Power Plants Failure Reports Analysis for Predictive Maintenance

Vincenza Carchiolo[1], Alessandro Longheu[2], Vincenzo di Martino[3] and Niccolo Consoli[2]
[1]*Dip. di Matematica e Informatica, Universita' di Catania, Viale Andrea Doria 6, Catania, Italy*
[2]*Dip. di Ingegneria Elettrica, Elettronica e Informatica, Universita' di Catania, Viale Andrea Doria 6, Catania, Italy*
[3]*BaxEnergy, Catania, Italy*

Keywords: Predictive Maintenance, Natural Language Processing, Ontologies, Wind Turbines, Renewable Energy.

Abstract: The shifting from reactive to predictive maintenance heavily improves the assets management, especially for complex systems with high business value. This occurs in particular in power plants, whose functioning is a mission-critical task. In this work, an NLP-based analysis of failure reports in power plants is presented, showing how they can be effectively used to implement a predictive maintenance aiming to reduce unplanned downtime and repair time, thus increasing operational efficiency while reducing costs.

1 INTRODUCTION

It can be nowadays considered as a matter of fact that climate changes and the increasing energy demand are competing factors that push for deep exploitation of renewable energies (IEA, b); in particular their market share is expected to grow up to the 40% the next five years (IEA, a), overcoming the contribution of coal and gas.

Among all sources as hydropower, geothermal, solar and others, the forecasts for wind as renewable energy endorse its position as one of the most relevant (estimated in 2018 as 43% higher with respect to 2015 (Council, 2019)). Wind power comes mainly from wind turbines, that are mechatronic devices using blades and rotor to convert wind into mechanical energy and shafts and generator to transform motion into electrical energy.

Wind turbines (WT) industry is growing faster and faster and larger devices are being developed, though WTs are continuously exposed to (possibly) extreme weather conditions, resulting in significant mechanical stress and moreover both on-shore, as well as offshore placement, is often characterized by restricted accessibility. This scenario can heavily impact on WTs reliability, often leading its components to fail during WTs' lifetime (Guolin et al., 2016), therefore making the Operation and Maintenance (O&M) a critical activity that actually impacts for up to 30% of the WT life cycle (Fischer et al., 2012).

In particular, since any technical intervention (e.g. replacing a WT damaged part) can be very expensive and, moreover, the lack of power generation during downtimes also impacts on revenues (Herbert et al., 2010), several strategies can be adopted to limit these expenses (Abichou et al., 2014), from condition monitoring (CM) systems that falls into the so-called signal processing approach (*signals* can be WT blades vibration, or acoustic emissions and/or thermography measurements of WT internal components), to numerical models of WTs, or data-driven strategies e.g. based on SCADA (supervisory control and data acquisition) (Márquez et al., 2012) (Nabati and Thoben, 2017).

All these strategies can be effectively exploited to endorse the *Predictive* Maintenance, which focuses on identifying the optimal time to perform maintenance, in particular after some WT working condition starts to decline and performance to decrease, but before failure occurs. It, therefore, aims to establish a trade-off between preventative maintenance, which uses strict time-based scheduling and may occur (and cost) too frequently, and reactive or run-to-failure maintenance, where components are repaired only after they have already failed (Selcuk, 2017).

Current studies on WT focused on predictive maintenance because good wind turbine reliability, together with predictable wind turbine maintenance schedules, will result in reduced cost of energy and then wind farm project success. This is even more important for offshore wind farms because of their higher initial capital cost and limited accessibility causing higher operational and maintenance costs (Qiu et al., 2012).

A traditional approach to predictive maintenance is based on the use of SCADA systems, that provide rich information about the plant itself giving signal information and component information.

Carchiolo, V., Longheu, A., di Martino, V. and Consoli, N.
Power Plants Failure Reports Analysis for Predictive Maintenance.
DOI: 10.5220/0008388204040410
In *Proceedings of the 15th International Conference on Web Information Systems and Technologies (WEBIST 2019)*, pages 404-410
ISBN: 978-989-758-386-5

There are a lot of research that, focusing on these systems as a primary source of entry, has achieved a good success in reporting failures and problems within the plant through power-curve and temperature analysis (Nabati and Thoben, 2017).

Some of these research outcomes have been recognized by industry and turned into applications. The main advance in this approach is declined in the use of artificial intelligence to analyze the collected information. For example (Huuhtanen and Jung, 2018) use convolutional neural networks (CNN) for monitoring the operation of photovoltaic panels. The predictive maintenance in this approach is activated when it is observed a large deviation between predicted and actual (observed) power curve, whereas (Helbing and Ritter, 2018), (de Azevedo et al., 2016) and (Romero et al., 2018) show example of the analysis of SCADA data through the use of deep learning for fault detection in wind turbines.

Improving wind turbine reliability requires to reduce downtime and increase availability by optimizing its design and imposing a well-organized maintenance schedule. This requires a full understanding of the system and a detailed analysis of its failure mechanisms and cases, therefore in addition to SCADA systems, a good approach is the exploitation of maintenance report, whose content can be effectively analyzed to extract relevant information about WT failure components, endorsing the predictive maintenance approach.

A maintenance report is a document in which there are important data about the WT and in which there probably is the main cause that led to the correspondent status of the plant at that time. Our goal is to find a correlation between the data shown in the maintenance reports and the possible causes of failure; to this purpose, we use Natural Language Processing (NLP).

NLP was formerly known as Computational Linguistics in the 60s (Wagner, 2016) and it uses computational techniques and artificial intelligence to understand, learn and synthesize human language content. The foundations of NLP lie in several disciplines as computer and information sciences, linguistics, artificial intelligence and robotics, psychology, philosophy, logic and mathematics, electrical and electronic engineering. Applications of NLP include machine translation, natural language text processing and summarization, user interfaces, multilingual and cross-language information retrieval, speech recognition, artificial intelligence, and expert systems, and many others. Recently, in particular in the last 20 years, this affected not only scientific research but also practical applications that have been successfully integrated into consumer products as the Apple Siri (Apple,) or Microsoft Cortana (Microsoft,) personal assistant, or even in more specific context, ranging from personal medical records data gathering, feeding and analysis (Carchiolo et al., 2015) (Carchiolo et al., 2015) to Twitter and/or Web data discovery for several purposes (Carchiolo et al., 2015) (Longheu et al., 2016).

Here we adopt a NLP-based approach together with Ontology-based information extraction for capturing syntactic and semantic relations within words, allowing to leverage maintenance phrases discovering what went wrong and determined the failure.

The rest of paper is organized as follows: In section 2 we address the question of analyzing maintenance report, whereas in section 3 the overall project that leverage information coming from reports is discussed; section 4 briefly concludes our work and show future directions.

2 MAINTENANCE REPORT NLP-BASED ANALYSIS

This section presents an analysis and methodology applicable to the data that have been extracted from maintenance reports of WTs.

In pursuing this goal, the first problem is to discover the meaningful information hidden in maintenance reports and to collect all useful data from.

The main problem is to discover all the information hidden in the maze of a maintenance report that it can, by its nature, be very varied in form and information. The information presents in a maintenance report can have different view:

- structured versus unstructured information
- fault reporting information vs. repairing operation
- measurable and non-measurable information

For example, the fault reporting can be described by either a well-coded data (almost structured) as the alarm code provided by a software tool or a qualitative description, via a natural language, of the unexpected behavior.

In the same way, the repairing operation description can contain the list of the WT replaced parts (each with numbers and costs), the operations carried out for the repair, the work hours but also information about the success of the technical intervention; the latter, probably, hidden in a descriptive sentence written in some natural language.

Fig 1 shows some examples of wind turbine maintenance report and it is notable how they contain a lot of information, with different spatial organization,

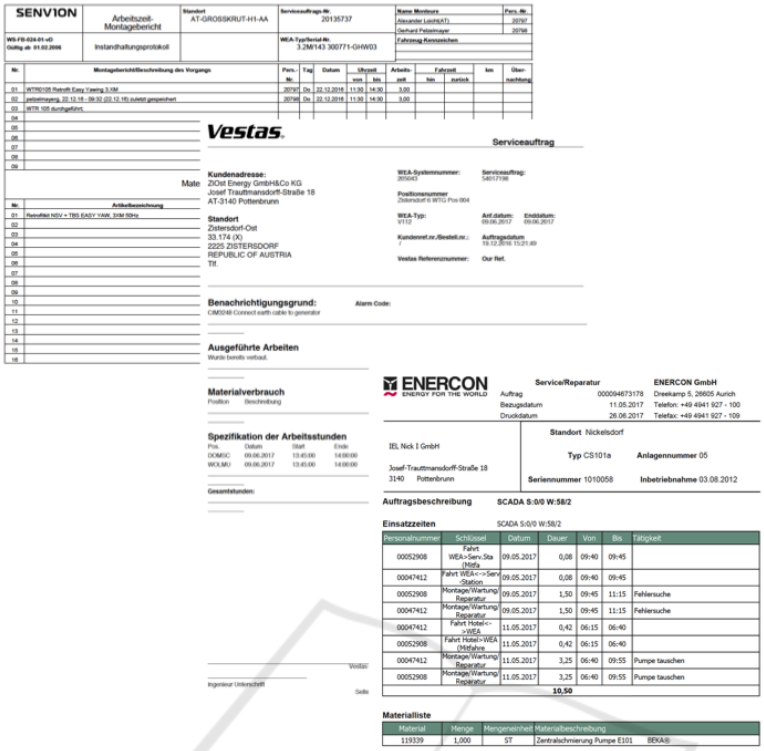

Figure 1: Maintenance Report Examples.

different degree of structuring (free test, lines, number, error code, and so on), several forms and sometimes using also different languages. In fact, each every wind turbine manufacturer use a customized form. It is clear that such data require different techniques due to their heterogeneity, e.g. data mining, sentiment analysis or similar.

Limited research activities have been conducted for detecting failures related to WTs. Sometimes, NLP techniques are applied for the identification of technology trends (Lee and Lee, 2013); they are though applied with the aid of large available input repositories. Those inputs consist of a lot of patent documents from which the analysis can start. The work uses those techniques to identify new innovation patterns in the field of energy technologies.

There are other similar works that focus on a large dataset of unstructured data and apply those techniques for knowledge discovery of accident information in the text. The involved text mining pipeline focuses on those inputs to recognize the main risk factors, which can be similar to the case of WTs maintenance report discussed in this paper. Those accident dataset has been collected over a 12-month period, through collecting data from different available accident datasets on the network. A smaller portion of those documents has been recognized by an expert analysis team and has been used to build the ontology terms in the model proposed in (Ertek et al., 2017); however, this study cannot be used in our use case since there are not equivalent failure dataset available in the network.

There are other researches on the text mining domain that focus on a different technique named FMECA (Failure Mode, Effects and Criticality Analysis), that is recognized as one of the earliest techniques for failure analysis (Bouti and Kadi, 1994). FMECA is used to analyze the reliability and safety of equipment with inductive logic. This work is pretty similar to our use case since it will involve the development of an ontology-based model with detailed concepts. Based on the analysis of the main research contents of fault diagnosis coming from machine tools there are four main concepts that has been recognized by this work. They are known as fault phenomenon, fault maintenance, fault cause and fault location and they are seen as fault events (Zhou et al., 2017). Those concepts will be used to retrieve text information inside the text with the use of clustering techniques. However, this approach cannot be useful to our domain case because it reflects a superset of our failure maintenance reports sentences and because the developed ontology has been build with a bigger dataset and with specific domain information

like frequencies of failures, severities, locations, motivations that are not available in our use case.

In (Küçük and Arslan, 2014) authors propose a semi-automatic approach to build an ontology model for wind energy domain. This process involve the crafting process of different Wikipedia articles related to the wind energy domain. However as the previous one this cannot be used to setup the proposed ontology model because it reflects a different domain case.

Our solution involves NLP techniques and Ontology-based information extraction for capturing syntactic and semantic relations within words in a text, as specified previously. Data is composed of maintenance reports regarding WTs maintenance activities and provided by Bax Energy. A main shortcoming is that reports provided as input are not enough, because only a small part of them contains analyzable data.

These reports make references to maintenance phrases that can be analyzed to find out what went wrong or not within the same report. The use of these techniques will allow identifying domain concepts within the text and if they respect the relationships of the proposed ontology. In fact, it contains all the relationships that have been built between a single failure and potential agents of failure that can provoke it, structural components of the plant that may incur at that specific failure and the condition that triggered that failure event for that specific component. So, the work shows the combination of a semantic module, that is based on the ontology information retrieval, with a set of natural language processing techniques that goes deep into the sentence to clean up it, tokenizes it and retrieves possible syntactic dependencies that are useful to build up domain individuals.

3 THE WEAMS PROJECT

In the context of WTs, if we want to find a correlation between the data shown in the maintenance reports and the possible causes of failure, first of all, it is necessary to gather a significant data amount for an analysis purpose. This data can already be collected from the developed software and also from different sources. In fact, the document will show how the grafting of different types of data, or rather the extension of our dataset can not only specify and further enrich our analysis purpose but also improve the final result in terms of accuracy.

In this section, we will present the solution of Weams project; the proposal is schematized in fig. 2, where several modules are present, i.e. the Data collector, data parser, data reader and matcher and finally the data writer joint with translate module. All these components are described in the following paragraphs.

3.1 Data Collector

The first module is named data collector and is responsible for the acquisition of the relevant parts of interest from maintenance reports. Uploaded documents pass through the PDFBox library (PDFbox,), used for preliminary reading of each string in the file.

The algorithm starts by collecting all relevant parts as defined in the user template. They are divided into static and dynamic parts, whose length may vary within each user document. Since the extraction area for fixed parts is fixed for every document, such parts can be extracted immediately, whereas dynamic parts require the algorithm looks for all the content that belongs to them. Read strings are stored into a specific data structure for further processing and associated with the relevant part of interest. In addition, the library provides important information like the reference coordinates within a 2D space of each character extracted from the document, to be used later to remap the content. Data acquisition also provides the removal of unwanted data as specified by the user (e.g. header/footer).

Once the *noise* data has been removed, the acquisition operation starts by comparing each read string with the one entered by the user in order to map that specific part. If this comparison succeeds, the algorithm will register each subsequent string in a structure that will be identified by the compared string. The structures can have different composition, depending on the nature of the current part. If it is reported from the template as table, the related structure will also have a list of *TextPagination* objects. Each object has a list of *TextPosition* properties, the whole information that are taken from the PDFBox library, and the relative page in which this string has been found. This information is important for the next step in which the content is ordered before building the related table structure.

When the part is reported as free text, the structure stores the starting point as X-Y 2D coordinates of each string, the related endpoint and the page where it has been found. Since this occurs for each page, the content of a page does not depend on others. Moreover, we consider the membership page of each string and its order and we compare with others' in the list to find the maximum value of that page. Such value is used as the start value of the first string on the next page. All strings that belong to pages after the first will see the value of the ordinate changed, which is

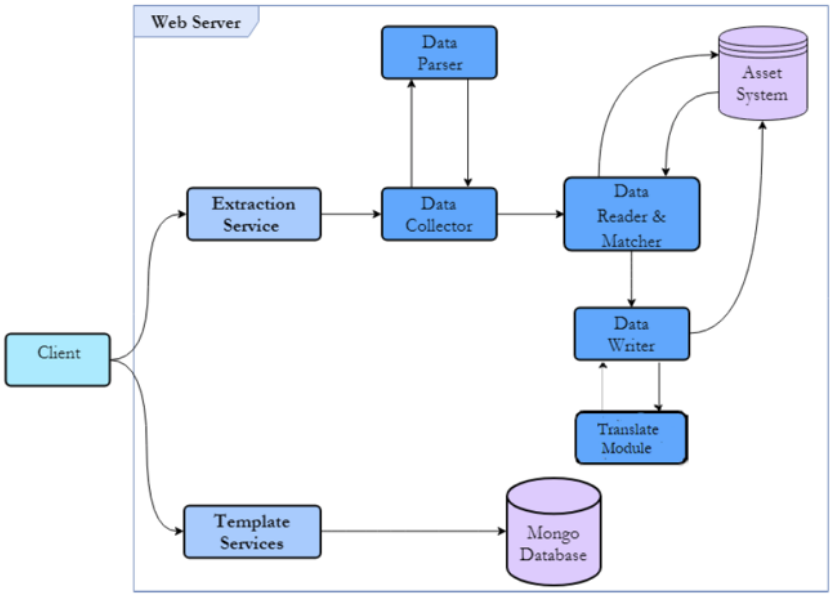

Figure 2: Weams Report Architecture.

not related to the previous page. This value is modified with the sum of the maximum value found on the previous page, i.e. the maximum ordinate (Y) of the last useful string on the previous page, and their current position on that page. This procedure for evaluating minimum and maximum values for each page is performed for all the pages in the document. After establishing a vertical data order, we then aggregate each row's content by a horizontal arrangement. Once these operations are completed, the content thus ordered and collected for each part of interest is ready to be further processed by the parser.

3.2 Data Parser

This module takes as inputs the different parts extracted from Data Collector and processes the parts in tabular format. In fact, the tabular information is still not homogeneous therefore it will be necessary to create a structure in rows and columns for each record mapped within the part being examined. The process of building the table foresees the use of the TrapRange library (Luong, 2015) which has been modified for our case. Specifically, the changes made allow a correct interpretation of tables whose headings are divided into several lines, or in which there are columns that subsequently branch into two or more columns and a correct interpretation of the rows of a table that can develop on more lines.

Concerning the first point, the user can specify inside the template which columns exhibit bifurcations, passing the index of the column of the interested table (0-based). This will allow the algorithm to exclude the word mapped to this index and to take into consideration the columns underlying it. As a result, the table will be in a structured and homogeneous form and the analysis of the number of rows and columns of the same can be carried out. This analysis is done by the software using TrapRange which identifies the limits of each row and each column of the table.

A drawback of this point concerns the identification of multi-lines rows, that are composed of several lines, that are not correctly recognized in the table. Once they are processed by the software they will have more lines than actually (the content is split unevenly into different lines). To make sure that the entire content is collected within a single row, a reference parameter has been chosen. It must be a value that exists within each table of interest and for this reason, it has been chosen as the content of the first column which represents always an identifying value, always reported, or a row index, always reported too. For collecting these reference points the algorithm create specific intervals for each word/character; these intervals record the height of the character and represent the range of integers that start from a lower minimum, the ordinate of the character, up to a higher maximum, the sum of the Y coordinate of the character plus the height of the character (in order to map the interval in which a character is positioned within the text). Intervals falling under the first column (that has its own interval range) constitute the content of the first column for each different row. In this way, excluding the null values, we have the positional ranges for each content of the first column for each row and the algorithm can proceed to the second step.

In summary, the algorithm computes an interval for each content that falls in the columns following the first column, with the same indexes already described, and the distance between the aforementioned range and each reference interval. The minimum distance value represents at which reference range that content refers. Once this procedure has been carried out for each reference interval a single line can be constructed. It is made up of as many cells as the number of columns in the table and each cell has its own content, even if distributed over several lines. Having done this for each part of interest the content is considered well structured and can be further processed to be written into the database.

3.3 Data Reader and Matcher

This module is responsible for the operation of reading the instances already present in the final system. The model file that has been created by the system on user entities (see the section below) comes in help for this purpose. It contains all the properties and relations and all data types of these properties. Moreover, it contains the address of each entity that has to be read from the system and written into the system; it is necessary to this module and to the writer module. In addition, in accordance with the parameters that have been set by the user during the creation of the template, some of these data must already be present in the system. As a result they must be referenced and not created and conversely, those that are not present in the database will be created. For the first case, the software reads the related objects already present in the system for each entity referenced by the user. For those entities that must match the software takes and stores the relative object that match. It will be sent as input to the writing module. Finally, the matching approach is based on the user-entered properties, different for each reference entity.

Note that for those objects that provide a match and consequently must already exist within the final system, the software will stop the processing of the current file giving an error in the case the entity is not matched/found. It will continue with the next file inserted by the user in the extraction process, restarting from the first module (the data collector).

3.4 Data Writer and Translate Modules

The writer module stores data present in the document on the final system. It receives as input all the references to instances already present in the system to be referenced (via an identification number) and the various patterns that will be used to create these objects (template patterns) before writing them. These patterns have been created to allow the user to extract different parts within a single block of text through regular expressions. The user must specify the name of the property the related content will refer to. Moreover, these patterns allow describing the properties of each column within a table. Then, the rich texts that will be involved in the analysis are translated through another module. It interfaces with a service exposed by Yandex dictionary (Yandex,) to detect text language and translate it into English (maintenance report include information in Deutch and other languages). It has been developed to facilitate the construction of a common reference dictionary. Finally, structured data is converted into JSON format and sent to the final system.

4 CONCLUSIONS

The approach discussed so far shows how textual data can be crafted and analyzed, starting from maintenance reports, with the application of a set of natural language processing and ontological techniques. The obtained results have been satisfactory, regarding the failure classification, notwithstanding the limited test dataset. This can be the first step on a true evaluation step that will be based on a much larger dataset than this. Its extension will also involve a larger ontological model that will be useful to encapsulate other failure phenomena or other more structural components within the plant or other components which will further enrich the text analysis. A deeper comparison with other works in the same domain (e.g. (Kusiak and Li, 2011)) is also planned.

ACKNOWLEDGMENTS

This work has been partially supported by Wind Energy Asset Management System (WEAMS) Project, endorsed by UE, Italian Ministry of Economic Development (MISE) and PON ”Imprese e Competitivita’ - Iniziativa PMI 2014-20”

REFERENCES

Abichou, B., Flórez, D., Sayed Mouchaweh, M., Toubakh, H., Francois, B., and Girard, N. (2014). Fault diagnosis methods for wind turbines health monitoring: a review.

Apple. Siri - apple. https://www.apple.com/siri. Accessed: Apr 2, 2019.

Bouti, A. and Kadi, D. a. (1994). A state-of-the-art review of fmea/fmeca. *International Journal of Reliability, Quality and Safety Engineering*, 01(04):515–543.

Carchiolo, V., Longheu, A., and Malgeri, M. (2015). Personal health record feeding via medical forums. In *2015 IEEE 19th CSCWD Intl. conf.*, pages 632–636.

Carchiolo, V., Longheu, A., and Malgeri, M. (2015). Using twitter data and sentiment analysis to study diseases dynamics. In *Proceedings of the 6th ITBAM conf.*, pages 16–24, New York, NY, USA. Springer-Verlag New York, Inc.

Carchiolo, V., Longheu, A., Malgeri, M., and Mangioni, G. (2015). Multisource agent-based healthcare data gathering. In *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 1723–1729.

Council, G. W. E. (2019). Global wind energy council (gwec) - global wind report. Technical report.

de Azevedo, H. D. M., Araújo, A. M., and Bouchonneau, N. (2016). A review of wind turbine bearing condition monitoring: State of the art and challenges. *Renewable and Sustainable Energy Reviews*, 56:368 – 379.

Ertek, G., Chi, X., Zhang, A. N., and Asian, S. (2017). Text mining analysis of wind turbine accidents: An ontology-based framework. *2017 IEEE Big Data conf.*, pages 3233–3241.

Fischer, K., Besnard, F., and Bertling, L. (2012). Reliability-centered maintenance for wind turbines based on statistical analysis and practical experience. *IEEE Trans. on Energy Conversion*, 27(1):184–195.

Guolin, H., Ding, K., Li, W., and Jiao, X. (2016). A novel order tracking method for wind turbine planetary gearbox vibration analysis based on discrete spectrum correction technique. *Renewable Energy*, 87:364–375.

Helbing, G. and Ritter, M. (2018). Deep learning for fault detection in wind turbines. *Renewable and Sustainable Energy Reviews*, 98:189 – 198.

Herbert, G. J., Iniyan, S., and Goic, R. (2010). Performance, reliability and failure analysis of wind farm in a developing country. *Renewable Energy*, 35(12):2739 – 2751.

Huuhtanen, T. and Jung, A. (2018). Predictive maintenance of photovoltaic panels via deep learning. In *2018 IEEE Data Science Workshop (DSW)*, pages 66–70.

IEA. International energy agency - modern bioenergy forecast. https://www.iea.org/newsroom/news/2018/october/. Accessed: "Aug 5, 2019".

IEA. International energy agency - world energy outlook. https://www.iea.org/weo/. Accessed: "Apr 2, 2019".

Küçük, D. and Arslan, Y. (2014). Semi-automatic construction of a domain ontology for wind energy using wikipedia articles. *Renewable Energy*, 62:484 – 489.

Kusiak, A. and Li, W. (2011). The prediction and diagnosis of wind turbine faults. *Renewable Energy*, 36(1):16 – 23.

Lee, K. and Lee, S. (2013). Patterns of technological innovation and evolution in the energy sector: A patent-based approach. *Energy Policy*, 59:415 – 432.

Longheu, A., Previti, M., and Mangioni, G. (2016). Tourism websites network: crawling the italian webspace. In *Proc. of 5th Intl. conf. on Data Analytics*, pages 131–136.

Luong, T. Q. (2015). Traprange: a method to extract table content in pdf files.

Microsoft. Cortana. your intelligent assistant across your life. https://www.microsoft.com/en-us/cortana. Accessed: "Apr 2, 2019".

Márquez, F. P. G., Tobias, A. M., Pérez, J. M. P., and Papaelias, M. (2012). Condition monitoring of wind turbines: Techniques and methods. *Renewable Energy*, 46:169 – 178.

Nabati, E. G. and Thoben, K. D. (2017). Big data analytics in the maintenance of off-shore wind turbines: A study on data characteristics. In *Dynamics in Logistics*, pages 131–140, Cham. Springer International Publishing.

PDFbox. Pdfbox open source java pdf library. http://www.pdfbox.org/.

Qiu, Y., Feng, Y., Tavner, P., Richardson, P., Erdos, G., and Chen, B. (2012). Wind turbine SCADA alarm analysis for improving reliability. *Wind Energy*, 15:951–966.

Romero, A., Soua, S., Gan, T.-H., and Wang, B. (2018). Condition monitoring of a wind turbine drive train based on its power dependant vibrations. *Renewable Energy*, 123:817 – 827.

Selcuk, S. (2017). Predictive maintenance, its implementation and latest trends. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 231(9):1670–1679.

Wagner, S. (2016). Natural language processing is no free lunch. In *Perspectives on Data Science for Software Engineering*, pages 175 – 179. Morgan Kaufmann, Boston.

Yandex. Yandex online dictionary. https://translate.yandex.com/. Accessed: Apr 2, 2019.

Zhou, Q., Yan, P., and Xin, Y. (2017). Research on a knowledge modelling methodology for fault diagnosis of machine tools based on formal semantics. *Advanced Engineering Informatics*, 32:92 – 112.