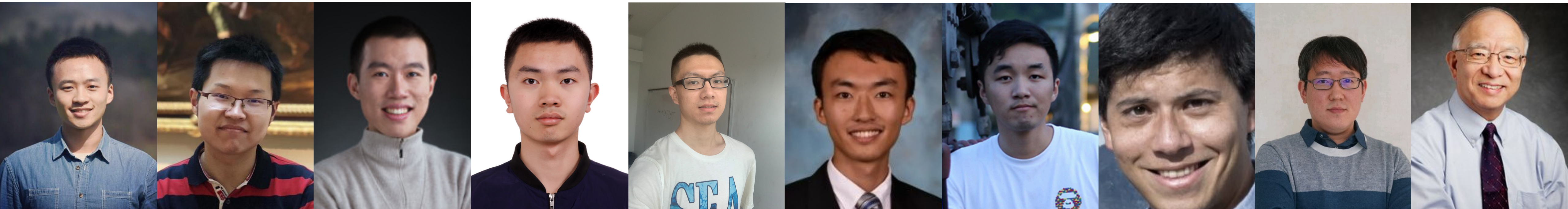
Discovering Hypernymy in Text-Rich Heterogeneous Information Network by Exploiting Context Granularity

Yu Shi[†*], Jiaming Shen[†*], Yuchen Li[†], Naijing Zhang[†], Xinwei He[†], Zhengzhi Lou[†], Qi Zhu[†], Matthew Walker[‡], Myunghwan Kim[§], Jiawei Han[†]

University of Illinois at Urbana-Champaign (UIUC)

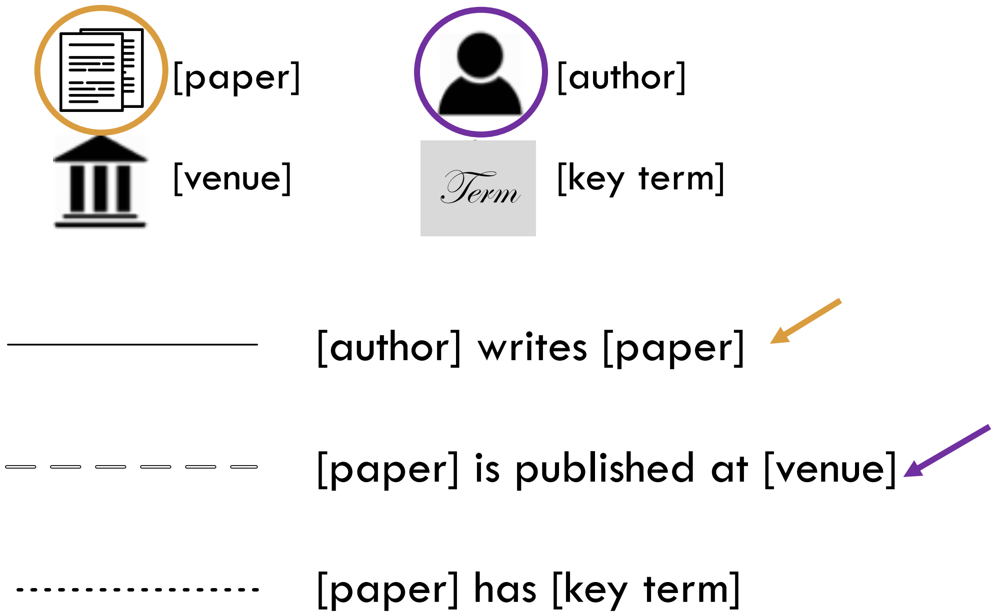
[‡]LinkedIn Corporation [§]Mesh Korea

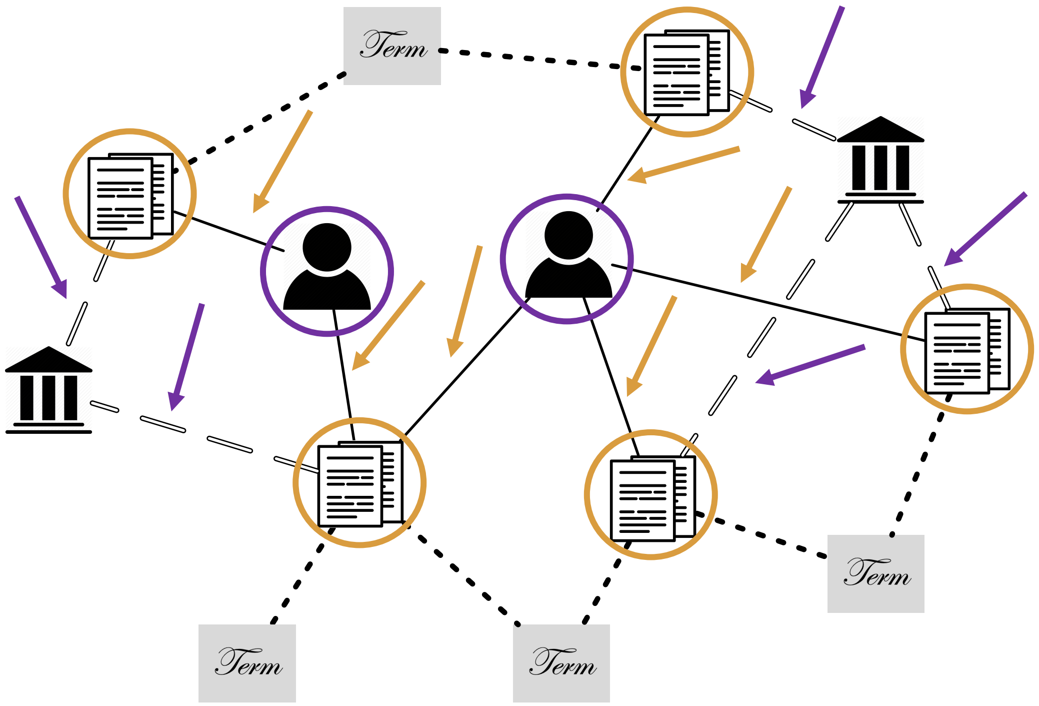
[*]These authors contributed equally to this work.



In real world applications, objects of different types can have different relations, which form heterogeneous information networks (**HINs**).

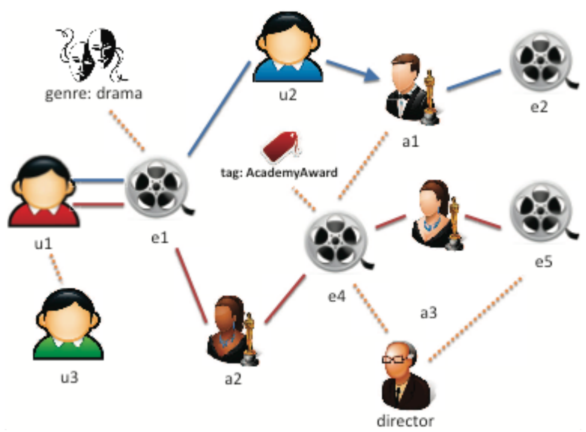
- **Typed** nodes: objects.
- **Typed** edges: relations.



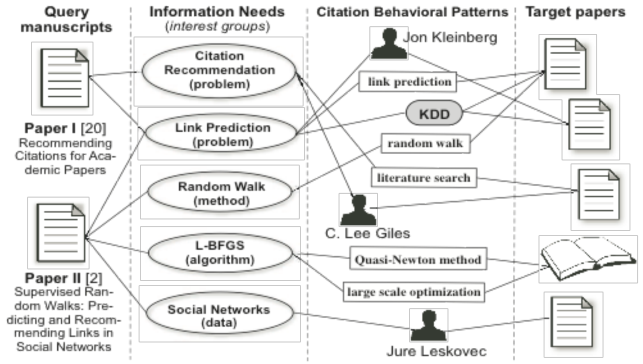


A toy bibliographic network

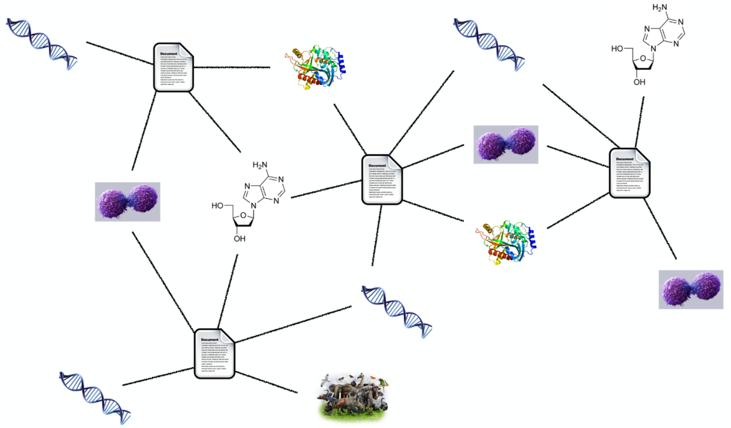
Heterogeneous information networks (HINs) are ubiquitous.



Movie Reviewing Network

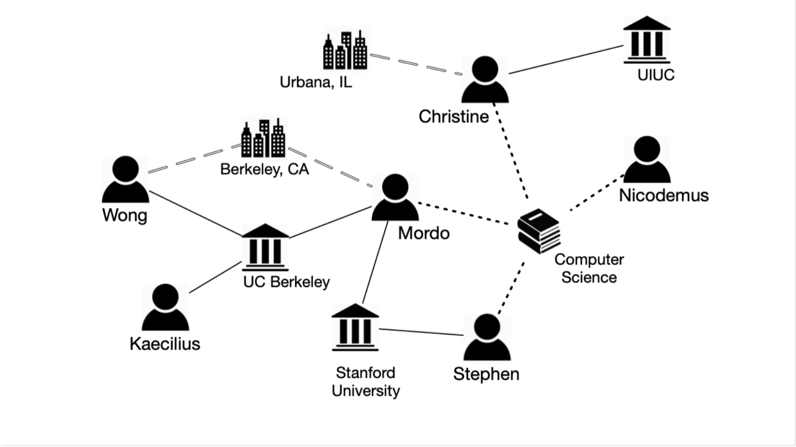


Bibliographic Network

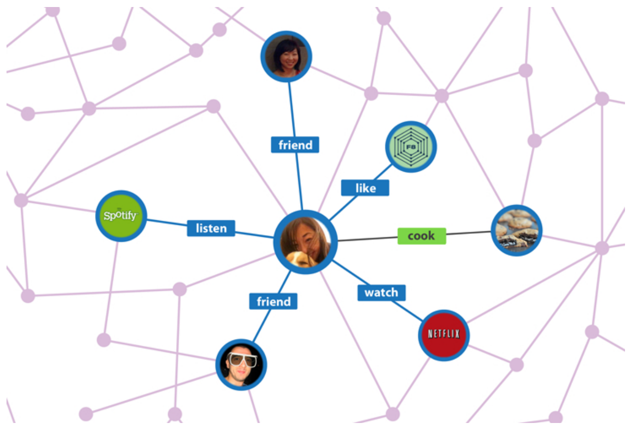


Biomedical Network

Economic Graph



Social Network



Facebook Open Graph

Figures partly from http://xren7.web.engr.illinois.edu/yahoo-dais_award.html, http://www.garrygolden.com/2015/07/06/linkedin-economic-graph-technosolutionism, and http://www.businessinsider.com/explainer-what-exactly-is-the-social-graph-2012-3

Background – Hypernymy Discovery

- We tackle the problem of hypernymy discovery – an important and fundamental problem in natural language processing.

> A **hypernym** is a word whose semantic field includes that of another word – its **hyponym**. Hypernymy or hyponymy is used to refer to such hyponym-hypernym relation.

- E.g., everything about a hyponym must be also about its hypernym
- Examples:

[hyponym] $\rightarrow$ **[hypernym]**
bird $\rightarrow$ animal
machine learning $\rightarrow$ computer science

machine learning $\nrightarrow$ data mining
computer science $\nrightarrow$ IT industry

Background – Hypernymy Discovery

- We tackle the problem of hypernymy discovery – an important and fundamental problem in natural language processing.
- Applications:

Knowledge base

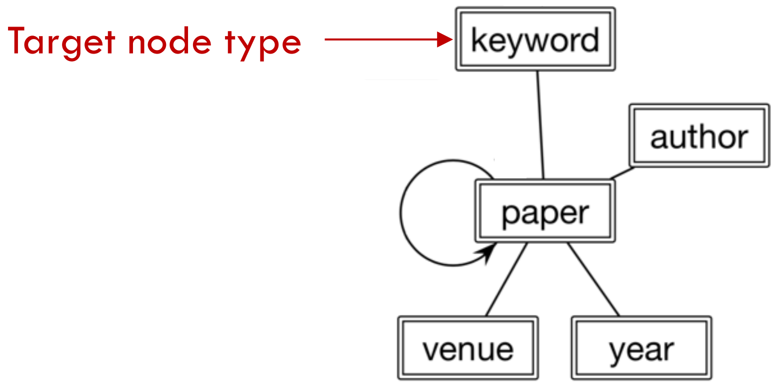
Taxonomy

Background – Hypernymy Discovery

- Expected output:

Hypernymy pairs with likelihood
…
[literature mining → data mining]: 0.99
…
[graph mining → data mining]: 0.98
…
…
…
[data mining → graph mining]: 0.05
…
…
[literature mining → python]: 0.02
…
…

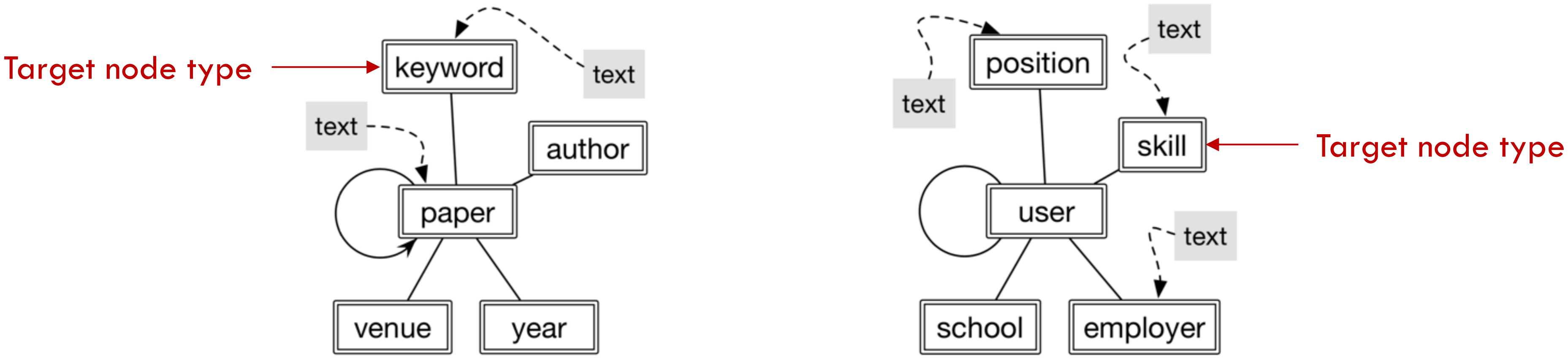
- Existing methods mainly discovery hypernymy from text corpora.
- Why from heterogeneous information networks?
  - Structured (in comparison with pure text)
  - Meaningful node type for hypernymy discovery
  - Rich semantics



(a) DBLP.

Text-rich heterogeneous information networks

- Existing methods mainly discovery hypernymy from text corpora.
- Why from heterogeneous information networks?
  - Structured (in comparison with pure text)
  - Meaningful node type for hypernymy discovery
  - Rich semantics



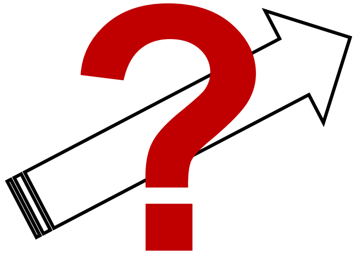
(a) DBLP. (b) LinkedIn.

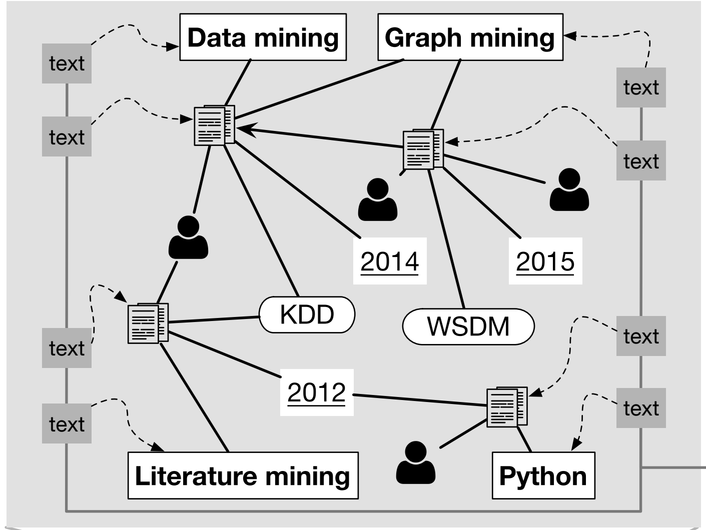
Text-rich heterogeneous information networks

- Existing methods mainly discovery hypernymy from text corpora.
- Why from heterogeneous information networks?
  - Structured (in comparison with pure text)
  - Meaningful node type for hypernymy discovery
  - Rich semantics

Hypernymy pairs with likelihood

…
[literature mining → data mining]: 0.99
…
[graph mining → data mining]: 0.98
…
…
…
[data mining → graph mining]: 0.05
…
…
[literature mining → python]: 0.02
…
…





Corpus

…
…
…
Literature mining is a specialised **data mining** method that is used to extract information (facts or data) from text…
…
…
Python is an interpreted high-level **programming language**…
…
…
…

Two families of methods for hypernymy discovery from text

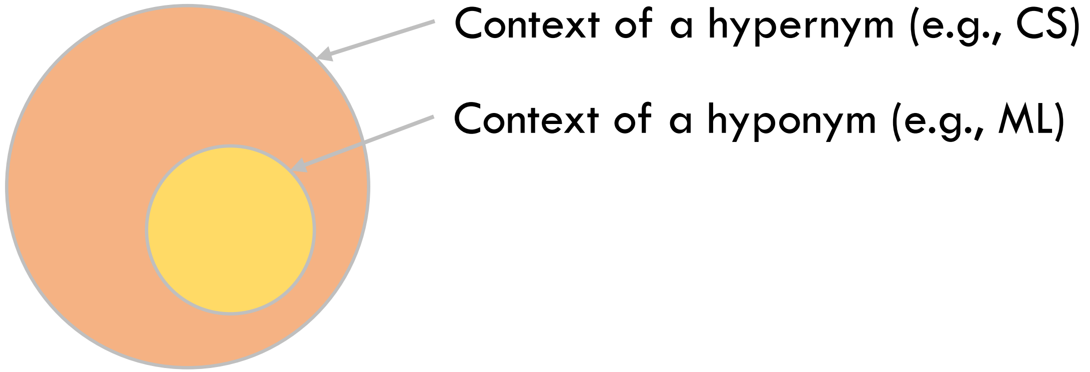
- **Textual-pattern–based**: Hearst pattern, etc.
  - High precision, low recall

Pattern Name	Pattern structure	Example
HEARST 1	CONCEPT such as (INSTANCE)+ ((and | or) INSTANCE)?	Cities such as Barcelona or Madrid
HEARST 2	CONCEPT (,?) especially (INSTANCE)+ ((and | or) INSTANCE)?	Countries especially Spain and France
HEARST 3	CONCEPT (,?) including (INSTANCE)+ ((and | or) INSTANCE)?	Capitals including London and Paris
HEARST 4	INSTANCE (,?)+ and other CONCEPT	Eiffel Tower and other monuments
HEARST 5	INSTANCE (,?)+ or other CONCEPT	Coliseum or other historical places

Table from https://www.researchgate.net/publication/220774728_Unsupervised_Web-based_Automatic_Annotation

Two families of methods for hypernymy discovery from text

- **Textual-pattern–based:** Hearst pattern, etc.
- **Distributional Inclusion Hypothesis (DIH):**
  - Find context distribution for each word based on co-occurrence
  - Given a dataset, **DIH** assumes the context of a **hypernym** (e.g., CS) should subsume of the context of a **hyponym** (e.g., ML)



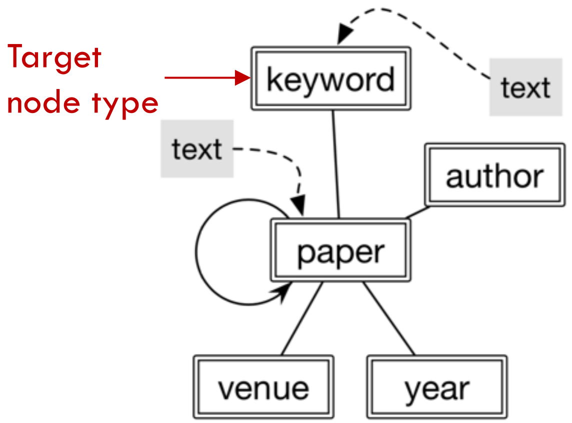
One DIH-based hypernymy measure: $M_1(t_1 \rightarrow t_2) = \frac{\sum_{c \in C^{(t_1)} \cap C^{(t_2)}} r_c(t_1)}{\sum_{C^{(t_1)}} r_c(t_1)}$

Two families of methods for hypernymy discovery from text

- **Textual-pattern–based**
- **Distributional Inclusion Hypothesis (DIH)**

When the input is a network instead of text, DIH-based method can still apply.

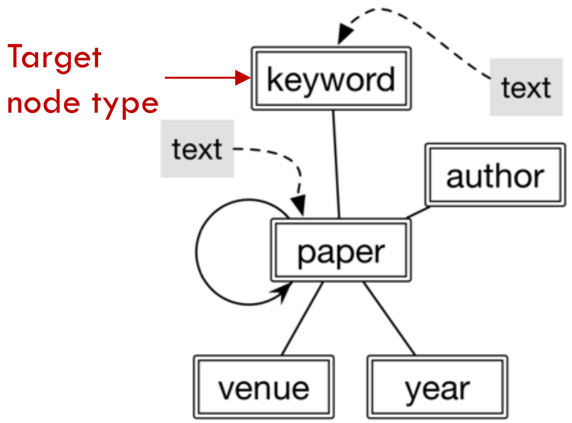
Contexts in HINs



(a) DBLP.

- How to define the **context** of a target node?
- The most straightforward way: neighbors
  - For DBLP, each unit of the context is a paper.
  - DIH ⇔ all **papers** tagged to the **hyponym** keyword (e.g., ML) should also be tagged to the **hypernym** keyword (e.g., CS).

Contexts in HINs

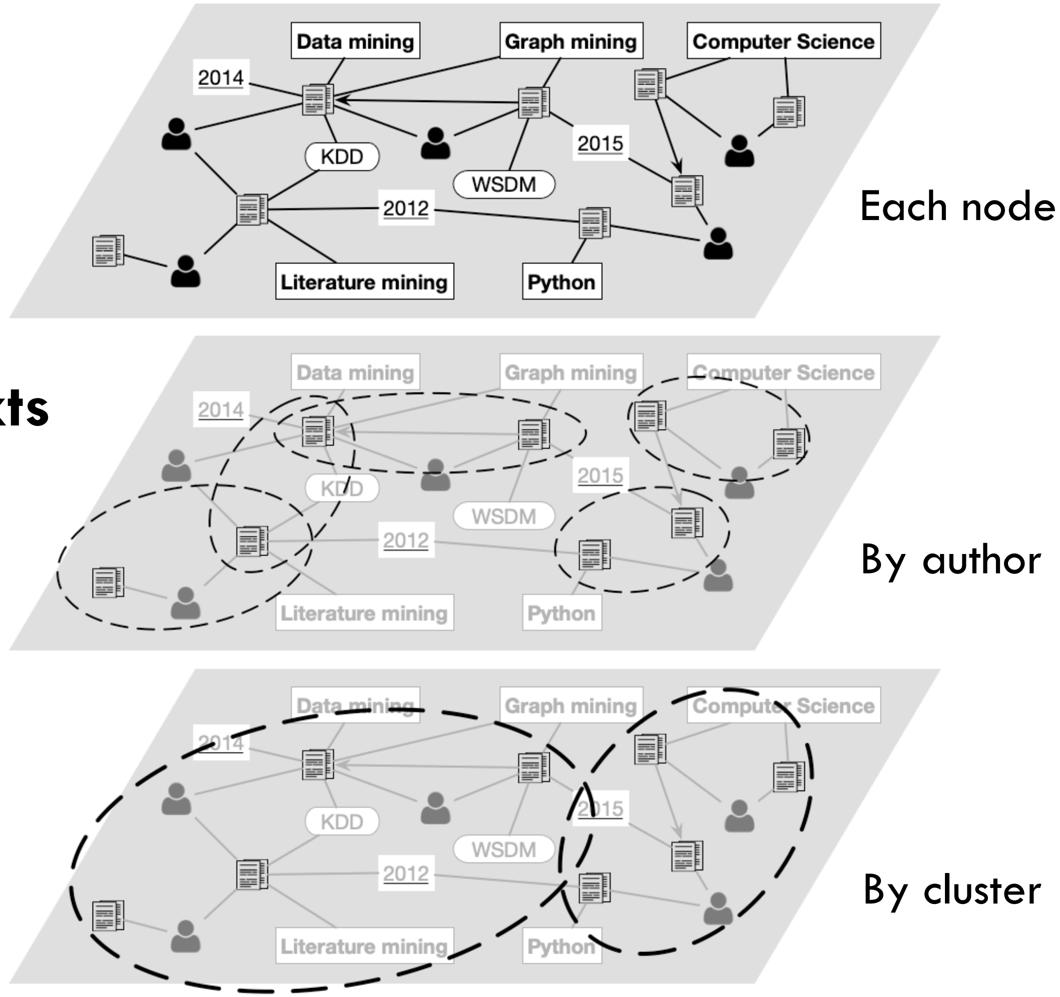


(a) DBLP.

- How to define the **context** of a target node?
- The most straightforward way: neighbors
  - For DBLP, each unit of the context is a paper.
  - DIH ⇔ all **papers** tagged to the **hyponym** keyword (e.g., ML) should also be tagged to the **hypernym** keyword (e.g., CS).
- With the rich type and rich semantics, one can easily define the context in other ways
  - Each context unit being **an author of the papers** tagged to the keyword (equiv. to grouping papers of the same author together)
  - Each context unit being a **cluster** of nodes

As such, an HIN can have **different meaningful contexts**

- with context units at different granularity.



In this view of the typed network, we have

- **nodes** of target types and
- different **contexts**

Computer Science

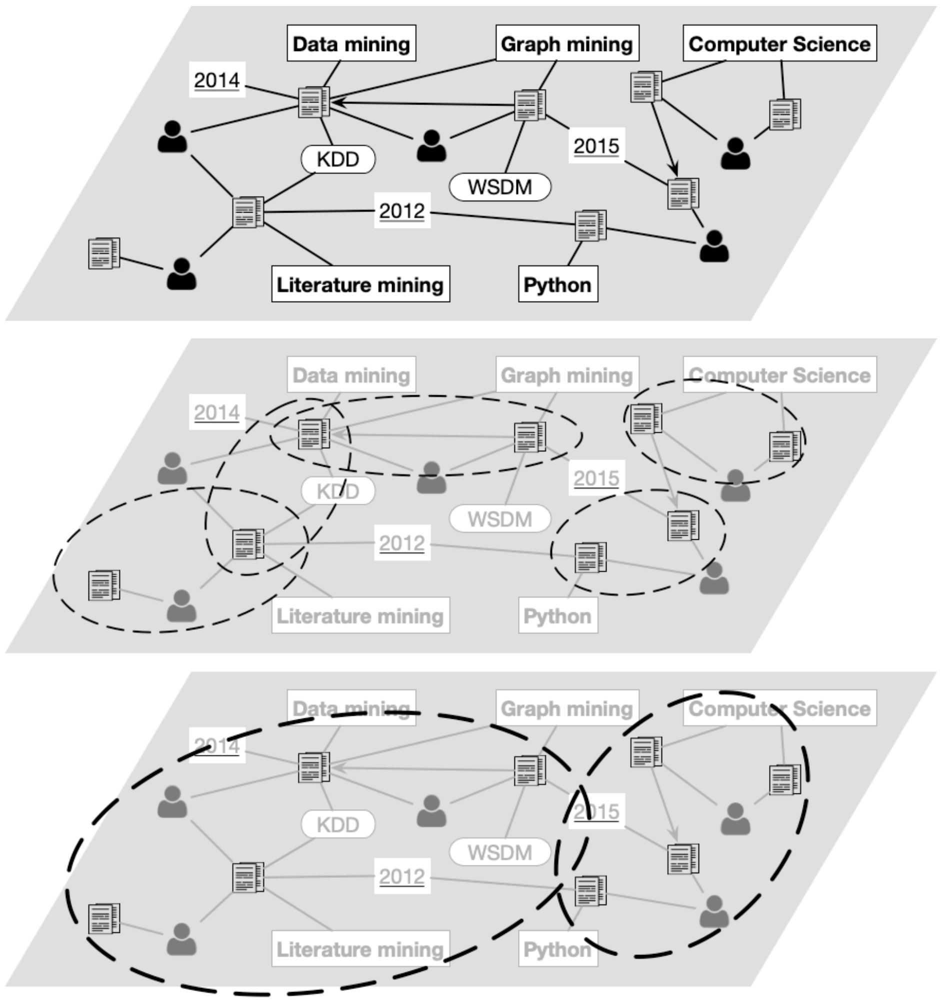
Data mining

Graph mining

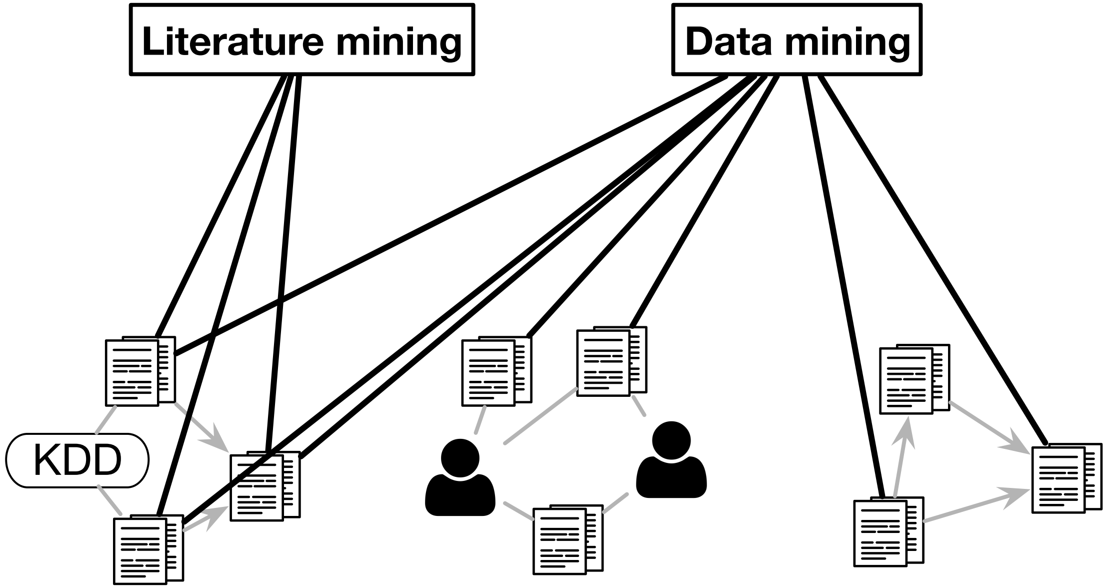
...

Literature mining

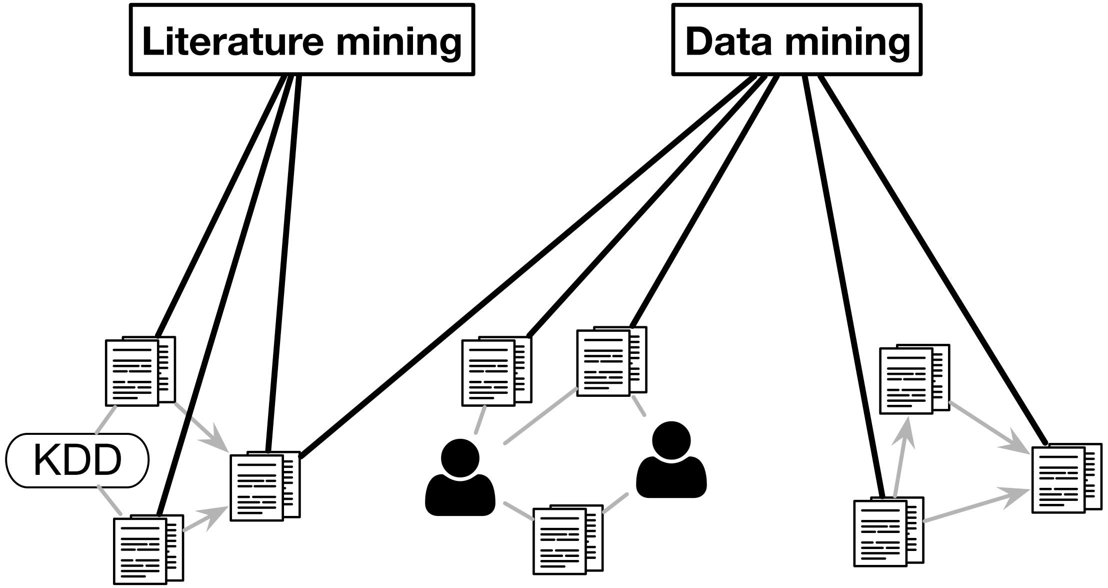
Python



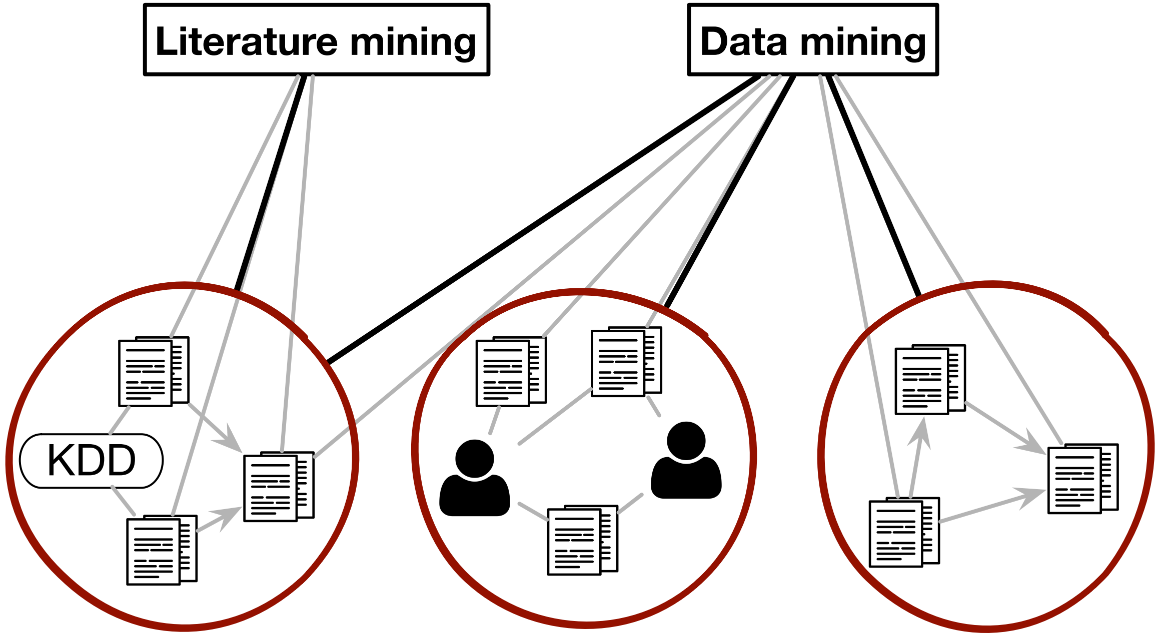
- Are the nodes and the contexts always compatible with each other?



- Ideally, all papers tagged to *literature mining* should also be tagged to *data mining*.



- Ideally, all papers tagged to *literature mining* should also be tagged to *data mining*.
- However, papers may **not always tag** the higher-level keyword data mining if they **already tagged** literature mining.



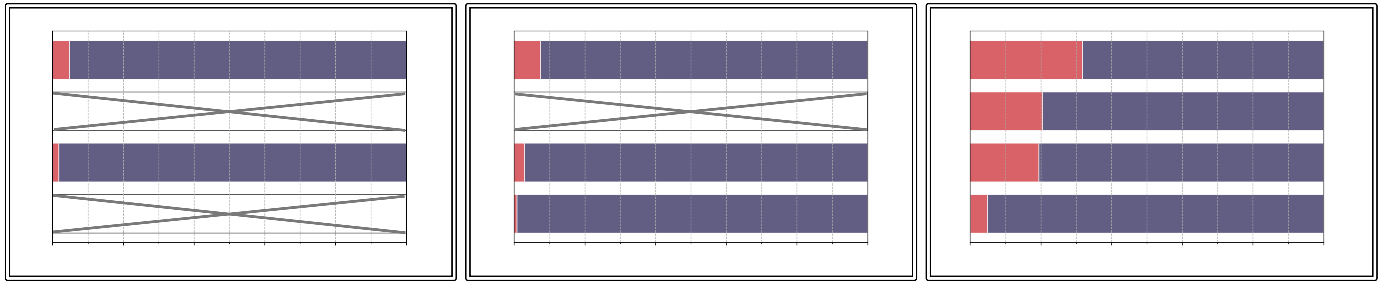
- Ideally, all papers tagged to *literature mining* should also be tagged to *data mining*.
- However, papers may **not always tag** the higher-level keyword data mining if they **already tagged** literature mining.
- That is, the simplest definition of context is not compatible with all hypernymy pairs.
- DIH still holds if we could cluster properly and define context at a **coarser granularity**.

Is there **one** context that is compatible with all hypernymy pairs?

- Based on the quantity computed from one popular DIH measure.
- Hypernymy relation can be discovered if the **red part** is much **shorter** can the **blue part**.

t_1 = **Association rule mining**	t_2 = **Data mining**
t_1 = **Frequent pattern mining**	t_2 = **Data mining**
t_1 = **Graph mining**	t_2 = **Data mining**
t_1 = **Literature mining**	t_2 = **Data mining**

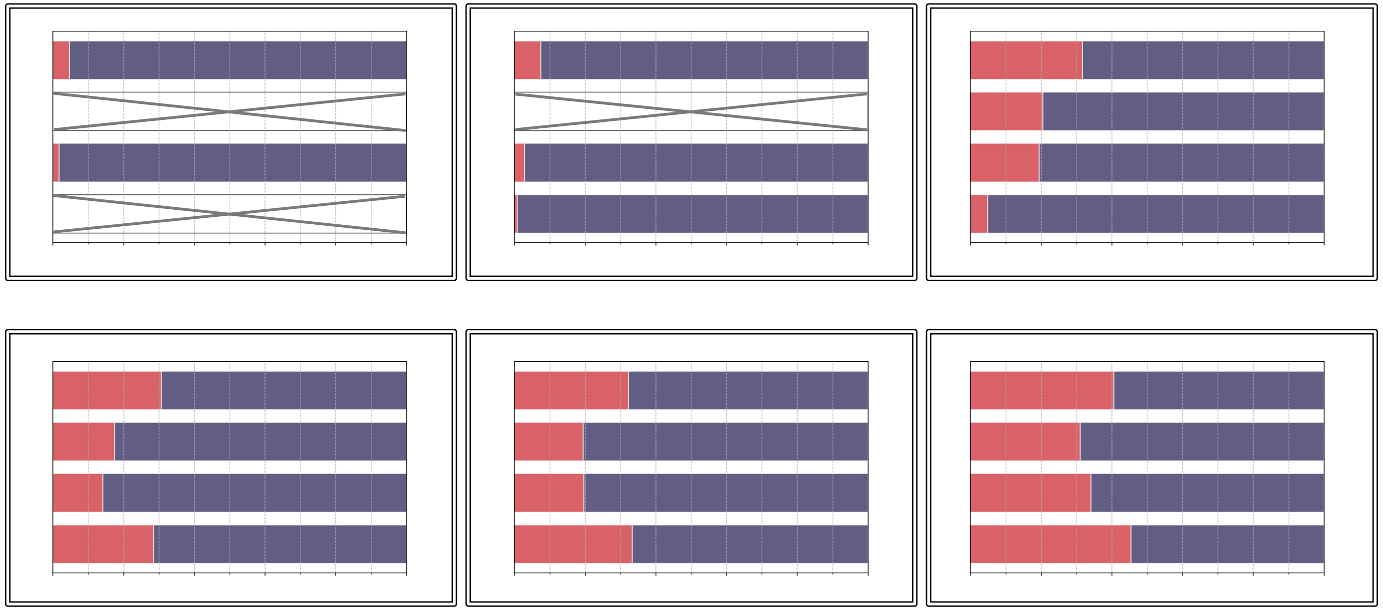
Simplest **Grp-by-W** **Clus-100**



Is there **one** context that is compatible with all hypernymy pairs?

- Based on the quantity computed from one popular DIH measure
- Hypernymy relation can be discovered if the red part is much **shorter** can the blue part

		Simplest	Grp-by-W	Clus-100
t_1 = Association rule mining	t_2 = Data mining			
t_1 = Frequent pattern mining	t_2 = Data mining			
t_1 = Graph mining	t_2 = Data mining			
t_1 = Literature mining	t_2 = Data mining			
t_1 = Supervised learning	t_2 = Learning algorithm			
t_1 = Unsupervised learning	t_2 = Learning algorithm			
t_1 = Semi-supervised learning	t_2 = Learning algorithm			
t_1 = Reinforcement learning	t_2 = Learning algorithm			

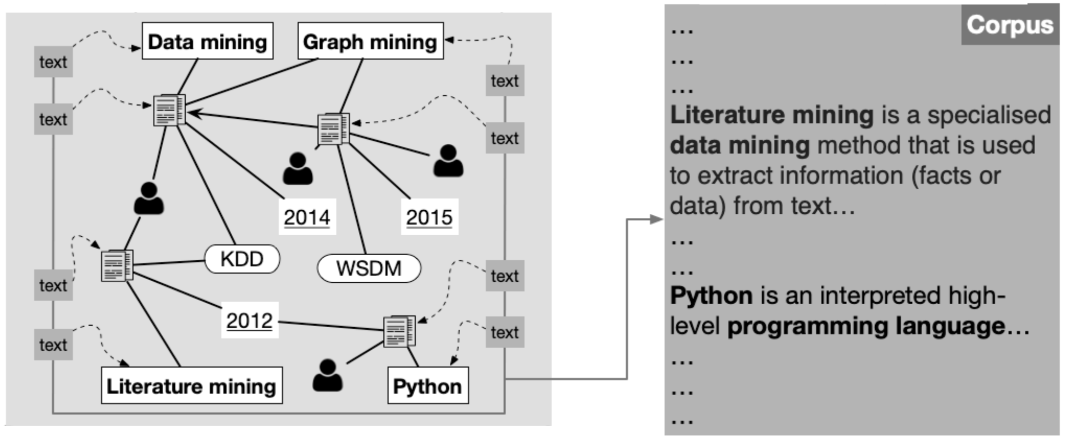


Different **hypernymy pairs** can have different compatibility with **different context.**

Based on the intuition that hypernymy discovery from typed networks should be done at multiple contexts with different granularities

- we propose the HDCG framework (**h**ypernymy **d**iscovery from multiple **c**ontext **g**ranularities).

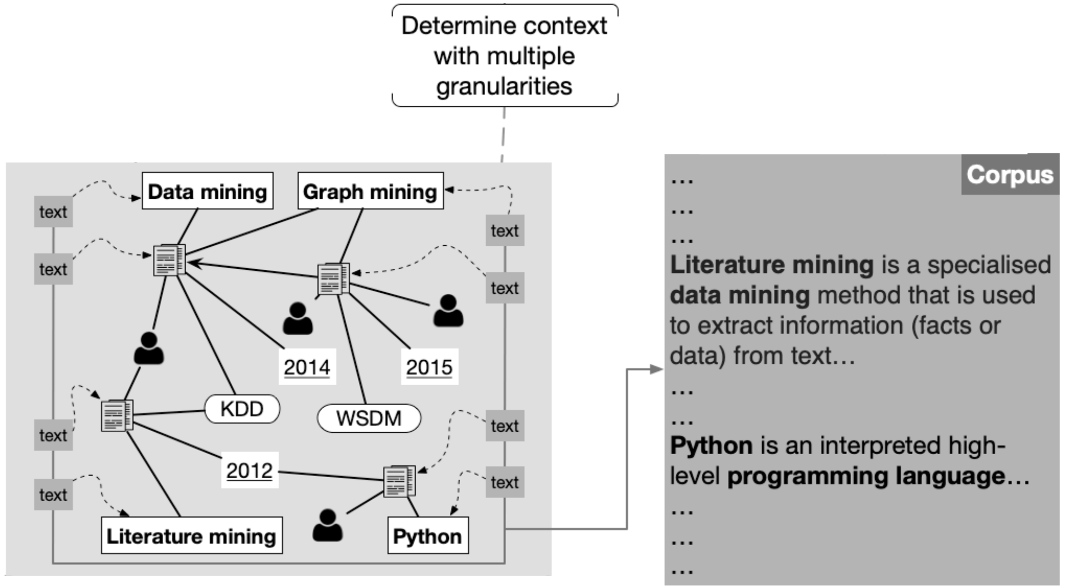
The HDCG Framework



The HDCG Framework

Determine contexts

- **Simplest**: each context unit is a neighbor
- **Grp-by-type**: group by a certain type of nodes (e.g., author)
- **Clus-K**: cluster the network into K clusters using embedding (HEER) + K-means



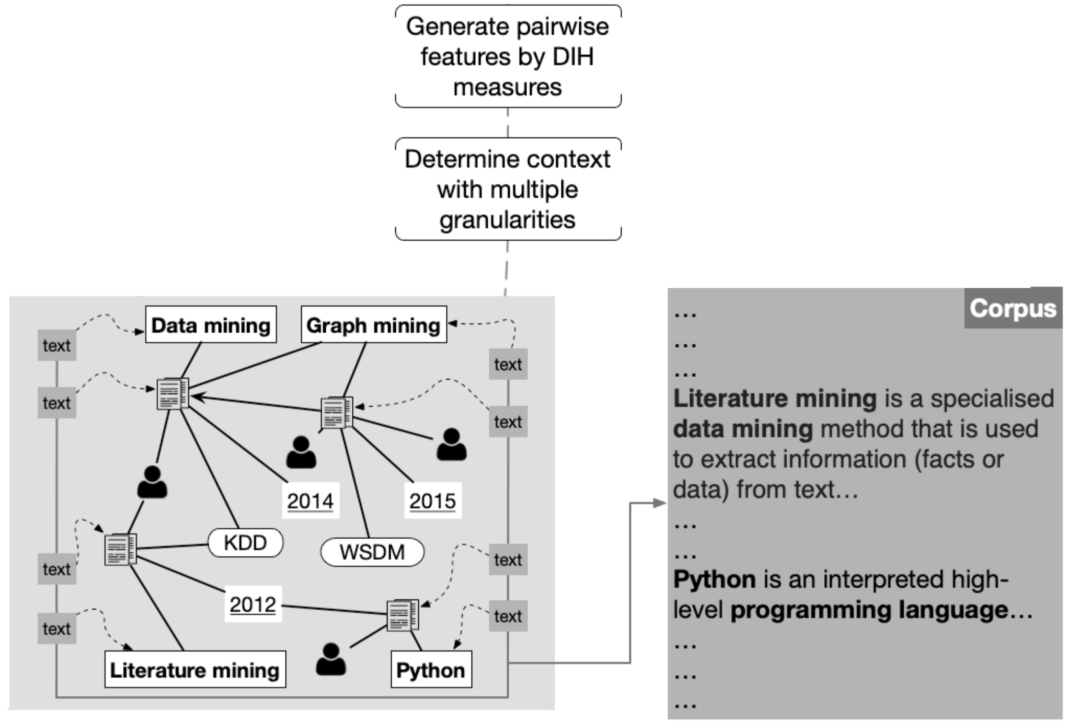
The HDCG Framework

<u>Generate DIH features</u> (pairwise)

For each pair of target nodes, apply 4 DIH measures in each context

- **M1** (*WeedsPrec*)
  $M_1(t_1 \to t_2) = \sum_{c \in C^{(t_1)} \cap C^{(t_2)}} r_c(t_1) / \sum_{c \in C^{(t_1)}} r_c(t_1).$
- **M2** (*invCL*)
  $M_2(t_1 \to t_2) = \sqrt{ClarkDE(t_1 \to t_2) \cdot [1 - ClarkDE(t_2 \to t_1]}.$
- **M3**
  $M_3(t_1 \to t_2) = ClarkDE(t_1 \to t_2) - ClarkDE(t_2 \to t_1).$
- **M4**
  $M_4(t_1 \to t_2) = \sum_{c \in C^{(t_1)} \cap C^{(t_2)}} \min(r_c(t_1), r_c(t_2)) / |C|.$

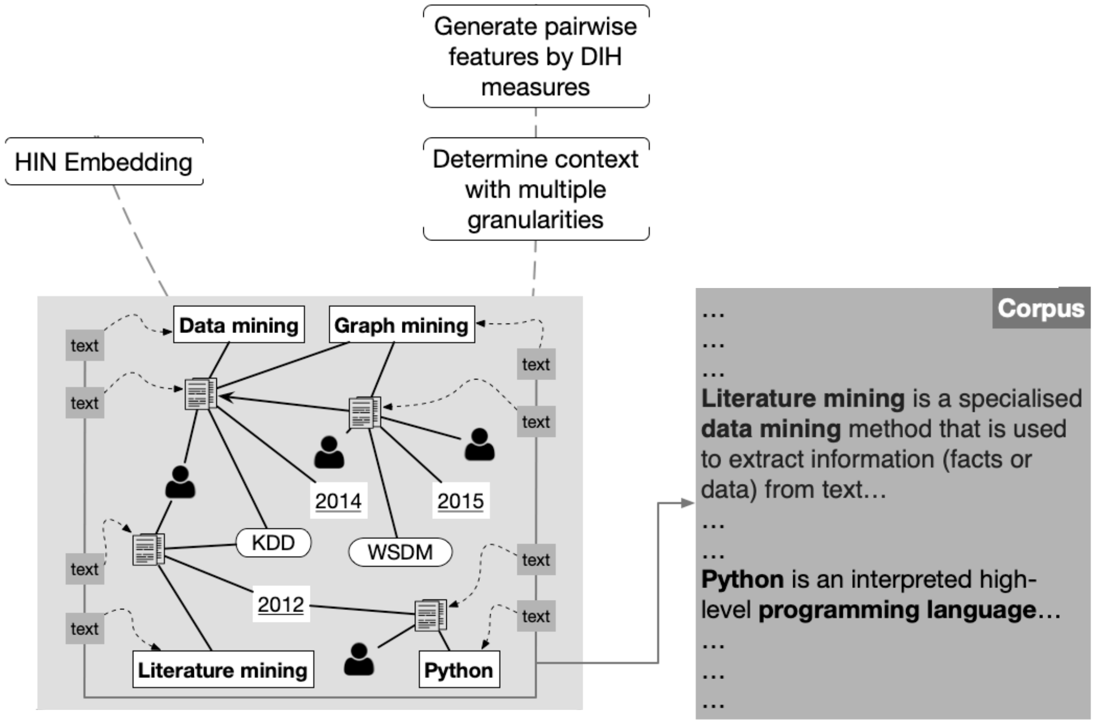
With 5 contexts, each pair has 4×5=20 DIH features



The HDCG Framework

<u>Generate nodewise features</u>

For each target node, apply an existing algorithm (HEER [1]) to learn its representation in the HIN



[1] Shi, Yu, et al. "Easing embedding learning by comprehensive transcription of heterogeneous information networks." In KDD, 2018.

The HDCG Framework

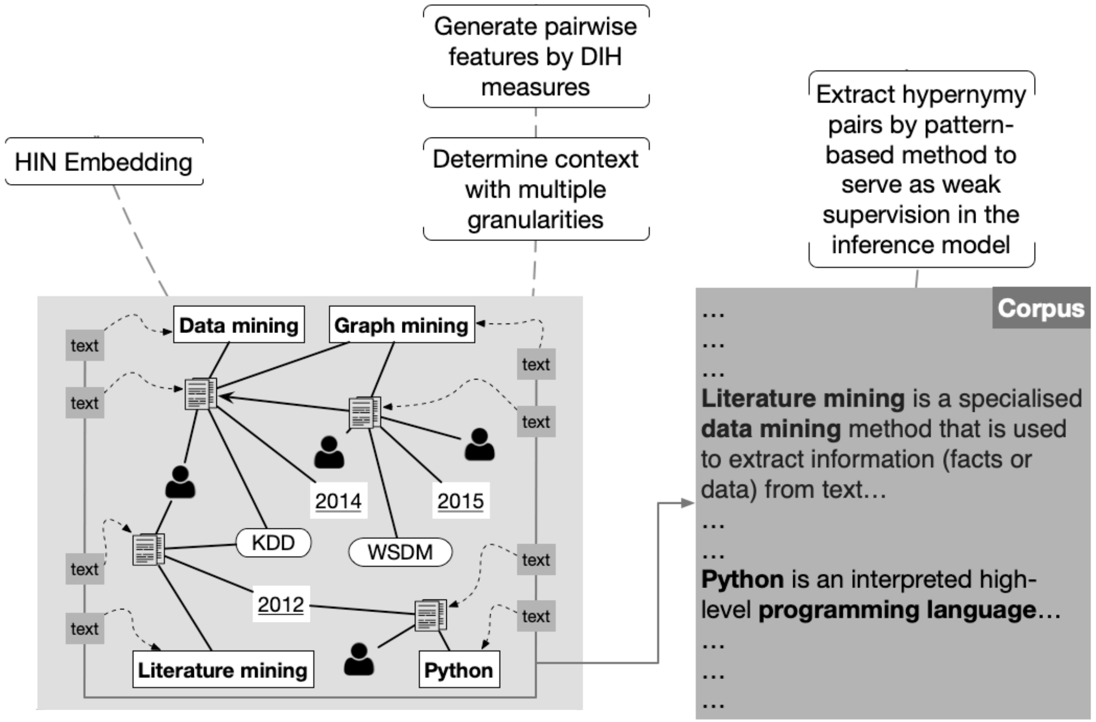
<u>Generate nodewise features</u>

For each target node, apply an existing algorithm (HEER [1]) to learn its representation in the HIN

<u>Generate weak supervision</u>

Using pattern-based–method (Hearst Pattern)

- **High precision**, low recall



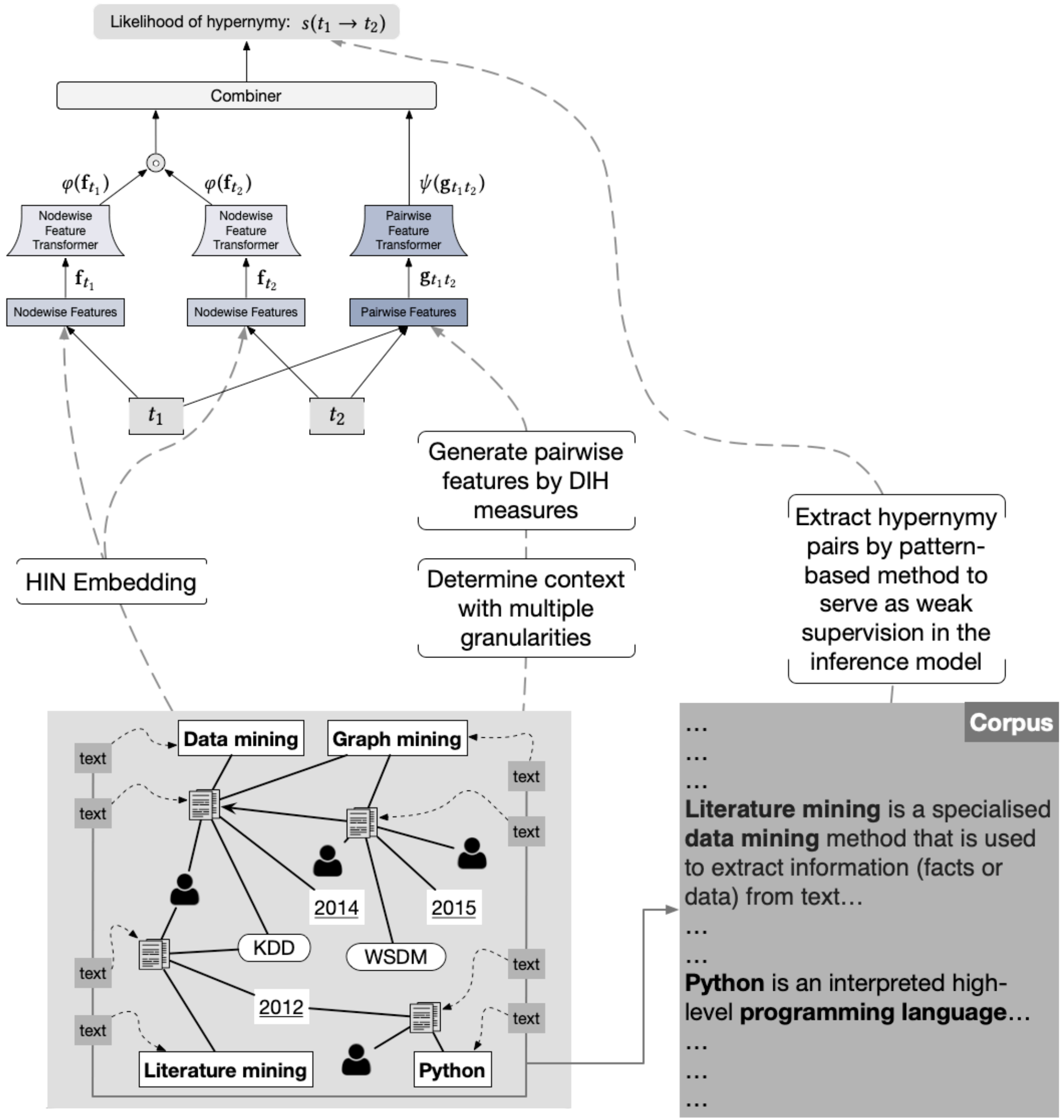
[1] Shi, Yu, et al. "Easing embedding learning by comprehensive transcription of heterogeneous information networks." In KDD, 2018.

The HDCG Framework

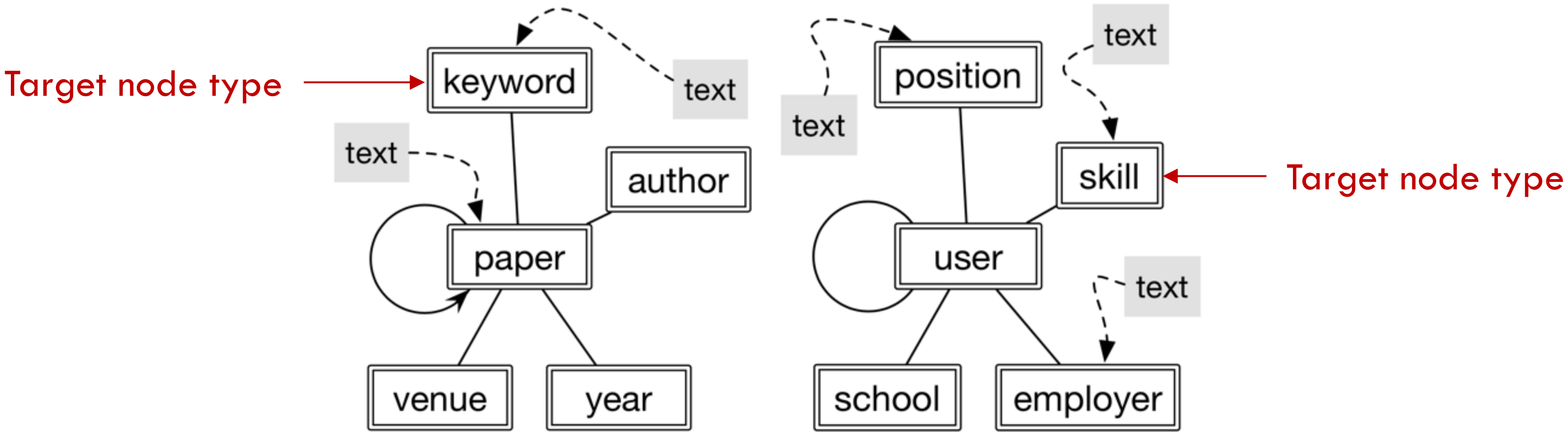
<u>Inference model</u>

A neural network model adapted from the Siamese Network

- Pairwise DIH features
- Nodewise features
- Weak supervision from corpus



Data Description



(a) DBLP. (b) LinkedIn.

Figure 3: The schemata of two text-rich HINs.

Table 1: Basic statistics for the DBLP and LinkedIn dataset, where K's stands for thousands and M's represents millions. The number of sentences is reported for the corpus size $|\Gamma|$.

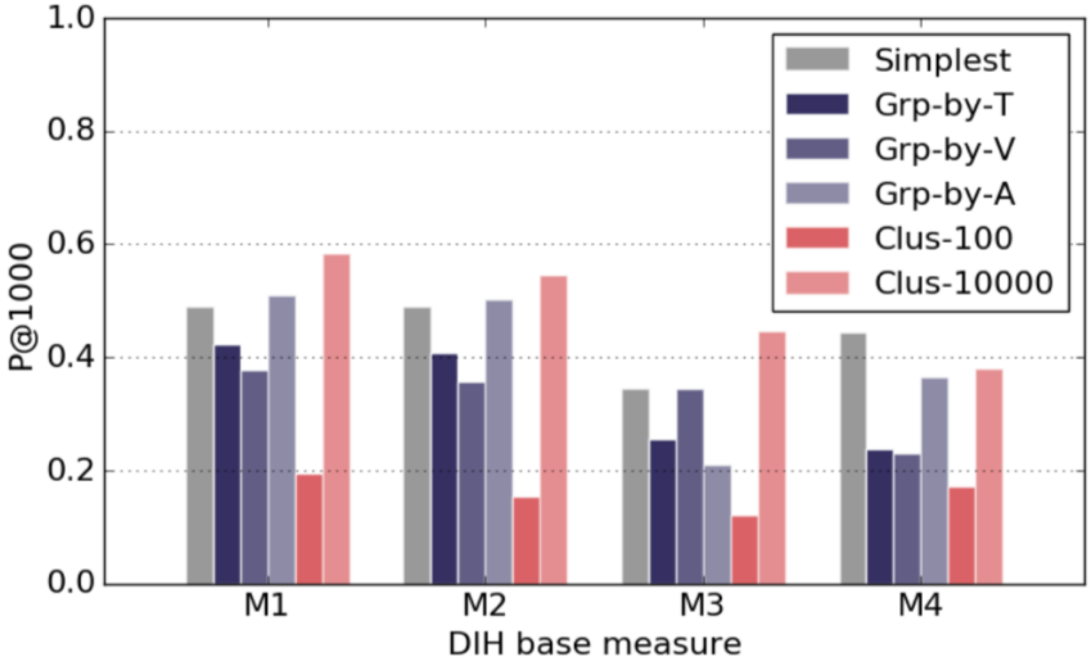
Dataset	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	$\|\mathcal{T}\|$	$\|\mathcal{R}\|$	$\|\mathcal{D}\|$	$\|\Gamma\|$
DBLP	3,715,234	20,594,906	5	5	32,688	10,147,503
LinkedIn	M's	hundreds of M's	5	5	5,000	tens of M's

The higher the better for all metrics

		Precision based		Reciprocal rank based				Precision based		Reciprocal rank based			
	Dateset	DBLP						LinkedIn					
	Metric	P@100	P@1000	MaMARR	MiMARR	MaMLRR	MiMLRR	P@100	P@1000	MaMARR	MiMARR	MaMLRR	MiMLRR
Pattern based	Hearst [13]	0.550	0.163	0.071	0.032	0.304	0.534	0.680	0.259	0.071	0.066	0.425	0.580
Network based	LAKI [21]	0.180	0.191	0.096	0.038	0.382	0.602	0.870	0.491	0.137	0.133	0.508	0.657
	Poincaré [26]	0.110	0.088	0.064	0.028	0.277	0.509	0.110	0.114	0.036	0.028	0.212	0.288
Text based, supervised	LexNET [36]	0.580	0.337	0.121	0.044	0.463	0.542	0.660	0.529	0.129	0.098	0.534	0.605
	HDCG-wo-CG	0.790	0.402	0.148	0.061	0.544	0.757	**0.920**	**0.847**	0.410	0.387	0.809	0.859
	HDCG	**0.880**	**0.620**	**0.358**	**0.148**	**0.745**	**0.865**	0.860	0.835	**0.447**	**0.414**	**0.842**	**0.890**

- HDCG-based models outperform all baselines.
- While the **state-of-the-art** LexNET (**text only**) generally outperform all other baselines, it is clearly worse than any HDCG-based model.
  - Validated the utility of introducing network signals in hypernymy discovery.

Feature Importance



Result of each single DIH feature (per context per DIH measure) in DBLP

- Simultaneously leveraging pairwise features from multiple contexts can bring in performance boost.
    - Compared w/ full HDCG: 0.620.
- No context granularity is always the best even in the same dataset.

Case Study: Taxonomy Construction

- Use existing unsupervised algorithm to construct a taxonomy (a DAG) from the output of HDCG.

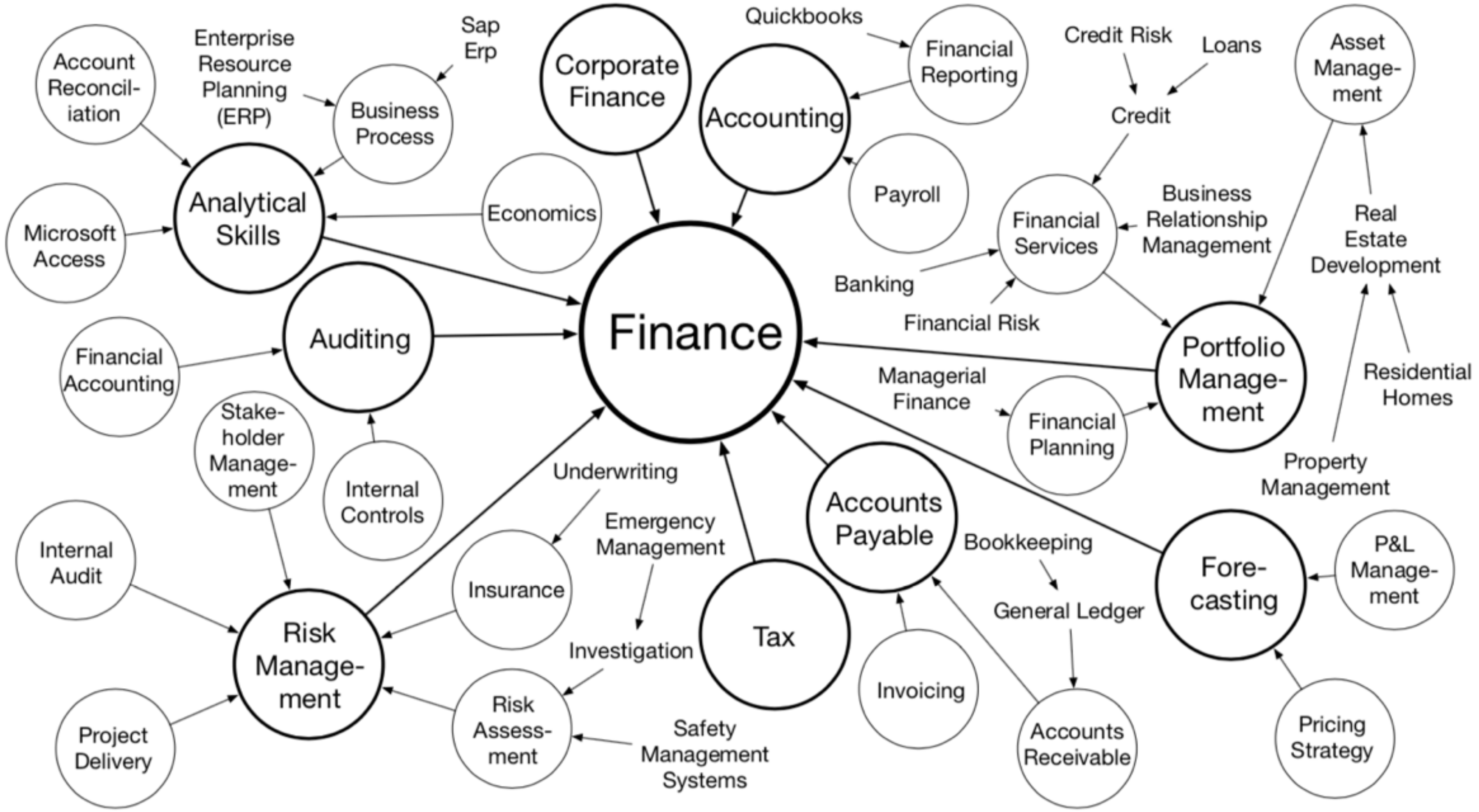


Figure 7: Partial view of a skill taxonomy constructed from the hypernymy discovered from the LinkedIn dataset.

- Generally reasonable. The discovered hypernymy pair with hypernymy scores are still useful when human labelers wish to seek recommendation in taxonomy construction.

Summary

- We propose to discover hypernymy from text-rich HINs, which introduce high-quality network signals in the task of hypernymy discovery.
- We identify the importance of modeling context granularity in distributional inclusion hypothesis (DIH).
- We then propose the HyperMine framework that exploits multi-granular contexts and leverages both network and textual signals for the problem of hypernymy discovery.
- Experiments and case study demonstrate the effectiveness of HyperMine as well as the utility of considering context granularity.