Benchmarking for Keyword Extraction Methodologies in Maintenance Work Orders

Thurston Sexton[1], Melinda Hodkiewicz[2], Michael P Brundage[1], Thomas Smoker[2]

[1] *National Institute of Standards and Technology, 100 Bureau Dr, Gaithersburg, MD 20899*
thurston.sexton@nist.gov
michael.brundage@nist.gov

[2] *Faculty of Engineering and Mathematical Sciences, The University of Western Australia, 35 Stirling Hwy, Crawley WA 6009*
melinda.hodkiewicz@uwa.edu.au
thomas.smoker@research.uwa.edu.au

ABSTRACT

Maintenance has largely remained a human-knowledge centered activity, with the primary records of activity being text-based maintenance work orders (MWOs). However, the bulk of maintenance research does not currently attempt to quantify human knowledge, though this knowledge can be rich with useful contextual and system-level information. The underlying quality of data in MWOs often suffers from misspellings, domain-specific (or even workforce specific) jargon, and abbreviations, that prevent its immediate use in computer analyses. Therefore, approaches to making this data computable must translate unstructured text into a formal schema or system; i.e., perform a mapping from informal technical language to some computable format. *Keyword spotting* (or, *extraction*) has proven a valuable tool in reducing manual efforts while structuring data, by providing a systematic methodology to create computable knowledge. This technique searches for known vocabulary in a corpus and maps them to designed higher level concepts, shifting the primary effort away from structuring the MWOs themselves, toward creating a dictionary of domain specific terms and the knowledge that they represent. The presented work compares rules-based keyword extraction to data-driven tagging assistance, through quantitative and qualitative discussion of the key advantages and disadvantages. This will enable maintenance practitioners to select an appropriate approach to information encoding that provides needed functionality at minimal cost and effort.

1. INTRODUCTION

Maintenance is a vital function in every industry, including manufacturing, construction, chemical, infrastructure asset management, resource extraction industries, and many others. It involves all actions necessary to ensure a piece of equipment is in a state suitable to safely and consistently perform a required function (AS IEC 60300.3.14, 2005). Thus the related theory of maintenance practice can be split into *strategy development* and *work management* components. (Márquez, 2007; Kelly, 1997; Palmer, 1999) This paper focuses on the *work management* component of maintenance.

Work management includes the maintenance processes of work identification, planning, scheduling, execution, completion, and review. Data generated through these processes is typically captured using a maintenance work order (MWO), and while data about maintenance tasks differs from domain to domain (or even company to company within a domain), some or all of the following data are usually collected: 1) the asset and/or its components, 2) observed symptoms, 3) the time of failure, 4) the time for maintenance, 5) possible causes, 6) actions taken, and 7) the name of the technician(s). They often contain a mixture of human-generated, unstructured text, and structured field entries. These fields usually take the form of drop-down menus, lists, or entry fields, and times/dates for items such as order creation, progress, and completion.

This data is primarily recorded by the technicians who actually perform the maintenance; given that there are multiple technicians within an enterprise, the human-generated data is often inconsistent, error-filled, and replete with domain specific jargon. For example:

Technician A: *"bearing broken at Station 1"*
Technician B: *"bearing failure at cutoff unit of S1."*

Both of these representations describe the same overall problem of "broken bearing," located at "Station 1," but they take very different forms, especially if the end goal is to perform automated analysis (like finding all instances of "broken bearing" on this data). If this data could be parsed, it could lead to calculations such as failure mode identification, rework, problem spot identification, and more accurate mean time to repair (MTTR) or mean time between failure (MTBF), which can lead to improved maintenance strategy, reduced risk of failure and improved maintenance efficiency.

There have already been some successes in parsing these types of records in other domains. This is specifically true in the medical field, given the parallels between MWO records and patient medical records: both record symptoms, diagnoses and actions taken using unstructured text. Indeed, considerable work has been done on mining text in patient records; Heinze et al. (2001) applied natural language processing (NLP) across a wide variety of medical domains, while Tremblay et al. (2009) specifically applies it to fall-related injuries in veterans, and the *MedIE* system extracts and mines text from clinical medical records, generally (Zhou et al., 2006). While these efforts are significant achievements, the medical field has significant advantages over the domains we are concerned with:

1. data-sets tend to be much larger and cover longer timespans by comparison, and
2. there are widely adopted controlled vocabularies available through medical ontologies.[1]

A number of research efforts exist in the engineering domains being addressed, that attempt to mitigate these issues.

1.1. Size of Datasets

How do maintenance practitioners in engineering obtain large quantities of data to robustly perform statistical analyses? In the authors experience within the manufacturing and mining equipment datasets, MWO dataset sizes for individual companies range from a few thousand records to upwards of one million MWOs each year. This quantity of MWO data is smaller than what most out-of-the-box solutions for NLP are built for (and regularly validated on), such as "tweets," or Amazon product reviews (Davidov et al., 2010).

One scheme to circumvent this issue promotes sharing data within a domain, and using a then-standardized dataset as training for data-driven tools that clean and analyze new data. This sharing is difficult when MWOs might contain proprietary information (machine identification numbers, technician names, specific processes for specific parts, etc.). There is ongoing research in anonymizing data-sets for this purpose, for example, focusing on "usefulness" as measured by information utility functions (Fang & Chang, 2008). Even using such an approach, information in a data-set will not be perfectly anonymized — there is a trade-off between privacy, and how much useful information remains for sharing.

1.2. Use of Ontologies

In the past, developers of Computerized Maintenance Management Systems (CMMS) have tried to ensure proper data structure through enforcing controlled vocabulary and problem code assignment for MWOs. In practice, these approaches have had limited success in improving data quality (Molina et al., 2013; Unsworth et al., 2011). With maintenance data especially, language used by one group (the maintenance technicians) can be quite different to that used by others (the engineers, or CMMS developers) (Murphy, 2010). Subsequently, the codes provided by engineers are often inadequate for expressing of the details of the event or action the maintainers take. Further, interpretations of events differ among the technicians themselves, and individuals might choose different codes for the same event.

There is growing interest in the potential value of ontologies to codify structures of *meaning* for maintenance. Early developments include the European project Proteus in 2005 from Rasoyska et al., with more recent work by, for example, Karray et al. (2012); Ebrahimipour & Yacout (2016). In the process-plant and engineering design sector, ISO15926 Standard Formal Ontology (ISO, 2003) could potentially be used for through-life support data. To date, however, there has been little uptake of ontological approaches by industry—in part because they have been developed in isolation. As a result, they are rarely interoperable, and lack scalability (Semy et al., 2004).

There remains a need for an agreed-upon upper ontology for maintenance. Current projects, such as the adaptation of Basic Formal Ontology (BFO) to the manufacturing sector (Arp et al., 2015), do include a sub-focus to provide an ontology for maintenance in manufacturing. Alternatively, the use of Natural Language Processing (NLP) to extract relevant information from the unstructured data sets promises to directly provide insights and analytics, even while maintainers continue to enter data in their own words. (Sharp et al., 2016; Sexton et al., 2017). This approach is somewhat ironically limited by the size of available training examples, mentioned previously.

[1] An ontology defines a machine-readable vocabulary to enable reasoning and with which queries and assertions are exchanged. Notable developments in medicine to underpin this capability include: SNOMED (Spackman et al., 1997), a nomenclature for human and veterinary medicine; the GENE Ontology for biology (Ashburner et al., 2000), a tool for the unification of biology; and the Unified Medical Language System (Bodenreider, 2004), a repository of biomedical vocabularies developed by the US National Library of Medicine.

1.3. Paper Outline

Informed by the dichotomy discussed, this paper compares two promising methods for automated data structuring through keyword extraction: a data-driven tagging method vs. a rules-based expert system. A publicly available mining equipment data-set is used to compare these methods for cognitive load on the human using these techniques, the ability of the method to calculate maintenance specific metrics (Median Time to Fail/ MTTF), and identification of problem-spots. The rest of the paper is structured as follows: Section 2 discusses background of both the rules-based method and the tagging method; Section 3 describes the data-set and how the two methods are compared, while Section 4 discusses these results; lastly, Section 5 presents conclusions and future work.

2. METHODS FOR ENCODING INFORMATION

We present a comparison of both previously discussed methodologies for encoding the tacit knowledge in MWOs into a more structured format. While obviously an incomplete overview of solutions to this common problem, we hope that the two selected methods are representative of two *archetypes* within the domain, namely: precisely-engineered, initially labour-intensive automation through design of "rules"; and data-driven, human-in-the-loop extraction that theoretically sacrifices precision for ease-of-use and statistical representativeness.

2.1. Rules-Based Methods

In rule-based data processing unstructured data is transformed into a predetermined format using explicit rule sets. These rule sets, often called "expert systems", are comprised of a series of 'if condition then perform action' statements where an action is performed if the given conditions are satisfied. Rule-based data processing requires progressive iteration of rule development and application in order to tune data sets to be capable of transforming unstructured data into the appropriate format (Rahm & Do, 2000; Prasad et al., 2011). An example output of a Rules-Based Method can be seen in Fig. 1.

It is important to have a purpose for the data structuring. In the case of maintenance work order records a common aim is to identify end-of-life events so that reliability metrics such as MTTF can be calculated. Other aims are to identify failure causes, track rework and develop troubleshooting capabilities. In each case a minimum viable data set to support the intended analysis needs to be identified. For calculation of time-to-end-of-life event, beyond just having a sufficient number of repeated events to sufficiently characterize the TTF distribution, the necessary data types are: an identifier of the asset or maintainable item that reached end-of-life, an identifier of the end-of-life event, the usage based on hours, distance, cycles or other measure to calculate life, and a means of identifying if the end-of-life event was due to censoring or not. Right censoring occurs when an item is removed before it reaches its end-of-life and when the observation period for data collection ends but the item is still in use.

Challenges in rule-based structuring include managing rule sets of increasing complexity and size. Rule sets are often executed sequentially with the order of rules important in determining the outcome of the transformation. Rule conflicts can occur which require the use of conflict resolution, often manual in nature to determine the appropriate output. As rule sets grow in size they become increasingly hard to manage, with each additional rule providing incrementally less benefit, yet with a possibility of degrading any previously executed rules.

2.2. Data-Driven Tags Method

Another approach to data-structuring is to derive patterns for recognition of good data using statistical aggregation, or any of several machine learning techniques. In this paradigm, text is processed in order to be represented "numerically". Previous work has compared several ways of using Natural Language Processing (NLP) on MWOs, including Bag of Words models and Word2Vec (Sharp et al., 2016). Regardless of the technique, the goal is to develop a computational representation of the text, that captures some fundamental statistics in the original language, and to develop a machine-learning (ML) pipeline to *predict* the correct organization of some set of work-orders. The primary issue then becomes creating a data-set to train the ML model, which is a labor-intensive task requiring at least a tacit rules system. Additionally, the amounts of data involved in this domain, as noted above, are smaller than typical use-cases for NLP, with high technicality, and not nearly sufficient examples to statistically cover the broad functionality-space.

To circumvent this issue, it is possible to use the concept of *tagging* as a form of user annotation of MWOs, to balance structure and flexibility. Tags are un-controlled, multi-label feature-assignments of text (or anything, potentially) that can be mapped quite easily to a bag-of-words representation. The problem now becomes creating a mapping between the existing language in historical MWOs and the set of tags that user might want to represent the MWOs through. Here, as in Sexton et al. (2017), we exploit statistical aggregation methods used in NLP (specifically, term-frequency/inverse-document-frequency weighting) to present users with the "most important" text-fragments—called "tokens"—*first*, allowing an annotator to generate a tag vocabulary list for post-facto automated extraction of these tags from historical MWOs. The output of *tagging* can be seen in Fig. 2. This allows the technicians to continue writing text with abbreviations and highly domain-specific, technical descriptions, while allow-

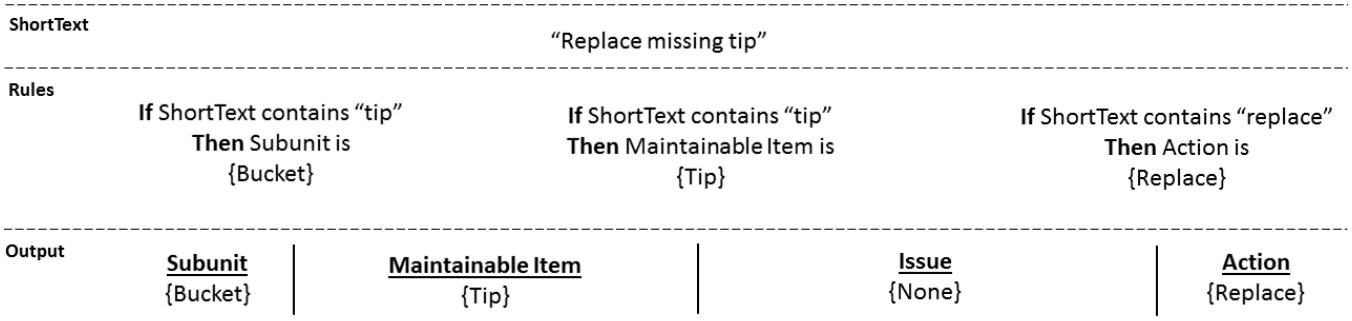

Figure 1. Illustrating the Rules Based Method for transforming MWOs. Rules are created given an expert's knowledge of the MWOs, however, they can grow with complexity as more situations are encountered. For example, if there are more "Maintainable items", the rules need to account for how to address multiple items. The Rules Based Method can handle misspellings, jargon, and abbreviations if the expert knows them beforehand. As seen in the figure, there is no rule for the "Issue" for this case, therefore, none is assigned.

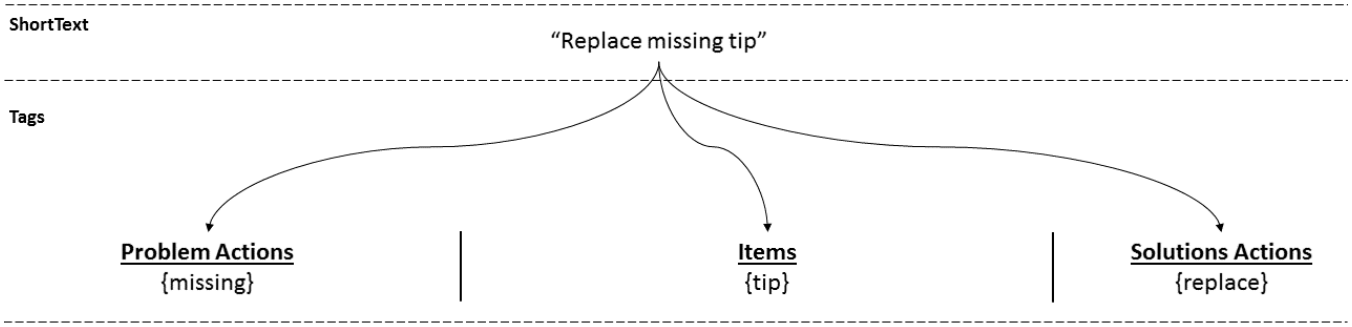

Figure 2. Illustrating the procedure for tagging a MWO. The tags are created using the original MWOs and are mapped to tags by a human expert. The output is the *Items*, *Problems*, and *Solutions*. This method creates structured data from the unstructured natural language MWO data. The tagging method can account for misspellings, jargon, and abbreviations in the original MWOs, but is dependent on how often these anomalies occur and how many tokens the human expert encounters. The tagging method, on its own, can not easily determine systems and subsystems (called Subunit in Fig. 1), while the Rules-Based Method can capture that information with the aid of a human expert.

ing an annotator/tagger to automatically extract tags from this text at their desired level of abstraction, thus drastically reducing the number of low-frequency concept-occurrences (see example experiment in Fig. 4).

Once a set of tags has been assigned to the set of MWOs, it is possible to perform queries by boolean set operations on tag occurrences. For example, if a sufficient number of related tokens have been mapped to the "broken" tag, a query over "broken" (conditional on particular machines) could be used as a good proxy for failure occurrence markers. Additionally, tag co-occurrences are naturally represented as graph "edges" between tag "nodes", making available a suite of graph database techniques and advantages.

3. DESCRIPTION OF EXPERIMENT

This comparative study consists of two steps: First, in the *information encoding* step, both methods described above are implemented on a single dataset, to perform the desired knowledge extraction. Subsequently, the structured forms are used to perform basic survival analysis, by approximating the labels for several major subsystems within the dataset, deriving these labels through the information encoded with both methods.

3.1. Information Encoding

The data set is an extract of maintenance work order records from a Computerized Maintenance Management database for five 1400 HP mining shovels. The data are publically available through the UWA Prognostics Data Library (Sikorska et al., 2016). Four fields are used in this analysis: *Date the Work Order is created, Asset identifier, Short Text,* and *Cost*. The *Short Text* field contains unstructured text populated by the individual generating the work order usually the asset operator, maintainers, maintenance planner or supervisor. Information that may be contained in the Short Text field is: the reason for the maintenance activity such as the observed symptoms of failure, a description of the work performed, the subunit(s) or maintainable items on which the work was performed and po-

sitional information (e.g right side, under). Typical examples of *Short Text* are 1) *Replace centre and LH lip shrouds* which describes the work performed (*(replace)* and the component *(centre and left hand lip shrouds)* and 2) *Broken grease line on bucket* which describes the problem *(broken grease line)* and the component (bucket). The data set used in this analysis (5485 records) has had the following cleaning prior to being made available in the data library: work orders were discarded due to issues such as incorrect functional location allocation, duplication, an absence of hours or costs logged and if the work order did not result in the repair or replacement i.e., inspections are not included.

3.1.1. Rules-Based Case Study

Rule files (also known as token files, not to be confused with NLP text "tokens") are constructed as a series of "if condition perform action"statements. Condition statements are comprised of between one to three logic statements: the location in the unstructured data-set to search; logical operator (choices of: equals, not equals, has, excludes, >, <); and patterns (regular expressions or numeric values) required to satisfy condition. Pattern matching (keyword spotting) is designed to be case insensitive and includes the ability to search through grammar, white space and alphanumeric search terms. Additional functionality includes: rule-conflict resolution to identify where multiple rules provide conflicting classifications on any given record, rule frequency statistics on how many times a rule action was activated, and records of the sequence of rules executed on each individual event.

Rules are developed using a piece-wise approach and stored in generic rule libraries, with one library per field to minimize interactions and mismatches between rule libraries. This allows the successful execution of other rules even in the presence of missing information that may cause other groups of rules to fail. The partitioning of rules allows development and reuse of rule libraries that can be used across multiple maintainable items. Generic rule libraries are developed by the selection and examination of training sets from more than one million MWOs for mobile mining assets such as haul trucks, shovels, excavators, loader and drills (Ho, 2015). These rule libraries are for a) maintenance action performed, b) failure mode or observed symptom of failure, c) maintainable item (for partial failures), d) location Identifier (e.g., "left rear" or "position 1"), e) active repair time, and f) down time. The development and subsequent reuse of generic rule libraries reduces development time.

All the rules are compiled into a token file. This token file was used to structure the data set for the five mining shovels used in this paper. The token file has 469 rules. Records require a minimum of three rules and can need as many as eleven.

Data about the problem and action are contained in the *Short-Text* field of the MWO. This has no set sentence structure, contains a high number of unique entries and technical jargon, abbreviations and spelling mistakes. Keyword spotting is used due to the prevalence of unambiguous keywords (e.g., "Engine" or "Leaking") which can be mapped directly to fields in the required minimum data-set such as asset or end of life event. Rules are developed manually to correct jargon, misspellings, variants and abbreviations.

The *Work-Order Type* and *Short-Text* fields are used to determine censoring status for the maintenance event. An event is classified as a failure if it contains one or more of the following criteria: *WorkOrder Type* code corresponding to a corrective maintenance code of the organization, a recorded failure mode, failure cause or symptom of failure in the *ShortText*, recording of a non-preventative maintenance action in the *ShortText*, recent job request or work order with a *ShortText* field recording symptoms of failure (job requests or work orders are classified as recent if they occurred since the previous scheduled service event), or active usage time of the maintainable item is less than half of the expected replacement interval recorded in the organization's maintenance plan. Events are classified as censored when the asset has been replaced; the work order type corresponds to the preventative maintenance code of the organization; and there are no recorded symptoms of failure or corrective maintenance actions, machine rebuilds or overhauls at fixed intervals, accidental damage, and secondary failures resulting from failure of a different maintainable items. Finally, all data is right censored at the end of the data collection period.

Data sets structured using the rule-based keyword spotting system are checked manually. Issues include duplicate rules and the need for conflict resolution if rule mismatches occur. For example when one rule classifies a work order as preventative yet another rule identifies the failure mode of breakdown. Identification of statistical outliers in time-to-failure data is used to review rules leading to the outliers. Other flags are when the time-to-failure interval is greater than fixed replacement interval specified by the organization's maintenance plan. Coded fields such as the Asset (also called Functional Location) and Work-Order Type fields can also contain inaccurate entries. Functional Locations fields are often recorded with values for an incorrect maintainable item or to a higher level in the functional location hierarchy. Numerical fields such as Total Actual Cost, Total Planned Costs and Man-hour fields are often populated by null values or with values that only reflect a partial cost of the maintenance work. Data quality issues in these coded and numerical fields necessitate the cross correlation of mismatching or missing information from text based fields. Interpretation of free-form text fields such as the Short-text field is required to cross-correlate with other data fields and extract relevant data elements that may be absent such as the identification of the maintainable item.

At the conclusion of data structuring an Excel file is created containing the new fields identifying the sub-system and maintainable item, the maintenance action, failure mode (if available), time since previous event, and a censoring indicator.

3.1.2. AI-Assisted Tagging Case Study

Starting from the same raw-text MWO descriptions as outlined above, text-fragment tokens are parsed from the entire corpus, and passed to a graphical user interface (GUI) that:

- presents an expert with tokens to "tag" in order of their TF-IDF score (Leskovec et al., 2014).
- suggests a list of potentially related tokens from the corpus, to promote terminology unification
- prompts a classification of each tag as "Problem", "Solution", or "Item".

An expert spent 60 minutes using this GUI to create and classify tags, saving their progress after every 10-minute interval. Fig. 3 demonstrates how quickly a large portion of the MWOs have a near-total classification rate—this can be calculated via *positive predictive value* (PPV)[2] by comparing the total number of raw tokens found in the work-order (i.e., the positives) to the number of tokens that have a valid tag mapping and classification (true-positives) [3] . Another way to measure the effectiveness of the tagging process as a form of terminology unification is by comparing the tag-frequency distribution to the original token frequencies (shown in Fig. 4). Far from the tokens—whose most common frequency is 1 by a large margin—tags created in this case-study are most likely to have between 5 - 15 occurrences, making this representation of the data much more amenable to statistical techniques.

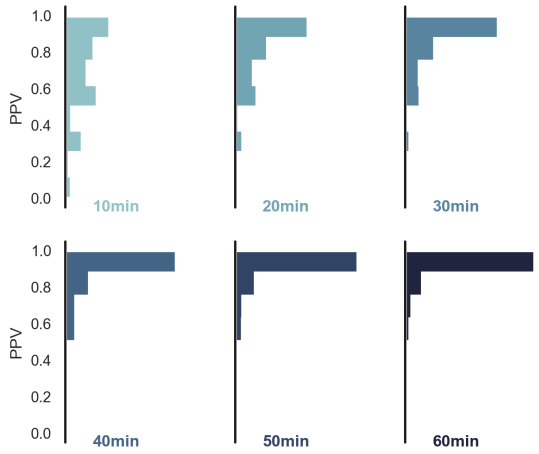

Figure 3. **Information encoding over time** — This figure demonstrates that the fraction of observed text-fragments/tokens that have a defined tag (with corresponding classification) increases rapidly as the tagger annotates the importance-ranked vocabulary list.

3.2. Application Case Study: Survival Analysis

To compare the usage of each structuring approach, a basic application of survival analysis is performed, with the goal of comparing the median time to fail for assets, according to several major subsystems. This can be done with both *parametric* and *non-parametric* models, from which Kaplan-Meier estimation and Weibull distribution models are represented here, respectively. All analysis was completed using the *lifelines* python package (Davidson-Pilon et al., 2018).

3.2.1. Kaplan-Meier Estimation

When there is a sufficient number of observations available, one can approximate the survival function of a population through a non-parametric estimator, the most well-known of

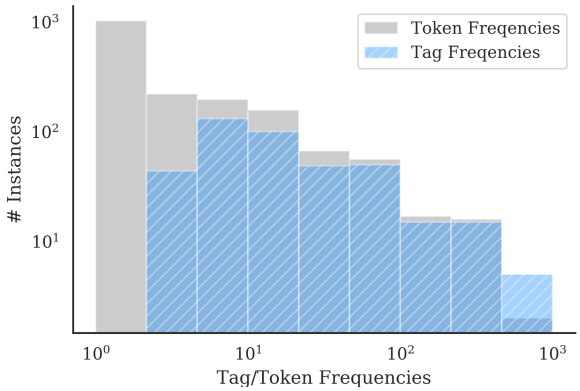

Figure 4. **Token vs. Tag frequency distributions** — the effect of mapping multiple low-occurrence tokens to some unified tag representation has a marked effect on the overall frequencies, dramatically decreasing the number of 1x or 2x occurring tags, and increasing the frequency of the most-recurring concepts, as is desired for statistical analysis.

[2]Typically called "precision" in an information-retrieval context.

[3]In theory, a complete rule-based method would map all observed, useful tokens to some structured information. In this way, we might assign the rules-based method a baseline PPV of 1.0 for all MWOs, to which the tag-based method is being compared.

which is the Kaplan-Meier (K-M) estimator (Meeker, 1998):

$$\hat{S}(t) = \prod_{i:t_i \leq t} \left(1 - \frac{d_i}{n_i}\right), \quad (1)$$

where each t_i is the time from initialization to the time of some event observation (a failure), d_i is the number of events occurring at t_i, and n_i is the number of population individuals known to survive or not have been censored at t_i. This allows censored data to be taken into account—for example, a part replaced in the course of scheduled maintenance has not *failed*, but is un-observable beyond its time of replacement, making this MWO a right censoring event.

Each point of $\hat{S}(t)$, then, is an estimate of the probability that any given member of a population will survive beyond the time t, given its previous survival up to that point.

3.2.2. Weibull Distribution Fit

It is often the case that, due to the rarity of actual failures for certain assets, one does not have enough data for a robust non-parametric estimate. In these cases, it is common among reliability engineers and others to assume that the set of "life-times" in a population is approximately exponentially distributed, and fit a distribution's parameters to available data. However, a strict exponential model assumes that the hazard rate of failure is constant in the population, meaning that the probability of failure is the same, no matter the age of an asset. When these properties of an exponential distribution cannot be assumed, then a Weibull distribution is often fit to the data; the survival function derived through this model (i.e., 1 − CDF) is given by:

$$S(t) = e^{-(t/\eta)^{\beta}}, \quad (2)$$

allowing for the β parameter to adjust the hazard rate as constant ($\beta = 1$), increasing ($\beta > 1$), or decreasing ($\beta < 1$). For this study, these Weibull parameters are estimated by fit the distribution to the observed failure inter-arrival times via Maximum Likelihood.

4. RESULTS & DISCUSSION

The primary way in which the two methods are compared is through the results of performing basic survival analysis on the data-set, after being structured with each approach. In the rules-based approach, each MWO is assigned to a "Major Subsystem" through application of one or more *rules*, along with a determination of whether the failure in this MWO was censored or not (through application of another rule). Since, in this data-set, the ID of each machine was noted in the MWOs, it is possible to calculate the running time for each machine between maintenance events, conditioned on each subsystem.

For the tag-based approach, it is necessary to approximate membership of each MWO into a subsystem by the set of tags extracted. The most straight-forward way to accomplish this is by selecting one tag that should be maximally representative of the subsystem (for example, the tag "bucket" for the bucket subsystem, etc.), and conditioning the failure inter-arrival times on that tag's occurrence. Obviously this will tend to under-estimate the number of failures, since there will be other objects or occurrences that are indicative of some particular subsystem. For example, if a boom is replaced, to which the bucket is attached (and therefore, also replaced), the "bucket" tag itself may not be explicitly extracted, since the bucket subsystem is only implicitly referenced via the "boom" tag. The single tag query for "bucket" would miss

Table 1. Results of the bench-marking experiment, organized by major subsystem. Queries for a set of multi-tag input $t_i \in T$ have an implied union: ($\cup T$). It is clear from the Weibull model that there are non-trivial decreases of the hazard rate occurring over time for all of the subsystems, but especially the "Engine" subsystem, and that this is indicated for both rules- and tag-based methods.

			MTTF (days)		Weibull Params.	
Major System	method	query	K-M	Weib.	β	η
Bucket	rules-based	Bucket	9.00	10.8	0.83±0.03	17±0.9
	single-tag	[bucket]	15.0	17.1	0.83±0.03	27±2
	multi-tag	[bucket, tooth, lip, pin]	9.00	10.5	0.82±0.02	16±0.9
Hydraulic System	rules-based	Hydraulic System	8.00	9.07	0.86±0.02	14±0.6
	single-tag	[hyd]	25.0	24.1	0.89±0.04	36±3
	multi-tag	[hyd, hose, pump, compressor]	9.00	9.74	0.89±0.02	15±0.7
Engine	rules-based	Engine	9.00	10.8	0.81±0.02	17±1
	single-tag	[engine]	10.0	11.8	0.79±0.03	19±1
	multi-tag	[engine, filter, fan]	8.00	9.31	0.81±0.02	15±0.8

this type of MWO. Additionally, the censoring of failure observations (here, the scheduled replacement of a part, e.g., before it had actually failed) was approximated with the occurrence of a "changeout" tag, for which the previous caveat also applies, but to action-words instead of subsystem item-words. The comparison between these single-tag estimates for Median Time to Failure (MTTF) and the rules-based estimates are shown in Table 1.

To approximate a remedy to the above "subsystem problem", and thus to derive more holistic approximations of the subsystem MTTF—with minimal annotation effort from a human—we also include an attempt by an expert to determine a reasonable set of subsystem-related tags, along with the corresponding approximation of the subsystem MTTF (to correspond to the rules-based method). We allowed the use of several (less than 5) tags that are *strictly members of the relevant subsystem*. These multi-tag approximations of the subsystem are simply the union of the set of MWO occurrences of each individual tag. As seen in Table 1, along with the K-M model comparisons shown in Fig. 5, these multi-tag approximations perform remarkably well at reaching a similar MTTF estimate to the rules-based methodology.

5. CONCLUSIONS & FUTURE WORK

In this study, two approaches to structuring unstructured data in the form of maintenance work orders were reviewed and bench-marked through the calculation of basic survival analysis models. While single-tag estimates tended to underestimate the failure rate, from Table 1, the average discrepancy between single-tag and rules-based estimates for MTTF, across the three tested subsystems and two methods, was only 7.7 days, with the majority of that discrepancy coming from the bucket subsystem (average of 16 days); when an expert is able to use his prior system knowledge, as was done through the previously-discussed multi-tag sets, that average discrepancy goes down to less than a day.

It is important to note that the methods discussed here are mainly compared between themselves — there is a distinct lack of a "gold-standard" measurement for, e.g., calculating the "true" subsystem MTTF, because the actual MTTF *per-subsystem* was never recorded in the first place. While it may be possible, going forward, to obtain such a well-curated reference dataset, the lack of this information speaks more broadly to the state of data availability and overall lack of standardized methodology through this process. We believe that the results here do not particularly advocate for one method over another; the rules-based keywords display a high level of thoroughness, but are only as complete as the number of hand-made rules being created, while the data-driven tags have a tendency to miss both rare events, and "obvious" physics-based relationships that inherently get encoded into a set of hand-made rules. Rather, we advocate for a combination of approaches going forward. The lack of a "gold-standard" is not uncommon in the broader information retrieval community, where the weighted opinions of "experts" are often combined to approximate an agreed-upon gold standard result. (Hripcsak & Rothschild, 2005)

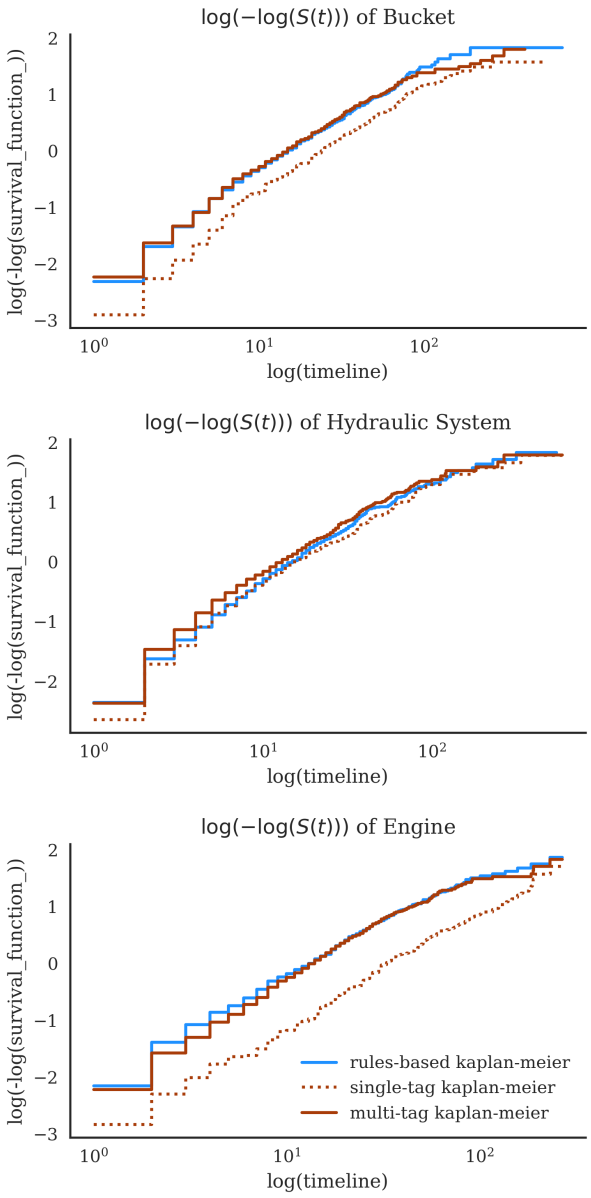

Figure 5. **Survival function comparison** — plotted on a log-scale, the multi-tag system approximation is clearly able to mirror the rules-based survival estimate across all relevant time-scales. Noticeable differences do occur for the lifespan extrema, though these effects are exaggerated in the plot and only last for small portions of the curve.

Given the at least one-week-difference in annotation labor time required between the two methods to achieve the reported results, the authors believe the tag-extraction method-

ology holds potential as an efficient tool for rapid MWO encoding. However, there are several key features of the rules-based approach that are lacking from the tag-based, and most important, perhaps, is the flexible definition of subsystem categorizations based on rule-matching. It would be very difficult to know, a priori, *which set of tags* and/or Boolean set operations would be "best" for approximating the classification of underlying subsystems.

Preferably, both rules-driven approaches, that encode some system-level from experts, *and* statistically sound empirical patterns from observation and data analytics, will continue to be explored as points of evidence toward a robust-yet-efficient standardized pipeline for encoding information from unstructured sources. Taking this further, we imagine a scheme where the development of taxonomies—or even ontologies—for expert systems are initialized and guided by latent patterns discovered from appropriate application of machine learning. Subsequent iterations of the machine learning pipelines for pattern discovery can then make use of human input via these "rule definitions", closing the loop that leads toward robust, hybridized, intelligence augmentation systems.

We suggest future efforts be directed toward the merging of automated tag extraction with the design of major functional relationships (encoded as rules), into an architecture for rapid, human-in-the-loop investigatory analysis. Such a system could take advantage of both the efficient data-processing from NLP techniques *and* the functional systems knowledge that human experts bring to the table.

ACKNOWLEDGEMENTS

This work was supported in part by the BHP Fellowship for Engineering for Remote Operations.

DISCLAIMER

The use of any products described in this paper does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that products are necessarily the best available for the purpose.

REFERENCES

Arp, R., Smith, B., & Spear, A. D. (2015). *Building ontologies with basic formal ontology*. MIT Press.

AS IEC 60300.3.14. (2005). *Dependability management application guide - maintenance and maintenance support*. SAI.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., ... others (2000). Gene ontology: tool for the unification of biology. *Nature genetics*, *25*(1), 25.

Bodenreider, O. (2004). The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, *32*(suppl_1), D267–D270.

Davidov, D., Tsur, O., & Rappoport, A. (2010). Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the fourteenth conference on computational natural language learning* (pp. 107–116).

Davidson-Pilon, C., Kalderstam, J., Kuhn, B., Fiore-Gartland, A., Moneda, L., Zivich, P., ... Rendeiro, A. F. (2018, April). *Camdavidsonpilon/lifelines: v0.14.1*.

Ebrahimipour, V., & Yacout, S. (2016). *Ontology modeling in physical asset integrity management*. Springer.

Fang, C., & Chang, E.-C. (2008). Information leakage in optimal anonymized and diversified data. In *International workshop on information hiding* (pp. 30–44).

Heinze, D. T., Morsch, M. L., & Holbrook, J. (2001). Mining free-text medical records. In *Proceedings of the amia symposium* (p. 254).

Ho, M. (2015). *A shared reliability database for mobile mining equipment* (Unpublished doctoral dissertation). University of Western Australia.

Hripcsak, G., & Rothschild, A. S. (2005). Agreement, the f-measure, and reliability in information retrieval. *Journal of the American Medical Informatics Association*, *12*(3), 296–298.

ISO. (2003). *Industrial automation systems and integration – integration of life-cycle data for process plants including oil and gas production facilities – part 2: Data model* (Tech. Rep.). Geneva, Switzerland.: International Standards Organisation.

Karray, M. H., Chebel-Morello, B., & Zerhouni, N. (2012). A formal ontology for industrial maintenance. *Applied Ontology*, *7*(3), 269–310.

Kelly, A. (1997). *Maintenance organization and systems*. Butterworth-Heinemann.

Leskovec, J., Rajaraman, A., & Ullman, J. D. (2014). *Mining of massive datasets*. Cambridge university press.

Márquez, A. C. (2007). *The maintenance management framework: models and methods for complex systems maintenance*. Springer Science & Business Media.

Meeker, W. (1998). *Statistical methods for reliability data*. John Wiley Sons.

Molina, R., Unsworth, K., Hodkiewicz, M., & Adriasola, E. (2013). Are managerial pressure, technological control and intrinsic motivation effective in improving data quality? *Reliability Engineering & System Safety*, *119*, 26–34.

Murphy, G. D. (2010). Testing a tri-partite contingent model of engineering cultures: A pilot study. *Reliability Engineering & System Safety*, *95*(10), 1040–1049.

Palmer, D. (1999). *Maintenance planning and scheduling handbook*. McGraw-Hill Professional Publishing.

Prasad, K. H., Faruquie, T. A., Joshi, S., Chaturvedi, S., Subramaniam, L. V., & Mohania, M. (2011). Data cleansing techniques for large enterprise datasets. In *Srii global conference (srii), 2011 annual* (pp. 135–144).

Rahm, E., & Do, H. H. (2000). Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, *23*(4), 3–13.

Rasoyska, I., Chebel-Morello, B., & Zerhouni, N. (2005). Process of s-maintenance: decision support system for maintenance intervention. *Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on*, *2*, 8 pp.–686.

Semy, S. K., Pulvermacher, M. K., & Obrst, L. J. (2004). Toward the Use of an Upper Ontology for U . S . Government and U . S . Military Domains : An Evaluation Military Domains : An Evaluation.

Sexton, T., Brundage, M. P., Hoffman, M., & Morris, K. C. (2017). Hybrid datafication of maintenance logs from ai-assisted human tags. In *Big data (big data), 2017 ieee international conference on* (pp. 1769–1777).

Sharp, M. E., Sexton, T. B., & Brundage, M. P. (2016). Semi-autonomous labeling of unstructured maintenance log data for diagnostic root cause analysis.

Sikorska, J., Hodkiewicz, M., D'Cruz, A., Astfalck, L., & Keating, A. (2016). A collaborative data library for testing prognostic models. In B. Lung & B. Zhang (Eds.), *European conference of the prognostics and health management society 2016.*

Spackman, K. A., Campbell, K. E., & Côté, R. A. (1997). Snomed rt: a reference terminology for health care. In *Proceedings of the amia annual fall symposium* (p. 640).

Tremblay, M. C., Berndt, D. J., Luther, S. L., Foulis, P. R., & French, D. D. (2009). Identifying fall-related injuries: Text mining the electronic medical record. *Information Technology and Management*, *10*(4), 253.

Unsworth, K., Adriasola, E., Johnston-Billings, A., Dmitrieva, A., & Hodkiewicz, M. (2011). Goal hierarchy: Improving asset data quality by improving motivation. *Reliability Engineering & System Safety*, *96*(11), 1474–1481.

Zhou, X., Han, H., Chankai, I., Prestrud, A., & Brooks, A. (2006). Approaches to text mining for clinical medical records. In *Proceedings of the 2006 acm symposium on applied computing* (pp. 235–239).