Graph Summarization Meets Outlier Detection

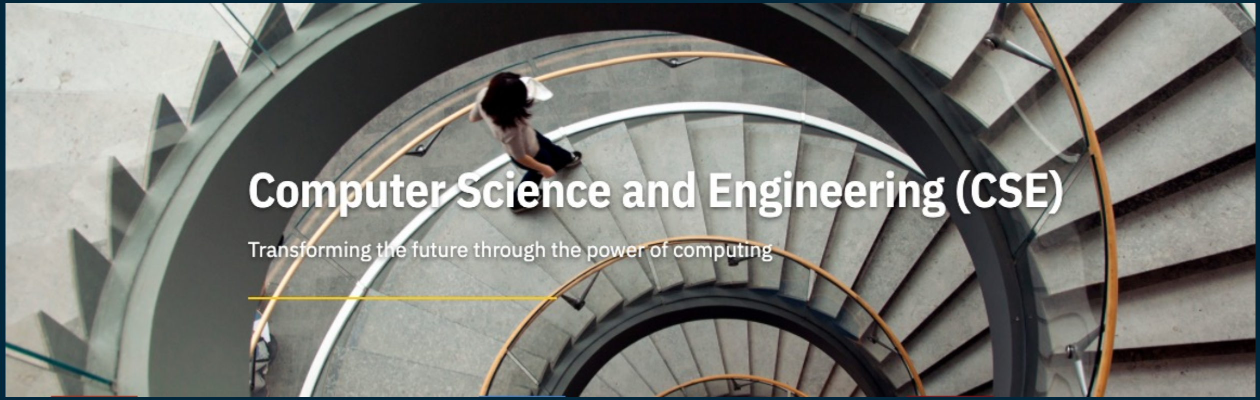
Danai Koutra
Morris Wellman Assistant Professor, CSE
Computational Medicine and Bioinformatics (courtesy)
Associate Director for AI, Michigan Institute for Data Science

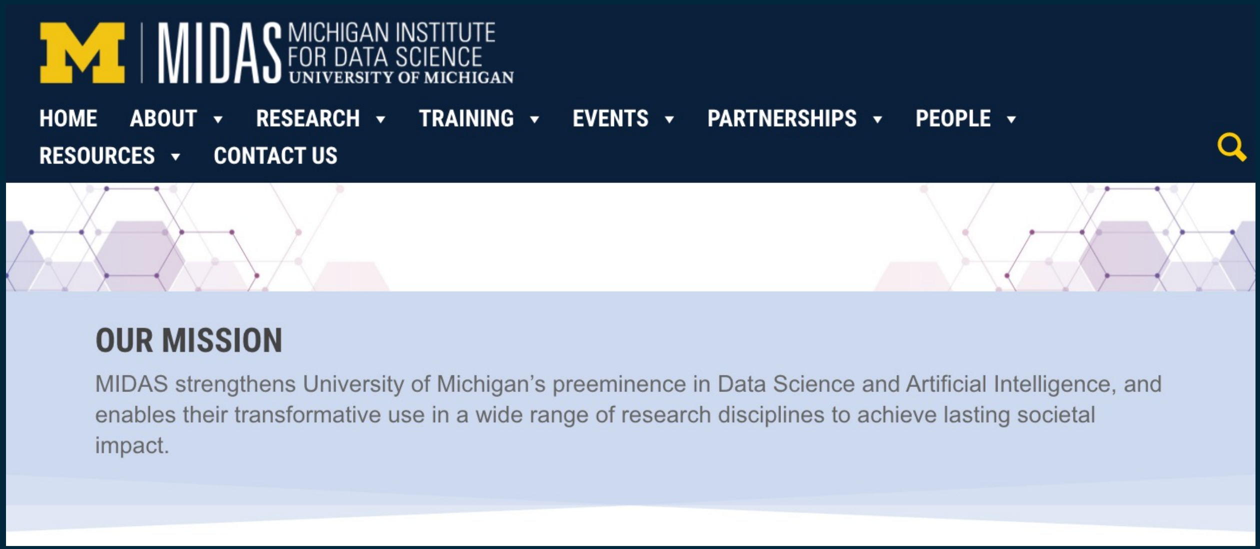
ACM SIGKDD, 6th Outlier Detection and Description (ODD) Workshop – August 15, 2021

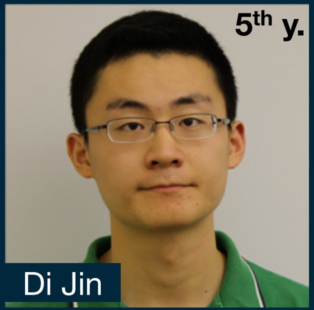
Joint work with: Caleb Belth, Christos Faloutsos, Brian Gallagher, Mark Heimann, Di Jin, Yike Liu, Ryan Rossi, Tara Safavi, Neil Shah, Chandra Sripada, Jilles Vreeken, Xinyi Zheng, …

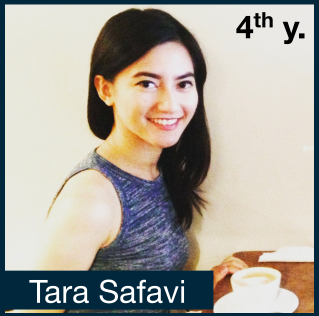
About me: Danai Koutra

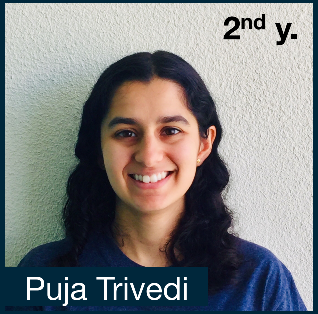
- Morris Wellman *soon-to-be* Assoc. Professor in CSE, University of Michigan (eff. Sep 1)
- Associate Director for AI, Michigan Institute for Data Science (MIDAS)











GEMS LAB

Welcome!

We are the **Graph Exploration and Mining at Scale (GEMS)** lab at the University of Michigan, founded and led by Danai Koutra. Our team researches important data mining and machine learning problems involving interconnected data: in other words, *graphs or networks*.

From airline flights to traffic routing to neuronal interactions in the brain, graphs are ubiquitous in the real world. Their properties and complexities have long been studied in fields ranging from mathematics to the social sciences. However, many pressing problems involving graph data are still open. One well-known problem is *scalability*. With continual advances in data generation and storage capabilities, the size of graph datasets has dramatically increased, making scalable graph methods indispensable. Another is the changing nature of data. Real graphs are almost always *dynamic*, evolving over time. Finally, many important problems in the social and biological sciences involve analyzing not one but *multiple* networks.

So, what do we do?

The problems described above call for **principled, practical, and highly scalable graph mining methods**, both theoretical and application-oriented. As such, our work connects to fields like linear algebra, distributed systems, deep learning, and even neuroscience. Some of our ongoing projects include:

- Algorithms for multi-network tasks, like matching nodes across networks
- Learning low-dimensional representations of networks in metric spaces
- Abstracting or "summarizing" a graph with a smaller network
- Analyzing network models of the brain derived from fMRI scans
- Distributed graph methods for iteratively solving linear systems
- Network-theoretical user modeling for various data science applications

We're grateful for funding from Adobe, Amazon, the Army Research Lab, the Michigan
for Data Science (MIDAS), Microsoft Azure, the National Science Foundation (NSF), and

Interested?

If you're interested in joining our group, send an email with your interests and CV to g
opportunities@umich.edu.

COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF MICHIGAN

MIDAS
MICHIGAN INSTITUTE FOR DATA SCIENCE
UNIVERSITY OF MICHIGAN

News

May 2020
1 paper accepted to KDD'20!

April 2020
Caleb receives an NDSEG Fellowship!

March 2020
Caleb receives an NSF GRFP!

February 2020
Danai receives a Google Faculty Research Award!

February 2020
Danai was recognised as an Outstanding Senior PC Member at WSDM'20!

January 2020
1 paper accepted to WebConf

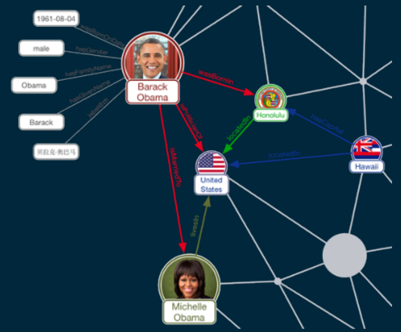
January 2020
Danai named Morris Wellman Professor!

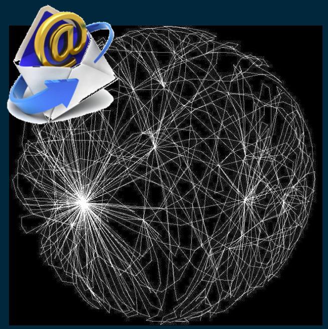
January 2020
Research Fellow Fatemeh Vahedian

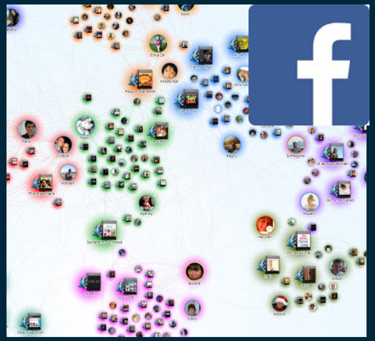
NSF
ARL
P&G
MIDAS
Adobe
amazon
Microsoft Azure

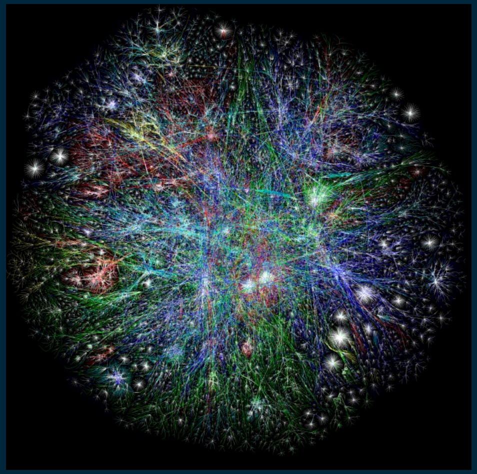
M CSE
GEMS LAB

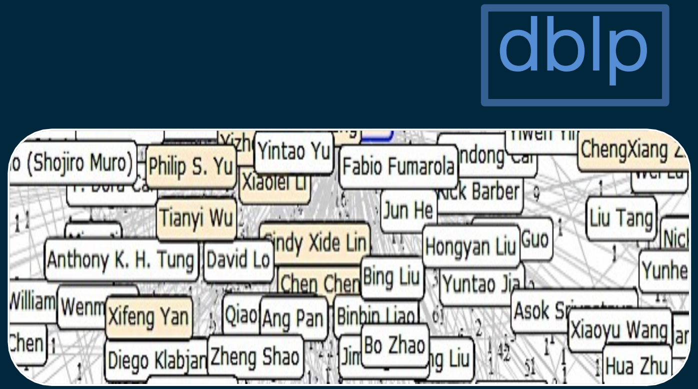
Graphs are everywhere!



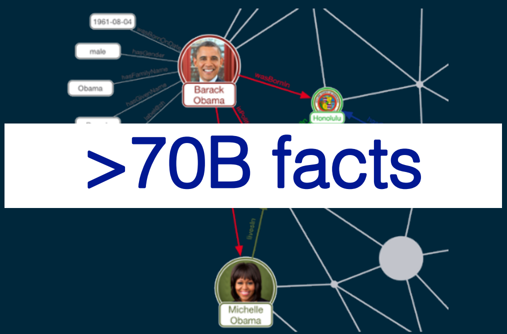


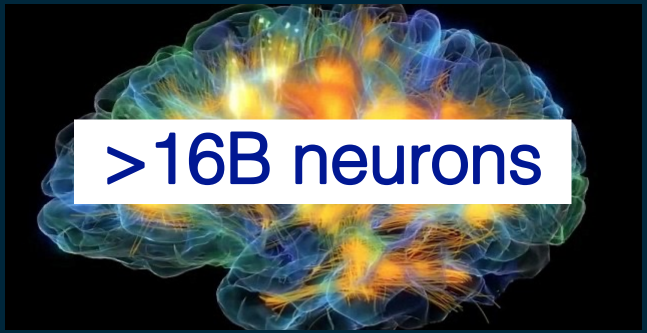


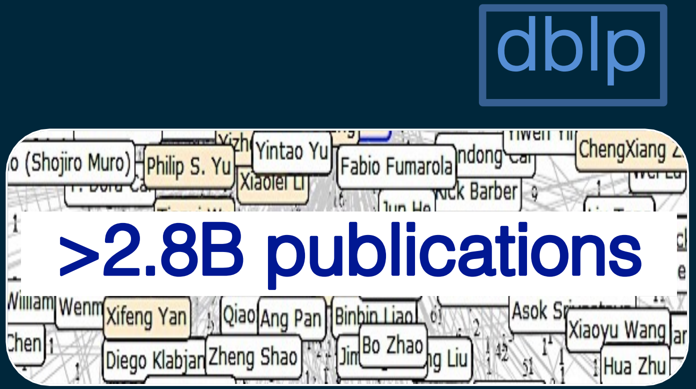




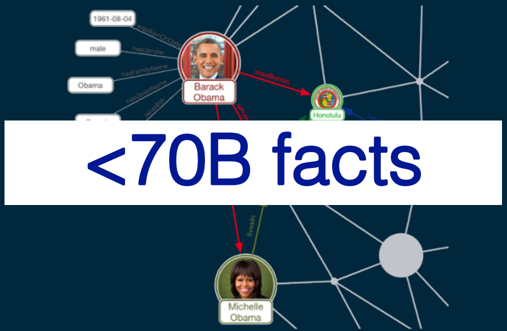
LARGE-scale Graph Data

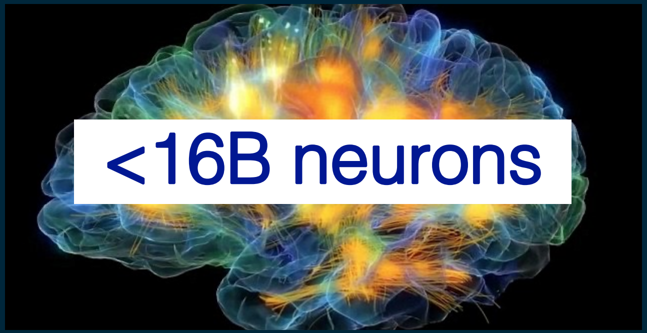






SUMMARIZATION of Big Datasets is Crucial!



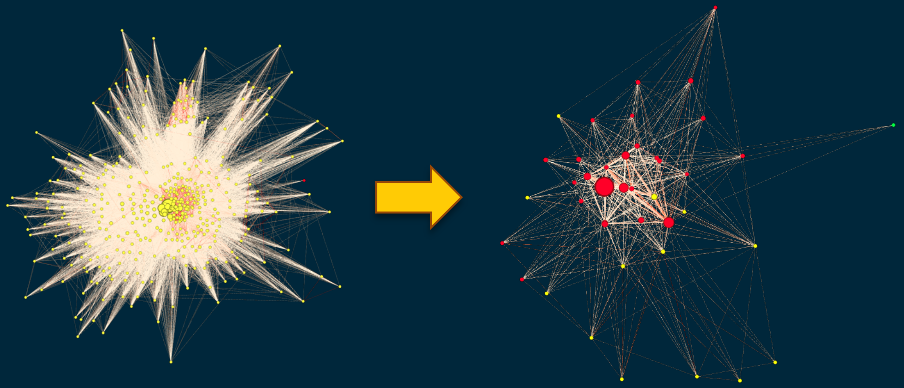


dblp

What is graph summarization?

Graph summarization seeks to find:

- a short representation of the input graph,
  - ✧ often in the form of an aggregated or sparsified graph, or a set of structures
- which reveals patterns in the original data and preserves specific structural or other properties, depending on the application domain.



Graph Summarization Methods and Applications: A Survey

YIKE LIU, TARA SAFAVI, ABHILASH DIGHE, and DANAI KOUTRA, University of Michigan, Ann Arbor

While advances in computing resources have made processing enormous amounts of data possible, human ability to identify patterns in such data has not scaled accordingly. Efficient computational methods for condensing and simplifying data are thus becoming vital for extracting actionable insights. In particular, while data summarization techniques have been studied extensively, only recently has summarizing interconnected data, or *graphs*, become popular. This survey is a structured, comprehensive overview of the state-of-the-art methods for summarizing graph data. We first broach the motivation behind and the challenges of graph summarization. We then categorize summarization approaches by the type of graphs taken as input and further organize each category by core methodology. Finally, we discuss applications of summarization on real-world graphs and conclude by describing some open problems in the field.

62

CCS Concepts: • **Mathematics of computing** → **Graph algorithms**; • **Information systems** → **Data mining; Summarization**; • **Human-centered computing** → *Social network analysis*; • **Theory of computation** → *Unsupervised learning and clustering*; • **Computing methodologies** → *Network science*;

Additional Key Words and Phrases: Graph mining, graph summarization

ACM Reference format:
Yike Liu, Tara Safavi, Abhilash Dighe, and Danai Koutra. 2018. Graph Summarization Methods and Applications: A Survey. *ACM Comput. Surv.* 51, 3, Article 62 (June 2018), 34 pages.
https://doi.org/10.1145/3186727

1 INTRODUCTION

As technology advances, the amount of data that we generate and our ability to collect and archive such data both increase continuously. Daily activities like social media interaction, web browsing, product and service purchases, itineraries, and wellness sensors generate large amounts of data, the analysis of which can immediately impact our lives. This abundance of generated data and its velocity call for data summarization, one of the main data mining tasks.

Since summarization facilitates the identification of structure and meaning in data, the data mining community has taken a strong interest in the task. Methods for a variety of data types

Y. Liu and T. Safavi contributed equally to this article.

This material was based on work supported in part by the National Science Foundation under grant IIS 1743088, Trove, and the University of Michigan. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or other funding parties. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

Authors' addresses: Y. Liu, T. Safavi, A. Dighe, and D. Koutra, Bob and Betty Beyster Building, 2260 Hayward St, Ann Arbor, MI 48109; emails: {yikeliu, tsafavi, adighe, dkoutra}@umich.edu.

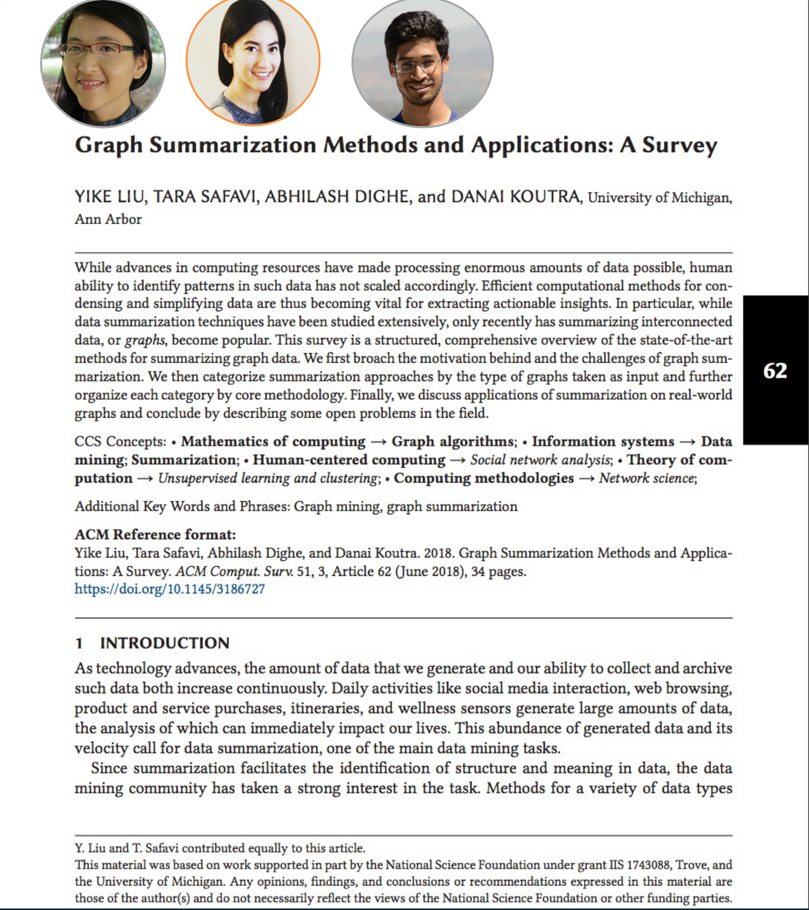
https://doi.org/10.1145/3186727

ACM Computing Surveys, Vol. 51, No. 3, Article 62. Publication date: June 2018.

[Yike Liu, Tara Safavi, Abhilash Dighe, Danai Koutra. ACM Computing Surveys '18]

Why graph summarization?

- Reduction of data volume + storage
  - ✧ e.g., fewer I/O operations
- Speedup of algorithms + queries
- Interactive analysis
- Influence analysis and understanding
- Noise elimination -> reveals patterns
- Privacy preservation



Graph Summarization Methods and Applications: A Survey

YIKE LIU, TARA SAFAVI, ABHILASH DIGHE, and DANAI KOUTRA, University of Michigan, Ann Arbor

While advances in computing resources have made processing enormous amounts of data possible, human ability to identify patterns in such data has not scaled accordingly. Efficient computational methods for condensing and simplifying data are thus becoming vital for extracting actionable insights. In particular, while data summarization techniques have been studied extensively, only recently has summarizing interconnected data, or *graphs*, become popular. This survey is a structured, comprehensive overview of the state-of-the-art methods for summarizing graph data. We first broach the motivation behind and the challenges of graph summarization. We then categorize summarization approaches by the type of graphs taken as input and further organize each category by core methodology. Finally, we discuss applications of summarization on real-world graphs and conclude by describing some open problems in the field.

62

CCS Concepts: • **Mathematics of computing** → **Graph algorithms**; • **Information systems** → **Data mining**; **Summarization**; • **Human-centered computing** → *Social network analysis*; • **Theory of computation** → *Unsupervised learning and clustering*; • **Computing methodologies** → *Network science*;

Additional Key Words and Phrases: Graph mining, graph summarization

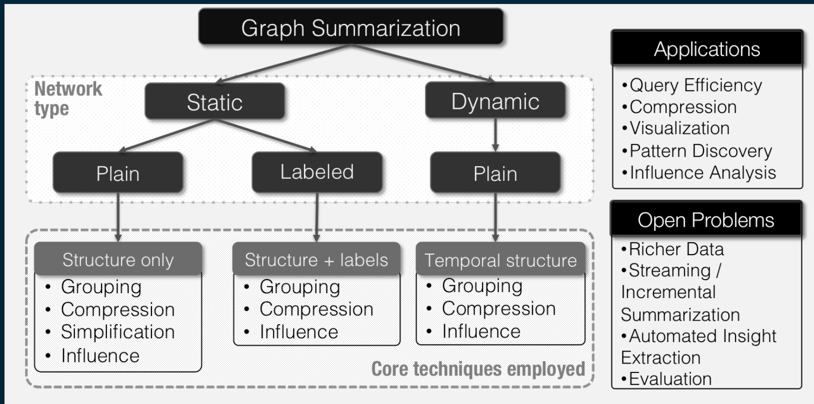
ACM Reference format:
Yike Liu, Tara Safavi, Abhilash Dighe, and Danai Koutra. 2018. Graph Summarization Methods and Applications: A Survey. *ACM Comput. Surv.* 51, 3, Article 62 (June 2018), 34 pages.
https://doi.org/10.1145/3186727

1 INTRODUCTION

As technology advances, the amount of data that we generate and our ability to collect and archive such data both increase continuously. Daily activities like social media interaction, web browsing, product and service purchases, itineraries, and wellness sensors generate large amounts of data, the analysis of which can immediately impact our lives. This abundance of generated data and its velocity call for data summarization, one of the main data mining tasks.

Since summarization facilitates the identification of structure and meaning in data, the data mining community has taken a strong interest in the task. Methods for a variety of data types

Y. Liu and T. Safavi contributed equally to this article.
This material was based on work supported in part by the National Science Foundation under grant IIS 1743088, Trove, and the University of Michigan. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or other funding parties.

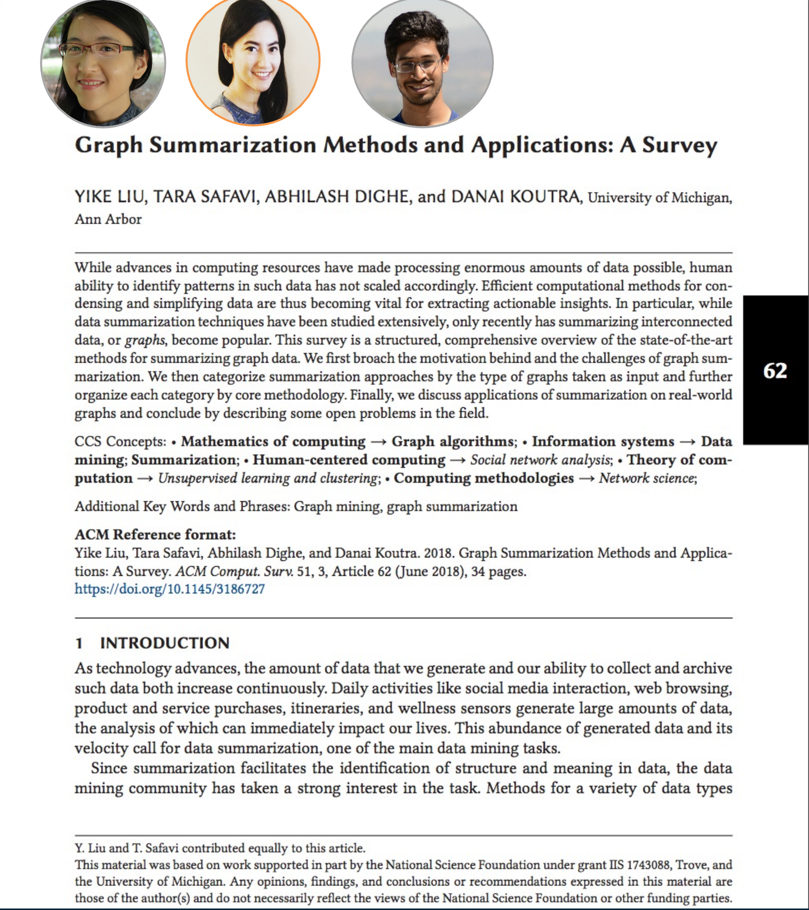


[Yike Liu, Tara Safavi, Abhilash Dighe, Danai Koutra. ACM Computing Surveys '18]

Why graph summarization?

- Reduction of data volume + storage
  - ✧ e.g., fewer I/O operations
- Speedup of algorithms + queries
- Interactive analysis
- Influence analysis and understanding
- Noise elimination -> reveals patterns
- Privacy preservation

\+ anomalies



Graph Summarization Methods and Applications: A Survey

YIKE LIU, TARA SAFAVI, ABHILASH DIGHE, and DANAI KOUTRA, University of Michigan, Ann Arbor

While advances in computing resources have made processing enormous amounts of data possible, human ability to identify patterns in such data has not scaled accordingly. Efficient computational methods for condensing and simplifying data are thus becoming vital for extracting actionable insights. In particular, while data summarization techniques have been studied extensively, only recently has summarizing interconnected data, or *graphs*, become popular. This survey is a structured, comprehensive overview of the state-of-the-art methods for summarizing graph data. We first broach the motivation behind and the challenges of graph summarization. We then categorize summarization approaches by the type of graphs taken as input and further organize each category by core methodology. Finally, we discuss applications of summarization on real-world graphs and conclude by describing some open problems in the field.

62

CCS Concepts: • **Mathematics of computing** → **Graph algorithms**; • **Information systems** → **Data mining**; **Summarization**; • **Human-centered computing** → *Social network analysis*; • **Theory of computation** → *Unsupervised learning and clustering*; • **Computing methodologies** → *Network science*;

Additional Key Words and Phrases: Graph mining, graph summarization

ACM Reference format:
Yike Liu, Tara Safavi, Abhilash Dighe, and Danai Koutra. 2018. Graph Summarization Methods and Applications: A Survey. *ACM Comput. Surv.* 51, 3, Article 62 (June 2018), 34 pages.
https://doi.org/10.1145/3186727

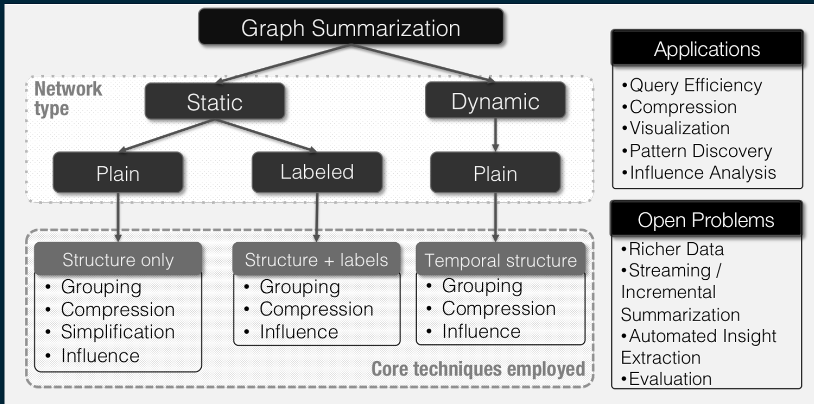
1 INTRODUCTION

As technology advances, the amount of data that we generate and our ability to collect and archive such data both increase continuously. Daily activities like social media interaction, web browsing, product and service purchases, itineraries, and wellness sensors generate large amounts of data, the analysis of which can immediately impact our lives. This abundance of generated data and its velocity call for data summarization, one of the main data mining tasks.

Since summarization facilitates the identification of structure and meaning in data, the data mining community has taken a strong interest in the task. Methods for a variety of data types

Y. Liu and T. Safavi contributed equally to this article.

This material was based on work supported in part by the National Science Foundation under grant IIS 1743088, Trove, and the University of Michigan. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or other funding parties.

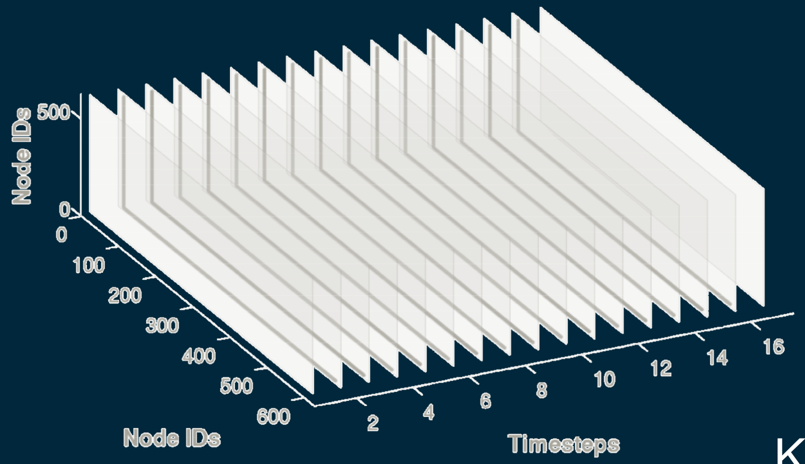


[Yike Liu, Tara Safavi, Abhilash Dighe, Danai Koutra. ACM Computing Surveys ’18]

Summarizing Large Networks: Overview

Survey: [CSUR'18]

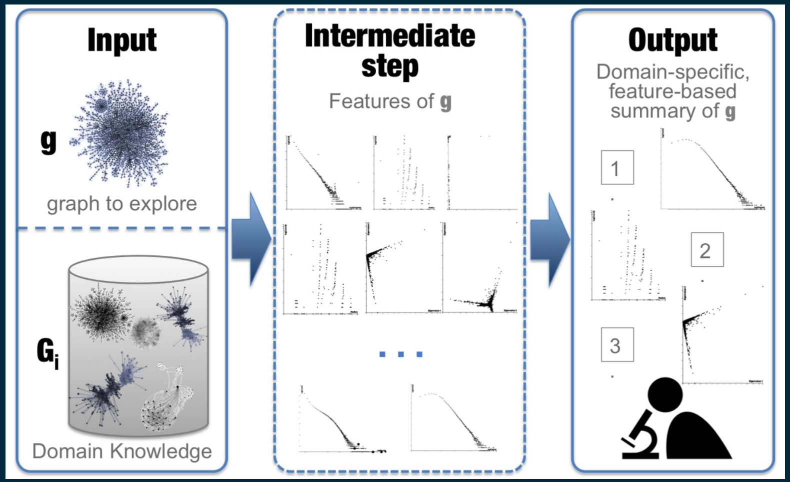
Structural Summaries



[SDM'14, KDD'15, Dat Bull Eng'17, SNAM'18, SDM'19, KDD'19a, KDD'20…]

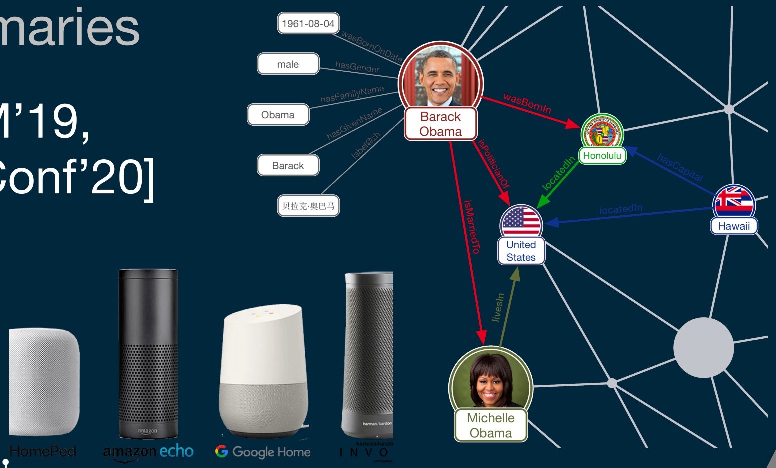
Domain-specific Summaries

[ICDM'17, KDD'19]



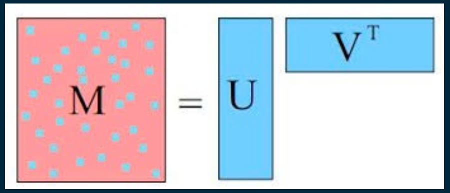
Query-on-the-edge + Rule-based Summaries

[ICDM'19, WebConf'20]



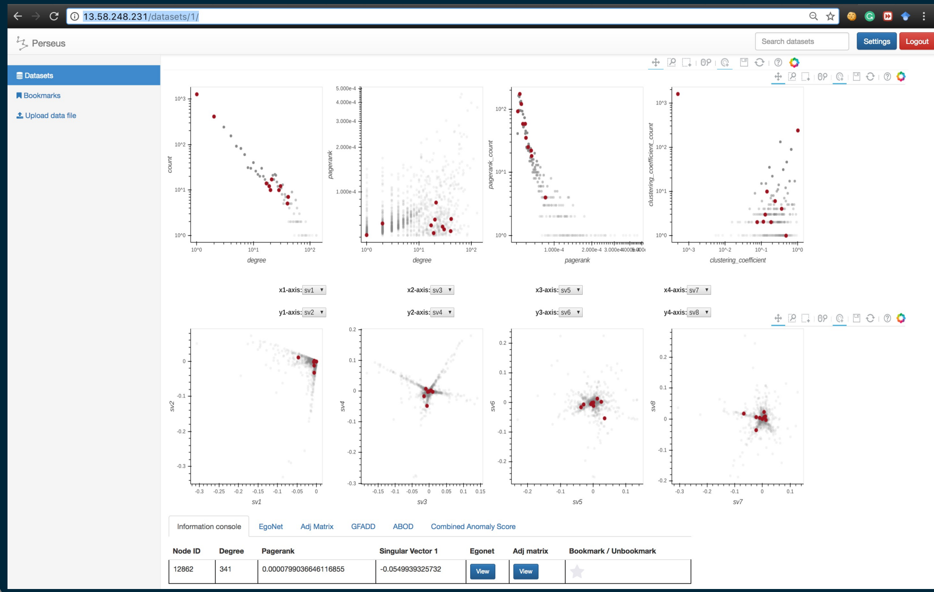
Latent Summaries

[KDD'19a, b]



Interactive Summaries

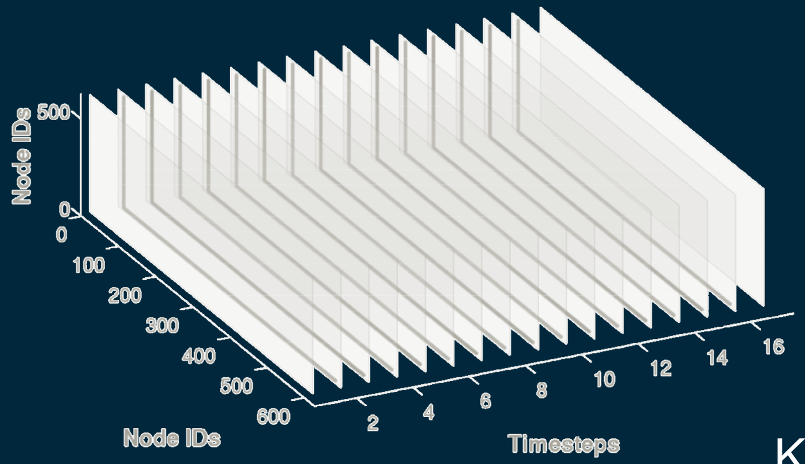
[VLDB'15, Informatics'17]



M CSE

GEMS LAB

This talk: Summarization Meets Outlier Detection



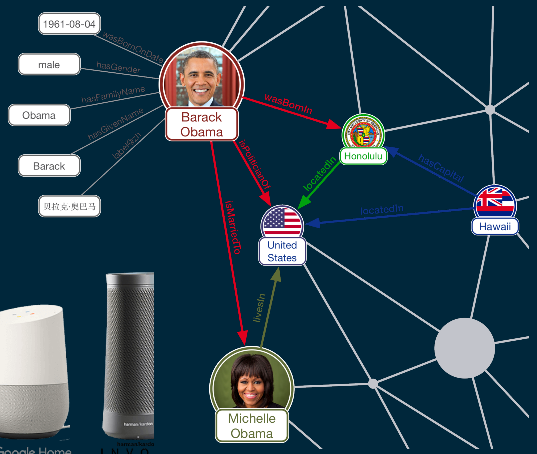
Structural
Summaries

[SDM'14, KDD'15,
Dat Bull Eng'17,
SNAM'18,
SDM'19,
KDD'19a, KDD'20…]

Query-on-the-edge + Rule-based

Summaries

[ICDM'19,
WebConf'20]

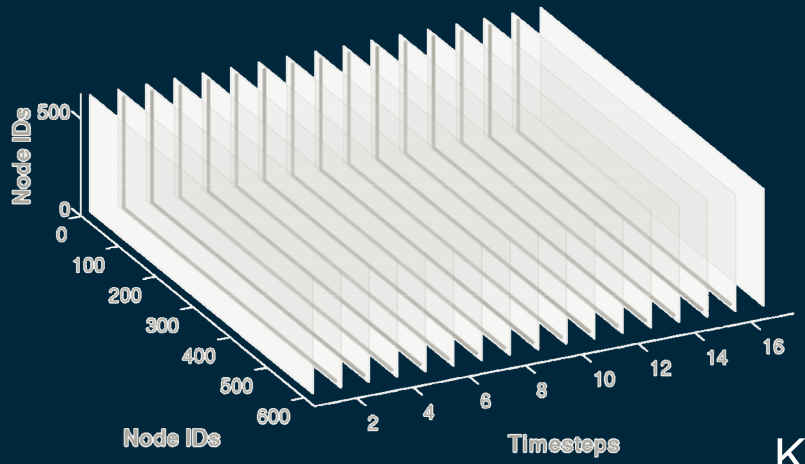


② Graph Streams:
Persistent and bursty activity detection
[KDD'20]

① Knowledge Graphs:
Unified error detection and completion
[WebConf'20]

M CSE
GEMS LAB

This talk: Summarization Meets Outlier Detection



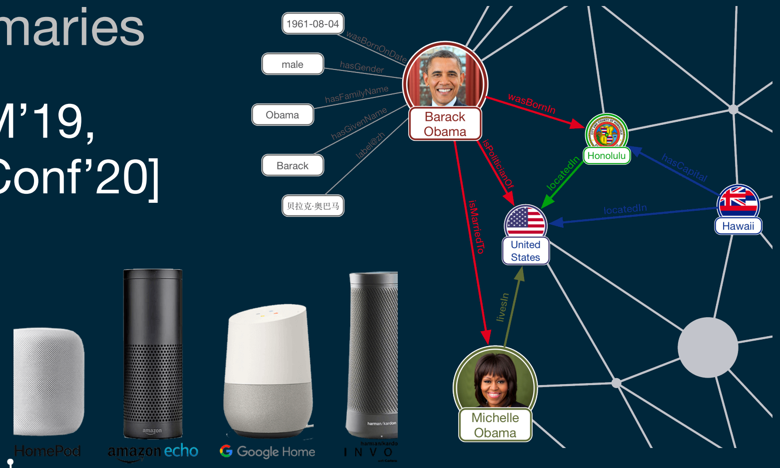
Structural Summaries

[SDM'14, KDD'15, Dat Bull Eng'17, SNAM'18, SDM'19, KDD'19a, KDD'20…]

② Graph Streams:
Persistent and bursty activity detection
[KDD'20]

Query-on-the-edge + Rule-based Summaries

[ICDM'19, WebConf'20]



① Knowledge Graphs:
Unified error detection and completion
[WebConf'20]

M CSE
GEMS LAB

Knowledge graphs (KGs)

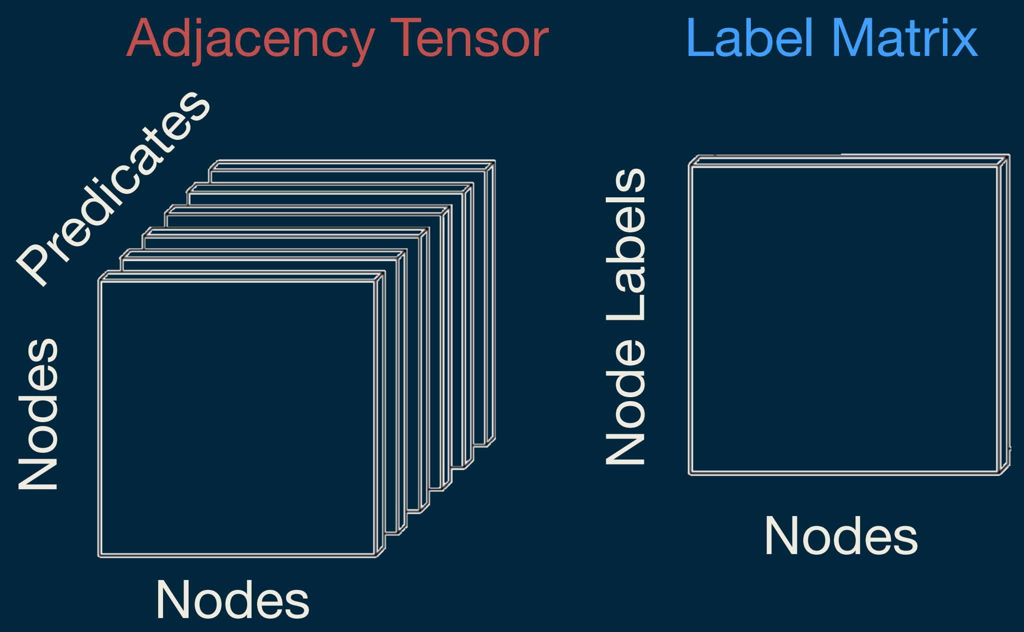
store general information about the world in the structure of a graph

edge = triple (subject node or head,
predicate or relation,
object node or tail)

M CSE GEMS LAB

Knowledge graphs (KGs)

can be represented as labeled, directed multi-relational graphs



practice

DOI:10.1145/3331166

Article development led by acmqueue
queue.acm.org

Five diverse technology companies show how it's done.

BY NATASHA NOY, YUQING GAO, ANSHU JAIN, ANANT NARAYANAN, ALAN PATTERSON, AND JAMIE TAYLOR

Industry-Scale Knowledge Graphs: Lessons and Challenges

KNOWLEDGE GRAPHS ARE critical to many enterprises today: They provide the structured data and factual knowledge that drive many products and make them more intelligent and "magical."

In general, a knowledge graph describes objects of interest and connections between them. For example, a knowledge graph may have nodes for a movie, the actors in this movie, the director, and so on. Each node may have properties such as an actor's name and age. There may be nodes for multiple movies involving a particular actor. The user can then traverse the knowledge graph to collect information on all the movies in which the actor appeared or, if applicable, directed.

Many practical implementations impose constraints on the links in knowledge graphs by defining a *schema* or *ontology*. For example, a link from a movie to its director must connect an object of type Movie to an object of type Person. In some cases the links themselves might have their own properties: a link connecting an actor and a movie might have the name of the specific role the actor played. Similarly, a link connecting a politician with a specific role in government might have the time period during which the politician held that role.

Knowledge graphs and similar structures usually provide a shared substrate of knowledge within an organization, allowing different products and applications to use similar vocabulary and to reuse definitions and descriptions that others create. Furthermore, they usually provide a compact formal representation that developers can use to infer new facts and build up the knowledge—for example, using the graph connecting movies and actors to find out which actors frequently appear in movies together.

This article looks at the knowledge graphs of five diverse tech companies, comparing the similarities and differences in their respective experiences of building and using the graphs, and discussing the challenges that all knowledge-driven enterprises face today. The collection of knowledge graphs discussed here covers the breadth of applications, from search, to product descriptions, to social networks:

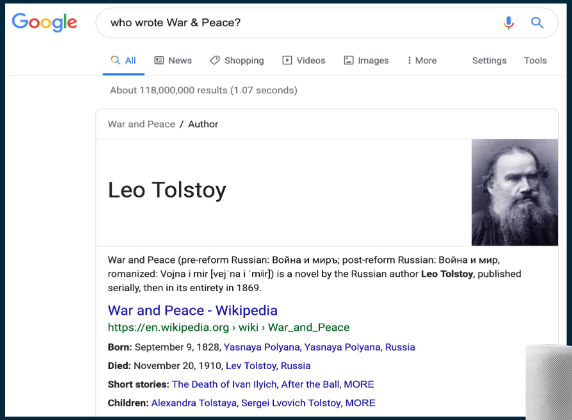
- Both Microsoft's Bing knowledge graph and the Google Knowledge Graph support search and answering questions in search and during conversations. Starting with the descriptions and connections of people, places, things, and organizations, these graphs include general knowledge about the world.
- Facebook has the world's largest social graph, which also includes information about music, movies, celebrities, and places that Facebook users care about.

Google
amazon
IBM
Microsoft
in
Bloomberg
AI2
Adobe
ebay

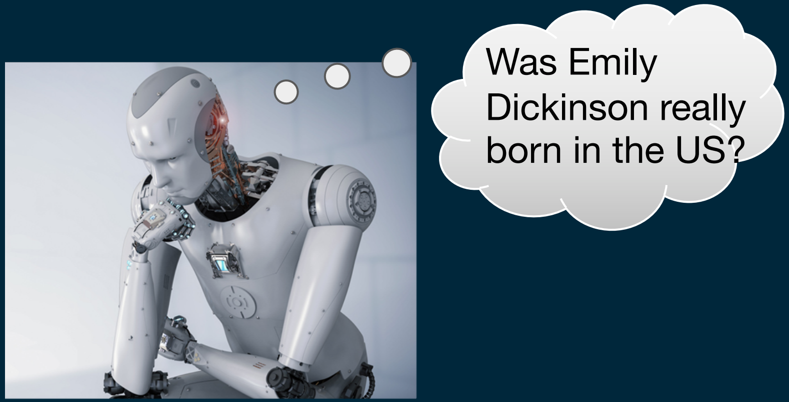
yago select knowledge
Freebase™
WIKIDATA
DBpedia
WORDNET
NELL

Applications of KGs

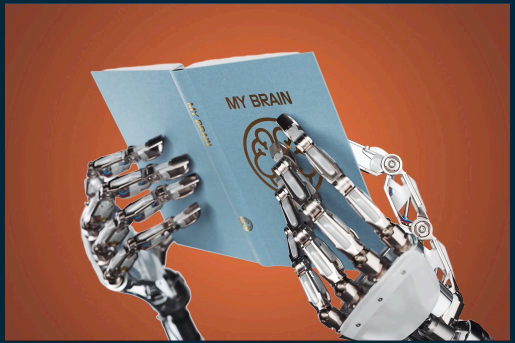
Question Answering & Chatbots



Automatic Fact Checking



Reading Comprehension



Semantic search
Biomedical applications
...

Financial applications
Recommendation systems
...

KGs are constructed via

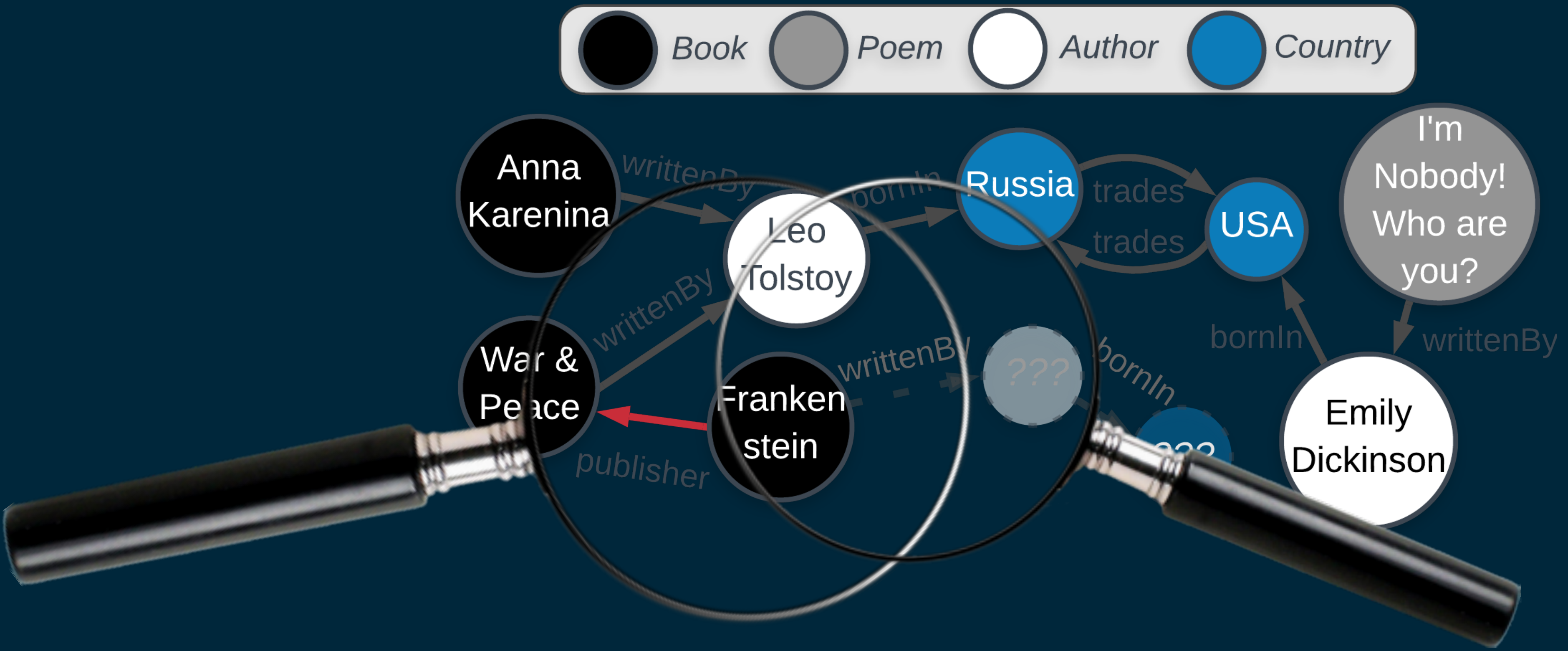
Crowd Sourcing

Web Crawling

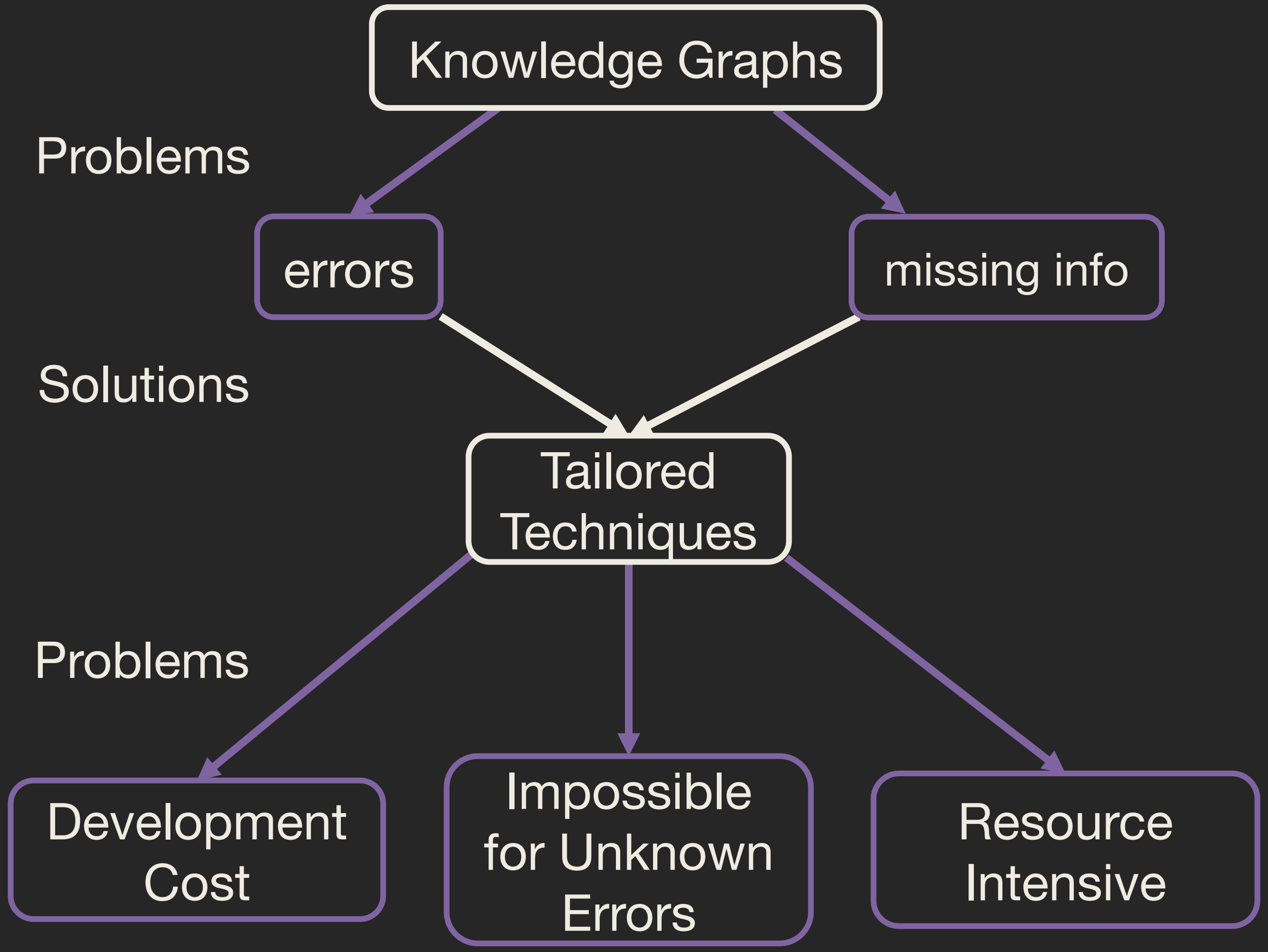
Gao, Liang, Han, Yakout, Mohamed. Building a Large-scale, Accurate and Fresh Knowledge Graph. Tutorial @ KDD’18.
Zalmout, Zhang, Li, Liang, Dong. All You Need to Know to Build a Product Knowledge Graph. Tutorial @ KDD'21.

...which leads to

errors and miss some information



Current Approach



Current Approach

Knowledge Graphs

Problems

errors

missing info

Solutions

Tailored Techniques

Problems

Development Cost

Impossible for Unknown Errors

Resource Intensive

Proposed Approach

Knowledge Graphs

Problem

abnOrmAl: errors & missing info

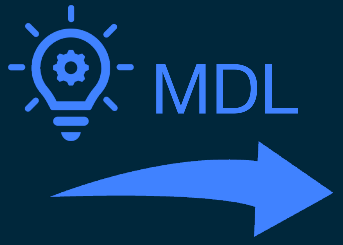
Solution

inductive summarization

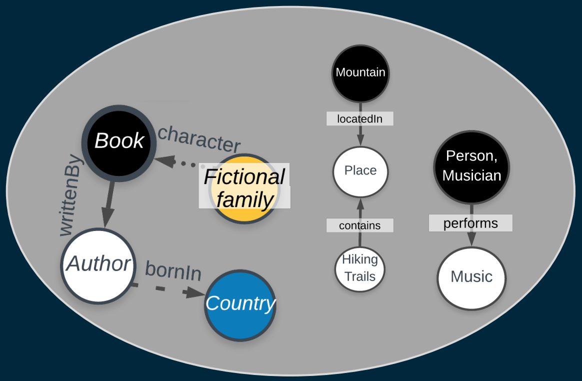
KGIST: Knowledge Graph Inductive SummarizaTion

Find a concise summary M of knowledge graph G, consisting of inductive, soft **rules** s.t.

$$\min L(G, M) = \underbrace{L(M)}_{\text{bits to describe } M} + \underbrace{L(G|M)}_{\text{bits to describe } G \text{ with } M}$$



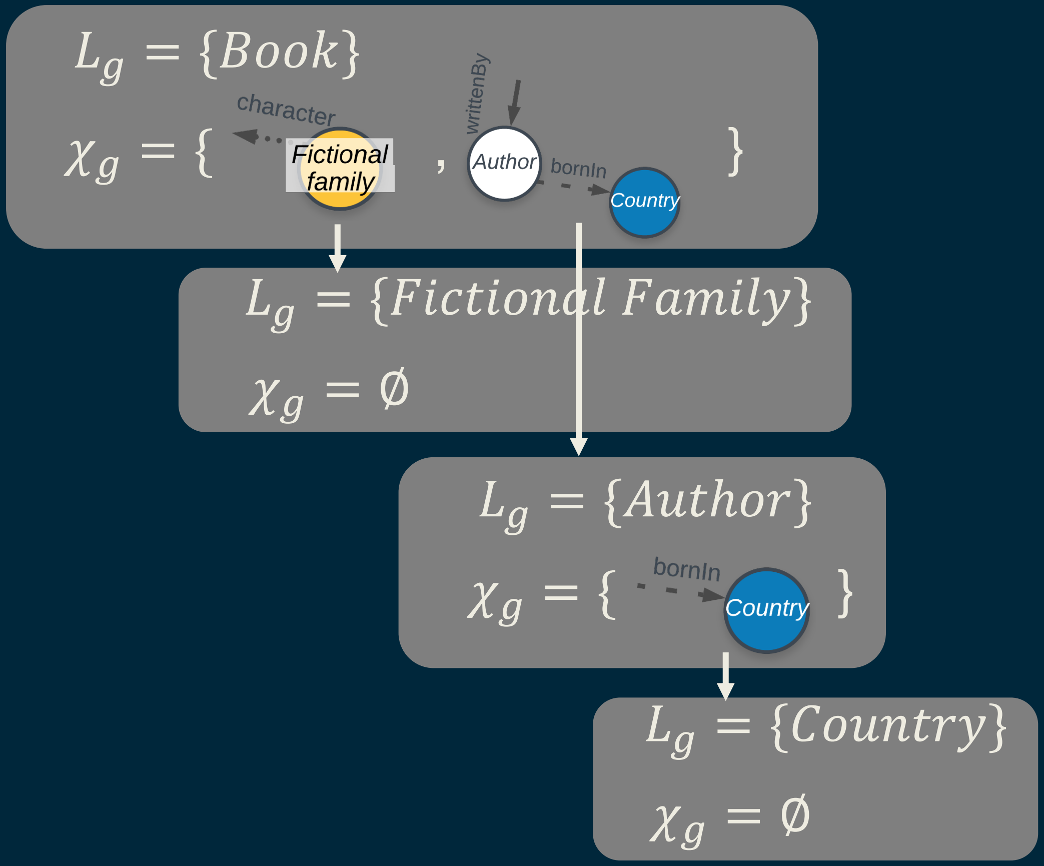
$M =$

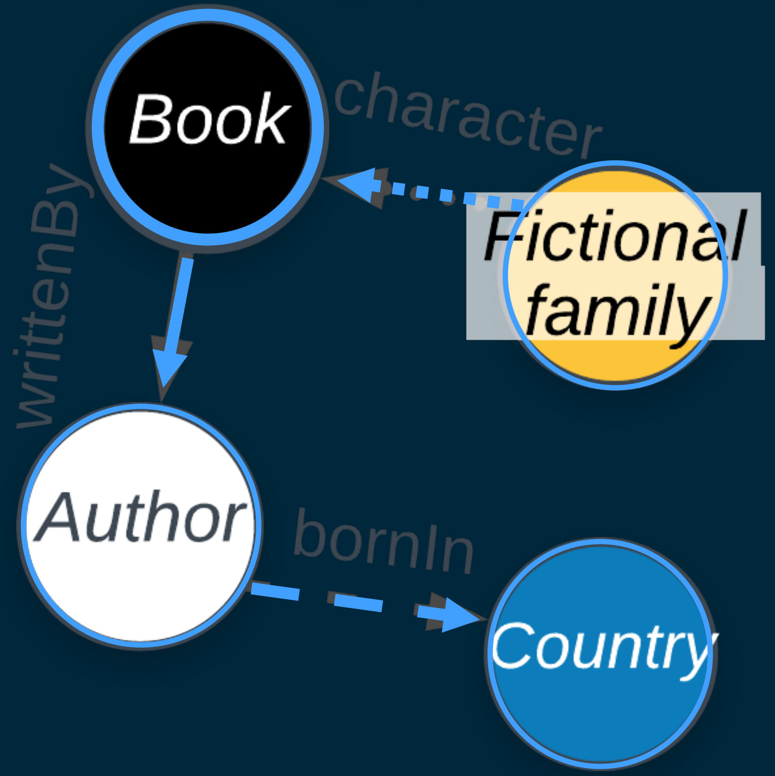


Rule $g = (L_g, \chi_g)$ = (root label, children rules)

We formulate rules recursively as rooted, directed, and labeled graphs

- A rule asserts things about nodes with the root labels, L_g





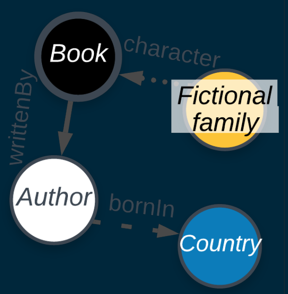
Correct assertions $\mathcal{A}_c^{(g)}$ & Exceptions $\mathcal{A}_\xi^{(g)}$

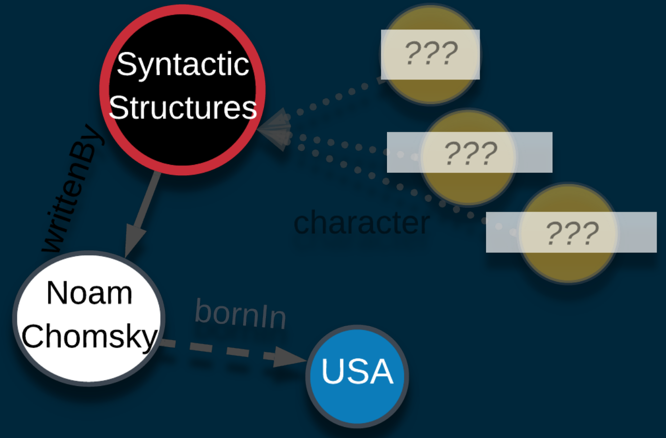
Guided traversals
that a rule implies

Failed guided traversals

instantiation of rule for
book “War & Peace”

Rule g





Exceptions indicate errors
or missing information

[Caleb Belth, Xinyi Zheng, et al. WWW ’20]

github.com/GemsLab/KGIST

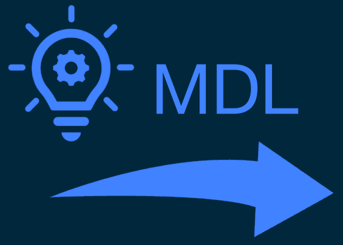
KGIST: Knowledge Graph Inductive SummarizaTion

Find a concise summ…

consisting of i…

min $L(G, M$…

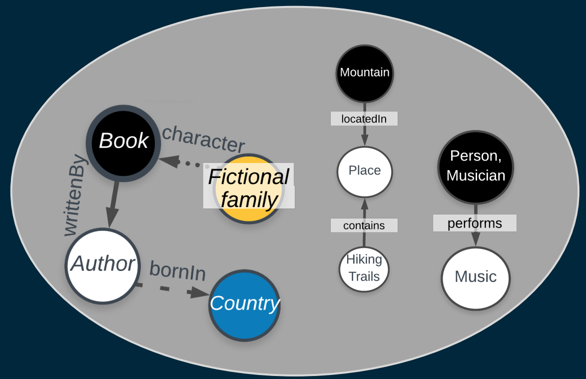
bits to describe M … bits to describ… M



$M =$

normal

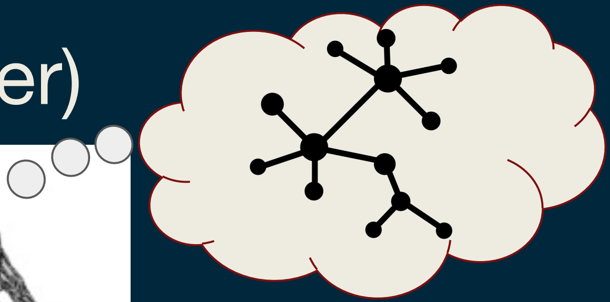
abnormal



Exceptions + unexplained parts

Deriving $L(G, M) = L(M) + L(G|M)$

Alice (sender)



Hey Alice, could you tell me about your KG?

Bob (receiver)

MDL Model: Overview

Alice

Sure! I'll send:

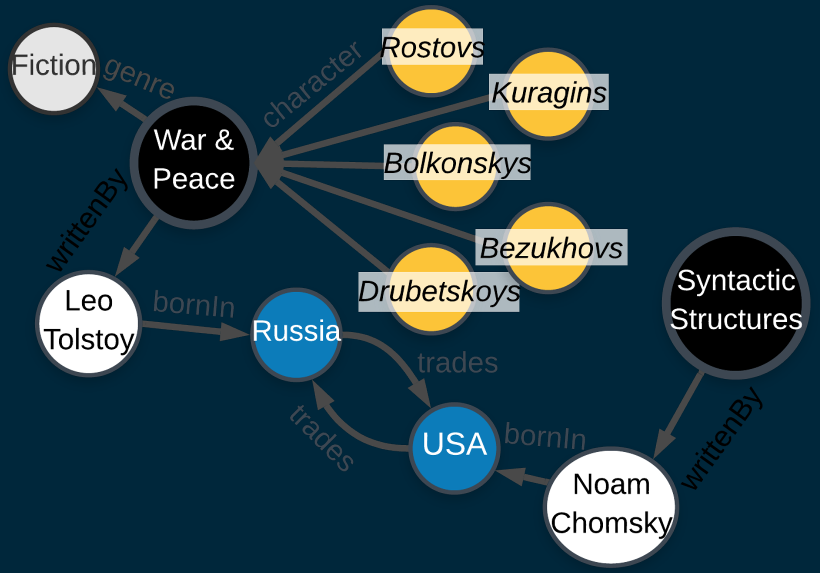
1) Model-independent information
2) A model M
3) Any error the model makes

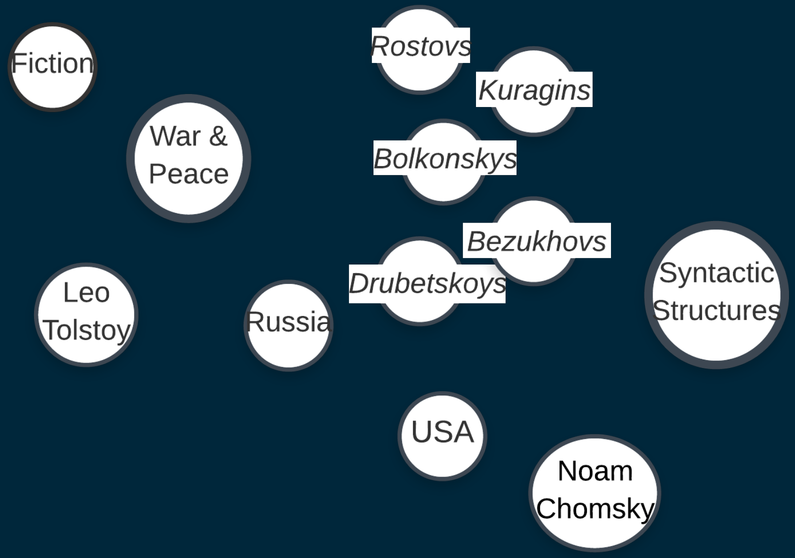
Ok, send the model the minimizes

$$L(G, M) = L(M) + L(G|M)$$

Bob

MDL Model: $L(G, M) = L(M) + L(G|M)$





Alice

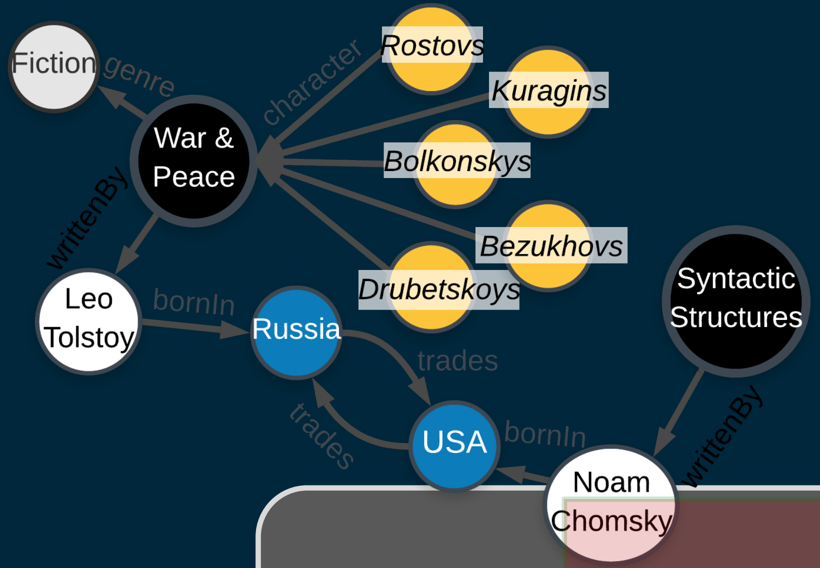
Model independent info:
nodes, # edges, node ids …

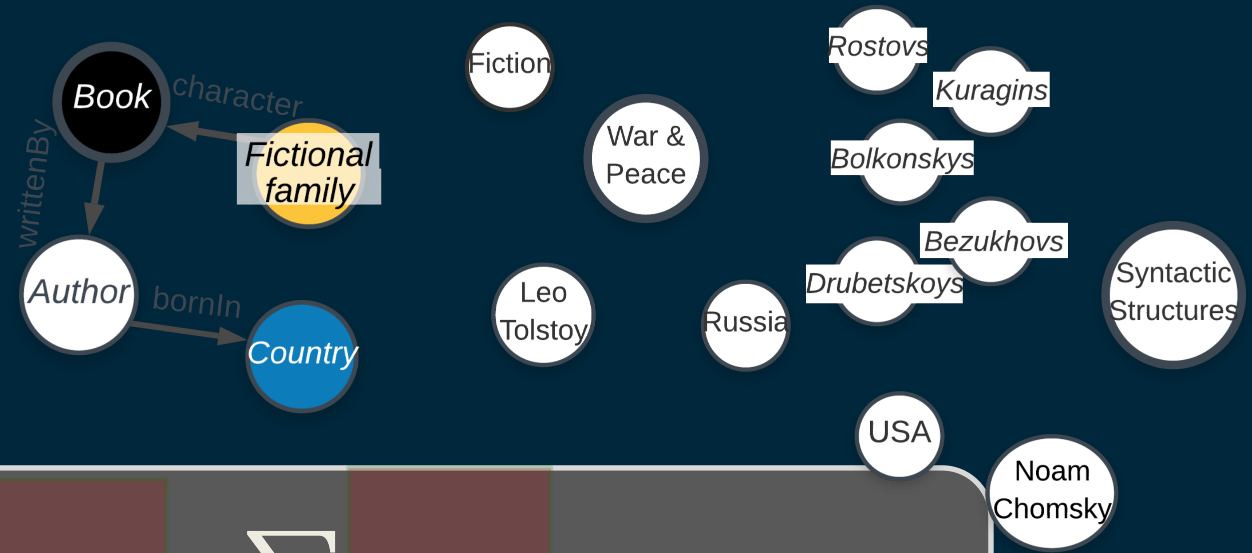
Bob

[Caleb Belth, Xinyi Zheng, et al. WWW ’20]

github.com/GemsLab/KGIST

MDL Model: $L(G, M) = L(M) + L(G|M)$



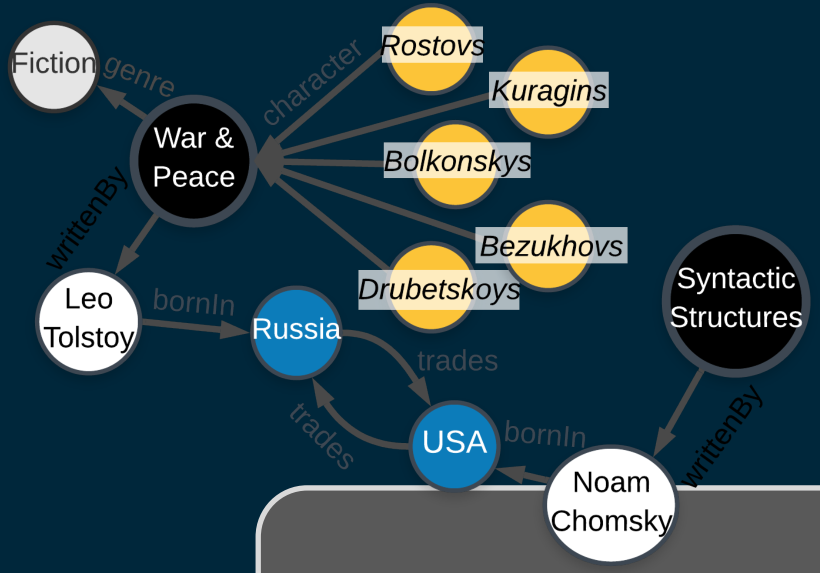


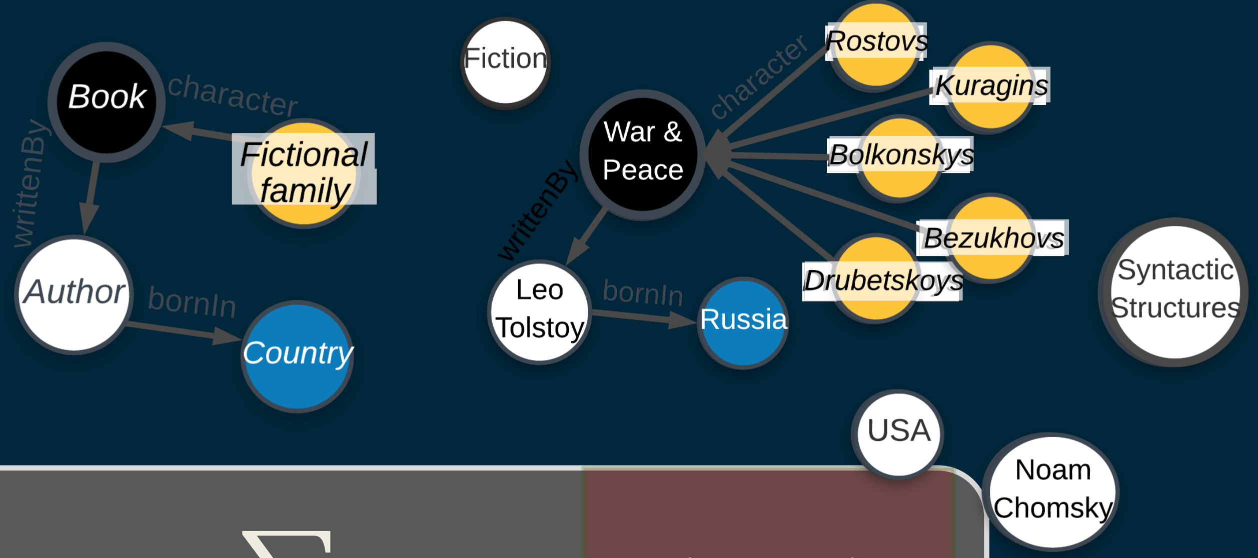
$$L(M) = \underbrace{\log(2 * |L_{\mathcal{V}}|^2 + |L_{\mathcal{E}}| + 1)}_{\text{\# rules}} + \sum_{g \in M} \Big(\underbrace{L(g)}_{\text{rule}} + \underbrace{L(\mathcal{A}^{(g)})}_{\text{assertions}} \Big)$$

Alice

Bob

MDL Model: $L(G,M) = L(M) + L(G|M)$





$$L(M) = \underbrace{\log(2 * |L_{\mathcal{V}}|^2 + |L_{\mathcal{E}}| + 1)}_{\#\text{ rules}} + \sum_{g \in M} \Big(\underbrace{L(g)}_{\text{rule}} + \underbrace{L(\mathcal{A}^{(g)})}_{\text{assertions}}\Big)$$

Alice

Bob

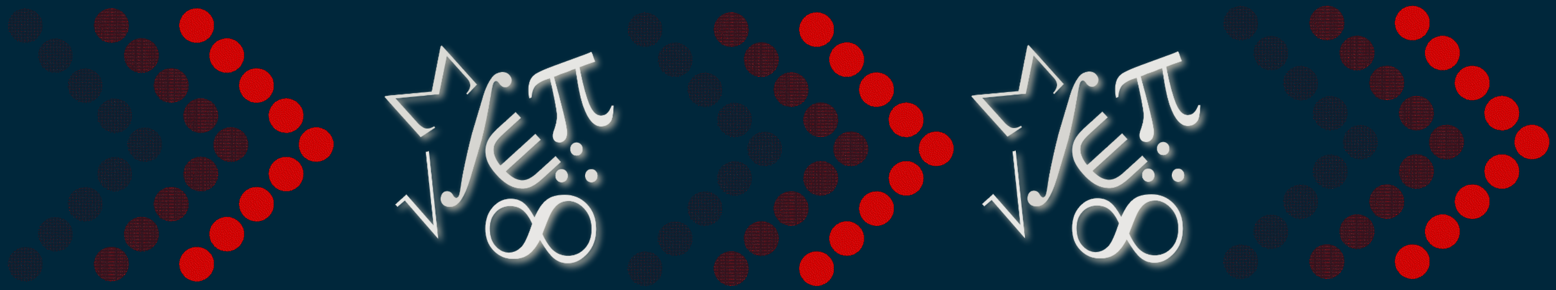
Great, now let me find all the books!

MDL Model: $L(G, M) = L(M) + L(G|M)$

- Alice continues with the assertions, traversals etc…

- Done with the definition of $L(M)$

Alice



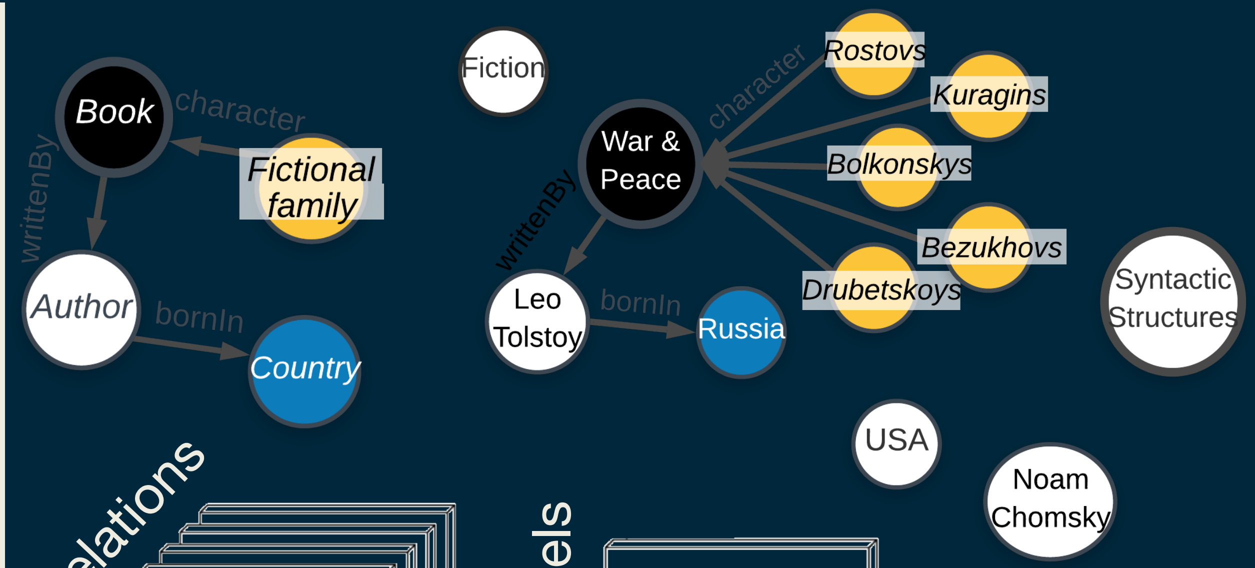
Bob

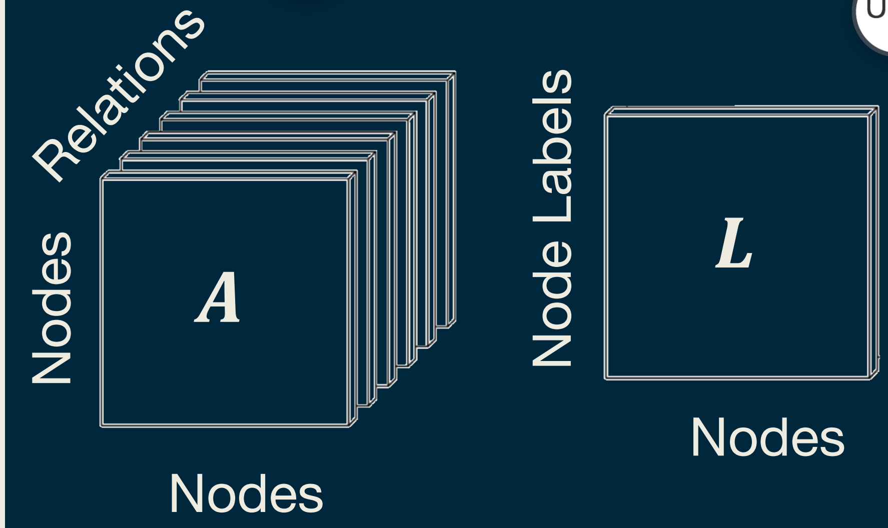
MDL Model: $L(G, M) = L(M) + L(G|M)$

$$L(G|M) = \mathrm{L}(\boldsymbol{L}^-) + L(\boldsymbol{A}^-)$$

Alice

I'll send the 1s in $\boldsymbol{L}$ and $\boldsymbol{A}$ that the rules didn't reveal



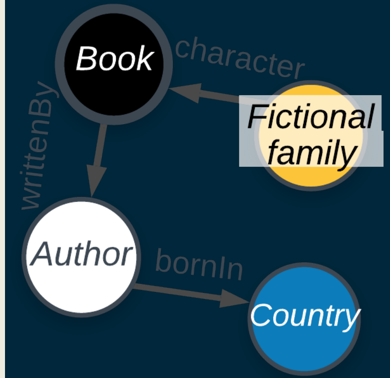


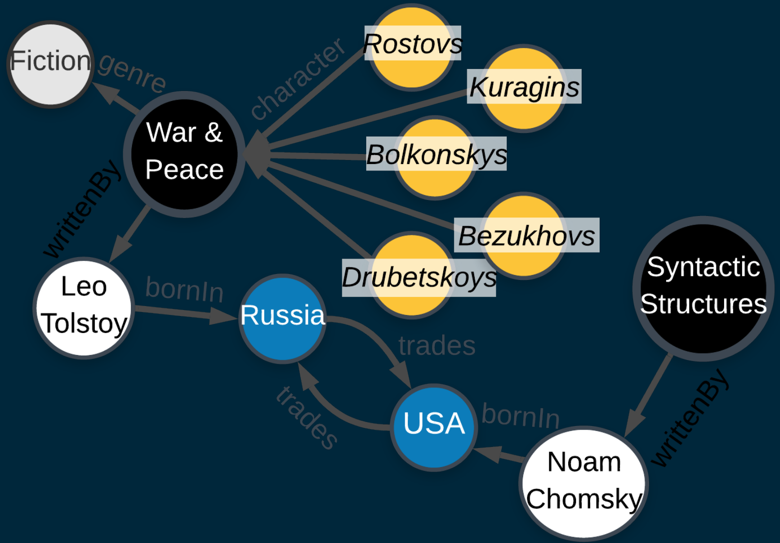
Bob

MDL Model: $L(G, M) = L(M) + L(G|M)$

There you go!

Alice





Bob

[Caleb Belth, Xinyi Zheng, et al. WWW '20]

github.com/GemsLab/KGIST

KGIST: Knowledge Graph Inductive SummarizaTion

Find a concise summary M of knowledge graph G, consisting of inductive, soft rules s.t.

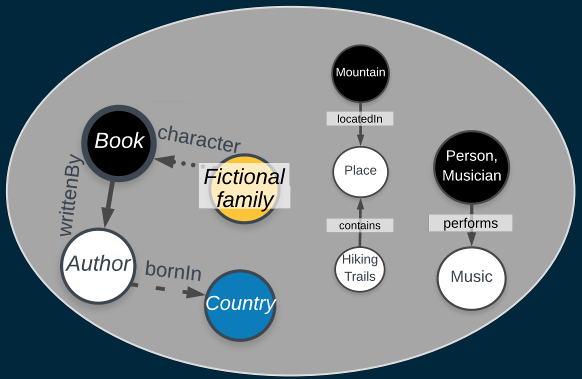
start with simple candidate rules & compose

$$\min L(G,M) = \underbrace{L(M)}_{\text{bits to describe } M} + \underbrace{L(G|M)}_{\text{bits to describe } G \text{ with } M}$$

?

MDL

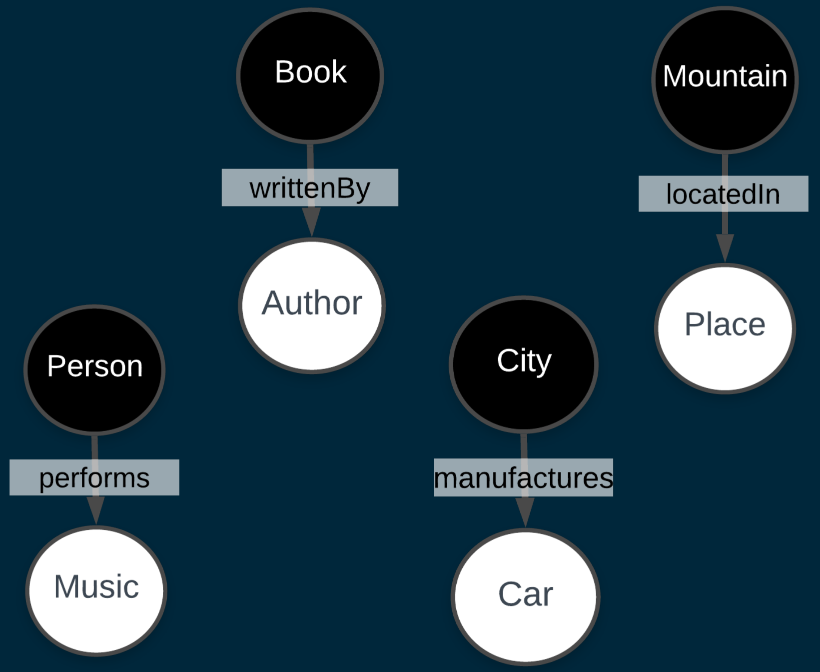
$M =$



[Caleb Belth, Xinyi Zheng, et al. WWW '20] github.com/GemsLab/KGIST

KGIST Method: Overview

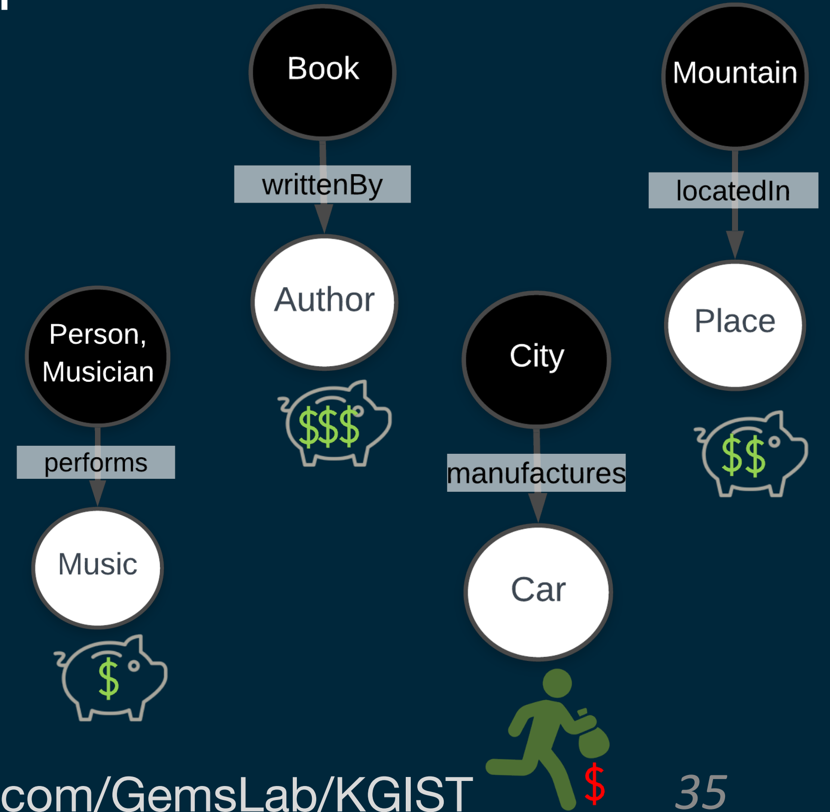
1. Generate candidate rules



KGIST Method: Overview

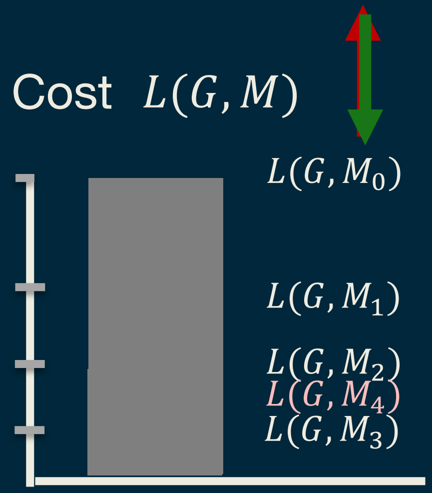
1. Generate candidate rules
2. Rank candidate rules
   - ✧ Based on how much they help explain/compress the KG

$$\underbrace{\Delta L(G|M_0 \cup \{g\})}_{\text{how much } g \text{ explains}} = \underbrace{L(G|M_0)}_{\text{\# bits w/o } g} - \underbrace{L(G|M_0 \cup \{g\})}_{\text{\# bits with } g}$$

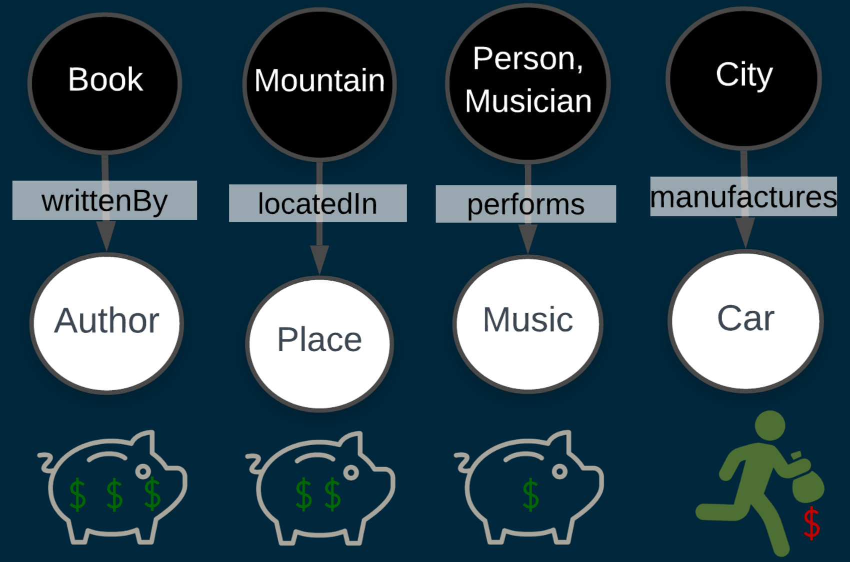


KGist Method: Overview

1. Generate candidate rules
2. Rank candidate rules
   - ✧ Based on how much they help explain/compress the KG
3. Select rules
   - ✧ Based on minimizing $L(G, M)$

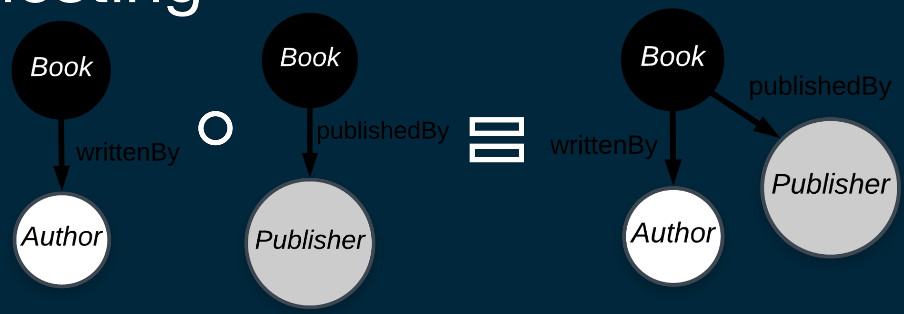


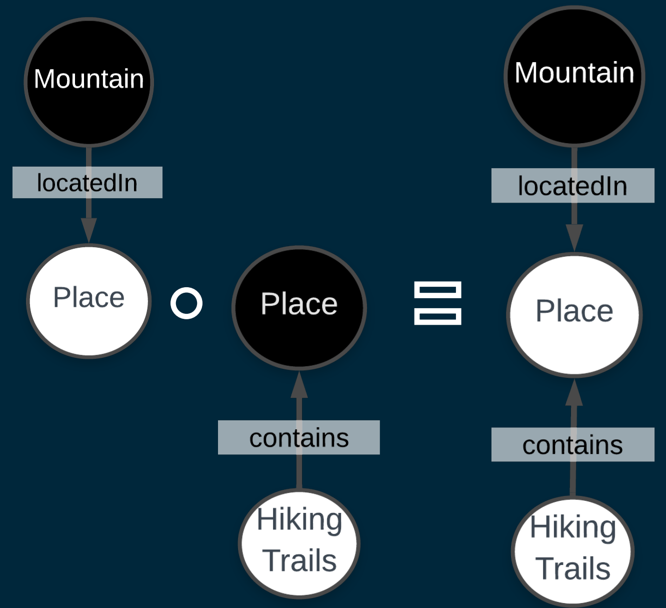
$M = \emptyset$



KGIST Method: Overview

1. Generate candidate rules
2. Rank candidate rules
   - Based on how much they help explain/compress the KG
3. Select rules
   - Based on minimizing $L(G, M)$
4. Refine rules
   - Merging and nesting





Current Approach

Knowledge Graphs

Problems

errors

missing info

Solutions

Tailored Techniques

Problems

Development Cost

Impossible for Unknown Errors

Resource Intensive

Proposed Approach: KGIST

Knowledge Graphs

Problem

abnOrmAl: errors & missing info

Solution

inductive summarization

M CSE GEMS LAB [Caleb Belth, Xinyi Zheng, et al. WWW ’20] https://github.com/GemsLab/KGIST

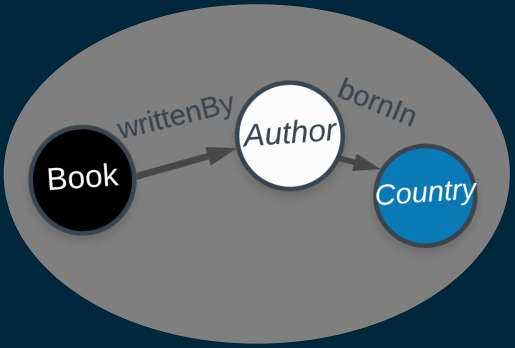
KGist Anomaly Scores

- Anomalous entities: violate many rules
  - ✧ *MDL intuition*: many bits to describe a node as an exception

$$\eta(v) = \sum_{\substack{g\in r(v): \\ v\in\mathcal{A}_\xi^{(g)}}} \frac{1}{|\mathcal{A}_\xi^{(g)}|}\log\binom{|\mathcal{A}^{(g)}|}{|\mathcal{A}_\xi^{(g)}|}$$ = # bits pointing out v as an exception

Alice

$M =$



Bob

KGIST Anomaly Scores

- Anomalous entities: violate many rules
  - *MDL intuition*: many bits to describe a node as an exception
- Anomalous triples: unexplained edges ($L(G|M)$) + anomalous endpoints

$$\eta(s,p,o) = \underbrace{\eta(s) + \eta(o)}_{\text{node endpoints}} + \underbrace{\boxed{\eta^{(p)}(s,p,o)}}_{\text{predicate}}$$

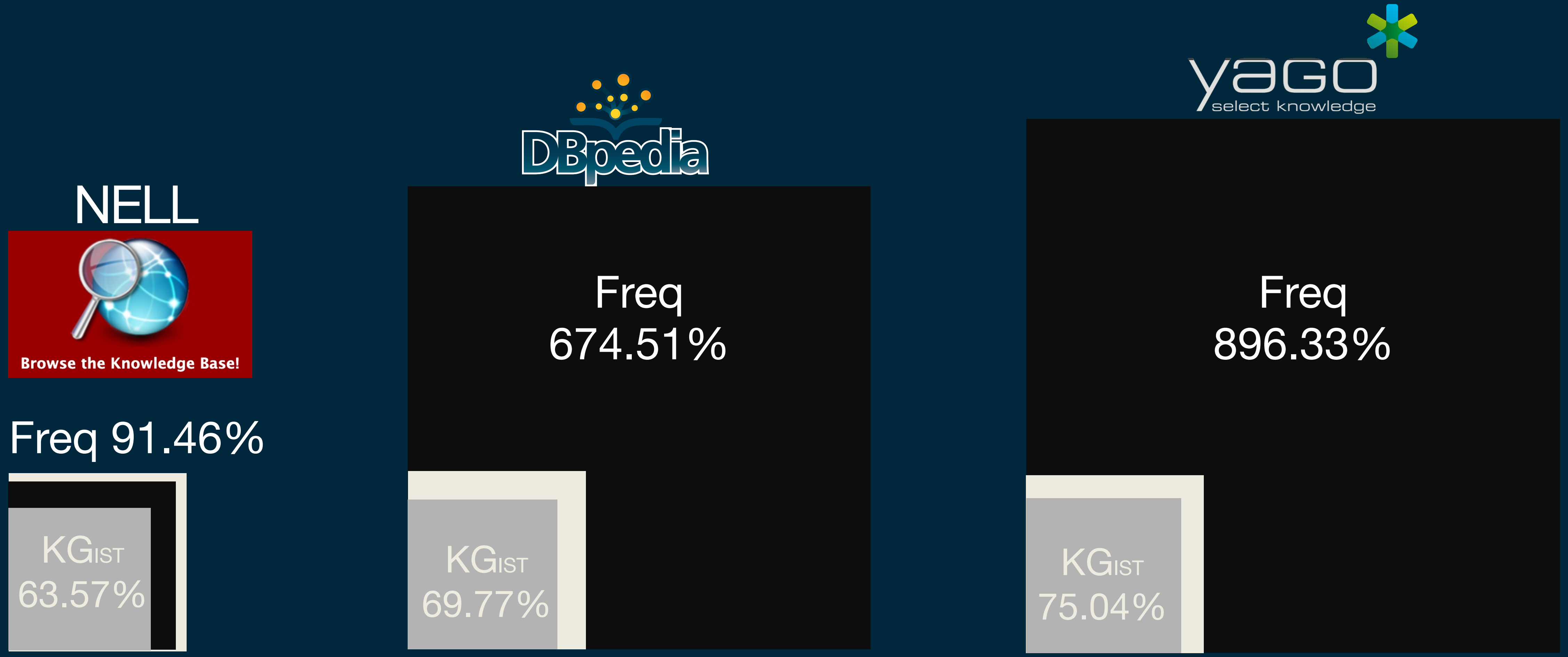
Alice

$$\boxed{\eta^{(p)}(s,p,o)} = \begin{cases} \frac{1}{|\boldsymbol{A}^{-}|} * \log\left(\frac{|\mathcal{V}|^2 * |L_{\mathcal{E}}| - |\boldsymbol{A}_M|}{|\boldsymbol{A}^{-}|}\right) & \text{if } \boldsymbol{A}^{-}_{s,o,p} = 1 \\ 0 & \text{otherwise} \end{cases}$$

= # bits describing unexplained triple

Bob

KGIST compresses real KGs significantly



② Findings

KGIST detects various types of errors

Browse the Knowledge Base!

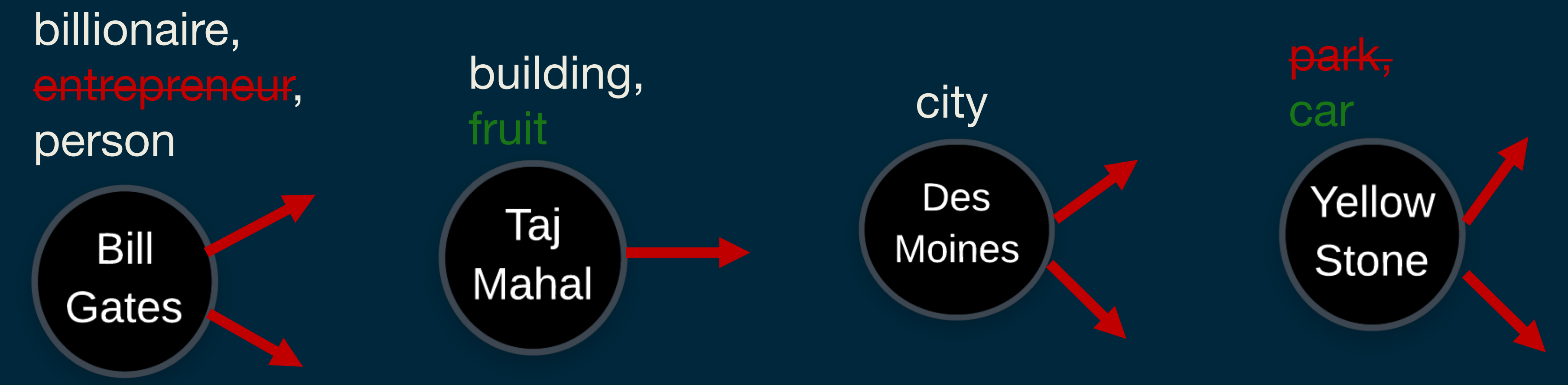
Metric	Supervised		Unsupervised			
	ComplEx	TransE	SDValidate	AMIE+	KGIST_FREQ	KGIST+m
AUC						
P@100						
R@100						
F1@100						

Select q% of all nodes and,

$\lesssim 0.0188$

$\lesssim 0.0369$

remove label | add label | inject 1 or 2 edges | replace label



KGIST performs best across all types of anomalies.

3 Findings

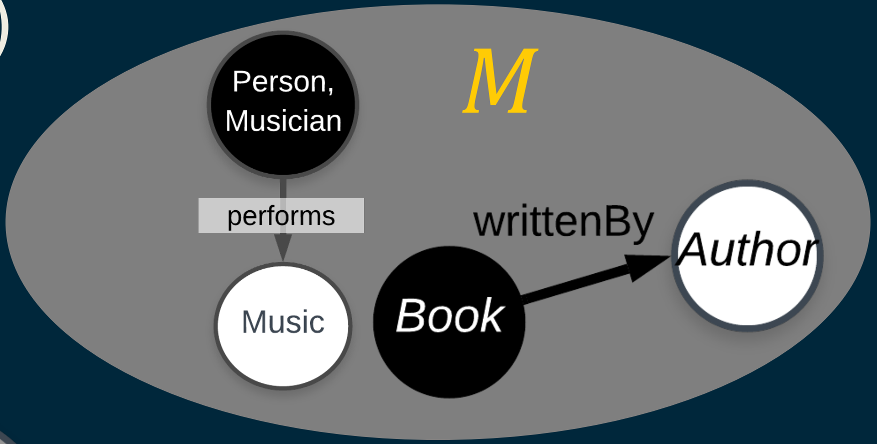
KGist identifies where information is missing

1. Remove entities / nodes (e.g. *Mary Shelley*)

Findings ③

KGIST identifies where information is missing

1. Remove entities / nodes (e.g. *Mary Shelley*)
2. Run KGIST on perturbed graph
3. Find *where* entities are missing



KGIST identifies where information is missing

Recall for location of missing node

Recall for location + label of missing node

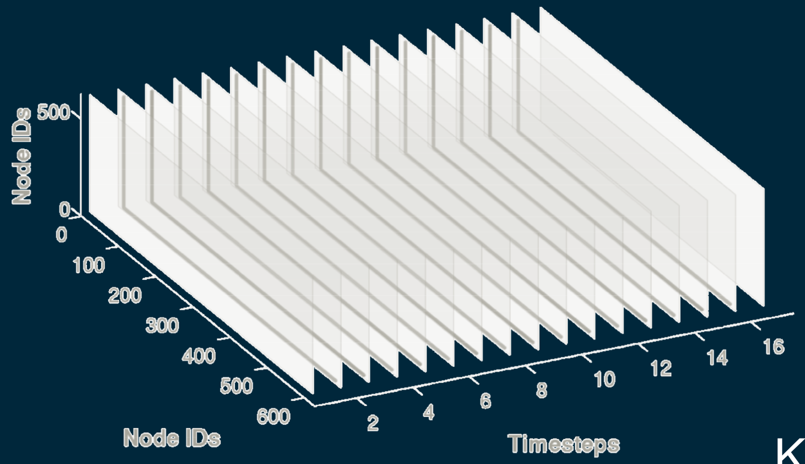
Dataset	Metric	Supervised LP	Supervised AMIE+C [16]	Unsupervised Freq	Unsupervised KGIST
NELL	Recall	N/A	0.6587 ± 0.03	0.4589 ± 0.02	0.7598 ± 0.02
	R_L	N/A	N/A	0.3924 ± 0.02	0.6636 ± 0.01
DBpedia	Recall	N/A	0.8187 ± 0.01	0.8049 ± 0.01	0.9288 ± 0.00
	R_L	N/A	N/A	0.7839 ± 0.01	0.9179 ± 0.00

Browse the Knowledge Base!

DBpedia

KGIST significantly outperforms the baselines.
It complements LP methods.

This talk: Summarization Meets Outlier Detection

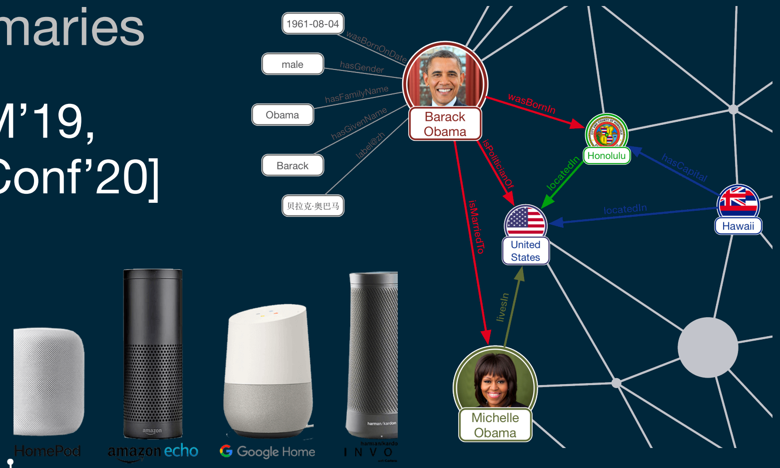


Structural
Summaries

[SDM'14, KDD'15,
Dat Bull Eng'17,
SNAM'18,
SDM'19,
KDD'19a, KDD'20…]

Query-on-the-edge + Rule-based
Summaries

[ICDM'19,
WebConf'20]



② Graph Streams:
Persistent and bursty activity detection
[KDD'20]

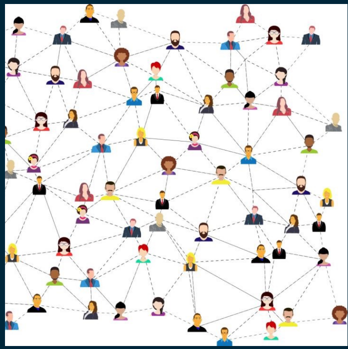
① Knowledge Graphs:
Unified error detection and completion
[WebConf'20]

M CSE
GEMS LAB

Summarizing Evolving Networks

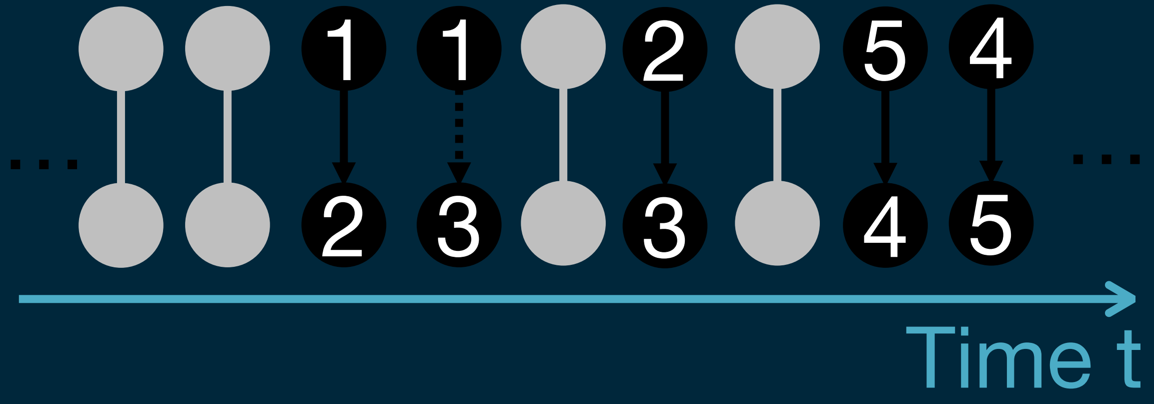
One possibility: summary of frequent graph patterns

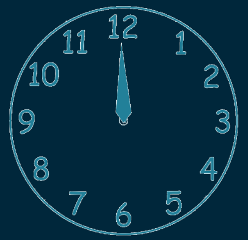
- Related topics:
  - ✧ Motif Mining
    - [Kovanen+, JSTAT'11], [Paranjape+, WSDM'17], [Liu+, WSDM'19]
  - ✧ Frequent Subgraph Mining
    - [Abdelhamid+,TKDE'17], [Aslay+, CIKM'18]

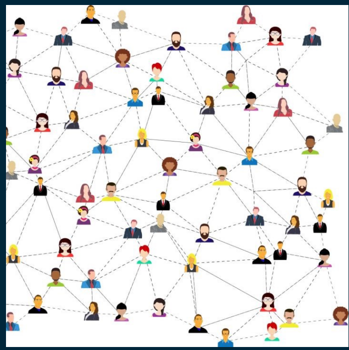


[Caleb Belth, X. Zheng, D. Koutra. ACM KDD'20] https://github.com/GemsLab/PENminer

Summarizing Evolving Networks

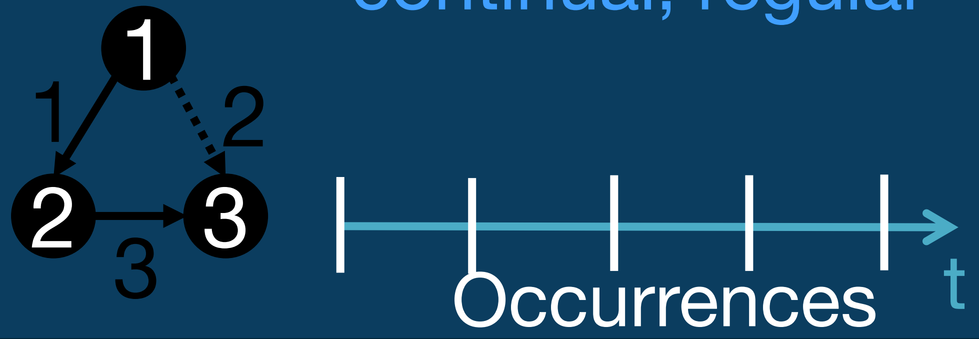






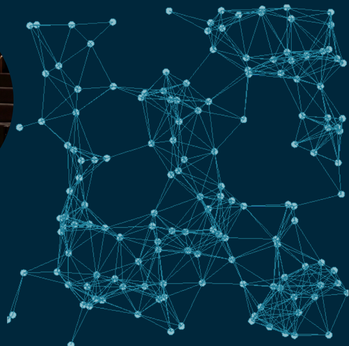
bursty anomaly

continual, regular



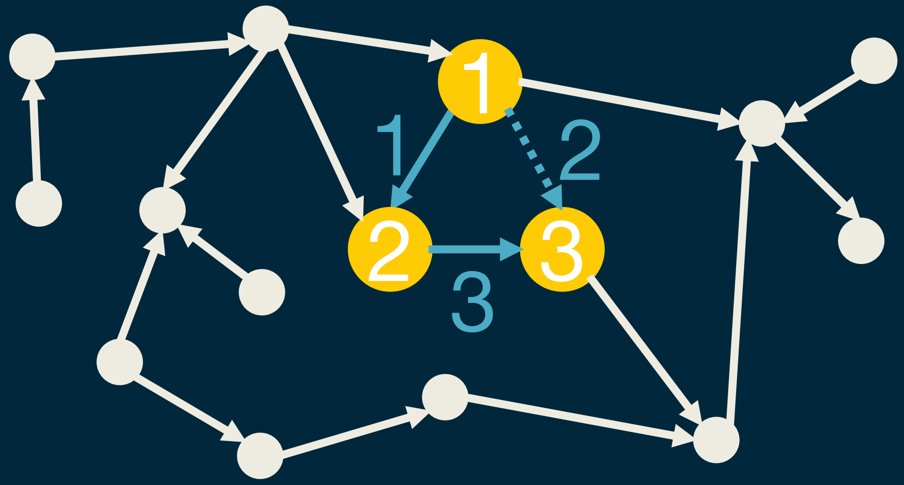
Summarizing Evolving Networks

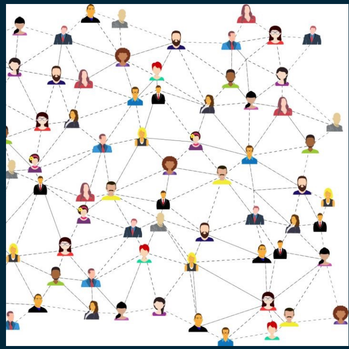




Summarize graph stream *G*,

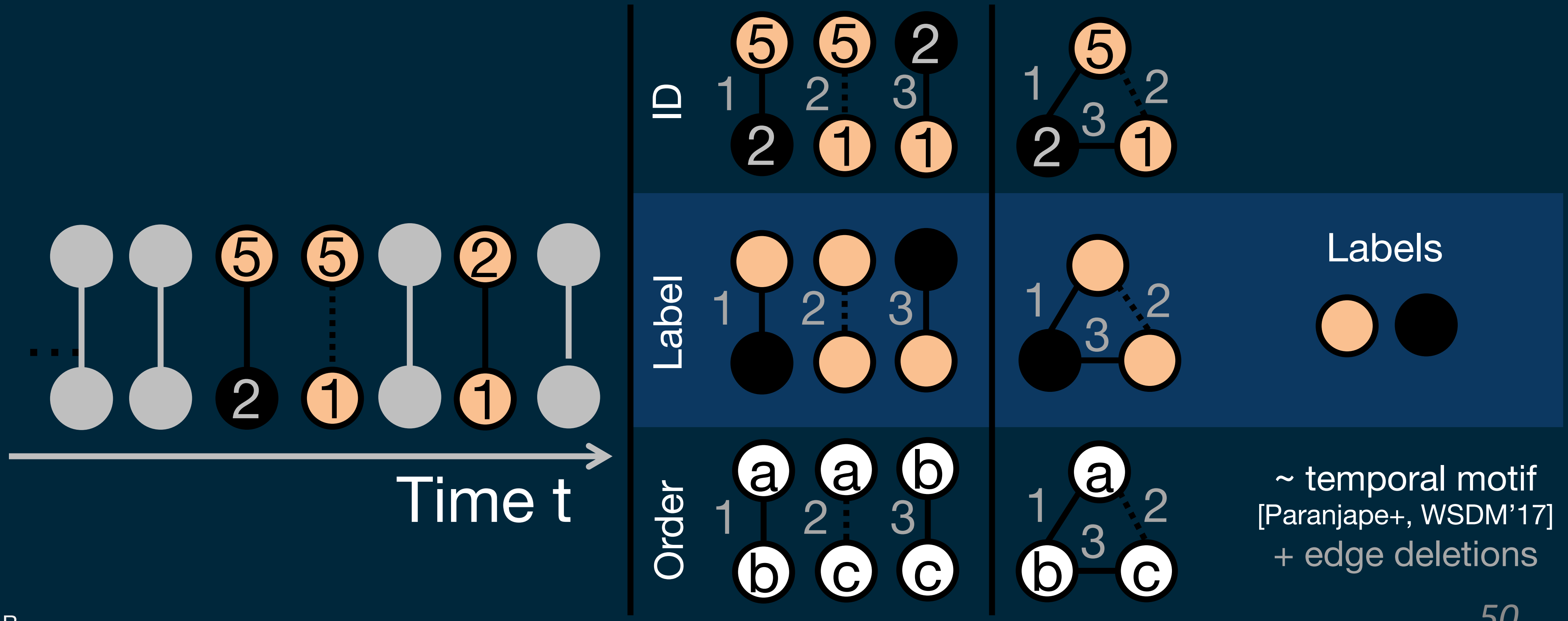
with *persistent* "activity snippets".






Activity Snippet

An activity snippet describes a sequence of activity among connected nodes in a network



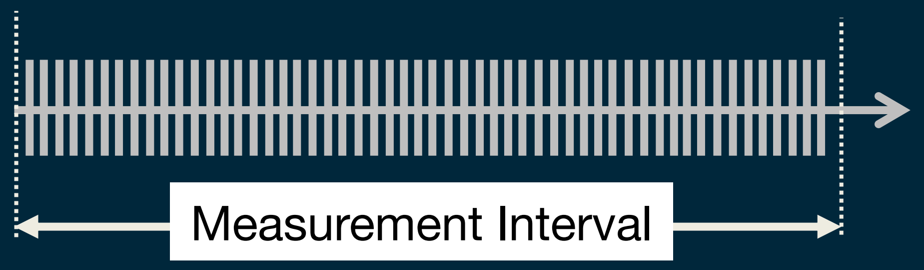
Axioms of Persistence

A1: Persistence should be 0 *iff* there are 0 occurrences

Axioms of Persistence

A1: Persistence should be 0 *iff* there are 0 occurrences

A2: As the interval becomes infinitely filled with unique occurrences, persistence should tend to infinity



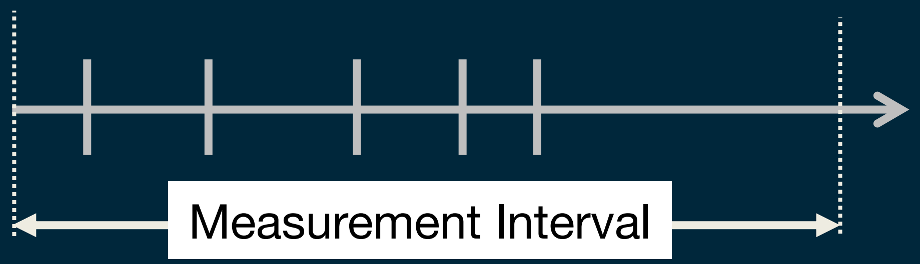
Persistence → ∞

Axioms of Persistence

A1: Persistence should be 0 *iff* there are 0 occurrences

A2: As the interval becomes infinitely filled with unique occurrences, persistence should tend to infinity

A3: Shifting all occurrences should not affect persistence



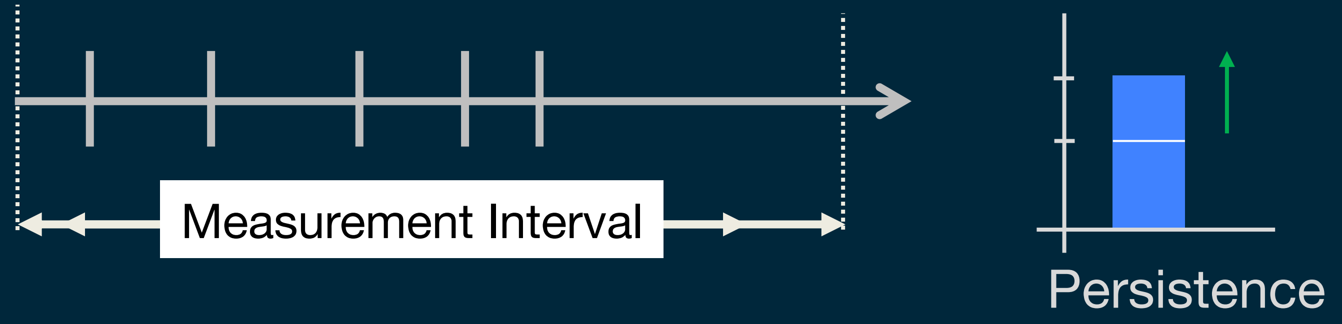
Axioms of Persistence

A1: Persistence should be 0 *iff* there are 0 occurrences

A2: As the interval becomes infinitely filled with unique occurrences, persistence should tend to infinity

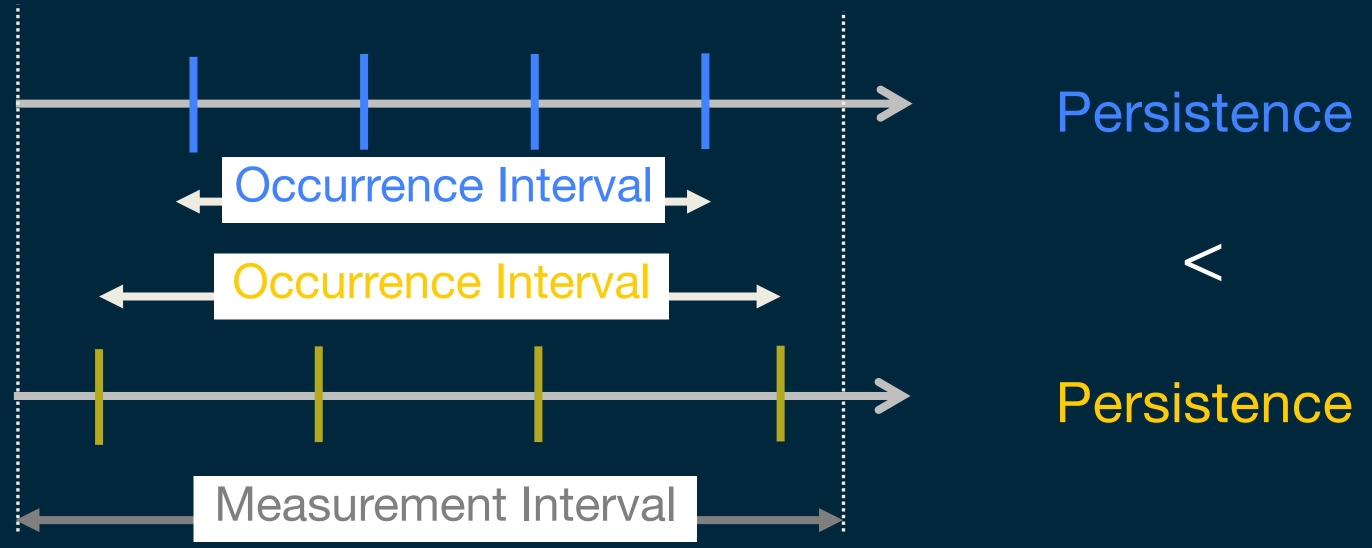
A3: Shifting all occurrences should not affect persistence

A4: Shrinking the interval of measurement leads to higher persistence



Properties of Persistence

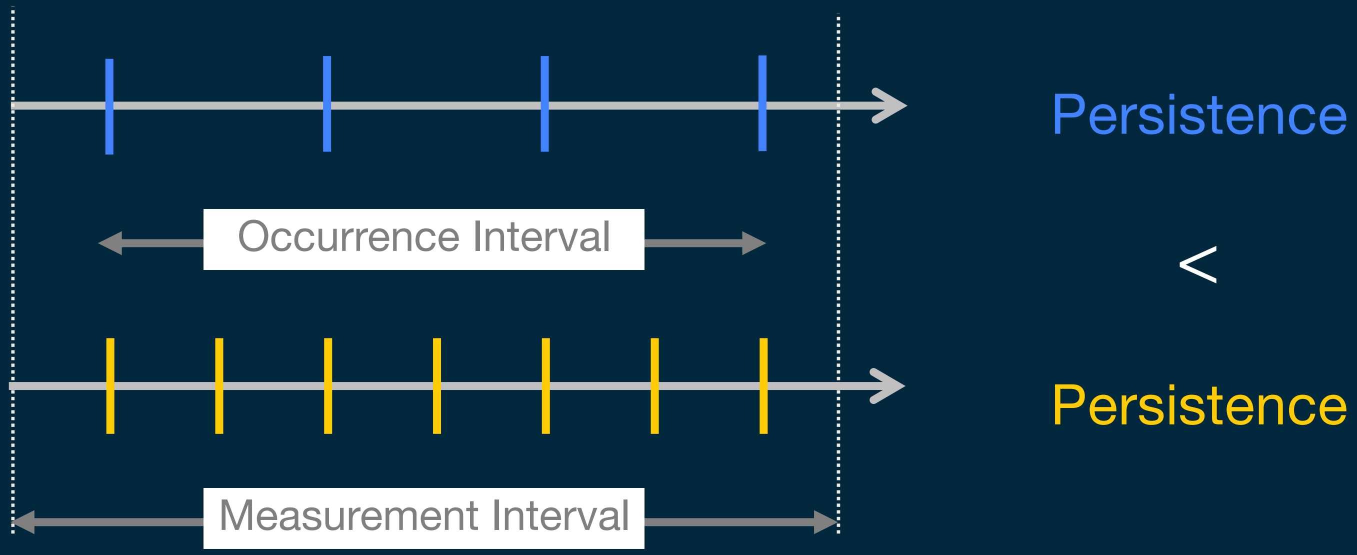
P1: For two snippets with n unique, uniformly-spaced occurrences, persistence is larger for the snippet with occurrences over a wider interval



Properties of Persistence

P1: For two snippets with n unique, uniformly-spaced occurrences, persistence is larger for the snippet with occurrences over a wider interval

P2: For two snippets with unique, uniformly-spaced occurrences spread out over the *same* interval, persistence is larger for the snippet with more occurrences

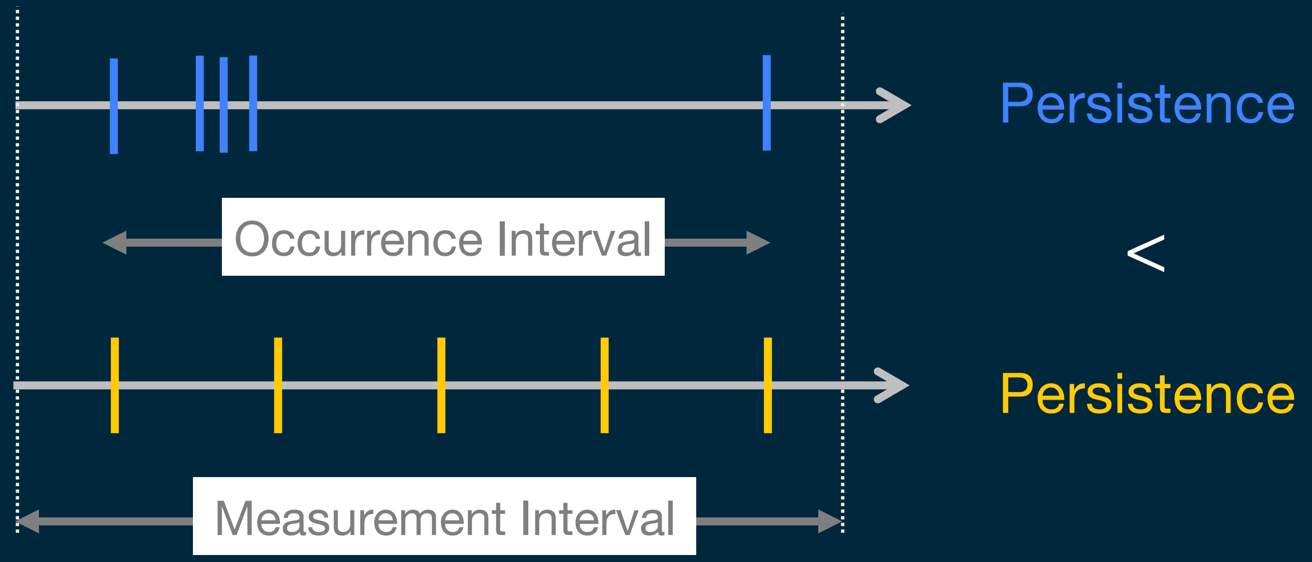


Properties of Persistence

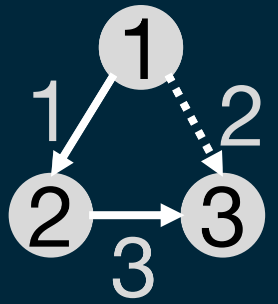
P1: For two snippets with n unique, uniformly-spaced occurrences, persistence is larger for the snippet with occurrences over a wider interval

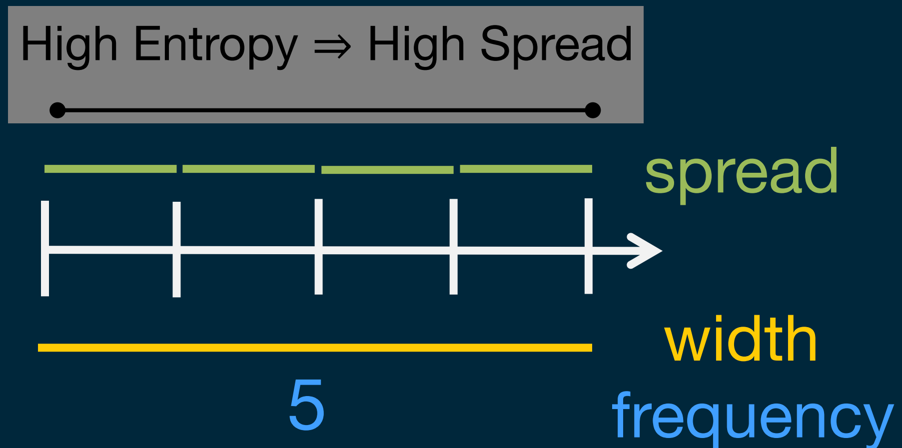
P2: For two snippets with unique, uniformly-spaced occurrences spread out over the *same* interval, persistence is larger for the snippet with more occurrences

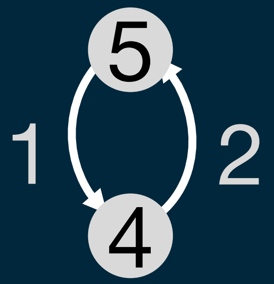
P3: The persistence of a snippet with n unique occurrences in an interval is maximized *iff* the occurrences are spread out uniformly

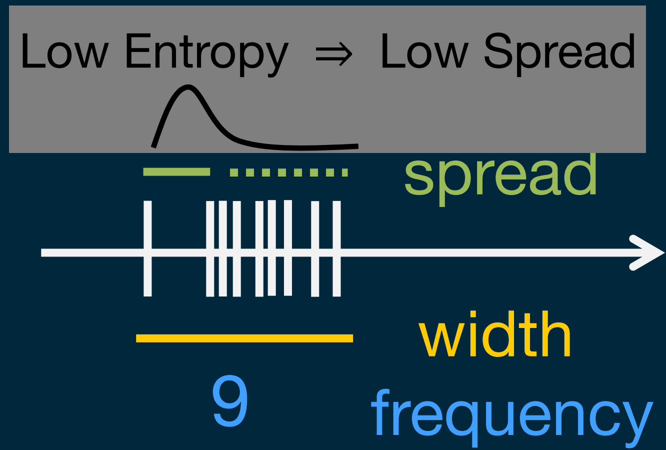


Measuring Snippets' Persistence









$$P(x;[t_s,t_e]) \triangleq W(x;[t_s,t_e])^{\alpha} F(x;[t_s,t_e])^{\beta} S(x;[t_s,t_e])^{\gamma}$$

$$\frac{|[t_f,t_l]_x| + 1}{|[t_s,t_e]| + 1}$$

$$\log_{10}\left(|O_x| + 1\right)$$

$$\begin{cases} \frac{H(\Gamma_x)}{\log|\Gamma_x|} + 1, & |\Gamma_x| > 1 \\ 1, & |\Gamma_x| \in \{0,1\} \end{cases}$$

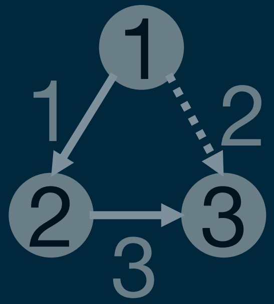
$$H(\Gamma_x) \triangleq -\sum_{g_i \in \Gamma_x} \frac{g_i}{|[t_f,t_l]_x|} \log \frac{g_i}{|[t_f,t_l]_x|}$$

Entropy of distribution of gaps between occurrences

[Caleb Belth, X. Zheng, D. Koutra. ACM KDD'20] https://github.com/GemsLab/PENminer

Measuring Snippets’ Persistence

High Entropy ⇒ High Spread



$$P(x;[t_s,t_e]) \triangleq W(x;[t_s,t_e])^{\alpha} F(x;[t_s,t_e])^{\beta} S(x;[t_s,t_e])^{\gamma}$$

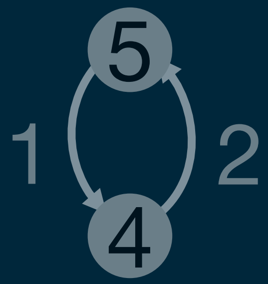
$$\log_{10}\left(|O_x|+1\right)$$

$$\begin{cases} \frac{H(\Gamma_x)}{\log|\Gamma_x|}+1, & |\Gamma_x|>1 \\ 1, & |\Gamma_x|\in\{0,1\} \end{cases}$$

Our Solution:
PENminer

Offline + Streaming variants

* Measure of persistence can apply to any data stream.

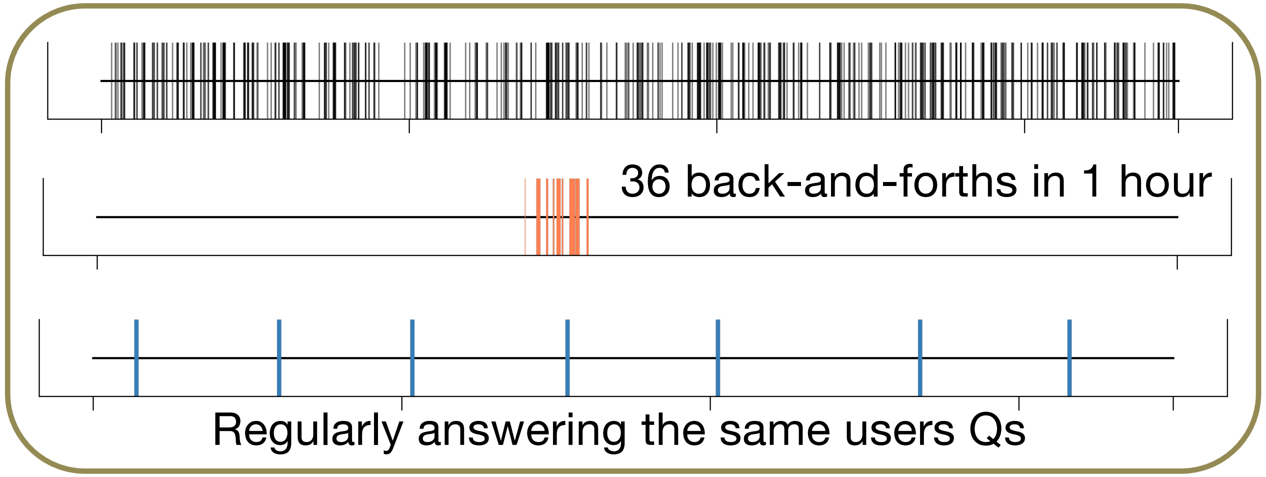


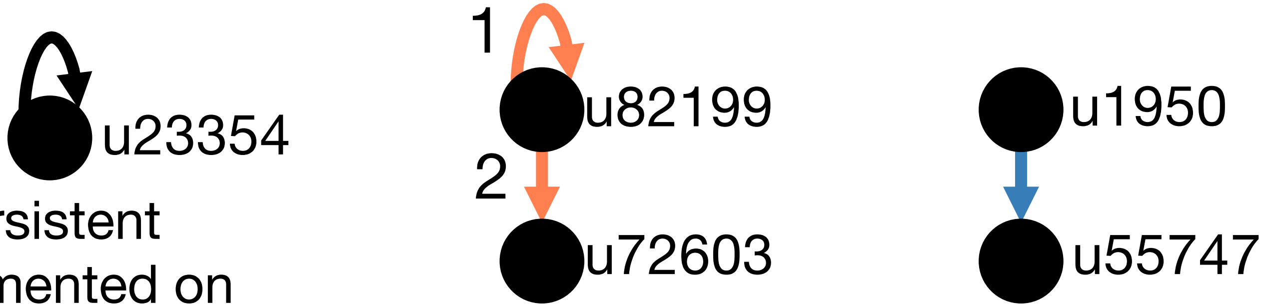
width
9
frequency

Entropy of distribution of gaps between occurrences

Engaged Discussions & Regular Interactions

1 Findings

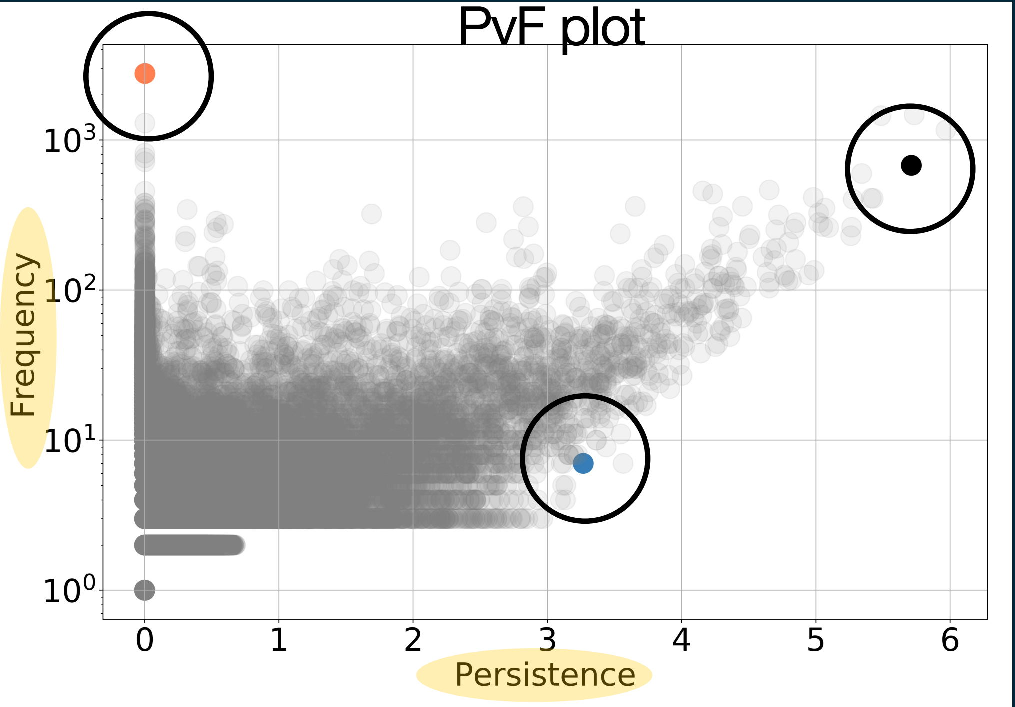




Persistent (Commented on Answer)

Bursty (Commented on Answer)

Subtly Persistent (Answered)



stackoverflow

Real-time Anomaly Detection

[Eswaran+, ICDM'18] [Dai+, PVLDB'14]
[Bhatia+, AAAI'20]

	Metric	Freq	SedanSpot	Midas-R	DS	sPENminer
Subtle	AUC					
	F1@100					
	F1@1K					
	F1@2K					
Bursty	AUC					
	F1@500K					
	F1@1M					
	F1@2M					
	Avg AUC					

subtly anomalous bike trips in Chicago IL

bursty network attacks in DARPA IP network

Subtle anomalies like these

Bursty anomalies like these

Setup:
- Represent each activity snippet at time t with a 2d point <frequency, persistence>
- Apply a streaming anomaly detection method (e.g., Random Cut Forest or RCF [Guha+, ICML'16])

Real-time Anomaly Detection

[Eswaran+, ICDM'18] [Dai+, PVLDB'14] [Bhatia+, AAAI'20]

	Metric	FREQ	SEDANSPOT	MIDAS-R	DS	sPENminer
Subtle	AUC	0.8325±0.02	0.4519±0.01	0.4520±0.02	0.7435±0.03	**0.9309**±0.00*
	F1@100	0.0505±0.01	0.0001±0.00	0.0000±0.00	0.0076±0.00	**0.0508**±0.01
	F1@1K	0.1812±0.00	0.0035±0.00	0.0003±0.00	0.0378±0.01	**0.2580**±0.03*
	F1@2K	0.1572±0.01	0.0098±0.00	0.0002±0.00	0.0561±0.01	**0.3292**±0.03*
Bursty	AUC	0.8450±0.00	0.6390±0.00	**0.9434**±0.00*	0.8632±0.00	0.8359±0.01
	F1@500K	**0.3089**±0.00*	0.2745±0.00	0.3019±0.00	0.3063±0.00	0.2978±0.00
	F1@1M	**0.5351**±0.00*	0.4527±0.00	0.5274±0.00	0.5295±0.00	0.5169±0.00
	F1@2M	0.7184±0.00	0.6309±0.00	**0.8378**±0.00*	0.8066±0.00	0.7770±0.01
	Avg AUC	0.8388 (2)	0.5455 (5)	0.6977 (4)	0.8034 (3)	**0.8834** (1)

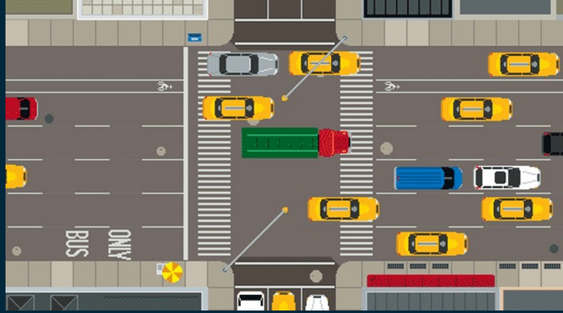
Subtle anomalies like these

Bursty anomalies like these

PENminer outperforms all baselines at the new task of finding subtle anomalies, and performs competitively at finding bursty anomalies against baselines designed specifically for that task.

3 Findings

Surprisingly Regular Taxi Trips



NYC Taxi data

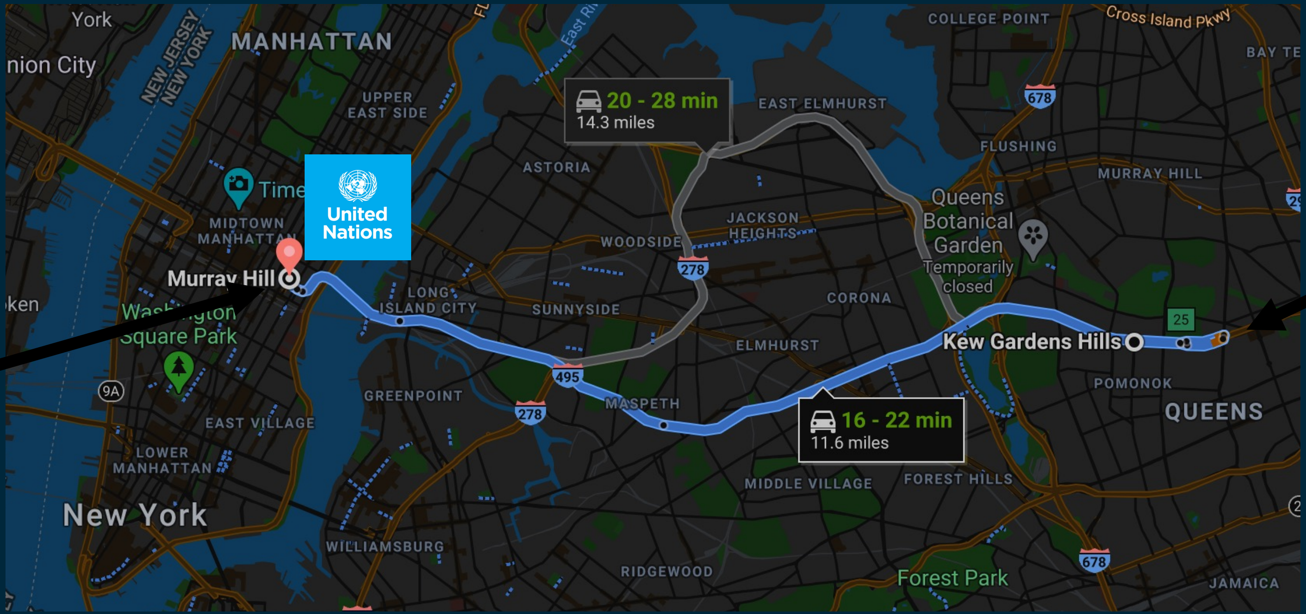
Mysterious trip every day

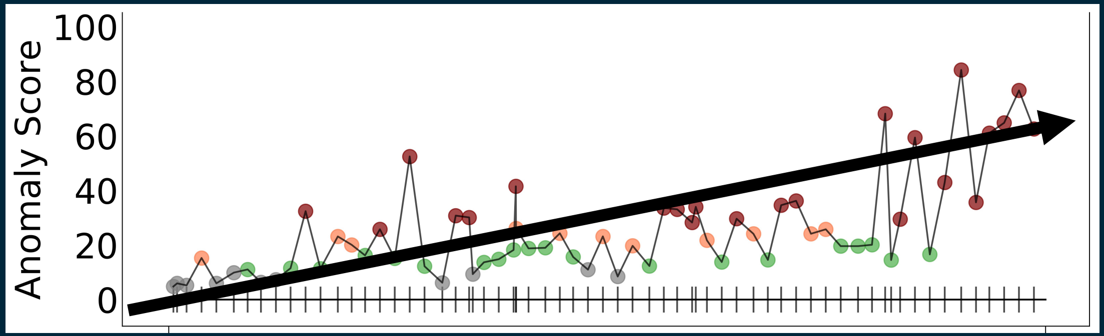
From Queens

To near UN building

Around midnight

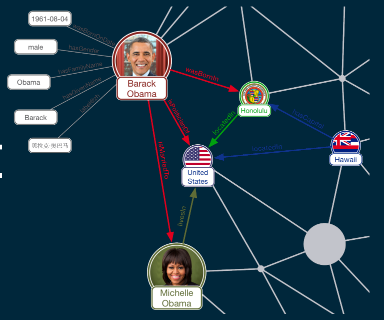
For over two months





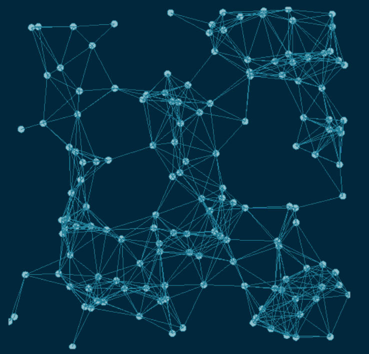
Recap: Graph Summarization Meets Outlier Detection

- Summarization can help identify *patterns* and anomalies in the data



- Rule-based summarization of KGs can help unify multiple refinement tasks that are traditionally solved by tailored approaches [WWW'20]
  - ✧ KGist can identify various types of errors in KGs and missing information

- Summarization of graph streams with persistent activity snippets [KDD'20]
  - ✧ Beyond just frequency; capture *how* patterns evolve
  - ✧ The relationship of frequency and persistence highlight anomalies



Talk based on the following papers

- Y. Liu, T. Safavi, A. Dighe, D. Koutra. Graph Summarization Methods and Applications: A Survey. ACM Computing Surveys 2018.

- Caleb Belth, Xinyi (Carol) Zheng, Jilles Vreeken, Danai Koutra. What is normal, What is Strange, and What is Missing in a Knowledge Graph: Unified Characterization via Inductive Summarization. The Web Conference (WWW/TWC) '20.

- Caleb Belth, Xinyi (Carol) Zheng, Danai Koutra. Mining Persistent Activity in Continually Evolving Networks. ACM SIGKDD ’20.

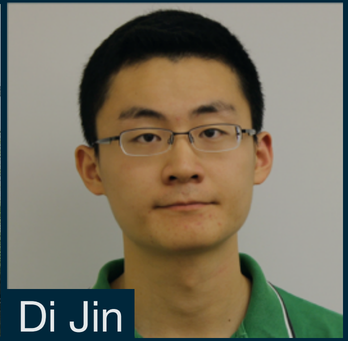
Thank you!
Questions?

Danai Koutra
dkoutra@umich.edu
GEMS LAB


Caleb Belth


Marlena Duda

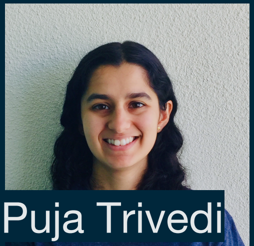

Mark Heimann


Di Jin


Tara Safavi

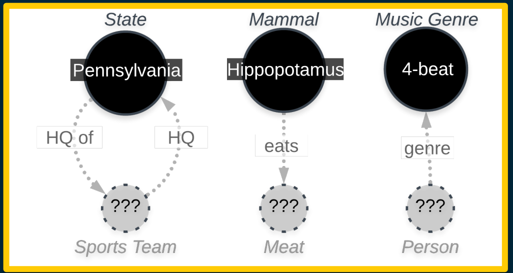

Yujun Yan

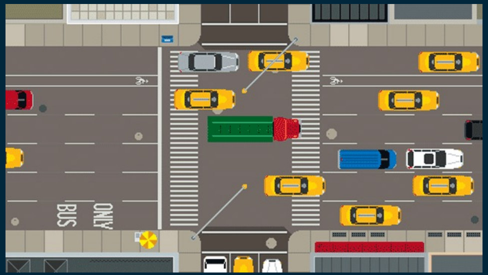

Jing Zhu

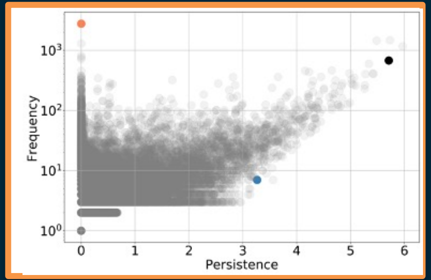

Puja Trivedi


Jiong Zhu

Graph Summarization Meets Outlier Detection







</> https://github.com/GemsLab/KGIST
https://github.com/GemsLab/PENminer

NSF ARL P&G MIDAS amazon Adobe Microsoft Azure