HAL open science

Toward Semi-autonomous Information

Michael Sharp, Thurston Sexton, Michael P. Brundage

To cite this version:

Michael Sharp, Thurston Sexton, Michael P. Brundage. Toward Semi-autonomous Information. IFIP International Conference on Advances in Production Management Systems (APMS), Sep 2017, Hamburg, Germany. pp.425-432, 10.1007/978-3-319-66923-6_50 . hal-01666222

HAL Id: hal-01666222

https://inria.hal.science/hal-01666222

Submitted on 18 Dec 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Toward Semi-Autonomous Information Extraction for Unstructured Maintenance Data in Root Cause Analysis

Michael Sharp, Thurston Sexton, and Michael Brundage

National Institute of Standards and Technology
Systems Integration Division,
100 Bureau Dr, Gaithersburg, MD 20899, USA
{michael.sharp,thurston.sexton,michael.brundage}@nist.gov
https://www.nist.gov/el/systems-integration-division-73400

Abstract. To facilitate root cause analysis in the manufacturing industry, maintenance technicians often fill out "maintenance tickets" to track issues and corresponding corrective actions. A database of these maintenance-logs can provide problem descriptions, causes, and treatments for the facility at large. However, when similar issues occur, different technicians rarely describe the same problem in an identical manner. This leads to description inconsistencies within the database, which makes it difficult to categorize issues or learn from similar cause-effect relationships. If such relationships could be identified, there is the potential to discover more insight into system performance. One way to address this opportunity is via the application of natural language processing (NLP) techniques to tag similar ticket descriptions, allowing for more formalized statistical learning of patterns in the maintenance data as a special type of short-text data. This paper showcases a proof-of-concept pipeline for merging multiple machine learning (ML) and NLP techniques to cluster and tag maintenance data, as part of a broader research thrust to extract insight from largely unstructured natural-language maintenance logs. The accuracy of the proposed method is tested on real data from a small manufacturer.

1 Introduction

Multiple industries often use root cause analysis (RCA) techniques to diagnose the underlying cause(s) of problems (Shrouti et al., 2013). Within the manufacturing industry, there are a variety of RCA techniques that are utilized: Six Sigma, including DMAIC (Define, Measure, Analyze, Improve, Control) and DFSS (Design-For-Six-Sigma) (BOUTI and KADI, 1994), Failure Mode and Effect Analysis (FMEA) (Liu et al., 2013), and fishbone diagrams, also known as Ishikawa diagrams (Juran and Godfrey, 1999) are just a few. While there are many techniques, instances of RCA are often problem-specific studies, where results are not readily available for wide retrieval in future studies. A framework was developed previously in Brundage et al. (2017) to help alleviate this issue,

providing mechanisms for accessing previous problems to aid in diagnosing the root cause. However, the developed framework relies on readily structured descriptions of causes-effects-treatments; generally such patterns are derived from raw information tracked via a Computerized Maintenance Management System (CMMS). Such clearly structured information is rarely found in practice, as technicians often inconsistently record informal prose rather than clearly filling in discrete fields for causes, effects, and treatments. Such inconsistencies make it difficult to perform diagnosis procedures. This paper begins to address that issue by providing NLP and ML techniques to prepare, clean, and tag the data for use in the diagnosis framework. It is aimed at cases where a CMMS may not be properly implemented, or in a well managed CMMS to help capture in-explicit information from any free form descriptions or comments within the system.

2 Motivation

Using NLP techniques in a maintenance data-set, unlike the more popular applications of NLP that have huge amounts of casual and/or complete-sentence phrases (such as Yelp® reviews or a Twitter® feed), requires treatment of documents generally smaller in nature, which at times have entries with fragmented sentence structure or are written in domain-specific shorthand. In addition, to the authors' knowledge, no definitive corpus or thesaurus of maintenance log terms and terminology currently exists that spans all companies in an industry setting. Nor, in many cases, would one be appropriate, as each company — or even each work site — will often develop their own short hand vocabulary and "tribal knowledge" set that could be meaningless to anyone not immersed in that environment. This work seeks to provide a method for working within these environments in order to characterize and categorize often dissimilar entries.

The authors envision a mature information extraction tool as autonomously tagging and structuring extracted information from the short, often fragmented entries that are common characteristics of many industrial maintenance logs. Achieving such vision would require both additional data sets, as well as comparative analysis with a wider breadth of existing NLP and ML techniques. The following is presented as a preliminary proof of concept (PoC), articulating the basic road map shown in Fig 1. The result of this PoC expresses the viability of implementing computer–augmented maintenance history analysis.

3 Methodology

Solutions to classification problems are generalized based on their method of training: supervised, semi-supervised, or unsupervised. This paper uses a supervised classification, meaning that pre-defined ground truth labels must be assigned to a training set by an "expert", and a trained model can then be used to predict labels for previously unseen entries. The data set used in this work consists of 779 hand-labeled entries from a manufacturing company's actual maintenance log over the period of several months.

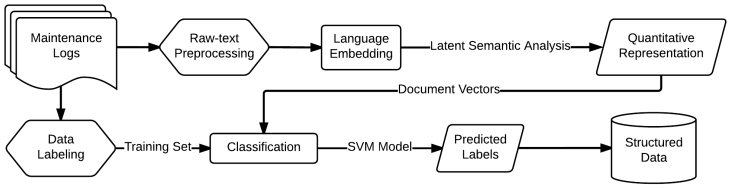

Fig. 1. An overview of the information extraction pipeline for maintenance log text, as used in this work.

3.1 Process Overview

Using text-mining methods in general, and specifically NLP as implemented here, requires several components:

1. Data collection and labeling
2. Text preprocessing and language embedding (vectorization)
3. Classification model training and validation

This process is summarized briefly in Fig. 1. The data set of 779 entries was manually labeled from short form problem descriptions to train a classifier (Sec. 3.2). To represent the logs numerically, vectorization was done using a version of Topic modeling closely related to Latent Semantic Analysis (Sec. 3.3). Finally, a Support Vector Machine (SVM) was trained to label documents based on the provided training set (Sec. 3.4).

3.2 Data Collection and Labeling

Each of the 779 entries was tagged with ground-truth "effect" labels via an "expert human classifier". During this "data labeling" step, the expert was tasked with tersely describing the observed "effect" of a problem recorded in the original log. For example:

Log Text	*Conveyor belt between machine and wash conveyor worn*
Effect Label	*Conveyor Belt Failure*

A total of 352 unique target "effect" labels were given by the expert, with the most common label occurring 39 times. A significant number of entries, 270 entries, have a unique label that is not repeated within the data set. To aid in classification quality for the more "important" effects, those with less than 4 occurrences, are re-labeled as "miscellaneous" or "uncharacterized". Placing these "uncharacterized" labels in the training set helps to solidify the model's recognition of high-frequency effects, and avoid prediction in areas of high uncertainty. In effect, one could interpret an uncharacterized label as any entry that is "not significantly represented" within the model[1]. Using a vector of zeros

[1] This may be viewed as equivalent to a form of outlier detection (or noise filtering), where the labels used frequently by the expert are considered "useful" for classification of future problems.

to represent these "not significantly represented" labels improves classification accuracy of identified labels by around 50%, while also providing well-defined identification of labels either unlike any in the training set, or those that the model is highly uncertain about. Any new or uncertain labels identified by the model could then be delivered to a user for better expert identification, and subsequent inclusion into a retrained model.

3.3 Text Preprocessing and Vectorization

To train a classifier on the labeled data, a numeric representation of the text-based maintenance logs (the documents) is needed. This semantic language embedding is done in two steps: 1) preprocess the text to cleanse it of non-useful artifacts, and 2) vectorize the processed text into manageable form by using a topic model.

Raw-text Preprocessing Cleaning each query (i.e. each issue as described in a string of words) by removing extraneous punctuation and inconsequential words (a.k.a. stop words) (Leskovec et al., 2014) is common practice in NLP techniques. After cleaning, the string is parsed into words and phrases up to N words long and placed in sparse word frequency matrix via the Bag of Words (BoW) technique. During the construction of this matrix, common pluralizations of words are combined (i.e. treated as a single entry) and tokens that have a very low occurrence rate within the corpus (less than 3 instances) are removed. In this context, a token is a word, or group of ordered words (phrase) that appear in the corpus. This dimensionality reduction aids both in convergence and processing time for the language embedding and classification algorithms.

Language Embedding To create a numeric feature space, useful for computerized classification, this work loosely follows the process used in Latent Semantic Analysis (Dumais, 2004). The entire set of text (the *corpus*) is represented as a term-document Matrix, a frequency-based vectorization of word occurrence (BoW, as referred earlier).

To reduce the dimensionality of this corpus word-frequency feature space, a reduced-rank Principal Component Decomposition was performed, and the top n-largest principal components were retained so that the rank-n approximation had 90% variance retention. One interpretation of each of the n the principal component vectors is as a weighted combination of words/phrases forming a *topic* in the corpus. Thus the name, "Topic Model".

Finally, due to very low term-overlap between maintenance topics within this corpus, the document vectors were weighted by the most common token found *elsewhere in the corpus*. This was done with an ordinary-least-squares (OLS) mapping, essentially a prediction on how a phrase would be most commonly talked about in the rest of the corpus. The authors hypothesize that this might take advantage of the natural structure of maintenance-like data, and preliminary results suggest accurate performance under this weighting scheme.

3.4 Classification: SVM Model

The last phase of the presented method is to train a supervised classifier on the expert diagnostic tags provided for each training entry. Here, a Support Vector Machine (SVM) classifier was selected for this task, which has been previously shown to have excellent ability in text classification (Joachims, 1998).

Data is randomly split into train- and test-groups, allowing the SVM to learn on a subset of the data, and then be validated by predicting labels on another, smaller subset (see Sec. 4.1 for a discussion on the effect of training-set size). The classifier was trained against binary vectors representing the target "ground truth" label. Upon subsequent input, it outputs a relative likelihood for each of the possible category labels, and assigns the most probable label to that input.

As a modification to the default labeling, if none of the potential labels had a relative likelihood 0.5 (50% likely) or more, the input was deemed too far away from previous training vectors and therefore an "uncharacterized set entry". This "uncharacterized set" labeling is crucial for identifying labels that are not well characterized in the model so that they can be further analyzed by the expert and retrained into the model, when Human-in-the-Loop training is possible.

4 Results

The results presented in this section enumerate classification performance averaged over twenty trials where the designation of training, testing, and validation samples are randomized between tests. The influence of training set size on performance is detailed, though this is recognized as only a small subset of controllable parameters in this model. A broad-sensitivity study, though outside the scope of this work, is a crucial part of future work (Sec. 5).

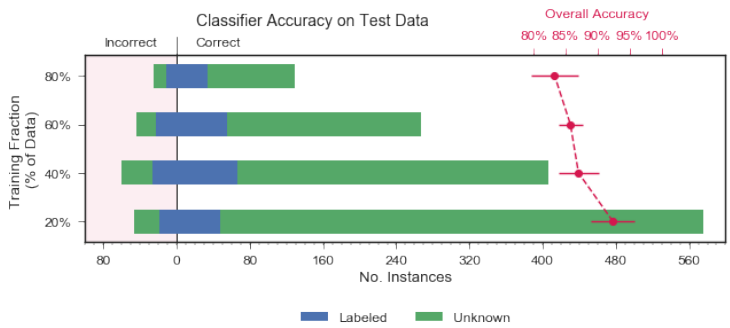

Fig. 2. The effects on classification accuracy of the fraction of data used for training the SVM. Mean number of correct and incorrect labellings over 20 trials are shown against the lower axis for each training-fraction setting. Additionally, the overall prediction accuracy is shown against the upper axis with 1σ uncertainty bars.

4.1 Training Set Size

To determine the robustness of the described method to the availability of training data, the SVM was trained on varying proportions of the data selected at random (see Fig. 2). Each increase in the proportion of training data is analogous to a human manually tagging entries marked by the algorithm as unknown, and/or correcting any observed mislabeling, then iteratively retraining the model. In this way we can simulate Human-in-the-Loop training. Note that as this data-set is finite in size, when an increasing amount (e.g. 80%) is used for training, the total number of validation instances or examples reported in the results must drop (e.g. to 20%) corresponding to the total available entries.

For this work, an "uncharacterized set entry" is a label that has too few (or zero) examples in the training set to be confidently characterized by the classification model. Intuitively, as the percentage of training examples increases, this number of "uncharacterized set entries" will likely decrease somewhat. However, for this data-set, and likely any coming from real world industry maintenance logs, there is a base level of these rarely occurring entries. Thus as important as it is to correctly classify or label those entries that can be characterized by the model, it is equally important to identify those that cannot. This not only lowers misclassification chances, it also allows such entries to be flagged for further investigations by an external human operator, who can then correctly label and add it to the model.

It is important to note that the classifier's ability to correctly identify well-documented labels (i.e. ones *not* in the "uncharacterized set") is relatively invariant across training-set sizes. This implies that, once the model has learned which labels are reliable, its confidence level in predicting them — and by extension, its performance — is likely to remain consistent, even as new information is added to the model. In other words, these results imply that once any arbitrary label obtains enough entries to be able to be characterized by the model, the expected correct classification rate of that label is consistent, regardless of what the label actually is. Thus adding additional labeled examples is more about extending the model coverage than improving performance; although intuitively performance should also somewhat improve.

In addition, the model is able to correctly isolate the"uncharacterized set entries" not previously seen in the training data, but this ability decreases with a broader selection of labeled entries in the training set. This indicates that, as more of the feature space is populated, an entry must be further away from any known labels to be identified as unknown. Consequently, a dynamic, rather than static threshold of confidence when classifying "uncharacterized set entries" as outliers may be more appropriate for this type of discrimination. Lowering the confidence criteria for labeling "uncharacterized set entries" from 50% to 40% in one of the tests caused an average increase of 3% in correct classification, but also a drop of 6% in the correct identification of "uncharacterized set" labels. Significant improvement might come from re-defining the confidence threshold on-the-fly, or defining it in a feature-specific manner.

As shown in Fig. 2, the increased *total* accuracy at lower levels of training data is driven by the model's increased ability to identify unknown entries. This should not lead one to assume that less training data is a favorable; rather, this reaffirms the need to accurately identify the "uncharacterized set entries" at higher proportions of training data, where the total feature space becomes more populated. As the well-characterized areas—the "known" areas of the feature space–become more dense, the "uncharacterized set" areas become harder to distinguish. Additional investigations into methods for managing and mitigating this effect will be part of on going work, perhaps utilizing more crisp kernel models in the classification algorithm, or other similar techniques

Additional investigations on n-gram parsing (Brown et al., 1992) and rare token exclusion have been performed, but a full review is left out due to space constraints. In brief, it was found that additional complexity added via n-gram parsing was unnecessary, likely due to the domain-targeted language of the data set. Conversely, regulating the minimal word token occurrence frequency did seem to have a significant effect on the model's performance. Removal of highly superfluous or infrequent terms aided classification through dimensionality reduction with indications that, for a given data set, there is an optimal occurrence frequency band to include in the model. Further investigations into the generalization to a broader range of data sets of these findings as well as additional areas of inquiry are left for future work.

5 Conclusions & Future Work

Virtual mountains of historic maintenance logs representing an untold wealth of diagnostic knowledge exist throughout industry. Without proper tools and techniques to analyze and contextualize that data, the usefulness of these maintenance logs is severely limited. Presented in this work is a proof-of-concept algorithmic framework for characterizing one aspect of that data. By categorically labeling the generally free form and fragmented text patterns associated with industrial maintenance logs, historical commonalities and recurring problem areas can readily be identified and targeted for process improvement.

The methodology detailed in this work is shown on a preliminary case study to consistently categorize and label a free form maintenance log entry from a set of known labels with over 70% accuracy. Additionally, the algorithm can correctly identify log entries as unique (or potentially needing better labeling) with over 85% accuracy.

Work in this area is a fertile ground for many avenues of continuing and future research. Especially apparent is the need for a broad overview of available methods for training and classifying natural-language texts in the form of maintenance logs. Automated selection of an optimal model for prediction of labels in a given industry or use case is crucial to ensure the best performance. Comparing other quantitative representations for language, like Word2Vec semantic embedding, will provide an excellent means of discovering maintenance-specific patterns in the text logs. In addition, the efficient utilization of domain-expert

knowledge will be crucial in implementing systems like this one, leading to a more dynamic ability to parse data with Human-in-the-Loop system schemes.

The authors believe a set of guidelines for selecting appropriate algorithms based on amount and quality of data, as well as the desired outputs, could accelerate maintenance information utilization. Taking advantage of information hidden in maintenance logs could help bolster productivity, improve maintenance practices, and ultimately save time and money wasted on patching trivial symptomatic problems instead of focusing on the root cause.

References

BOUTI, A., KADI, D.A.: A state-of-the-art review of fmea/fmeca. International Journal of Reliability, Quality and Safety Engineering 01(04), 515–543 (1994), http://www.worldscientific.com/doi/abs/10.1142/S0218539394000362

Brown, P.F., Desouza, P.V., Mercer, R.L., Pietra, V.J.D., Lai, J.C.: Class-based n-gram models of natural language. Computational linguistics 18(4), 467–479 (1992)

Brundage, M.P., Kulvantunyou, B., Ademujimi, T., Rakshith, B.: Smart manufacturing through a framework for a knowledge-based diagnosis system. In: Proceedings of the ASME 2017 International Manufacturing Science and Engineering Conference, MSEC2017, June 4-8, 2017, University of Southern California, Los Angeles, CA (2017)

Dumais, S.T.: Latent semantic analysis. Annual Review of Information Science and Technology 38(1), 188–230 (2004), http://dx.doi.org/10.1002/aris.1440380105

Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: European conference on machine learning. pp. 137–142. Springer (1998)

Juran, J., Godfrey, A.B.: Quality handbook. McGraw-Hill (1999)

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

Liu, H.C., Liu, L., Liu, N.: Risk evaluation approaches in failure mode and effects analysis: A literature review. Expert Systems with Applications 40(2), 828 – 838 (2013), http://www.sciencedirect.com/science/article/pii/S0957417412009712

Shrouti, C., Franciosa, P., Ceglarek, D.: Root cause analysis of product service failure using computer experimentation technique. Procedia CIRP 11, 44 – 49 (2013), http://www.sciencedirect.com/science/article/pii/S221282711300543X