Hybrid Datafication of Maintenance Logs from AI-Assisted Human Tags

Thurston Sexton, Michael P. Brundage, Michael Hoffman, and KC Morris
Information Modeling & Testing Group
National Institute of Standards & Technology
Gaithersburg, MD, 20899, USA
Email: thurston.sexton@nist.gov

***Abstract*—One of the main challenges of applying AI to certain datasets derives from the datasets themselves being unstructured, unclear, and ambiguous. Furthermore, the insights that are to be gained reflect the quality of the data itself; if the data is skewed, so will be the insights. This problem is not unique to AI technology. People looking back at logs of past events often struggle to understand what was recorded, and to put together a timeline amongst a range of actors. AI technology can help humans sort the data out, but it does not provide the same insight often found in the background knowledge of human participants. This *contextual* weakness has made unstructured data hard to process. In our work, we have studied typical manufacturing maintenance logs to explore whether and how we can apply AI technologies to gain more insight from this—often vast and under-used—data-source. Our approach combines AI techniques for NLP, machine learning, and statistical processing with human contextual knowledge to quickly develop structured semantics reflecting unique datasets.**

***Keywords*-Natural Language Processing; Intelligent manufacturing systems; Predictive maintenance; Tagging;**

I. Introduction

The typical manufacturing maintenance log consists of composed notes from operators or maintenance technicians on problems encountered, along with some information about the solutions applied. Often, much of the information is left out, and the records contain numerous abbreviations and jargon. Some organizations are better than others about trying to classify these records to track maintenance history, but more often than not the problem details are still described in common language. Maintenance logs are typically only used when there is a recurring or "seismic" problem, at which point a full-fledged investigation is undertaken. Then, it is up to the investigator to try and make sense of what has been recorded. That person brings a good deal of background knowledge to this task, specifically a good understanding of the jargon in these records.

Our research has focused on developing a methodology for extracting useful *knowledge* from these records using emerging technologies for natural language processing (NLP) and statistical analyses to identify trends in the data. In this work we "datafy" the maintenance logs, making them more useful for statistical insight. We approach this problem through a hybridization—machine learning techniques, augmented with human guidance—to give meaningful structure to the data. Early results have been promising and are reported on here. First we provide an overview of the problem, followed by our approach and preliminary findings.

A. Why Datafication of Maintenance Logs?

Datafication has been defined as "the transformation of social action into online quantified data" [1]. In the example we present, we take the "social action" of a maintenance technician creating some historical record of events (for the eventuality that it becomes useful to another person), and transform it into data useful specifically for computation. In other words, it is the process of structuring data in a way that adds value to it, by facilitating the transmission of human contextual knowledge used in understanding this data.

Maintenance management in a manufacturing facility is an important part of the manufacturing process, since reducing machine downtime leads to increased productivity. When a machine breaks down and the subsequent maintenance work-order is issued, the associated information is either manually written down, input into a database, or created using maintenance management software. Meanwhile, during the machine downtime, the time spent diagnosing the issue is often larger than the time spent carrying out the repair of the machine [2]. In other words, getting at the knowledge within this work-order data, and using it to diagnose and address problems faster, will mean real improvements to a manufacturing process.

The problems encountered in using this data reflect the human-centric nature of the maintenance activity, and include:

1) Technicians often describe problems informally, leading to inconsistencies and inaccuracies in the data.
2) Certain maintenance data, such as the actual root cause of a problem, is not always being collected.
3) Once data is collected, it is *not* often subsequently used for future diagnosis.

The first two of these factors can be attributed to a reliance on human background or contextual knowledge. The latter factor is due to a lack of clear and simple procedures for doing so—also a human interaction issue. Our research addresses both aspects of the problem.

The framework presented in [3] shows how these factors can be systematically addressed. In this paper, we explore

Table I
EXAMPLE OF TYPICAL INPUT FIELDS IN WORK-ORDER DATA

Input field	Input type
Issue ID	Integer
When work-order was issued	Time/Date
Description of Problem	Raw Text
Asset ID	Protected String
⋮	⋮
Resolution of Problem	Raw Text
When work-order was completed	Time/Date

how to apply AI techniques to structure the data in a way that will be more easily understandable and processed—this is the datafication of these maintenance logs. The resulting data set contains maintenance work orders that have been consistently tagged for clarity and analysis.

B. The Structure of Maintenance Logs

To understand the problem of datafication for manufacturing maintenance logs, let's consider the data more closely. Maintenance logs, especially when we consider the work-orders themselves, contain basic natural language descriptions of the maintenance issues. They're often organized into tabular form, and include some kind of contextual data, as illustrated in Table I.

In other cases, data comes directly from sensors, organized according to some format dictated by the device. While some standards are emerging, at present the devices can vary greatly and the sensor data is essentially a structured set of jargon. In practice, sensor data can be used to facilitate the creation of work orders by monitoring a variety of physical signals created by a machine. Work orders can sometimes be generated automatically if certain sensor measurements indicate the machine is not functioning as expected. These work orders can be used in the same manner as those that are created manually and contain similar data.

For the purposes of this paper, we focus on the human-generated, natural-language maintenance data—specifically, on "Description of Problem" and "Resolution of Problem" fields commonly found as in Table I. These fields are often a free text description. For example, Technician 1 might describe a "hydraulic leak" as "leaking hydraulic fluid at Machine H1", while Technician 2 might write "hyd leak at cutoff unit of H1." Both of these descriptions represent the same overall problem of "hydraulic leak" located at "Machine H1", but described in two different ways. The same applies for the eventual problem resolution text. These inconsistencies makes it impossible to automatically search a database to find something as simple as "common resolutions for all cases of hydraulic leak".

In an ideal scenario, a maintenance technician would be able to search instances of "hydraulic leak" and find suggested solutions from previous instances. Locating these logs would support the maintenance technician as an augmentation to his existing experience, hopefully aiding to quickly solve the problem. Additionally, this knowledge could give maintenance managers a more accurate way to track problems and corresponding repairs throughout the facility.

The common assumption behind recording these logs is that a *human* will refer back to them; as such a human is assumed to interpret them. That assumption, however, makes it very difficult for machine computation in any meaningful way. Many manufacturers have recognized this problem, and try to enforce more rigor in these descriptions of problems and solutions, perhaps using interfaces like drop-down menus to control vocabulary. Computerized Maintenance Management System (CMMS) solutions, such as IBM Maximo[1], address some of these issues through drop down menus and enforced categorization [4]. Many other commercial CMMS products, such as Maintenance Connection or eMaint CMMS, offer similar approaches to maintenance management. However, in our conversations, the problems are not completely resolved through these systems; the controlled vocabularies do not fully cover the situational complexity, and much of the useful data for diagnostics is still covered in the comments. The interfaces to these systems are often cumbersome to use and do not commonly succeed in altering technician recording behavior or eventual data structuring. The analysis that can be performed by these CMMS platforms—while certainly powerful—is often limited by the inconsistent structure of the human-generated data. Furthermore, small and medium enterprises (the bulk of the US manufacturing base) often do not have the wherewithal for these solutions.

As a workable solution to these problems, we propose to take the free text descriptions found in these logs and apply the concept of *tagging* to provide context for such data. This allows maintenance managers and technicians to properly analyze the data—perhaps even as a preparation mechanism for available CMMS systems—and thereby obtain diagnostic and prognostic aid from a trove of previous solutions.

II. HYBRID DATAFICATION OF MAINTENANCE LOGS

Underlying the task of datafication is a fundamental need to *classify* the content of available unstructured data, which is to say, provide structure for it. Our challenge is to classify the data from the manufacturing maintenance logs into the diagnostic framework shown in Fig. 1. Each word or token in a maintenance log will be fit into one of these categories. Using this structure we can then identify meaningful patterns in the data. Classification in this context is approached,

[1]Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

generally, in one of two ways: manually, given some framework within which users are allowed to operate and make classification decisions; and via automation, generally using Natural Language Processing (NLP).

Due to the nature of manufacturing maintenance as a natural-language domain, neither of the above are particularly satisfactory approaches. In our experience, the scale of available natural-text maintenance log data exists in the $10^3 - 10^5$ (number of entries) range—this is far too labor-intensive to manually classify under some standardized framework, especially considering the need for time-crunched, domain-expert employees to divert time toward such a menial task. On the other hand, NLP is typically suited toward large corpuses of documents, not the short, technical issue descriptions found in maintenance logs. [2] Complicating the problem further, these logs are full of highly domain-specific technical terms and user-specific lingo. The classification is very difficult to automate without customized NLP libraries tailored to the jargon, further adding to the burden on manufacturers trying to implement datafication techniques.

Our approach is to *hybridize* the two paradigms, centering around the use of data *tagging*. "Tags" are simply annotation terms for some resource — in this case, maintenance work-orders — and are generally chosen without a controlled vocabulary or boundary [5]. Tagging is very flexible — being a direct reflection of the tribal knowledge and domain vocabulary of specific users, it can adapt quickly to new domains by simply adding and removing tags. Additionally, tags can be used to reconstruct robust dynamic hierarchies of concepts, sometimes referred to as "folksonomies" [6].

Rather than separating data-points into discrete bins or strict relations, a user of a tagging system simply assigns characteristics directly onto data instances . However, while being more intuitive for humans than strictly unambiguous taxonomic classification, this "fully-manual tagging" methodology is still labor intensive. For this reason we propose a hybrid system, augmenting a human's ability to tag natural-language maintenance documents, while preserving the flexibility and robustness of a folksonomy. This is accomplished through easily implemented, foundational techniques from NLP as a way to optimize a human tagger's time investment. In this way, we hope to encourage adoption and understanding of techniques that balance human usability with analytic efficacy for the analysis of largely unused maintenance data.

A set of tags meant to give structure to maintenance logs should in essence list qualities associated with the 1) problem, 2) solution, and 3) items (the objects directly relevant to the issue such as machine, resources, parts, etc.) Consequently, we have adopted a "meta-classification" for the tags themselves, namely, the enumerated classes above. Fig. 1 shows an example tagging in this style, which is accomplished via the process outlined in Section II-A below and in Fig. 2.

A. Importance-based Vocabulary Tagging

It follows from the above discussion that imposing a protected vocabulary when tagging items would be counter-productive; however, a vocabulary list functioning as a look-up table, able to match important concepts to their "tag" representation (without repeated human input), is very well-suited to technical domains like manufacturing maintenance. NLP allows us to extract an initial vocabulary list which can then be validated and refined by experts. NLP extracts a list of "tokens" [3] that are ranked in order of importance. For the purpose of this study, we turn to a widely used importance metric called Term Frequency & Inverse Document Frequency (TF-IDF), which balances the frequency of occurrence of some word within a document (i.e. issue) and the frequency in the overall corpus of documents. [7]. Passing this list of tokens to a set of domain experts for "meta-classification" is highly efficient for quickly building a list of common vocabulary, as we demonstrate in the following case-study. The job of the experts is to determine

[2]NLP has been applied to short documents with many instances (i.e. Twitter®) with general success; however, the size of our datasets are not nearly of that scale, and the language is generally much more technical.

[3]In the context of NLP, tokens are either individual words, or combinations of words (n-grams) that have a discrete meaning. For example, "ice", "cream", and "ice cream" might all be tokens, with "ice cream" being a 2-gram.

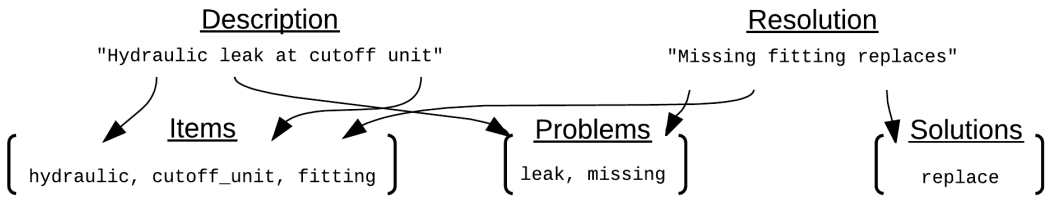

Figure 1. Illustration of a maintenance issue tagging process, mapping from two raw-text inputs—here an issue description and corresponding resolution—to a set of categorized tags. Note the correction of "replaces" to "replace".

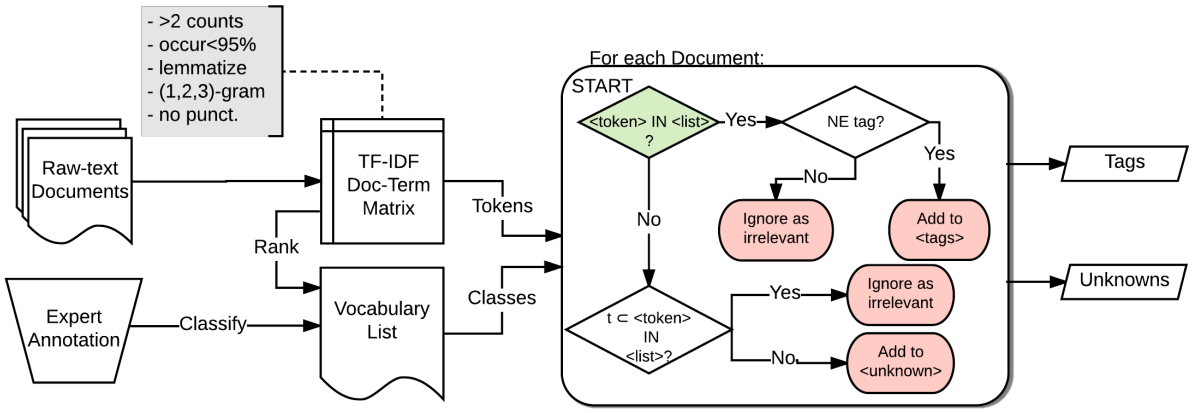

Figure 2. A flow chart illustrating the process from raw-text work-order documents and expert annotation to tags. The key is to pass users a ranked list that is already in (approximately) "tag-like" form, via various pre-processing techniques and ordered by a TF-IDF "importance" heuristic. In the keyword-extraction step, a series of binary decisions determines if incoming tokens are already classified and whether to mark them as **Tags** or **Unknowns**.

if the tokens fall into one or a subset of the following categories:

1) **Item**, **Problem**, or **Solution** named-entity (NE) tag
2) **Redundant** or un-useful tokens
3) **Stop-word** and **Ambiguous** tokens.

The NE tag classifications (1) are then treated as definitive information, and all instances of these tokens need no further human input to be correctly tagged. We allow the experts to add an alias or *preferred label* for any of these token classes, thereby automating the creation of a thesaurus that can take local jargon and abbreviations into account.

The redundant tokens (2) do not add additional information, and can be ignored. As an example, consider the recognized tokens from the phrase "bar feeder chain". Extracted tokens from 3-gram TF-IDF might be:

{bar, feeder, chain,
bar feeder, feeder chain,
bar feeder chain}

However, in this specific process, a "feeder chain" might not make sense without "bar", and a feeder might not be an object existing outside of "bar feeder". Such a scheme is depicted in Fig. 3. With necessary redundant classifications in place, the resulting tags extracted using expert-classified vocabulary might be

{bar, bar feeder, bar feeder chain}

Similarly, stop words or ambiguous tokens (3) are deemed either non-important filler words or words too ambiguous to give a strict Item/Problem/Solution designation. This process of annotation results in a classification and alias list in order of TF-IDF "importance", as seen in Table II. It is then computationally inexpensive to return to the the Bag of Words representation of the original raw text issue descriptions and

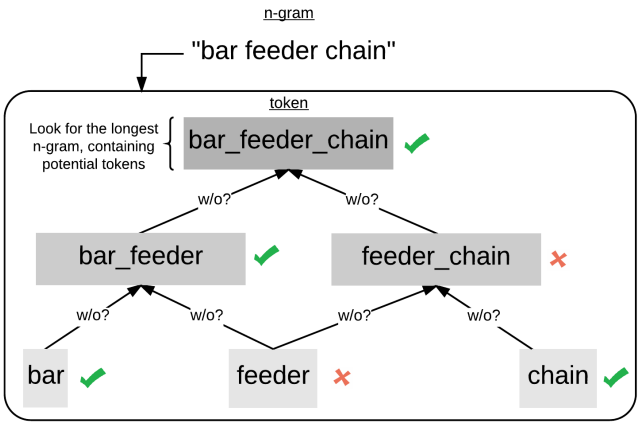

Figure 3. An example heuristic rule for determining what extracted tokens might be considered "redundant". Each arrow is a determination if, within the context of a given corpus, some n-gram token makes sense to talk about *without* its (n+1)-gram parent. If not, it can be ignored, since it will almost always occur within one of the sensible n-gram parents.

Table II
EX. TOP TOKENS, CLASSIFICATIONS, AND ALIASES

Token	NE	Alias
replace	S	replace
unit	I	unit
motor	I	motor
spindle	I	spindle
leak	P	leak
valve	I	valve
replaced	S	replace
⋮	⋮	⋮

flag extracted tokens as useful tags, redundant/ambiguous, or as completely unknown due to incompleteness of the vocabulary list. This automated tagging approach is described in the case study section below using an industry data-set.

B. Human Critique and Completion

Once a first pass has been completed by the automated vocabulary-based tagger, humans may now provide critique of the set of tags for each issue. This can be handled with a combination of approaches.

As a consequence of the vocabulary method above, each maintenance issue will now have a list of **Unknown** tokens. It is then possible to perform a TF-IDF importance ranking *on only the unknown tokens*, then passing this to a human expert for classification. This iterative process is hypothesized to quickly reduce the number of unknown tokens while better completing the vocabulary list than the TF-IDF heuristic ranking alone.

Another possibility, which particularly becomes useful when multiple human agents can tag resources in a data-set, takes a nod from previous taxonomic and crowdsourcing work [8], [9]. If users are to ever manually add or remove tags from individual issues directly, it is important that they are directed in such a way as to

1) maximize the utility of their time investment, and
2) optimize completeness of the issue-to-tag mapping

This can be achieved by maximizing the *expected information gain* over the set of available issues, and queuing issues for the users to tag, ordered in this way.

The human critique may serve another, potentially important, goal: the separation of Items into problem-items and solution-items. These categories are highly ambiguous and the vocabulary list may not be useful to differentiate between them, since the tags for the two categories are nearly mutually-inclusive. Consequently, these must be determined on a per-issue level. The benefits for doing so are a current area of research, especially concerning diagnostic prediction using the tagged data; a minimal example of such diagnostics is further discussed in the *Diagnostics* part of Section III-B below.

III. Industry Case Study

To test the vocabulary-based tagging procedure outlined above, a manufacturing data-set containing 3,438 raw-text descriptions and resolutions of maintenance issue work-orders was analyzed.

First, two domain experts collaboratively tagged as many individual issues as possible, resulting in 1,814 structured data-points via fully-manual tagging. The experts reported being able to tag approximately 100 issues/hour, giving a rough estimate of 18 hrs, 8min to tag that number of issues. They were asked to list the Problem, Solution, and Item tags, further differentiating between problem-items and solution-items.

Next, they were given a TF-IDF ordered list of extracted tokens, with the task of meta-classification described previously, assigning one of: Item (**I**), Problem (**P**), Solution (**S**), Ambiguous (**U**), or Stop-word (**X**), along with any preferred labels to replace lingo, misspellings, etc. The experts reported finishing 1000 vocabulary items in approximately 1 hour, while ultimately completing 1,362 classifications, giving an estimate of 1hr, 22min spent.

A. Datafication Quality

To estimate the quality of the automated tagging, we assume that any issues with *no* Unknown tags are considered fully "datafied", or tag-complete. Additionally, for issues where no known tags were extracted at all, the datafication process was completely non-useful. The top of Figure 4 reports both of these groups as fractions of the entire data-set, versus the size of the available vocabulary list. The automated tagger datafies nearly all issues, and extracts all available information from over half of the data-set with only 1,000 tokens in its vocabulary.

1) Comparison to Fully-Manual Tagging: Another way to quantify the tag quality is by directly comparing the tag-set from the vocabulary tagger to the human-generated fully-manual tags as a test-set. This requires a suitable metric. Typically, accuracy is defined as the ratio of "correct" to "incorrect" label instances, which in this context would be how many times one set of tags perfectly matches the other set. This is an overly-harsh metric for the performance of multi-label classification, since it ignores *how close* we got to the correct output in each case. The Hamming score (S_H) measures the "distance" between the two sets of tags, averaged over the data-set, which is a more forgiving and useful metric [10]. Given a set of n resources, where the ith resource has "true" tags T_i and predicted tags P_i:

$$S_H = \frac{1}{n}\sum_{i=1}^{n}\frac{|T_i \cap P_i|}{|T_i \cup P_i|} \tag{1}$$

However, this is somewhat difficult to interpret. Additionally, we can use the more intuitive notions of *precision*, *recall*, and their weighted harmonic mean F_β*-score*. [4] Simplistically, precision is a classifier's ability to avoid false-positives, and recall is its ability to *not miss* true-positives. F_β then combines them, attributing β times more importance to recall than to precision:

$$Pr = \frac{1}{n}\sum_{i=1}^{n}\frac{|T_i \cap P_i|}{|P_i|} \tag{2}$$

$$Re = \frac{1}{n}\sum_{i=1}^{n}\frac{|T_i \cap P_i|}{|T_i|} \tag{3}$$

$$F_\beta = \frac{1}{n}\sum_{i=1}^{n}(1+\beta^2)\frac{Pr_i Re_i}{\beta^2 Pr_i + Re_i} \tag{4}$$

Since we do not really trust that the original tags given by humans were all-inclusive, we want to place more importance on recall in our measure — for example, twice as much

[4]There are several ways to aggregate these scores across a data-set; here we have adopted the simple mean over all resources/issues (macro-average).

— leading to our usage of F_2. The results of calculating these values, assuming the manually-tagged issues define the "true" tag set, are in the bottom of Fig. 4.

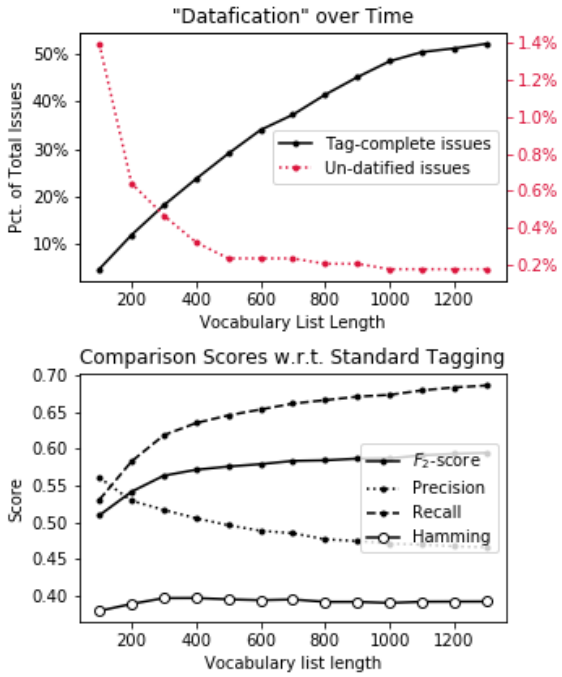

Figure 4. The top chart illustrates the return on labor investment for meta-classification of the tag vocabulary. The bottom chart shows the information-retrieval scores for the vocabulary-tagger when calculated against un-assisted human-derived tags.

It is interesting to note that the Precision actually goes down with more vocabulary; this arises from Precision-punishing false-positives. This implicitly assumes that the human-derived tags are "the whole truth", when in reality these tags only consist of what was deemed important. This, coupled with the way that Precision losses and Hamming scores level out rather quickly with further increases in vocabulary size, indicate that it is likely better to start with the auto-tagging procedure as a "gold standard", since we can be certain extracted tokens *existed* in the original text.

2) NLP Comparative Study: Another common way to assist datafication efforts with NLP is by using Machine Learning models that, for example, mimic the tagging patterns of observed human-generated tags. Generally this requires a training set, i.e. there are resources with existing "true" tags, which are used to train a multi-label classifier. [5] This obviously requires some initial time-investment from human taggers to create this training set, but once this is completed, the model has potential to quickly tag remaining

[5]Mutli-label classification is the task of assigning some set of target labels to an input, meaning that the output "classes" may not be mutually exclusive (as would be the case in multiclass classification). This is necessary for tag prediction, since resources must receive multiple tags.

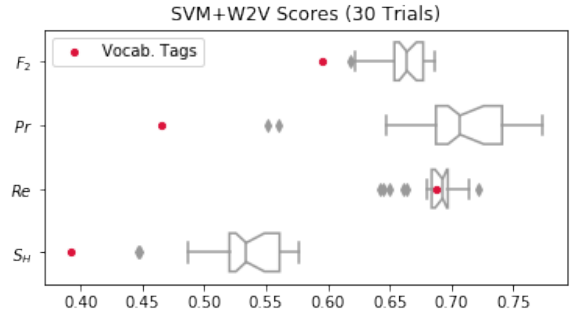

Figure 5. Box-plot comparison of the performance for Vocabulary tagger vs. multi-label SVM classification using Word2Vec semantic embeddings of the raw-text issues. Notches indicate 95% confidence intervals for median score, via 10,000 bootstrap samples.

resources.

For this study, a linear-kernel support-vector machine (SVM) was used in a One-vs-Rest multi-label scheme[6]. Inputs were 1,814 Word2Vec semantic embeddings [12], one for each raw-text maintenance issue tagged by the experts. Target (training) labels were the expert-generated tags themselves. Stochastic Gradient Descent (SGD) was used to minimize the hinge loss, and an elastic-net penalty was used to reduce over-fitting [13]. The model was written in Python using the Sci-kit Learn and SpaCy libraries [14], [15]. Results compared to the vocabulary-based tagger w.r.t. the expert tags are displayed in Fig. 5.

It is apparent that the SVM out-performs the vocabulary tagger in every available metric, especially in Precision (i.e. a much lower false-positive rate). Recall, however, that we do not expect the expert taggers to generate all-inclusive labels, which prompted the usage of an F_2 metric. The F_2-score for both methods are remarkably close. Considering the drastic (over an order of magnitude) human time-investment discrepancy between implementing the two methods, the 7-point gain for the SVM is not a particularly worth-while return on investment.

Still, the increased precision is a desirable feature of the SVM. Once the initial vocabulary-based tags have been through the Human-Critique phases described above, an SVM using Word2Vec would potentially pick up on contextual tag patterns not existing explicitly in the raw text. This layer of human-machine hybridization should provide a robust and scalable approach to the continued tagging of issues, especially when the corpus size becomes considerably larger.

B. Potential Applications of Datafication by Tags

In this section, we provide an initial foray into the usage of tagged maintenance data for 1) Diagnostic assistance, and 2) Prognostics and health management (PHM). While there

[6]Called Binary Relevance method in the multi-label classification literature [11]

is not space here for a holistic tag-based approach to either field (nor would any single study necessarily suffice), we hope to demonstrate the effectiveness of tagged maintenance issues at capturing some useful information from otherwise unused natural-language data.

1) Diagnostics: The most straight-forward application of tags is to query on the occurrence rate for certain tag combinations, as proxy for occurrence rate of certain issues. This is especially effective for its simplicity. Using the case-study data, the most common raw-text description was an "Accumulator check request", occurring uniquely 14 times. After automated vocabulary tagging, the issue count for occurrence of $\{\text{accumulator}\} \cup \{\text{check request}\}$ was 73. Additionally, the rate for $\{\text{hydraulic}\} \cup \{\text{leak}\}$, which was not in the top 20 most common (raw-text) issue descriptions, in fact occurred in 159 separate issues. The high occurrence of hydraulic leaks was not known to the maintenance management previously.

Extending this idea, tag occurrence can be shown over time, visually illustrating issue "hot-spots", as demonstrated in the top of Fig. 6. Note the increase of hydraulic leaks in the summer months, information which could potentially be correlated with specific jobs underway in those periods, that might be causing such behavior.

Another possible use-case concerns solving new maintenance work-orders. Given the set of existing tags, along with some new problem and item tags as input from a maintenance tech, an algorithm might predict the set of most likely solution tags, along with other item tags of significance to this issue. If the additional step of classifying Items as "problem-items" and "solution-items" is taken as part of the Human-Critique phase, it is possible to predict the object involved in solving some input problem. This might be done with a type of "maintenance-tag language", using the classified tags as an n-gram language model, such as in [16]. An visual implementation of this style of diagnostic assistance is demonstrated in Fig. 7.

2) Preliminary Prognostics: Prognostics is primarily concerned with predicting the useful life of some system or component before it ceases to function [18]. This remaining useful life (RUL) is often calculated using empirical observations and/or a model of the process failure rates. The expected time between actual failures is called the failure inter-arrival time (δ_f), and it is generally modeled as a stochastic process from which failures are drawn. [19]

Note that failures are not—in general—the same as maintenance requests, since a machine able to be repaired has not actually failed or reached 0 RUL. With this in mind, even if we make assumptions that failures are a constant-rate stochastic process, it is often quite difficult to calculate RUL values, let alone when there is a lack of sensor or other pre-existing data in a CMMS system. However, with the tagging system proposed above, it is possible to approximate failures as occurring when some set of pre-defined tags appear, such as $\{\text{broken}\} \cup \{\text{replace}\}$. A calendar view of the tag co-occurrences over time, with this explicit condition, is shown in the bottom of Fig. 6. Using that data, is now possible to approximate δ_f as drawn from, for example, an exponential distribution $\delta_f \sim \text{Exp}(\lambda)$, where λ is the rate parameter in a constant-rate Poisson Process. The approximation, while certainly rough, can be visually verified as reasonable for the "spindle" and "motor" tags in Fig. 8.

As a final item of interest, it is perhaps useful to consider a *taxonomy* of maintenance concepts. Determining which tags are most discriminatory for use in prognostics is a non-trivial problem, and a taxonomy might help to organize the tags as inheriting information or even classification from other, more generalized tags. However, generating such a taxonomy or ontology is a labor-intensive design task, and could not easily be generalized across multiple manufacturing domains. One possibility is the use of automated taxonomy-generation algorithms built around the sorting of tag co-occurrences according to their graph-theoretic centrality. One such tool, the Heymann algorithm [20], has shown preliminary success at hierarchically ordering the item tags by generality. This will be discussed at length in further research, but initial results suggest that this automatically generated taxonomy is easy for humans to critique for robustness This may prove a much simpler task than designing a process taxonomy from scratch.

IV. Conclusion

The work outlined in this paper is part of a larger framework presented in [3]. The framework is the basis for a knowledge base of manufacturing performance problems, to support diagnosis of performance issues. The use of manufacturing work order data is one method for identifying these problems, and others are being researched as well. The tagging procedures described in this research promise to provide a rich source of data for the knowledge base.

The progress shown thus-far in datafication of maintenance logs addresses one of the primary challenges of constructing such a knowledge base—formulating reusable semantics around data being collected by manufacturers today. By most estimates, a significant portion of today's manufacturing data is not being used—or perhaps only analyzed later, when significant problems occur. A barrier to using the data more proactively is the use of inconsistent terminology and lack of context for the data. Our tagging approach can help address these problems.

In addition, the insights gained through the datafication and analysis processes themselves suggest that meaningful insights may be gained from the data independent from its inclusion in a knowledge base. The goal for this work is to provide methods to improve manufacturing performance and help to address common challenges that industries face. We envision publishing a formal methodology and tool that manufacturers could use to learn from their own data sets.

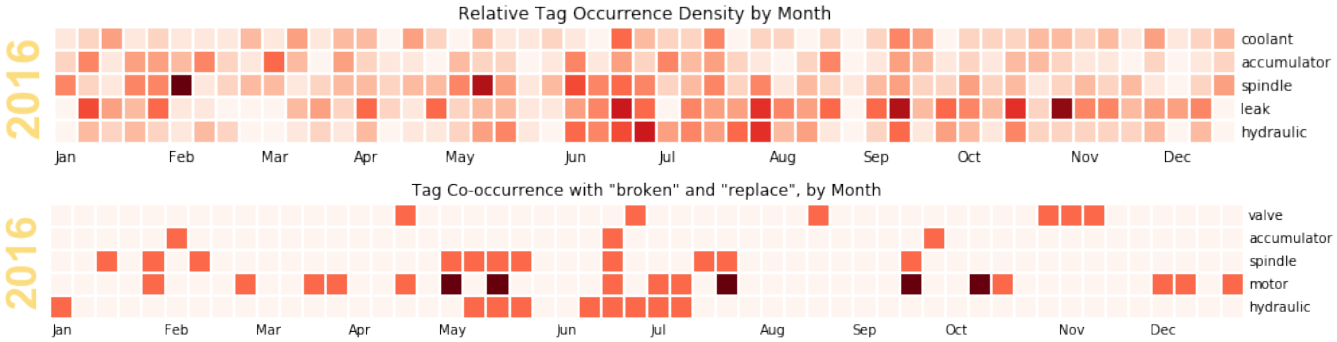

Figure 6. Illustration of the usefulness of tags in quickly analyzing maintenance data as a time-series. Additionally, an approximation to "failure rates" of parts might be approximated by returning only tags co-occurring with both the "broken" and "replace" tags. This rough approximation may be useful for preliminary RUL models.

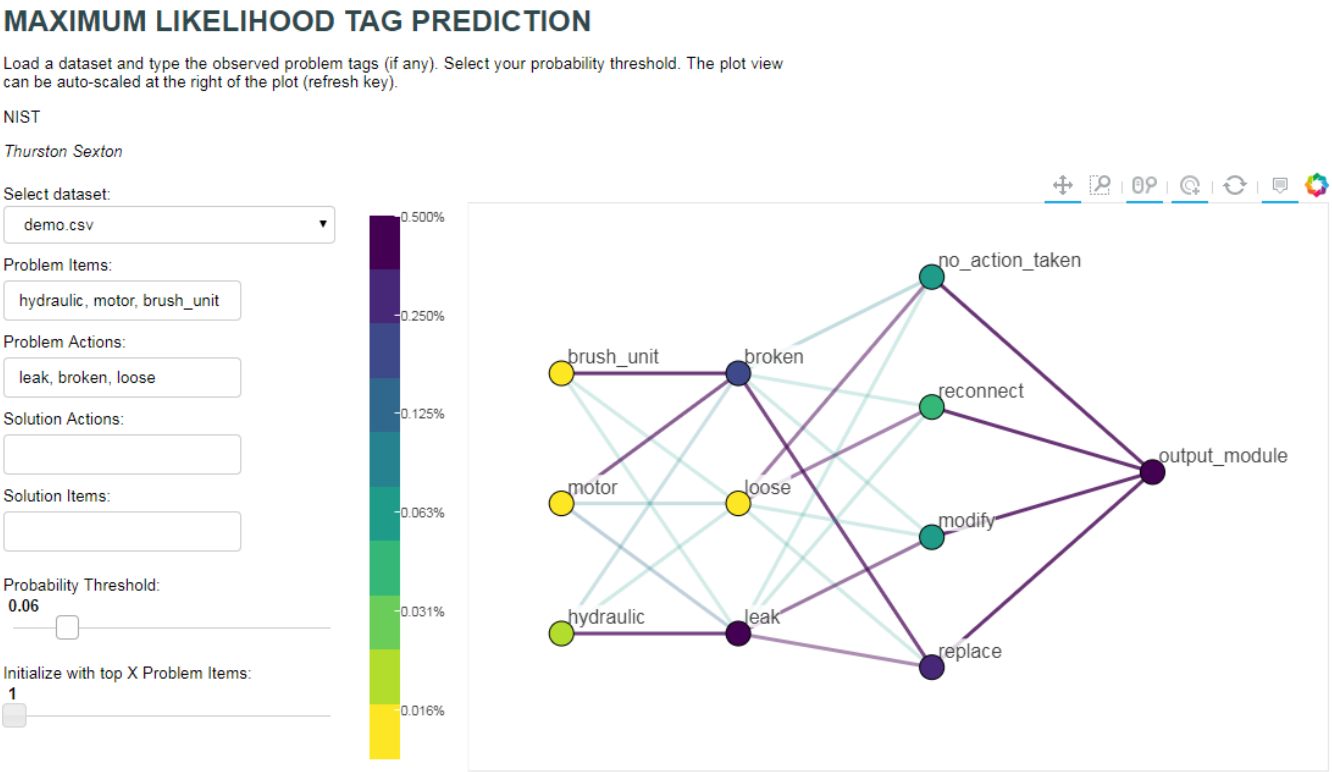

Figure 7. A simple app interface demonstrating the usefulness of further categorizing Items into "problem items" and "solution items". Simply using ordered tag co-occurrences previously observed in the database, one might calculate the unsmoothed likelihood of observing some solution tags given some input problem tags, as illustrated above. Each node is weighted by its likelihood given the nodes before it (left-to-right), and the edges indicate how much each node is contributing to the likelihood of its successors. In this example, given an issue like "Hydraulic leak; motor is broken with a loose brush unit", the suggested solution is to replace the output module. Note that only nodes above some tunable probability threshold are displayed. Created using interactive widgets via the Bokeh visualization library [17].

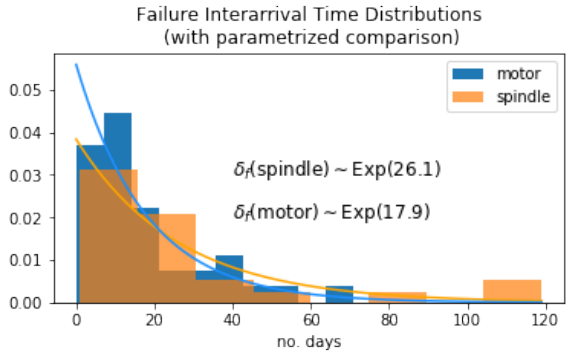

Figure 8. Approximate failure inter-arrival time (δ_f) distributions extracted using vocabulary-based tags. Basic models for spindle or motor failure could perhaps assume a constant-rate Poisson Process, allowing one to predict failures, e.g., for *a* motor once every 18 days, on average.

Future work will also include reproducing our results with data from a broader variety of organizations and sources, as well as understanding the idiosyncrasies of computation with sensor data. Finally, we plan to explore additional interface technologies that help humans in validating and augmenting data sets produced through automated means of processing. These mechanisms, along with the previously discussed human-critique methodologies, are areas for future research.

References

[1] E. Dumbill, "A revolution that will transform how we live, work, and think: An interview with the authors of big data," *Big data*, vol. 1, no. 2, pp. 73–77, 2013.

[2] R. L. Kegg, "One-line machine and process diagnostics," *CIRP Annals - Manufacturing Technology*, vol. 33, no. 2, pp. 469 – 473, 1984. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0007850616301688

[3] M. P. Brundage, B. Kulvantunyou, T. Ademujimi, and B. Rakshith, "Smart manufacturing through a framework for a knowledge-based diagnosis system," in *Proceedings of the ASME 2017 International Manufacturing Science and Engineering Conference, MSEC2017*. American Society of Mechanical Engineers, 2017.

[4] K. Doyle, J. Saulman, and B. Cary, "IBM solution approach for enterprise asset management," IBM Software Group, Tech. Rep., 1 2014, document Version 1.2.

[5] M. Strohmaier, C. Körner, and R. Kern, "Understanding why users tag: A survey of tagging motivation literature and results from an empirical study," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 17, pp. 1–11, 2012.

[6] T. Vander Wal, "Folksonomy," 2007.

[7] A. Rajaraman and J. D. Ullman, *Mining of Massive Datasets*. Cambridge University Press, 2011.

[8] X. Chen, P. N. Bennett, K. Collins-Thompson, and E. Horvitz, "Pairwise ranking aggregation in a crowdsourced setting," in *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM, 2013, pp. 193–202.

[9] J. Bragg, D. S. Weld *et al.*, "Crowdsourcing multi-label classification for taxonomy creation," in *First AAAI conference on human computation and crowdsourcing*, 2013.

[10] S. Godbole and S. Sarawagi, "Discriminative methods for multi-labeled classification," *Advances in knowledge discovery and data mining*, pp. 22–30, 2004.

[11] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.

[12] T. Mikolov, K. Chen, G. Corrado, J. Dean, L. Sutskever, and G. Zweig, "word2vec," 2014.

[13] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[15] M. Honnibal and M. Johnson, "An improved non-monotonic transition system for dependency parsing," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, September 2015, pp. 1373–1378. [Online]. Available: https://aclweb.org/anthology/D/D15/D15-1162

[16] Y. Lv and C. Zhai, "Positional language models for information retrieval," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2009, pp. 299–306.

[17] Bokeh Development Team, *Bokeh: Python library for interactive visualization*, 2014. [Online]. Available: http://www.bokeh.pydata.org

[18] G. J. Vachtsevanos, F. Lewis, A. Hess, and B. Wu, *Intelligent fault diagnosis and prognosis for engineering systems*. Wiley Online Library, 2006.

[19] Y. Xie, K. Tsui, M. Xie, and T. Goh, "Monitoring time-between-events for health management," in *Prognostics and Health Management Conference, 2010. PHM'10.* IEEE, 2010, pp. 1–8.

[20] P. Heymann and H. Garcia-Molina, "Collaborative creation of communal hierarchical taxonomies in social tagging systems," Stanford, Tech. Rep., 2006.