Organizing Tagged Knowledge

Similarity Measures and Semantic Fluency in Structure Mining

Thurston Sexton (NIST) Mark Fuge (UMD)

8 August 2019

National Institute of
Standards and Technology
U.S. Department of Commerce

Background

How to "Get Smart"?

Maintenance is expensive ($50 billion for USA in 2016 [1]) and expertise-driven, *but*... **Smart manufacturing** technologies can reduce costs!

SME's still not employing these technologies

- High Cost to implement; Risk is high with incorrect implementation
- Lack of Support/Expertise in manufacturing
- Leads to a lack of high quality (or understood) sensor data

Have little/no data

Difficult to assess impacts of new technologies

How to “Get Smart”?

Except...*that’s not entirely true.*

- Untapped source of data...natural language documents
- **Maintenance Work Orders**
- These come with severe issues! [2]

jargon/misspellings

```
Hyd leak at saw atachment
```

abbreviations

```
HP coolant pressure at 75 psi
```

lack of context

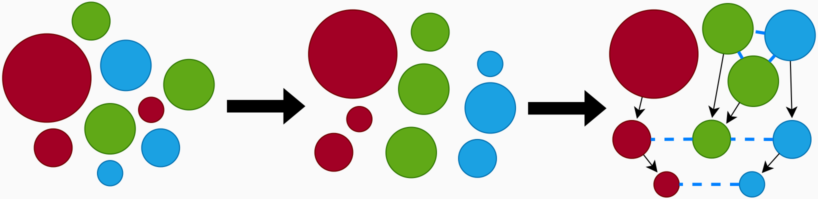
```
Replaced – Operator could have done this!
```

Want

What kinds of useable system knowledge could be gained from such data?

Things that could assist in diagnostics, prognostics, schedule, etc.?

1. What **components**/concepts are **relevant** to our system
2. How these components are **related**



Tagging Example

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

Tagging Example

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

Missing the second half: **tag relationships** (this work!)

→ Structure Mining (e.g. Representation Learning)

Modeling Tag Relationships

Compare the Archetypes

At a high level:

1. Context (global) - *e.g. Cosine Similarity*
   Summary: "Tags found in similar contexts are similar."
    - Excellent Recall;
    - Used in structure mining/representation learning literature
    - Hard to tune, mistakes correlation with relation

Compare the Archetypes

At a high level:

1. Context (global) - *e.g. Cosine Similarity*
   Summary: “Tags found in similar contexts are similar.”
   - Excellent Recall;
   - Used in structure mining/representation learning literature
   - Hard to tune, mistakes correlation with relation
2. Sequence (local) - *e.g. Markov Chain*
   Summary: Tags come “out” of the user in order*!*
   - Better Precision;
   - Used to model generative human processes
   - possibility to miss large-scale patterns

Which one? How to combine?

Cosine Similarity

Cosine of the angle between example vectors:

MWO 1: "Hyd leak at saw attachment"

[`hydraulic`, `saw`] $

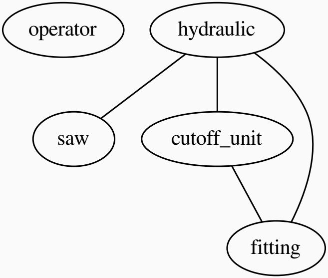
MWO 2: "Hydraulic Leak at cutoff unit; Missing fitting replaced"

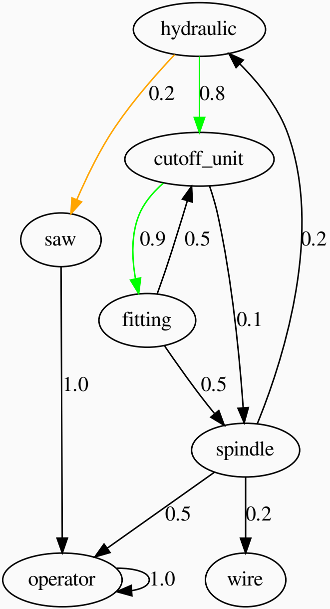
[`hydraulic`, `cutoff_unit`, `fitting`]

MWO 3: "Replaced – Operator could have done this!"

[`operator`]

$$u_{\text{saw}} = (1, 0, 0, \cdots)$$
$$u_{\text{hydraulic}} = (1, 1, 0, \cdots)$$
$$u_{\text{operator}} = (0, 0, 1, \cdots)$$





Each “tag” is a **state** with transition probabilities to other tags. Tags on a resource are observed random walks through tag-states.

Random Walk 1

hydraulic $\longrightarrow$ cutoff_unit $\longrightarrow$ fitting

Random Walk 2

hydraulic $\longrightarrow$ saw

Random Walk 3

operator

Hold up a minute...

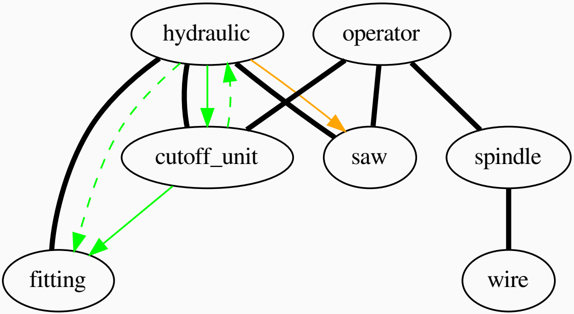
Let's step back. What might actually be going on? What could a reasonable set of tags be coming from?

Random Walk 1

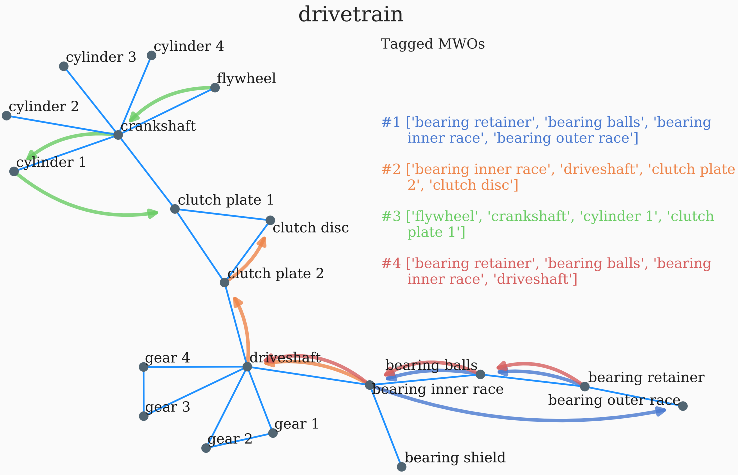
hydraulic ⟶ cutoff_unit ⟶ fitting

Random Walk 2

hydraulic ⟶ saw



More Realistic Example



Component network model from Walsh et al. [3]

Semantic Fluency Data

Audience Participation

VOLUNTEER NEEDED

VOLUNTEER NEEDED

What patterns were there? Why did *those* transitions happen?

Another list :

Semantic (or verbal) Fluency Tasks ask participants to recall objects in a category and write them down, as quickly as possible.

- dog

Another list :

Semantic (or verbal) Fluency Tasks ask participants to recall objects in a category and write them down, as quickly as possible.

- dog
- cat

Another list :

Semantic (or verbal) Fluency Tasks ask participants to recall objects in a category and write them down, as quickly as possible.

- dog
- cat
- lion

Another list :

Semantic (or verbal) Fluency Tasks ask participants to recall objects in a category and write them down, as quickly as possible.

- dog
- cat
- lion
- tiger

Another list :

Semantic (or verbal) Fluency Tasks ask participants to recall objects in a category and write them down, as quickly as possible.

- dog
- cat
- lion
- tiger
- elephant

Another list :

Semantic (or verbal) Fluency Tasks ask participants to recall objects in a category and write them down, as quickly as possible.

- dog
- cat
- lion
- tiger
- elephant
- wolf

Another list :

Semantic (or verbal) Fluency Tasks ask participants to recall objects in a category and write them down, as quickly as possible.

- dog
- cat
- lion
- tiger
- elephant
- wolf
- ...

Another list :

Semantic (or verbal) Fluency Tasks ask participants to recall objects in a category and write them down, as quickly as possible.

- dog
- cat
- lion
- tiger
- elephant
- wolf
- ...

Another list :

Semantic (or verbal) Fluency Tasks ask participants to recall objects in a category and write them down, as quickly as possible.

- dog
- cat
- lion
- tiger
- elephant
- wolf
- ...

Household

Felines

“African”

Canines

What have we learned?

Key observations:

1. Recalled items **jump** between overlapping **contexts** (global)
2. **No** item **repetition** through the list (local)
3. **Past** items influence **sequential** context-changes

What have we learned?

Key observations:

1. Recalled items **jump** between overlapping **contexts** (global)
2. **No** item **repetition** through the list (local)
3. **Past** items influence **sequential** context-changes

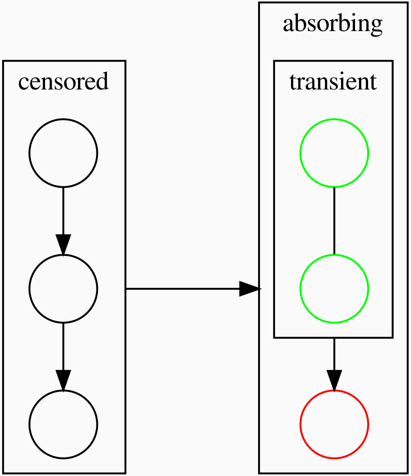
This describes a combination of *both* modelling archetypes! How to describe computationally? **Initial-Visit Emitting Random Walks**

INVITE - Jun et al. (2015); Zemla & Austerweil (2018) [4,5]

- All visited nodes are "allowed" and *hidden*
- **infinite** number of paths can generate a given observation!

key insight

Split each "random walk" of length K into $K-1$ **absorbing** random sub-walks.



Experiments

What are we missing if a process like INVITE *is* generating our data?

- If we assume Bag-of-Words applies
- If we assume Markov property applies

What are we missing if a process like INVITE *is* generating our data?

- If we assume Bag-of-Words applies
- If we assume Markov property applies

Let's Find Out - We are looking to quantify model effects on *edge probability*.

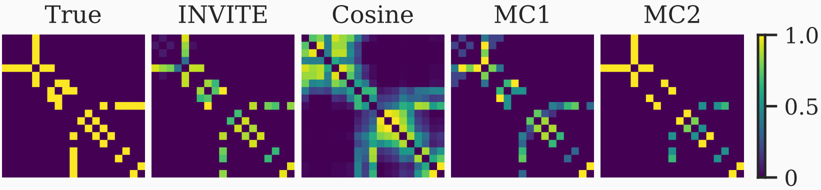
- **recall** is our ability to select all relevant edges, and
- **precision** is the relevance of our selected edges.

The edge probability threshold (σ) we choose has a huge effect on these.

Experiment 1 - Example

To illustrate, return to the example of Walsh et al. [3]

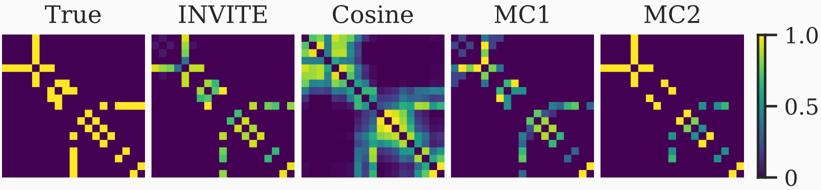
- adjacency matrix representation (recovered edge probabilities):
- $C = 20$ random walks of length $l = 4$



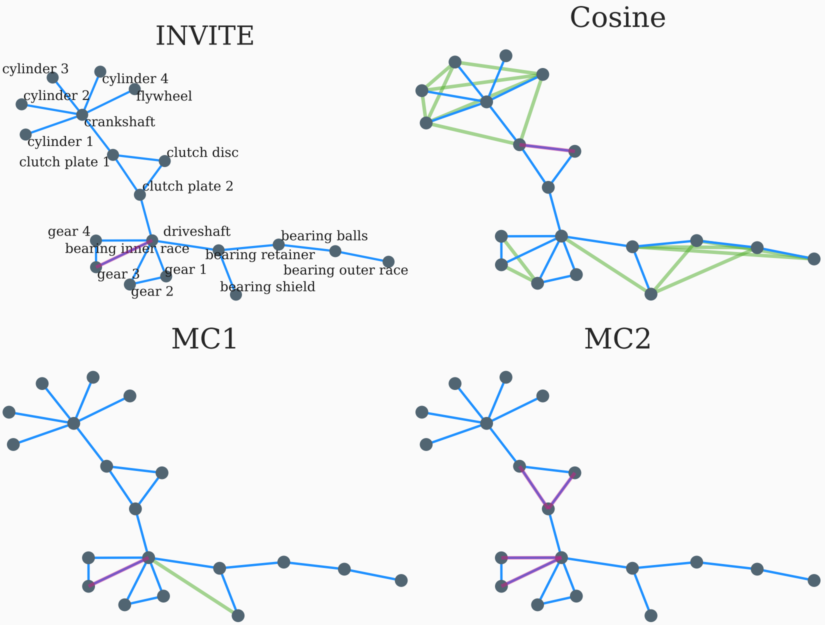
Experiment 1 - Example

To illustrate, return to the example of Walsh et al. [3]

- adjacency matrix representation (recovered edge probabilities):
- $C = 20$ random walks of length $l = 4$

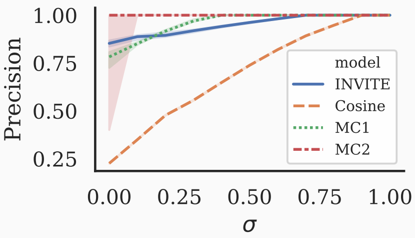


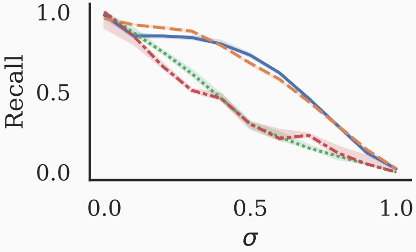
Let's threshold for optimal F_1 score...


INVITE
cylinder 3
cylinder 4
cylinder 2
flywheel
crankshaft
cylinder 1
clutch plate 1
clutch disc
clutch plate 2
gear 4
driveshaft
bearing balls
bearing inner race
bearing retainer
gear 3
gear 1
bearing outer race
gear 2
bearing shield
Cosine
MC1
MC2

Experiment 1 - Summary Curves

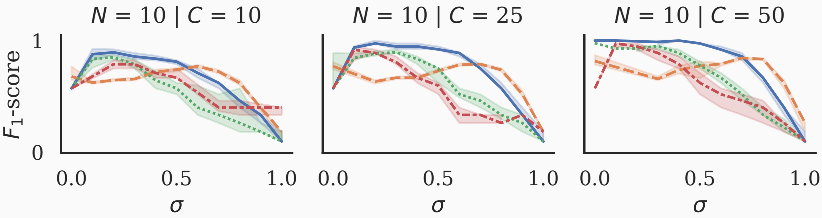
Across 90 graphs, $N, C \in \{10, 25, 50\}$:

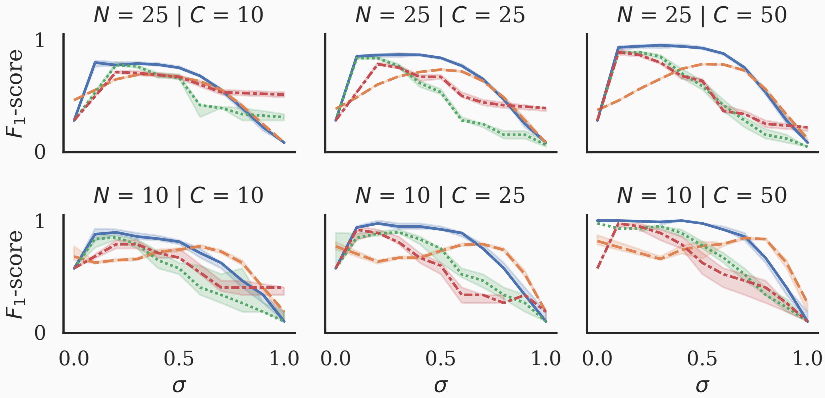




- **Precision**: INVITE ~ Sequential Probabilities
- **Recall**: INVITE ~ Contextual Similarity

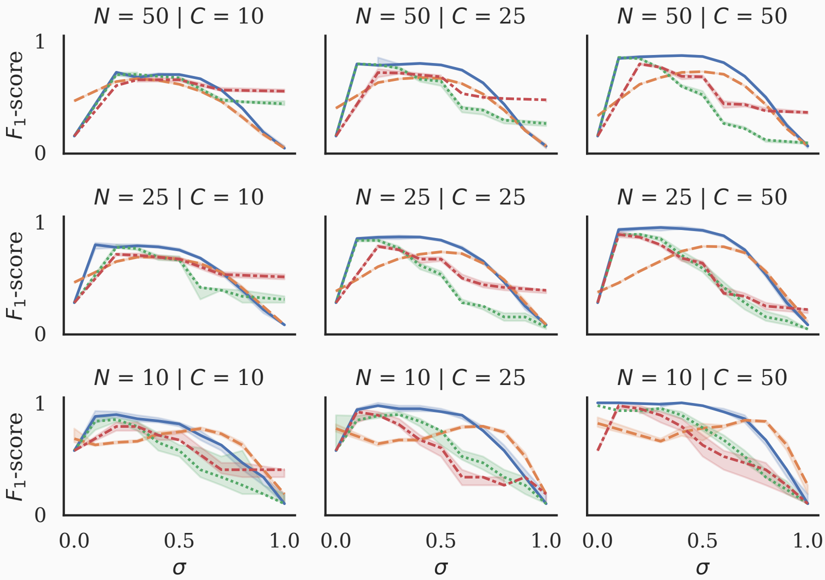
Experiment 1 - Performance Curves





N = 25 | C = 10
N = 25 | C = 25
N = 25 | C = 50
N = 10 | C = 10
N = 10 | C = 25
N = 10 | C = 50
F_1-score
1
0
0.0
0.5
1.0
σ

Experiment 1 - Performance Curves



Exp. 2 - Excavator Case Study

Excavator - Dataset

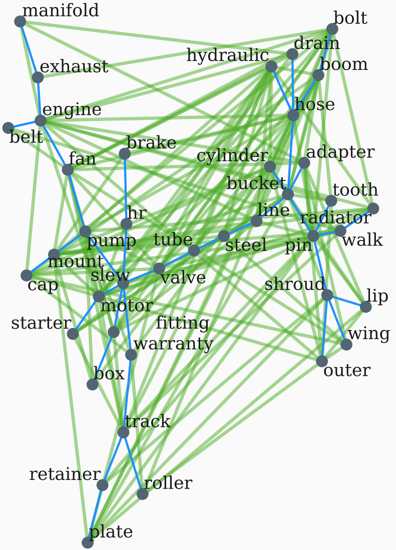
1. **Mining Dataset**: Excavator Maintenance Work orders [6,7]
    - 8 Excavators, 8264 MWOs
    - Bespoke keyword-recognition tool; recognize failure “major subsystem”
    - Labor intensive… months (and a dissertation!)
2. Tags - Nestor Toolkit [8]
    - Compare Survival Analysis - tags vs. keyword-recognition [9]
    - Subsystems approximated by expert-determined “tag-sets”

PROBLEM

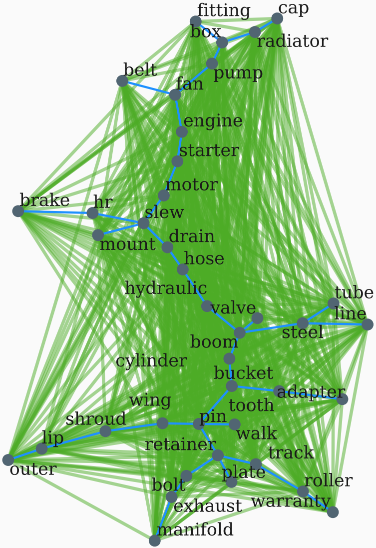
How to estimate subsystem by tags? **Which tags**?

Excavator - Recovered Structure

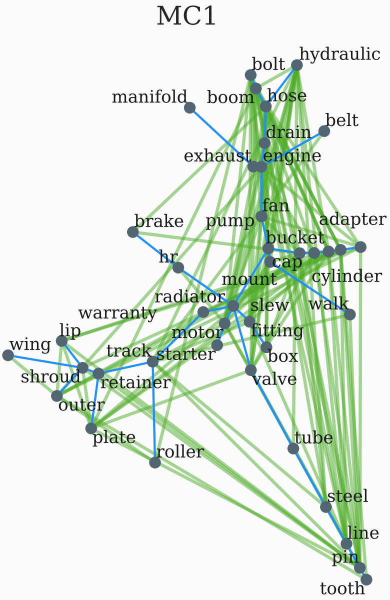
INVITE

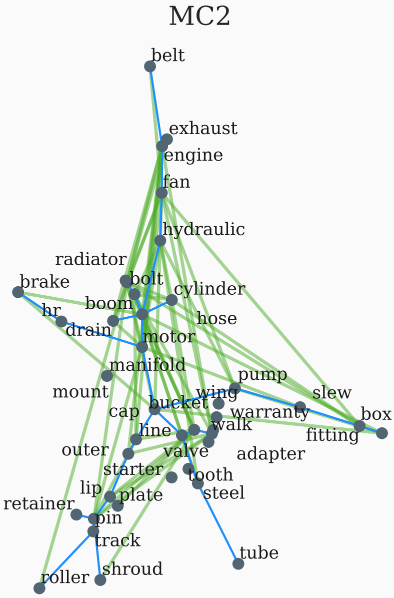


Cosine



Excavator - Recovered Structure





Excavator Data

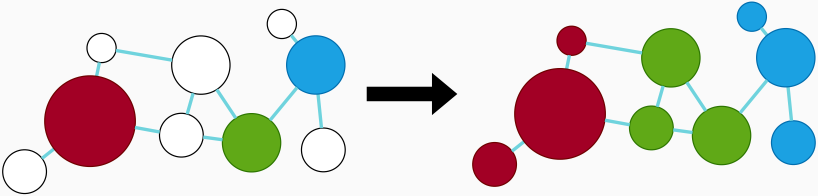
Some tags are trivially known (use these!)

- `bucket` $\longrightarrow$ "Bucket Subsystem"
- `hydraulic` $\longrightarrow$ "Hydraulic System"
- `engine` $\longrightarrow$ "Engine Subsystem"

Thanks to the keyword-tagger, we can compare ground-truth tag-subsystem allocations:

- `hose` : *probably* **hydraulic**
- `teeth`: *always* **bucket**
- `bolt`: all of them?

More generally, this is a problem of estimating labels from the data *topology*; **Label Spreading**



We apply the label spreading algorithm of Zhou et al (2004) [10]

Excavator - Results

Comparing tag label distributions: **ground-truth** (from keyword extractor) v.s. **predicted** classification (from label spreading)

Excavator - Results

Comparing tag label distributions: **ground-truth** (from keyword extractor) v.s. **predicted** classification (from label spreading)

	F_1^*	ΣKL	μ_{KL}
INVITE	**0.83**	**13.5**	**0.35**
Cosine	0.71	16.5	0.42
MC1	0.80	17.0	0.44
MC2	0.66	17.6	0.45

INVITE consistently performs best, across a wide range of σ.

Conclusions

Key Contributions - Modeling Similarity

1. If censored random walks *are* taking place
    - Context-based models could work; **hard to tune**
    - Sequential models can **miss latent** relationships
    - Accounting for censoring improves **precision and recall**

Key Contributions - Modeling Similarity

1. If censored random walks *are* taking place
   - Context-based models could work; **hard to tune**
   - Sequential models can **miss latent** relationships
   - Accounting for censoring improves **precision and recall**
2. In analytics tasks down-stream (i.e. Semi-Supervised Classifier)
   - Use similarity model assumptions as pre-processor
   - Incorporating INVITE can better map to users' organizational intuitions.

INVITE is only the beginning...

- Relevant tags may be skipped entirely
    - Too general
    - Too specific
    - Hidden nodes? Node hierarchies?

Future Work

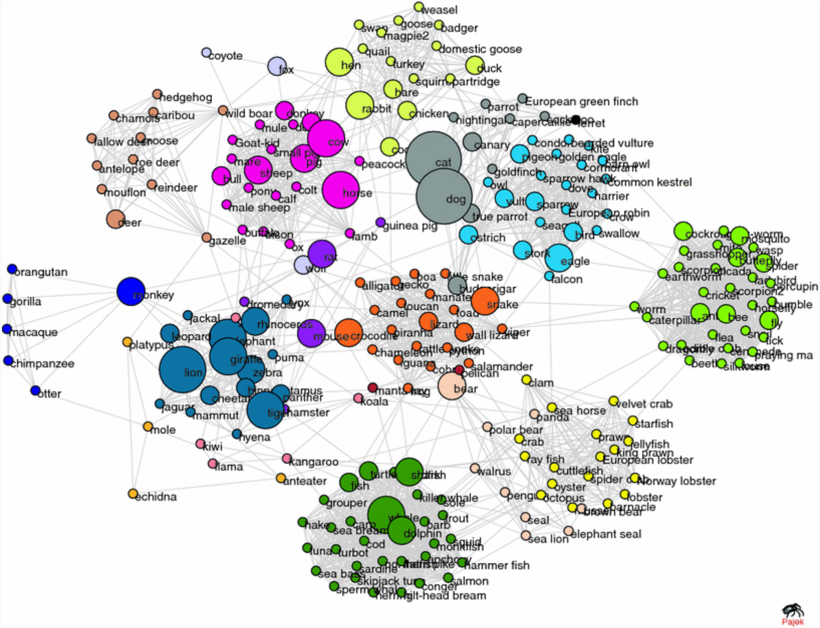
INVITE is only the beginning...

- Relevant tags may be skipped entirely
  - Too general
  - Too specific
  - Hidden nodes? Node hierarchies?
- Active Learning of representation learning
  - Real-time feedback to *build trust*
  - Exploit embeddings, probabilistic models, etc.
  - Mixture models for different *types* of relationships

Thank You! Questions?

Backup

Animals Network - Goni et al. (2011) [11]



INVITE - Absorbing Random (sub-) Walks

Partition at k^{th} *observed* transition ($t_k \to t_{k+1}$):

- q transient states
- transition matrix $\mathbf{Q}^{(k)}_{q\times q}$
- r absorbing states with $q \to r$ transitions as $\mathbf{R}^{(k)}_{q\times r}$

Markov transition matrix $\mathbf{M}^{(k)}_{n\times n}$ has the form:

$$\mathbf{M}^{(k)} = \begin{pmatrix} \mathbf{Q}^{(k)} & \mathbf{R}^{(k)} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \quad (1)$$

Partition at k^{th} *observed* transition $(t_k \rightarrow t_{k+1})$:

- q transient states
- transition matrix $\mathbf{Q}^{(k)}_{q\times q}$
- r absorbing states with $q \rightarrow r$ transitions as $\mathbf{R}^{(k)}_{q\times r}$

Markov transition matrix $\mathbf{M}^{(k)}_{n\times n}$ has the form:

$$\mathbf{M}^{(k)} = \begin{pmatrix} \mathbf{Q}^{(k)} & \mathbf{R}^{(k)} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \tag{1}$$

Probability P of chain starting at t_k being *absorbed* into state $k+1$, letting $\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1}$, is [12]:

$$P\left(t_{k+1} | t_{1:k}, \mathbf{M}\right) = \mathbf{N}^{(k)} R^{(k)}\Big|_{q,1} \tag{2}$$

Summary

The probability of being absorbed at $k+1$ conditioned on jumps $1:k$ is the probability of observing our $k+1$ INVITE tag.

Likelihood of $\mathbf{M}$ given observed *censored* chain $\vec{t}$ is:

$$\mathcal{L}\left(\vec{t}\,|\,\theta\,;\mathbf{M}\right) = \theta(t_1)\prod_{k=1}^{T-1} P\left(t_{k+1}\,|\,t_{1:k}\,;\,\mathbf{M}\right) \tag{3}$$

This implies that a "folksonomy" of tag lists $\mathbf{C} = \left\{\vec{t}_1, \vec{t}_2, \cdots, \vec{t}_c\right\}$ can recover an $\mathbf{M}$ through optimization:

$$\mathbf{M}^* \leftarrow \arg\min_{\mathbf{M}} \quad \sum_{i=1}^{C}\sum_{k=1}^{T_i-1} -\log\mathcal{L}\left(t_{k+1}^{(i)}\middle|t_{1:k}^{(i)}, \mathbf{M}\right) \tag{4}$$

How to optimize?

- Still nearly intractable for large numbers of tags, given search-space
- Analytic gradient given in Jun & Zemla et al. [4] has restrictions
- Binarizing edge states [5] removes edge weights entirely.

Model Inference - Implementation Details

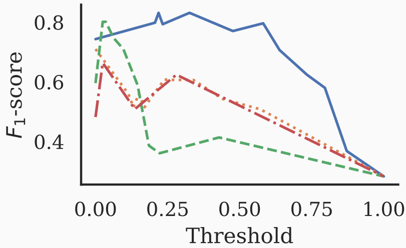
How to optimize?

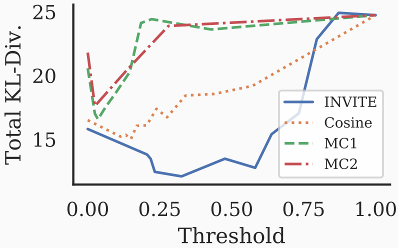
- Still nearly intractable for large numbers of tags, given search-space
- Analytic gradient given in Jun & Zemla et al. [4] has restrictions
- Binarizing edge states [5] removes edge weights entirely.

We leverage **automatic differentiation**

- Tensor library PyTorch to minimize the loss via ADAM [13]
- Flexible w.r.t symmetry, bound transformations, etc during training

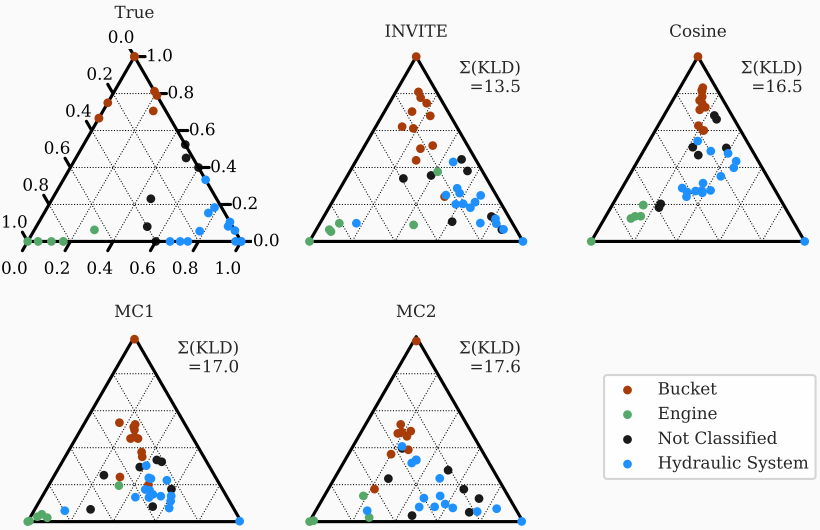
Excavator - Results ($l \geq 3$)





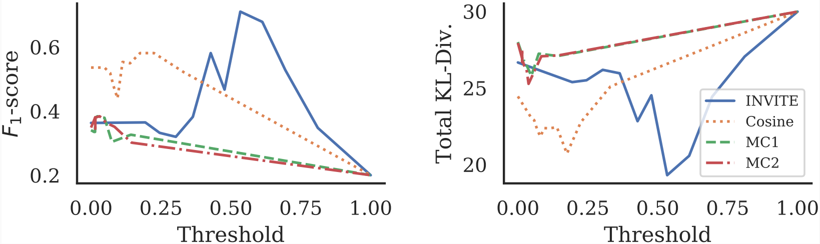
Excavator - Distribution Comparison ($l \geq 3$)

at optimal F_1-score:



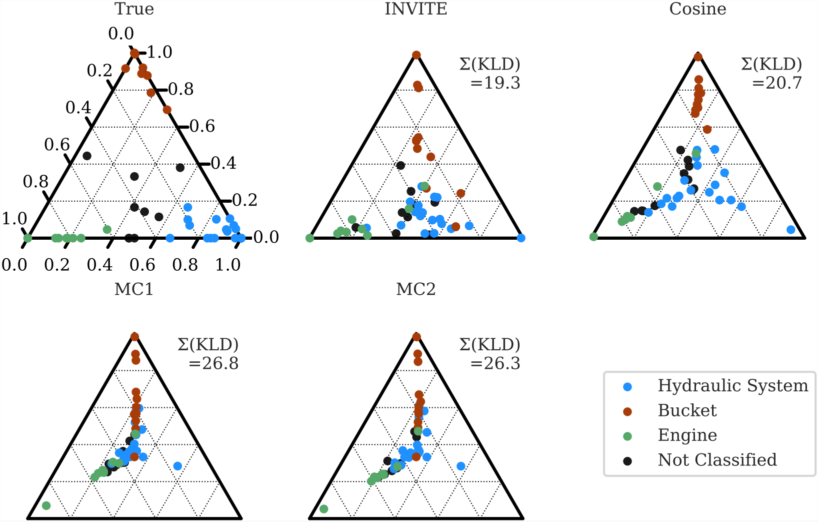
Excavator - Results ($l \geq 2$)

- MWOs with at least 2 tags, each occurring at least 10x
  - $C = 1712$



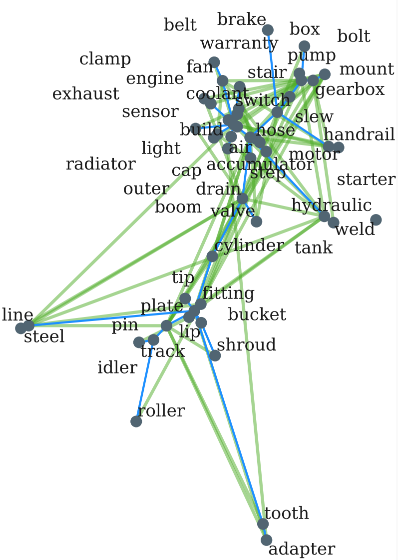
Excavator - Distribution Comparison ($l \geq 2$)

at optimal F_1-score:

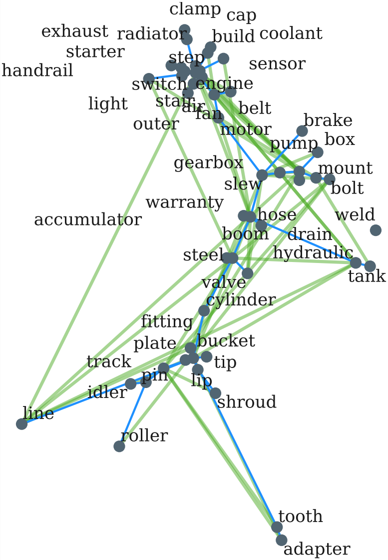


Excavator - Recovered Structure ($l \geq 2$)

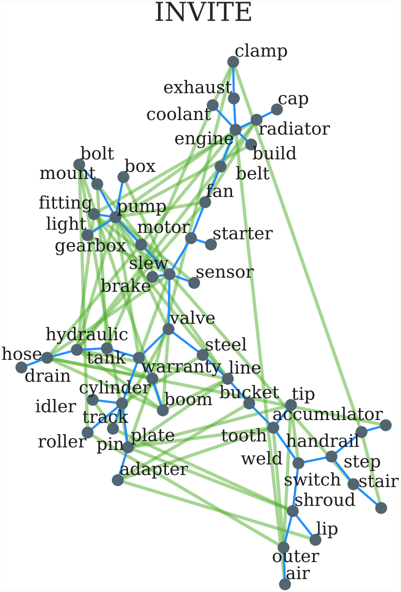
MC1

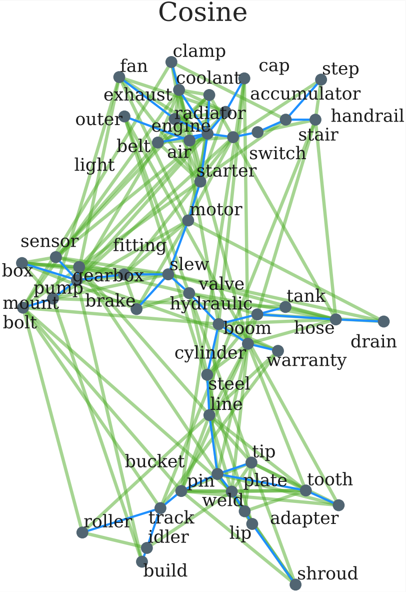


MC2



Excavator - Recovered Structure ($l \geq 2$)





References

[1] Thomas, D. S., and Thomas, D. S., 2018, *The Costs and Benefits of Advanced Maintenance in Manufacturing*, US Department of Commerce, National Institute of Standards; Technology.

[2] Sexton, T., Brundage, M. P., Hoffman, M., and Morris, K. C., 2017, "Hybrid Datafication of Maintenance Logs from Ai-Assisted Human Tags," *2017 Ieee International Conference on Big Data (Big Data)*, IEEE, pp. 1769–1777.

[3] 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.

[4] Jun, K.-S., Zhu, J., Rogers, T. T., Yang, Z., and others, 2015, “Human Memory Search as Initial-Visit Emitting Random Walk,” *Advances in Neural Information Processing Systems*, pp. 1072–1080.

[5] 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.

[6] 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,” *Annual Conference of the Prognostics and Health Management Society 2017*.

[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] 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."

[9] Sexton, T., Hodkiewicz, M., Brundage, M. P., and Smoker, T., 2018, "Benchmarking for Keyword Extraction Methodologies in Maintenance Work Orders," *PHM Society Conference*.

[10] Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B., 2004, "Learning with Local and Global Consistency," *Advances in Neural Information Processing Systems*, pp. 321–328.

[11] Goñi, J., Arrondo, G., Sepulcre, J., Martincorena, I., Mendizábal, N. V. de, Corominas-Murtra, B., Bejarano, B., Ardanza-Trevijano, S., Peraita, H., Wall, D. P., and others, 2011, "The Semantic Organization of the Animal Category: Evidence from Semantic Verbal Fluency and Network Theory," Cognitive processing, **12**(2), pp. 183–196.

[12] Doyle, P. G., and Snell, J. L., 2000, "Random Walks and Electric Networks," arXiv preprint math/0001057.

[13] 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," *NIPS-W*.