Evaluating Word Representations in a Technical Language Processing Pipeline

Ajay Varma Nandyala[1], Sarah Lukens[2], Sundararam Rathod[3], and Pratiksha Agrawal[4]

[1,2,3,4] *GE Digital, San Ramon, CA, 94583-9130, USA*

ajayvarma.nandyala@ge.com
sarah.lukens@ge.com
sundararam.rathod@ge.com
pratiksha.agrawal@ge.com

ABSTRACT

The recent explosion of advancements in natural language processing (NLP) are encouraging in the industrial sector for leveraging the volumes of unstructured, technical data that currently sit unused. However, results from direct application of many NLP pipelines to technical text often fail to address the business needs of industrial companies. One requirement for satisfactory performance is an effective representation of the unstructured text in a form which contains the information required for an application task. We know of no standard methodology for evaluating word representations for technical text tailored to industry needs. In this paper, we propose guidance and methods for evaluating the performance of word representations for industrial use-cases.

1. INTRODUCTION

Recent advancements in natural language processing (NLP) have created possibilities and opportunities for utilizing the knowledge found in unstructured natural language data. State of the art (SOTA) performance continues to be achieved, now often through developments in language models based on Transformer architecture (Pathak, 2021). NLP advancements have moved so quickly in the field of mainstream artificial intelligence (AI) that a gap is emerging with respect to industrial business needs and the nature of the technical language data found in industry. Technical Language Processing (TLP) has been proposed to describe the iterative approach for tailoring NLP tools to technical data to meet the business needs of industry (Brundage, Sexton, Hodkiewicz, Dima, & Lukens, 2021). There is great opportunity to use advances from NLP to help industry utilize their unstructured data in their businesses to competitive advantage, but great care must be taken for companies to realize this value potential.

Industry is not the only field where such opportunities exist. Other domains, such as medical, legal, financial sectors, have been making advancements in the area of tailoring NLP tools to their domain and their domain specific challenges and applications (Wang, Liu, Afzal, Rastegar-Mojarad, Wang, Shen, Kingsbury, & Liu, 2018; Zhang, Chen, Yang, Lin, & Lu, 2019; Zhong, Xiao, Tu, Zhang, Liu, & Sun, 2020; Chalkidis, Fergadiotis, Malakasiotis, Aletras, & Androutsopoulos, 2020; Chen, Huang, & Chen, 2020). Domain adaptation refers to a class of approaches concerned with extending the learning from a source domain with abundant labeled data to a target domain with little or no annotated data (Ben-David, Blitzer, Crammer, Kulesza, Pereira, & Vaughan, 2010). We can look to success in other domains for guidance in the technical domain.

One of the key areas in advancement in NLP has been in data representation, which is the stage in the model pipeline where raw text is converted to a form that can be used by a numerical algorithm. In the past decade, approaches have been developed which go beyond the basic treating words as items in a bag to approaches which incorporate word similarities based on semantics and capturing different contexts as to how words are used in sentences. As such approaches have matured in sophistication, they have also grown in complexity and computational requirements, increasing the technical expertise needed to successfully wield such approaches in applications.

All of these factors lead to the question of how to determine which approach is the right approach for an industrial application. There are several factors to consider, but ultimately, the real decision is determining the trade-off between some form of quality of the model outputs, the costs associated with the execution of the model and the potential business value realized. In other words, model selection should come from determining if gains from technological

complexity are worth it relative to the price of execution. For industrial applications today, where using these technologies is still nascent, making such evaluations is critical.

An additional consideration is the ability to explain the behavior of the selected approach. It is well known within the mainstream AI community that as developments increase in complexity, the ability to explain the exact mechanisms which are contributing to SOTA performance are unclear (Kovaleva, Romanov, Rogers, & Rumshisky, 2019) and in fact, there is emerging evidence in which such SOTA models fail to generalize learned knowledge (D'Amour, Heller, Moldovan, Adlam, Alipanahi, Beutel, Chen, Deaton, Eisentein, Hoffman, Hormozdiari, Houlsby, Hou, Jerfel, Karthikesalingam, Lucic, Ma, Mclean, Mincu, Miatani, Montanari, Nado, Natarajan, Nielson, Osborne, Raman, Ramasamy, Sayres, Schrouff, Seneviratne, Sequeria, Suresh, Veitch, Vladymyrov, Wang, Webster, Yadlowsky, Yun, Zhai, & Sculley, 2020). Gaining the trust of the practitioner in the adoption of such analytics depends on the ability to explain model behavior, particularly as observed phenomena in industry is typically explained by principles of engineering and physics.

Further, when digging into the details of the different architectures and designs of different representation approaches, there are fundamental differences which make comparison between different approaches indirect unless some form of evaluation is performed against a downstream task. Which leads to another challenge today in industry, which is lack of accepted benchmarks for performance. Other communities, such as the biomedical field, have developed their own benchmarks for measuring performance and validating models (Gu, Tinn, Cheng, Lucas, Usuyama, Liu, Naumann, Gai, & Poon, 2020; Wang et al. 2018), but industry still has much work to get there.

The purpose of this paper to review performance evaluation methodologies for word representations and present how these approaches can be tailored for technical text. This paper serves as a review article, providing a survey of two areas: current work to date using technical language data for industrial use cases and current advancements in mainstream AI and domain adaptation, particularly from the biomedical field. The paper focuses on methods for assessing data representation approaches in the absence of benchmark performance metrics, and how such approaches can be used to compare the performance of different data representation approaches independent of a downstream task. We include a case study as an example of the methods reviewed here, and hope this organization of material will help guide industry towards developing performance benchmarks in TLP as well as make suggestions towards processes for analyzing behaviors which arise from using more complex word representations.

The rest of the article is organized as follows. Section 2 provides background on work that has been done to date in terms of TLP and industrial use-cases and work done by other domains to adapt mainstream AI tool and evaluate performance. Section 3 reviews some methodologies for evaluating different representation families, with emphasis on comparing different representation families and lack of industry benchmarks. Section 4 presents a case study illustrating the evaluation approach on an open-source dataset. The paper ends with concluding discussions and suggestions for the technical language processing community.

2. BACKGROUND

2.1. Technical language processing and industrial data

TLP formalizes tailoring advances in NLP to meet industrial business needs, providing tools and best practices which industrial companies can adopt in order to utilize their currently underutilized unstructured text in their business processes. For example, maintenance work orders (MWO) are one common source of unstructured text in industry which are typically used as the data source for any reliability analysis. Across the industry, the semi-structured fields in this data, such as failure codes, event types, etc. are often missing or inconsistently coded while the description fields often contain the richest information (Sexton, Brundage, Hoffman, and Morris 2017; Lukens, Naik, Saetia, & Hu, 2019; Hodkiewicz & Ho, 2016). However, new challenges emerge when trying to systemically parse this data. These text fields often contain misspellings, abbreviations along with technical complexities such as hierarchical structures or implicit meanings (Hodkiewicz, Kelly, Sikorska, & Gouws, 2006; Lukens, Naik, Hu, Doan, & Abado, 2017). TLP offers a path for developing and standardizing how these data sources are used in industry.

From the perspective of realizing business value, the most central component of TLP is that TLP is use-case driven. Alignment with business objectives and outcomes dictate the requirements, any modeling decisions made, and performance measures used. There has been encouraging work published in the literature successfully demonstrating how developing tools which utilize the free text fields can drive different industrial use cases, which we review in the remainder of this section.

Use cases based on MWO data are often for reliability applications. However, the source of MWO data is from Enterprise Asset Management (EAM) or Computerized Maintenance Management Systems (CMMS). The data is entered into the EAM/CMMS for maintenance purposes and not usually with the intent to use the data for reliability purposes. TLP offers an approach for bridging the gap between data quality challenges and the ability to leverage knowledge locked in descriptions for reliability purposes.

One reliability-centric use case which uses failure event and maintenance information from MWO data is opportunity

identification, where improvement initiatives for asset performance are identified, tracked and measured. Structured information on corrective work events can be used to quantity information about failure histories and maintenance work. Such structured information enables the consistent evaluation of reliability metrics (Gunay, Shen, & Yang 2019; Hodkiewicz et al. 2016), analyses requiring failure mode levels of granularity in the data such as Weibull analysis (Sexton, Hodkiewicz, Brundage & Smoker, 2018) and estimates of maintenance time durations by types of maintenance actions (Navinchandran, Sharp, Brundage, & Sexton, 2019). Applications for reliability decision making which use these calculated measures include system reliability simulations and Reliability-Availability-Maintainability (RAM) analysis (Seale, Hines, Nabholz, Ruvinsky, Eslinger, Rigoni, & Vega-Masionet, 2019; Lukens, Markham, Naik, & Laplante, 2019; Hodkiewicz, Bastioudis, Radomiljac, & Ho, 2017). Applications related to developing and measuring the effectiveness of a maintenance strategy include quantifying failure mode and effects analysis (FMEA) (Yang, Shen, Chen, & Gunay, 2018) and integration of data with Reliability Centered Maintenance (RCM) (Lukens & Markham, 2018; Sikorska & Hammond, 2007).

Related to reliability-centric applications are use-cases where knowledge databases are populated with the structured information extracted from the unstructured text fields. Rajpathak (2013) developed a diagnosis ontology for systematically organizing fault information from repair text. There has been work developing larger knowledge frameworks for industrial equipment in manufacturing where creating knowledge databases which integrate tools for structuring the unstructured data fields (Pereira, 2020; Mckenzie, Matthews, Goodman, & Bayoumi, 2010; Brundage, Kulvatunyou, Ademujimi, & Badarinath, 2017).

Another area of application is in real time suggestion systems, such as through prescriptive analytics and recommender systems. Systems have been proposed which utilized structured databases of failures and actions taken to make recommendations for maintenance actions in real time (Bastos, Lopes, & Pires, 2012; Bokinsky, Mckenzie, Bayoumi, & McCaslin, 2013; Federspiel & Villafana, 2003). Other proposed real-time suggestion systems estimate workload based on the nature of the failure mode (Usuga Cadavid, Grabot, Lamouri, Pellerin, & Fortin, 2020) and suggest where to route a work order based on information from the text (Bouabdallaoui, Lafhaj, Yim, Ducoulombier, & Bennadji, 2020). A reliability-centric real-time system which flags potential data quality errors during work order closure to ensure clean usable data in the first place was proposed by Gao, Woods, Liu, French, and Hodkiewicz (2020).

None of the use cases summarized above proposes to use technical language data in isolation, nor is the process of extracting structured information from the unstructured text the end goal. The end goal is always the decision-making enablement which arises from how the extracted knowledge can be integrated in industrial work processes. In general, model performance measures for TLP need to reflect the business goals of the final desired task.

2.2. Data representation and the NLP pipeline

The generic NLP pipeline can be summarized through two main components: the pre-processing pipeline and the text analysis pipeline. The preprocessing pipeline consists of the general steps for cleaning raw text, such as lowercasing, rules for handling special characters, numbers, etc. The text analysis portion of the pipeline are the steps concerned with converting preprocessed text to desired outputs (Brundage et al. 2021). The analysis task at the end is the ultimate desired use-case. For example, often the use case is machine learning classification, which requires labeled data for model training and validation.

Data representation is the portion of the pipeline concerned with converting raw text data into a form that can be used by the desired analytic algorithm. This could be for the straightforward case of converting the text into a form that matches the inputs needed to train a classifier (or layers in a neural network) or for merging text data features with non-text data (Geigle, Mei & Zhai, 2018). Representation is the area where many high profile advances in NLP have emerged such as embedding models such as Word2Vec and GloVe and attention- or transformer-based approaches such as BERT (Devlin, Chang, Lee, & Toutanova, 2018) and GPT-3.

The analysis task stage in the pipeline is where decision level information is ultimately extracted from the text. Often this is in the form of supervised learning through a classification problem but may also be in the form of named-entity recognition (NER), document summarization, information retrieval, machine translation, sentiment analysis and question and answering to name a few. This work is focused on reviewing ways for evaluating the data representation portion of the pipeline which are independent of the downstream task. Reviewing model behavior earlier in the pipeline can ultimately improve downstream performance. No matter what model is used, without an effective feature representation in any machine learning pipeline it is impossible to achieve optimal performance.

2.3. Looking to other domains for guidance

Other specific domains, such as biomedical, legal, finance have made considerable progress in tailoring NLP tools to their domain. Challenges with adapting mainstream NLP to biomedical applications have been long acknowledged by the biomedical community, such as insufficiency of pre-trained models from general domain corpora (such as news articles and web content) due to reasons such as domain-specific vocabulary, lack of biomedical domain knowledge in formal or consumer training datasets and abundance of unlabeled data (Zhang et al. 2019). Consequently, there exists publicly-

available biomedical datasets and performance benchmarks which the biomedical community is actively developing and releasing (see next section), and available pre-trained data representations such as BioWordVec (Zhang et al. 2019) and BioBERT (Lee, Yoon, Kim, Kim, Kim, Ho So, and Kang 2020).

Other domains have followed, such as "LegalAI" which focuses on tailoring tools from NLP to help legal tasks such as retrieval of relevant legal documents, matching similar cases and legal question answering (Zhong, Xiao, Tu, Zhang, Liu, & Sun , 2020), and leading to pre-trained models using data from different cases and legislation such as LEGAL-BERT (Chalkidis et al. 2020). In the finance domain, the lack of publicly available datasets has been identified as a major challenge for researchers working on NLP and finance (Chen et al. 2020).

2.4. Performance measurement and validation for word representations

Approaches for evaluating the performance of word representations can be categorized as qualitative and quantitative measures. Qualitative measures are "softer" in nature, but can be designed to provide insights into behaviors, and can also be formalized to be used in rigorous evaluations, such as through design of a checklist or a pass/fail screening process. For instance, qualitative tests can also be used as checkpoints in a pipeline with "necessary but not sufficient" criteria. If a choice made in preprocessing or word representation fails a particular qualitative test, it will not be sufficient for the downstream task and should be reviewed.

Quantitative measures are broken into intrinsic and extrinsic categories. Intrinsic evaluations evaluate system output in terms of predefined criteria about the desired purpose and function of the system itself, while extrinsic evaluations measure the impact of the word representations on the specific downstream tasks (Clark & Lappin, 2013). Quantitative performance measures, both intrinsic and extrinsic, are commonly evaluated against an industry benchmark.

For word embedding families, datasets containing measurements of the closeness of meaning of two words, or semantic relatedness are used, and the distances measuring term relatedness of the embedding is then compared against this. Publicly available datasets, where humans have manually ranked the relatedness, such as WordSim353 (Agirre, Alfonseca, Hall, Kravalova, Pasca, & Soroa, 2009), are often used for benchmarking these tasks. For domain-specific tasks, in the biomedical community, Wang et al. (Wang et al. 2018) was able to test word embeddings on four different published biomedical measurement datasets containing semantic similarity measures. In the absence of publicly available datasets, other approaches have been used for intrinsic quantitative assessments as well, such as using pseudo-labeling to create "ground truth" data (Khabiri, Gifford, Vinzamuri, Patel, and Mazzoleni, 2019).

Datasets commonly used for transformer- or attention-based families consist of sentence pairs where given the first sentence, the model will predict the second sentence. Two mainstream examples are SQuAD, which is a collection of question/answer pairs, and SWAG, which contains sentence-pair completion examples (Devlin et al. 2018). In the biomedical field, there is the Biomedical Language Understanding BLUE benchmark consisting of 5 tasks across 10 databases (Gu et al. 2020). An example of the creation of a publicly available question and answer pair dataset from the Health Sciences sector comes from a study which evaluated transformer-based approaches for answering questions about COVID-19. In this study, questions were pulled from a Kaggle competition and experienced medical experts reviewed the quality of different model responses. The dataset was then made publicly available by the authors (Oniani & Wang, 2020).

Currently, much of the literature evaluating performance of word representations in the technical domain focuses on extrinsic performance measurements on some supervised learning (classification task) with measurements such as accuracy, f1 scores, precision and recall. Performance comparisons of a supervised learning task is especially attractive when comparing different representation approaches, due to the differences in how context and semantic similarities are handled between different representations.

One challenge today in industry is that there is no benchmark labeled dataset to use for assessment based on classifier performance and many challenges to overcome to get there. There are many reasons behind this. For one, many companies view their data as proprietary so the sharing of data itself is challenged (Brundage et al. 2021). Secondly, there are additional extra challenges with supervised learning tasks using maintenance work order descriptions. The rest of this section will review supervised learning studies to-date to motivate the need for the creation and adoption of additional performance evaluations.

Accurate reliability calculations, such as for industry metrics such as "mean time between failure" or MTBF, require understanding which events were failure events (and often the failure mode information too). In many CMMS, there is a breakdown indicator field intended to indicate failure events but rarely used in practice (Lukens et al. 2017). A binary failure classification approach has been used to characterize if a work order description describes a failure or a non-failure event (Arif-Uz-Zaman, Cholette, Ma, & Karim, 2017; Lukens et al. 2018; Lukens et al. 2019). Such a classifier is useful to enable accurate (or at least consistent) reliability calculations. While useful for improving accuracy of many reliability metrics, one major hurdle to overcome in developing a golden dataset is through the need to revisit the

definition of failure, a very sensitive topic in industry. It is generally accepted in industry that a failure is "when an asset is unable to perform its required function" (SMRP, 2017), but there is much flexibility around what is meant by "required function". Two identical descriptions for two different assets may describe a failure in one case and a nonfailure in another, depending on asset-specific information. More discussion of this trade-off is found in Lukens and Markham (2018).

As discussed in Section 2.1, often information required for reliability analysis such as failure mode, maintainable item and action performed is missing or inconsistent. Such information is vital for many use-cases and well outlined in (Hodkiewicz et al. 2016) (Hodkiewicz et al. 2006). Often such failure event information can be extracted from the description and often it cannot. A single entry may have a variable number of relevant labels from a given description (zero, one or multiple). For example, in the zero relevant label case, Hodkiewicz and Ho (2016) reported between 1.9% to 18.3% of work orders missing this information in descriptions across five different organizations. Further, in the cases when the information can be extracted, the number of possible labels can quickly become unmanageable and class imbalance becomes a real challenge. Hodkiewicz and Ho (2016) reported 101 unique items, 74 unique actions and 21 unique failure modes for their use-case. Seale et al. (2019) reported over 1200 unique component labels in the training corpus! Success with such supervised learning models has been achieved through alignment to the business purpose, and often in practice, success in these situations has been achieved through rules-based approaches (Gunay et al. 2019; Sexton et al. 2018; Sexton et al. 2017).

In general, due to the complexity and sheer volume of different equipment types, components, parts and maintainable items and the complexities of their taxonomic nature, any classifier attempting to predict these labels from text description fields will face challenges. Seale et al. (2019) ultimately achieved high performance through developing an approach based on the hierarchical structure of the taxonomy, combining industry-specific expertise with the machine learning approach. Saetia, Lukens, Pijcke, and Hu (2019) developed a man-in-the-loop hybrid approach to handle the 110 labels from classifying equipment to a standard in the EAM/CMMS using the equipment short description field. Bouabdallaoui et al. (2020) had a different application; in this study, they were interested in routing the work order to the right department. Since the 107 labels in their data was not necessary for satisfactory performance with respect to their use case, domain experts were used to reduce the number to 10 possible labels (Bouabdallaoui et al. 2020).

Another classification application by Usuga Cadavid et al. (2020) was interested in predicting priority and workload duration from the work order text. To handle class imbalance, k-means clustering was used to create four categories for labeling the data. The decision to use this clustering approach was made with consideration of an end goal which as to enable performance metrics on a classification task with a manageable number of labels for comparing different modeling approaches.

3. PERFORMANCE CHECKS FOR WORD REPRESENTATION

Figure 1 shows the NLP pipeline with emphasis on performance checks for representation families. The blue squares denote the main pipeline steps considered in this work (preprocessing, representation and analysis task) and the yellow hexagons show areas where performance checks can be integrated. The gray cylinders show where industry benchmarks ideally could be used (if available) for performance evaluation. The input to the pipeline is the raw text data from its initial source. For benchmarking model performance, the test data source may come from an industry provided standard dataset. Currently there are a couple available datasets for maintenance work orders (Reviewed in (Brundage et al. 2021)). Text corpora characterization refers to basic descriptive statistics on the corpus, such as number of documents, size of the vocabulary, number of words per document, etc. We place these characterization checks before and after the pre-processing step as a way of understanding the impacts pre-processing had on the data before input to the data representation step.

After data representation, there are a couple of performance checks before setting up the analysis task. The first area of checks are the qualitative checks, which act to both give insight on explaining the representation and may be used as pass/fail criteria to help either modify components in the pre-processing or representation stages. Intrinsic quantitative assessments apply additional levels of rigor for benchmarking performance independent of the downstream task. Ideally, there may be datasets or information that can be used here as industry benchmarks, but other approaches may be used as well. The next step is the analysis task, which is commonly supervised learning but may be a different task such as summarization or question/answering. In terms of typical model benchmarks, the analysis task is usually supervised learning and extrinsic quantitative performance measures are measures such as accuracy, specificity and recall.

An additional set of measurements across the full pipeline are some indicator of computational resources and requirements. The level of resources needed may depend on the size of the dataset, data connectivity factors, the technological requirements of the techniques used, the number of people involved and their levels of training needed and the business needs of the desired use case. There is always a trade-off to consider. During the model design phase, the balance between performance gains for achieving a specific task should be weighed against the additional computational costs and additional overhead needed to achieve such gains.

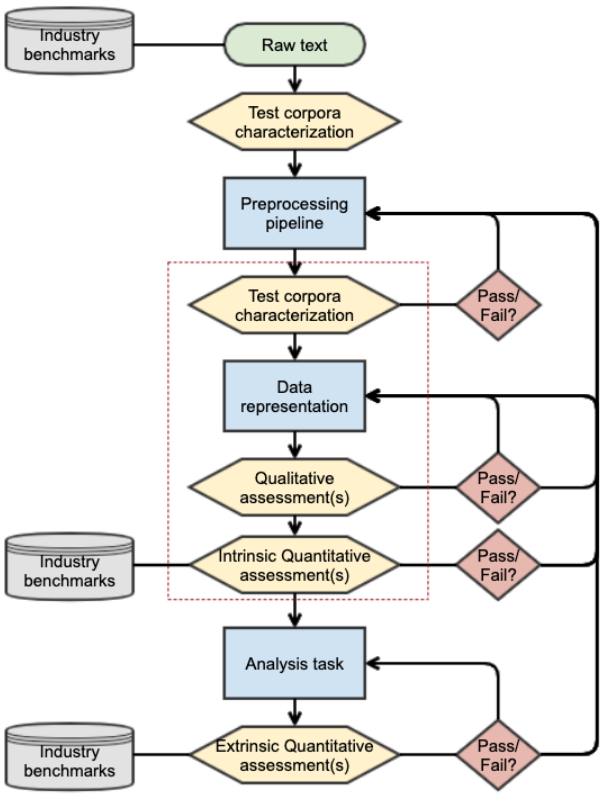

Figure 1. TLP pipeline with Pass/Fail evaluations steps as described in this paper. The red dashed box marks the pipeline steps which are the scope of this work. Explicitly missing from this picture is an evaluation of computational resources in order to weight any trade-offs in the context of the use-case.

3.1. Text corpora characterization and pre-processing

Text corpora characterization refers to basic descriptive statistics characterizing the corpus. Comparisons of such characterizations before and after the preprocessing stage of the pipeline can provide insights which may impact decision making around designing and developing the word representation approach. For example, the number of words per document may impact the size of the window for training a word embedding, as the average document length is commonly used as an initial window size for model training. Another example is the distribution of rare words. Different components of preprocessing such as stop word removal, stemming, lemmatization, special character or number removal or replacement and use of knowledge dictionaries or spell checkers may change the vocabulary distribution. If a corpus contains a large amount of spelling errors, there can be a long tail of rare terms which would be significantly reduced if replaced by correct spellings. If a pre-trained representation family is used this will additionally impact the number of words that are out-of-vocabulary, or OOV.

3.2. Data representation

In this section, different data representation families are summarized in the context of what elements of words and language are captured and different ways to analyze and measure performance. We focus our paper on three families: Bag of words (BOW)-based, embeddings-based, and language models which are attention- or transformer-based. There are other families such as RNN-based approach such as ELMo (Peters, Neumann, Iyyer, Gardner, Clark, Lee, & Zettlemoyer, 2018) which are not covered in this review. It is important to note that each family has inherent differences, based on how they are constructed, and as a result, measurement and assessment techniques are unique to the family. Despite these differences, there are still ways to interpret and understand behaviors between the different approaches.

The BOW-based representation family refers to a class of representations based on tokenization of the pre-processed text. A bag-of-words, or frequency vector, is the sum of all one-hot vectors which contain frequency but not ordering of words in a corpus. Considerably less storage is needed for frequency vectors than one-hot vectors, and normalization approaches such as TFIDF are often used to handle numerical challenges. However, in a BOW-based representation, word ordering, context and semantic similarity are not captured. For example, the words “valve” and “vlv”, which are contextually the same will be treated as separate features, unless captured in preprocessing.

Word vectors were designed to capture the influence neighboring words may have on the meaning of a word. Word embeddings-based representations represent each word as a dense vector or embedding. Word vectors are dense vectors and numerical vector representations of word semantics, capturing neighboring words (specified window size, usually <10 tokens).

Information, such as semantic similarity with other words in the vocabulary, is encoded in each word vector. As a result, vector operations can be performed for measuring the similarity between two words even if they are not exact matches. Words like “valve” and “vlv” would be expected to be close to each other in such a vector space, as they would be expected to have similar neighboring words in a training data corpus. On the other hand, these representations are context independent. Context independent means that one word will have one vector, independent of its context. For example, the word “pressure” will be represented by one vector despite the terms “pressure vessel”, “high pressure” and “pressure transmitter” having different contexts.

Word embeddings require a pre-training step, which adds additional complexity to the BOW based approach. Pre-training also enables use of additional data beyond the data in the task corpus, resulting in the possible convenience of out-of-the-box pre-trained embeddings. Word embeddings are

used with a supervised learning classifier in the form of a shallow layer where the weights are what is determined in the pre-training processes. The neural network weights are trained through setting up neighborhoods of surrounding words. The result of the training exercise is a dense word vector of a specified dimension for each word in the corpus. Skip grams and continuous bag of words (CBOW) are two common ways of setting up the system to train. Skip grams are trained as each "word" is the model input, and the target are the words around it, while CBOW uses the surrounding words as model input and each "word" is the target. The "accuracy" in training these models is measured against predicting surrounding words for a given word.

Attention- or transformer-based representations are another family of word representations based on transformer architecture, or attention. In attention- or transformer-based models, such as bidirectional encoder representations from transformers (BERT), word positions are represented by positional encodings. Which means that in addition to accounting for the contribution of neighboring words like word embeddings do, BERT additionally takes word order and word position into account.

BERT models are trained through two semi-supervised tasks; Masked language modeling (MLM) and next sentence prediction (NSP) (Devlin et al. 2018). At a very high level, MLM models mask a certain percentage of words (eg: 15%) and the model is trained to predict the missing words. NSP models create a binary label training set by taking sentences from a corpus and creating input data of sentence pairs, where half the time the second sentence truly follows the first and half the time it does not.

Like word embeddings, language models are pre-trained. However, unlike embedding models, semantic similarity is captured through the similarity between two sentences. Due to these differences, many of the qualitative approaches which makes sense for analyzing embedding models (such as reviewing similar words by semantic similarity) are not straightforward when analyzing language models and vice-versa. And measurements of semantic similarity which may make sense for embedding or language models may not make sense for BOW based.

3.3. Qualitative assessments

In this paper, three different areas or approaches for qualitative assessment are covered. Additional assessments specific to individual representation families do exist, but the three reviewed here were selected because some type of comparison between different families can be made in each of these cases (even if not perfect). We believe this approach will be helpful when selecting a general representation approach for a given task and promote understanding and explanations for observed results.

1. **Word similarity** – evaluate similarity between two different words.
2. **Sentence similarity** – evaluate similarity between two different sentences.
3. **Word clustering** – understanding patterns for how different words are related or similar.

Word similarity. Measuring the similarity between two words is most naturally associated with analysis on word embedding models. For a word embedding representation, calculating the cosine similarity between two-word vectors returns a measure of their distance in vector space. A common and useful qualitative check used for word embeddings is to compare the closest matches (say 5-10 words) for a sample of relevant of words across different representations. In Levy & Goldberg (2014), this proposed approach gave insight on their proposed embedding approach using parts of speech compared to classic skip gram trained word2vec. This approach was further utilized by Wang et al. (2018) for the biomedical domain to interpret the results of embeddings trained on news and web content compared to embeddings trained on domain specific content. This approach was also used in the industrial domain with the words "steam" and "oil" in Khabiri et al. (2019)

However, word similarity does not make direct sense when evaluating a BOW-based or a transformer/attention-based representation. In the BOW-based, word similarity is out-of-scope of the modeling assumptions, so insights here are out-of-scope. However, in the BERT case, understanding related-ness of different words may provide insights, provided your test is well-defined and understand. The major factor to understand is the context dependence of BERT. The word "pressure" in the sentence "under pressure to fix pressure vessel with low pressure readings on the xmtr" will return different "similar words" depending on what sort of test you do.

To look at some tests, we look at what some practitioners are proposing. For example, one straightforward proposed test uses the MLM trained feature of BERT to understand how a model performs with domain- (or corpus-) specific words. Such a test consists of inputting a simple sentence of the form "[domain word] is a [MASK]" and reviewing the model outputs. Such a test may be helpful for evaluating the capabilities of a pre-trained model on a domain just by seeing what sorts of words it returns and how BERT handles the original word (if OOV, it will split it up) (Rajasekharan, 2020).

Another approach used by practitioners is use k-nearest neighbors to approximate similarity between words (Khandelwal, Levy, Jurafsky, Zettlemoyer, & Lewis, 2019; Rajani, Krause, Yin, Niu, Socher, & Xiong, 2020). Similar words have been used to qualitatively provide insights in the biomedical field for comparing BioBERT with clinical

BERT (Alsentzer, Murphy, Boag, Weng, Jin, Naumann, & McDermott, 2019).

Sentence similarity. Measuring the similarity between two sentences is more natural for both BOW-based and transformer/attention-based representation families, which use NSP in their training. For BOW-based representations, similarity between two documents (in the corpus) can be measured by taking the dot product, or cosine similarity, which will indicate some similarity of words. For word embedding-based representations, measuring the similarity between two sentences needs additional extra work, and again, may be useful if what you are doing and how you are doing it is well defined.

In general, approaches for measuring sentence similarity in word embeddings look at the similarity of the different word vectors in the sentence. The Word Mover's Distance (WMD), measures the similarity between two sentences as the shortest distance they need to travel in word embedding space (Kusner, Sun, Kolkin, and Weinberger, 2015), and Sum and Mean of word embeddings (SOWE and MOWE) measures have also been used for sentence similarity. There are a few challenges using these approaches as the contextual and semantic meaning of the individual words may be lost. Further, the different lengths of documents or sentences have poor effect on such operations. TFIDF weights can also be used in a weighted average for handling the variability in the length of documents or sentences, but do not solve all of the problems in capturing semantic and contextual meaning. (Choudhary, 2020)

Word clustering. Word clustering is another useful qualitative approach which can show the relationships of different words. For word embedding-based families, this type of approach is straightforward. The high dimension word vectors can be projected to a two- or three-dimension space for a person to review. The ability to view and interact with domain specific terms in a visual vector can provide insights at how the trained embeddings are handling semantic similarities across a broad number of words at once.

As in the similar words case, such an exercise is not straightforward for the transformer/attention-based approach. One approach is a visualization of how the same word can mean different things. In this approach, sentences where the same word is used in different contexts are selected, the word is masked and then the BERT suggested words are projected in two dimensions (Lucy & Bamman, 2021; Wiedemann, Remus, Chawla, & Biemann, 2019). Such visualizations have been insightful for understanding how the attention/transformer-based representation handles polysemous words.

3.4. Evaluation of required resources

In addition to performing various qualitative and quantitative performance checks, it is important to also evaluate the resources needed for a defined task. In industry, unlimited computational resources and personnel may not be available, so understanding the different resources needed are important measurements for completeness. At high level, a checklist of the different areas to consider in addition to the model performance is presented in Table **1**.

Table 1 Overview of different resource requirements to consider in additional to model performance when selecting a representation approach

Category	Considerations
Technology	Computational requirements to train/deploy model
	Model training resources needed such as size to store, time to train, time to deploy, etc.
Data	Size of data, data storage, frequency of data updating etc.
	Data connectivity - how does data connect to model? How does model write back to data?
	Data quality considerations
People	Who are the stakeholders needed? What are the skillsets needed for the different stakeholders?
Processes	How will this model be used in industrial work processes?
Business drivers	Use-case consideration and business case justification

4. CASE STUDY: ASSESSING DIFFERENT REPRESENTATIONS ON MAINTENANCE DATA

In the case study, we walk through the different steps of the workflow for evaluating word representations (Figure 1). The case study illustrates how the different evaluation approaches can be used to examine how data representations handle technical language in maintenance work order descriptions.

4.1. Description of the data used

The data used for this case study is an open source dataset describing 5,485 work orders for 5 excavators (The Prognostics Data Library, 2017). A sample of the dataset is shown in Table 2. Due to its public availability, this dataset has been used in several studies already (Yang, Baraldi, & Zio, 2020; Sexton & Fuge, 2020; Sexton et al. 2018; Sexton & Fuge, 2019; Hodkiewicz et al. 2016). In the study, we compare the performance of different representation approaches from each of the families and also by using pre-trained and publicly available representations for comparison.

Table 2 Sample of excavator dataset, made publicly available by the UWA Prognostics Data Library

Date	Asset ID	Original Short text	Cost ($)
6/16/06	C	C/OUT RH NO-3 TRACK ROLLER-FAILED	17719.07
07/04/11	C	Repair hyd oil leak	2609
1/23/11	C	OIL LEAK ON BOOM PIPING	1317.13
1/20/05	D	Replace LH turbo	2212.87
9/24/08	C	RECTIFY ELECTRICAL FAULT (LOW PRESSURE)	115.9

4.2. Pre-processing

Minimal pre-processing is done to promote reproducibility of this work and to maintain focus on data representation. The details of the pre-processing steps taken are as follows: (1) data is converted to lowercase by replacing all upper-case alphabets to their corresponding lowercase alphabets. (2) Regex is used to remove all numerical and punctuation characters from the input data. (3) Whitespaces are removed from data which are already present or may have been created by removal of punctuation.

4.3. Word representations

We compare the performance of different word representations across the three main families considered (TFIDF, BOW and attention/transformer) with combinations of out-of-the-box libraries, pre-trained models and models we trained ourselves. The representations used are summarized in Table 3. For representations using the BOW family, the TFIDFVectorizer() function from the python sklearn library (Pedregosa, Varoquax, Gramfort, Michel, Thirion, Grisel, Blondel, Muller, Nothman, Louppe, Prettenhofer, Weiss, Dubourg, Vanderplas, Passos, Cournapeau, Brucher, Perrot, & Duchesnay, 2011) was used with all of the default parameter settings. The default settings include only using 1-grams.

For representations in the word embeddings, we compare both pre-trained out-of-the-box models as well as training our own custom embedding on the excavator dataset. For the pre-trained models, we used two pre-trained word embedding models in the gensim library (Rehurek & Sojka, 2010), the Word2Vec model trained on Google News (Google-News-300) and the GloVe model trained on Wikipedia 2014 + Gigaword5 (Pennington, Socher, & Manning, 2014). The vector dimensions of 100 for GloVe-100 and 300 for Google News where chosen as only 300 is publicly available for Google News.

For our custom trained word embeddings, we followed the Word2Vec approach with the skip-gram architecture. In the skip-gram architecture, the input word is combined with the context words (words surrounding the input word) to create the training dataset and train the model. We chose 100 as the vector dimension and trained it on the excavator dataset which contained 1965 words of which many were domain specific and hence wanted to capture the fine differences. We set the window as 3 and negative sampling parameter to 5. These parameters were set based on the average length of sentences.

For attention-/transformer-based representations, we used out-of-the-box BERT (Devlin et al. 2018). We selected the BERT large uncased which is of 24-layer, 16 attention heads, 1024 hidden dimensions, and 336M Parameters. For sentence semantic similarity, we used Sentence-BERT (SBERT) (Reimers & Gurevych, 2019) as BERT requires massive computational resources. SBERT is a modification of BERT which uses Siamese network to generate sentence embeddings which can be compared using cosine. We selected BERT base model bert-base-nli-mean-tokens for Sentence Embeddings.

Table 3 Summary of the different representations used in the case study.

Case name	Representation Family	Description
TFIDF	BOW-based	Fit on excavator dataset
GloVe-100	Embeddings-based	Pre-trained GloVe-100, trained on Wikipedia 2015 + Gigaword5
Google-News-300	Embeddings-based	Pre-trained Word2Vec model, trained on Google News
Custom Word2Vec	Embeddings-based	Custom trained Word2Vec on the excavator dataset
BERT	Attention-/transformer-based	Pre-trained BERT large uncased

4.4. Text corpora characterization

Characterizations of the Excavator dataset before and after the pre-processing steps described in 4.2 are reported in Table 4. Reporting the measurements before and after pre-processing returns some indication of how pre-processing steps may have affected the corpus. For our simple example, a reduction in vocabulary size is mostly due to the mix of work orders all in capitals and those mostly in lowercase in the original document. Due to the left-skew of vocabulary word frequency (a handful of words are typically used most of the time), the measure “Count of words in 80% of the total word usage” is proposed to give indication of the skew based on the Pareto 80-20 principle. This measure sorts the words by their frequencies in the corpus and counts the number of words that make up 80% of the total usage frequency (sum of all word frequencies). For the Excavator dataset, after pre-

processing, we observe that while there are 1935 distinct words, 237 of them (12%) are used 80% of the time.

Table 4 Text corpora characterization for Excavator dataset, before and after pre-processing steps

Measure	**Before Pre-processing**	**After Pre-processing**
Number of observations	5485	5485
Vocabulary size (distinct word count)	3237	1935
Count of words 80% of the total word usage (%)	563 (17%)	237 (12%)
Mean word count per document	4.92	4.62
Standard deviation word count per document	1.61	1.4

4.5. Qualitative – word similarity

The first qualitative check is word similarity. In order to illustrate both how different word "similarities" can be computed and how the different representation families need to be treated differently, we select a combination of different representations and approaches and present a sample of the results in Table 5. In Table 5, we list 9 different domain specific words (all present in the Excavator dataset) across 4 different categories. The three categories of Item, Problem/State and Action are commonly used to classify maintenance work orders and we add an additional category for called "Service" for domain words such as grease and oil (Sexton et al. 2017; Gao et al. 2020).

It is important to note that comparing the words returned between the different word similarity approaches is not a direct comparison. Rather, it is more of a comparison/understanding for how the different models handle domain specific words. For the embedding-based approaches, it is related to different measures of semantic similarity but for BERT these may be different. In all cases, we did not report any words that had three or less characters.

For the word embedding models, cosine similarity was used to identify the top five similar words. We reported the most similar words from the pre-trained GloVe-100 against the context embeddings from our custom trained embeddings on the excavator dataset. In Word2Vec, there are two approaches to find similarity between words; find words which have the same words surrounding each of the words (target embeddings) or find the context of each of the words (context embeddings). For reporting the similarity of words in our custom Word2Vec, we chose to go with context embeddings as we wanted to understand if the model was able to capture the context of the target word correctly and further could help us to find the different states/actions/services that support the target word and can be used for downstream analytic. Note the returned pool of possible similar words from the custom Word2Vec are limited to the excavator corpus, while the pre-trained models (GloVe-100 and BERT) can return words from the pre-training corpus.

In order to compare against BERT representations, we reported the top words by two different measures. The first measure, "fill-in-the-blank" is based on filling in the blank from a very simple sentence, based on the MLP training step intended to understand how BERT handles domain specific vocabulary (Rajasekharan, 2020). For each word, one of two possible simple sentences is used. If the word is an Action word, such as "replace", we entered "[Word] the [MASK] (eg: "Replace the _____"). For the other categories, we used "[Word] is a [MASK]" (eg: "Bucket is a _____"). We then reported the top five returns for the blank mask.

The second approach used to compare similar words is the "words as a sentence" approach in which we treated each word in the document as its own "sentence". While the contextual information of words is lost in this approach, the "one-word sentences" returned come the pre-trained BERT alone. The results are interesting to browse, as shown in Table 5.

Reviewing the results in Table 5 we observe that pre-trained models often find similar words which are out of domain. For example, "instrument" is often treated as a musical instrument rather than a category of equipment. "Word as sentence" BERT turns out to be very good in some cases at finding similar words by how they are spelled as well as context. But in the case of "grease", more of the words matched seem to be related to cooking instead of industry. This sort of analysis is very revealing towards the difficulties in explaining model behavior as well as possible complications that could arise from using out-of-the-box pre-trained models in industrial use-cases.

4.5 Qualitative – sentence similarity

The second qualitative check is through evaluating similar sentences. As mentioned in the previous section, such an approach is more natural for BOW-based and Attention-/transformer-based representations than word embeddings. For the TFIDF representation from the BOW-based family, the distance between two sentences was measured through using the cosine distance between two document vectors. For BERT, the distance was measured through cosine distance using SBERT. For the word embeddings, sum and mean of word embeddings (SOWE and MOWE) were used with TFIDF weighting cosine distance as similarity measure. One interesting consideration for the sentence similarity analysis over the similar word analysis is that the answers are restricted to sentences in the corpus, while in the similar word study the answers can be any word used in pre-training the representation. The results showing the most similar returned sentences for 5 sample sentences are reported in Table 6.

Table 5 Selected domain specific words from Excavator dataset from 4 different categories relevant to maintenance work orders and a comparison of how the approaches domain specific word through the top 5 most "similar" words returned by method, across 4 different representation approaches & method combinations.

Semantic Category	**Root word**	**Similar Words with GloVe-100**	**Context Words with Custom Word2Vec**	**Fill-in-the-blank with BERT (Masked)**	**"Words as sentence" similarity with BERT**
Action	replace	replacement, successor, succeed, installed, newly	teeth, locking, handle, bucket, shrouds	function, constant, point, break, matrix	replacing, replaces, replaced, replacement, substituted
	repair	maintenance, damaged, rebuild, upgrade, rebuilding	rail, bent, commercial, damaged, return	trade, condition, change, process, function	repairing, repaired, fixing, restore, heal
Item	bucket	bag, tub, toilet, laundry, pail	lines, broken, spare, busted, grease	place, tree, person, whole, bottle	pump, barrel, jug, pile, sack
	motor	automobile, auto, electric, toyota, honda	starter, slew, cooling, leaking, blocking	force, type, car, key, power	motorized, propulsion, engine, wheel, torque
	instrument	keyboard, percussion, sound, acoustic, piano	cluster, globes, inoperator, place, blower	key, guitar, piano, notation, note	item, device, mechanism, components, instrumentation
Service	grease	roasting, lard, baking, butter, coated	line, steel, low, pipe, text	color, remix, weapon, solvent, food	apron, steaming, sauce, skinned, patches
	oil	petroleum, crude, gas, fuel, energy, supply	cuting, swivel, mtr, area, propel	solvent, gas, fuel, resource, fluid	petroleum, tanker, fuels, gas, pipeline
Problem	leak	spill, leakage, bp, contamination, explosion	massive, major, bad, relief, oil	person, place, break, hole, bridge	Leaking, compromised, defects, contaminated, error
	broken	breaking, apart, cracked, fractured, neck	lines, line, grease, clamp, block	child, break, song, key, whole	shattered, fractured, damaged, cracked, wrecked

Table 6 Selected possible maintenance work order short descriptions and comparison of how different approaches match the description

Sentence	**TFIDF**	**GloVe-100**	**Custom Word2Vec**	**BERT**
replace missing tip	replace tip	replace missing tip adaptor	replace missing tip adaptor	replace missing tip adaptor
rebuild grease pump	rebuild lube pump	grease pump leaking	rebuild lube pump	install grease pump
blown grease line on bucket	blown grease line	grease line on bucket broken	replace grease line on bucket	blown grease line at pump
replace tooth on shovel	replace tooth on bucket	replace tooth on bucket	replace tooth on bucket	replace tooth on bucket
instrument tower panel broken	Instrument panel lights us	metal found in hydraulic screens	broken tooth	filter housing mount bolts broken

As expected, in many cases (such as "replace missing tip"), all the approaches were consistent in returning straightforward similar sentences. For the sentence "replace tooth on shovel", all approaches returned "replace tooth on bucket". For "rebuild grease pump", the pre-trained models captured the relationships between "grease" and "pump", while "rebuild" and "pump" was the higher focus on both the custom trained embedding model and TFIDF. The sentence "instrument tower panel broken" returned interesting responses as the word "instrument" was not frequently used in the corpus, but the concept of "broken" was. TFIDF was the only representation which returned a sentence with "instrument", possibly due to higher emphasis on the relationship between word and documents than the other models.

4.6. Qualitative –evaluating word clusters

The qualitative evaluation of how different words are related from the model is an exploratory exercise that can provide a lot of insight on model behavior. Due to the high number of words in a corpus and interest mainly in technical term relationships, we first identify a meaningful subset of words. We extracted 349 technical terms from the excavator dataset selected by ordering words by their TFIDF weighting and selecting 349 full words which could be seen as domain-specific (eg: words such as "replace", "valve", "exhaust"). Words which were acronyms or possibly generic (such as "in", "to", "and") were omitted.

When using pre-trained models for representations, it is useful to measure the amount of "coverage" between the words in the corpus used for developing the use case and the corpus for the pre-trained. For this case, the pre-trained Google-News-300 contained 329 out of 349 words (94% coverage) and GloVe-100 contained 328 (94% coverage). An interesting subsequent qualitative analysis is reviewing which words are included and which are excluded. The 21 words not present in the GloVe-100 were mostly misspellings or domain specific words such as "prelube", "gearcase", "pressurizer", Similarly, Google-News-300 did not contain domain words like "adaptors". The high quantity of misspellings such as "comditioning" or abbreviations such as "overtemp" indicate that creating dictionaries to handle such words in the pre-processing stage may be beneficial.

We show how the word embeddings are visualized for the 349 technical terms in a two-dimensional plot using t-distributed stochastic neighbor embedding (t-SNE) in Figure 2. The entire word cloud is very noisy, so illustrative subsets were selected for visualization. For GloVe-100 (Figure 2 (a)) we see two trends. First, as would be expected, words in similar contexts are clustered together such as "adjust", "modify", "install", "upgrade". Not only are these words all descriptors of maintenance actions, but they are maintenance actions which are typically associated with improvement work (as compared to maintenance actions which are typically preventative or corrective). The other trend is that similar versions of similar words tend to cluster, such as "replace", "replacing" and "replaced".

For the custom trained model, the target embeddings were used (Figure 2 (b)). Many of the relationships observed were not as straightforward as the GloVe-100 case, which could possibly be due to the very small amount of data used to build this model. Compare the 400,000 words used to train GloVe-100 with the 1935 words used to train the custom Word2vec. The small dataset may not be sufficient enough to capture many complexities. For satisfactory performance for pre-trained models, a suitably sized training dataset is necessary.

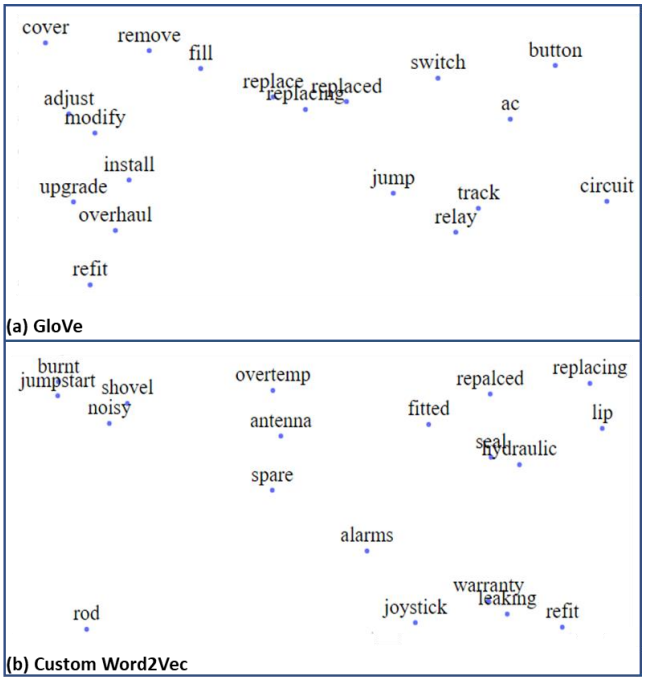

Figure 2 Examples of word clusters in the visualization of word embeddings using t-SNE. (a) Pre-trained GloVe-100 and (b) custom trained Word2Vec on the Excavator dataset which was trained on considerably a smaller corpus. Observe how similar words and similar word contexts are more related in the GloVe-100 representation.

4.7. Resources used in the case study

In addition to the results observed in this case study, it is important to explicitly consider the resources needed. The computational resources needed to use the pre-trained models or to train the word embeddings with the excavator dataset are not massive and could be done on a personal laptop. However, even in this simple use case it is important to note out that such a laptop has configuration requirements for running python and different python packages, which may be the case for a data scientist but not necessarily someone like a reliability engineer or reliability manager. On the other hand, many domain specific interpretations (such as words

and concepts specifically related to excavators) rely on a domain expert.

5. DISCUSSION

In this paper, different approaches for the qualitative performance evaluation of word representations in NLP were reviewed specifically for industrial use-cases for general downstream tasks. A case study using a publicly available maintenance dataset was used to illustrate the different approaches. Some key points emerged from this simple organization of information such as the importance of text pre-processing before data representation. The abundance of misspellings and variations in domain-specific spellings can make the data noisy and ultimately cause performance challenges in the downstream check. Another key point is the need to explicitly consider the trade-off between model performance and available resources when determining how to represent words for an industrial use case.

In terms of pre-processing, the study here used a very simplistic approach. There have been more in-depth promising works in this area, such as a pre-processing pipeline presented by Gao et al. (2020) which considers different entities/tags and uses an out-of-box spellchecker on common English words in order to focus on domain specific words. For word embeddings, we saw how visualizing words in a cluster can be used to evaluate how different misspelled words may appear. In fact, this insight may explain some motivation behind the work of Hansen, Coleman, Zhang, and Seale (2020), who used a word embeddings approach to assist in tagging data for document classification.

In terms of publicly available resources for the community, in addition to the need for public datasets, we have identified and motivated the need for creating datasets around word relatedness pairs or sentence pairs in ways that make sense with short descriptions (since often technical language is not in formal multi-sentence form). There is need in the industry for performance benchmarks. Such work could be built on top of public datasets and would be more realistic to develop and release over a benchmark labeled dataset for supervised learning.

The scenarios in the case study were designed to illustrate the reviewed qualitative methods in a reproducible way. By design, the simplest collection of cases needed to achieve the comparison were selected. This explains that while we elected to custom train a Word2Vec model, we did not elect to train, fine-tune or continued pre-train a BERT model. For the custom Word2Vec, this study observed that the size of the dataset (~5000 observations with ~2000 tokens) was probably too small to adequately train a word embedding model in application. As an extension, it may be overkill to train a BERT model which has over 1 million parameters with this dataset. There has been work exploring and comparing performance for different combinations of pre-trained BERT with fine tuning and continued pre-training for domain adaptation, which is future work for the industrial domain (Gururangan et al., 2020).

The performance evaluation methods reviewed in this paper were by no means exhaustive. The methods were selected as they appeared to be the most commonly used in the literature reviewed. The clustering qualitative test shown here was mostly specific to word embeddings, but there are additional visualization approaches developed for explaining and understanding model behavior specific to attention-/transformer-based representations. For instance, there are tools which offer visualizations of how the model assigns attention weights which have been shown to help qualitatively explain model behavior (Vig, 2019).

Ultimately, we hope that the methods reviewed in this paper will be useful for the technical community in understanding different considerations for determining which data representation to use and to help explain model behaviors. While it may be extremely tempting to use the latest method from the mainstream AI community with technical text, hopefully the considerations suggested in this paper will help guide and justify the model decision process based on industrial business needs.

ACKNOWLEDGEMENT

The authors acknowledge Devang Gandhi, Xiaohui (Mark) Hu, Veerappan Muthaiah, and Mahesh Asati for their mentorship and support.

NOMENCLATURE

AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
BOW	Bag of Words
CMMS	Computerized Maintenance Management System
EAM	Enterprise Asset Management
FMEA	Failure Mode and Effects Analysis
MLM	Masked Language Modeling
MTBF	Mean Time Between Failures
MWO	Maintenance Work Order
NER	Named entity recognition
NLP	Natural Language Processing
NSP	Next Sentence Prediction
OOV	Out of Vocabulary
RAM	Reliability Availability Maintainability
RCM	Reliability Centered Maintenance
SOTA	State of the Art
TFIDF	Term Frequency Inverse Document Frequency
TLP	Technical Language Processing

REFERENCES

Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Pasca, M., & Soroa, A. (2009). A study on similarity and

relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, (19-27).

Alsentzer, E., Murphy, J.R., Boag, W., Weng, W., Jin, D., Naumann, T., Mcdermott, M.B.A (2019). Publicly available clinical BERT embeddings. *arXiv preprint* arXiv:1904.03323.

Arif-Uz-Zaman, K., Cholette, M.E., Ma, L., & Karim. A. (2017). Extracting failure time data from industrial maintenance records using text mining. *Advanced Engineering Informatics, 33*, 388-396. doi: 10.1016/j.aei.2016.11.004

Bastos, P., Lopes, I., Pires, I. (2012). A maintenance prediction system using data mining techniques. *World Congress on Engineering*. London, p. 1448-1453. ISBN 978-988-19252-2-0

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, Alex., Pereira, K., & Vaughan, J.W. (2010). A theory of learning from different domains. *Machine learning, 79*(1), 151-175. doi: I 10.1007/s10994-009-5152-4

Bokinsky, H., McKenzie, A., Bayoumi, A., & McCaslin, R. (2013). Application of natural language processing techniques to marine V-22 maintenance data for populating a CBM-oriented database. In *AHS Airworthiness, CBM, and HUMS Specialists' Meeting*. 463-472.

Bouabdallaoui, Y., Lafhaj, Z., Yim, P., Ducoulombier, L., & Bennadji, B. (2020). Natural Language Processing Model for Managing Maintenance Requests. *Buildings, 10*(9), 160.

Brundage, M.P., Kulvatunyou, B., Ademujimi, T., Badarinath, R. (2017). Smart manufacturing through a framework for a knowledge-based diagnosis system. In *International Manufacturing Science and Engineering Conference.* American Society of Mechanical Engineers.

Brundage, M.P., Sexton, T., Hodkiewicz, M., Dima, A., & Lukens, S. (2021). Technical language processing: Unlocking maintenance knowledge. *Manufacturing Letters, 27*, 42-46. doi:10.1115/MSEC2017-2937

Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., & Androutsopoulos, I. (2020). LEGAL-BERT: The Muppets straight out of law school. *arXiv preprint.* arXiv:2010.02559

Chen, C., Huang, H., & Chen, H. (2020). NLP in FinTech applications: past, present and future. *arXiv preprint* arXiv:2005.01320.

Choudhary, V. (2020). Calculating Document Similarities using BERT, Word2Vec, and other models. Retrieved from *Towards Data Science.* https://towardsdatascience.com/calculating-document-similarities-using-bert-and-other-models-b2c1a29c9630.

Clark, A., & Lappin, S. (2010). *The handbook of computational linguistics and natural language processing.* 197-220. doi: 10.1002/9781444324044.ch8.

D'Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., Chen, C., Deaton, J., Eisentein, J., Hoffman, M.D., Hormozdiari, F., Houlsby, N., Hou, S., Jerfel, G., Karthikesalingam, A., Lucic, M., Ma, Y., Mclean, C., Mincu, D., Mitani, A., Montanari, A., Nado, Z., Natarajan, V., Nielson,C., Osborne, T.F., Raman, R., Ramasamy, K., Sayres, R., Schrouff, J., Seneviratne, M., Sequeira, S., Suresh, H., Veitch, V., Vladymyrov, M., Wang, X., Webster, K., Yadlowsky, S., Yun, T., Zhai, Z., & Sculley, D. (2020). Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint.* arXiv:2011.03395

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for language understanding. arXiv preprint.* arXiv:1810.04805

Dima, A., Lukens, S., Hodkiewicz, M., Sexton, T. & Brundage, M.P. (2021). Adapting natural language processing for technical text. *Applied AI Letters*. doi:10.1002/ail2.33

Ethayarajh, K. & Jurafsky, D. (2020). Utility is in the Eye of the User: A Critique of NLP Leaderboards. *arXiv preprint*. arXiv:2009.13888

Federspiel, C. & Villafana, L. (2003). Design of a maintenance and operations recommender. *ASHRAE Transactions, 109*(2), 677-683.

Gao, Y., Woods, C., Liu, W., French, T., & Hodkiewicz, M. (2020). 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 (ESREL)*. Venice, Italy.

Geigle, C., Mei, Q., Zhai, C. (2018). Chapter 2: Feature engineering for text data. In Gong D. & Liu, H. (Eds.). *Feature engineering for machine learning and data analytics*. 15-54.

Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J., & Poon, H. (2020). Domain-specific language model pretraining for biomedical natural language processing. *arXiv prepri*n. arXiv:2007.15779

Gunay, H.B., Shen, W., & Yang C. (2019). Text-mining building maintenance work orders for component fault frequency. *Building Research & Information*, 47(5), 518-533. doi:10.1080/09613218.2018.1459004

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., & Smith, N. A. (2020). Don't Stop Pretraining: Adapt Language Models to

Domains and Tasks. *arXiv preprint*. arXiv:2004.10964

Hansen, B., Coleman, C., Zhang, Y., & Seale, M. (2020). Text Classification and Tagging of United States Army Ground Vehicle Fault Descriptions in Support of Data-Driven Prognostics. In *Proceedings of the Annual Conference of the PHM Society*. Virtual. doi:10.36001/phmconf.2020.v12i1.1154

Hodkiewicz, M., & Ho, M.T. (2016). Cleaning historical maintenance work order data for reliability analysis. *Journal of Quality in Maintenance Engineering*. 22(2),146-163. doi:10.1108/JQME-04-2015-0013

Hodkiewicz, M., Kelly, P., Sikorska, J., & Gouws, L. (2006). A framework to assess data quality for reliability variables. *Engineering Asset Management*, (137-147). Springer, London. doi:10.1007/978-1-84628-814-2_15

Hodkiewicz, M., Batsioudis, Z., Radomiljac, T., & Ho, M. T. (2017). Why autonomous assets are good for reliability - the impact of 'operator-related component' failures on heavy mobile equipment reliability. In *Proceedings of the Annual Conference of the PHM Society*. St. Petersburg, FL. doi: 10.36001/phmconf.2017.v9i1.2449.

Khabiri, E., Gifford, W.M., Vinzamuri, B., Patel, D., & Mazzoleni, P. (2019). Industry specific word embedding and its application in log classification. In *Proceedings of The 28th ACM International Conference on Information and Knowledge Management*, (2713-2721). doi: 10.1145/3357384.3357827

Khandelwal, U., Levy, O., Jurafsky, D., Zettlemoyer, L., Lewis, M. (2019). Generalization through memorization: Nearest neighbor language models. *arXiv preprint.* arXiv:1911.00172

Kovaleva, O., Romanov, A., Rogers, A., & Rumshisky A., (2019). Revealing the dark secrets of BERT. *arXiv preprint.* arXiv:1908.08593

Kusner, M.J., Sun, Y., Kolkin, N. I., Weinberger, K.Q. (2015). From word embeddings to document distances. *Proceedings of the 32nd International Conference on Machine Learning*, (957-966).

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., Ho So, C., & Kang, J. (2020). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, *36*(4), 1234-1240. doi: 10.1093/bioinformatics/btz682

Levy, O., & Goldberg, Y. (2014). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, (302-308).

Lucy, L., & Bamman, D. (2021). Characterizing English Variation across Social Media Communities with BERT. *arXiv preprint.* arXiv:2102.06820

Lukens, S., & Markham, M. (2018). Data-driven Application of PHM to Asset Strategies. In *Annual Conference of the PHM Society*. Philadelphia, PA. doi: 10.36001/phmconf.2018.v10i1.245

Lukens, S., Markham, M., Naik, M., Laplante, M. (2019). Data-driven approach to estimate maintenance life cycle cost of assets. In *Model-Based Enterprise Summit (MBE2019)*. doi:10.13140/RG.2.2.14005.32489

Lukens, S., Naik, M., Hu, X, Doan, D.S., & Abado, S. (2017). The role of transactional data in prognostics and health management work processes. In *Proceedings of the Annual Conference of the PHM Society* (517-528). St. Petersburg, FL. doi: 10.36001/phmconf.2017.v9i1.2473

Lukens, S., Naik, M., Saetia, K., Hu, X. (2019). 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. doi: 10.36001/phmconf.2019.v11i1.836

McKenzie A., Matthews M., Goodman N., Bayoumi A. (2010). Information Extraction from Helicopter Maintenance Records as a Springboard for the Future of Maintenance Text Analysis. In *Trends in Applied Intelligent Systems. IEA/AIE 2010. Lecture Notes in Computer Science*. 6096. García-Pedrajas N., Herrera F., Fyfe C., Benítez J.M., Ali M. (eds). Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-13022-9_59

Navinchandran, M., Sharp, M. E., Brundage, M. P., & Sexton, T. B. (2019). Studies to Predict Maintenance Time Duration and Important Factors From Maintenance Workorder Data. In *Annual Conference of the PHM Society*. Scottsdale, AZ. doi:10.36001/phmconf.2019.v11i1.792

Oniani, D., & Wang, Y. (2020). A qualitative evaluation of language models on automatic question-answering for covid-19. In *Proceedings of the 11th ACM International Conference on Bioinformatics*, (1-9). Computational Biology and Health Informatics.

Pathak, A. (2021). Comparative Analysis of Transformer based Language Models. In *CS & IT Conference Proceedings.*

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Muller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. (2011). *Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research.* arXiv:1201.0490

Pennington, J., Socher, R., Manning, C. (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, (1532-1543). doi:10.3115/v1/D14-1162

Pereira, P.C. (2020). Text-Mining Maintenance Records to Automate the Identification and Grouping of Failure Modes. *Offshore Technology Conference*. doi:10.4043/30737-MS

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint*. arXiv:1802.05365

Rajani, N.F., Krause, B., Yin, W., Tiu, T., Socher, R., & Xiong, C. (2020). Explaining and improving model behavior with k nearest neighbor representations. *arXiv preprint*. arXiv:2010.09030

Rajasekharan, A. (2020, November 4). Maximizing BERT model performance. Retrieved from *Towards Data Science* https://towardsdatascience.com/maximizing-bert-model-performance-539c762132ab

Rajpathak, D. G. (2013). An ontology based text mining system for knowledge discovery from the diagnosis data in the automotive domain. *Computers in Industry, 64*(5), 565-580. doi:10.1016/ j.compind.2013.03.001

Rehurek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks*, (45-50), May 22, 2010. Valletta, Malta. http://is.muni.cz/publication/884893/en

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 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)*, Association for Computational Linguistics, (3982–3992). Hong Kong, China.

Saetia, K., Lukens, S., Pikcle, E., Hu, X. (2019). Data-driven approach to equipment taxonomy classification. In *Annual Conference of the PHM Society*. Scottsdale, AZ. doi: 10.36001/phmconf.2019.v11i1.818

Seale, M., Hines, A., Nabholz,G., Ruvinsky, A., Eslinger, O., Rigoni, N., & Vega-Masionet, L. (2019). Approaches for Using Machine Learning Algorithms with Large Label Sets for Rotorcraft Maintenance. In *2019 IEEE Aerospace Conference* (1-8), Big Sky, MT, USA. doi: 10.1109/AERO.2019.8742027

Sexton, T.B., Fuge, M. (2019). Using Semantic Fluency Models Improves Network Reconstruction Accuracy of Tacit Engineering Knowledge. In *ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. doi:10.1115/DETC2019-98429

Sexton, T., and Fuge, M. (2020). Organizing Tagged Knowledge: Similarity Measures and Semantic Fluency in Structure Mining. *Journal of Mechanical Design, 142*(3). doi:10.1115/1.4045686

Sexton, T.B., Brundage, M.P., Hoffmanm, M.L., Morris, K.C. (2017). Hybrid datafication of maintenance logs from AI-assisted human tags. In *2017 IEEE international conference on big data* (1769-1777), Boston, MA, USA. doi: 10.1109/BigData.2017.8258120

Sexton, T., Hodkiewicz, M., Brundage, M.P., & Smoker, T. (2018). Benchmarking for keyword extraction methodologies in maintenance work orders. In *Annual Conference of the PHM Society*. Philadelphia, PA. doi: 10.36001/phmconf.2018.v10i1.541

Sikorska, J., Hammond, L., & Kelly, P. (2007). Identifying failure modes retrospectively using RCM data. In *ICOMS Asset Management Conference*. Melbourne, Australia.

The Prognostics Data Library (2021). Retrieved from UWA Systems Health Lab: https://prognosticsdl.ecm.uwa.edu.au/pdl/

Usuga Cadavid, J.P.,Grabot, B., Lamouri, S., Pellerin, R.,Fortin, A. (2020). Valuing free-form text data from maintenance logs through transfer learning with CamemBERT. *Enterprise Information Systems*, 1-29. doi:10.1080/17517575.2020.1790043

Vig, Jesse. (2019). A multiscale visualization of attention in the transfromer model. *arXiv preprin*t. arXiv:1906.05714

Wang, A. S. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint*. arXiv:1804.07461

Wang, Y., Liu, S., Afzal, N., Rastegar-Mojarad, M., Wang, L., Shen, F., Kingsbury, P., & Liu, H. (2018). A comparison of word embeddings for the biomedical natural language processing. *Journal of biomedical informatics, 87*, 12-20. doi:10.1016/j-jbi.2018.09.008

White, L. T. (2015). How well sentence embeddings capture meaning. *Proceedings of the 20th Australasian document computing symposium*, (1-8).

Wiedemann, G., Remus, S., Chawla, A., Biemann, C. (2019). Does BERT make any sense? Interpretable word sense disambiguation with contextualized embeddings. *arXiv preprint*. arXiv:1909.10430

Yang, C., Shen, W., Chen, Q., Gunay, B. (2018). A practical solution for HVAC prognostics: Failure mode and effects analysis in building maintenance. *Journal of Building Engineering, 15*, 26-32. doi: 10.1016/j.jobe.2017.10.013

Yang, Z., Baraldi, P., Zio, E. (2020). A novel method for maintenance record clustering and its application to a case study of maintenance optimization. *Reliability Engineering & System Safety, 203*(2), 107103. doi:10.1016/j.ress.2020.107103

Zhang, Y., Chen, Q., Yang, Z., Lin, H., & Lu, Z. (2019). BioWordVec, improving biomedical word embeddings with subword information and MeSH. *Scientific data*, *6*(1), 1-9. doi: 10.1038/s41597-019-0055-0

Zhong, H., Xiao, C., Tu, C., Zhang, T., Liu, Z., and Sun, M. (2020). How does NLP benefit legal system: A summary of legal artificial intelligence. *arXiv preprint.* arXiv:2004.12158

BIOGRAPHIES

Ajay Varma Nandyala was born and graduated in India with a Computer Science Degree and a minor in Artificial Intelligence in 2015. He completed his master's from University of North Texas in Information Science major with focus on Pattern Recognition and Natural Language Processing in 2018. He joined GE Digital as a Data Scientist in 2020 and works from Atlanta, GA, USA. His work has involved developing recommendation systems using Natural Language Processing, developing timeseries forecasting models, and developing computer vision models. His current research interests are in the areas of Natural Language Processing, Pattern Recognition, and their applications to a broad range of fields.

Sarah Lukens lives in Roanoke, Virginia and is a Data Scientist at GE Digital. Her interests are focused on data-driven modeling for reliability applications by combining modern data science techniques with current industry performance data. This work involves analyzing asset maintenance data and creating statistical models that support asset performance management (APM) work processes using components from natural language processing, machine learning, and reliability engineering. Sarah completed her Ph.D. in mathematics in 2010 from Tulane University with focus on scientific computing and numerical analysis with applications in fluid-structure interaction problems in mathematical biology. Prior to joining Meridium (now GE Digital) in 2014, she conducted post-doctoral research at the University of Pittsburgh and the University of Notre Dame building data-driven computational models forecasting infectious disease spread and control. Sarah is a Certified Maintenance and Reliability Engineer (CMRP).

Sundararam Rathod lives in Bengaluru, India and is a Data Scientist at GE Digital. He completed his master's from King's College London, UK in the field of Intelligent Systems in 2016 and joined GE in 2017. His work has involved applying anomaly detection techniques to prevent failures in gas turbines using sensor data, natural language processing techniques to build prescriptive analytics and improving efficiency in manufacturing plants using unsupervised methods. His focus lies in utilizing industrial data to build models to help in monitoring and improving industrial plant performance.

Pratiksha Agrawal lives in Durgapur, India and is a Data Scientist at GE Digital. She completed her B.Tech in Biotechnology in 2019 from Indian Institute of Technology, Roorkee, India. She joined GE Digital as a Data Science Specialist in July 2019 and in her role, she has analyzed the historical GE data and used it to elevate asset performance management (APM). She is currently involved in building analytics for anomaly prediction, Key Performance Indicator (KPI) forecasting in manufacturing domain. Her interest lies in utilizing unsupervised machine learning methods to maximize the industrial productivity and employ text mining or natural language processing to reduce the manual data combing efforts.