Received 11 February, 2021; | Revised 6 April, 2021; | Accepted 24 May, 2021

DOI: xxx/xxxx

LETTER

Adapting natural language processing for technical text †

Alden Dima*[1] | Sarah Lukens[2] | Melinda Hodkiewicz[3] | Thurston Sexton[4] | Michael P. Brundage[4]

[1]Information Technology Laboratory, National Institute of Standards and Technology, Maryland, USA
[2]GE Digital, California, USA
[3]Faculty of Engineering and Mathematical Sciences, The University of Western Australia, Western Australia, Australia
[4]Engineering Laboratory, National Institute of Standards and Technology, Maryland, USA

Correspondence
*Alden Dima, 100 Bureau Drive
Gaithersburg, MD 20899-8970
Email: alden.dima@nist.gov

Abstract

Despite recent dramatic successes, Natural Language Processing (NLP) is not ready to address a variety of real-world problems. Its reliance on large standard corpora, a training and evaluation paradigm that favors the learning of shallow heuristics, and large computational resource requirements, makes domain-specific application of even the most successful NLP techniques difficult. This paper proposes Technical Language Processing (TLP) which brings engineering principles and practices to NLP specifically for the purpose of extracting actionable information from language generated by experts in their technical tasks, systems, and processes. TLP envisages NLP as a socio-technical system rather than as an algorithmic pipeline. We describe how the TLP approach to meaning and generalization differs from that of NLP, how data quantity and quality can be addressed in engineering technical domains, and the potential risks of not adapting NLP for technical use cases. Engineering problems can benefit immensely from the inclusion of knowledge from unstructured data, currently unavailable due to issues with out of the box NLP packages. We illustrate the TLP approach by focusing on maintenance in industrial organizations as a case-study.

KEYWORDS:
natural language processing, technical language processing, technical data, maintenance records, domain adaptation

1 | INTRODUCTION

Natural Language Processing (NLP) has recently made rapid and significant advances across a wide variety of tasks. These were enabled by improvements in language models that predict characters, words, or sentences from surrounding context, which have become a central theme in NLP research [1,2,3]. The foremost example, Generative Pre-trained Transformer 3 (GPT-3), has been dubbed the "most powerful language model ever" [4] and recently demonstrated strong performance on many existing data sets for a variety of NLP tasks such as translation, question answering, unscrambling words, and news article generation [5]. Early users have shown its ability to generate text ranging from guitar tablature, to website layouts, to computer code [4].

For engineers and technical analysts wishing to use NLP as part of their analyses of technical processes, there is less reason to be optimistic. Despite impressive results with standard challenge data sets, an open question remains as to what state-of-the-art (SOTA) models are actually learning [6]. In particular, claims that NLP systems understand language or the meaning of text are overblown as evidenced by the failure of SOTA models to generalize learned knowledge in a human-like manner [1,3,6,7].

There is also concern that the current NLP training and evaluation paradigm naturally favors models for which large amounts of data are available [3]. This may not be an issue for academic or research NLP systems: they are often successful when trained on "standard" text that comes from e.g. English news wire and other literature [8,9]. However, text encountered in technical applications, such as in industrial operations, differs significantly from these benchmarks, causing performance of deployed NLP systems to drop [8,10], often to unacceptable levels.

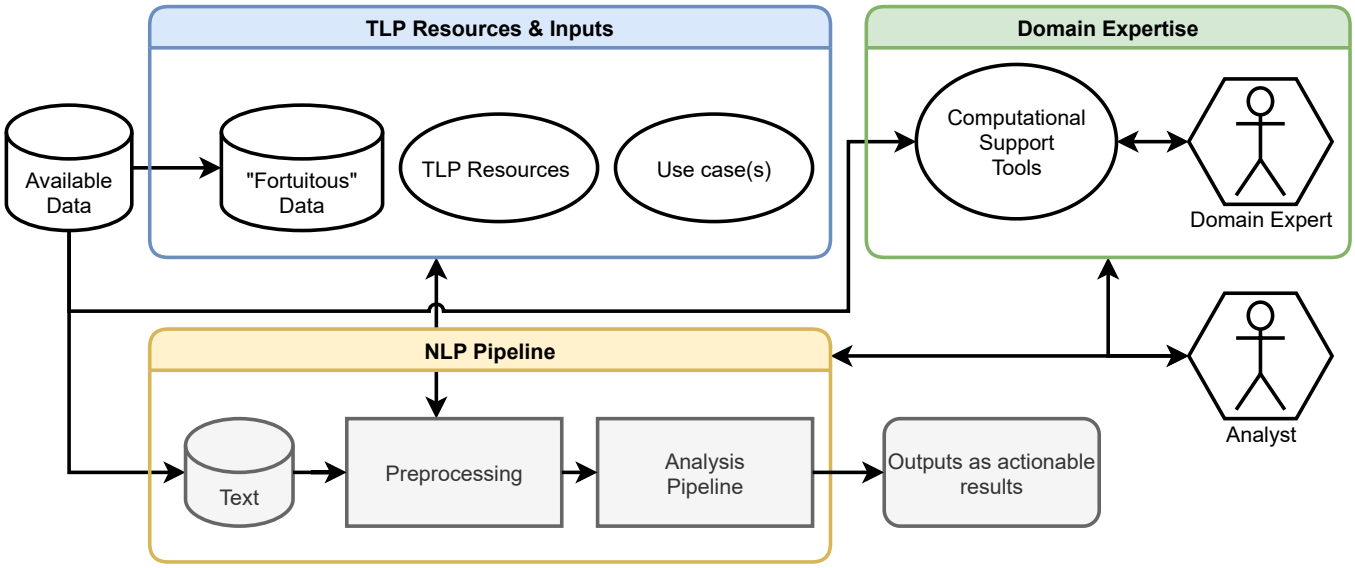

FIGURE 1: Technical Language Processing expands the system boundary beyond the traditional NLP pipeline to include users, engineering use cases, TLP resources such as dictionaries, as well as other "fortuitous" data sources (Section 6) which aid in the interpretation of the primary text data.

Despite the volume of text data in industrial engineering, it is in many ways a low-resource domain from the NLP perspective. The traditional response to addressing these domains in machine learning is transfer learning in which the models generated from annotated data from resource-rich domains are adapted for the low-resource domain [11,12,13]. However, these approaches often assume that the differences between two different domains is constrained in particular ways. For example, the lexical, grammatical, and terminological differences between "standard" English and that found in industrial maintenance logs, has spawned a whole set of domain-specific NLP adaptations that are largely outside of mainstream NLP [9,14].

The classical NLP goal of having computers attain human-like language abilities [3] may also bias NLP towards impressive — but complex and resource-intensive — technologies, while ignoring those that are more in line with practical engineering needs [15,16]. With all this in mind, we sought an approach which will help bridge the gap between the promise of NLP and the realities confronted in many technical domains.

Technical Language Processing (TLP) is our proposed human-in-the-loop, iterative approach to tailor NLP tools to technical data that explicitly considers industrial engineering use cases as inputs along with the raw text (see Fig. 1. Our intention is to address perceived shortcomings of applying standard NLP to technical text data. As an engineering discipline, TLP includes explicit notions of process and can catalog and disseminate successful patterns of application. The TLP process builds specialized resources from existing components including NLP techniques such as tokenizers and embeddings. Some of the burden on domain experts is alleviated via computational support tools that elicit expert input when necessary. Analysts also benefit from TLP resources such as industry standards and technical dictionaries. TLP strives to improve its resources and computational support tools to reduce error and increase confidence in analyses through collaboration between analysts and domain experts. Community-driven TLP resource development is iterative and influenced by text analysis.

Our goal for this paper is to further argue for the creation of an NLP field that focuses on the technical text that appears in the computer-mediated communication used to support business processes within specialized domains. We will focus on industrial maintenance as our motivating example and we consider the need for TLP when analyzing the text found in maintenance management systems.

The remainder of this paper is organized as follows. We will discuss maintenance, along with its records and text, and the challenges that they present in Section 2. We will then question in Section 3 whether algorithms have the ability to generalize what they learn and show how TLP addresses this concern. Section 4 introduces two issues related to the use of large data sets and Section 5 examines the problems with domain adaptation for technical text. We discuss the benefits of "fortuitous" data in Section 6 and discuss computational costs and TLP's strategies for mitigating them in Section 7. Using a set of ethical concerns, we present three general risks of applying existing NLP to technical text and why we believe that TLP can help in Section 8. We close with Section 9, a

summary of how TLP addresses the challenges of applying NLP to technical text.

2 | WHAT IS MAINTENANCE?

The health and prosperity of a nation is built on its infrastructure. Consider our roads, water and power networks, buildings, and manufacturing capacity. The assets that provide these services require maintenance. The management of this maintenance is often an invisible process until something fails [17]. Asset maintenance involves a wide variety of stakeholders such as asset owners, operators, contractors, original equipment manufacturers and specialist service providers. All these stakeholders keep their own records about assets. Maintenance records are created by maintenance technicians, engineers and operators. Collectively we call this group 'maintainers'. To become a maintainer requires years of training involving learning the language of engineering and maintenance, developing physical, chemical, structural, electrical and digital knowledge of how assets and asset systems function, and how they fail [18]. Maintainers' training enables them to share information efficiently using common mental models, often codified in standards and standardized or well-known procedures. Maintainers use their expertise to describe the maintenance work they perform, usually in a free text format. Much information, especially about relationships is implicit, and jargon and abbreviations are widely used [19]. The language of engineering and maintenance is challenging for non-maintainers (and computers) to understand.

2.1 | Maintenance Records and Text

A maintenance work order (MWO) is created for every maintenance activity. It may be generated by a maintainer on noticing that an asset needs maintenance work or by a Computerized Maintenance Management System (CMMS), in which case the original work order text would have been generated as semi-structured text by a maintenance planner [20]. Examples of both are shown in Table 1. Hundreds, sometimes thousands of MWOs, are generated each month depending on the complexity of the organization. Currently, without NLP tools that are fit for purpose, all these MWO records need to be read by humans in order to be planned, scheduled and executed. In the past, records were kept on paper, but are nowadays stored in unstructured text fields in relational database systems and spreadsheets. These MWO records are akin to medical records for an individual [21], and are vital to efforts that estimate the reliability of the asset and potential for functional failures. However there are a number of challenges in extracting knowledge from these texts.

2.2 | Maintenance Text Challenges

The text taken from maintenance management systems deviates from "standard" English in a number of ways. As shown in Table 1, the sample MWO's describe the state of an asset and/or the work that needs to be done. Work order description fields can usually be characterized as containing at least one verb such as 'replace' to describe desired action or word that describes the asset state such as 'plugged'. In general, such entities in MWO corpora are unbalanced with a relatively small number of verbs describing maintenance work and the observed state and a large number of n-grams [27] used to describe the assets. As yet there is no widely agreed structure for named entity recognition for those seeking to create annotated data sets. A number of different named entity recognition classes are being used for MWO annotation: Item-Activity-State [10], Item-Problem/ Symptom-Solution/ Action [28,29].

The familiar assumptions of NLP often mislead in the analysis of maintenance text. For example, while the overall number of maintenance records can be similar to the number of documents in an NLP corpora, the MWO text tends to be much smaller (Table 2). Maintenance text itself is often more similar to shorthand notation than standard English text [9,10]. Stop words, commonly removed in NLP, provide important context for the interpretation of MWO [14]. As seen in Table 1, many of the words are domain-specific and most are abbreviations or acronyms, some created by specific individuals or groups of maintainers [9,10,30], that are used inconsistently and interchangeably [14,31] and are not consistently marked with periods [9]. Words can be misspelled, omitted, or run together and longer words are often contracted with sporadic apostrophes [9,14]. Unlike "standard" English where each form of punctuation has a specific use, punctuation in maintenance data is typically used interchangeably to separate distinct ideas [14]. These lexical issues can lead to semantic ones. Multiple instances of parts, actions, and symptoms co-exist in a single record and their correct associations must be established [31]. Many individual concepts are expressed using multiple words, that must be parsed as a single unit to get the intended meaning [14]. The same concept can also be referred to in many different ways, 'frontShockAbsorber', 'shockAbsorbedFront', 'shockFrtAbsorber' and 'brakeAbsorber' all refer to the same part but are lexically inconsistent [29].

As a result of the many challenges associated with maintenance text, there have been domain adaptations, largely ad hoc, some of which we will discuss in Section 5.

3 | DO ALGORITHMS UNDERSTAND?

The excitement of SOTA NLP is often conveyed with claims that these systems understand or capture meaning of the text being analyzed [1]. But what does this really mean? The "symbol

TABLE 1: Example Maintenance Work Orders; abbreviations and typos appear in the original.

Asset identifier	Functional location	Work order description	Date
Pneumatic System 2222	10.01.20-AS222	Replace air dryer silencer	02/09/17 07:57
02 Sump pump	PUMP-AI025-SUP002	pump electrically dead trouble shoot. if u/s disconnect for c/o	04/06/19 08:34
h5 Motor	H5	Replaced pin in pendant and powered machine -Possible short in pendant cable	04/06/20 08:34
Grinding Ball Mill BM001	40.03.05-ML001	1W Mech Insp Ball Mill BM001	04/12/20 --:--
Thickener Concentrate	TR0003	DSHT Cons Thkner rplace bed press.	16/03/16 06:12
Lighting and small power	SE00401	Lighting uprgrades ,grnd flr filter bld	05/06/19 08:27
Fuel tank	EDD0020	Fuel tank leak	04/02/18 09:42
VAV Box	AHU	RESET FAN FAILS AND START EQUIOPMENT	17/02/18 13:42
Tractor	TRD0250	Reseal RH F/drive Komatsu	08/03/19 09:42
150428	216 B54	Emergency retract solonoid failure	02/12/17 13:45
Pump-Centrifugal	ESI-DD01	Control valve may be plugged	04/03/19 12:22
Motor, Exchanger, 20 HP	Flare System 05E112	05E112B -replace damaged motor	06/08/19 15:16

TABLE 2: Comparison of sizes (count and average words-per-document, if reported) of selected work order collections and typical NLP training corpora. The average words per document (WPD) for the work order records is smaller than those of the NLP training documents.

Source	Type	Count	WPD
South African fuel service stations [22]	MWO	373,344	18
Helicopter Maintenance Records [9]	MWO	100,000	—
UWA Excavators Maintenance Records [23]	MWO	5485	5
Reuters-21578 [24,25]	NLP	21,578	160
Reuters Corpus Vol. I [26]	NLP	804,414	200

grounding problem", which occurs when symbols are interpreted based on other symbols in a circular fashion rather than their meaning in the external world, is a concern when evaluating the ability of computational machines to understand the intrinsic meaning in language [32]. NLP systems operate under the distributional hypothesis that words surrounding a word in question give clues to its meaning and when taken in aggregate, all of its contexts appear to give us what we seek [2]. This may especially be not true in technical text though techniques using contextual information have been developed in the automotive industry [31].

The ability to generalize, when a model behaves as expected in novel situations beyond the training context, is closely related to the problem of meaning [33]. Challenges with proper generalization of SOTA NLP systems suggest that such systems are not able to meaningfully learn from their training data, evidenced by inconsistent results when input data differs in distribution from training data and the need for significant retraining to adapt models to new tasks [3,34,35].

Some of this semantic disability can be traced to the current training and evaluation paradigm which does not encourage human-like generalization by having the test data drawn from the same distribution as the training data [3]. Under these conditions, many SOTA learning systems learn shallow heuristics that work for the training data instead of really learning the expected generalizations [6,7,34,36,35,37,38]. The current paradigm and shallow heuristics conspire to create models that are, in a sense, overfitted to particular data sets and lack the ability to generalize as their creators intended [3,39]. As a result, claims that these models offer a human-level capacity for real-world meaning and understanding are exaggerated [1].

TLP is an adaptation of and is firmly rooted in NLP. There is nothing precluding the use of any and all useful NLP approaches. By expanding the system boundary away from algorithms and data pipelines to include humans in the loop, we hope to overcome grounding issues.

Unlike NLP approaches that learn from text in an exclusively unsupervised fashion [2], TLP allows and encourages iterative

human intervention and supervision at every stage; we describe this aspect of TLP in detail in [21]. This connection to the outside world goes beyond merely interfacing with sensors which some believe to be sufficient [32]. We see TLP as leveraging humans to provide a rich source of semantic information and meaningful action through their ability to discriminate between, manipulate, identify, describe, and respond to real world objects, events and states. This will allow us to inject meaning into analyses.

TLP can help tackle the problem of generalization by promoting the use and development of computational resources such as annotation tools that support hybrid datafication via artificial intelligence-assisted human tagging, where datafication refers to the process of structuring text information to facilitate the understanding of its context [40]. These NLP-based tools allow for the manual injection of real-world knowledge into the learning process by providing ontological information that can guide categorization and generalization. Two such tools, Nestor [41] and Redcoat [42], allow for the tagging of short technical text, such as found in maintenance work order descriptions, with annotations that facilitate processing. Machine learning systems can then use these tags as a signal to promote generalization by helping to mitigate the shallow heuristics and spurious correlations that could otherwise affect learning.

4 | MORE DATA ISN'T THE ANSWER

We believe that the problem of learning of shallow heuristics is further exacerbated by two issues associated with analyses of large amounts of data: spurious correlations and the low probability of regularities due to the underlying phenomena of interest [43]. Very large data will contain spurious correlations that exist solely due to the size of the data and not because of any other intrinsic property. Such correlations cannot be distinguished algorithmically from other types of correlations and can overwhelm detection of the "true correlations". Second, even though "true correlations" are the signals sought during analyses, the probability of regularities due to the underlying phenomena appearing in the data is low. The larger the data analyzed, the greater the chance that spurious correlations dominate the results and lead to erroneous conclusions.

The domain-specific pre-processing and data normalization steps in TLP help improve the visibility of system behaviors against their naturally noisy backdrop as well as reducing the opportunity for spurious correlations. Because NLP is focused on "standard" English, we believe that pre-processing and data normalization do not receive sufficient attention. In TLP, these issues become areas of active interest and we expect that practitioners across different TLP domains will share and evaluate experiences and approaches. Over time, TLP will develop a systematic framework for preprocessing and normalization that can be easily adapted to new technical domains.

5 | WHAT ABOUT DOMAIN ADAPTATION?

Domain adaptation is a class of approaches that attempt to transfer learning from a task in a source domain with abundant annotated data to a similar task in a target domain, one with little or no annotated data [12]. An underlying assumption is that there exists a resource-rich domain that is similar enough to the low-resource domain; this is unclear for the technical domains that we are considering, such as maintenance. Adding to the uncertainty, the NLP literature sometimes equates domain adaptation with transfer learning, [13] it lacks a consistent definition for the concept of a domain, and its notion of domain adaptation focuses on assumptions that are unrealistic for technical text.

One such assumption is that syntactic structures and parts of speech (POS) are stable between two domains because they reflect intrinsic properties of a shared, clean natural language whose only differences are the appearance, roles, or distributions of certain domain-specific words [12,13]. Shared features can then be leveraged. So for example, there are known shared words whose POS tags can be used to predict POS tags for unknown words.

There is then an expectation that NLP systems will work sufficiently well when trained either using annotated source domain data alone or with a combination of a small set of annotated data from the new domain combined with the annotated data from the source domain [12]. Normalizing the target domain's data to make it more closely resemble the data used to originally train the system also seems viable [8]. However, given the grammatical, spelling, and usage issues present in technical text, these approaches will likely not work in general, though they might be useful in some contexts. For maintenance, not only are typical NLP systems not suited [9], but neither are standard domain adaptation techniques.

Like other technical domains, a variety of bespoke maintenance-specific NLP adaptations have appeared in the literature. Out-of-the-box pre-processing pipelines require modifications. As part of their work with military aircraft maintenance, Bokinsky et al. [14] and McKenzie et al. [9] made adaptations to Natural Language Toolkit functionalities, such as introducing a token "sterilizer", which addressed observed challenges of inconsistent punctuation, necessary punctuation and words with no semantic difference through injection of special rules - replacing all punctuation with an identical special punctuation token and all tokens containing numbers with a special identical code token.

We see the presence of bespoke NLP adaptations as evidence that the lack of a well-developed notion of a domain is

a central issue that hampers domain adaptation for TLP. We favor the approach taken by Plank[8] which critiques the current approaches to domain adaptation as focusing on the dichotomy between the source and target domains without a real interest in their essential differences[8]. She states that there is little research that addresses how text varies and how these variations affect the use of NLP and proposes a definition of a domain as a region in a high-dimensional variety space. This space is an unknown high-dimensional space whose dimensions include many latent variables beyond the text itself such as social factors. The concept of a **variety space** is defined by a set of variables, some of which are latent, that describe the different ways that texts and their contexts can differ[8]. Variety spaces can accommodate variables that exist outside of the text like gender or geographic location, the medium used, or area of domain expertise. A domain is a region in this space where it can be said that texts are similar; it is a bounded cluster of points in this variety space.

In the conventional NLP conception of domains, all that can be considered formally is the text by itself. One problem with this restricted way of thinking about domains is that two texts can appear to be very similar. By using the variety space definition, one can formalize the need for using two separate dictionaries to decode the terms found in them and process them accordingly.

6 | MAKING USE OF FORTUITOUS DATA

To enable interpretation, Plank[8] also argues for the value of "fortuitous" data associated with text that includes metadata and data from other sources which is usually ignored during NLP analyses. In maintenance, this data includes data extracted from other fields in the maintenance management system, such as cost or time spent, as well as information obtained from purchase systems, weather databases, and maintenance manuals. She claims that the pairing of fortuitous data with learning algorithms allows for rapid adaptation to new varieties of language. In particular, she argues for rapidly gathering annotated data and the increased use of unsupervised and weakly supervised methods.

While for many use-cases, a rules-based approach can handle the presence of zero, one or multiple labels on a single maintenance work order, this can challenge supervised learning approaches [28,30] and performance depends on the handling of class imbalance. Seale[44] handled the challenge of 1200 different component classes by injecting additional information relevant to the physical systems into the model training systems through "privileged information" which is a form of fortuitous data[8]. A common example of this knowledge in engineering is that components have natural hierarchical structuring and this taxonomic information can be used to identify correct and incorrect components. Another example is knowledge of the cause and event relationships to predict components involved in a failure or repair activity.

We believe that TLP can further develop the idea of fortuitous data by encouraging community development and use of shared computational resources. The creation and use of knowledge dictionaries, typified by ConceptNet[45], to improve the semantic processing of natural language and provide additional assistance in pre-processing the data, managing word tagging and/or any special rules have gained traction. Such dictionaries are reusable, developed/tuned as a data pre-processing step across the data and often use common NLP tools to assist in their creation [10,28,40]. Sexton et al.[40] developed an importance based vocabulary tagging system using term frequency–inverse document frequency (TF-IDF) weighting[27]. Gao et al.[10] used spellcheckers (`pyspeller`) and string distance (`fuzzywuzzy`) to support dictionary creation process for domain specific uses. Such dictionaries have helped manage misspellings and variations of the same terms in pre-processing for word representations[46,47]. POS tagging has also been customised, examples include the use of a modified version of the widely-used Penn Treebank Set and custom tags for domain-specific concepts[9,14] and context-relevant State-Activity-Item tags[10].

On the surface, these resources can help mitigate the lexical variations in technical text and simplify domain adaptation between similar technical domains by constraining terminological variation to the intrinsic differences found between facilities[21]. But from a deeper perspective, they provide a source of standardized fortuitous data; they represent shared knowledge that can be used to understand the latent variables associated with a domain and help define the proper context for their interpretation. This knowledge can also help with the comparison of domains and further foster the sharing and adaptation of analysis approaches.

7 | DOING MORE WITH LESS

Engineering researchers have started to widely use SOTA NLP approaches to mine text data[15,16]. However, there is also a tendency to gloss over its high computational costs[15]. For example, the article introducing GPT-3[5] does not mention its estimated cost of 355 graphics processing unit (GPU) years or $4.6 M (USD) [48].

Even smaller efforts can incur large costs; Strubell, Ganesh, and McCallum[49] examined the cost of a representative NLP research project: 27 GPU-years for training and tuning costing in excess of $100K (USD) for cloud compute time and $9870 for electricity.

How can we ensure accessibility to the benefits of NLP to those who can't afford large computing clusters? Can something be done to mitigate the computational requirements?

In some domains, crowd sourcing [50], the use of large numbers of people across a network performing an information processing task, has successfully added complex NLP [51] to an analysis with minimal computational cost. In technical domains, however, the data is often proprietary business information that cannot be shared outside of the organization. This dramatically limits the usefulness of crowd sourcing.

We will instead focus on TLP's engineering mindset that encourages discussions about the most practical approaches for achieving real-world goals. For example, Xu et al. [52] used neural word embeddings and convolutional neural networks (CNN) to perform text classification. Their CNN model took 14 hours to train. Following an approach that is congruent with TLP, Fu and Menzies [15] performed a replication study that used an optimizer to fine tune a traditional support vector machine (SVM) to achieve similar performance while decreasing training time by a factor of 84.

Subsequently, Majumder et al. [16] repeated the replication study using local learning via clustering the data prior to training an SVM on each cluster. They reported a 570× speed up on a single core and a 965× speed up eight cores relative to Xu et al. while achieving F1 score results within 2 %. While from an NLP perspective, the classification scores did not improve, the accessibility and usefulness did by mitigating the need for large computational resources while achieving useful results.

These results show the value of applying an engineering perspective to an application domain text analysis problem instead of solely using the current NLP state of the art. Because the NLP literature focuses on advancements along its frontiers, applied results, particularly those which address computational costs, are relegated to the literature of disparate domain-specific communities, such as software engineering. TLP allows for the aggregation and dissemination of these patterns of usage within its community.

8 | THE RISKS OF THE STATUS QUO

There are always risks which accompany the application of technology, tools and techniques to any domain, including maintenance text. Some of these risks are of a more practical nature. One central risk is lack of trust—due to missing, incomplete, and inconsistent information, practitioners do not trust their maintenance data, and by extension, do not trust the outputs from application of NLP to this data. Another risk is that many groups will likely develop ad-hoc solutions to particular issues and, in general, mistakes will be made and solutions will be re-invented many times. This re-iteration and reinvention effectively represents a tax on an entire industry, one that could be greatly reduced by shared conventions and standards.

A more pernicious set of risks can be articulated using a set of ethical concerns which were originally intended for the broad societal use of algorithms but they apply to this more focused use as well [53]. The concerns are **unjustified actions**, **inscrutable analyses**, and **systemic bias**; maintenance-based examples and their consequences are shown in Table 3.

Correlations emerge from the analysis of data and actions may be taken from these findings. When the causal link is unknown or not determined, the action may be unjustified as well as costly and ineffective. The use of correlation to guide actions is not without pitfalls; spurious correlations in the data coupled with the low-probability of finding legitimate regularities and the tendency of SOTA NLP to learn shallow heuristics can result in unjustified actions to address questionable concerns identified by mining text-based records. For maintenance, this is likely to be further exacerbated by the lexical noisiness due to variations in spelling, abbreviation, and punctuation found in the data.

Actions can be justified by examining the relationships between data and conclusions. Though often hard to discern, it is reasonable to expect that they are available for inspection. Such scrutiny can help decide between competing conclusions drawn from different analyses of the same data, or identify ungeneralizable conclusions drawn from accidental features of the data. Responsibility is an important component of engineering ethics and we see ability to analyze and justify technical actions as key to being able to accept meaningful responsibility.

With the large amounts of complex data and machine learning that are used by SOTA NLP, the rationale behind analysis results can easily be obscured inside of inscrutable algorithmic black boxes that impede human understanding and criticism. The results and implied courses of action then have to be accepted at face value with a lack of confidence. With competing analyses, a final course of action must then be determined by outside means driven by the personal biases of those left to make the decision.

It is well known that analyses follow the "garbage in, garbage out" principle and the quality of the results is heavily dependent on the quality of the data. However, analyses are also inherently biased by assumptions baked into tools and methodologies. These biases are often propagated into the conclusions. For maintenance, any tendency of the NLP analytics pipeline to overlook certain issues results in resources being instead allocated to other activities. Because of their dependence on large amounts of annotated training data, the use of popular SOTA NLP techniques in technical domains such as maintenance may cause a type of sampling bias; issues for which

TABLE 3: Examples of risks and consequences with NLP-based analyses in industrial maintenance

Risks	Example Situations	Possible Consequences
Unjustified Actions	*The failure mode causing the most unplanned downtime was not identified because of large variation in misspellings*	*Improvement initiatives were not focused on highest opportunity areas*
Inscrutable Analyses	*An analysis uses complex algorithms and large amounts of data that are hard to understand*	*Lack of confidence in analysis results; final course of action decided by other means*
Systemic Bias	*A company's maintenance analytics pipeline tends to overlook certain issues*	*Resources routinely allocated to other areas*

training data is readily available will lead to machine learning systems that can find them. Text preprocessing can also affect analysis results[54] and due to the large lexical and grammatical variations in maintenance text, the use of common, out-of-the-box NLP preprocessing techniques may not work well for this domain. This means that the apparent importance of certain maintenance-related issues could be systematically diminished or exaggerated because of the mismatch between the assumptions behind commonly-used NLP techniques and the requirements of maintenance.

With its emphasis on iterative, human-in-the-loop style analyses, TLP naturally fosters human understanding throughout the pipeline. By adapting NLP to the domain under investigation, we see increased opportunity for simpler and more understandable analyses. Tailoring earlier stages of the pipeline to the domain, such as preprocessing and parsing, allows important semantics earlier into the analysis to simplify the algorithms used later. For example, instead of using an opaque deep neural network to classify messy text data, a simpler, interpretable classifier can be used on the preprocessed and normalized text that facilitates understanding the rationale behind analyses, justifying the resulting actions, and finding hidden biases.

9 | SUMMARY

NLP has made significant progress in recent years towards achieving human-level performance on a variety of natural language tasks. Despite this, engineers and technical analysts seeking to use state-of-the-art NLP for real-world tasks face concerns that it may not live up to expectations, require more annotated training data than is available, be too complex to understand the rationale behind its analyses, require excessive computational resources, and inject biases into the final results.

We have proposed a human-centered, iterative approach to NLP, technical language processing (TLP) to address these issues for engineering domains. By focusing on the needs of engineering text analysis and not being driven to achieve human-level language performance, TLP practitioners are free to choose the most practical techniques to address the challenge at hand while achieving a thoughtful balance between raw analytical performance and the available resources. In place of the aesthetic of stringing together complex algorithmic black boxes and hoping for the desired outcome, TLP encourages human intervention to inject domain knowledge and meaning at each stage of the analysis as detailed in our previous paper[21]. This can help mitigate the accumulation of systemic technical bias in the final analysis. By adapting NLP to focus on the challenges of engineering text, TLP can bring the promise of text analysis to industry.

ACKNOWLEDGEMENT

Hodkiewicz acknowledges funding from the BHP Fellowship for Engineering for Remote Operations and the Australian Research Council Centre for Transforming Maintenance through Data Science (Industrial Transformation Research Program Grant No. IC180100030).

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. Bender EM, Koller A. Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data. In:

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics; 2020: 5185–5198.

2. Emerson G. What are the Goals of Distributional Semantics?. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,* Association for Computational Linguistics; 2020: 7436–7453.

3. Linzen T. How Can We Accelerate Progress Towards Human-like Linguistic Generalization?. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,* Association for Computational Linguistics; 2020: 5210–5217.

4. Heaven WD. OpenAI's new language generator GPT-3 is shockingly good—and completely mindless. https://www.technologyreview.com/2020/07/20/1005454/openai-machine-learning-language-generator-gpt-3-nlp/; 2020. Accessed: 2020-12-8.

5. Brown TB, Mann B, Ryder N, et al. Language models are few-shot learners. *arXiv:2005.14165* 2020.

6. McCoy T, Pavlick E, Linzen T. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*Association for Computational Linguistics; 2019; Florence, Italy: 3428–3448.

7. Glockner M, Shwartz V, Goldberg Y. Breaking NLI Systems with Sentences that Require Simple Lexical Inferences. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers),* Association for Computational Linguistics; 2018; Melbourne, Australia: 650–655.

8. Plank B. What to do about non-standard (or non-canonical) language in NLP. *Bochumer Linguistische Arbeitsberichte* 2016: 13-20.

9. McKenzie A, Matthews M, Goodman N, Bayoumi A. Information Extraction from Helicopter Maintenance Records as a Springboard for the Future of Maintenance Text Analysis. In: *Trends in Applied Intelligent Systems,* Springer Berlin Heidelberg; 2010: 590–600.

10. Gao Y, Woods C, Liu W, French T, Hodkiewicz M. Pipeline for machine reading of unstructured maintenance work order records. In: *Proceedings of the 30th European Safety and Reliability Conference and 15th Probabilistic Safety Assessment and Management Conference*; 2020.

11. Pan SJ, Yang Q. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 2009; 22(10): 1345–1359.

12. Ben-David S, Blitzer J, Crammer K, Kulesza A, Pereira F, Vaughan JW. A theory of learning from different domains. *Mach. Learn.* 2010; 79(1): 151–175.

13. Li Q. Literature survey: domain adaptation algorithms for natural language processing. tech. rep., Department of Computer Science, The Graduate Center, The City University of New York; 2012.

14. Bokinsky H, McKenzie A, Bayoumi A, et al. Application of Natural Language Processing Techniques to Marine V-22 Maintenance Data for Populating a CBM-Oriented Database. In: pdfs.semanticscholar.org; 2013.

15. Fu W, Menzies T. Easy over hard: a case study on deep learning. In: *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering,* ESEC/FSE 2017. Association for Computing Machinery; 2017; New York, NY, USA: 49–60

16. Majumder S, Balaji N, Brey K, Fu W, Menzies T. 500+ Times Faster than Deep Learning: (A Case Study Exploring Faster Methods for Text Mining StackOverflow). In: *2018 IEEE/ACM 15th International Conference on Mining Software Repositories (MSR)*; 2018: 554–563.

17. Russell A, Vinsel L. Innovation is overvalued. Maintenance often matters more. *Aeon* 2016; 12: 2020.

18. Hodkiewicz MR. Maintainer of the future. *Australian Journal of Multi-disciplinary Engineering* 2015; 11(2): 135–146.

19. Orr JE. *Talking about machines: An ethnography of a modern job*. Cornell University Press . 2016.

20. Woods C, Hodkiewicz M, French T. Requirements for Adaptive User Interfaces for Industrial Maintenance Procedures: A discussion of context, requirements and research opportunities. In: *Proceedings of the 31st Australian Conference on Human-Computer-Interaction*; 2019: 322–326.

21. Brundage MP, Sexton T, Hodkiewicz M, Dima A, Lukens S. Technical Language Processing: Unlocking Maintenance Knowledge. *Manufacturing Letters* 2020. doi: 10.1016/j.mfglet.2020.11.001

22. Malan F. Extracting Failure Modes from Unstructured, Natural Language Text. Master's thesis. Stellenbosch University. 2020.

23. Hodkiewicz MR, Batsioudis Z, Radomiljac T, Ho MT. Why autonomous assets are good for reliability–the impact of 'operator-related component'failures on heavy mobile equipment reliability. In: *Annual Conference of the Prognostics and Health Management Society*. 8. ; 2017.

24. Lewis DD. *Reuters-21578 text categorization test collection, Distribution 1.0 README*. David D. Lewis Consulting and Ornarose, Inc.; 2004.

25. Timonen M. Categorization of very short documents. In: *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* SciTePress; 2012: 5–16.

26. Lewis DD, Yang Y, Rose TG, Li F. RCV1: A New Benchmark Collection for Text Categorization Research. *J. Mach. Learn. Res.* 2004; 5(Apr): 361–397.

27. Jurafsky D, Martin J. *Speech and Language Processing*. Upper Saddle River, New Jersey: Prentice Hall . 2000.

28. Sexton T, Hodkiewicz M, Brundage MP, Smoker T. Benchmarking for keyword extraction methodologies in maintenance work orders. In: *PHM society conference*. 10. ; 2018.

29. Rajpathak D, Chougule R. A generic ontology development framework for data integration and decision support in a distributed environment. *International Journal of Computer Integrated Manufacturing* 2011; 24(2): 154–170.

30. Hodkiewicz M, Ho MTW. Cleaning historical maintenance work order data for reliability analysis. *Journal of Quality in Maintenance Engineering* 2016; 22(2): 146–163.

31. Rajpathak DG. An ontology based text mining system for knowledge discovery from the diagnosis data in the automotive domain. *Computers in Industry* 2013; 64(5): 565–580.

32. Harnad S. The symbol grounding problem. *Physica D* 1990; 42(1): 335–346.

33. Doumas LAA, Puebla G, Martin AE. Human-like generalization in a machine through predicate learning. *arXiv:1806.01709* 2018.

34. Yogatama D, Masson d'Autume dC, Connor J, et al. Learning and Evaluating General Linguistic Intelligence. *arXiv:1901.11373* 2019.

35. Augenstein I, Derczynski L, Bontcheva K. Generalisation in named entity recognition: A quantitative analysis. *Comput. Speech Lang.* 2017; 44: 61–83.

36. Gururangan S, Swayamdipta S, Levy O, Schwartz R, Bowman SR, Smith NA. Annotation Artifacts in Natural Language Inference Data. 2018.

37. Poliak A, Naradowsky J, Haldar A, Rudinger R, Van Durme B. Hypothesis Only Baselines in Natural Language Inference. *arXiv:1805.01042* 2018.

38. Geirhos R, Jacobsen JH, Michaelis C, et al. Shortcut Learning in Deep Neural Networks. *arXiv:2004.07780* 2020.

39. D'Amour A, Heller K, Moldovan D, et al. Underspecification presents challenges for credibility in modern machine learning. *arXiv:2011.03395* 2020.

40. Sexton T, Brundage MP, Hoffman M, Morris KC. Hybrid datafication of maintenance logs from AI-assisted human tags. In: *Proceedings of the 2017 IEEE International Conference on Big Data (BIGDATA)*; 2017.

41. Sexton TB, Brundage MP. Nestor: A Tool for Natural Language Annotation of Short Texts. *J. Res. NIST* 2019; 124.

42. 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: 193–198.

43. Calude CS, Longo G. The Deluge of Spurious Correlations in Big Data. *Found. Sci.* 2017; 22(3): 595–612.

44. Seale M, Hines A, Nabholz G, et al. Approaches for Using Machine Learning Algorithms with Large Label Sets for Rotorcraft Maintenance. In: *2019 IEEE Aerospace Conference*IEEE. ; 2019: 1–8.

45. Speer R, Chin J, Havasi C. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. *AAAI* 2017; 31(1).

46. Saetia K, Lukens S, Pijcke E, Hu X. Data-driven approach to equipment taxonomy classification. In: *Proceedings of the PHM Society Conference*; 2019.

47. Navinchandran M, Sharp ME, Brundage MP, Sexton TB. Studies to predict maintenance time duration and important factors from maintenance workorder data. In: *Proceedings of the Annual Conference of the PHM Society*. 11. ; 2019.

48. Li C, Balaban S, Balaban M. OpenAI's GPT-3 Language Model: A Technical Overview. https://lambdalabs.com/blog/demystifying-gpt-3/; 2020. Accessed: 2020-12-9.

49. Strubell E, Ganesh A, McCallum A. Energy and Policy Considerations for Deep Learning in NLP. *arXiv:1906.02243* 2019.

50. Estellés-Arolas E, Guevara G.-L.-dF. Towards an integrated crowdsourcing definition. *J. Inf. Sci. Eng.* 2012; 38(2): 189–200.

51. Callison-Burch C, Ungar L, Pavlick E. Crowdsourcing for NLP. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorial Abstracts*Association for Computational Linguistics; 2015; Denver, Colorado: 2–3.

52. Xu B, Ye D, Xing Z, Xia X, Chen G, Li S. Predicting semantically linkable knowledge in developer online forums via convolutional neural network. In: *2016 31st IEEE/ACM International Conference on Automated Software Engineering (ASE)*; 2016: 51–62.

53. Mittelstadt BD, Allo P, Taddeo M, Wachter S, Floridi L. The ethics of algorithms: Mapping the debate. *Big Data & Society* 2016; 3(2): 1–21.

54. Denny MJ, 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.

How to cite this article: Dima A., S. Lukens, M. Hodkiewicz, T. Sexton, and M. Brundage (2021), Adapting natural language processing for technical text, *Applied AI Letters*, *2021;00:1–6*.