode was observed the ability to relate a variety of phrases to a common term is required. For example the following examples are all associated with a replacement activity: C/O, Change out, change out, CO, change out, swap, replace, and rep” (6). In a lexica, equivalents such as synonyms or alternative spellings are normally lemmatised – semantically grouped under a single representative key word, or lemma (9).

3.6. Models of the document collection

In preparation for further analysis, the documents and terms need to be represented in a consistent way, i.e. according to a model. Most models can be divided into two major categories: statistical and geometrical. Statistical models represent the documents based on observable regularities: term frequencies, probabilities, and other statistical inference. Statistical models depend on a good representation (number of occurrences) of terms, and are therefore suitable for larger documents (28). Geometrical models represent documents in a vector space, where they are compared based on distances and angles (8). Models that do not consider word order are known as bag-of-words models.

3.6.1. TDM

The term-document matrix (TDM) is a matrix where each row corresponds to a term and each column to a document (8) (or vice versa in other sources (25)). In a binary TDM, the presence or absence of each word in each document is noted by a binary 1 or 0, respectively (9). This translates the document collection into vector space: term space represented by row vectors, and document space represented by column vectors. Such representation provides a good overview of the document collection and its vocabulary. Based on non-zero values in the appropriate row or column vectors, it is easy to retrieve terms that are present in a given document, or documents that include a given term (8). Being in the vector space, the TDM classifies as a geometrical model, and allows document distance measures to be calculated directly. In addition, the TDM representation can be used as an input to learning and classification algorithms (25).

TDM is a “bag-of-words” model, and consequently does not carry word order information. However, Banchs (8) notes that this mostly does not affect semantic analysis, such as information retrieval and document categorisation.

3.6.2. TF-IDF

Insensitivity to the number of term occurrences can be compensated by replacing binary TDM with a weighted alternative.

The simplest approach is to replace each binary 1 with a count of all occurrences of the term in the document (or the entire collection) known as term frequency (TF). However, this is likely to highlight mainly stop words. To make practical conclusions about a document, the weighting should instead reflect how much unique or contextual information each word carries.

Inverse document frequency (IDF) is the ratio between the number of all documents in the collection and the number of documents that contain a term - i.e. inverse of the proportion of documents that contain the term. The rarer a term, the more unique is a document that includes it, and the higher its weighting (8,25). Mathematically, rare terms are the most informative in a dataset and can be used to describe or set apart particular documents (8).

However, as thoroughly discussed in (28), statistical inference based on rare terms is meaningless - a model or distribution fitted to a very small sample cannot be accurate. Document collections often include a large proportion of terms that occur only once, especially if multi-word compounds are treated as individual entities. Based on single or rare occurrences, it is not reasonable to draw conclusions about the remaining dataset, or probability of the term occurring in other collections.

The product of the two weights – known as TF-IDF weighting – has been described as the optimal weighting scheme. It applies a relatively higher weighting in the middle part of the term frequency scale, which reduces the complications of applying TF and IDF individually (8).

3.7. Dimensionality reduction

The so-called “Curse of dimensionality” applies in text mining. A TDM is mostly a sparse matrix (one with a large proportion of zeros), because not all documents contain all terms in the vocabulary, and vice versa (8). A large matrix takes considerable time to process, and therefore it is desirable to carry out analysis (e.g. construct a TDM) with the minimal possible set of terms (9). A smaller dataset is faster to process, easier for the operator to verify, reduces clustering complexity. A straightforward means of dimensionality reduction is lemmatising, which was described in section 3.5. A normalised TDM, such as a TF-IDF weighted one, can also save processing time compared to one where direct occurrence counts are used (9).

3.8. Similarity, clustering and classification

Presence of common words in different documents is a fundamental part of similarity assessment, based on which documents can be grouped (26). However, semantically equivalent documents do not necessarily need to be composed of exactly the same words. Various similarity measures are used. Although geometrical measures usually compute the distance, or dissimilarity, similarity is simply the opposite of that value: either *s=1/d* or *s=1-d* (*s – similarity; d – distance*) (8).

Firstly, similarity measures can be used on the character level, which is useful for identifying and bypassing spelling errors. According to Banchs (8), the most common of these is the Levenshtein distance, which he defines as “the minimum number of character editions that are required to transform one of the strings into the other” (p 62). Therefore the distance is given as a discrete measure. There can be three types of editions:

- insertions – adding a new character in any position in the string
- deletions – removing a character without replacing it
- replacements – substituting one character in a string with another one in its place

- transpositions - switching the position of two characters – is added in the Damerau-Levenshtein distance. Together these four editions make up over 80% of spelling errors in search engines (29).

Other measures are used to compare entire documents on the token level. The Dice and Jaccard coefficient both give a normalised count of common non-zero elements between two vectors (despite different normalisation factors) (8). Both represent a binary dependency (present or absent) on non-zero elements, not the absolute values of these elements. That makes them suitable for comparing nominal variables, including text represented as a sequence of tokens (26).

Clustering and classification are approaches for grouping documents or terms according to a similarity measure. In classification, the classes are predefined – it is therefore a supervised approach. Clustering, on the other hand, is unsupervised – it works solely based on database contents, does not require external knowledge base or prior knowledge of documents (9).

Unsupervised clustering algorithms define clusters during the process. Much less user input is required and it is easier to adapt between fields the results may be unexpected and need to be confirmed by an expert. Domain knowledge and familiarity with the dataset are essential at this stage (26,30). Supervised methods are more case-specific and require manual inputs. However, they can generally be tuned to a higher accuracy as long as they are applied within the context for which they are developed. Supervised methods are also better suited for small datasets, large proportions of unique or rare entries and noisy data.

Both clustering and classification are implementations of machine learning, a form of artificial intelligence. Machine learning is the application of computational methods to perform an inductive process. In text mining, it can also be used for data cleaning (8,25). Machine learning algorithms perform best on large amounts of data. The data needs to include a training set – a subset which is representative of the whole dataset, to allow the algorithm to learn the association rules between known values, before predicting the unknowns (8,9).

A suitable dataset is not easily achieved with unstructured data such as CMMS, where up to 40% of entries are unique. In such cases, different types of supervised methods can be used, known as rule-based algorithms. In these, the algorithm performs transformations that have been fully defined by the operator (6).

3.9. Output analysis

The previous steps in the text mining workflow are means to extract certain information from the sources. There is usually an added interface through which the user can exploit the information gained – a statistical analysis procedure or a data visualisation step (9,25). Alternatively, results such as common terms or key words could be associated with the documents and stored, so that they can be queried later. For example, most search engines and scientific databases use this functionality to retrieve quick search results (29).

4. Overview of SAP work order data

This section aims to give a detailed description of SAP work order free text data in preparation for text mining method development. In accordance with Objective 2, the purpose is to highlight any problems that could affect the choice or effectiveness of text mining methods. Each of the following paragraphs discusses a category of problems found in the data, while characteristic examples are given in Table 2.

Table 2. Common issues identified in the training dataset, with real examples from the text.

Sparsity	• Field length limit: 40 characters • Poor grammatical structure • Both excessive and lacking punctuation present
Term frequency	• 41.8% of tokens were unique • 4657 tokens per 12425 free texts
Spelling mistakes	• 25 different ways to spell “replace” • 18 different ways to spell “investigate”
Duplicates	• “Fault finding ... fault” • “Failure” and “fault” • “Failure” and “error”
Homonyms	• “Stop” – activity or failure mode • “Seal” – activity or part (gasket) • “Grease” – activity or lubricant
Compound	• “Gear Box “

mistakes	
Abbreviations	• G/BOX or BOX meaning “gearbox” • “SWGR” meaning “switchgear” • “Inv” meaning “Investigate” or also “Inverter”
Ambiguity	• “Oil error” • Circuit break; thermal break; yaw break

4.1. Short, sparse texts

Due to sparsity and noise, the grammatical structure of WO free texts is absent or distorted. This limits the use of existing NLP applications. Part-of-speech recognition could help with classification (verbs represent mostly maintenance activities, nouns mostly part names). But cannot be used to full advantage, as word associations and hierarchies become ambiguous under a poor grammatical structure.

Short and sparse texts have become one of the research fields in text mining due to the rise of social media and instant messaging. Issues with SMS and tweets also apply here: both these forms of communication suffer from abbreviations and ungrammatical expression.

4.2. Noise

In case of the training dataset free text fields, noise appears in several forms, such as spelling mistakes, excessive punctuation, unnecessary words, duplicate words, etc. Across the dataset, there was a wide array of issues such as misguiding component references, mismatch between component description and functional location code.

4.3. Missing and incomplete data

In the training dataset, empty cells were not a major concern, although some were present (only one WO was excluded as a result). However, partially incomplete data might be harder to detect and treat than empty cells. Excessive use of stopwords is a common example where a field appears full of information, but little of it is significant or associable to a particular activity or component. WO data is often incomplete, inconsistent, includes missing fields – the extent of the problem varies, so that there is no standard effective approach (7). Hodkiewicz confirms this by stating that for rule-based cleaning approaches, rules mostly need to be recreated for every separate case (6).

4.4. Duplicates

WOs carried out over several days may be represented as duplicates for each day, but still represent one activity to repair one failure (6). Duplicates were also present within fields. Repetition of (nearly) equivalent terms within text fields was a common practice. This was caused by limited information being available to the operator at the time of creating the work order. When a fault code is raised through SCADA, for example, the corresponding part or sensor is often all that is known about the fault, and therefore it is common to include both of the most likely activities in the work order (22).

4.5. Multiple senses

Across the entire dataset, there were certain cases where a token could represent different taxonomical entities at different levels of hierarchy. For example, in case of "yaw gearbox", it was important to recognise the functional location as the yaw system, not the gearbox; the gearbox in this case was the subassembly.

Rule-based approaches are not well suited for handling sense ambiguity. When a word appears in a document collection in several senses, it invokes a conflict between rules for each sense. It is important to recognise such conflicts and bring them to the operator’s attention for manual resolution to avoid false interpretation (6).

4.6. Spelling and abbreviations

The numerous spelling mistakes for many words in the texts indicated that a standard dictionary was not sufficient to capture all variations of each term. Spelling mistakes can make tokens harder to match, and can additionally add a level of ambiguity to the meaning.

Due to the 40-character limit, longer terms were often abbreviated to a stem, mostly the first 2-3 characters. A stem this short can be common to words of different meanings. Sense disambiguation can be used as an additional step to distinguish the correct meaning semantically, but a straightforward matching of stems to dictionary terms is in this case ineffective.

5. Text mining algorithm

This chapter is concerned with the development of the text mining methodology, in short concluding Objective 3. The first section sets the overall approach and functional requirements that motivated the process. It is followed by more detailed sections for each individual stage or functionality, where each decision is explained with regard to the data description and literature review above. The completed methodology is illustrated by a flow chart in the end of the chapter (Figure 2) and complete scripts and custom functions have been included in Appendices 1 and 2, respectively.

5.1. Approach

5.1.1. Methodical goals

Challenges in work order free text mining due to data type have already been discussed in the previous chapter. In addition, the following methodical demands had to be considered throughout the process.

5.1.1.1. Required outputs

In the present work, the aim of the text mining process was to structure the information to allow categorical data analysis. The focus of the analysis was on corrective maintenance, and was also limited to turbine assets, which were highlighted by MacLean (22) as requiring most attention. The following requirements were considered, in order for the outputs to be usable as described in Chapter 2:

- Different aspects of the task needed to be extracted, most importantly maintenance activity, component, and failure mode.
- Accurate frequency information was required for each term in these main classes.

5.1.1.2. Precision

Modern text mining systems mostly work with large collections (on the order of 10^6) of full text documents or web sites. When the length of a document exceeds a paragraph, terms and phrases commonly start to repeat. That means there are several occurrences based on which an algorithm could draw conclusions about the context of the document, and associate it with a few outputs. Semantic ambiguity can be reduced with statistical measures. This is loosely known as a many-to-one approach.

According to Hodkiewicz et al. (6): “Extracting the output fields from the Shorttext field is a one-to-many mapping rather than the many-to-one mapping that machine learning systems excel at.” Here, the number of words in a work order is roughly similar to, and often smaller than, the number of semantic classes they represent. There is little additional information for ambiguity resolution. Therefore a rule-based method is better suited in the WO application.

The classification success rate for each term – not only each document as a whole - needs to be high in every instance.

5.1.1.3. Adaptability

The desired output required a classification of terms, some of which follow a hierarchical relationship in practice. However, as described in Chapter 2, component hierarchies and taxonomies are not comprehensive, or universal. Therefore the aim was to develop an inductive approach, which does not depend on previous knowledge about the dataset, but uses information contained in free text records themselves as an alternative to RDS functional location codes, to establish how the terms in each maintenance task are related.

5.1.2. Development workflow

Method development in this study was a heuristic process carried out in the stages below. The algorithm was developed in a modular layout, so that steps could be added to it later.

1. Empirically review the problem
    a. Data input
    b. Desired output
2. Identify necessary steps
3. Identify data issues in each step
4. Choose appropriate methods
5. Test methods on small sample of data to identify issues
    a. Record conflicts
    b. Either choose other method or adjust
6. Verify on training dataset
7. Apply on test dataset
8. Analyse results and generate outputs

5.1.3. Assumptions

Method development was based on the following assumptions:

1. Work orders and all words within them are in English
2. Every free text describes a single work order and single objective.
3. Every WO describes work done on a single turbine – repeated tasks are assigned a separate WO for each turbine.
4. Work orders are treated independently: if a common cause failure occurs in different components, they are considered different failures.
5. In case of a replacement at a high hierarchical level, all necessary subcomponent replacements are included in the high level procedure.

5.1.4. Software

5.1.4.1. MATLAB

There are specialist tools and programming languages developed specifically for complex applications in text and data mining. However, not all capabilities of specialist text mining tools are necessary in this application. Moreover, these tools are not commonly available, and would need to be obtained, likely with a financial cost as well as a learning period for the user (25). Therefore using an existing software instead would make the work order mining process more widely accessible. Matlab is a familiar and widely used software throughout the engineering sector, and is capable of the tasks (8). Another advantage of Matlab is its effectiveness at data analysis and reporting, which can be used after the data/text mining stage, without the need to switch between software.

5.1.4.2. MS Excel

Simple manual manipulation of two-dimensional data is more convenient in Excel than in Matlab. For ease of access, inputs to the algorithm as well as outputs were stored as .xlsx files whenever possible. When data manipulation by the user was necessary, data were exported from Matlab to Excel, handled as required, and then imported to Matlab to continue analysis.

5.2. Data cleaning and filtering

The first step after gathering inputs and loading files was filtering. Upfront filtering was found to significantly reduce processing time. By default, all WOs where the raw data contained an empty field in any of the necessary columns, were filtered out. Additional filters could be specified by the user. The maintenance type filter allows to enter SAP abbreviations for the maintenance types of interest. In this work, MCP (Minor Corrective Maintenance), MCG (Major Corrective Maintenance), and MRL (Local Reset Maintenance) were entered, as the main focus was on corrective maintenance (22). The work area filter allowed to select which of the following - turbine assets (A), transformers and switchgear (T), or site infrastructure (C) - should be included in the analysis. This was set as A in the present work.

Finally, the turbine model corresponding to each asset was retrieved, and the user could then select ones of interest. The size of the example datasets was considered insufficient to allow for statistical inference for individual turbines. All turbines of each site were included, and since only one model was used per site, the different datasets also represent different turbine models.

In the cleaning step that followed, syntactic issues such as excessive punctuation or whitespace were resolved first. 10 rules were constructed and maintained directly within the algorithm. This approach was chosen due to numerous conflicts that required the rule sequence be carefully maintained.

In addition, stop words were removed, lists of which were maintained in .xlsx tables. Stop list was based on (31). Some words that were meaningful but not important for this analysis, for instance qualifications of the technicians involved in the work, were also removed together with stop words. A separate list was made for tokens whose meaning could not be identified, including severely misspelled words or abbreviations. The two types of unimportant words were separated, so that depending on later analysis, the removal of meaningful stop words could be switched off in the script, while the unidentified words are still removed.

A separate step was added to treat words where spaces may be missing. All words were compared to pairs of tokens in the lexicon. If it was found that two known tokens make up a word that had been found, the user was prompted to decide whether to split that word into those two tokens. If the choice was made to split, then this was recorded in a .xlsx table, so that the same decision could be repeated without prompting in the future.

All data cleaning steps took place within the “CleanTokenise” script (Appendix 2, **Error! Reference source not found.**Figure 2), using custom functions from Appendix 3.

5.3. Lexicalisation and tokenisation

The present work is concerned with a particularly well-defined domain that can be described as the domain of “wind turbine maintenance operations”. This gave two main arguments for choosing a lexical approach. Firstly, a narrow domain meant the size of its vocabulary was also limited, and could be lexicalised with reasonable effort. Secondly, the dataset contained domain jargon and specialist terms, which could be more easily resolved by including background knowledge.

Some major works such as main bearing replacements take place very rarely, but mean a significant downtime and expense for the wind farm operator. It is important to capture rare terms as well as more common ones. The method should not depend on a good statistical representation of every term. In coherence with Evert in chapter 3.4, a suitable statistical approach was not found during literature review. Common methods such as TF-IDF (8) are ineffective for term recognition in case of very rare terms. Similarly, the algorithm by Justeson (27) requires at least two instances of a term, and more to increase the probability of identifying significant terms. Therefore an approach was chosen where all terms need to be manually listed for matching.

5.3.1. Mistakes and abbreviations

Various ways of abbreviation and spelling mistakes were commonly used in the documents, some of which required careful comparison with full terms to be recognised. One of the approaches used to deal with such issues is known as stemming. Stemming reduces a lemma to the fewest first letters by which it can still be uniquely identified, and all words beginning with that stem are then assigned under the same stem (9).

However, stemming algorithms may struggle if terms appear in a non-lexical form – this includes prefixes and suffixes, and also mistakes (9). Considering the extent of mistakes in the present vocabulary, some words would need to be reduced to a stem of only 2 to 3 characters to avoid mistakes within the stem. That, on the other hand, means that more than the intended term may begin with the same characters as the short stem. Some such cases could be solved by allowing stems to be longer, and allowing their matching within a certain Levenshtein distance. Importantly, among the matches within 1 Levenshtein distance to a stem, is the stem 1 character shorter. For a stem length of 4-5 characters, a distance higher than 1 would therefore be impractical, as the above issues concerning short stems would arise. On the other hand, longer stems start to approach term length, in which case the usefulness of stemming is reduced, and full terms can be matched instead. The use of the Levenshtein distance for terms is similar to stems. For longer words, a higher distance could be allowed. However, words or stems with more spelling differences than the allowable distance can still not be matched.

It was considered that for the present dataset, stemming and Levenshtein distance measures were unreliable due to excessively noisy data. This further justified manual lexicalisation and lemmatising of all variants of each word, to increase singular identification of terms.

5.4. Term classification

According to the output requirement set in 5.1.1.1, term classification was motivated by the distinct types of information that free text terms represented. The aim was to convey the maximum amount of information with the minimal number of classes. Not all possible hierarchical relationships were therefore registered as a separate class. Similarly, not all parts of speech were separately tagged.

The approach was to classify all terms that were discovered, but not to restrict a class to the terms of a predefined taxonomy. This was due to several reasons: 1) Different turbine models can consist of different components; 2) Semantically equivalent ideas can be expressed in vastly different ways; 3) the use and range of domain vocabulary can change between datasets; 4) failure modes and their frequencies vary depending on particular components and site conditions.

During initial testing of the algorithm, particular types of information were clearly distinguished, that largely overlap with (6,21). These were: maintenance activities, failure modes, component codes, and component names. Some less discriminative terms were also tokenised, which were classed as supporting details, or parameters and values, which could be used to identify sensors or further specify failure modes. The full list of classes is shown in Table 3.

5.4.1. Component taxonomy

Exact component taxonomy may differ depending on the RDS used by the turbine manufacturer, on a given turbine model or year of manufacture. If an operator’s maintenance databases were based only on functional location (part) codes, then a translation would be necessary for them to be analysed on an equal basis. For example, when one manufacturer considers the pitch system to be part of the blade system, but another considers it a part of the hydraulic system, then a dictionary could be used to associate the two.

However, a one-for-one translation of part codes may not always be possible, as technologies are constantly evolving, and not all models use the same functional components. Newer models tend to have a more complex control system with numerous sensors and more power electronics that did not exist in older turbines. The increasingly wide use of direct drive machines with permanent magnet generators means yet another change of component taxonomy at a very high (system and subsystem) level.

The advantage of free texts is that they do not follow any particular RDS, so the extracted information can be classified directly according to any taxonomy. Instead of a predefined component map, the present study assumes a hierarchical relationship between components, but leaves the exact taxonomy to be defined by the information obtained during the process. Han et al. (26) call this a schema-level specification, and compare the level of predefinition to classifying streets, cities and countries.

Table 3. Term classes with arbitrary examples drawn from text (the whole list does not represent a single WO).

Class number	Class name	Arbitrary example
1	Activity	Investigate
2	System	Yaw
3	Subsystem	Gearbox
4	Basic function	Cooling
5	Product	Fan
6	Component	Sensor
7	Failure mode	Failure
8	Parameter/type	Temperature
9	Part code	FG008
10	Other	Outside

It is important to note that the hierarchical levels are described by different terms depending on source. The present work adapted the hierarchical terminology from RDS-PP (11) with the addition of the component level, based on examples from Tavner (2) and the KKS system (20). Importantly, while Tavner uses the term "system" to mean the entire wind turbine, then in RDS-PP, the wind turbine is a "Main system". Consequently, "system" in this work means the level that accommodates the generator, rotor, etc. Another distinction is that the hierarchy in this work consisted of 5 levels below the wind turbine itself, compared to 4 used by Tavner (2). The difference was mainly in the lower two levels, where components (cables, hoses, bolts) could in this way be distinguished from products (motors, fans). A full hierarchy with examples is available in Table 3.

5.4.2. Supervised classification

After rule-based transformations were completed, a large proportion of entries remained where the system was not named. These could be fully elaborate (subsystem-function-product-component); only a high-level value (subsystem); or a low-level value (e.g. product, component) without specifying the higher taxonomical units. Missing taxonomical values could usually be estimated empirically by a person with sufficient background knowledge. It was desirable to develop a text mining method that is equally adaptive. The assumption was made that all named parts follow a hierarchical relationship. A system could be determined based on its constituent parts; a more accurately defined list of parts would result in a more accurately defined system. Some systems also have characteristic failure modes or maintenance activities.

Such predictions can be made using a supervised or unsupervised classification algorithm. In the present case, examples of correct component-to-system mapping were available in the form of well-worded WOs that contained both system and component names. It was assumed that all system levels present in the data were also mentioned in these examples. With possible levels (response classes) thus defined, a supervised algorithm was preferred.

The Matlab Classification Learner was used to compare algorithms and select the optimal solution. The KNN (k Nearest Neighbour) algorithm showed the highest overall cross-validation accuracy in preliminary tests with the same dataset (followed by complex decision tree), and was therefore chosen as the default. This choice was confirmed by the fact that the KNN algorithm was listed in several sources (9,26) as among the most widely and successfully used classifiers. The following parameters were then determined as offering the highest accuracy estimates on the training dataset, and were used in every case of the final analysis: Hamming distance; squared inverse distance weighting; 10 nearest neighbours; 10-fold cross-validation. Prediction accuracy during individual analyses was estimated based on the confusion matrix provided in the toolbox.

5.5. Rule sets

The proportion of unclassified or falsely classified tokens was reduced during development by creating new rules. A heuristic, rather than methodical process was chosen for this due to dataset size and large proportion of unique entries, which were found to cause unexpected rule conflicts and invalidate any methodical assumptions about the dataset. Rule-based transformations were carried out after converting to document vectors. Each rule was created based on the following sequence:

1. Data issue heuristically identified
2. Information gathered about the issue:
    a. frequency: unique/rare/common?
    b. type: syntactic or semantic issue?
    c. variations: does the issue appear in one common form that can be solved by a single, or are there conflicts?
3. Syntactic issues were documented and resolved together during the cleaning phase; semantic issues were considered as follows.
4. If a semantic issue was unambiguous and corresponded to a known construct, a rule was added to the appropriate .xlsx rule table (sections 5.5.1 to 5.5.4).

This was the simplest case and was done regardless of its frequency, for example when any two words were found to form a compound word every time they co-occurred.

5. If the issue was found to be representative of frequent cases, but not a known construct, a new rule type with an accompanying .xls table was created, employing a different algorithm (sections 5.5.1 to 5.5.4).
6. If a rule conflict could not be avoided or would have required more than two conditional rules, the case was left untreated. Leaving rare cases of raw data was preferred to false transformations.

5.5.1. Compounds

Rules for identifying compound terms were implemented before other major transformations: if any component words were modified by other rules, the compounds would not be identified correctly. Terms that are correctly spelled as closed compounds were lexicalised as normally, but a reference rule was created that connects its two parts, should they be mistakenly spelled as a loose compound. For loose compounds, surrogate tokens were created where the parts of the compound were separated by a dot (as demonstrated in Table 4). The rule then exchanged the separate tokens in the TDM for the surrogate. This effectively consumed all words that formed the compound, so that they would not be considered as part of a compound and their individual meanings at the same time. 123 rules of this class were used altogether, with examples given in Table 4. In addition, this syntax could be used to reduce ambiguity in the way demonstrated by changing “blade pitch” to “pitch” in the table. Compound rules were applied as part of the “RuleTransform” script (Appendix 2, Figure 2).

Table 4. Examples of rules for joining compound words and removing excess words

IF This word is present	AND This word is present	AND This word is present	THEN Replace all with
Low	Speed	Shaft	Main.Shaft
Ball	Bearing	-	Ball.Bearing
Blade	Pitch	-	Pitch

5.5.2. Mapping of equivalent terms

Two approaches were used to treat equivalent tokens. Firstly, obvious cases were lemmatised at the lexicalisation stage. Secondly, if there was additional need to assign one lemma under another, then mapping was done directly in the lexicon. A separate column was created in the .xlsx file that contained the lexicon, wherein references from one lemma to another could be manually added in the form of token numbers. Synonym references were then resolved within the "RuleTransform" script (Appendix 2, Figure 2).

5.5.3. Sense disambiguation rules

These rules were used for words whose meaning varied depending on collocated words, or also the absence of certain other words. Altogether, 14 presence-based rules were used in the form presented in Table 5, and another 4 absence-based rules in the form shown in Table 6. Both sets of rules were applied within the "RuleTransform" script (Appendix 2, Figure 2).

Table 5. Rules for words whose meaning depended on other words present

IF	AND	THEN
Word 1 is present	Word 2 is present	Word 2 is replaced with
Circuit	Brake	Breaker

Table 6. Rules for words whose meaning depended on the absence of certain words

IF	AND	THEN
This word is present	This word is NOT	Word 1 is replaced with
Sonic	Anemometer	Sonic.anemometer

5.5.4. Classification rules

In addition to the previous rules where tokens were replaced, there were another 33 rules which enabled tokens to be reclassified, as shown in Table 7. Since the TDM terms had not been classified at the time when the first rule-based transformations were carried out ("RuleTransform", Appendix 2, Figure 2), classification rules were applied as a separate step later ("Reclassify", Appendix 2, Figure 2).

Table 7. Rules for reclassification

IF		(Optional) IF		THEN		(Or) THEN	
This class	Includes	This class	Includes	This class	Is set to	Then this word	Is changed to
(any)	Anemometer	-	-	System	Meteorological	-	-
Failure mode	(is empty)	Activity	Stop	-	-	Stop	Failure mode

5.6. Model of document collection

The TDM model of the document collection was used, as it could facilitate rule-based transformations, and be used for categorical data analysis later. Its limitations as a bag-of-words model were acknowledged: the method distorts any grammatical structure and positional information in the text, so that grammatical analysis of its outputs is not meaningful. However, this limitation was not significant in the current context, as previously confirmed by Arif-Uz-Zaman et al. (21). The grammatical structure and word order in the free texts were already inconsistent and difficult to exploit. Most words occurred only once per documents, so the extent of frequency information wasted was minimal.

5.7. Conflict resolution

There were examples where several terms of the same class were present, allowing the WO to be classified in multiple ways. According to the assumptions made in section 5.1 that each WO represents a single task, such conflicts needed to be unanimously resolved. Each entry had to be reduced to a single failure mode, single activity, and a unanimous component identification. This was particularly necessary in the system class, which carried most importance in the output analysis.

There were three main considerations when choosing a solution:

1. It was necessary to identify the most important or most discriminative term;
2. It needed to apply across the dataset, regardless of which terms were causing each conflict;
3. It had to be adaptable to other datasets, where different terms may be important.

The most basic approach was that if there were several tokens present in a cell, only the first one was left, but this was considered unreliable.

A user-defined term hierarchy would offer reliable solutions to the first two conditions, but not always the third. Also a user's judgement may be biased depending on which cases are taken as an example to determine where a word stands in the hierarchy. Using examples from the dataset, "Error" is a meaningful word compared to "not OK" in one document, but is definitely less informative than "Overtemperature" in another.

Computational approaches can provide a more neutral ranking. In this word, the TF-IDF ranking (calculated across the entire dataset, not individual documents) was used to determine term hierarchy. This approach also satisfies the third condition, as the hierarchy can be recalculated for each new dataset. However, the outcome may not always be according to user preference (e.g. when there is a particular term of interest), so it may be necessary to review the hierarchy before implementing it.

The TF-IDF term hierarchy was applied on the classified overlay of the TDM (Appendix 2, Figure 2). Where more than one term of any one class was present in a document, only the highest ranking one was maintained. Each class of each document was treated separately. That allowed the same term hierarchy be used universally, but from it, only terms within the same class could be ranked against each other (and inferior ones removed). After settling conflicts in the classified overlay of the TDM, the basic TDM was also updated.

5.8. Output analysis

5.8.1. Data formats

Visualisation was considered the most intuitive way to present categorical data to a user when precise numerical information is not required. It can be useful when numerous classes of information need to be compared (such as to select subgroups for further study), or when simple decisions need to be made at speed (such as to check whether a newly created transformation rule has functioned).

Word clouds were used to highlight the most important terms in the vocabulary at each stage of the analysis.

The Pareto chart has been used previously to illustrate the failure rates of wind turbine components (18). A particularly informative version of the Pareto chart was used by (13), which presented two hierarchical levels simultaneously: a Pareto chart on the system level (rotor, control system, etc.), with each column in turn containing a Pareto chart of its subsystems (blades, sensors, etc.). This double-level chart was also adapted for the present study. A Matlab function was created ("ParetoPlot", Appendix 2, Figure 2) that presents an overall Pareto plot on the system level, with a subplot for each system showing its most frequent subunits. The same script also outputs the two classes of input data on a frequency-sorted basis to allow for its further numerical analysis.

5.8.1. Standardising vocabulary

Assuming that the terminology and maintenance profile will be similar for each wind farm in the future, the most common terms found in historic records should be the foundation for a standardised vocabulary in the future. Similarly to conflict resolution (section 5.7), term frequency does not directly reflect how important or discriminative a term is. Yet when constructing a standard vocabulary, the operator's and technicians' preference for specific terms should be taken into account. Replacing their accustomed vocabulary with one produced via term importance weighting such as TF-IDF may cause confusion. The TF and TDF ranking were used to compare how informative the vocabulary is provided by each.

5.9. Complete methodology

The flow chart in Figure 2 presents the sequence in which the scripts in Appendix 2 formed the final methodology. Each stage is initiated manually and it is possible to alter the sequence, provided that the workspace variables that each stage uses have been created by the previous stages. In the original sequence, the variables are created and named so that user manipulation is not necessary. However, execution stops for lexicalisation if new tokens are found (B), and pauses for model fitting (D). The outputs described in the following section were obtained by the scripts under the "OUTPUT" headings in the figure, but intermediate results are kept in workspace variables and can be accessed by other means of categorical data analysis.

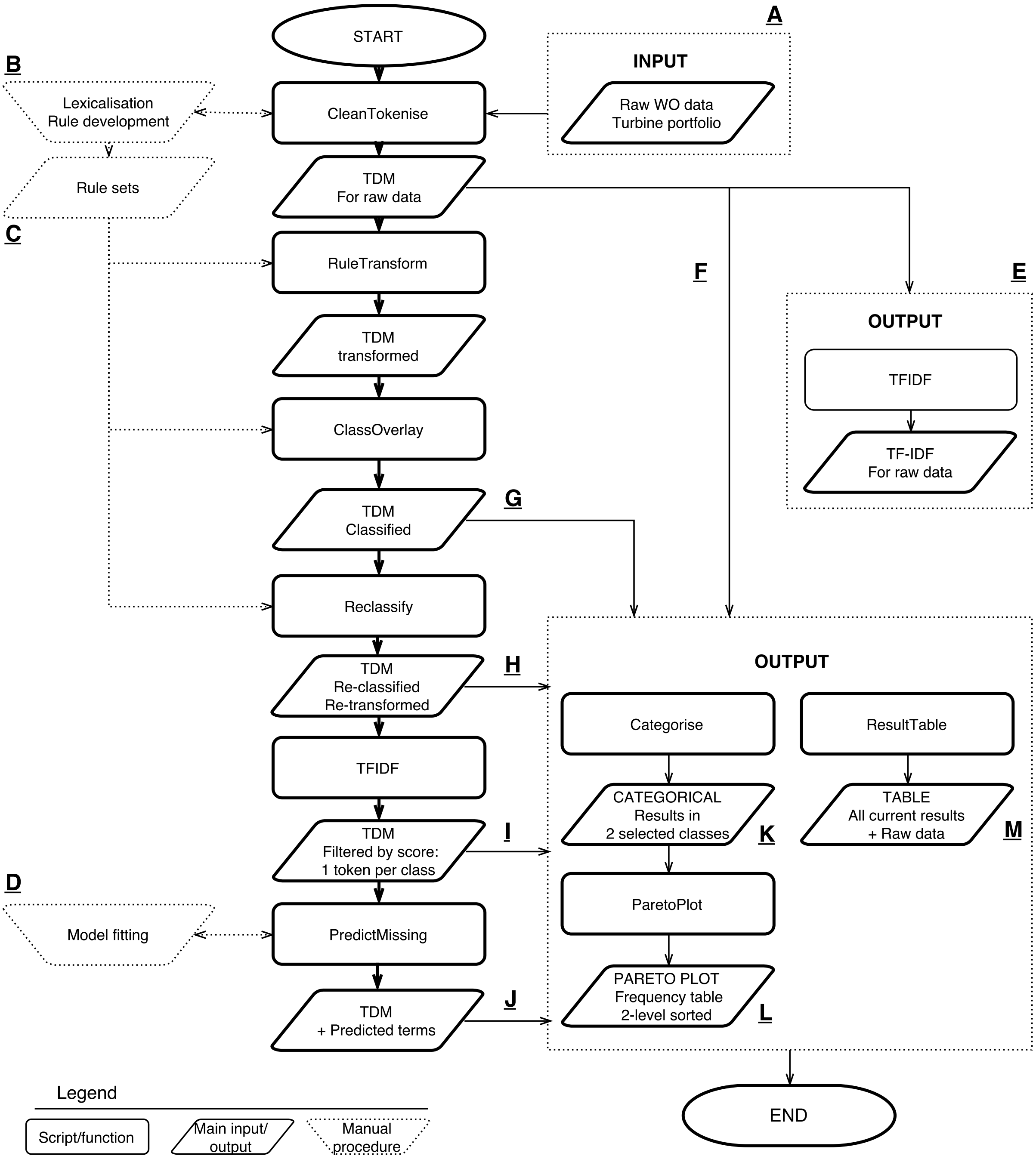
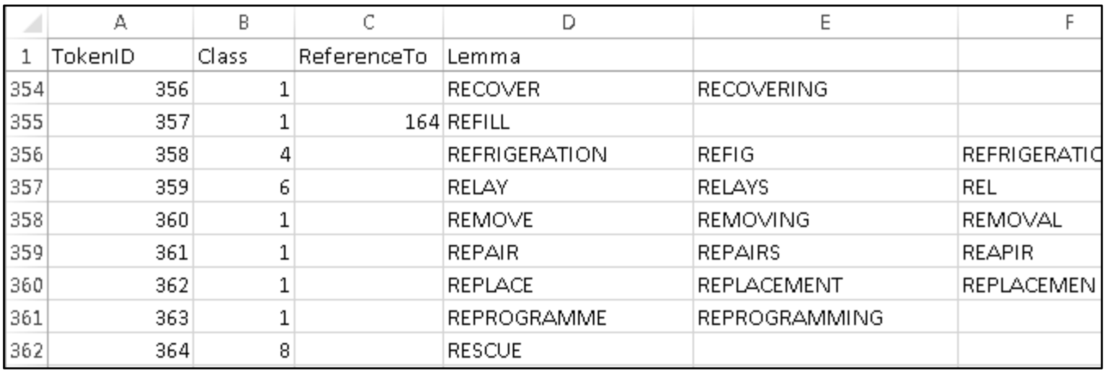


Figure 2. The text mining workflow

6. Results and improvements

6.1. Lexicalisation

Lexicalisation was one of the foundations of this study, before other operations could be carried out. An example of the resulting lexicon in MS Excel is shown in Figure 3. Different words are assigned in rows under each lemma, and assigned a numeric TokenID. Class numbers are also managed by the same table. The ReferenceTo column allows the user to specify a higher lemma. For example, by following reference TokenID 164 in the table, all cases of “Refill” will be merged under “Fill”. Note also how the TokenID does not match its row index – this is caused by rearrangement and deletion during lexicon development, and is the reason for using additional TokenID values, not simply indices.



	A	B	C	D	E	F
1	TokenID	Class	ReferenceTo	Lemma		
354	356	1		RECOVER	RECOVERING	
355	357	1	164	REFILL		
356	358	4		REFRIGERATION	REFIG	REFRIGERATIC
357	359	6		RELAY	RELAYS	REL
358	360	1		REMOVE	REMOVING	REMOVAL
359	361	1		REPAIR	REPAIRS	REAPIR
360	362	1		REPLACE	REPLACEMENT	REPLACEMEN
361	363	1		REPROGRAMME	REPROGRAMMING	
362	364	8		RESCUE		

Figure 3. Excerpt from the lexicon

Based on the training dataset, a lexicon of 2443 words and compounds was created, arranged under 592 lemmas. The test dataset introduced an additional 227 terms and 97 new lemmas, and in effect created a more comprehensive vocabulary for use in future analyses. On the other hand, the introduction of new rare terms complicated classification and grouping of the lexicon terms.

Manual lexicalisation involved approximately 15 days of labour throughout the duration of the project. This includes the identification of semantic relationships, resolving word sense conflicts, associating abbreviations and jargon with the equivalent terms, classifying terms, etc. The amount of time spent on this basic task signifies again how a standardised nomenclature could help to reduce the workload in analysing maintenance records.

The lexicon can be used as a foundation for similar projects in the future, without significant modification in its structure or the semantic associations between terms.

A major drawback of manual lexicalisation is the need to recognize and capture all new terms, and compounds in particular. The present method also depends heavily on the operator's judgement on grouping and classifying terms. Justeson et al. (27) give a warning that lexical methods of term recognition do not guarantee a match, because terms in a given context might be too specific, and not included in the dictionary.

The inclusion of a term extraction algorithm could increase the effectiveness, and reduce time consumption. Yet as discussed in Chapters 3 and 5, the dataset is noisy and inconsistent for reliable automatic term extraction.

Miner (9) describes the use of singular value decomposition (SVD) as a way to reduce the representation of each document in a TDM to its most meaningful, descriptive values. This is achieved by determining which combinations of terms, when other terms are removed and documents much shortened, result in the highest remaining distinction between documents. Miner describes SVD in a context of larger documents and with the purpose of dimensionality reduction. However, in the present work, an improved term ranking mechanism would be highly useful in two applications. Firstly, as part of a supervised lexicalisation process, it could recognise candidate terms for lexicalisation by the user. Secondly, the term importance values calculated initially could be used later to replace the TF-IDF measure for conflict resolution.

6.2. Data cleaning

A summary of the effects of each stage in the text mining process is given in Table 8 and Table 9 for the training and test datasets, respectively.

Data cleaning was the first stage of the process, and chiefly intended to standardise the document collection for further methods. Yet the removal of noise and stop words already had its effect in increasing the similarity between documents, or rather, making semantically similar documents appear similar by syntactic comparison. The number of different WOs was thereby reduced by 38% and 29% for the training and test datasets, respectively (Table 8, Table 9).

Table 8. Main parameters during the text mining process - TRAINING dataset

Stage	Number of documents	Number of different WOs	Number of words	Size of vocabulary (including compounds)	Percentage of unknown systems/ subsystems
Raw	12425	4657	63032	2443	N/A
Filtered	6372	3473	33904	1464	N/A
Cleaned (F)	6371	2150	25674	494	N/A
Transformed	6371	2022	22764	486	N/A
Classified (G)	6371	2022	22764	486	4357/1215
Reclassified (H)	6371	1912	25778	488	1843/1215
TF-IDF filtered (I)	6371	1756	23763	438	1843/1215
KNN prediction (J)	6371	1697	25504	438	102/1215

Table 9. Main parameters during the text mining process - TEST dataset

Stage	Number of documents	Number of different WOs	Number of words	Size of vocabulary (including compounds)	Percentage of unknown systems/ subsystems
Raw	3431	1530	17873	1047	N/A
Filtered	1431	926	8358	684	N/A
Cleaned (F)	1431	653	5461	317	N/A
Transformed	1431	622	4695	313	N/A
Classified (G)	1431	622	4695	313	919/386
Reclassified (H)	1431	608	5121	315	624/386
TF-IDF filtered (I)	1431	588	4842	276	624/386
KNN prediction (J)	1431	568	5347	276	119/386

6.3. Similarity and grouping

Table 10 presents the most frequent free text entries after the first filtering, and after all stages of transformations had been completed. The grouping and counting is based on direct similarity on a character level. The most frequent WOs after filtering maintained the same, or very similar, frequency after the transformations. Among the slightly less frequent WOs that followed (data not shown), many were related to inspection and servicing. MacLean (22) attributes the latter to a stricter supervision and regulation of safety-related tasks. However, it is clear that the more frequent and standardised a procedure, the more uniform and standardised is its description.

On the other hand, certain tasks that were most frequent in the final results, were not initially in the top ten. These were therefore expressed in several different ways in the original data, and the information extraction process interpreted them to a similar form. One of the major factors in homogenising equivalent expressions was the KNN classifier's successful prediction of "control" and "lifting.system": if some of the entries were previously described without the system, and some included it, then after processing they were all expressed in the same way. Cleaning and removing excess words such as "AT" and "AAT" also had a major role in homogenising expressions. Data cleaning effects can be seen in more detail in Table 13 and Table 14 (Appendix 1).

Table 10 also highlights some of the negative effects that the transformations had. The most significant one was misinterpretation of the initially most frequent WO (ranked 7th in the end). The classifier predicted "gear" to mean "yaw gear", although most likely it was an abbreviation for the gearbox, and the system should be "drive train". What is more, the "oil filter" was removed by the TF-IDF filtering, although it was the more discriminative term in this case. "G/box" was correctly identified as gearbox, and recognised as belonging to the drive train, but the term "gearbox" itself was missing from the final text. Also some excess words were not removed (e.g. "To" in this example).

Table 10. Most frequent WOs in the TRAINING dataset after first filtering and in the final results.

Rank	First filtering		Final (J)	
1	'INSPECT GEAR OIL FILTER AND EQUIPMENT'	62	'INVESTIGATE **CONTROL** INTERBUS ERROR'	113
2	'***AT***- REPLACE ***G/BOX*** STAT & HOSES'	61	'MODIFY ***LIFTING.SYSTEM*** HOIST SYSTEM'	82
3	'WTG ENTRY FOR HV TRANSFORMER WORKS'	61	'REPROGRAMME TURBINE SMP'	75
4	'CARRY OUT PERSONNEL LIFT WORKS'	61	'REPLACE GENERATOR SLIP.RING FAN'	68
5	***'AAT*** TO RE-INSTALL ELEVATOR COMPONENTS'	61	'REPAIR CONTROL INTERBUS ERROR'	64
6	'AT-CHANGE POWER FACTOR / GROUND CONTROL'	60	'INSTALL LIFTING.SYSTEM ELEVATOR COMPONENTS ***TO***'	63
7	'AT - INVESTIGATE COMMS ISSUE AFTER TRIP'	60	'INSPECT ***YAW GEAR EQUIPMENT'***	62
8	'AT-SNAGGING AND PAINTING WORKS'	60	'WORK LIFTING.SYSTEM ELEVATOR'	62
9	'AT- INST. GENERATOR TO SUPPLY POWER WTG'	60	'PAINTING'	61
10	'INSTALL NEW BREAKER PLUG SOCKET'	60	'REPLACE ***DRIVE.TRAIN*** PRESSURE.STAT HOSE'	61

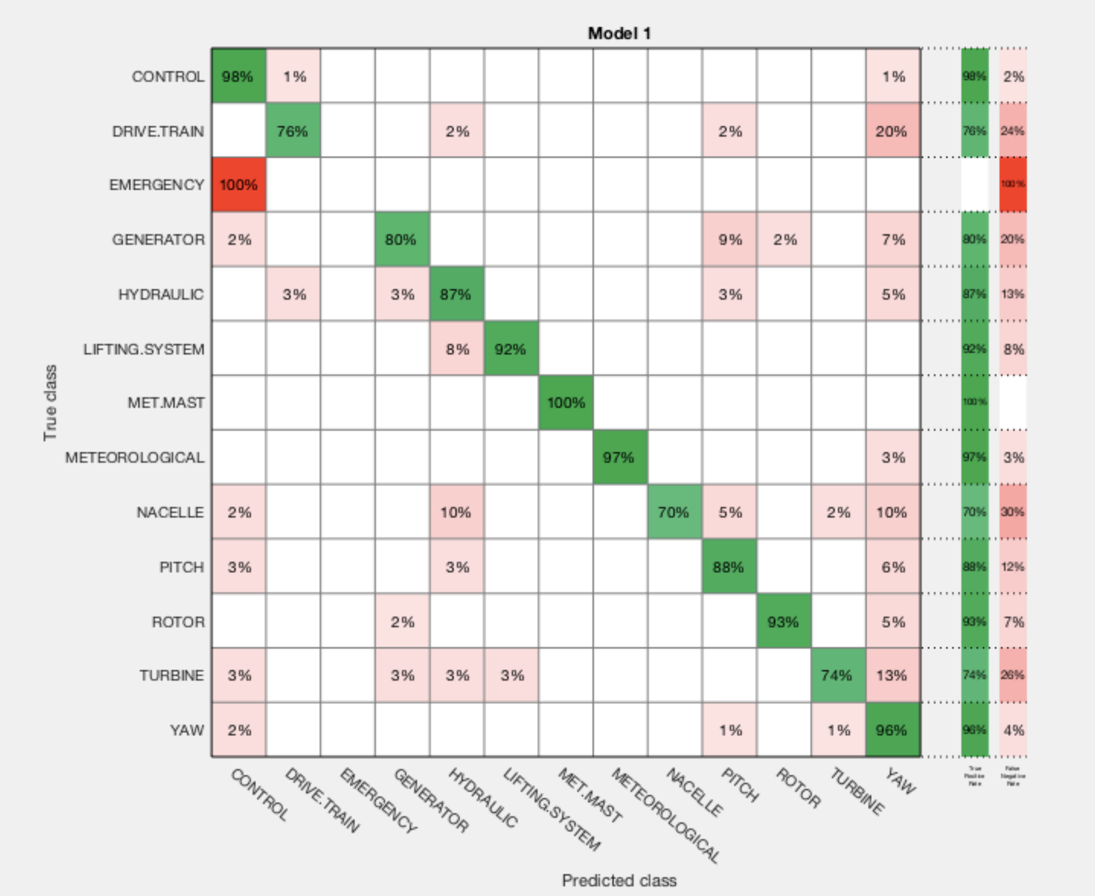
6.4. Class conflicts

Some conflicts were detected in component taxonomy, mostly on lower levels of component taxonomy (data not shown). A common example was that "fan" and "motor" are both classed as products, but motors are also part of cooling fans. This indicates that the hierarchical classification may need further tuning. Alternatively, if it were known that all turbines in the analysis follow a particular taxonomy, then that predefined taxonomy could be used within this algorithm. Using a more standardised taxonomy such as the one developed by the Reliawind project (13) would also offer better comparison between different studies. However, as the authors point out, the industry still lacks a universal taxonomy.

Some of the conflicts where two terms of the same class were present were caused by a single WO actually describing two separate replacements. Such cases were outwith the assumptions of this study. However, it is in principle possible to extract the two component names separately. Sometimes stop words could be used for this. Words like "and", "in", "of", "between" often identify which part names are used as location identifiers, and which represent the subjects of a maintenance task. These distinctions could be made with by straightforward syntactic approach, but stop word removal would need to be carefully reconsidered.

6.5. Classification and clustering

Above, the KNN classifier was found to misinterpret the system as “Yaw” instead of the expected “Drive train” (Table 10). The confusion matrix for that classifier is shown in Figure 4.



	CONTROL	DRIVE.TRAIN	EMERGENCY	GENERATOR	HYDRAULIC	LIFTING.SYSTEM	MET.MAST	METEOROLOGICAL	NACELLE	PITCH	ROTOR	TURBINE	YAW	True Positive Rate	False Negative Rate
CONTROL	98%	1%											1%	98%	2%
DRIVE.TRAIN		76%			2%					2%			20%	76%	24%
EMERGENCY	100%														100%
GENERATOR	2%			80%						9%	2%		7%	80%	20%
HYDRAULIC		3%		3%	87%					3%			5%	87%	13%
LIFTING.SYSTEM					8%	92%								92%	8%
MET.MAST							100%							100%	
METEOROLOGICAL								97%					3%	97%	3%
NACELLE	2%				10%				70%	5%		2%	10%	70%	30%
PITCH	3%				3%					88%			6%	88%	12%
ROTOR				2%							93%		5%	93%	7%
TURBINE	3%			3%	3%	3%						74%	13%	74%	26%
YAW	2%									1%		1%	96%	96%	4%

Figure 4. Confusion matrix created when learning KNN classifier for test dataset

It shows that the same happened for in 20% (which corresponds to 9 times) of the cases during cross-validation. It can be concluded that the learning and prediction datasets were different in terms of the prediction variables present – insead of 20%, all of the gearbox entities were falsely identified. In addition, the confusion matrix shows that 100% of “Emergency” values were misinterpreted. In absolute count, this was 1 occasion. This demonstrates that the KNN classifier is a method that needs to be verified not just by cross-validation during training, but via the predicted results.

Despite effective term classification based on the lexicon, there remained an impractically large number of different WOs (Table 8 and Table 9), and different levels in each component class (Table 15 and Table 16). Many semantically equivalent WOs differed by a few tokens so that they could not be matched directly on a character level. Regardless, the KNN classification algorithm was trained with a high accuracy rating (89 to 96% depending on predictor classes). This indicates that the outputs may be suitable for training a similar classifier to match equivalent WOs based on a distance measure, rather than direct comparison. However, for a more accurate analysis in the future, the 40 character limit should be used to the full extent by the person assigning the work orders.

An even more practical approach would be to match all work orders to a predefined taxonomy based on, for instance, the Jaccard distance (prior confirmation would be needed that the turbine model under consideration follows that particular taxonomy). This was presently not attempted as one of the underlying aims of this method was to develop an algorithm independent of a predefined taxonomy. The method was initially applied without any defined component relationships, however that caused considerably more coarse results (a more spread-out taxonomy). For instance, instead of classifying all sensors as part of the control system, that approach assumed that every system had its own sensors as a subsystem. This was not considered informative, nor correct in practice. Therefore some of the most significant cases such as the above example were referred to their correct locations by specially defined rules.

6.6. Distribution of maintenance efforts

The first results using the ParetoPlot script are shown in Figure 6 (Appendix 1). It can be seen that around 68% of the system level taxonomical units are unknown, which means the results are inaccurate for practical use. Additionally, multiple tokens per class were interpreted as separate categories, which appeared as the dense text labels. This clearly identifies that firstly, more narrow categories have to be selected, and secondly, that further specification of categories was necessary. Both aspects were improved, as shown for example in Figure 5.

The present methodology requires careful selection of grouping variables. Some variable combinations for plotting/analysis are more meaningful than others. For example, if the intention was to find out how many generators were replaced, it is not sufficient to choose “generator” as the system and “replace” as the activity. Doing so would ignore any additional component descriptions that followed “generator”, including a large number of cooling fans etc. A false impression of very frequent generator replacements would be created. Instead, the entire component hierarchy should be concatenated and used as subcategories of the activity “replace”. In that case, subcomponents like “generator cooling fan” and “generator” itself could be clearly distinguished.

Unknown values formed a large proportion of most classes. Unknown values in system and subsystem names were attributed to the fact that certain components could be unambiguously identified without specifying a system. The operator may not consider a full description necessary, assuming that the text would be read by another human. Similarly, some components can be identified by failure mode. There are also cases where failure mode or maintenance activity cannot be estimated at the time of writing the work order – this often takes place before a technician is able to investigate the fault in person (22). Regardless of the cause, unknown values posed difficulties in systematic classification of each WO.

6.7. Data fusion

WO free texts contain valuable information, but not all consequences of a fault or remedial action can be found in this one source. For maximal value in maintenance planning, each task should be associated with the number of technicians and number of working hours, as well as the cost and lead time of any components, materials or externally provided services. This can be achieved by using data mining methods on alternative source fields (6), incorporating present results with technicians’ labour time tables, SCADA fault code databases. WO tables also contain other fields apart from free text, such as functional location, maintenance type, and dates. Arif-Uz-

Zaman et al. (21) used a data fusion approach where work orders provided details of component identification, failure mode and remedial activities undertaken, while consequences of each event were obtained from downtime data. Thereby full ABC analysis and failure modelling would be possible, which offer significantly more grounds for maintenance planning than failure rates alone, as described in Chapter 2.

6.8. Verification and confidence

Due to the high proportion of unknown values in most classes, the actual frequency of terms in each class cannot be accurately estimated. To make any practical conclusions from data with such high uncertainty, a reliability or confidence interval is essential. Most work orders or components, when grouped, follow a certain frequency distribution. The same can be expected of the unknowns. However there is no basis to assume that the two distributions are similar: the unknowns may be caused by a bias such as operator decision; it may also include values that are not found in the known groups. Nevertheless, it is possible to estimate a distribution for the unknowns, either based on expert opinion or some assumptions. The distribution could then be used to estimate the actual frequency in each group. Confidence intervals could be calculated and used together with the frequency data in the analysis.

Nevertheless, there is useful information in a dataset regardless of the unknowns. Successfully identified values give a minimal frequency for each group, which provides a best-case estimate of how many corresponding spare parts may be necessary over time. The worst case, or maximum possible failure frequency in each group, would be the sum of itself and all the unknown values.

Another factor that can influence the results is that the rule-based algorithms here depend heavily on operator’s judgement. Any new dataset or discovered rule exception required significant user input to update the lexicon and term associations, as well as other rules. It would be desirable to reduce the operator to the role of a supervisor, rather than the principal decision-maker. Some methods such as additional machine learning steps and statistical term recognition have been proposed above. However, also shown, the data conditions reduce the accuracy and reliability of the suggested improvements (8,25). Different authors agree (6,13,21) that semantic ambiguity and poor statistical representation (e.g. unique terms/spellings) are inevitable in this type of data. Any improvement methods should therefore be designed with this consideration. A confidence metric should be applied to highlight any instances of low accuracy, which could then be forwarded to the user for reviewing.

Alternatively, verification could be done by comparing a sample set of results to a previously approved manually annotated set. A benchmark set would be useful for further testing and development of wind turbine specific text mining tools. This should be a fully tagged dataset of work orders, including component classifications, failure modes and maintenance activities. Several turbine models and technologies should be included. The size of the dataset should be sufficient to provide a representative learning set for machine learning algorithms. Such a dataset would allow training and testing sets to be taken from a uniformly reliable source, so that methods could be validated to a higher standard and compared to one another.

6.1. Computational performance

During algorithm development, several decisions were made taking computing time requirement into account. Testing the algorithm revealed that this is not a major concern with the current dataset and lexicon size: running time was under three minutes in all cases, including classifier training, but excluding rule creation and other manual inputs. Computing time may become an issue when repeated runs of the algorithm are required, for example when testing new rule sets. To save time, it is advisable to test and adjust one section of code at a time, or run the full algorithm with a reduced sample size.

During development, the modular layout and saving of intermediate outputs increased the transparency of the process. This enabled the user to take outputs for testing after each stage, and simplified fault-finding where that was necessary. After the recommendations in this chapter have been applied to improve the text mining methodology, it is advisable to make certain adjustments to its layout. Scripts should be arranged in an automated continuous workflow as far as possible. Variables should be cleared from memory when not needed. Related variables, e.g. TDM and its classified overlay, should be arranged in a structure, which allows more effective referencing between functions.

6.2. Adaptation by the industry

6.2.1. Advantages and economic benefit

The method development process above, and in particular the distribution of labour intensity, highlighted several advantages that an automated WO text mining process has over the manual alternative.

MacLean (12) had previously performed the classification process manually on the test dataset in this work, and estimated the labour time at around two weeks (10 working days). The economic benefit of automating the WO free text analysis was considered directly equivalent to the expenses of carrying out equivalent analysis manually. Since manual handling of domain-specific data requires background knowledge, it was assumed that the minimal qualification level for this work would be associate professional, but a professional might be more effective. The two pay scales were obtained from Labour Force Survey data (32), and were £798 per week for professional, and £671 per week for associate professional levels. The average value of £734.5 per week was used. Assuming a constant earning rate for 52 weeks per year, the annual salary accounts to £38194. This was used as input to The True Cost of an Employee calculator (33), which was the most comprehensive tool found for estimating an employee's cost to an employer. National insurance rate and threshold were updated, but otherwise default values were used in the calculator. The cost of the employee was found to be £34.09 per hour. Assuming that a working day lasts 8 hours, this accounts for £2727 pounds for the 10 days that it takes to manually transform 3431 WOs in the test set. Assuming a similar rate of productivity for larger datasets, the training set would cost the company 3.62 times more to analyse, which is £9876. For comparison, the algorithm runs for under 3 minutes without lexicalisation. With lexicalisation of the additional 227 tokens of the test dataset, the total time was under two hours, which accounts for less than £70.

For a company that operates several wind farms and has an internal terminology or reporting structure, it might be necessary to customise the rules and lexicon, but they are complemented with each dataset, and the effort is thereby reduced every time. Whereas when done manually, the workload is related to the number of records in a dataset, regardless of previous experience. The work is also tedious and repetitive, which may reduce employee satisfaction and decrease their productivity.

Manual analysis may be more transparent and trusted than an algorithm. The operator's expert judgement should be the benchmark to determine how each text entry should be treated and classified to best inform a particular analysis. However, a person's judgement is inevitably subjective, and may vary over time. How a person classifies a particular term or work order is not necessarily consistent. This is especially true for large, diverse, and granular (detailed) datasets. Two main types of bias may affect the manual WO analysis procedure: anchoring bias (failing to adjust the approach when new information is found that actually demands it), and inconsistency bias (forgetting or changing assumptions or approaches during process, here due to dataset size) (34). Noisy data affects a person's judgement during manual classification in the same way as when writing rules for an algorithm. A well-defined set of rules may therefore offer consistency equal to or better than a human, but will perform the classification task considerably faster.

Automation is advantageous if the analysis is done on a larger scale. If an operator of several wind farms needs to perform a comparative analysis, then the records from each site first need to be analysed individually. Assuming dataset size similar to the one in this study, the manual workload would be two weeks per site. The work could be done by the same person over a longer period, or by different persons simultaneously. In the former case, the drift in judgement over time may affect the results; in the latter case, it would be difficult to achieve consistency between the people (34).

As outlined in Chapter **Error! Reference source not found.**, the ultimate purpose of WO free text analysis is to obtain reliability data to inform the wind farm operator during a decision-making process. Those benefits were considered equal regardless of the methods used for text analysis, and were not separately quantified.

6.2.2. Data collection

Despite the requirement to fine-tune the algorithms and methods, the fundamental classification approach in this work was found effective. As existing terms could be accommodated under this system, it is also possible to adapt a similar classification for data collection in the future. Classifying parts of each work order at its creation would save the effort - and uncertainty - of text mining, if the intention is to use such data in a reliability analysis.

6.2.3. Standard terminology

Standard technical terms are coined to facilitate effective communication between experts in a field (27). The term list of 1330 terms in the present results was inconveniently long to be used by field technicians. A choice needs to be made which historical terms to standardise.

To illustrate how the choice of vocabulary would depend on the choice of ranking scale, the top 10 words of the maintenance activity, system, and failure mode classes were ranked using both TF and TF-IDF weighting Table 11. In this case, the TF-IDF ranking in did not provide suitable grounds for an overall importance ranking. Term frequency ranking gave more informative and relevant terms that could all be expected from a standard list. Yet the TF-IDF ranking brought up terms such as “Contactor” or “Battery” that are not frequently used, but are still uniquely descriptive in wind turbine maintenance. These also need to be included in the database to maintain the level of detail. Regardless of how terms are ranked, the choice of should be led by the natural development of technical language: frequently used terms should be short and concise (27).

Table 11. TF and TF-IDF ranking of terms before and after transformations in TRAINING dataset

	After cleaning (E)				After TF-IDF-based sorting (I)			
Rank	Word	TF	Word	TF-IDF	Word	TF	Word	TF-IDF
1	'REPLACE'	468	'FAULT'	16.91	'REPLACE'	425	'HIGH'	16.91
2	'RESET'	441	'CABINET'	16.91	'BASE.RESET'	322	'TOP'	16.91
3	'BASE'	314	'CB'	16.91	'INVESTIGATE'	215	'FAILURE'	16.90
4	'REPAIR'	201	'MODULE'	16.91	'REPAIR'	201	'BATTERY'	16.90
5	'ERROR'	170	'OK'	16.91	'CONTROL'	189	'CONTACTOR'	16.90
6	'YAW'	158	'VALVE'	16.91	'ERROR'	170	'FRECUENCY'	16.90
7	'TOP'	153	'HIGH'	16.90	'YAW'	133	'TOG'	16.89
8	'ACCUMULATOR'	130	'BATTERY'	16.90	'ACCUMULATOR'	110	'NO'	16.89
9	'INVESTIGATE'	100	'CONTACTOR'	16.90	'HYDRAULIC'	84	'PROPORTIONAL.VALVE'	16.89
10	'HYDRAULIC'	86	'NITROGEN'	16.90	'PITCH'	83	'ANEMOMETER'	16.88

6.2.4. User interface

Firstly, for work order creation, instead of a plain list of standard terms it is possible to use a tree-style interface where terms are classified similarly to the results here. When specifying a work order, the user would choose terms from one class at a time, starting from more general, e.g. on the system level with only 13 separate choices. With all historical combinations of tokens known to the computer, the database would perform as a truth model (26). Each choice the user makes would narrow down the remaining combinations, and with it reduce the number of terms they are allowed to choose in each class. The full lexicon could be made available in this way, but if the amount of information forwarded at any one time is small, it is not expected to overwhelm the user (34).

Secondly, to be able to extract most value from historical WO records, the industry should be provided with an automated tool for WO text mining. The text mining process developed in this project is not capable of providing results at the required reliability level. However, the TDM model, lexical approach, term classes and in some cases KNN-based predictive capabilities could be effective as part of such a toolbox.

7. Conclusions

The present thesis is an investigation into the opportunities and challenges in mining work order free text data. Literature review was carried out in two chapters. The first chapter established that failure rates, downtimes, repair times, lead times, and cost can be used for evidence-based decision making in wind turbine maintenance. The Pareto and ABC analysis were identified as the principal methods in which these data can be exploited. It was also concluded that failure frequency information for this purpose can be extracted from work order free text data. The second review chapter provided an overview of text mining methods that could be used to extract the failure information from work order free text in an actionable format.

Following the results of the two review chapters, the development of a robust text mining methodology for the specific domain of wind turbine SAP work orders was undertaken. During preparation of the algorithms, a list of main issues detected in this data type was constructed and supplied with examples, which can be used to inform future studies in this field. In addition, a lexicon of 1330 tokens under 597 lemmas was developed to support the rule-based algorithm. The algorithm itself was constructed of 9 separate functional modules, each based on a custom Matlab script.

The method was used on two separate sets of SAP work orders from large-scale wind farms in Scotland. In both cases, the number of different work orders was significantly reduced by cleaning and transformations. Furthermore, missing values were predicted by a KNN classifier, which reduced the extent of uncertainty in the final results.

Several advantages of using the algorithm in an industrial context were identified, including consistency and cost savings compared to performing the same analysis manually. To help advance the use of work order data mining methods in the industry, two possible user interfaces and a means to standardise maintenance terminology were also proposed.

The results of this work conform with previous studies in the field, showing work order free texts to be a difficult data type to mine. Missing and noisy data, ambiguous expression, and spelling errors were found to degrade the quality of the results most significantly; other more specific issues were also described.

The effect of noisy data was more severe on rare terms, where each instance contributes more to its final interpretation, but statistical basis to determine the correct one is limited. This was noticed to influence both the rule-based approach, and especially the machine learning classifier. However, a well-defined confidence metric was not found that would allow to quantify the negative effects. This was recognised as one of the major shortcomings of the present analysis. It is strongly recommended that for further analysis using the methods in this study, or the suggested improvements, a framework should be developed where each stage of the analysis could be associated with a suitable accuracy or error metric, so that final results could be presented with a confidence interval. This would both provide a basis for fine-tuning the methodology, as well as increase the appeal of the analysis as a decision-making tool in the industry.

8. References

1. Letcher TM. Wind energy engineering. A handbook for onshore and offshore wind turbines. Academic Press; 2017.

2. Tavner P. Offshore Wind Turbines: Reliability, availability and maintenance. 2012. 293 p.

3. McMillan D, Dinwoodie IA, Wilson G, May A, Hawker G. Asset modelling challenges in the wind energy sector [Internet]. 2014 [cited 2017 May 9]. Available from: https://pure.strath.ac.uk/portal/en/publications/asset-modelling-challenges-in-the-wind-energy-sector(ec0f0152-893b-4399-8a6c-297ee9bdd60d)/export.html

4. Carroll J, McDonald A, McMillan D. Failure rate, repair time and unscheduled O&amp;M cost analysis of offshore wind turbines. Wind Energy [Internet]. 2016 Jun [cited 2017 May 30];19(6):1107–19. Available from: http://doi.wiley.com/10.1002/we.1887

5. Parle J, Gibson J, Reese C. Wind Industry Work Order Information Flow Survey. 2013.

6. Hodkiewicz M, Ho MT-W. Cleaning historical maintenance work order data for reliability analysis. J Qual Maint Eng [Internet]. 2016 May 9;22(2):146–63. Available from: http://www.emeraldinsight.com/doi/10.1108/JQME-04-2015-0013

7. SAP enhancements improve coal plant maintenance practices. Power Eng (Barrington, Illinois) [Internet]. 2008;112(2):72–4. Available from: http://www.scopus.com/inward/record.url?eid=2-s2.0-41149170681&partnerID=40&md5=233a7c491ef0c527aa000c15b9018792

8. Banchs RE. Text Mining with MATLAB. New York: Springer; 2013.

9. Miner G. Practical text mining and statistical analysis for non-structured text data applications. Academic Press; 2012. 1053 p.

10. Froelich J, Ananyan S. Decision Support via Text Mining. In: Handbook on Decision Support Systems. 2008.

11. Richnow J, Rossi C, Wank H. Designation of wind power plants with the Reference Designation System for Power Plants RDS-PP. VGB Powertech. 2014;38–44.

12. MacLean G. Private conversation. Glasgow; 2017 Mar 22;

13. Wilkinson M, Hendriks B, Spinato F, Delft T Van. Measuring wind turbine

reliability, results of the reliawind project. Eur Wind Energy Assoc Conf [Internet]. 2011;1–8. Available from: http://www.gl-garradhassan.com/assets/downloads/Measuring_Wind_Turbine_Reliability_-_Results_of_the_Reliawind_Project.pdf

14. Wilson G, McMillan D. Assessing Wind Farm Reliability Using Weather Dependent Failure Rates. J Phys Conf Ser [Internet]. 2014;524(Torque):12181. Available from: http://stacks.iop.org/1742-6596/524/i=1/a=012181?key=crossref.b8dc65677e053e3646aa421e267e73ed

15. Wilson G. Quantifying the Relationship Between Wind Turbine Component Failure Rates and Wind Speed. University of Strathclyde; 2015.

16. Mckenney BL, Ogilvie AB, Peters VA. Using Wind Plant Data to Increase Reliability. Sandia Report. 2011.

17. Rademakers H. Braam, M. B.Zaaijer, G. J. W. Van Bussel LWMM. Assessment and optimisation of operation and maintenance of offshore wind turbines. Proc EWEC. 2003;8–12.

18. Hill RR, Stinebaugh J a, Briand D, Benjamin AS, Linsday J. Wind Turbine Reliability : A Database and Analysis Approach. Sand2008-0983. 2008;(February):60.

19. Spinato F, Tavner PJ, van Bussel GJW, Koutoulakos E. Reliability of wind turbine subassemblies. IET Renew Power Gener [Internet]. 2009;3(4):387. Available from: http://digital-library.theiet.org/content/journals/10.1049/iet-rpg.2008.0060

20. Königstein H, Müller H, Kaiser J. RDS-PP – Transition from the KKS to an international standard. VGB Powertech. 2007;87(8):64–72.

21. Arif-uz-zaman K, Cholette ME, Li F, Ma L, Karim A. A Data Fusion Approach of Multiple Maintenance Data Sources for Real-World Reliability Modelling. In: Proceedings of the 10th World Congress on Engineering Asset Management (WCEAM 2015) [Internet]. 2016. p. 69–77. Available from: http://link.springer.com/10.1007/978-3-319-27064-7

22. MacLean G. Private conversation. Glasgow; 2017 Jun 7;

23. Galar D, Kans M, Schmidt B. Big Data in Asset Management: Knowledge Discovery in Asset Data by the Means of Data Mining. In: Koskinen KT, Kortelainen H, Aaltonen J, Uusitalo T, Komonen K, Mathew J, et al., editors. Proceedings of the 10th World Congress on Engineering Asset Management (WCEAM 2015) [Internet]. Cham: Springer International Publishing; 2016. p. 161–71. Available from: http://dx.doi.org/10.1007/978-3-319-27064-7_16

24. Witten IH, Frank E, Hall M a. Data Mining: Practical Machine Learning Tools and Techniques [Internet]. Complementary literature None. 2011. 664 p.

Available from: http://books.google.com/books?id=bDtLM8CODsQC&pgis=1

25. Feldman R, Sanger J. The Text Mining Handbook [Internet]. Cambridge: Cambridge University Press; 2006 [cited 2017 May 7]. Available from: https://www.cambridge.org/core/product/identifier/9780511546914/type/book

26. Han J, Kamber M. Data Mining: Concepts and Techniques. Elsevier. 2006.

27. Justeson JS, Katz SM. Technical terminology: some linguistic properties and an algorithm for identification in text. Nat Lang Eng. 1995;1(1):9–27.

28. Evert S. The statistics of word cooccurrences : word pairs and collocations [Internet]. 2005 [cited 2017 Jul 7]. Available from: https://elib.uni-stuttgart.de/handle/11682/2573

29. Croft WB, Metzler D, Strohman T. Information retrieval in practice. 2009;

30. Cambria E, White B. Jumping NLP curves: A review of natural language processing research. IEEE Comput Intell Mag. 2014;9(2):48–57.

31. Rose S, Engel D, Cramer N, Cowley W. Automatic Keyword Extraction from Individual Documents. In: Text Mining: Applications and Theory. 2010. p. 1–20.

32. Office for National Statistics. EARN06: Gross weekly earnings by occupation - Office for National Statistics [Internet]. [cited 2017 Aug 25]. Available from: https://www.ons.gov.uk/employmentandlabourmarket/peopleinwork/earningsandworkinghours/datasets/grossweeklyearningsbyoccupationearn06

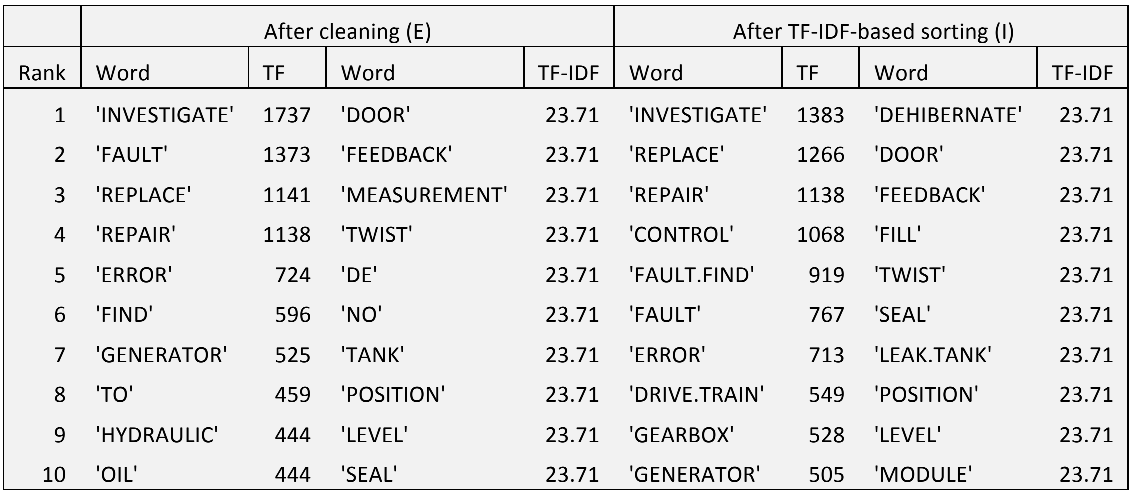
33. True Cost of an Employee Calculator [Internet]. [cited 2017 Aug 25]. Available from: http://www.accountingservicesforbusiness.co.uk/calculators1/true-cost-of-an-employee/

34. Meyer MA, Booker JM. Eliciting and analyzing expert judgment : a practical guide. Society for Industrial and Applied Mathematics and American Statistical Association; 2001.

9. Appendices

9.1. Appendix 1. Detailed results

Table 12. Comparison of term ranking scales - TEST dataset



	After cleaning (E)				After TF-IDF-based sorting (I)			
Rank	Word	TF	Word	TF-IDF	Word	TF	Word	TF-IDF
1	'INVESTIGATE'	1737	'DOOR'	23.71	'INVESTIGATE'	1383	'DEHIBERNATE'	23.71
2	'FAULT'	1373	'FEEDBACK'	23.71	'REPLACE'	1266	'DOOR'	23.71
3	'REPLACE'	1141	'MEASUREMENT'	23.71	'REPAIR'	1138	'FEEDBACK'	23.71
4	'REPAIR'	1138	'TWIST'	23.71	'CONTROL'	1068	'FILL'	23.71
5	'ERROR'	724	'DE'	23.71	'FAULT.FIND'	919	'TWIST'	23.71
6	'FIND'	596	'NO'	23.71	'FAULT'	767	'SEAL'	23.71
7	'GENERATOR'	525	'TANK'	23.71	'ERROR'	713	'LEAK.TANK'	23.71
8	'TO'	459	'POSITION'	23.71	'DRIVE.TRAIN'	549	'POSITION'	23.71
9	'HYDRAULIC'	444	'LEVEL'	23.71	'GEARBOX'	528	'LEVEL'	23.71
10	'OIL'	444	'SEAL'	23.71	'GENERATOR'	505	'MODULE'	23.71