Proceedings of the ASME 2019 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2019
August 18-21, 2019, Anaheim, USA

IDETC2019-98429

USING SEMANTIC FLUENCY MODELS IMPROVES NETWORK RECONSTRUCTION ACCURACY OF TACIT ENGINEERING KNOWLEDGE

Thurston Sexton*
Systems Integration Division
Engineering Laboratory
National Institute of Standards and Technology
Gaithersburg, Maryland 20871
Email: thurston.sexton@nist.gov

Mark Fuge
Dept. of Mechanical Engineering
University of Maryland
College Park, Maryland 20742
Email: fuge@umd.edu

ABSTRACT

Human- or expert-generated records that describe the behavior of engineered systems over a period of time can be useful for statistical learning techniques like pattern detection or output prediction. However, such data often assumes familiarity of a reader with the relationships between entities within the system—that is, knowledge of the system's structure. This required, but unrecorded "tacit" knowledge makes it difficult to reliably learn patterns of system behavior using statistical modeling techniques on these written records. Part of this difficulty stems from a lack of good models for how engineers generate written records of a system, given their expertise, since they often create such records under time pressure using shorthand notation or internal jargon. In this paper, we model the process of maintenance work order creation as a modified semantic fluency task, to build a probabilistic generative model that can uncover underlying relationships between entities referenced within a complex system. Compared to more traditional similarity-metric-based methods for structure recovery, we directly model a possible cognitive process by which technicians may record work-orders. Mathematically, we represent this as a censored local random walk over a latent network structure representing tacit engineering knowledge. This allows us to recover implied engineering knowledge about system structure by processing written records. Additionally, we show that our model leads to improved generative capabilities for synthesizing plausible data.

1 INTRODUCTION

Due in part to an explosion of interest in statistical modeling techniques, specifically machine learning (ML), much recent effort has been devoted to using various forms of engineering data for training these models. These models, trained on historical engineering data to detect patterns of classification, fault detection, performance estimates, etc., promise to reliably automate many of these labor-intensive tasks, freeing the time of designers and maintainers for more high-level decisions. However, in technical fields like engineering, the available historical data is often difficult to use directly — the experts creating it in the past generally assumed it would be read and adapted by colleagues or experts in their own field. This causes analysts to, quite often, lack the information needed to appropriately represent and process this data. One cannot simply use, *e.g.*, written lab notebooks, technical reports, or maintenance work-orders (MWOs) as is, taking them at face value: words and concepts with more general meaning to the layman will have domain-specific technical application that must be accounted for if a statistical model is to learn a robust representation of the semantic space. In this paper, our goal is to infer how the original data creators/experts structure their own knowledge about the problem at hand. This "structured knowledge" can then be used create more reliable models for engineering learning tasks.

This paper presents initial techniques to automatically infer key parts of this tacit structured knowledge, and explores a mechanism to extract it from observations/historical records written

*Address all correspondence to this author.

by human experts. To do this, we frame the act of recording engineering events as a type of memory recall, which we assume occurs within a broader "network" of system relationships that structure the expert's knowledge about a system's behaviors (but that we do not have direct access to and thus must infer through examples). Specifically, we show that:

1. By explicitly modeling work-order generation as non-Markovian memory recall over learned object relationships, we can more accurately recover those relationships than by using more traditional token similarity measures, and subsequently,
2. learning such relationships provides a generative model of each object's conditional relationships in the form of a graph, for which performing a random walk from points of interest (*e.g.*, a Failed part) will synthesize more realistic new data.

We demonstrate this on two examples of maintenance work orders: (1) synthetically generated work orders from real-world engineering systems with a known ground-truth structures; and (2) actual maintenance work orders from an excavator. In both cases, we show that by building a probabilistic model that accounts for (and subsequently learns) how experts structure their implicit knowledge of a domain, one can often achieve significantly better performance (as measured by standard information retrieval metrics) than existing methods of structure recovery.

2 RELATED WORK

Using data to infer the underlying structure of a complex system is a long-standing goal within both systems engineering and other domains that depend upon accurate network recovery, such as: biological systems and disease transmission vector modeling [1, 2]; uncovering economic interactions and social networks [3, 4]; inferring physical models by learning governing equations [5, 6]; or even description generation in computer vision, and quantifying how humans reason about belonging and causality in ambiguous images or contexts [7, 8]. For written (text-based) documents specifically, we can group major methods to perform structure recovery from unstructured written documents into roughly three camps: (1) prescriptive rule definition, (2) training statistical models (NLP), and (3) "folksonomies" and tag-based crowdsourcing.

2.1 Prescriptive Rules

The most straight-forward way to make tacit knowledge computable is to explicitly design the relationships as they are assumed to exist. An expert (or set of experts) define what objects are allowed to exist in the domain, and how those concepts relate to each other. These rules are then mapped onto the observed data, similar to constructing a thesaurus. This manually constructed rule-set can take the form of ontologies, *e.g.*, but they are always structured representations formed from from mixtures of domain expertise and example data, which can then be used to parse remaining data, and restrict the format of future data. For example, ISO-15926 defines a data model [9, 10] with which one can constrain engineering records to have precise, unambigious meanings, and later work built on the standard construct ontologies with which to reason over these meanings and their relationships [9, 11–13].

In practice, however, a particular domain or data-set will not have existing, applicable ontologies or data structures, and time investment needed to create them for sufficiently generalized usage is commonly out-of-scope for analysts to dedicate. Some work has been done to automate this process [14], but such techniques generally require us to rely on language-specific syntactical rules (*i.e.*, grammar). Data-entry errors and shorthand are ubiquitous in technical records, where grammar is often low-priority if system-familiarity is assumed. In these cases, sophisticated systems of rules are still often developed, potentially with reduced formalism or scope, taking the form of keyword recognition and filtering rules to find a priori "useful" patterns for analysis [15, 16] In engineering design, similar manually-created rule-sets that define concept relationships are involved in constructing Design Structure Matrices (DSMs), which are often derived from expert input or technical/project documents [17–19]. Regardless, this paper assumes that the need for low-cost, low effort estimates of a system's "rules" is not met by requiring a designer to manually intervene.

2.2 Natural Language Processing

Rather than build patterns manually, natural language processing (NLP) often deals with the use of significant quantities of text to discover latent patterns automatically. This requires finding mathematical representations of text, like "bag-of-words" weightings [20], topic models [21, 22], or semantic vector embeddings [23, 24]. These transformations enable the use of text-based documents in statistical models that can, for instance, train a classifier to automate labeling of work orders [25]. The success of this approach is fundamentally linked to the notion that supervised ML models use *labeled training data* to learn these patterns, and the quality of the model increases with the amount of available labeled data-points — using this approach with few labels presents a problem of diminishing returns. Time saved by automating document classification scales with the amount of time spent labeling document classifications.

This trade-off is problematic in highly technical and jargon-filled domains, where existing models from more generalized training sets cannot be easily or reliably transitioned. In addition, the actual patterns being "learned" are quite often difficult to interpret and use for humans [26], stemming from the so-called "black box" nature of these models, despite an inherent need to

justify our engineering decisions with evidence-based reasoning. This paper puts forward an unsupervised model, able to function with few training samples, but only as a stepping stone in the process toward encoding the types of prescriptive knowledge that can be used to communicate and train future operators/designers.

2.3 Folksonomies

In contexts where dedicated annotation labor can be difficult to secure, significant research has been done to present less restrictions to casual annotators, and understand how natural classification and labeling schemes arise in social communities, *e.g.* online tagging efforts [27]. Tags, a form of multi-label classification, allow concepts to be derived freely in the course of work, where repeated and cross-contexutal usage leads to a naturally-arising set of useful, domain-specific concepts; this is commonly referred to as a *folksonomy*, a portmanteau of "folk" and "taxonomy" [28]. Because folksonomies generally ask users to determine minimal representative labels rather than strict classifications (*i.e.*, tags), each label can be seen in multiple contexts. The predominant way to analyze these tags, then, is by their co-occurrences with each other: intuitively, highly co-occurring tags are considered "similar." [29, 30]. A basic, but commonly used measure of this co-occurrence is the *cosine similarity*: if, over a set of C documents, tag t_k has binary vector $u_k = \{\mathbf{1}_c(t_k) : c \in C\}$, then the cosine similarity s between the binary occurrence vectors of the tags $t_1; t_2$ is defined as:

$$s(t_1; t_2) = \frac{u_1 \cdot u_2}{\|u_1\| \, \|u_2\|} \tag{1}$$

This measure has seen consistent usage in folskonometric methods to structuring relationships between tagged concepts in useful ways [31–33]. Because various annotators will perceive the importance and relevance of each tag differently in each context, these ambiguities are typically overcome through crowdsourcing, by having large numbers of users tag. This allows a statistical "smoothing" over differences in expertise. However, in the case of technical tags from a few experts, this benefit from large numbers of annotators is not something that we can count on. Additionally, the types of relationship information we might want is not purely statistical/distributional similarity, as experts creating documents will have several core views about what "being related" in their system entails. Consequently, we believe it is important to exploit potential cognitive processes by which these tags might be produced, to enforce a greater degree of information *precision* than typical similarity measures might allow for in their desire for increased information *recall*.

3 MODELING WORK ORDER CREATION

As discussed above, common techniques for discovering structure in human-annotated or natural-language data primarily rely on frequency and co-occurrence information of discrete objects/concepts. These are powerful and easy-to-apply models of speech or the written word, but can miss key causal links implied in the original text, which are difficult to extract this way without significant amounts of data or relevant pre-training. Instead, this paper proposes that by explicitly modeling the conditional dynamics of how humans recall concepts within this data—which, for the purposes of this work will be limited to MWO's—we can extract the conditional relationships between the mentioned objects or concepts that best match what was recorded by the experts.

This section first describes the concept of *semantic fluency*—a existing psychological theory of concept recall—and how that theory relates to the construction of written engineering documents, specifically MWOs in this paper. We then describe a computational method to implement the concept of semantic fluency using Initial-Visit Emitting Random Walks (INVITE) [34]—a probabilistic model of graph walks that is non-Markovian.

3.1 Semantic Fluency

When a technician begins to record a MWO, they try to search their memory for words that represent concepts relevant to the MWO itself. These consist of items, problems that were encountered with some items, and how other items were used to solve these problems [35]. The exact psychological mechanisms by which a person searches through their memory is still an active area of research and has been modeled in various ways. Some recent studies [36] propose that concepts are recalled sequentially by foraging in "semantic patches"—in brief, that humans sequentially recall concepts that are "near" each other in some person-specific semantic space built through experience.

Specifically, these patches are thought of as existing in a high-dimensional concept-space,[1] and the likelihood that some concept is recalled next is based on combining both associative and categorical knowledge into a similarity measure between the current recalled entity and the next. The classic psychological experiment for this model is the Semantic (or, Verbal) Fluency test:

1. Recall and record an object (*e.g.*, an animal);
2. Record the next object of this type you think of;
3. Continue recording for the remaining time

The reader is encouraged to try this process out for themself. One advantage of this test lies in not restricting (or having

[1] Though less applicable in technical or domain-specific corpuses where examples are too few and far between, this is the intuition that leads to the success of vector-based semantic embeddings like `gloVe` or `word2vec` [23, 24].

to specify apriori) the relationship between objects required to record subsequent ones. For example:

$$\text{dog} \rightarrow \text{cat} \rightarrow \text{tiger} \rightarrow \text{lion} \rightarrow \text{elephant} \rightarrow \text{wolf} \cdots$$

For example, it is common for animal-based semantic fluency lists to start with household pets, potentially switching to entirely unrelated categories like "large cats," for further exploration, before either retracing back to a previous category (*e.g.*, canines to "wolf" via "dog") or onward via new similarities (*e.g.*, African animals to "elephant" via "lion"). Different people can create different fluency lists, owing to differences in how they psychologically structure relationships between concepts.[2]

The key contribution of this work is to propose that explicitly modeling this process lends itself well to recovering engineering knowledge from text-based technical records. While, technicians are not purely sampling arbitrary system concepts, as you might a list of animals, we nevertheless assume that each subsequent concept written in a MWO is directly conditional on what was written previously. [3] Then, an MWO consists of "jumps" between concepts that depend upon previously "visited" concepts. This assumption allows us to infer relationships between concepts given examples of MWOs. This boils down to two key components of the technician's cognitive task when recalling relevant information to write down MWO's:

- A technician records concepts sequentially, as he or she recalls unique defining characteristics of the MWO.
- They recall these characteristics by remembering links between them, and any recently recalled characteristics.

This differs from a standard Bag of Words model—where all entities occurring in a document are assumed to be linked through co-occurrence—and from n^{th}-order language models—where relations are limited to the nearest (or, previous) n entities. In technical shorthand (like MWOs), objects listed later on may be linked to any of the previously mentioned objects, not strictly those directly adjacent to it. For instance, the MWO *"Leaking hydraulic valve; cleaned oil spill and replaced O-ring"* consists of a sequence of concepts (*leak→hydraulic→valve...*), not all of which share the same causal structure: perhaps "hydraulic", "valve", and "leak" are all potentially subsets of a hydraulic "system", but "replace", "clean", and "oil" all have potential to span subsystems. Similarly, in this MWO, "oil" would likely be considered as linked with "leak", more than it would to the *closer* entity "replace". This illustrates nicely the trade-off between categorical and associative memory foraging that [36] discusses at length, and is precisely the feature of MWOs we exploit to extract a more sparse representation of system relationships through the Initial-Visit Emitting Random Walks semantic fluency model, which we detail next.

3.2 Initial-Visit Emitting Random Walks

Based on the above modelling assumptions, we demonstrate the application of an Initial-Visit Emitting Random Walks (INVITE) as initially described by [34], on recovering system structures from MWOs. Say the set of components or concepts in our system is denoted by the node-set n. A set of T tags [4] can be denoted as a Random Walk (RW) trajectory $\mathbf{t} = \{t_1; t_2; t_3; \cdots t_T\}$, where $T \leq n$. However, this limit on the size of T assumes tags are a set of unique entries: any transitions between previously visited nodes in $\mathbf{t}$ will not be directly observed, making the transitions observed in $\mathbf{t}$ strictly non-Markovian, and allowing for a potentially infinite number of possible paths to arrive at the next tag.

Instead of directly computing over this intractable model for generating $\mathbf{t}$, the key insight from the original INVITE paper comes from partitioning $\mathbf{t}$ into $T-1$ Markov chains with absorbing states, where previously visited nodes are transient states, and unseen nodes are absorbing. It is then possible to calculate the absorption probability into the k^{th} transition ($t_k \rightarrow t_{k+1}$) using the *fundamental matrix* of each partition. If the partitions at this jump consist of q transient states with transition matrix amongst themselves $\mathbf{Q}^{(k)}_{q\times q}$, and r absorbing states with transitions into them from q as $\mathbf{R}^{(k)}_{q\times r}$, the Markov chain $\mathbf{M}^{(k)}_{n\times n}$ has the form

$$\mathbf{M}^{(k)} = \begin{matrix} \mathbf{Q}^{(k)} & \mathbf{R}^{(k)} \\ \mathbf{0} & \mathbf{I} \end{matrix} \tag{2}$$

where $\mathbf{0}$, $\mathbf{I}$ represent lack of transition between/from absorbing states. It follows from [38] that the probability P of a chain starting at t_k being absorbed into state $k+1$, letting $\mathbf{N} = (\mathbf{I}-\mathbf{Q})^{-1}$, is given as

$$P(t_{k+1}|t_{1:k}; \mathbf{M}) = \mathbf{N}^{(k)} R^{(k)}_{q;1} \tag{3}$$

The probability of being absorbed at $k+1$ conditioned on jumps $1:k$ is thus equivalent to the probability of observing the

[2] *e.g.*

$$\text{dog} \rightarrow \text{walk} \rightarrow \text{run} \rightarrow \text{gym} \rightarrow \cdots \quad \text{vs}$$
$$\text{dog} \rightarrow \text{home} \rightarrow \text{family} \rightarrow \text{meal} \rightarrow \cdots$$

[3] This is standard practice in the language modeling domain [37].

[4] While traditional application of "tagging" assumes the set of labels to be strictly un-ordered (as in multi-label classification), we follow [15,35] by assuming tags are generated directly from text by keyword recognition. It is thereby trivial to reverse the process, assigning each tag a position as the first time its corresponding keyword was recognized in the original text.

$k+1$ INVITE tag. If we approximate an a priori distribution of tag probabilities to initialize our chain as $t_1 \sim \text{Cat}(n; q)$ (which could be empirically derived or simulated), then the likelihood of our observed tag chain $\mathbf{t}$, given a Markov chain, is

$$\mathscr{L}(\mathbf{t} \mid q; \mathbf{M}) = q(t_1) \tilde{\prod}_{k=1}^{T-1} P(t_{k+1} \mid t_{1:k}; \mathbf{M}) \tag{4}$$

Finally, if we observe a corpus of tag lists $\mathbf{C} = \{\mathbf{t}_1; \mathbf{t}_2; \cdots; \mathbf{t}_c\}$, and assume q can be estimated separately from $\mathbf{M}$, then we can finally frame the problem as minimizing our loss function, the negative log-likelihood of our corpus over $\mathbf{M}$:

$$\mathbf{M}^* = \arg\min_{\mathbf{M}} \sum_{i=1}^{C} \sum_{k=1}^{T_i-1} -\log P\left(t_{k+1}^{(i)} \mid t_{1:k}^{(i)}; \mathbf{M}\right) \tag{5}$$

3.3 Implementation

As stated in Eq. 5, the optimization is constrained; in addition to requiring row-stochasticity, the matrix N is only guaranteed to exist if self-transitions are disallowed, as proved in [34]. Similar to that implementation, we introduce a softmax re-parameterization of $\mathbf{M}$ that allows the optimization to be unconstrained in $\mathbb{R}^{n \times n}$, and guaranteeing row-stochasticity.

$$M_{i;j} \leftarrow \frac{\exp(M_{i;j})}{\sum_j \exp(\mathbf{M}_i)_j}$$

However, we introduce several modifications to this re-parameterization:

Edge Weights Because it is important for our purposes to estimate the weight (*i.e.*, importance) of each relationship, we to not require (as in [39]) that the structure of $\mathbf{M}$ is un-weighted—in this case each relationship would either exists or not exist. However, sparsity of $\mathbf{M}$ is still desirable, so we apply an L_1-penalty to the loss function, adding an $(\alpha/T) \cdot \|\mathbf{M}\|_1$ term to Eq. 5. The parameter α should generally be tuned via cross-validation where possible, but to demonstrate effectiveness in an unsupervised setting (as is expected to be the case when no "true" $\mathbf{M}$ is yet known), we use $\alpha = 0.01$, which was found to be robust to sensitivity trials for one log-factor in either direction.

Edge Direction In addition, Eq. 5 implies that $\mathbf{M}$ represents a *directed* graph. Though we model each tag as being generated conditional on preceding tags alone, we wish to preserve the intuition that relationships between tags are still assumed to be bi-directional, while not strictly enforcing $\mathbf{M}$ to be symmetric (undirected), as in [39]. Put simply, one-directional relationships can be useful when they are strictly the case (*e.g.*, oil→leak), but we may not wish to encourage one-directional relations that are quirks of imbalanced data and how people talk (gear 1 ↔ gear 2) To ensure the recovered weights in each direction are meaningful, and to speed-up recovery of what we assume is a "symmetry-dominant" $\mathbf{M}$, we bias it toward symmetry via an update to each entry prior to softmax:

$$M_{i;j} \leftarrow \max\left(M_{i;j}; M_{j;i}\right)$$

Because of these alterations, the analytic gradient for the INVITE loss function described in [34] no longer applies; instead, we make use of automatic differentiation as a means to ensure accurate gradient calculations under the above modifications [40]. The package `autograd` [41] was used to exploit a number of convenience functions for doing so, in the Python programming language.

4 EXPERIMENTS

The first experiment demonstrates the tractability of the INVITE model in the context of MWO-type data by generating synthetic MWOs from real engineering systems as described in [42]. We use these synthetic MWOs to (1) measure the network recovery accuracy of the INVITE model, (2) compute the sample efficiency of the INVITE model, and (3) compare the INVITE model to co-occurrence similarity thresholding models currently used in the state-of-the-art.

Second, we apply our proposed method to a corpus of real excavator MWOs, for which a "true" underlying structure does not yet exist. We compare the plausibility of work orders sampled from our network estimation to the original dataset and benchmark our model with respect to purely associative sampling.

Due to the high dimensionality of Eq. 5, and the noisy nature of observations, we use Stochastic Gradient Descent (SGD) to perform optimization of $\mathbf{M}$. Specifically, we use the ADAM algorithm [43], which modifies the gradient estimation for each iteration with first- and second-order momentum estimates from previous iterations, improving convergence behavior. Because each tag transition is considered a reliable observation, and the underlying structure of $\mathbf{M}$ is generally sparse relative to the complete adjacency graph between the set of all tags, a learning rate of 0.9 was used, but with minibatches of 5 censored lists each. Exponentially-weighted learning-rate decay was used, along with time-discounted averaging of $\mathbf{M}$, with settings as suggested by [44].

4.1 Exp. 1: Recovering Known Engineering Networks

To validate the ability of our method to accurately reconstruct engineering networks under varying data quantities, we

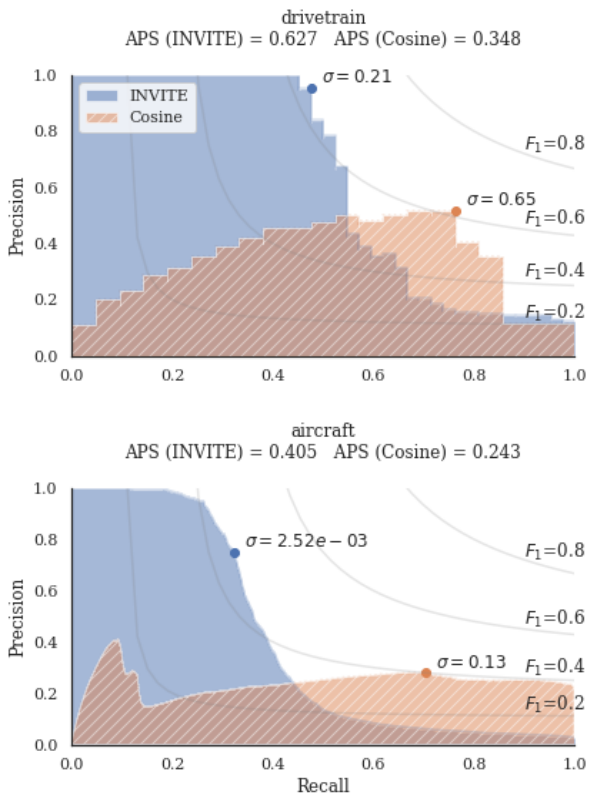

FIGURE 1: Comparing INVITE and Cosine-Similarity thresholding performance for recovering true network structure. **Top**: Recovery performance (precision vs. recall) for the *drivetrain* network. Trained on 18 samples ("work orders"), at 3 tags each. **Bottom**: Same comparison for the more complex *aircraft* network, trained on 1634 samples at 5 tags each. Also shown are the F_1-score iso-lines, along with F_1-optimal thresholds (s) for each model setting.

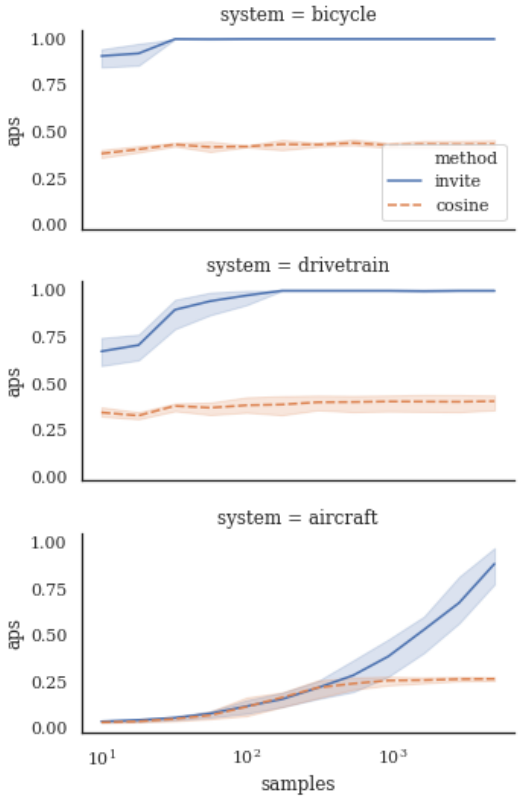

FIGURE 2: Mean average precision score (APS) for the three system networks of [42], shown with mean APS over sample lengths $T \in \{3; 4; 5\}$, and a 1000-bootstrap-sample 95% confidence interval. INVITE consistently outperforms similarity thresholding in low-data, low-complexity scenarios. In complex networks, performance is comparable until a significant number of samples are available, after which a lack of sparsity causes the cosine method's performance to plateau.

first synthesize censored tag lists from true component networks described in [42, 46]: a bicycle ($n = 10$), an automotive drivetrain ($n = 18$), and an aircraft ($n = 375$). Drawn layouts for each network are provided for reference in the appendix. For each network, censored random walks were generated by performing a random walk over the nodes until either 100 transitions or all nodes have been visited. The first unique visit to each node was recorded to simulate censoring, and the lists were clipped to the first 3,4, or 5 node visits, to reflect the typical number of tags seen in real MWO datasets (see Exp. 2, below, for an example). The number of censored lists used to train the models was evenly sampled at 11 intervals on a log-scale from 10 - 5000 lists, for a total $3 \times 11 \times 3 = 108$ trials.

Because the original networks are relatively sparse (See Table 1), the classification of edges as "existing" or "not existing" can be framed as a class-imbalanced information retrieval problem. Given some measure of node similarities (entries in the recovered adjacency matrix), we wish to threshold $\mathbf{M}$ such that, for a given threshold value $\mathsf{s} \in [0; 1]$, the entries of a thresholded adjacency matrix $\mathbf{M}^{\mathsf{s}}$ are given by:

$$M^{\mathsf{s}}_{i,j} = \begin{cases} 1; & \text{if } M^{*}_{i,j} \geq \mathsf{s} \\ 0; & \text{otherwise} \end{cases}$$

Prior to selecting a specific threshold, it is useful to recognize how robustly each model performs under varying threshold values, since the underlying "true" networks are available. For class-imbalanced learning problems like this, precision-recall (*P-R*) curves can elucidate model robustness under varying threshold sensitivities [47]. In Fig. 1, the precision-recall curves for

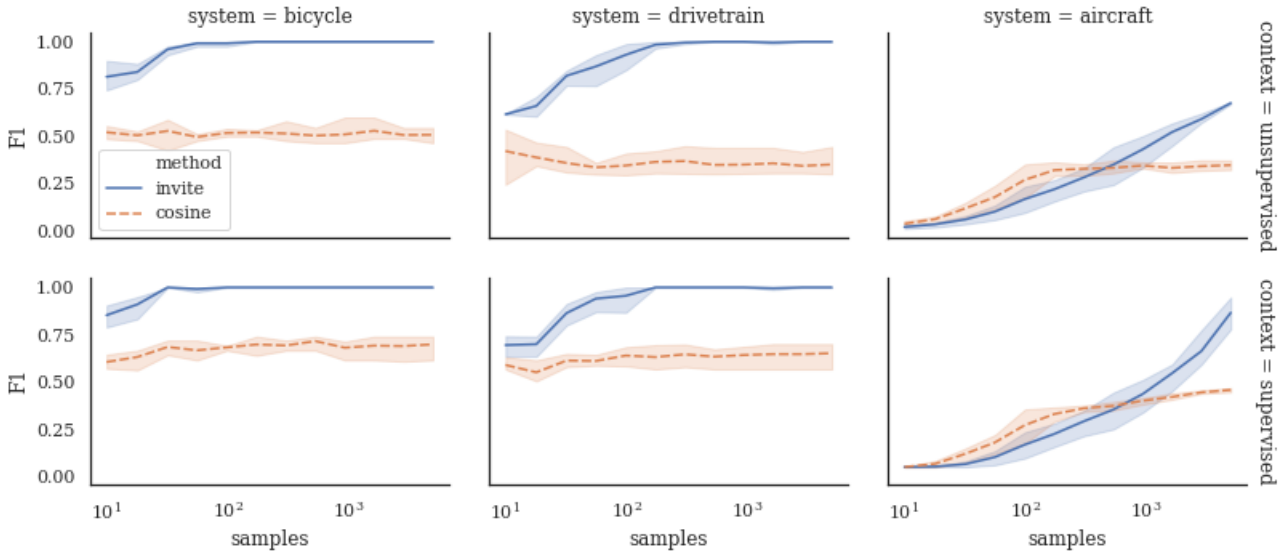

FIGURE 3: Mean F_1 reconstruction scores for the three system networks of [42], shown with mean score over sample lengths $T \in \{3;4;5\}$, and a 1000-bootstrap-sample 95% confidence interval. In an unsupervised context (top row), thresholds for node similarity were selected using a knee-finding heuristic [45], for where the EDF of edge-weights showed maximum curvature. In a supervised context (bottom row), the optimal threshold was selected as one that maximized the model's F_1-score. The INVITE method significantly outperforms pure co-occurrence similarity thresholds as the number of samples increases, and because the EDF is much more spread-out for cosine similarities, picking a "good" threshold is much more difficult than the sparsity-inducing INVITE models.

TABLE 1: Engineering component network summary for Experiment (1). Network models adapted from [42].

Model	Nodes	Sparsity
bicycle	10	80:0%
drivetrain	18	88:4%
aircraft	375	97:5%

the drivetrain model demonstrate that INVITE can quickly recover simpler networks with under 20 observations at relatively few "tags" each, while Cosine Similarity only robustly captures the global structure of the network—precision is relatively invariant over wide ranges of recall. This is even more pronounced for more complex networks, with the INVITE model capable of achieving either high precision or high recall, while the Cosine Similarity threshold has difficulty improving it under any circumstance. One way to summarize this robustness under varying threshold is calculate the average the precision score (APS) gained by each threshold's increase of recall R:

$$\text{APS} = \sum_{s} [R(s_i) - R(s_{i-1})] P(s_i) \tag{6}$$

The APS score will not give a "good" s, but instead summarizes the total "goodness" of each model across possible s. APS scores for the INVITE and Cosine Similarity models are shown against training set size in Fig. 2. APS eventually plateaus for the cosine model in every case. INVITE can perfectly recover the bicycle and drivetrain structures after around 100 samples. For the aircraft network, while INVITE has nearly identical performance to cosine similarity below 500 samples, INVITE's APS almost reaches 1.0 with 5000 samples.

In practice, selecting the value for s will depend on whether training examples are available: if not, a heuristic threshold such as knee-finding can be applied; if examples are available, it is possible to use performance measures appropriate for imbalanced learning problems (*e.g.*the F_1-score), and optimize the threshold for this value. In the common case that no training labels are available (no "true" structures are known), a common heuristic for thresholding values posits that diminishing returns occur for the retrieval function after the point of maximum curvature on the empirical distribution function (EDF) of values to threshold at the point of diminishing returns—*e.g.*, using a so-called "knee-finding" algorithm. To test the performance of both the cosine-similarity (bag-of-words) and INVITE recovered networks with respect to the originals, we apply the `kneedle` algorithm [45] to calculate a threshold s. The F_1-score can then be calculated for $\mathbf{M}^s$ as for each training-set size (see unsuper-

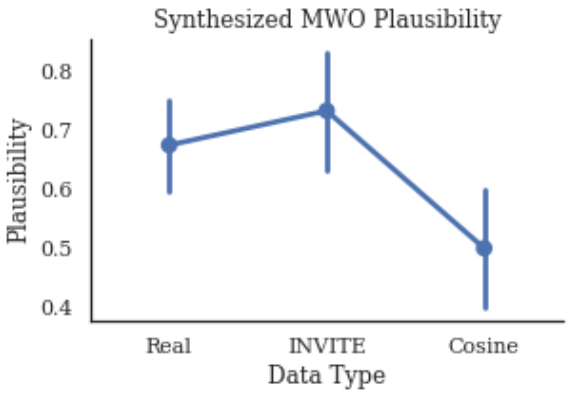

FIGURE 4: Mean plausibility ratio for 100 MWOs, both real and synthesized by sampling $\mathbf{M}^s$ recovered from INVITE and Cosine methods. Confidence intervals show the inter-quartile range of 1000 bootstrap samples.

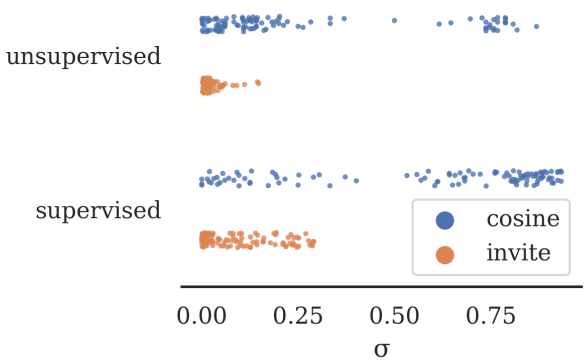

FIGURE 5: Thresholds selected by knee-finding heuristic (unsupervised) and optimal F_1-score (supervised). The cosine similarity performance was more sensitive to s selection than INVITE.

vised context in the top row of Fig. 3). If parts of the underlying structure are known a priori, it is possible to tune s so that the F_1-score of $\mathbf{M}^s$ vs. the "test-set" (the true $\mathbf{M}$) is maximized. These can likewise be found on the bottom of Fig. 3

4.2 Exp. 2: Real-World Excavator MWOs

To assess the applicability of INVITE to real-world scenarios, we apply our model to tags annotated for a mining dataset (8264 MWOs) pertaining to 8 similarly-sized excavators at various sites across Australia [15, 48]. The tags were created by a subject-matter expert spending 1 hour of time in the annotation assistance tool `nestor` [49], using a methodology outlined in [50]. The tag annotations were limited to objects (bolt, motor, fan, *etc.*), problems (leak, missing, cracked, *etc.*), and solutions (replace, repair, stick, *etc.*) that occurred at least 50 times each in the original corpus, for a total of 77 unique tags. Subsequently, the same settings for solving Eq. 5 were used as in the previous experiment, though the optimization was initialized with the cosine similarity matrix to speed convergence.

To test whether the INVITE model was able to learn a robust representation of the system structure, we perform blind tests of the generative capability of each recovered network. First, the starting tag probability q was set as the observed distribution of first tags in the original dataset. Then, censored random walks of length $T = 5$ were sampled from both an INVITE and a cosine-similarity recovered network, without thresholding. This is intented to preserve weighted relationships between tags, for the purposes of data synthesis. The expert was then given a list of 100 randomly mixed MWOs, made from 40 real work-orders,[5] 30 INVITE censored lists, and 30 cosine-similarity censored lists. The resulting lists were filtered to only contain lists of tags not explicitly found in the original data. The expert was then asked to blindly classify each MWO as being "plausible" or "not plausible", such that an MWO would or would not reasonably occur based only on the tags in each. The fraction of real, INVITE, and cosine-generated work orders marked as "plausible" can be found in Fig. 4. Both the real and INVITE-synthesized MWOs are within a similar plausibility range, between $60\% - 80\%$, while the cosine similarity MWOs are between $40\% - 60\%$ plausible, overall.

5 DISCUSSION

To enable optimization in continuous space, our model does not enforce un-weighted, un-diriected graphs, as in Zemla & Austerweil's extension of the INVITE model that optimizes in discrete space [39]. Their version is intended to induce sparsity without artificially introducing new tuning parameters, as we have in introducing s and a. In practice, however, it is reasonable to select a low-valued positive $s \ll 1$, or use a knee-finding heuristic on the EDF of edge weights. This is because the L_1 penalty on edge weights, plus the tendency of INVITE to route random walks through commonly-visited chokepoint nodes, naturally drives "unnecessary" edge weights to near-zero probability. As seen in Fig. 5, the "knee" in the edge-weight EDF for INVITE was nearly always near-zero, for all experiments. Even when oracle information was allowed to tune s, the best F_1-score was still to be found strictly below $s = 0.3$, with a majority below 0.1. In this sense, the signal-to-noise ratio of our INVITE model is quite high, making the selection of a good s much less

[5]The process in [49] displays extracted concepts in order of their statistical importance, for the purposes of keyword recognition. As such, the annotator does not interact with the original work-orders directly during tagging.

difficult.

In truth, this thresholding does not completely solve the problem of knowledge extraction. If the goal of automatically extracting knowledge graphs is to suggest whether causal relationships *exist* between tags, and not primarily to synthesize data, weights are not needed in communicating these links, and may obfuscate important understanding below vastly more obvious relationships.

Another issue related to thresholding is how the row-stochastic constraint affects edge-value distribution: technically, each row in our model will be re-normalized every iteration, independent of other rows. This means that the row-normalization inherently de-symmetrizes $\mathbf{M}$. In reality, though we might model some asymmetry in node relations[6], the modality of direction being discovered by sentence-structure (ordering of the written tags) is not equivalent to the types of directionality we might want to discover. In memory, it could be beneficial to assume that the probability of transitioning between tags should be bi-directional, and allow desirable directionality to be proxied by local tree-like structures that reduce centrality of tags farther out. This bi-directional assumption implies making $\mathbf{M}$ a doubly-stochastic matrix. This has the added benefit of placing $\mathbf{M}$ on a simplex, *i.e.*, the space of permutation-invariant matrices belonging to the Birkhoff Polytope. There are recent developments [51, 52] in this space that could prove highly useful at reducing the state-space we search over.

Finally, our method is not intended to serve as a complete, end-to-end processing of natural language text into structured knowledge. Ultimately, the final structuring will need to be performed by humans. Instead, we believe the most efficient tools to assist in knowledge recovery will pose annotation questions in a lower-dimensional state-space, easier for a human to verify or edit *quickly*. Figure 6 illustrates how we believe INVITE takes steps toward this goal. Common structure recovery techniques, like cosine-similarity thresholding, tend to discover global structure quite well, but over-estimate the connectivity of local communities where hierarchical relationships are unknown-yet-assumed by the data. By definition, co-occurrence (bag-of-words) metrics are treating these work-orders more like un-censored random walks, starting from any tag and transitioning to any other in the list. Consequently, the local resolution of the structure it approximates is going to be fundamentally limited for tree-like communities, more reflecting a 2^{nd}- or 3^{rd}-order power graph[7] of the true, underlying structure.

In contrast, INVITE tends to concentrate edges to nodes that are highly central, forming "chokepoints" where global-scale transition mechanisms are unknown, but largely preserving local tree-like structures in outer communities. From an active learning perspective, humans tend to be quite good at verifying global-scale connections as viable or not—editing spurious connections in every over-dense local community is a much more difficult task for us than recognizing spurious individual connections to a small set of highly abstract concepts.

We believe this feature can be exploited to create better knowledge-structuring assistance tools in an active learning context. Such a tool could additionally benefit from a recent explosion in interest for preserving hierarchical relationships in vector space, *e.g.*, via Poincaré embeddings [54]. Additional care must be taken to allow flexible annotation of different kinds of relationships,[8] and allow for multiple (potentially disagreeing) annotators, subsequently suggesting relationship types for review. We envision a type of "topic model" over the space of knowledge graphs [55], or potentially a set of independent "graph components" that maximally explain the distribution of edge types in a community [56].

6 CONCLUSIONS AND FUTURE WORK

This paper presented a method to recover a structured representation of engineering knowledge from unstructured written documents (specifically, Manufacturing Work Orders), based on initial-visit emitting random walks (INVITE). Compared to previous methods, our technique preserves local connectivity structures, even in tree-like communities. This can lead to (1) better generative capability for synthesizing plausible documents (such as work-orders) in a simulation context; and (2) allowing us to cast the knowledge-structuring problem in probabilistic context that is potentially amendable to active-learning; this can minimize the number of local-scale edits needed relative to global-scale, abstract connections that humans can easily spot and correct.

Overall, the model we describe here can enable experts and novices alike to benefit from tacit system knowledge contained within frequently unused mountains of technical work-orders, by quickly prototyping computable representations of this knowledge for downstream usage in analysis pipelines. We believe that by explicitly incorporating cognitive theories into our modeling assumptions about how technicians might represent and then recall their knowledge in maintenance work-orders, we can accelerate the training and use of unsupervised data-driven expert systems in engineering design.

ACKNOWLEDGMENT

Thanks to Dr. Michael Brundage (NIST) for his efforts annotating and rating MWOs, and to Dr. Melinda Hodkiewicz (Univ. Western Australia) for providing the excavator data, and for many enjoyable and enlightening discussions on this topic.

[6] *e.g.*in hierarchies: "gear-1" may be a member of "gearbox," making the link gear-1 $\rightarrow$ gearbox a stronger link in a technician's head than the other direction.

[7] The graph G's n^{th}-order powergraph $P(G;n)$ has an edge between any two nodes if the minimum path length between those nodes in G is at most n.

[8] *e.g.*, Walsh *et al.*actually construct three types of structured system representations: functional, parametric, and component (which we use here)

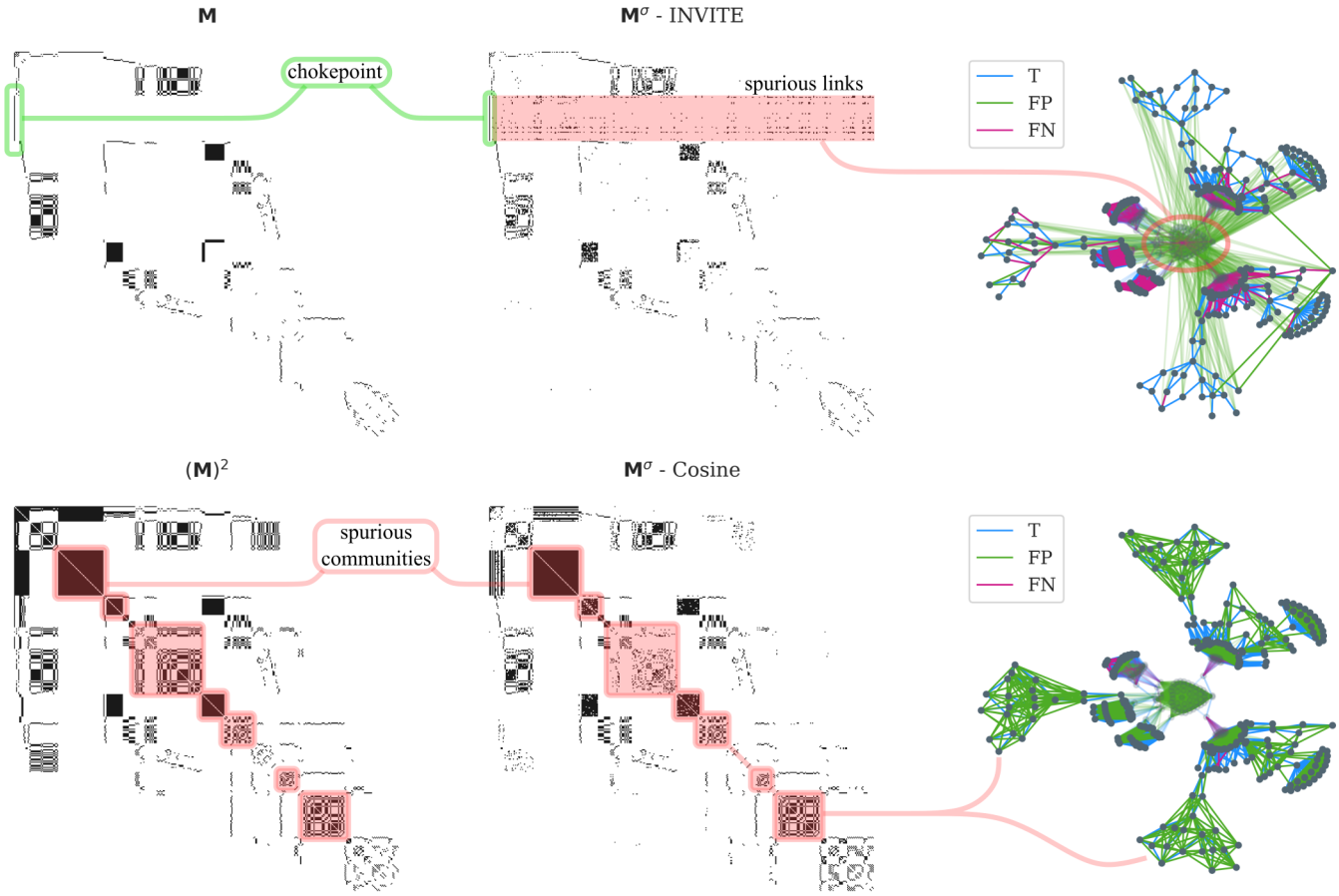

FIGURE 6: Comparing aircraft model network reconstruction for F_1-optimal INVITE (F_1 = 0.75) and Cosine Similarity (F_1 = 0.49) methods. Shown are the original "true" adjacency matrix $\mathbf{M}$, its 2nd-order power graph $(\mathbf{M})^2$, and the thresholded adjacency matrices $\mathbf{M}^s$ for both INVITE and Cosine Similarity. For visualization, the matrix rows/columns are sorted by the closeness centrality of each node [53], better indicating which nodes form core/integral components in the system (upper-left) and which are more likely a part of localized "edge" communities (bottom and right). These edge communities are highlighted in the graph layouts on the right, where nodes in the top 25th percent most central (and their edges) are transparent. INVITE directly estimates the underlying structure of $\mathbf{M}$ by accounting for node censoring in observations, concentrating uncertain edges into a few highly-connected "chokepoint" nodes. Cosine similarity mistakes co-occurrence of components in a sample for direct relationships, forming dense, spurious communities throughout the graph that are reflective of higher-order powers of $\mathbf{M}$, as shown here. Concentrating the false-positive relationships (FP) in a few highly central nodes makes INVITE a viable candidate for querying human experts for annotation/critique in an active-learning context.

7 DISCLAIMER

The use of any products described in this paper does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that products are necessarily the best available for the purpose.

REFERENCES

[1] Guimerà, R., and Sales-Pardo, M., 2009. "Missing and spurious interactions and the reconstruction of complex networks". *Proceedings of the National Academy of Sciences,* ***106***(52), pp. 22073–22078.

[2] Gomez-Rodriguez, M., Leskovec, J., and Krause, A., 2012. "Inferring networks of diffusion and influence". *ACM Transactions on Knowledge Discovery from Data (TKDD),* ***5***(4), p. 21.

[3] Linderman, S., and Adams, R., 2014. "Discovering latent network structure in point process data". In International Conference on Machine Learning, pp. 1413–1421.

[4] De Paula, Á., Rasul, I., and Souza, P., 2018. "Recovering social networks from panel data: identification, simulations and an application".

[5] Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al., 2018. "Relational inductive biases, deep learning, and graph networks". *arXiv preprint arXiv:1806.01261*.

[6] Raissi, M., Perdikaris, P., and Karniadakis, G. E., 2017. "Machine learning of linear differential equations using gaussian processes". *Journal of Computational Physics,* ***348***, pp. 683–693.

[7] Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., et al., 2017. "Visual genome: Connecting language and vision using crowdsourced dense image annotations". *International Journal of Computer Vision,* ***123***(1), pp. 32–73.

[8] Speer, R., Chin, J., and Havasi, C., 2017. "Conceptnet 5.5: An open multilingual graph of general knowledge". In Thirty-First AAAI Conference on Artificial Intelligence.

[9] ISO 15926-1:2004, 2004. Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities – Part 1: Overview and fundamental principles. Standard, International Organization for Standardization, Geneva, CH, July.

[10] Leal, D., 2005. "ISO 15926" life cycle data for process plant": An overview". *Oil & gas science and technology,* ***60***(4), pp. 629–637.

[11] ISO/TS 15926-8:2011, 2011. Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities – Part 8: Implementation methods for the integration of distributed systems: Web Ontology Language (OWL) implementation. Standard, International Organization for Standardization, Geneva, CH, Oct.

[12] Batres, R., West, M., Leal, D., Price, D., Masaki, K., Shimada, Y., Fuchino, T., and Naka, Y., 2007. "An upper ontology based on ISO 15926". *Computers & Chemical Engineering,* ***31***(5-6), pp. 519–534.

[13] Klüwer, J. W., Skjæveland, M. G., and Valen-Sendstad, M., 2008. "Iso 15926 templates and the semantic web". In Position paper for W3C Workshop on Semantic Web in Energy Industries; Part I: Oil and Gas.

[14] Kumar, N., Kumar, M., and Singh, M., 2016. "Automated ontology generation from a plain text using statistical and nlp techniques". *International Journal of System Assurance Engineering and Management,* ***7***(1), pp. 282–293.

[15] Hodkiewicz, M., and Ho, M. T.-W., 2016. "Cleaning historical maintenance work order data for reliability analysis". *Journal of Quality in Maintenance Engineering,* ***22***(2), pp. 146–163.

[16] Ho, M., 2015. "A shared reliability database for mobile mining equipment". PhD thesis, University of Western Australia.

[17] Eppinger, S. D., and Browning, T. R., 2012. *Design structure matrix methods and applications*. MIT press.

[18] Browning, T. R., 2016. "Design structure matrix extensions and innovations: a survey and new opportunities". *IEEE Transactions on Engineering Management,* ***63***(1), pp. 27–52.

[19] Ellinas, C., Allan, N., Durugbo, C., and Johansson, A., 2015. "How robust is your project? from local failures to global catastrophes: A complex networks approach to project systemic risk". *PloS one,* ***10***(11), p. e0142469.

[20] Robertson, S., 2004. "Understanding inverse document frequency: on theoretical arguments for idf". *Journal of documentation,* ***60***(5), pp. 503–520.

[21] Steyvers, M., and Griffiths, T., 2007. "Probabilistic topic models". *Handbook of latent semantic analysis,* ***427***(7), pp. 424–440.

[22] Blei, D. M., Griffiths, T. L., and Jordan, M. I., 2010. "The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies". *Journal of the ACM (JACM),* ***57***(2), p. 7.

[23] Mikolov, T., Chen, K., Corrado, G., and Dean, J., 2013. "Efficient estimation of word representations in vector space". *arXiv preprint arXiv:1301.3781*.

[24] Pennington, J., Socher, R., and Manning, C., 2014. "Glove: Global vectors for word representation". In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543.

[25] Sharp, M., Sexton, T., and Brundage, M. P., 2017. "Toward semi-autonomous information". In IFIP International Conference on Advances in Production Management Systems, Springer, pp. 425–432.

[26] Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., and Blei, D. M., 2009. "Reading tea leaves: How humans interpret topic models". In Advances in neural information processing systems, pp. 288–296.

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

[28] Vander Wal, T., 2007. Folksonomy. http://vanderwal.net/folksonomy.html.

[29] Specia, L., and Motta, E., 2007. "Integrating folksonomies with the semantic web". In European semantic web conference, Springer, pp. 624–639.

[30] Mousselly-Sergieh, H., Egyed-Zsigmond, E., Gianini, G.,

Döller, M., Kosch, H., and Pinon, J.-M., 2013. "Tag similarity in folksonomies". In INFORSID, Vol. 29, Inforsid, pp. 319–334.

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

[32] Henschel, A., Woon, W. L., Wachter, T., and Madnick, S., 2009. "Comparison of generality based algorithm variants for automatic taxonomy generation". In Innovations in Information Technology, 2009. IIT'09. International Conference on, IEEE, pp. 160–164.

[33] Schvaneveldt, R. W., Durso, F. T., and Dearholt, D. W., 1989. "Network structures in proximity data". In *Psychology of learning and motivation*, Vol. 24. Elsevier, pp. 249–284.

[34] Jun, K.-S., Zhu, J., Rogers, T. T., Yang, Z., et al., 2015. "Human memory search as initial-visit emitting random walk". In Advances in neural information processing systems, pp. 1072–1080.

[35] Sexton, T., Brundage, M. P., Hoffman, M., and Morris, K. C., 2017. "Hybrid datafication of maintenance logs from ai-assisted human tags". In 2017 IEEE International Conference on Big Data (Big Data), IEEE, pp. 1769–1777.

[36] Hills, T. T., Todd, P. M., and Jones, M. N., 2015. "Foraging in semantic fields: How we search through memory". *Topics in Cognitive Science,* ***7***(3), pp. 513–534.

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

[38] Doyle, P. G., and Snell, J. L., 2000. "Random walks and electric networks". *arXiv preprint math/0001057*.

[39] Zemla, J. C., and Austerweil, J. L., 2018. "Estimating semantic networks of groups and individuals from fluency data". *Computational Brain & Behavior,* ***1***(1), pp. 36–58.

[40] Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M., 2018. "Automatic differentiation in machine learning: a survey". *Journal of Marchine Learning Research,* ***18***, pp. 1–43.

[41] Maclaurin, D., 2016. "Modeling, inference and optimization with composable differentiable procedures". PhD thesis.

[42] Walsh, H. S., Dong, A., and Tumer, I. Y., 2019. "An analysis of modularity as a design rule using network theory". *Journal of Mechanical Design,* ***141***(3), p. 031102.

[43] Kingma, D. P., and Ba, J., 2014. "Adam: A method for stochastic optimization". *arXiv preprint arXiv:1412.6980*.

[44] Bottou, L., 2012. "Stochastic gradient descent tricks". In *Neural networks: Tricks of the trade*. Springer, pp. 421–436.

[45] Satopaa, V., Albrecht, J., Irwin, D., and Raghavan, B., 2011. "Finding a" kneedle" in a haystack: Detecting knee points in system behavior". In 2011 31st International Conference on Distributed Computing Systems Workshops, IEEE, pp. 166–171.

[46] Haley, B. M., Dong, A., and Tumer, I. Y., 2016. "A comparison of network-based metrics of behavioral degradation in complex engineered systems". *Journal of Mechanical Design,* ***138***(12), p. 121405.

[47] Saito, T., and Rehmsmeier, M., 2015. "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets". *PloS one,* ***10***(3), p. e0118432. An optional note.

[48] Hodkiewicz, M. R., Batsioudis, Z., Radomiljac, T., and Ho, M. T., 2017. "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 2017.

[49] Madhusudanan Navinchandran, F., Bones, L., Brundage, M., Hoffman, M., Moccozet, S., and Sexton, T., 2018. Nestor: a toolkit for quantifying tacit maintenance knowledge, for investigatory analysis in smart manufacturing.

[50] Sexton, T., Hodkiewicz, M., Brundage, M. P., and Smoker, T., 2018. "Benchmarking for keyword extraction methodologies in maintenance work orders". In PHM Society Conference, Vol. 10.

[51] Adams, R. P., and Zemel, R. S., 2011. "Ranking via sinkhorn propagation". *arXiv preprint arXiv:1106.1925*.

[52] Linderman, S. W., Mena, G. E., Cooper, H., Paninski, L., and Cunningham, J. P., 2017. "Reparameterizing the birkhoff polytope for variational permutation inference". *arXiv preprint arXiv:1710.09508*.

[53] Sabidussi, G., 1966. "The centrality index of a graph". *Psychometrika,* ***31***(4), pp. 581–603.

[54] Nickel, M., and Kiela, D., 2017. "Poincaré embeddings for learning hierarchical representations". In Advances in neural information processing systems, pp. 6338–6347.

[55] Gerlach, M., Peixoto, T. P., and Altmann, E. G., 2018. "A network approach to topic models". *Science advances,* ***4***(7), p. eaaq1360.

[56] Park, B., Kim, D.-S., and Park, H.-J., 2014. "Graph independent component analysis reveals repertoires of intrinsic network components in the human brain". *PloS one,* ***9***(1), p. e82873.