Organizing Tagged Knowledge: Similarity Measures and Semantic Fluency in Structure Mining

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

Recovering a system's underlying structure from its historical records (also called structure mining) is essential to making valid inferences about that system's behavior. For example, making reliable predictions about system failures based on maintenance work-order data requires determining how concepts described within the work order are related. Obtaining such structural information is challenging, requiring system understanding, synthesis, and representation design. This is often either too difficult or too time-consuming to produce. Consequently, a common approach to quickly eliciting tacit structural knowledge from experts is to gather uncontrolled keywords as record labels—i.e., "tags." One can then map those tags to concepts within the structure and quantitatively infer relationships between them. Existing models of tag similarity tend to either depend on correlation strength (e.g. overall co-occurrence frequencies), or on conditional strength (e.g. tag sequence probabilities). A key difficulty in applying either model is understanding under what conditions one is better than the other for overall structure recovery. In this paper, we investigate the core assumptions and implications of these two classes of similarity measures on structure recovery tasks. Then, using lessons from this characterization, we borrow from recent psychology literature on semantic fluency tasks to construct a tag similarity measure that emulates how humans recall tags from memory. We show through empirical testing that this method combines strengths of both common modeling paradigms. We also demonstrate its potential as a pre-processor for structure mining tasks via a case study in semi-supervised learning on real excavator maintenance work-orders.

Nomenclature

ML	Machine Learning
NLP	Natural Language Processing
MWO	Maintenance Work Order
DSM	Design Structure Matrix
RW	Random Walk
INVITE	Initial-visit Emitting (Random Walk)
SGD	Stochastic Gradient Descent

1 INTRODUCTION

Many engineering and design tasks rely on having an accurate representation of a system's structure. This *structured knowledge*, made up of concepts and concept-relations, can then be used to create more reliable models for engineering learning tasks. For example, such structures include ontologies for industrial data and reliability analysis [1–3], Design Structure Matrices (DSMs) for quantitative design of complex systems [4–6], or rule-sets for normalizing reliability data for e.g. survival analysis [7–9]. Though some effort has been spent automating the process of building these "knowledge structures" [10], even these cases require significant manual effort to collate validated vocabularies and syntactical rules; in general, obtaining such structured knowledge can be challenging since closed form descriptions and characterizations of structure are often either too difficult or too time-consuming to produce. Manual construction of bespoke, application-specific engineering ontologies are often cost-prohibitive to create and maintain, and the use of general purpose concept networks [11–13] often lack needed domain knowledge.

[*] Address all correspondence to this author.

In light of these difficulties, many researchers have realized a need to rapidly acquire this structured knowledge from their staff's expertise, whether through elicitation [14], or by *learning* from their data (i.e., historical records). The latter is often easier or more reliable to collect from experts when time-constraints and demanding responsibilities play a significant role in data creation. This process of learning structured data from written historical records is often referred to as *structure mining*, or in the machine learning community, a special case of representation learning on discrete data (e.g. graphs) [15, 16].

In technical fields like engineering, design, and manufacturing, performing structure learning faces two key difficulties that this paper helps address. First, performance of existing structure learning approaches hinges on an appropriate definition of similarity among concepts. As we describe in Section 2, common choices for this similarity fall into two camps—correlation versus conditional strength. This paper compares the merits of both approaches and demonstrates conditions under which both struggle to accurately infer ground-truth structure (§4). Second, available historical data is often difficult to use directly; the domain experts creating it generally assume it will be read and adapted by colleagues or other experts in their own field. This means an analyst 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 meaning.

This paper addresses this problem by adapting models of memory recall in psychology to posit a statistical model that accounts for how experts may generate tags given prior experience or context (§3). This model also forms a middle-ground between existing similarity measurement tools, and sheds light on the differences among those models.

The next sections describe our perspective on the use of structure learning while dealing with tags and historical records —i.e., using Maintenance Work Orders to infer system structure. We use this concrete example to highlight why structure learning is difficult, what practical issues one faces when evaluating such techniques, and then summarize the paper's key research questions.

1.1 Example of Maintenance Work Orders and Tags

In contexts where annotation is costly, significant research has been done to empower casual annotators, and understand how natural classification and labeling schemes arise in social communities. When restricted vocabularies and categories for record annotation are not available or practical, users are often allowed to assign uncontrolled keywords to a record, a process referred to as "Tagging". This allows concepts to be derived freely in the course of work, as repeated and cross-contextual usage, often among multiple users, leads to a naturally-arising set of useful, domain-specific concepts [17–20].

Historical Record (MWO) Annotation Comparison

"Hydraulic Leak at cutoff unit;
Missing fitting replaced"

Categorization:	
Subsystem	`142_HYD_SYSTEM`
Error Code	`ERR_142A`
Action Taken	`PART_ORDERED`
Tags:	
objects	cutoff_unit, hydraulic, fitting
problems/actions	leak, replace

This freedom implies that "tags" have not been directly controlled—that is, picked from a fixed list known ahead of time. They lack a designed model of individual tag *relationships*. Therefore the crucial step required to use tags for structure mining is to determine relationship strength: mathematically modeling pairwise tag similarities (or conversely, distances). As will be discussed in Section 2, methods for approximating concept relationships in unstructured multisets like tags vary widely, and have a variety of implications.

1.2 Evaluating Similarity Measures between Tags

Because so many down-stream structure mining and analysis tools require some underlying assumption of what makes concepts similar, it is important to consider the impact of selecting a similarity model.

How does one evaluate whether a given similarity measure is "good" for a given problem? To unpack common evaluation measures, assume we can represent our system structure as a weighted graph $G = \{V, E\}$, where the node-set V represents concepts in our system (assumed to be known, one for each tag), and the edges E are weighted based on similarity between these. Our modeling assumptions can influence two key properties of E that are crucial to a successfully recovered structure:

Precision & Recall: Detected relationships should be distinctly recognizable, and those detections should be reliably useful. This implies graph sparsity, and ensures that indirect similarity through an intermediary concept is not conflated with true similarity. Any non-zero edge weight should therefore correspond to a real concept connection of some kind. In other words, *of the detected edges, most should be relevant*, and *of relevant edges, most should be detected*.

Robustness: Since structure is not known a priori, some amount of filtering on edge weights will take place to enforce the previous properties. The quality of a recovered structure should be robust to changes in filter strictness. This implies *relevant edges should not be quickly lost as an increased edge-weight thresholds remove unwanted detections*.

1.3 Research Questions

This paper investigates the performance of two common similarity measures with respect to these traits: co-occurrence frequency-based (typified by cosine similarity), and conditional sequence probabilities (typified by k^{th}-order Markov chains). We then show that, while each has strength in one of the above desideratum, the other can be lacking. This necessitates a hybridized approach to "interpolate" between the two measures. To accomplish this, we frame the act of tagging historical records, typified by maintenance work orders (MWOs), as a type of semantic memory recall from within the expert's internal "knowledge-graph"—this leverages the concept of *semantic fluency*, which we define in §3. Specifically, we investigate:

R1 Whether incorporating mechanisms for non-Markovian jumps improve the precision and recall of structure recovery compared to frequency-based or Markovian relationship measures

R2 Whether the relationship-graph learned through this model shows improved accuracy of learning tasks that require an assumed similarity measure, compared to the traditional measures.

We empirically test (1) precision and recall for learned similarity measures from multiple synthetically generated tag data-sets (using several known structures) via a memory-recall model; and (2) semi-supervised concept classification using tagged maintenance work orders from a mining excavator operation, not having a previously known concept-relationship structure.

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 achieve significantly better performance (as measured by precision and recall) than existing methods of relationship recovery.

2 RELATED WORK

Using data to infer the underlying structure of a complex system is a long-standing goal within domains that depend upon accurate network recovery, such as: biological systems and disease transmission vector modeling [21,22]; uncovering economic interactions and social networks [23,24]; inferring physical models by learning governing equations [25]; or even description generation in computer vision, and quantifying how humans reason about belonging and causality in ambiguous images or contexts [12,13]. It is beyond the scope of this work to exhaustively compare state-of-the-art in representation learning[1]; still, a common theme found among these techniques is an assumed definition for the "distance" between observed data. For numerical data, a common assumption is that distance between observations with N features is an L-norm between the N-dimensional vectors (e.g. Euclidean distance being the L_2-norm), though often a more robust characterization of distances exists on a lower dimensional manifold embedding within that space [26].

Learning useful structures from non-numerical data, like tags or networks, is a rapidly progressing research area. From a mechanism for extracting latent taxonomies from tagged documents [27], to extracting interconnected term- and topic-hierarchies through nested stochastic block models [28] or hyperbolic embeddings [29, 30]. Once again, all of these tools assume an a priori estimate of what being "related" means: how similarity and distance are defined in the latent feature space. Therefore, to make the best use of these burgeoning tools, it is paramount to characterize the impacts of one's chosen similarity measure, and ensure that the choice matches well with properties of the data and subsequent models being used.

2.1 Global Frequency and Context

A common way to encode similarity between observations with discrete-valued features (whether tags, graphs, or natural language documents) starts with making the intuitive assumption that features occurring across similar contexts *are similar.* This style of similarity measure naturally arises when using frequency-based mathematical representations of text via natural language processing (NLP). These include "bag-of-words" weightings [31], topic models [32, 33], or semantic vector embedding [34, 35]. In these vector representations of an observation, then, the similarity between two observations is less about how "close together" the co-occurrence frequency magnitudes are, and more about occurrence frequency correlations —the vector *direction* similarity. This is encoded in the cosine similarity measure, i.e. the cosine of the angle between the vectors.

Rather than a corpus of documents, we are concerned specifically with the set of tags assigned to records. This set of tags, especially when created by multiple users, is commonly referred to as a *folksonomy*, a portmanteau of "folk" and "taxonomy" [36]. 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, much like words in text. The predominant way to analyze tag similarity, then, is by their co-occurrences with each other [37,38]. If, over a set of C records, 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 is applied across many NLP and folskonometric methods to structuring relationships between tagged concepts in useful ways, including the taxonomy extraction and hyperbolic embedding work mentioned above [27, 30, 39]. For this work, while significant advances have been made in contextual, set-based measures on e.g. topic models or semantic embeddings, the latent relations being 'learned"

[1] readers are directed to the literature review by Bengio et al. [16].

are quite often difficult to interpret for humans [40], stemming from the so-called "black box" nature of these models. We therefore make use of Equation 1 for ease of interpretation and broad acceptance.

The power of cosine similarity comes from its computational simplicity, and an ability to deal with high-dimensional feature sets (*e.g.*the set of all unique tags in a folksonomy). These context-based similarity measures (which also include Jaccard similarity, mutual information, and the like), base their approach on treating tags as unordered sets. This has the distinct advantage of picking up on un-obvious relationships between tags that co-occur in wildly varying contexts, quickly recovering global-scale structures with minimal observations. We should expect that most relevant relationships are quickly retrieved this way, i.e. cosine similarity typically exhibits a high *recall* score in structure recovery.

However, one can imagine adding a tag to a document that is related to, say, the previously added tag, but not necessarily to the first tag added; so when is co-occurrence a coincidence? This line of reasoning implies a separate model, where annotating each tag implies a probability to use or not use some subsequent tags.

2.2 Local Sequence Probability

On the opposite side of treating tags or text as an unordered set, one might think of tagging as a sequential stream of tag additions. Once again taking a cue from NLP, one might assume that each subsequent concept written in text is directly conditional on what was written previously. Predicting the probability of observing a word based on the previous, n locally-observed words (in order) is known as an n^{th}-order language model [41].

For tags, say assigning a tag to a document is equivalent to being in that tag's "state", and the relations between states is the probability of transitioning between those states. Assigning tags would then be a process satisfying the Markov property; thus, for an n^{th}-order tag Markov model, the probability of observing any i^{th} tag in a sequence is $P(t_i|t_{i-1},\cdots,t_{i-n})$. In practice, given a data-set of observed tag sequences, this means finding the maximum likelihood estimate for transition probabilities between tags, in the form of conditional probability tables.

This is a powerful (though over-simplifying) model, and many techniques seek to apply a similar reliance on the sequential nature of textual language or tagging to predict subsequent relevant tags. Hidden Markov Models (HMMs), for instance, treat each state as a *distribution* of tag "emmission" probabilities, and train to find transitions between these distributions. These are often used both for tag recommendation, and for predicting other system feature relationships with tags or keywords [42]. Other success has been found using recurrent neural networks as language models, capable of storing sophisticated, long-distance contextual information while predicting a sequence [43].

Because the intuition behind the these sequence-based models comes from nearby tags having a strong influence on each other, one way to quickly estimate the relationship strength of two tags is to estimate the probability of observing them in sequence:

$$s(t_1,t_2) = \max\left[P(t_1|t_2), P(t_2|t_1)\right] \tag{2}$$

This preserves symmetry in the similarity measure, allowing us to compare it to the cosine similarity above. Since our similarity is calculated from a sequence, and the model is estimated only from observed sequences, we expect a high fraction of total predicted relationships to be truly relevant, i.e., precision score should be high.

Still, what if tag relationships exist that are rarely observed directly, due to an third, highly common tag? What if there are biases in tag ordering due to quirks of user reporting? Rather than having to choose between skewing toward recall or precision, is there a model that more naturally fits the mechanics of tagging, to avoid systematic failure to improve either metric?

3 MODELING TAGS AS MEMORY RECALL

As discussed above, common techniques for discovering structural relationships in tagged data primarily rely on either frequency and co-occurrence information, or conditional sequence probabilities of discrete objects/concepts. These are powerful and easy-to-apply models used ubiquitously for speech or the written word, but can also lead to systematic misbehavior under the conditions that user taging presents.

Instead, this paper tries to address shortcomings in relationship recovery by explicitly emulating the dynamics of how humans might recall concepts from memory, and apply this memory recall to estimating tag relationship structures.

This section first describes the concept of *semantic fluency tests*—an existing tool in psychology literature for testing concept-relationship recall—and how the surrounding theory relates to tagging engineering records. We then describe a computational method to implement the concept of semantic fluency using Initial-Visit Emitting Random Walks (INVITE) [44]—a non-Markovian probabilistic model for sampling semantic-fluency-type data from an underlying concept-relationship network.

3.1 Semantic Fluency

When a user begins to tag a record, they try to search their memory for concepts that are relevant to the record itself, in the context of the engineered system it pertains to. In the interest of recovering latent relationships between system components as understood by, e.g. a technician, we restrict our discussion on tags to ones representing objects/items directly (though they may additionally concern problems that were encountered with some items, or how other items were used to solve these problems [20]).

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 [45] 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,[2] 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 concept and the next. By thresholding this high-dimensional association "map", binarizing it as "is related"/"is not related",we can represent this map as a graph[3], where concepts are nodes and an edge represents "is related". Memory recall, then, consists of a sort of "walk" along this graph.

A classic psychological experiment to measure what such a graph might look like is the Semantic (or, Verbal) Fluency test. Given an object type (*e.g.*, animal):

1. Recall and record an object of that type;
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 themselves. One advantage of this test lies in not restricting (or having to specify a priori) the relationship between objects required to record subsequent ones. For example:

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

As in this example, it is common for animal-based semantic fluency lists to start with household pets, potentially switching to 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").

Altering the scope of such a task to "system object that is relevant to a given record" instead of "object that is an animal," represents a task that is remarkably similar to how the user tagging task was construed in previous sections. In this model, each subsequent tag assigned to a record constitutes a"jump" in the user's internal "tag network", which depends in some way upon previous tag jumps for that record.

Thus, any attempt to recover the associative strength between concepts should necessarily incorporate these context "jumps" (canines, big cats, household pets, African animals) in a way that allows for "retracing your steps" to previous concepts when exploring some new context. One model that incorporates these precise features mathematically is the recently proposed INVITE model [44].

3.2 Initial-Visit Emitting Random Walks

The described semantic fluency model for tagging boils down to two key components of a user's cognitive task when recalling relevant tags:

- They submit tags sequentially, as they recall **unique** defining concepts related to the record.
- They recall each concept by traversing relationship links between **it**, and **any** recently recalled concepts.

Fig. 1 Illustrates such traversals by using a drive-train component network from Walsh et al. to stand in for a user's latent understanding of a system's structure. In that figure, each "MWO" begins with a some initially sampled tag, with subsequent tags potentially stemming from a "jump" to distant (non-adjacent) nodes in the network. This illustrates hidden jumps due to initial-visit censoring. The resulting tags could still be reasonable for a MWO where those components were involved: Example #1 could represent the text *"Had to replace bearing retainer; bearing balls showed excess wear. Inner and outer bearing races cleaned."* Despite not being directly connected, they share a common region in the graph, with each subsequent tag accessible in memory from *one of the previous tags*.

This differs from a standard Bag of Words model—where all tags are assumed to be linked through co-occurrence on a record (i.e. only global graph topology matters), and from n^{th}-order Markov models—where tag relations are limited to the nearest (or, previous) n entities (i.e. only local sequences of observed tags matter). Additionally, in neither of these models are tags explicitly modeled as unique within the record.

This illustrates nicely the trade-off between categorical and associative memory foraging that [45] discusses at length, and is precisely the feature of tagging we investigate when extracting a more realistic representation of tag relationships through the mathematical framework of Initial-Visit Emitting Random Walks.

Say the set of components or concepts that have a corresponding tag in our system is denoted by the node-set N. A user-given set of T [4] for a specific record can be denoted as a Random Walk (RW) trajectory $\mathbf{t} = \{t_1, t_2, t_3, \cdots t_T\}$, where $T \leq N$. This limit on the size of T assumes tags are a set of unique entries: any transitions between previously visited tags 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 *through previously visited ones*.

Instead of directly computing over this intractable model for generating $\mathbf{t}$, the key insight from the original INVITE paper [44] comes from partitioning $\mathbf{t}$ into $T-1$ Markov chains with absorbing states, where previously visited tags are "transient" states, and unseen tags 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 among themselves $\mathbf{Q}^{(k)}_{q\times q}$,

[2] 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` [34, 35].

[3] Also called an *associative network [46]*.

[4] While some sources use "tagging" as a proxy for a set of strictly unordered labels (as in multi-label classification), we preserve the mechanism by which the tags were generated in the first place, i.e., in a *specific* order.

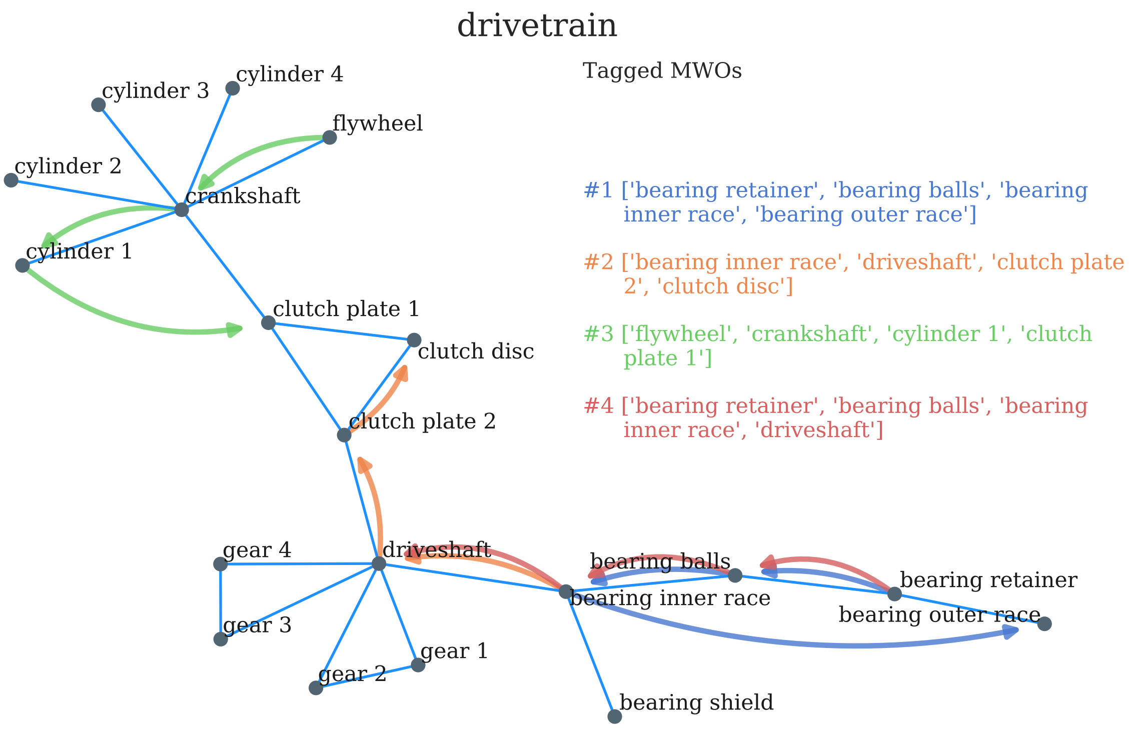
drivetrain
Tagged MWOs
#1 ['bearing retainer', 'bearing balls', 'bearing inner race', 'bearing outer race']
#2 ['bearing inner race', 'driveshaft', 'clutch plate 2', 'clutch disc']
#3 ['flywheel', 'crankshaft', 'cylinder 1', 'clutch plate 1']
#4 ['bearing retainer', 'bearing balls', 'bearing inner race', 'driveshaft']
cylinder 4
cylinder 3
flywheel
cylinder 2
crankshaft
cylinder 1
clutch plate 1
clutch disc
clutch plate 2
gear 4
driveshaft
bearing balls
bearing retainer
bearing inner race
bearing outer race
gear 3
gear 1
gear 2
bearing shield

Fig. 1: Exmaple observations of INVITE samples on a drive-train network model from Walsh et al [47].

and r absorbing states with transitions into them from q as $\mathbf{R}^{(k)}_{q\times r}$, the Markov transition matrix $\mathbf{M}^{(k)}_{n\times n}$ has the form

$$\mathbf{M}^{(k)} = \begin{pmatrix} \mathbf{Q}^{(k)} & \mathbf{R}^{(k)} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \tag{3}$$

where $\mathbf{0}$, $\mathbf{I}$ represent lack of transition between/from absorbing states. It follows from [48] 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

$$P\left(t_{k+1}|t_{1:k}, \mathbf{M}\right) = \mathbf{N}^{(k)} R^{(k)}\Big|_{q,1} \tag{4}$$

The probability of being absorbed at $k+1$ conditioned on jumps $1:k$ is thus equivalent to the probability of observing the $k+1$ INVITE tag. If we approximate an a priori distribution of tag probabilities to initialize our chain as $t_1 \sim \mathrm{Cat}(n, \theta)$ (which could be empirically derived or simulated), then the likelihood of our observed tag chain $\mathbf{t}$, given a transition matrix, is

$$\mathcal{L}\left(\mathbf{t}\,|\,\theta;\mathbf{M}\right) = \theta(t_1)\prod_{k=1}^{T-1} P\left(t_{k+1}\,|\,t_{1:k}\,;\mathbf{M}\right) \tag{5}$$

Finally, if we observe a folksonomy of tag lists $\mathbf{C} = \{\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_c\}$, and assume θ can be estimated independently of $\mathbf{M}$, then we can frame the problem of structure mining on observed INVITE data as a minimization of negative log-likelihood of our folksonomy given $\mathbf{M}$:

$$\mathbf{M}^* \leftarrow \underset{\mathbf{M}}{\arg\min} \quad \sum_{i=1}^{C}\sum_{k=1}^{T_i-1} -\log\mathcal{L}\left(t^{(i)}_{k+1}\middle|t^{(i)}_{1:k}, \mathbf{M}\right) \tag{6}$$

3.3 Implementation

As formulated in Eq. 6, 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 [44]. 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.

$$M_{i,j} \leftarrow \frac{\exp\left(M_{i,j}\right)}{\left[\sum_j \exp\left(\mathbf{M}_i\right)\right]_j}$$

We introduce a modification to this re-parameterization. Eq. 6 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) while learning from samples, as in [49]. Put simply, one-directional relationships can be useful to model when they are largely the case (*e.g.*, cat→lion), 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 speed-up recovery of what we assume is a "symmetry-dominant"

M, we can bias the optimization toward symmetry via an update to each entry prior to the softmax step:

$$M_{i,j} \leftarrow \max\left\{M_{i,j}\,,M_{j,i}\right\} \tag{7}$$

In folksonomies where the recovered weights in each direction are known to be meaningful, this can be skipped.

4 EXPERIMENTS

Per the above discussion, the following experiments and case studies are done by comparing the recovered similarity measures, in the form of tag-relationship graphs, between a cosine similarity measure, 1st- and 2nd-order Markov chain models, and the proposed INVITE-based similarity model.

To address R1 from §1.3, the first experiment demonstrates the effectiveness of incorporating mechanisms from the INVITE model when tag-style data is generated in the manner of semantic fluency tests. We synthesize tagged records as censored random walks on a sample of random small-world networks, as well as on networks representing real engineering systems, as described in [50].

We use these synthetic tags to (1) measure the network recovery accuracy of the various similarity measure models using standard information retrieval metrics, (2) determine the ability of INVITE-based similarity to hybridize precision and recall efficiency of the other models, and (3) Illustrate qualitatively the key failure modes of various modeling assumptions when INVITE mechanics are not taken into account.

In the second experiment, addressing R2, we determine the performance of the similarity measures as pre-processing steps to accomplishing a semi-supervised tag classification task. We utilize a folksonomy of real, tagged excavator MWOs, for which a "true" underlying system structure is not known a priori. Classification scores and divergence from true multinomial tag classification distributions are presented.

For all experiments, we address the way in which different models perform under similarity *thresholding*. Thresholding is important since, as is universally the case in representation learning, we do not generally have a "ground-truth" representation to tune parameters against. As described briefly in §1.2, it is the performance characteristics over a *range* of thresholds that we seek to improve. After normalizing the relationship strength of any given edge into the range $M^*_{i,j} \in [0,1]$, we threshold **M** such that, for a given threshold value $\sigma \in [0,1]$, the entries of a thresholded similarity matrix $\mathbf{M}^\sigma$ are given by:

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

These networks should be sparse, to be informative about the existence of important relationships while ignoring noisy ones. This implies class-imbalance between edges and non-edges as target predictions. For imbalanced learning problems like this, precision (P, the ratio of true-positive edges to total detected edges) and recall (R, the ratio of true-positive edges to total true edges) at each threshold can elucidate model robustness under varying threshold sensitivities [5] [51]. Combining both into a single metric for balancing these two desirable traits is primarily done using an F_β-*measure*:

$$F_\beta = (1+\beta^2)\frac{PR}{\beta^2 P + R} \tag{8}$$

In this paper, we use the most common case of $\beta = 1$ to equally balance the importance of precision and recall.

Because of the alterations described in §3.3, the analytic gradient for the INVITE loss function described in [44] no longer applies; instead, we make use of automatic differentiation as a means to ensure accurate gradient calculations under these modifications. The package `PyTorch` [52] was used for for optimization with automatic differentiation, in the Python programming language. For calculating maximum likelihood estimates for the Markov-chain models, we have made use of the Python package `pomegranate` [53]. Code will be made available in an associated repository for reproduceability[6].

4.1 Exp. 1: Recovering Known Networks

To validate the ability of our method to accurately reconstruct engineering networks compared to other methods, we first synthesize censored tag lists from true tag-relationship networks under a variety of conditions.

Randomized Graphs Random graphs were generated, consisting of Watts-Strogatz randomized connections between $N \in \{10,25,30\}$ nodes. For the purposes of comparison across networks, the mean degree was set as $K_{WS} = 4$ with the re-wiring coefficient set to $\beta_{WS} = 0.166$ [54] [7]. Then, synthetic folksonomies were generated consisting of $\|C\| \in \{10,25,30\}$ "tagged documents" (*i.e.*censored random-walks on a given graph). In this experiment, each document/random-walk was assigned $\|T\| = 4$ tags. The median F_1-score across the for 10 different graphs are shown for each N/C combination in Figure 2. The precision and recall curves are also shown, collapsed over all 90 random graphs.

Discussion As measured by F_1-score, the INVITE-based similarity measure consistently out-performs both the Markov chain and cosine similarity measures across a wide range of thresholds, for all graph/random walk settings. More interesting, and more useful for practitioners in an unsupervised setting, is the *shape* of these curves, and how

[5] Recall is alternatively known as *sensitivity*, while precision is alternatively known as Positive Predictive Value (PPV).

[6] `https://github.com/tbsexton/organizing-tags`

[7] This Watts-Strogatz setting, while not necessary for the purposes of our experiment, can give networks with experimentally similar properties to real cognitive associative networks; see [49].

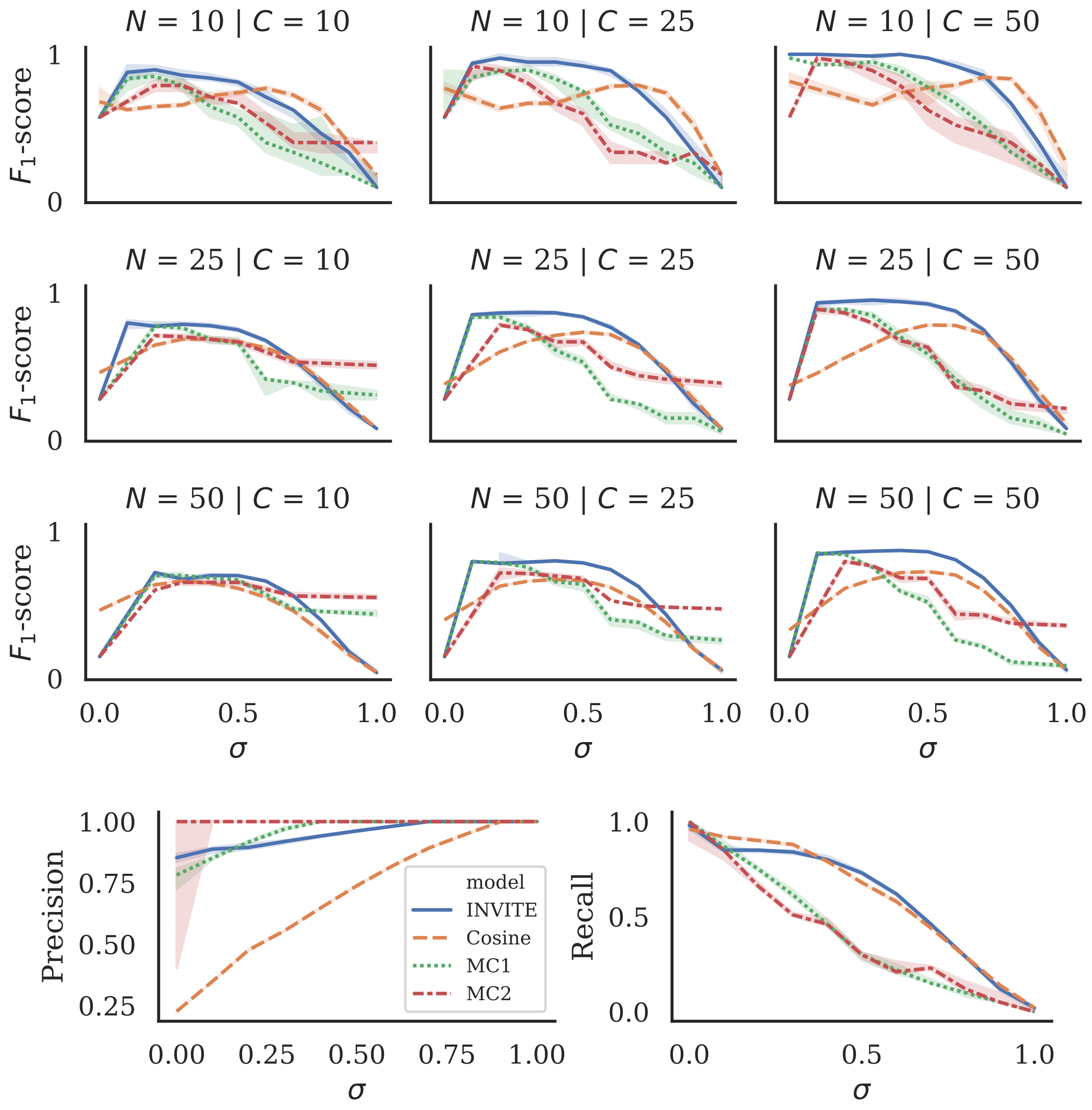

Fig. 2: **Top**: F_1-scores for various combinations of network size N and number of "tagged records"/random-walks C, shown as a function of similarity threshold σ. Median over 10 trials for each setting, shown with 95% confidence interval (1000 bootstrap samples). **Bottom**: Precision and recall across all 90 trials, for all 9 setting combinations.

they change. For low-complexity networks, cosine similarity is relatively stable over all thresholds. Then, as complexity increases, much more filtering has to take place (higher σ) before it reaches best performance. This is contrary to the sequence-based Markov chains, which show dramatically better performance at thresholds *barely* above 0, but suffering at higher specificity.

Meanwhile, the INVITE-aware similarity shows a sharp increase at low-σ, like the markov-model, while retaining the smoothness of the cosine model as σ is tuned higher. This tendency to capture the strengths of each is more clear if precision and recall are shown separately, as in the bottom of Figure 2, where the precision behavior of the INVITE model matches that of the Markov similarity (it's presumed strength; recall §1.2). At the same time, it's recall behavior more closely resembles that of the cosine similarity model (again, the strong-suit of that paradigm).

This dynamic—the trade-off between models that favor recall vs. precision, can be made clearer with a concrete example.

Real System Networks To qualitatively understand the underlying failure/success modes of each measure, we turn to the real system networks presented in [47, 50]. We start with their drivetrain model ($N = 18$), which is simple enough for visualization while demonstrating common patterns in engineered systems. We sample $C = 20$ random walks of length $l = 4$, some of which were used above in Figure 1.

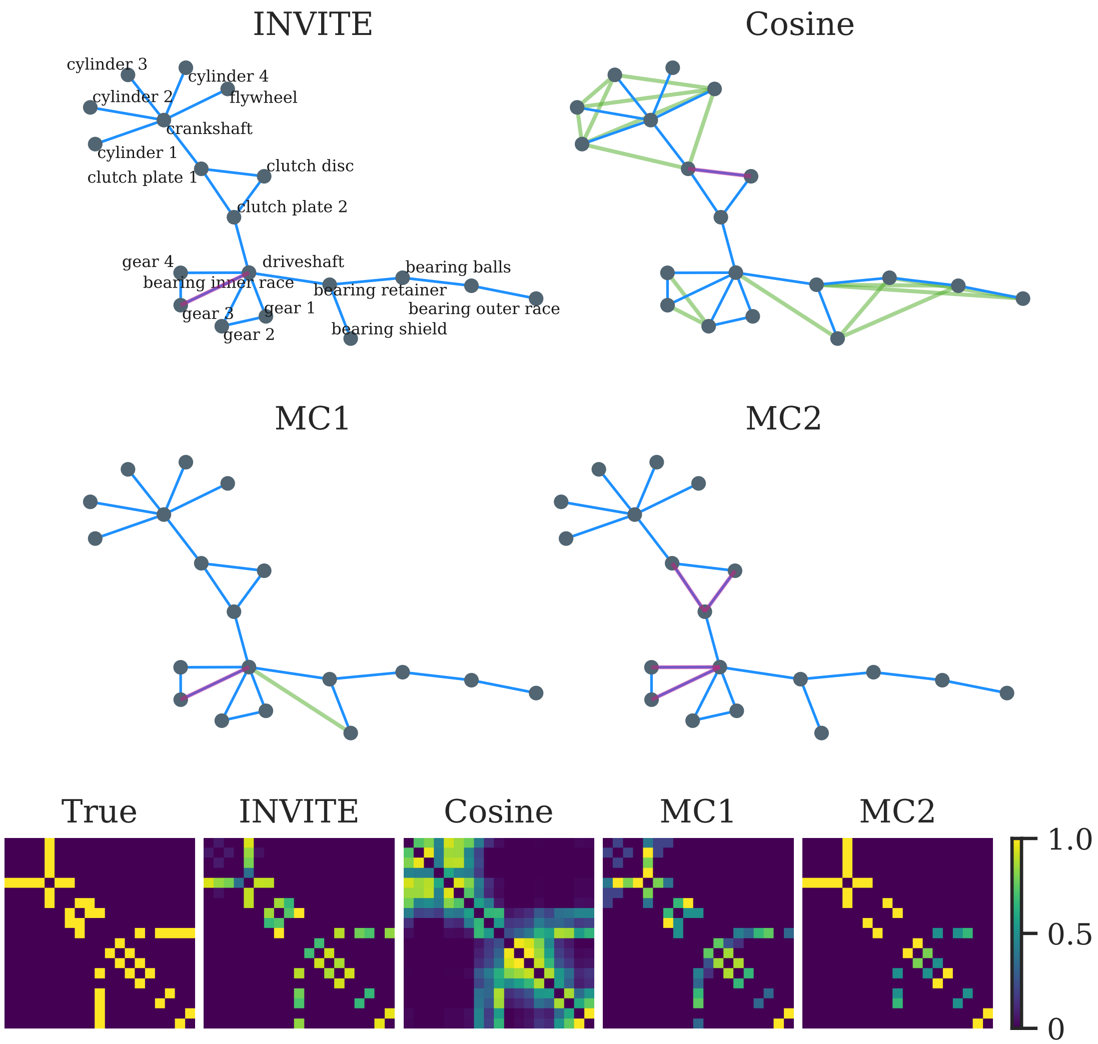

Fig. 3: F_1-optimal thresholded networks, with un-threshholded adjacency-matrix representations of M^*.

These settings were chosen intentionally as low-performing, to illustrate failure modes in each model type in Figure 3, where optimal F_1-thresholded networks are shown with false positive (green) and false negative (red) edge predictions, along with the un-thresholded similarity matrix $\mathbf{M}^*$.

The cosine similarity model has quickly detected *all* relevant edges, as seen in the un-thresholded matrix, but it has also over-estimated the connectivity of local communities. Engineered systems often display hierarchical connectivity patterns, where many low-level parts are only similar indirectly because of their connection to a key higher-level component (*e.g.*all cylinders and the flywheel to the crankshaft, or all gears and bearing inner race to the driveshaft). Because the observed transitions are censored from seeing previously visited nodes—just like a user only tags each concept once, even while they continue to use it to recall other concepts —the cosine model sacrifices higher-component connections to preserve the perceived frequency with which low-level tags co-occur.

The Markov models, on the other hand, demonstrate remarkably few false-positives. Instead, accuracy is limited by the number of available observations for each sequence of two or three tags. Since the number of possible paths is so large, true relationships might only be realized as a direct sequence a single time, or not at all, when so little data is available. This means true edges are quickly lost when the model's certainty about their existence is no better than false edges detected in a censored INVITE jump. The INVITE model balances aspects from both models, by quickly gaining certainty about the overall structure, while still allowing for exploration to re-route potential connections through edges that make more sense *sequentially*.

For the interested reader, the same exercise was performed on a reduced version of the Airplane network, also from [47, 50]. Due to complexity of the visualization, nodes with identical names (barring a numerical identifier) were merged into a single concept-tag. Figures 6 and 7 in the supplementary materials show F_1-scores, precision-recall curves, and Average Precision scores (APS), while Figure 8 replicates Figure 3 for this more complex network. Readers will note that once again, certain desirable behaviors of the Cosine model are exhibited by INVITE (*e.g.*, a smooth rise in F_1 over a wide range of middling thresholds, with maximum at a mid-to-high value), along with desirable traits of the Markov model (e.g. significant F_1 at near-zero thresholds).

4.2 Exp. 2: Real-World Excavator MWOs

Unlike the previous synthetic experiments, one does not in general have access to a ground-truth network that validates any chosen similarity measure. Not having labeled data or targets to supervise the learning process is one of the key difficulties in representation learning [16]. To assess the applicability of the INVITE-based similarity measure to real-world scenarios, we apply our model to tags annotated for a mining dataset pertaining to 8 similarly-sized excavators at various sites across Australia [7,55].

The tags were created by a subject-matter expert spending 1 hour of time in the annotation assistance tool `nestor` [56], using a methodology outlined in a previous benchmarking study for that annotation method [9].

That work compared the ability of tags to estimate survival curves and mean time-to-failure, when compared with a custom-designed keyword extraction tool based on classifying the maintenance issues by subsystem. While certain sets of tags were able to predict time-to-failure with high accuracy for certain subsystems, a key problem identified in that work is in knowing beforehand "which tags best represent a given subsystem?"

Some tags are sufficient-but-unnecessary conditions to represent a subsystem —*e.g.*the "hydraulic" tag indicates a Hydraulic System MWO, but so might a "valve", s.t. hydraulic is implied but not present. Consequently, we can treat the problem of assigning tags to a subsystem as a semi-supervised multi-class classification problem: given a few known tag→subsystem assignments, and a similarity value between all pairs of tags, *classify each un-assigned tag as belonging to a subsystem.*

To test the ability of the similarity measures to accomplish this, the top three most common subsystems in the data were used as classes, namely, Hydraulic System, Engine, and Bucket. The tags "hydraulic", "engine", and "bucket" were assigned to those subsystems as known labels, respectively. Tags were filtered to only include ones of high-importance and sufficient information: only work orders containing at least 3 unique tags, and only tags that occurred at least 10 unique times within the those work orders, were included for this analysis ($C = 263$ MWOs, $N = 40$ tags). Then the number of occurrences for every tag can be compared across subsystems, giving each tag a ground-truth multinomial (categorical) probability distribution for occurring within each subsystem, as shown in Figure 4. We determine ground-truth classification labels as subsystems that account for $\geq 60\%$ of each tag's occurrences. Tags more balanced than that are considered "unknown subsystem".

Implementation We proceed in a similar way as before in training the similarity measures for each tag. Note that the tagging annotation process used by [56] assigns tags when they are recognized in raw text through one of many alias'. Therefore, the ordering of tags for these MWOs is strictly based on the order in which English is written—this makes the order any pair's occurrence quite meaningful. As discussed in §3.3, we skip the symmetrization step of Equation 7 until after training is complete.

To perform semi-supervised classification on the recovered relationship graphs, we use a label-spreading algorithm described in [57], which itself was inspired by spreading activation networks in experimental psychology [58, 59]. The result of this algorithm is tags having a score for each class, with the classification being the maximally scored class for that tag. These class assignments can then be compared to the ground-truth labels, which we have done by weighted macro-averaging of the F_1-score (see the top of Figure 5).

Discussion The classification of the INVITE-based similarity measure far outperforms the other measures as a preprocessor for label-spreading, when measured by average F_1-score. However, since these "classifications" are actually thresholded multinomial distributions (with some tags regularly occurring across multiple subsystems), how do we know if an underlying structure has actually been recovered, rather than simply a black-box classifier that happens to perform well at this setting?

To begin answering this question, we might ask whether the relative scores returned by label-spreading are similar to the original multinomial distributions themselves, rather than the overall classification. To find out, we use softmax normalization [8] to transform each tag's scores into a "predicted multinomial", before finally calculating the Kullback-Leibler divergence (KLD) between the true and predicted multinomials for every tag. The total KLD, summed over all tags, is also shown in Figure 5, along with positions of each tag's multinomial as projected onto the 2-simplex for the true and F_1-optimal predicted distributions. Once again, the INVITE performs much better at this task, over a wide range of σ (lower is better).

A reason for the performance disparity can be seen in the simplex projections: recovered topology via INVITE-similarity does a much better job of separating the three classes, while not letting any single tag overcompensate by dominating a subsystem's area. even the "unknown" tags are correctly placed roughly between Bucket and Hydraulic System regions, reflecting the true topology of the system. Interested readers are encouraged to find the best-performing recovered networks visualized in Figure 9, further demonstrating how the properties of each similarity measure behave radically differently.

One other point of note is the number of tags-per-MWO: these results were calculated using MWOs with at least three tags each, but the vast majority of documents in this dataset had fewer than this. The same similarity measures were calculated using more data (having at least 2 tags each), and performance decreased *across the board.* INVITE-based similarity still performed best, with Cosine similarity now closer

[8] For visualization, a temperature parameter was added to softmax, and this was optimized for minimum KLD via Brent's method [60] for each similarity measure independently to provide an equal footing for comparison.

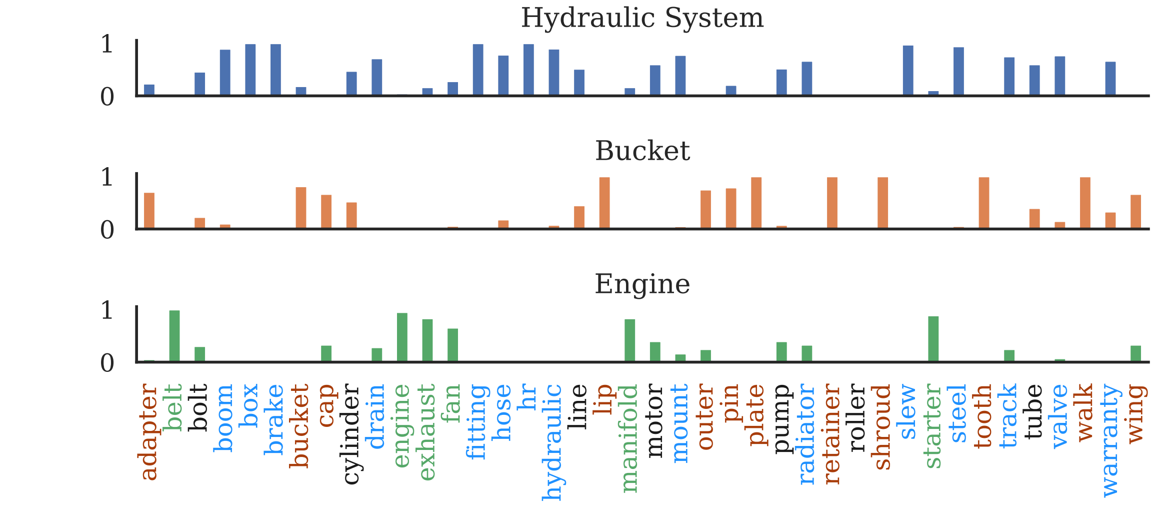

Fig. 4: Ground truth tag multinomial distributions across the top 3 subsystems. "Unknown-subsystem" tags shown in black.

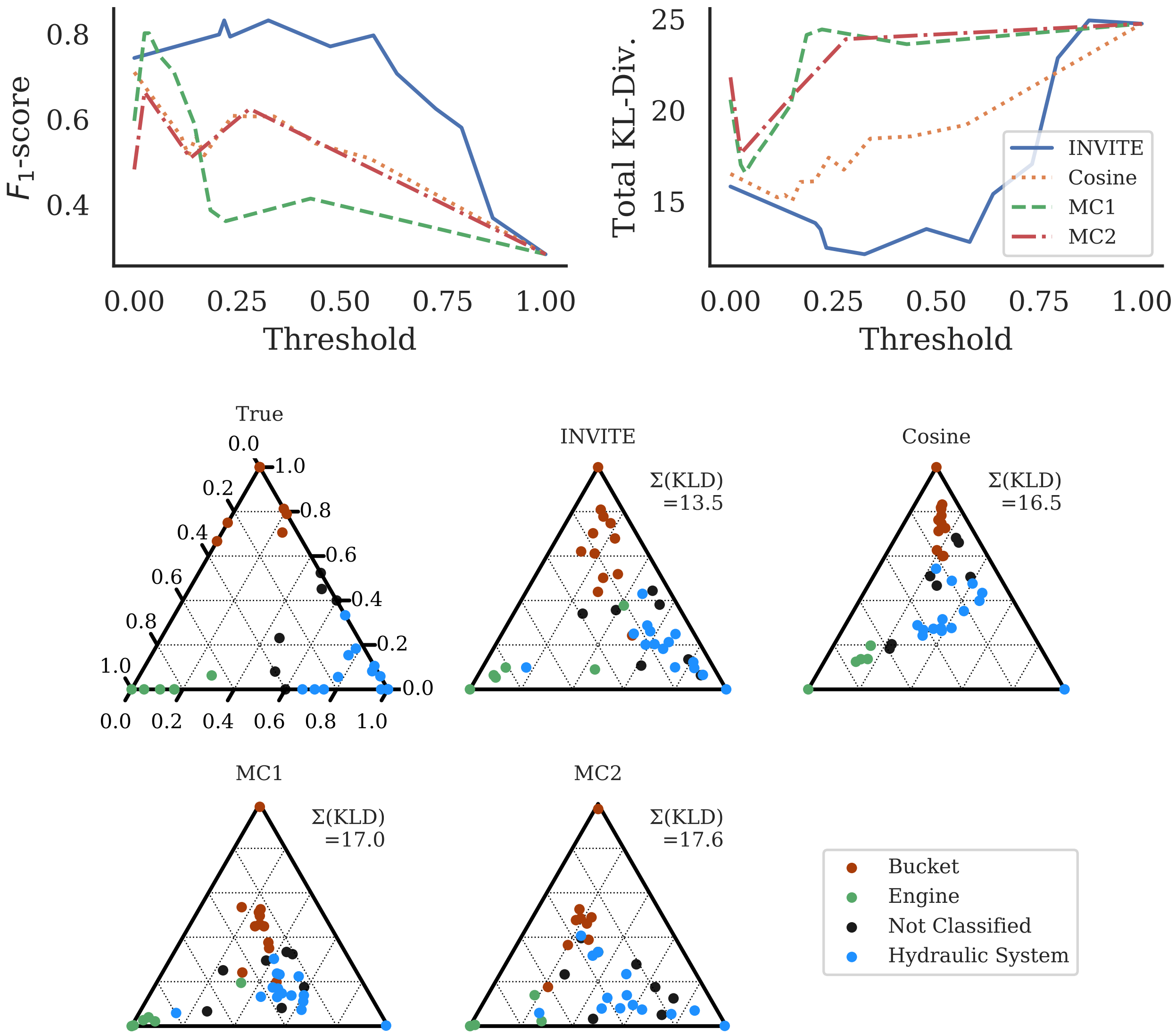

Fig. 5: **Top, left**: (weighted) macro-averaged F_1-scores for semi-supervised classification (label spreading) for using each similarity measure as a pre-processor. **Top, right**: Total KL-divergence for each method between ground-truth multinomal and class score distribution after softmax normalization. **Bottom**: Tag multinomials projected onto a 2-simplex, colored by ground-truth classification.

to it. This decrease indicates a base level of noise in common, catch-all tags that actually reduces the amount we learn about structure from data. In a sense, quality may beat quantity for some types of representation learning. Interested readers can find all additional results in Figure 9.

5 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 locally hierarchical communities. This can lead to better pre-processing for down-stream structure mining and representation learning tasks, as well as for analytics or predictions that better map to expert users' intuitions about how concepts within a system are organized. Both of these have the opportunity to increase trust in data-driven decision support systems, which are increasingly adopted and used without necessarily considering how humans will interact with them [61].

Plenty of work remains to be done to achieve these goals. While the INVITE-based similarity measure performed quite will in our tests, there are still discrepancies between the model it adheres to, and what one might observe in a real folksonomy. For instance, if a "hydraulics" tag is considered too general or abstract for a team that concerns itself largely with hydraulic work, this tag may be skipped as being implied through context. INVITE requires tags to be observed at least once in a record to be reached, but a better method might account for hidden paths or extra, unseen nodes that greatly improve the model's likelihood, much like a form of the "Steiner-tree" problem [62].

Additionally, such a similarity measure could be used for knowledge-structuring-assistance more generally*e.g.*in an active learning context. Such a tool could additionally benefit from a recent explosion in interest for preserving hierarchical and knowledge-graph relationships in vector space, *e.g.*, via Poincaré and "Box-lattice" embeddings [63, 64]. Care must be taken to allow flexible annotation of *different kinds* of relationship strengths,[9], while INVITE assumes a single, generic "similarity". Such a system should allow for multiple (potentially disagreeing) annotators, occasionally suggesting detected relationship types for review to become accepted as ground-truth. We envision a type of "topic model" over the space of knowledge graphs [65], or relationship graphs a combination of independent "graph components" that maximally explain the distribution of edge types in a community [66].

Overall, the model we describe here can enable experts and novices alike to benefit from tacit expertise contained within frequently-unused mountains of tagged technical records, by quickly prototyping quantitative representations of this knowledge as concept-relationship graphs for downstream usage in analysis pipelines. We believe that by explicitly incorporating cognitive theories into our modeling assumptions about how users might represent and then recall their knowledge while tagging, we can accelerate the training and use of unsupervised data-driven expert systems in engineering design.

Acknowledgements

Thanks to Dr. Melinda Hodkiewicz (Univ. Western Australia) for providing and annotating the excavator data, for providing feedback on the 'Nestor' tagging process, and for many enjoyable and enlightening discussions on this topic.

6 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] 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.

[2] 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.

[3] 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.

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

[5] 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.

[6] 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.

[7] 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.

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

[9] Sexton, T., Hodkiewicz, M., Brundage, M. P., and Smoker, T., 2018. "Benchmarking for keyword extrac-

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

tion methodologies in maintenance work orders". In PHM Society Conference, Vol. 10.

[10] 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.

[11] Miller, G. A., 1998. *WordNet: An electronic lexical database*. MIT press.

[12] 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.

[13] 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.

[14] Pantförder, D., Schaupp, J., and Vogel-Heuser, B., 2017. "Making implicit knowledge explicit–acquisition of plant staff's mental models as a basis for developing a decision support system". In International Conference on Human-Computer Interaction, Springer, pp. 358–365.

[15] Hadzic, F., Tan, H., and Dillon, T. S., 2010. *Mining of data with complex structures*, Vol. 333. Springer.

[16] Bengio, Y., Courville, A., and Vincent, P., 2013. "Representation learning: A review and new perspectives". *IEEE transactions on pattern analysis and machine intelligence,* ***35***(8), pp. 1798–1828.

[17] 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.

[18] Macgregor, G., and McCulloch, E., 2006. "Collaborative tagging as a knowledge organisation and resource discovery tool". *Library review,* ***55***(5), pp. 291–300.

[19] Huang, Y.-M., Huang, Y.-M., Liu, C.-H., and Tsai, C.-C., 2013. "Applying social tagging to manage cognitive load in a web 2.0 self-learning environment". *Interactive Learning Environments,* ***21***(3), pp. 273–289.

[20] 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.

[21] 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.

[22] 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.

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

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

[25] 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.

[26] Chen, W., Fuge, M., and Chazan, J., 2017. "Design manifolds capture the intrinsic complexity and dimension of design spaces". *Journal of Mechanical Design,* ***139***(5), p. 051102.

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

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

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

[30] Nickel, M., and Kiela, D., 2018. "Learning continuous hierarchies in the lorentz model of hyperbolic geometry". *arXiv preprint arXiv:1806.03417*.

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

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

[33] 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.

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

[35] 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.

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

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

[38] Moussely-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.

[39] 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.

[40] 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.

[41] 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.

[42] Bergamaschi, S., Guerra, F., Rota, S., and Velegrakis, Y., 2011. "A hidden markov model approach to keyword-based search over relational databases". In International Conference on Conceptual Modeling, Springer, pp. 411–420.

[43] Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., and Khudanpur, S., 2010. "Recurrent neural network based language model". In Eleventh annual conference of the international speech communication association.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

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

[49] 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.

[50] 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.

[51] 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.

[52] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A., 2017. "Automatic differentiation in pytorch". In NIPS-W.

[53] Schreiber, J., 2018. "Pomegranate: fast and flexible probabilistic modeling in python". *Journal of Machine Learning Research,* ***18***(164), pp. 1–6.

[54] Watts, D. J., and Strogatz, S. H., 1998. "Collective dynamics of 'small-world' networks". *nature,* ***393***(6684), p. 440.

[55] 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.

[56] 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.

[57] Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B., 2004. "Learning with local and global consistency". In Advances in neural information processing systems, pp. 321–328.

[58] Anderson, J. R., 2013. *The architecture of cognition*. Psychology Press.

[59] Shrager, J., Hogg, T., and Huberman, B. A., 1987. "Observation of phase transitions in spreading activation networks". *Science,* ***236***(4805), pp. 1092–1094.

[60] Brent, R. P., 1971. "An algorithm with guaranteed convergence for finding a zero of a function". *The Computer Journal,* ***14***(4), pp. 422–425.

[61] Brundage, M. P., Sexton, T., Hodkiewicz, M., Morris, K., Arinez, J., Ameri, F., Ni, J., and Xiao, G., 2019. "Where do we start? guidance for technology implementation in maintenance management for manufacturing". *Journal of Manufacturing Science and Engineering*, pp. 1–24.

[62] Ivanov, A. O., and Tuzhilin, A. A., 1994. *Minimal NetworksThe Steiner Problem and Its Generalizations*. CRC press.

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

[64] Vilnis, L., Li, X., Murty, S., and McCallum, A., 2018. "Probabilistic embedding of knowledge graphs with box lattice measures". *arXiv preprint arXiv:1805.06627*.

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

[66] 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.

A Supplementary Material: Airplane (Walsh et al.)

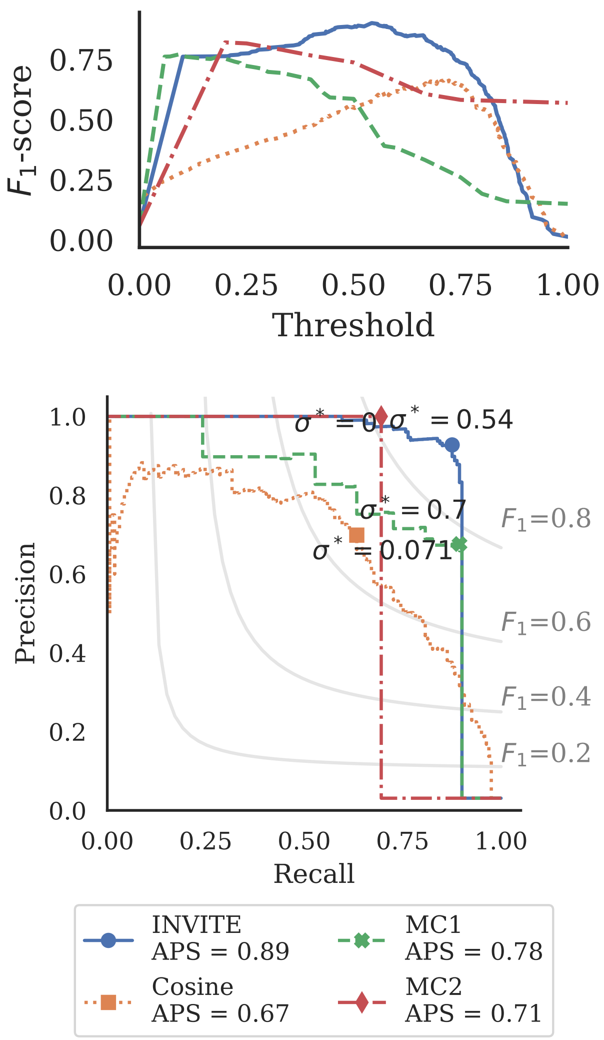

Fig. 6: F_1-scores and precision-recall curves for the airplane network, using $C = 100$ random walks. Precision/Recall is reported alongside Average Precision Score for the whole threshold range, and the threshold value for which F_1-score was optimal

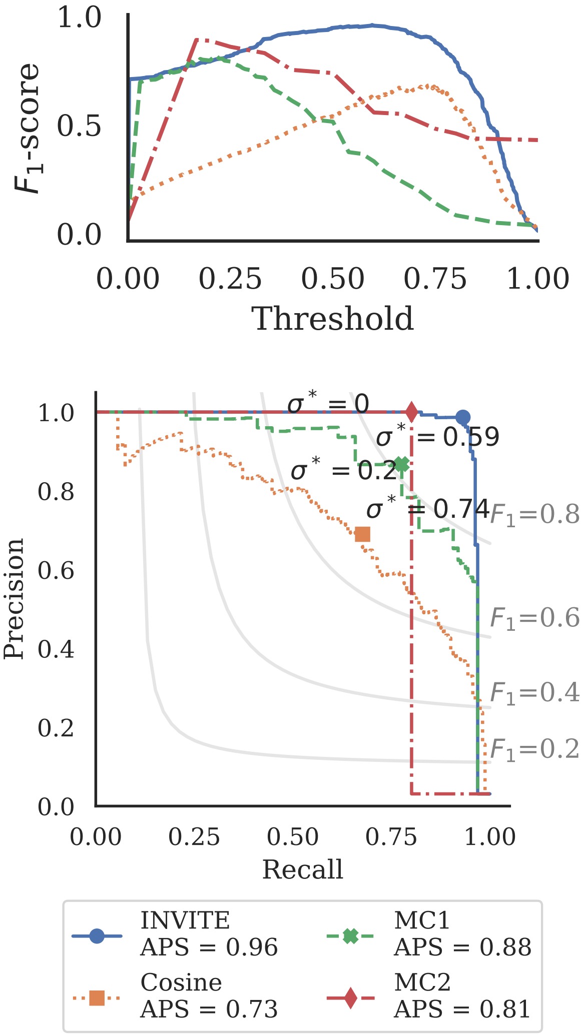

Fig. 7: Repeating Figure 6 using $C = 200$ random walks.

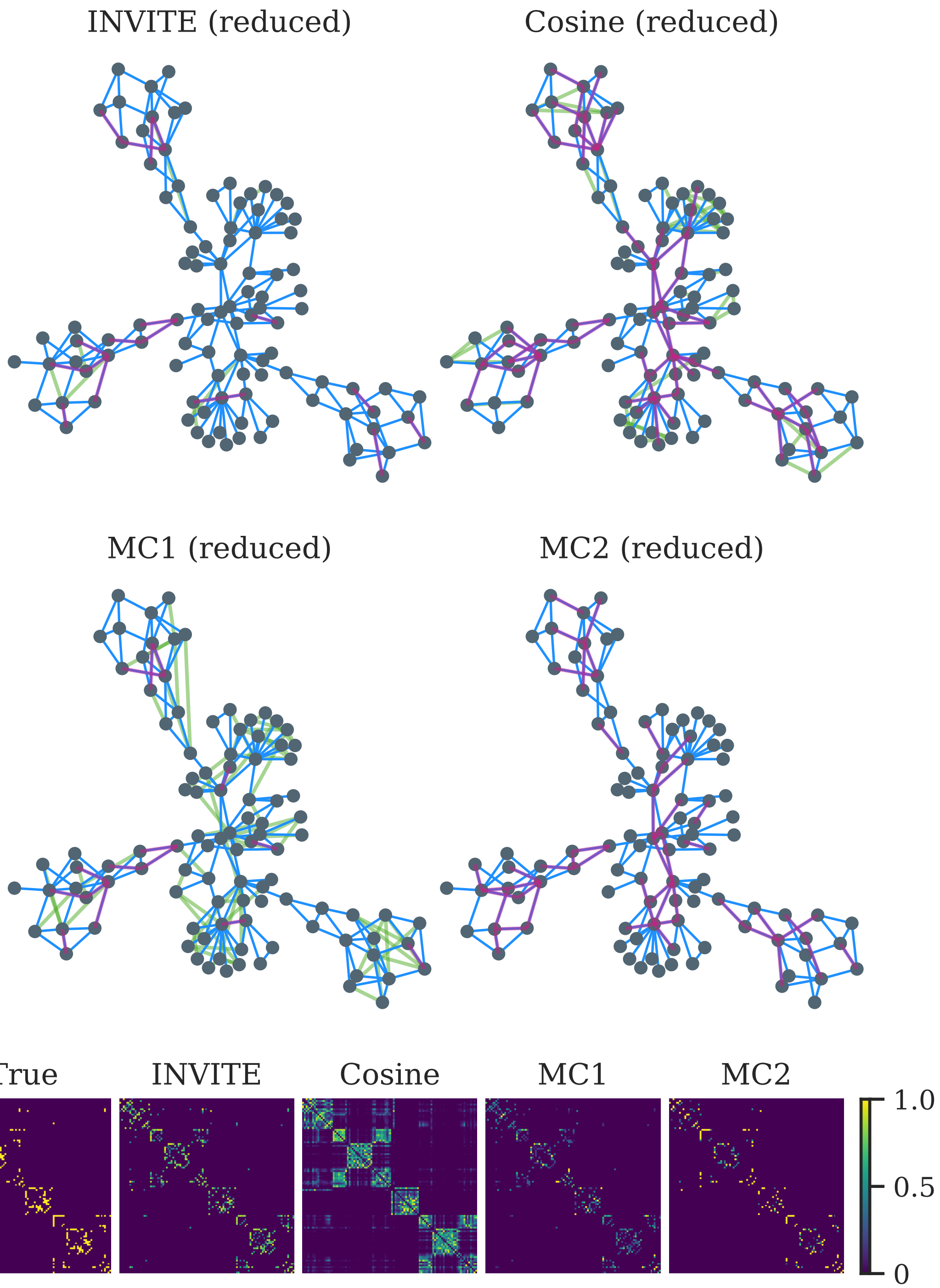

Fig. 8: Repeating the Figure 3 visualizations for the reduced Airplane network, and $C = 100$ random walks.

B Supplementary Material: Excavators

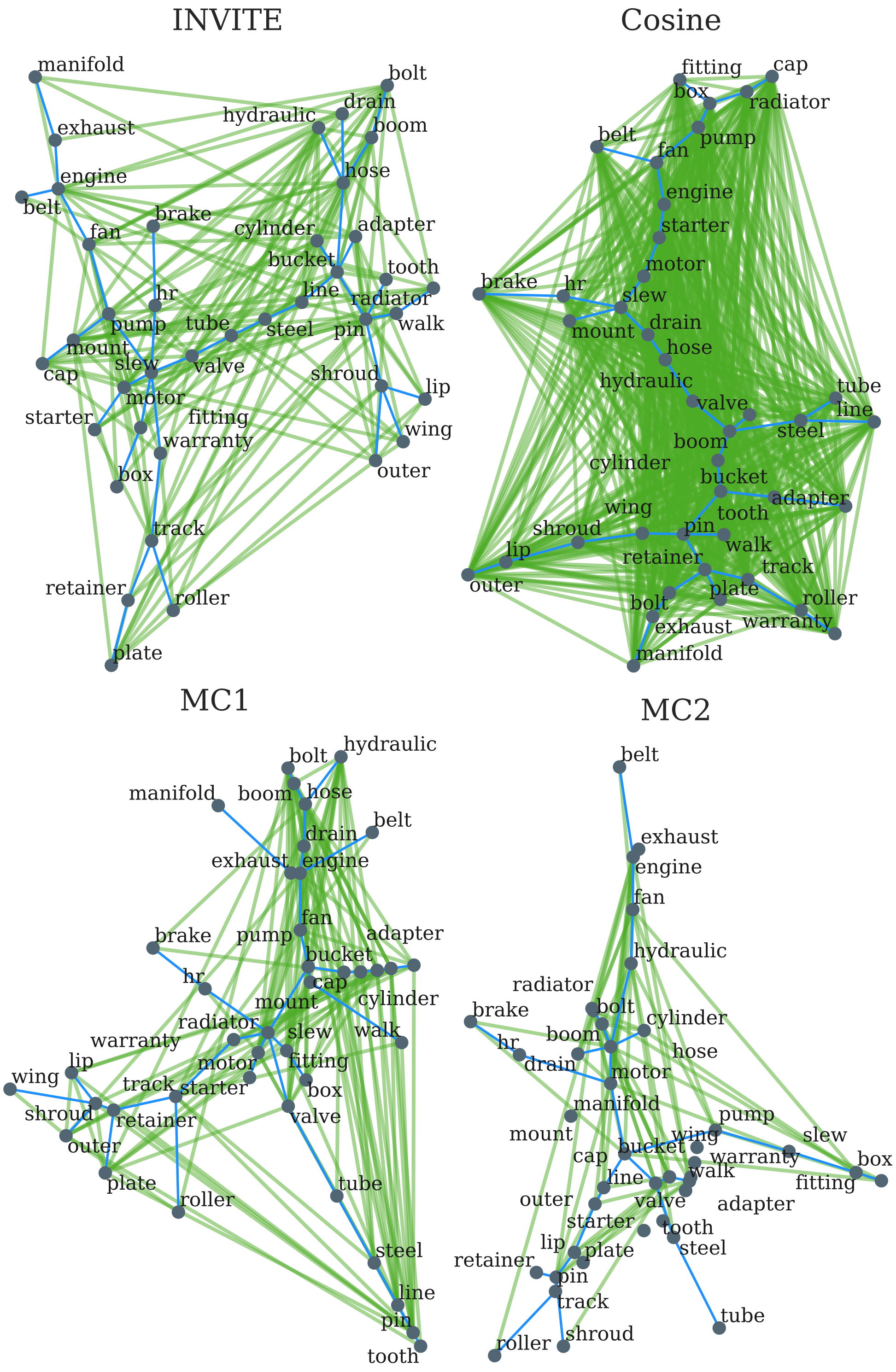

Fig. 9: F_1-optimal tag structures recovered from excavator MWOs having at least 3 tags ($C = 263$). Maximum spanning tree appearing in blue, with other edges appearing in green. Note the two markov-chain models isolated certain tags to achieve better performance—an sign that the learned topology was less relevant.

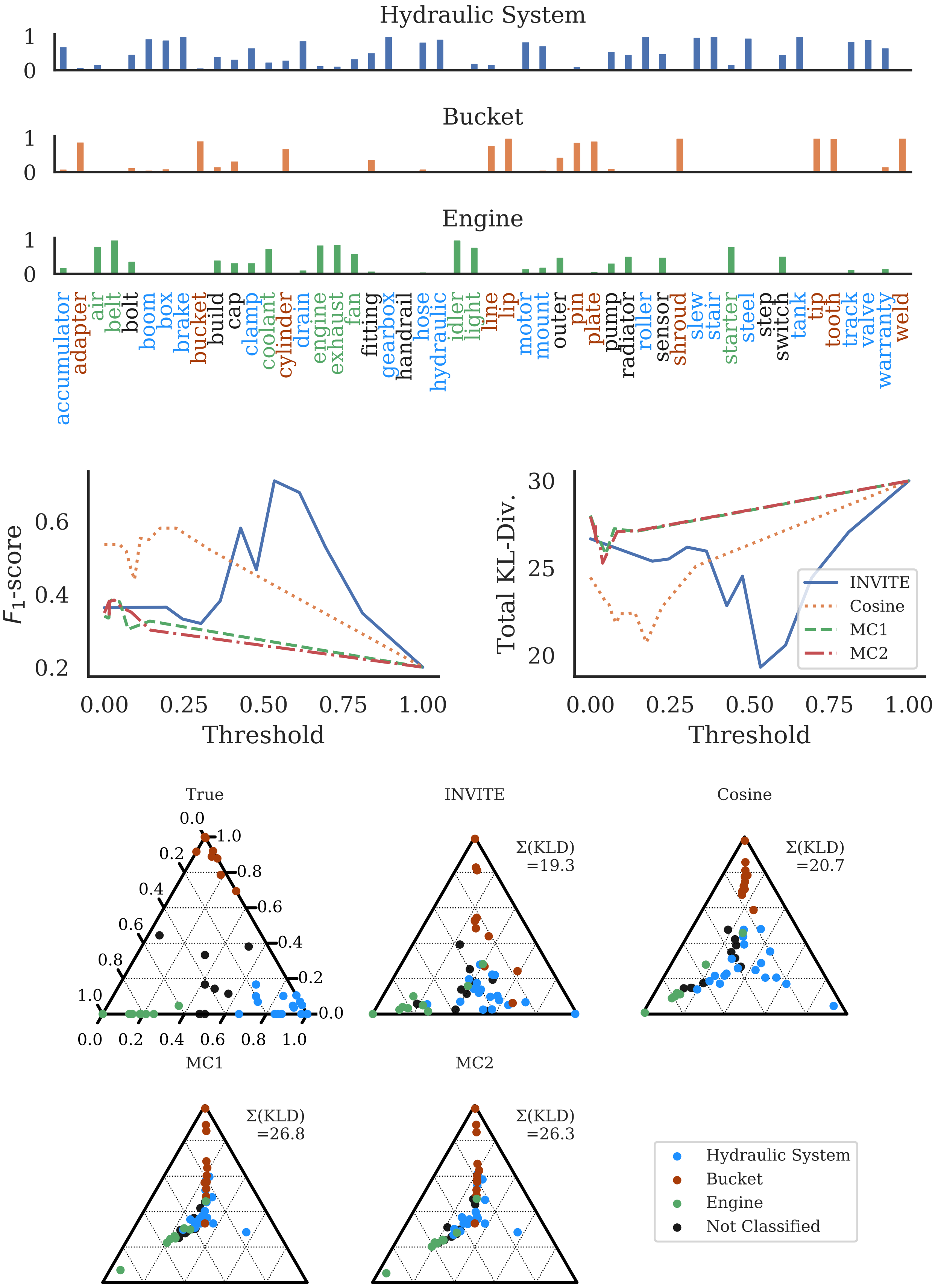

Fig. 10: Results for the excavator data where any MWO having at least 2 tags was allowed. The increase in MWO from 263 to 1712 is indicative of the sparsity in this data, and how varied the tags can be. Additionally, the decrease in performance indicates how important increased expert elicitation at the individual MWO-level can be toward extracting useful knowledge with representation learning.

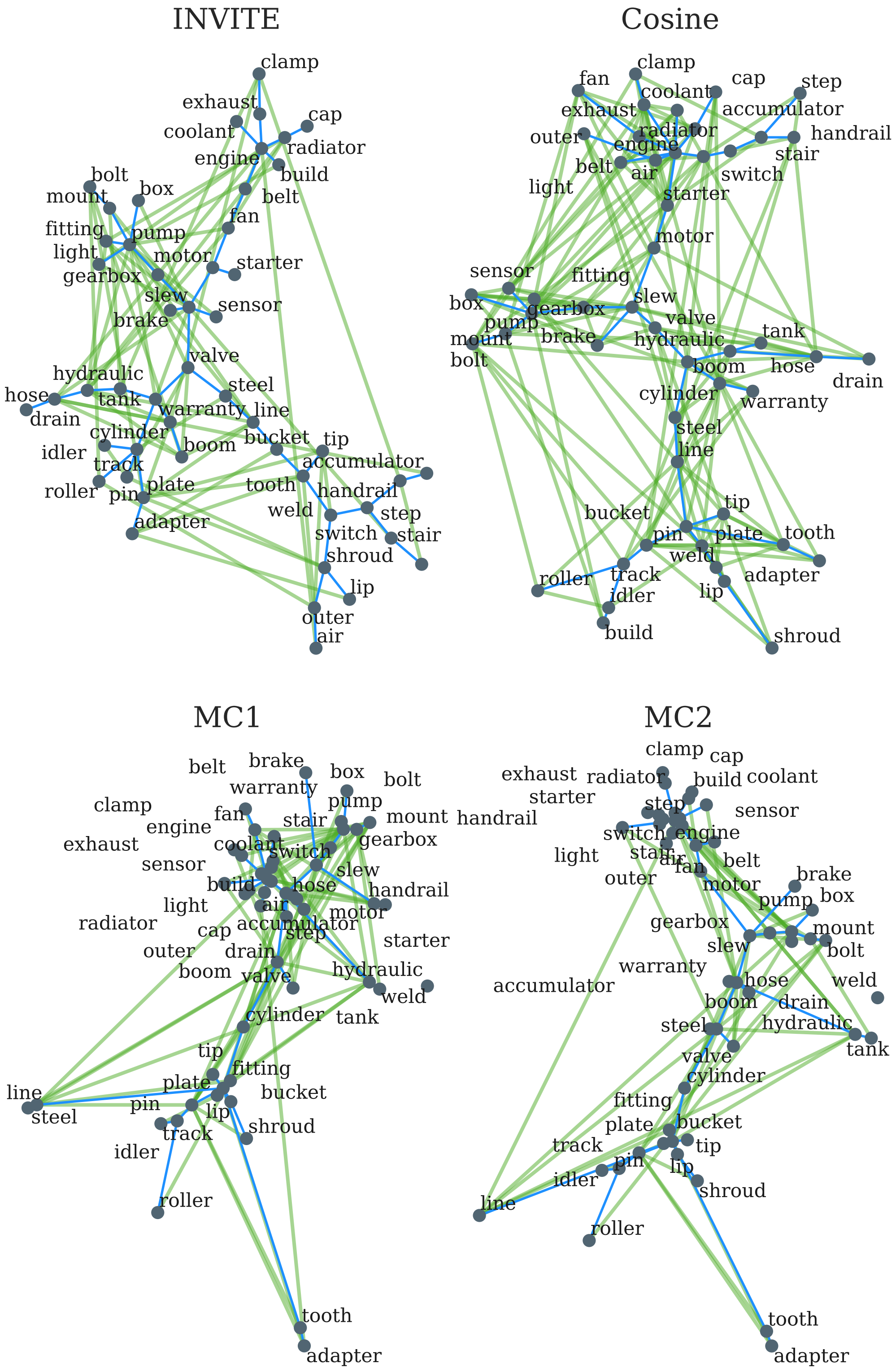

Fig. 11: F_1-optimal tag structures recovered from excavator MWOs having at least 2 tags ($C = 1712$). Networks are ploted with highlighted maximum spanning tree appearing in blue, with other edges appearing in green. While performance was lower across the board (even with far more training data), clear communities around the three subsystems can be clearly seen for both INVITE and Cosine models, and the sparsity for Cosine is much improved, with more data to filter spurious connections against.