Technical Language Processing: Unlocking Maintenance Knowledge

Michael P. Brundage[a,*], Thurston Sexton[a], Melinda Hodkiewicz[b], Alden Dima[a], Sarah Lukens[c]

[a]*National Institute of Standards and Technology, Gaithersburg, MD 20899, USA*
[b]*Faculty of Engineering and Mathematical Sciences, University of Western Australia, Perth, 6009, Australia*
[c]*GE Digital, San Ramon, CA, 94583, USA*

Abstract

Out-of-the-box natural-language processing (NLP) pipelines need re-imagining to understand and meet the requirements of engineering data. Text-based documents account for a significant portion of data collected during the life cycle of asset management and the valuable information these documents contain is underutilized in analysis. Meanwhile, researchers historically design NLP pipelines with non-technical language in mind. This means industrial implementations are built on tools intended for non-technical use cases, suffering from a lack of verification, validation, and ultimately, personnel trust. To mitigate these sources of risk, we encourage a holistic, domain-driven approach to using NLP in a technical engineering setting, a paradigm we refer to as *Technical Language Processing (TLP)*. Toward this end, the industrial asset management community must collectively redouble efforts toward production of and consensus around key domain-specific resources, including: 1) goal-driven data representations, 2) flexible entity type definitions and dictionaries, and 3) improved access to data-sets – raw and annotated. This collective action allows the maintenance community to follow in the path of other scientific communities, e.g., medicine, to develop and utilize these public resources to make TLP a key contributor to Industry 4.0.

1. Current Paradigm

Maintenance Work Orders (MWO) capture the health history of an asset: the "clinical notes" of an asset management system. Maintainers enter data relating to equipment inspections, diagnoses and corrective actions into their organization's Computerized Maintenance Management Systems (CMMS). Typical MWO health data contain (a) messy, unstructured raw text; and (b) inconsistent, incomplete, incorrect, or missing codes and data formats [1]. These data issues are ubiquitous throughout industry, so MWOs – rich with the health history of the asset – largely sit untouched and are seldom analyzed in a robust and reproducible way. Imagine if, to diagnose complaints of indigestion and shortness of breath, a physician ignored past notes, studies, or health history and only used diagnostic testing. By studying diagnostic tests, without additional context provided by records, this doctor could miss connections beyond stomach issues, like a heart attack. This process is how most mainte-

Table 1: A list of example maintenance work orders

Example Maintenance Work Orders
A *Hyd leak and hydrau power pack noise reported;*
B *Hydromat smroke; Replaced hyd machine;*
C *Hyd leak at saw atachment. Hydromat Saw 012, hydpump not working; rep with new HS012.*

*Corresponding author: mpb1@nist.gov

nance jobs are analyzed, without the context and health history provided by MWOs.

2. Problems with NLP on Technical Text

To analyze technical text data, the scientific community – medicine in particular – successfully adapted key natural language processing (NLP) tools to their text-based data [2, 3, 4]. Asset-rich organizations are also investigating NLP to analyze their MWO text data [5, 6, 7]. Easy-to-access NLP packages[1] allow analysts to stitch together out-of-the-box packages into reasonable pipelines that can preprocess and analyze their text [8, 9]. In the following sections, we describe NLP core concepts and pitfalls with technical text. We illustrate how these pitfalls can be overcome through a community-driven effort to move the maintenance profession forward.

Fig. 1 shows an example of an NLP workflow with technical language. The left side shows common NLP steps applied to the text of MWO *C* from Table 1. The right side presents pros and cons of each of these steps. Each of these steps and the results as the MWO text is transformed is described below.

2.1. Preprocessing

Raw text data is first preprocessed, as follows:

Tokenization: The process of separating text into meaningful units such as words and numbers [10, 11]. In technical language, this step will fail to account for words that are linked accidentally, such as hydpump.

Stop Word Removal: The elimination of common words, such as *this, that, the* that convey little semantic meaning [11]. Using the stopword list from Natural Language Toolkit (NLTK)[2] removes words like 'saw' and 'not' reversing the meaning of the MWO: instead of the correct "pump was NOT working", the output is the "pump WAS working".

Cleaning: The removal of characters and tokens, e.g., punctuation and numbers from the text to reduce noise that can negatively impact analysis [12, 13]. In technical data, this step removes valuable information such as asset numbers (i.e., HS012) or similar entities.

Stemming/Lemmatizing: The reduction of inflected words to a common base form [13], for example:

$$\{\text{boat, boats, boat's, boats'}\} \Rightarrow \text{boat}$$

In this example technical data processed with NLTK's stemmers failed to properly merge "hyd" "hydraulic" and "hydaulic".

2.2. Text Analysis

Text analysis refers to a pipeline that extracts decision-level information. To do this automatically, without constant intervention, known data-labels *annotations*, along with "inputs" — numerically-encoded *representations* of the preprocessed text — are used to "train" a model that learns to accomplish a *task* on new data[3].

Annotation: Automated analysis ironically requires an initial investment in manual labeling. This annotation can take many forms, including categorization, tagging, and highlighting values in already collected data, such as from a CMMS. In practice labels – e.g., fault codes, maintenance work type – are often captured inconsistently [1] or incorrectly [14]. Thus, allocating labor to create an accurate, large annotated dataset is necessary. This step can be prohibitively cost and time expensive for many manufacturers and can create a perception that these datasets are valuable intellectual property.

Data Representation: The preprocessed text must be converted into a proper data representation for the desired analytic algorithm. For example, Bag-of-Words (BoW) models are common, where each word and its count in the MWO text is represented as an element in a vector without ordering information [13]. Other approaches consider syntactic structures and/or may contain features from machine learning, such as word embeddings [15]. Many of these models make assumptions about quantity or length of text that are rarely questioned, yet rarely hold true in short, technical text.

Analysis Tasks: Text analyses are tasks that map inputs to desired outputs. These tasks can include a) single label classification (e.g., mapping text to an expected fault code), b) multi-label classification (e.g., mapping text to several fault codes), or

[1] For instance: OpenNLP (Apache), NLTK, SpaCy, PyNLPl (Python), Stanford CoreNLP

[2] https://www.nltk.org/

[3] This only covers a subset of possible analyses, namely, *supervised learning*. Other analysis is possible, but not discussed here. The core idea remains: data with some annotation and encoding assumptions are used to accomplish analysis tasks.

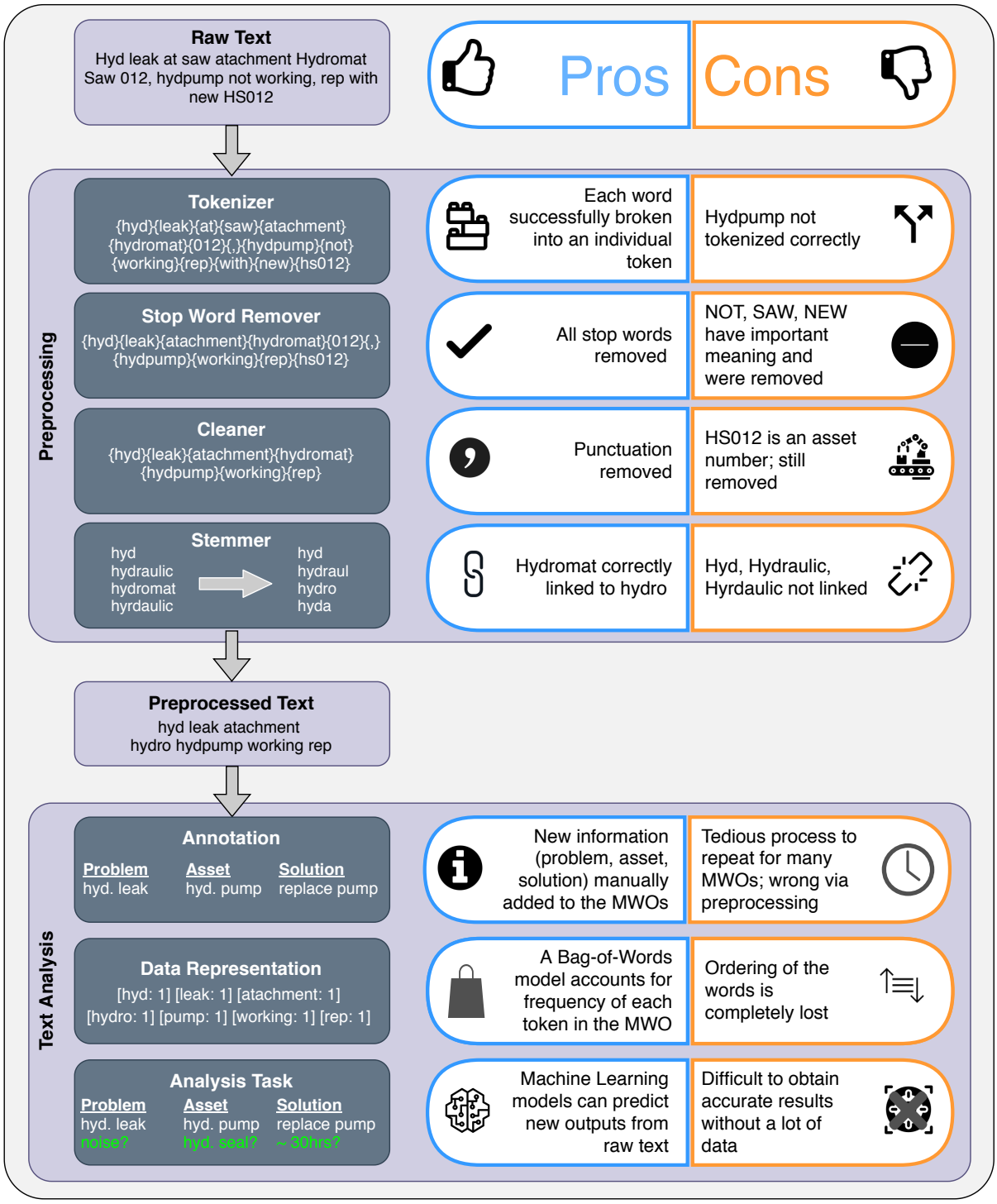

Figure 1: A common NLP workflow for technical language text data. The left side of the figure illustrates the output of each step. The right side shows the pros and cons of each step when applied to technical text. This example is meant to be illustrative of real problems that can arise with NLP pipelines on technical text.

c) Named Entity Recognition (NER) (e.g., locating and classifying entities within the text into predefined categories). The task outputs for technical domains require large, detailed inputs, drastically increasing annotation requirements.

In practice, the lines between input and output are not well-defined. An analyst might use intermediary tasks and representations to enrich annotations and cascade into further tasks. A holistic approach to improving one component will inevitably improve the others; a stolid adherence to a given pipeline can prevent progress all-around.

The issues of labor cost and valuable intellectual property surrounding annotated datasets often loom large in text analysis. However, hope exists for possible high-performing analysis pipelines: Seale et al. used similar steps as described above and achieved an impressive 96 % accuracy in [8]. But to achieve that same success on a wide scale, history from other disciplines shows that annotated datasets and shared model-training pipelines are vital to create collective domain expertise and tooling [16]. Lacking a similar effort as other domains, maintenance has not seen the same NLP-driven renaissance. Without maintenance community support and resources, text analysis cannot achieve the needed level of development for wide-scale adoption.

By lowering barriers to entry for text analysis through the development of efficiency-boosting tools and a more human-centered annotation approach, engineers have a unique opportunity to simultaneously learn from other domains and improve on their processes. A new approach is needed to adapt NLP methods to industry use cases in a scalable and reproducible way: Technical Language Processing (TLP).

3. Technical Language Processing

TLP is a human-in-the-loop, iterative approach to tailor NLP tools to engineering data. Figure 2 shows a conceptual diagram of a TLP methodology, where:

A Engineering use cases are explicitly considered as an input along with the raw text.

B NLP resources, such as tokenizers or embeddings, are transparent parts of a process that builds specialized TLP resources.

C Computational support tools alleviate some burden on domain experts, while continually eliciting their support as appropriate.

D Collaboration between analysts and domain experts to improve TLP resources and computational support will reduce error and increase trust in analyses.

E Community-driven TLP resources are iteratively developed and used in a transparent, reproducible way.

F Text analysis influences future TLP resource development.

So how do we make TLP a reality? Industrial leaders, standards organizations, professional societies, and researchers must work together to develop robust and widely applicable community resources and TLP solutions.

3.1. Data Representations

Maintainers need to understand potential data representations and how they apply to maintenance use cases. As an example, if the goal is to understand the causes of a problem, a Bag-of-Words model may fail to make use of key information, e.g., word order.

Maintainers could focus on classifying specific words (e.g., hydraulic pump or leak) to their classifications (e.g., component and problem action) rather than on classifying the work-order as a whole. Computational support is readily available for this approach. Algorithms can rank terms by estimated importance, presenting individual words or phrases one at a time. This allows for rapid annotation of the most-used concepts, providing a sense of priority and system trust.

3.2. Entity Types Definitions and Dictionaries

Once the TLP data representations are understood, the entity types used as annotation and intermediate inputs must be researched. In traditional NLP, Named Entity Recognition (NER) is used to discover entities such as Persons (e.g., Mike) or Organizations (e.g., Google).

No wide-spread community consensus or adoption exists for agreed on entity sets or hierarchies in maintenance. For entity types to be scaled across different asset data sets, researchers and industry must collaborate to determine standard entity types for industry use cases.

Data dictionaries provide mappings for ubiquitous terms in industry (e.g., “Replace” = “Solution Action”). Such dictionaries allow experts to spend more time labelling facility-specific words. Computational tools – active learning or adaptive annotation systems – make facility-specific tagging easier for maintainers.

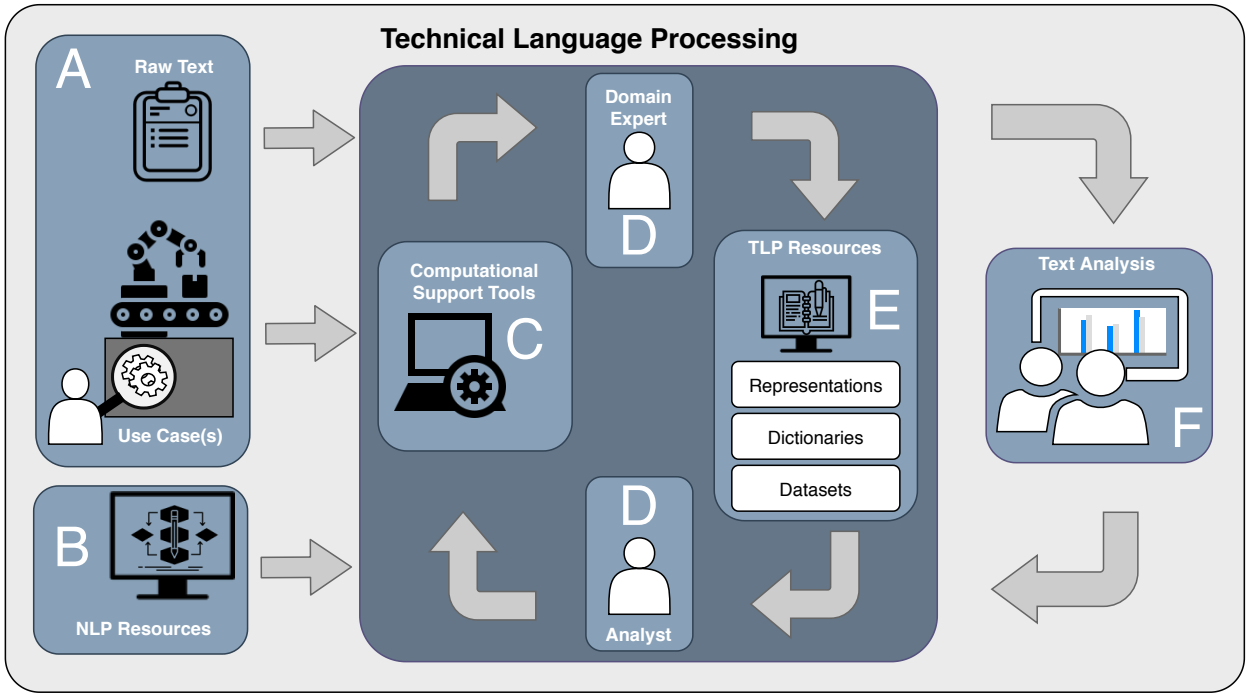

Figure 2: A conceptual diagram of the TLP workflow. This workflow illustrates the iterative, human-in-the-loop process to tailor NLP tools for engineering text-based data.

3.3. Raw and Annotated Datasets

Once these resources are developed, they can be incorporated into real industry datasets as training data. Industry datasets should contain the raw text data, but also other maintenance information (e.g., dates, cost) and entity information (e.g., components, problems). These datasets form the foundation for the development of open TLP tools.

4. Conclusions

TLP can unlock maintenance knowledge hidden in short text, providing needed insights from the asset health history while making maintenance decisions. The research community has started developing maintenance resources, but they are currently small in size, in their infancy, and not as diverse as other domains. Table 2 provides a list of available resources, including a TLP Community of Interest (COI)[4] [30]. We present an opportunity for the entire maintenance community to work together to accelerate development and adoption of these resources to make TLP a reality.

[4]https://www.nist.gov/el/tlp-coi

NIST Disclaimer

The use of any products described in this paper does not imply endorsement by NIST, nor does it imply that products are necessarily the best available for the purpose.

References

[1] Hodkiewicz, M., Kelly, P., Sikorska, J., Gouws, L.. A framework to assess data quality for reliability variables. In: Engineering Asset Management. Springer; 2006, p. 137–147.

[2] Chen, X., Xie, H., Wang, F.L., Liu, Z., Xu, J., Hao, T.. A bibliometric analysis of natural language processing in medical research. BMC Medical Informatics and Decision Making 2018;18(1):14.

[3] Meystre, S., Haug, P.J.. Natural language processing to extract medical problems from electronic clinical documents: performance evaluation. Journal of biomedical informatics 2006;39(6):589–599.

[4] Zhou, L., Hripcsak, G.. Temporal reasoning with medical data—a review with emphasis on medical natural language processing. Journal of biomedical informatics 2007;40(2):183–202.

[5] Brundage, M.P., Weiss, B.A., Pellegrino, J.. Summary report: Standards requirements gathering workshop for natural language analysis. NIST AMS 100-30 2020;.

[6] Lukens, S., Naik, M., Saetia, K., Hu, X.. Best practices framework for improving maintenance data quality to enable asset performance analytics. In: Proceedings of the Annual Conference of the PHM Society. Scottsdale, AZ; 2019,.

Table 2: A list of maintenance specific TLP resources. These resources are a starting point for the maintenance community to make TLP a reality.

	Ongoing Maintenance TLP	**Description**
TLP Resources		
Representations	Token-based [17, 18]	These papers describe a token-based method for MWO annotation
	Embeddings - CamemBERT [19]	This paper uses transfer learning to predict the criticality and duration of maintenance issues
Dictionaries & Entity Types	ISO 15926 [20]	Reference data for recording information about process plants
	ROMAIN [21]; IOF Maintenance WG [22]	Maintenance management ontology
	ISO 14224 [23]; NERC-GADS Data Reporting Instructions [24]	Bases for the collection of reliability and maintenance (RM) data for equipment in oil and gas and electric utility industries, respectively
Datasets	Excavator Maintenance dataset [25]	The `Excavators_Raw&Cleaned` dataset provides annotated MWOs
	NYC Maintenance dataset [26]	This dataset is raw maintenance work orders for park equipment
Development Support		
Computational Tools	Nestor [27, 28]	A free toolkit that helps maintainers annotate their MWOs by tagging
	Redcoat [29]	A web-based annotation tool that supports collaborative hierarchical entity typing
Overviews & Expertise	TLP COI [30]	Emerging Community of Interest to discuss TLP needs and solutions
	NLP workshop report [5]	Workshop report on current trends, successes, and challenges with respect to NLP for maintenance in manufacturing.
	NLP standards needs [31]	Discussion on standards needs for NLP in maintenance
	Data Entry & Human Factors [14]	Analysis of potential error sources and mitigation for maintenance data-entry

[7] Rajpathak, D., De, S.. A data-and ontology-driven text mining-based construction of reliability model to analyze and predict component failures. Knowledge and Information Systems 2016;46(1):87–113.

[8] Seale, M., Hines, A., Nabholz, G., Ruvinsky, A., Eslinger, O., Rigoni, N., et al. Approaches for using machine learning algorithms with large label sets for rotorcraft maintenance. In: 2019 IEEE Aerospace Conference. IEEE; 2019, p. 1–8.

[9] Sharp, M., Sexton, T., Brundage, M.P.. Toward semi-autonomous information extraction for unstructured maintenance data in root cause analysis. In: IFIP International Conference on Advances in Production Management Systems. Springer, Cham; 2017, p. 425–432.

[10] Palmer, D.D.. Tokenisation and sentence segmentation. Handbook of natural language processing 2000;.

[11] Manning, C.D., Schutze, H.. Foundations of Statistical Natural Language Processing. Cambridge, MA: The MIT Press; 1999.

[12] Denny, M.J., Spirling, A.. Text preprocessing for unsupervised learning: Why it matters, when it misleads, and what to do about it. Polit Anal 2018;26(2):168–189.

[13] Manning, C.D., Raghavan, P., Schütze, H.. Introduction to Information Retrieval; vol. 20. Cambridge University Press; 2008.

[14] Sexton, T., Hodkiewicz, M., Brundage, M.P.. Categorization errors for data entry in maintenance work-orders. In: Proceedings of the Annual Conference of the PHM Society; vol. 11. 2019,.

[15] Geigle, C., Mei, Q., Zhai, C.. Feature engineering for text data. In: Dong, G., Liu, H., editors. Feature engineering for machine learning and data analytics; chap. 2. Boca Raton, FL: CRC Press; 2018, p. 15–54.

[16] Sheikhalishahi, S., Miotto, R., Dudley, J.T., Lavelli, A., Rinaldi, F., Osmani, V.. Natural language processing of clinical notes on chronic diseases: systematic review. JMIR medical informatics 2019;7(2):e12239.

[17] Sexton, T., Brundage, M.P., Hoffman, M., Morris, K.C.. Hybrid datafication of maintenance logs from ai-assisted human tags. In: 2017 ieee international conference on big data (big data). IEEE; 2017, p. 1769–1777.

[18] Sexton, T., Hodkiewicz, M., Brundage, M.P., Smoker, T.. Benchmarking for keyword extraction methodologies in maintenance work orders. In: PHM Society Conference; vol. 10. 2018,.

[19] Cadavid, J.P.U., Grabot, B., Lamouri, S., Pellerin, R., Fortin, A.. Valuing free-form text data from maintenance logs through transfer learning with camembert. Enterprise Information Systems 2020;0(0):1–29. doi:\bibinfo{doi}{10.1080/17517575.2020.1790043}. https://doi.org/10.1080/17517575.2020.1790043; URL https://doi.org/10.1080/17517575.2020.1790043.

[20] ISO, . ISO/TS 15926-4:2019 Industrial automation systems and integration — Integration of life-cycle data for process plants including oil and gas production facilities — Part 4: Initial reference data. Geneva Switzerland; 2019.

[21] Karray, M.H., Ameri, F., Hodkiewicz, M., Louge, T.. Romain: Towards a bfo compliant reference ontology for industrial maintenance. Applied Ontology 2019;14(2):155–177.

[22] IOF, . "IOF Maintenance WG"; 2020. Available at https://www.industrialontologies.org/?page_id=92. Accessed 09-03-20.

[23] ISO, . ISO 14224:2016 Petroleum, petrochemical and natural gas industries — Collection and exchange of reliability and maintenance data for equipment. Geneva Switzerland; 2016.

[24] NERC, . Generating Availability Data System Data Reporting Instructions. Atlanta, GA; 2020.

[25] Lab, U.S.H.. "Prognostics Data Library"; 2020. Available at https://prognosticsdl.ecm.uwa.edu.au/pdl/. Accessed 09-03-20.

[26] Data, N.O.. "Asset Management Parks System (AMPS) - Work Orders"; 2020. Available at https://data.cityofnewyork.us/Environment/Asset-Management-Parks-System-AMPS-Work-Orders/8sdw-8vja. Accessed 09-03-20.

[27] Sexton, T.B., Brundage, M.P.. Nestor: A tool for natural language annotation of short texts. J Res NIST 2019;124.

[28] NIST, . "Nestor"; 2020. Available at https://www.nist.gov/services-resources/software/nestor. Accessed 09-03-20.

[29] Stewart, M., Liu, W., Cardell-Oliver, R.. Redcoat: A collaborative annotation tool for hierarchical entity typing. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations. 2019, p. 193–198.

[30] NIST, . "Technical Language Processing Community of Interest"; 2020. Available at https://www.nist.gov/el/tlp-coi.

[31] Sexton, T., Brundage, M.. Standards needs for maintenance work order analysis in manufacturing. Proceedings of the 2019 Model-Based Enterprise (MBE) Summit 2019;.