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[†]

[†]Department of Computer Science, University of Illinois Urbana-Champaign
[‡]LinkedIn Corporation [§]Mesh Korea
{yushi2, js2, li215, nzhang31, xhe17, zlou4, qz3, hanj}@illinois.edu
[‡]mtwalker@linkedin.com [§]mykim@cs.stanford.edu

ABSTRACT

Text-rich heterogeneous information networks (text-rich HINs) are ubiquitous in real-world applications. Hypernymy, also known as *is-a* relation or *subclass-of* relation, lays in the core of many knowledge graphs and benefits many downstream applications. Existing methods of hypernymy discovery either leverage textual patterns to extract explicitly mentioned hypernym-hyponym pairs, or learn a distributional representation for each term of interest based its context. These approaches rely on statistical signals from the textual corpus, and their effectiveness would therefore be hindered when the signals from the corpus are not sufficient for all terms of interest. In this work, we propose to discover hypernymy in text-rich HINs, which can introduce additional high-quality signals. We develop a new framework, named HyperMine, that exploits multi-granular contexts and combines signals from both text and network without human labeled data. HyperMine extends the definition of "context" to the scenario of text-rich HIN. For example, we can define typed nodes and communities as contexts. These contexts encode signals of different granularities and we feed them into a hypernymy inference model. HyperMine learns this model using weak supervision acquired based on high-precision textual patterns. Extensive experiments on two large real-world datasets demonstrate the effectiveness of HyperMine and the utility of modeling context granularity. We further show a case study that a high-quality taxonomy can be generated solely based on the hypernymy discovered by HyperMine.

KEYWORDS

Hypernymy Discovery; Heterogeneous Information Network; Text-rich Network; Distributional Inclusion Hypothesis

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3357866

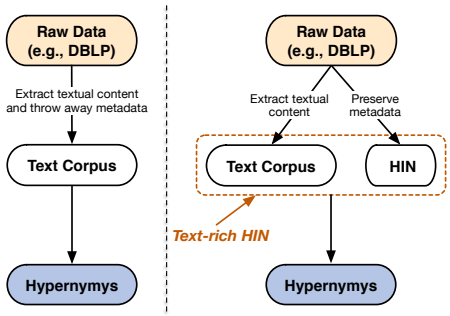


Figure 1: Comparison between previous work on hypernymy discovery only from text (left figure) and the proposed task of discovering hypernymy from text-rich HIN (right figure).

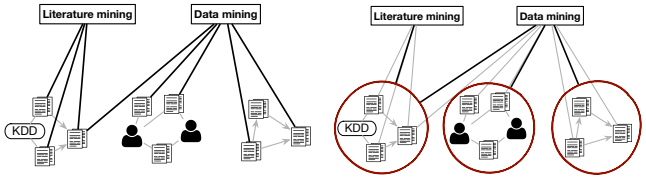
ACM Reference Format:
Yu Shi[†⋆], Jiaming Shen[†⋆], Yuchen Li[†], Naijing Zhang[†], Xinwei He[†], Zhengzhi Lou[†], Qi Zhu[†], Matthew Walker[‡], Myunghwan Kim[§], Jiawei Han[†] . 2019. Discovering Hypernymy in Text-Rich Heterogeneous Information Network by Exploiting Context Granularity. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3357866

1 INTRODUCTION

Heterogeneous information network (HIN), as a powerful data model, has been widely studied since the past decade [40, 49, 59]. Many real-world HINs contain nodes associated with rich textual information [54, 55, 60], and we refer to them as *text-rich HINs*. Typical examples of text-rich HINs include bibliographical networks (*e.g.*, PubMed, DBLP) where nodes representing research papers are associated with their contents and social networks (*e.g.*, Facebook, LinkedIn) in which nodes representing users are attached with their self-descriptive text. These text-rich HINs encapsulate both structured and unstructured information and empowers many downstream tasks such as document clustering [54], topic modeling [35], and event detection [63].

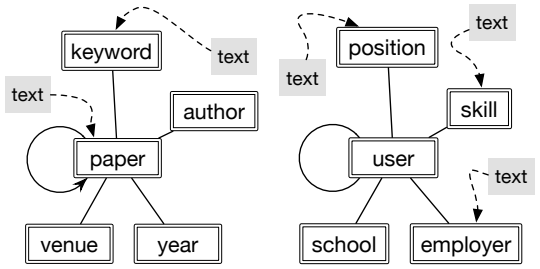
Hypernymy, also known as *is-a* relation or *subclass-of* relation, is a semantic relation between two terms. For example, "*panda*" is a "*mammal*" and "*data structure*" is a subclass of "*computer science*".

[⋆]These authors contributed equally to this work.



(a) Each paper node is defined as a context unit. **(b) Each red circle is defined as a context unit.**

Figure 2: Example of the two definitions of context units with different granularities. By redefining a more coarse-grained context, we have the contexts of "*Literature mining*" to be a subset of the contexts of "*Data mining*".



(a) A bibliographical network (DBLP). **(b) A professional social network (LinkedIn).**

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

Furthermore, we refer to a term t_1 as the *hypernym* of term t_2 and t_2 as the *hyponym* of t_1, if t_2 can be categorized under t_1. In the above case, "*panda*" and "*data structure*" are hyponyms while "*mammal*" and "*computer science*" are hypernyms. Hypernymy lays the foundations of many knowledge bases and knowledge graphs such as YAGO [48], DBpedia [2], and WikiData [52]. Discovering high-quality hypernymy can also benefit many downstream applications such as question answering [45, 58], query understanding [14], and taxonomy construction [22, 37, 62].

Existing work on hypernymy discovery focuses on detecting hypernymy pairs from massive text corpora. These methods typically fall into two categories — pattern-based methods [13, 32, 36] and distributional methods [8, 45, 56]. Pattern-based methods leverage high-precision textual pattern (*e.g.*, "X such as Y") to extract hypernymy pairs. However, these patterns are usually language-dependent and have low recall, in the sense that they can only match explicitly stated hypernymy pairs in text. On the other hand, distributional methods, primarily based on the distributional inclusion hypothesis [64], assume that the context of a hyponym is a subset of the context of a hypernym. A variety of textual contexts are defined, including nearby words in local window [64], adjacent words in parse tree [3], or documents containing the term [39], with different term-context weighting measures [18, 33, 56]. However, this approach usually performs poorly when only one type of textual context is used [32, 46]. Moreover, to simultaneously model all types of contexts, this approach requires additional training hypernymy pairs which are often unavailable.

In this work, we propose to discover hypernymy from text-rich HINs. The motivation for studying this problem is two-fold. First, text-rich HIN is a better data model to preserve all the information in raw data (c.f. Figure 1). For example, there are three existing competitions related to hypernymy discovery [4, 5, 7] and all of them use Wikipedia as their raw data. However, they only extract textual contexts from Wikipedia and simply discard all other structured information such as hyperlinks and Wikipedia categories, which causes severe information loss. Second, when the input corpus is small (*e.g.*, the ACL Anthology where only NLP papers are included), existing methods may not work as there are less statistical signals such as co-occurrence and less chance of matching a textual pattern [61]. However, if documents in this corpus are linked (*e.g.*, by citation relations), we can model these documents using a text-rich HIN (*e.g.*, the ACL Anthology Network [27]). Then, we can leverage the network part of the text-rich HIN to derive additional high-quality signals, which to some extent increases the applicability of hypernymy discovery method.

To discover hypernymy from text-rich HINs, we develop a new unsupervised framework, named HyperMine, that exploits contexts of different granularities and combines signals from both text and network without any human labeled data. HyperMine extends the definition of "context" in distributional inclusion hypothesis (DIH) to include nodes in the network. For instance, in the DBLP network, we can define the context of each *keyword* node to be the set of *paper* nodes directly linked to it, as depicted in Figure 3a. However, such a simple definition of context may not be proper for hypernymy discovery, as shown in the following example and Figure 2a:

Example. *We are given a pair of keywords "literature mining" (t_1) and "data mining" (t_2) and aim to predict whether it is a hypernymy pair based on DIH. We denote all papers linked to "literature mining" as $C^{(t_1)}$ and those linked to "data mining" forms $C^{(t_2)}$. If the DIH holds in this case, we should have $C^{(t_1)} \subseteq C^{(t_2)}$ as "data mining" is a hypernym of "literature mining". However, a paper with "literature mining" linked as a keyword may not need to additionally tag the more general "data mining", and thus such scenario would violate the assumption of the DIH. This can happen whenever the hypernym is too general, and the contextual units are too fine-grained.*

HyperMine resolves the above issue by redefining the "context" to be a group of semantically relevant papers, instead of each single paper. Under this new definition, the desired property $C^{(t_1)} \subseteq C^{(t_2)}$ would hold as shown in Figure 2b. Such new contextual unit has more coarse granularity and is more general. In fact, we observe that the generality of a hypernymy pair (*i.e.*, the overall generality of two terms involved in a hypernymy pair) is highly coupled with the granularity of the context, Therefore, we design HyperMine to incorporate multiple contexts with different granularities.

Finally, we need to combine signals from the textual part and the network part of the input data without resorting to human labeled training data. To tackle this challenge, we first leverage textual patterns to extract a small set of quality hypernymy pairs from the textual part of text-HIN. The intuition is that while pattern-based methods have low recall, they tend to achieve high precision and thus will return a small set of quality hypernymy pairs. Then, we serve this small "seed" set into a hypernym inference model as weak supervision. This model takes two terms as input; calculates their nodewise features (*e.g.*, embeddings in the text-HIN) and pairwise features (*e.g.*, DIH measures based on multi-granular contexts), and

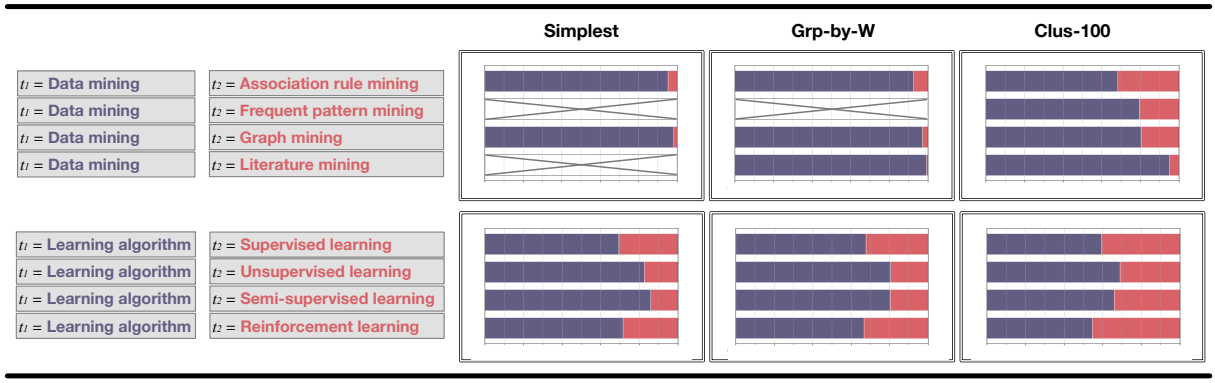


Figure 4: Using the DBLP dataset, we demonstrate the observation that different hypernymy pairs should be discovered at different context granularities. Each row in the plot corresponds to a hypernymy pair $t_1 \rightarrow t_2$, and each column corresponds to a context granularity. In each bar, the length ratio between the left blue part and the right red part is proportional to the ratio between $M_1(t_1 \rightarrow t_2)$ and $M_1(t_2 \rightarrow t_1)$.

returns a score indicating how likely they constitute a hypernymy pair. To better leverage information from the weakly-labeled set, we train the hypernymy inference model using a contrastive loss that penalizes the model whenever it predicts a higher score between a target term with its non-hypernym than the same target term with its true hypernym.

In summary, this paper makes the following contributions: (1) We propose to discover hypernymy from text-rich heterogeneous information networks (HINs), which introduces additional high-quality signals beyond text corpora; (2) We identify the impact of context granularity on distributional inclusion hypothesis (DIH) and propose the HyperMine framework that exploits multi-granular contexts and meanwhile combines signals from both textual and network data in text-HINs, and (3) We conduct extensive experiments to validate the utility of modeling context granularity and the effectiveness of leveraging HIN signals in hypernymy discovery. We further present a case study showing that HyperMine is able to discover high-quality hypernymy pairs for taxonomy construction.

2 PROBLEM FORMULATION

We first elaborate on some related concepts and useful notations and then formulate our problem.

*Definition 2.1 (**Heterogeneous Information Network** [49]).* An *information network* is a directed graph $G = (\mathcal{V}, \mathcal{E})$ with a node type mapping $\varphi : \mathcal{V} \rightarrow \mathcal{T}$ and an edge type mapping $\psi : \mathcal{E} \rightarrow \mathcal{R}$. When $|\mathcal{T}| > 1$ or $|\mathcal{R}| > 1$, the network is called a *heterogeneous information network* (HIN). An HIN is referred to as a **text-rich HIN** if a portion of its nodes are associated with textual information that collectively constitute a *corpus* $\mathcal{D}$.

Given the typed essence of HINs, the network schema $\tilde{G} = (\mathcal{T}, \mathcal{R})$ [49] is used to abstract the meta-information regarding the node types and edge types in an HIN. Figure 3a illustrates the schema of the DBLP network, where a *paper* node may have additional textual information from its content and a *keyword* node may be associated with its description from Wikipedia. Similarly, the schema of a social network in Figure 3b consists of 5 node types with *skill*, *employer*, and *position* having textual information.

*Definition 2.2 (**Target Node Type and Vocabulary**).* A *target node type* $T \in \mathcal{T}$ is a node type where each node of this type corresponds to a textual term (*i.e.*, a word or a phrase)[1]. We refer to the set of all such terms as the *target vocabulary* $-$.

In the heterogeneous bibliographic network, DBLP, the target node type can be *keyword*, and in a heterogeneous professional social network, LinkedIn, the target node type can be *skill*.

*Definition 2.3 (**Context in DIH Measures**).* Measures based on distributional inclusion hypothesis are defined on a given domain of *context* C, over which each term in the target vocabulary $-$ has a relevance distribution. Given a term $t \in -$ and a *contextual unit* $c \in C$, we denote the relevance between t and c as $r_c(t)$.

Additionally, we denote the subdomain of the context that are relevant to term t by $C^{(t)} := \{c \in C \,|\, r_c(t) \neq 0\}$. The primary intuition of measures based on distributional inclusion hypothesis is that $C^{(t_1)}$ should include $C^{(t_2)}$ if t_1 is a hypernym of t_2, and one widely-used DIH measure, *WeedsPrec* [56], is given by

$$M_1(t_1 \rightarrow t_2) = \frac{\sum_{c \in C^{(t_1)} \cap C^{(t_2)}} r_c(t_2)}{\sum_{C^{(t_2)}} r_c(t_2)}. \quad (1)$$

The higher the *WeedsPrec* score, the more likely t_1 is a hypernym of t_2. In the traditional task of hypernymy discovery from a corpus, the typical definition of C is the set of all words that co-occur with term t in the corpus. Finally, we define our task as follows.

*Definition 2.4 (**Problem Formulation**).* Given a text-rich HIN $G = (\mathcal{V}, \mathcal{E})$ and a target node type T corresponding to a target vocabulary $-$, the problem of *hypernymy discovery from this text-rich HIN* aims to discover a list of hypernymy pairs with confidence scores where the hypernyms and the hyponyms are terms from $-$.

3 EXPLOITING CONTEXT GRANULARITY

In this section, we further illustrate our observation that different hypernymy pairs should be discovered with contexts of different granularities. Such observation motivates us to design the HyperMine framework in the next section. Figure 4 presents the

[1]Therefore, in the following, we use "*term*" and "*node*" interchangeably.

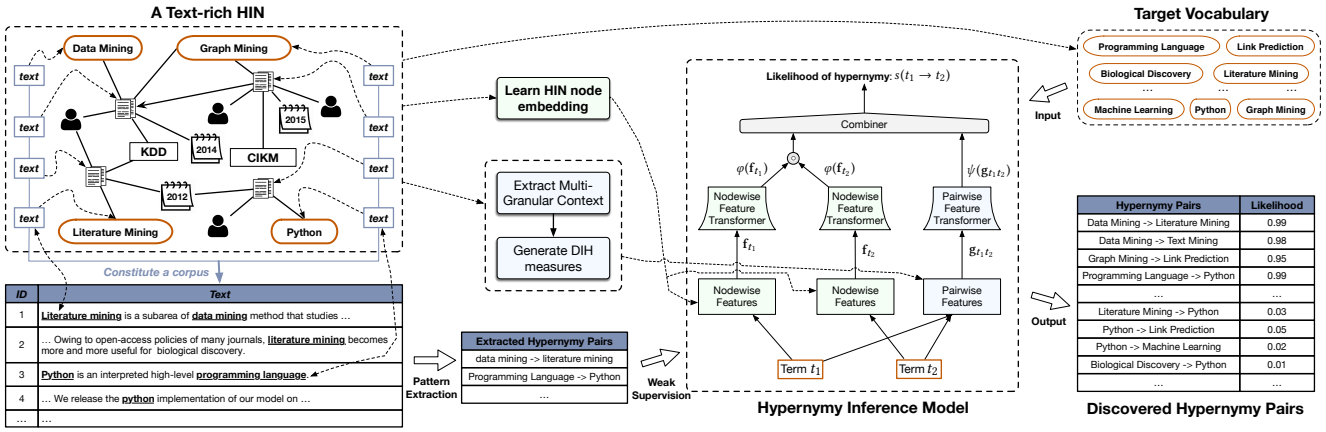


Figure 5: The overview of the proposed HyperMine **framework. We discover hypernymy by exploiting rich signals from the network data besides the corpus. A hypernymy inference model is trained using weak supervision extracted by high-precision patterns from the textual part of the text-rich HIN. A rich pool of features with comparably good term pair coverage is generated from the network part.**

WeedsPrec scores M_1, as calculated by Eq. (1), between eight hypernymy pairs computed at three contexts with different granularities — *Simplest*, *Grp-by-W*, and *Clus-100*, where *Simplest* has the finest granularity and *Clus-100* has the coarsest granularity. We will introduce their concrete definitions in Section 4.2. For each bar, the blue part on the left end is expected to be clearly longer than the red part on the right end, so that *WeedsPrec* may be able to reveal t_1 as a hypernym of t_2 instead of the reversed order.

In the first column (*Simplest*) of the figure, the ratio between $M_1(t_1 \to t_2)$ and $M_1(t_2 \to t_1)$ cannot be visualized for two pairs involving "data mining" since their M_1 are trivially zero. This undesirable result is the outcome of the fact that "frequent pattern mining" and "data mining" are never linked to the same contextual unit in the DBLP dataset to be described in Section 5.1, and likewise for "literature mining" and "data mining". In the context represented by the second column (*Grp-by-W*), *WeedsPrec* is still trivially zero for one pair. Fortunately, if we choose the context in the last column, *WeedsPrec* can generate scores for all four pairs with hypernymy "data mining". As a result, by leveraging a coarser context, we can obtain more features and thus improve the recall.

However, the context in the last column (*Clus-100*) is not always the best for all hypernymy pairs. In the example of "reinforcement learning" (t_1) and "learning algorithm" (t_2), $M_1(t_1 \to t_2)$ and $M_1(t_2 \to t_1)$ are close to each other, which makes it hard to decisively assert "reinforcement learning" is a hyponym of "learning algorithm". On the other hand, the distinction between $M_1(t_1 \to t_2)$ and $M_1(t_2 \to t_1)$ are much clear at the *Simplest* context in the first column. We interpret this result as the generality of a hypernymy pair is coupled with the granularity of the context, and hypernymy relations should, therefore, be revealed at multiple granularities.

Lastly, we emphasize that resolving the problem of the DIH shown in Figure 2a and Figure 4 by exploiting context granularity is easier with the availability of HINs as input. This is because, in an HIN, one can easily define semantically meaningful contextual units using explicit network structures such as grouping by a specific node type or more complex structures including meta-paths or motifs [40, 49]. Furthermore, one may also use network clustering methods to derive contextual units on a broad spectrum of granularities by varying the number of clusters.

4 THE HYPERMINE FRAMEWORK

We tackle the problem of discovering hypernymy from text-rich HINs by combining signals from both text and network using a hypernymy inference model. This model infers the likelihood of a term $t_1 \in \Gamma$ being a hypernym of another term $t_2 \in \Gamma$. To learn this model without human labeled data, we leverage high-precision patterns to extract a set of hypernymy pairs from the corpus, which serves as the weak supervision. Then, we exploit the information encoded in HIN and generate a rich pool of multi-granular features for every single term and every term pair. These features help to increase the recall. Particularly, we encode the network signals into nodewise features using HIN embedding and into pairwise features using the DIH-based measures under various context granularities. Note that while the training of hypernymy inference model is weakly supervised, the whole HyperMine framework is unsupervised, as shown in Figure 5.

4.1 Weak Supervision Acquisition

We generate weak supervision for the hypernymy inference model from the corpus $\mathcal{D}$ of the input text-rich HIN. As a pioneering method, the Hearst pattern [13] has been shown to have decent precision [22, 53, 57]. We use this method to extract a list $\mathcal{S} = \{(t_i^{\wedge}, t_i^{\vee})|_{i=1}^{|\mathcal{S}|}\}$ of hypernymy pairs from the corpus $\mathcal{D}$. In Section 5, we will quantitatively evaluate the validity of this method for generating weak supervision.

4.2 Nodewise Feature Generation

Network embedding has emerged as a powerful representation learning approach, which has been proven effective in many application scenarios [9]. A network embedding algorithm generally learns an embedding function $f : \mathcal{V} \to \mathbb{R}^d$ that maps a node to a

d-dimensional vectorized representation. In our HyperMine framework, we generate a feature $\mathbf{f}_v := f(v)$ for each node $v \in \mathcal{V}$ using a network embedding algorithm designed for HINs from an existing study [41]. As each node of the target type in HIN represents a term, we can use this embedding function to derive the nodewise feature $\mathbf{f}_t$ for each term t in the target vocabulary.

4.3 Pairwise Feature Generation

As introduced in Section 1 and 3, the DIH measures can be naturally extended to generate pairwise features from networks, whose power can be further unleashed by modeling context granularity. For each term pair $(t_1, t_2) \in \Gamma \times \Gamma$, we generate a feature using one of the many DIH measures under one specific context definition. Each DIH measure in our framework is henceforth referred to as a *base DIH measure*. In the following, we introduce the base DIH measures used in the HyperMine framework, together with various approaches to defining contexts with different granularities.

4.3.1 Base DIH measures. For a given context C, we define the following four DIH measures.

• **M1** (*WeedsPrec* [56]) is one of the pioneering DIH measures defined as below:

$$M_1(t_1 \to t_2) = \frac{\sum_{c \in C^{(t_1)} \cap C^{(t_2)}} r_c(t_2)}{\sum_{C^{(t_2)}} r_c(t_2)}.$$

• **M2** (*invCL* [18]) is another widely used DIH measure which considers not only how likely t_1 is a hypernym of t_2 but also how unlikely t_2 is a hypernym of t_1.

$$M_2(t_1 \to t_2) = \sqrt{ClarkDE(t_1 \to t_2) \cdot [1 - ClarkDE(t_2 \to t_1)]},$$

$$ClarkDE(t_1 \to t_2) = \frac{\sum_{c \in C^{(t_1)} \cap C^{(t_2)}} \min(r_c(t_1), r_c(t_2))}{\sum_{c \in C^{(t_2)}} r_c(t_2)}.$$

• **M3** is a variant of **M2** and shares the same intuition as **M2**.

$$M_3(t_1 \to t_2) = ClarkDE(t_1 \to t_2) - ClarkDE(t_2 \to t_1).$$

• **M4** is a symmetric distributional measure. Although it does not directly capture the inclusion intuition of the DIH, we use it to quantify the relevance of the term pair.

$$M_4(t_1 \to t_2) = \frac{\sum_{c \in C^{(t_1)} \cap C^{(t_2)}} \min(r_c(t_1), r_c(t_2))}{|C|}.$$

4.3.2 Context Definition. A simple way to define context given an HIN and a target node type is to let every node linked to nodes of the target type be a contextual unit. We call the context C defined in this way the *Simplest* context. As discussed in Section 1, atop the *Simplest*, one may redefine contextual units by grouping the original ones that are semantically relevant. With the availability of HIN data, we adopt the following two approaches to alternatively define the context in a broad spectrum of context granularities.

• **Define the context by explicit network structures.** Many explicit network structures can be found in HINs such as the node types, the edge types, the meta-paths, and the meta-graphs [40, 49]. Using these structures, one can design methods to group the original contextual units in the *Simplest* together to derive new contextual units. In this paper, we adopt the most straightforward way and group together all contextual units linked to nodes of a specific node type. We refer to this approach as *Grp-by-type* with *type* being a specific node type. As an example, in the DBLP dataset, a contextual unit in *Grp-by-author* is the collection of all papers written by a particular author. We consider a term $t \in -$ relevant to a contextual unit in *Grp-by-type* as long as t is relevant to at least one original unit that is grouped into the new unit.

• **Define the context by network clustering.** Another way to derive semantically meaningful groups is by network clustering. A great many clustering algorithms have been proposed for clustering HINs [40, 49]. With an intention to experiment with a simple algorithm while leveraging the rich information from HINs, we perform the classic K-means algorithm on the node features $\mathbf{f}_v \in \mathbb{R}^d$ to derive K clusters. Similarly, a term $t \in -$ is relevant to a cluster-based contextual unit as long as t is relevant to at least one original unit in this cluster. This approach is henceforth referred to as *Clus-K*.

Given a term pair $(t_1, t_2) \in - \times -$, we compute a single score using each one of the base DIH measures together with one context type. Therefore, the pairwise feature $\mathbf{g}_{t_1 t_2}$ for term pair (t_1, t_2) has the dimensionality equals to the number of base DIH measures times the number of context types. In this study, we focus our investigation on the benefit of introducing HIN signals and the utility of modeling context granularity, and we hence always set the relevance to be binary, *i.e.*, $r_c(t) = 1$ if relevant and 0, otherwise.

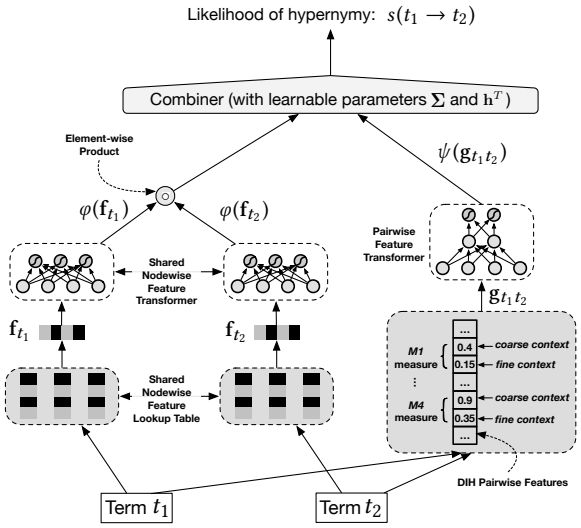


Figure 6: Our proposed hypernymy inference model in HyperMine**.**

4.4 Hypernymy Inference Model

We aim to obtain a model that calculates the likelihood of a pair of terms (t_1, t_2) being a hypernymy pair, using weak supervision extracted from the text part and features from the network part of the input text-rich HIN. The architecture of our hypernymy inference model has three major components, as depicted in Figure 6. The first component is a nodewise feature transformer $\varphi(\cdot)$ that takes the raw nodewise feature $\mathbf{f}$ as input and transforms it into a new embedding space where the hypernymy semantics can be better captured. We design this transformer to be a simple linear layer with dropout followed by a non-linear activation using $tanh(\cdot)$ function. Following the core idea of the Siamese Network [6], we

apply the same nodewise feature transformer to both term t_1 and t_2. The second component is a pairwise feature transformer $\psi(\cdot)$ that acts upon the DIH-based pairwise features. Similarly, we design the pairwise feature transformer using a fully connected neural network with two hidden layers of size N and $N/2$, where N is the dimension of $\mathbf{g}_{t_1t_2}$. Again, we apply dropout for regularization and use $tanh(\cdot)$ function for activation. This pairwise feature transformer can capture the interaction across pairwise features derived from different contexts. The third component is a combiner that aggregates both nodewise and pairwise features after transformation and calculates the hypernymy score by

$$s(t_1 \to t_2) = \varphi(\mathbf{f}_{t_1})^T \Sigma \varphi(\mathbf{f}_{t_2}) + \mathbf{h}^T \psi(\mathbf{g}_{t_1t_2}), \quad (2)$$

where Σ is a diagonal matrix and $\mathbf{h}$ is a vector.

To learn the parameters in both Σ and $\mathbf{h}$, we expect hypernymy pairs to have higher hypernymy scores than non-hypernymy pairs. Therefore, we use the following contrastive loss for model learning

$$\mathcal{L} = \sum_{(t^\wedge, t^\vee) \in \mathcal{S}} \sum_{t^\times \in \mathcal{N}(t^\wedge)} \max\left[0, 1 - s(t^\wedge \to t^\vee) + s(t^\wedge \to t^\times)\right], \quad (3)$$

where $\mathcal{N}(t^\wedge)$ is a randomly sampled set of negative terms such that for any term $t^\times$ in the set, $(t^\wedge, t^\times) \notin \mathcal{S}$. For each pair in $\mathcal{S}$, L negative pairs are sampled. This contrastive loss essentially penalizes the model whenever it predicts a higher score between a target term with its non-hyponym than the same target term with its true hyponym. By minimizing this loss, we can learn our hypernymy inference model.

5 EXPERIMENTS

In this section, we quantitatively evaluate the effectiveness of HyperMine on two real-world large text-rich HINs. An additional case study is also presented using taxonomy construction as our downstream application.

5.1 Data Description

Datasets. To the best of our knowledge, there is no standard benchmark dataset on hypernymy discovery in text-rich HIN. In this work, we use two large real-world HIN datasets[2] for the evaluation.

• **DBLP** is a bibliographical network in the computer science domain, with five node types — *author* (A), *paper* (P), *keyword* (W), *venue* (V), and *year* (Y), and five edge types — authorship, keyword usage, publishing venue, and publishing year of a paper, and the citation relation from a paper to another. The text affiliated to a *paper* node is the title of that paper, and the text associated with a *keyword* node is the raw string of this keyword plus its Wikipedia page if the page exists. We define *keyword* to be our target node type. To generate a set of ground truth hypernymy pairs, we resort to the ACM Computing Classification System (CCS)[3] which organizes computer science topics into a tree-structured taxonomy. A keyword in the vocabulary – is mapped to a topic term in CCS if they can be linked to the same Wikipedia entry using WikiLinker. A positive hypernym-hyponym label is recorded if two keywords are mapped to two CCS terms that have ancestor-descendant relation in the CCS taxonomy. Finally, we obtain 10,055 positive hypernym-hyponym pairs. Then, for each positive pair, we generate ten negative pairs by fixing the hypernym (hyponym) and randomly sampling five non-hyponym (non-hypernymy) keywords. During such negative sampling process, a keyword is always randomly sampled from the set that can be mapped to CCS terms.

• **LinkedIn** is an internal profession social network that has five node types — *user*, *skill*, *employer*, *school*, and *position*, and five edge types – users possessing skills, working for employers, attending schools, holding job positions, and being connected with other users. The text associated to a *skill*, a *position*, and an *employer* are respectively the Wikipedia page on this skill, users' descriptions for this position, and the job posting description created by this employer. The entire network is down-sampled to include only users from a major metropolitan area in the US as well as nodes and edges directly linked to these users. We further filter skills and keep the top 5,000 regarding the number of users having a skill. We define *skill* be the target node type. Positive hypernym-hyponym pairs were curated by the company that owns the data, and the label curation process is independent of our experiments. We take the same process as described above to generate negative pairs.

The schemas of these two HINs are depicted in Figure 3, and we summarize the statistics of the datasets in Table 1.

Table 1: Basic statistics of DBLP and LinkedIn datasets, where 'M' represents million and corpus size $|\mathcal{D}|$ is the number of sentences.

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert-\rvert$	$\lvert\mathcal{D}\rvert$
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

5.2 Compared Methods

We compare our framework with the following methods.

• **Hearst patterns (Hearst)** [13] is a classic pattern-based method for hypernymy discovery from text.

• **LAKI** [21] is a document representation method which first learns a keyword hierarchy based on word embeddings (*i.e.*, nodewise features) and DIH measures at the *Simplest* context, and then assigns documents to this hierarchy. Since LAKI also incorporates both pairwise and nodewise features, we compare our framework with it in order to show that the higher expressive power of our hypernymy inference model indeed helps.

• **Poincaré Embedding (Poincaré)** [25] is an embedding learning algorithm in the hyperbolic space. It can embed an input taxonomy, represented as a directed acyclic graph (DAG), into a hyperbolic space, and then uses learned node embeddings to predict more hypernymy pairs in the DAG. We take it as a baseline because, to the best of our knowledge, this class of algorithms is the only existing ones that are relevant to hypernymy and take graphs or networks as input.

• **LexNET** [43] is a state-of-the-art algorithm for hypernymy discovery from text. LexNet integrates dependency path based signals with distributional signals for predicting hypernymy.

• **HyperMine-wo-CG** is an ablated version of HyperMine which does not model context granularity and derives all the DIH measures based on raw context features.

• **HyperMine** is the full version of our proposed framework[4].

[2] Available at http://bit.ly/HyperMine-dataset.

[3] https://www.acm.org/publications/class-2012

[4] Code available at: https://github.com/ysyushi/HyperMine

Table 2: Quantitative evaluation results of hypernymy discovery from the DBLP and the LinkedIn datasets.

Dateset	DBLP						LinkedIn					
Metric	P@100	P@1000	MaMARR	MiMARR	MaMLRR	MiMLRR	P@100	P@1000	MaMARR	MiMARR	MaMLRR	MiMLRR
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
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 [25]	0.110	0.088	0.064	0.028	0.277	0.509	0.110	0.114	0.036	0.028	0.212	0.288
LexNET [44]	0.580	0.337	0.121	0.044	0.463	0.542	0.660	0.529	0.129	0.098	0.534	0.605
HyperMine-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
HyperMine	**0.880**	**0.620**	**0.358**	**0.148**	**0.745**	**0.865**	0.860	0.835	**0.447**	**0.414**	**0.842**	**0.890**

5.3 Evaluation Metrics and Experiment Setups

Evaluation Metrics. We report evaluation results using two precision metrics and four ranking metrics. The two precision metrics are precision at k (**P@k**) with $k \in \{100, 1000\}$, which are computed as the number of positive pairs among the top-ranked k pairs divided by k. The four ranking metrics are macro mean average reciprocal rank (**MaMARR**), micro mean average reciprocal rank (**MiMARR**), macro mean largest reciprocal rank (**MaMLRR**), and micro mean largest reciprocal rank (**MiMLRR**). To calculate these four ranking metrics, we first group all pairs sharing the same hypernym t in the evaluation data. Then, for each pair, its reciprocal rank (RR) is the reciprocal of the rank of this pair within the group. Since there can be multiple positive pairs in a group, the average reciprocal rank for the group is the average over the RR's of all positive pairs, and the largest reciprocal rank is the largest RR among the RR's of all positive pairs. Finally, we compute the macro mean and the micro mean across all groups to get the final four metrics, where the macro mean assigns uniform weights for all group when calculating the mean and the micro mean assigns weights proportional to the number of positive pairs in each group. We also note that the optimal value the perfect model can achieve for MiMARR and MaMARR may be smaller than 1. This can be explained with an example where a group has three positive pairs, then the highest average reciprocal rank for this group would be $(1/1 + 1/2 + \ldots + 1/6)/6 = 0.408 < 1$. For all six metrics, greater value indicate better performance.

To obtain rank lists, we calculate a confidence score for each pair in the evaluation. All methods expect for baseline Hearst will directly return such confidence score. For example, our hypernymy inference model will return a hypernymy likelihood $s(t_1 \rightarrow t_2)$ which can be naturally viewed as the confidence score. As for baseline Hearst, we simply assign a score of 1.0 for all of its extracted pairs and assign a score of 0.0 for all other pairs. When two pairs have the same scores, we randomly break the ties so that a model predicting all ties would get the same evaluation result as random guess. For fairness across different runs of evaluation and across different models, we fix the random seed in the evaluation pipeline.

Experiment Setups. When determining context by explicit network structure, we use *Grp-by-A*, *Grp-by-V*, *Grp-by-W* for DBLP and *Grp-by-P*, *Grp-by-S*, *Grp-by-U* for LinkedIn. When determining context by clustering, we select two different values for K: 100 and 10000, which yields two contexts *Clus-100* and *Clus-10000*. Therefore, we have totally 6 different contexts (5 derived contexts plus the simplest context), which, times 4 distinct base DIH measures, gives $4 \times 6 = 24$ pairwise features.

For both datasets, we learn the 128-dimension HIN node embedding using HEER [42]. We tune all hyper-parameters of compared methods using 5-fold cross validation on our weak supervision dataset. For the hypernymy inference model in HyperMine, we use a neural network with one hidden layer of size 256 as the nodewise feature transformer. We set negative sampling ratio $L = 10$, the dropout rate for $\varphi(\cdot)$ to 0.7, and dropout rate for $\psi(\cdot)$ to 0.1.

5.4 Quantitative Evaluation Results

The main quantitative evaluation results are presented in Table 2. Overall, the HyperMine-based methods outperform all baselines under all metrics in both datasets by large margins with only one exception for P@100 in the LinkedIn dataset. Furthermore, the full HyperMine model clearly outperforms HyperMine-wo-CG in DBLP and has a competitive performance with HyperMine-wo-CG in LinkedIn dataset.

Notably, the state-of-the-art corpus-based method LexNET mostly excels among all baselines. However, it still performs significantly inferior to HyperMine and HyperMine-wo-CG, which demonstrates the benefit of introducing network signals in the task of hypernymy discovery. Also only taking corpus as the input, Hearst further underperforms LexNET on most metrics. It is worth noting that the precision of Hearst drops drastically when the k in P@k changes from 100 to 1000 in both datasets. In comparison, LexNET has P@100 similar to Hearst, while the former has clearly better P@1000. This outcome further verifies the existing observation that Hearst tends to extract a limited number of term pairs, *i.e.*, low recall, while the precision on the extracted pairs could be decent.

In addition, Poincaré is selected as a baseline because it represents a line of research that is both relevant to hypernymy and takes graphs or networks as input. We find that Poincaré has the worst performance among all baselines, which might be because this algorithm is not designed for discovering hypernymy from data. Furthermore, LAKI also generally performs worse than the other two HyperMine-based models with one exception for P@100 in LinkedIn. We analyze why it achieves better performance in LinkedIn compared to in DBLP dataset in the next paragraph.

The same context granularity can have different importances in different datasets. HyperMine clearly outperforms its partial model HyperMine-wo-CG in DBLP dataset, which demonstrates the utility of leveraging multiple context granularities. However, the comparison between HyperMine and HyperMine-wo-CG has mixed results in LinkedIn dataset. We interpret this as the *Simplest* context is too informative in LinkedIn since each user is linked to more skills on average than each paper is linked to keywords.

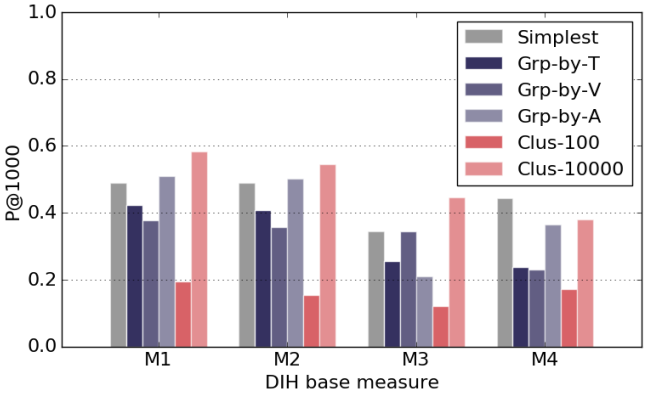


Figure 7: The importance of each DIH feature based on different base measures and different context granularities.

As a result, introducing more features from other context granularities may not bring in a significant performance boost. In fact, low-quality noisy features may even dampen the top-ranked pairs resulting in a lower precision, especially when k is small. This also explains the better performance of LAKI in LinkedIn, since LAKI indeed uses a DIH measure at the *Simplest* context.

Simultaneously leveraging pairwise features from multiple context granularities can introduce performance boost. In Figure 7, we plot out the evaluation result using each single network-based DIH feature on the DBLP dataset. We only present the metric P@1000 and omit the other metrics due to space limitations, and that similar conclusion can be reached based on other metrics. Comparing Figure 7 with Table 2, it can be seen that the proposed method using multi-granular context features achieves elevated performance compared with every single feature. This finding corroborates our previous observation that different hypernymy pairs may be revealed from different context granularities.

No context granularity is always the best even in the same dataset. From the performance of each single feature in Figure 7, it can be seen that not a context granularity is the best under all DIH base measures. For instance, *Clus-10000* has the best performance when coupled with M_1, M_2, and M_3, while in the case of M_4, *Simplest* is the best. Also, while *Simplest* is the best with M_4, it is even slightly worse than *Grp-by-A* when coupled with M_1 and M_2.

5.5 Case Study: Taxonomy Construction

In this section, we show that HyperMine can discover high-quality hypernymy pairs that are useful for taxonomy construction. If we consider each discovered hypernymy pair as a directed edge, putting all pairs together will yield a graph potentially with cycles. However, by definition, a taxonomy is restricted to be a directed acyclic graph (DAG). Therefore, we resort to a simple heuristic algorithm used in many existing taxonomy construction studies [17, 51], which repeatedly finds one cycle in the current graph and then randomly breaks an edge. We consider the resulting DAG as a crude taxonomy.

Due to the scalability limit of the above cycle breaking algorithm, we construct the initial graph with 500 most popular skills and then keep 5000 edges with the highest hypernymy scores. We present the result after the cycle breaking algorithm in Figure 8, which includes the part of the DAG rooted by node *Finance* and all of its hyponyms within the fourth-order neighborhood.

Since only 500 top skills are left when constructing the graph, one should expect the recall of the taxonomy should be limited. The recall aside, the constructed taxonomy has decent overall quality. If we refer to a hyponym of t that also has an edge from t as a child of t, seven out of nine children of *Finance* makes sense except *Tax* and *Analytical Skills*, where the latter two are related to *Finance* but are not precisely its hyponyms. One level deeper, descendants of the seven children are reasonable as well, which further corroborates the effectiveness of our proposed HyperMine framework.

As a final remark, even when such an unsupervised approach may not directly yield a perfect taxonomy, the discovered hypernymy pairs with confidence scores are still useful. For example, when human labelers wish to expand an existing taxonomy to incorporate more nodes, they can seek recommendations from our HyperMine framework.

6 RELATED WORK

In this section, we discuss related work for text-rich HINs and hypernymy discovery.

Heterogeneous Information Network. Heterogeneous information network (HIN) has been heavily studied for its ubiquity in real-world scenarios and its ability to encapsulate rich information [40, 49, 59]. Researchers have demonstrated that using HINs can benefit a wide range of tasks such as classification, clustering, recommendation, and outlier detection [40, 49, 66]. Many real-world HINs are also text-rich with certain types of their nodes associated with additional textual information [10, 55, 60]. One typical example is the HIN with node type *document* or *research paper*. The content of each document or paper provides textual information highly relevant to this node, and such text-rich HINs have been studied in tasks such as clustering [54], topic modeling [35], and literature search [38].

Distributional Method for Hypernymy Discovery. Distributional methods constitute one major line of research for hypernymy discovery [50, 53] and can be adapted to hypernymy discovery from network data. Early studies proposed symmetric distributional measures for hypernymy discovery that only capture relevance between terms [20]. More recently, researchers have investigated into asymmetric measures based on the *distributional inclusion hypothesis* (DIH) to comply with the asymmetrical nature of hypernymy [12, 53, 65]. Examples of popular DIH measures include WeedsPrec [56], APinc and balAPinc [16], ClarkeDE [8], cosWeeds, invCL [18], and WeightedCosine [28].

Closely related to our effort in rectifying the DIH by modeling context granularity, several studies have also studied the validity of the DIH. These studies have also suggested the DIH may not always hold accurate and proposed solutions orthogonal to ours [30, 31, 34]. Santus et al. [34] propose an entropy-based measure SLQS that do not rely on the DIH, while some other studies suggested only certain units in the context should be used to generate features [30, 31]. We note that these approaches do not contradict with ours, because they are all based on the default context granularity, while we

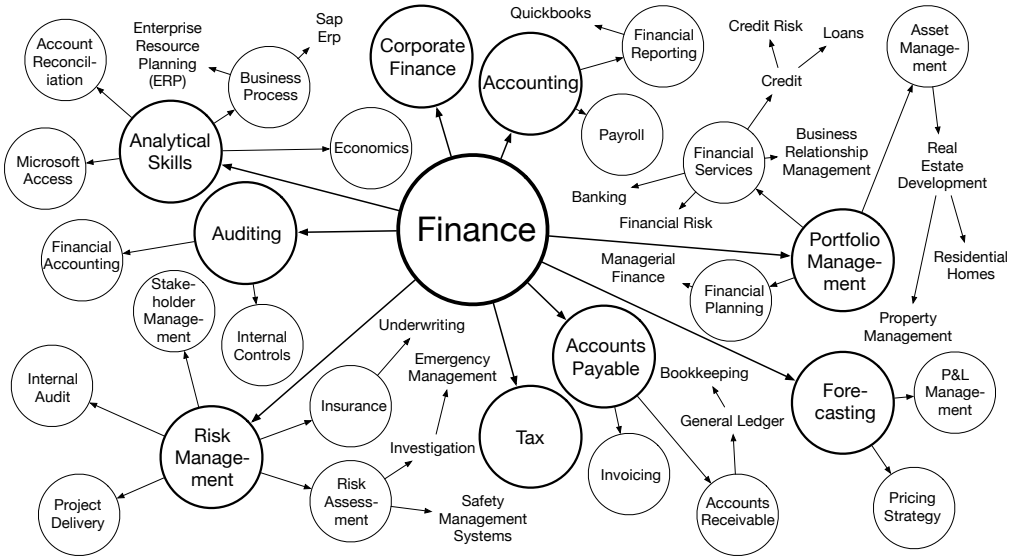


Figure 8: A partial view of the skill taxonomy constructed from the hypernymy pairs discovered by HyperMine **in the LinkedIn dataset.**

argue that DIH would hold at proper context granularities for each hypernym-hyponym pair.

Pattern-based Method for Hypernymy Discovery. Hearst, *et al.* [13] pioneered the line of pattern-based hypernymy discovery methods which leverage hand-crafted lexico-syntactic patterns to extract explicitly mentioned hypernymy pairs from a text corpus. A substantial number of methods have been proposed to extend the original six Hearst patterns [11, 17, 57]. It has been shown that Hearst pattern based methods tend to achieve high precision with compromised recall [22, 32, 53]. Attempts have also been made to further improve the recall [1, 24, 47]. In our framework, we use the straightforward Hearst pattern-based method to extract weak supervision pairs in the hope of yielding pairs with decent precision and without much additional engineering.

Supervised Method for Hypernymy Discovery. With additional supervision available, researchers have proposed models to infer hypernymy based on the representation of a term pair [19, 29, 44]. Methods for deriving such representations include the aforementioned pattern-based methods and distributional methods as well as the compact, distributed representations generated from models such as word2vec [23], GloVe [26], and SensEmbed [15].

7 CONCLUSION AND FUTURE WORK

In this work, we propose to discover hypernymy from text-rich HINs, which avails us with additional rich signals from the network data besides corpus. From real-world data, 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.

Future work can explore more methods to derive contextual units. For example, we can use more complex structures (*e.g.*, network motifs) and HIN-specific clustering methods to further unleash the utility of modeling context granularity. Besides, it is also of interest to extend our framework to consider polysemy in datasets that possess such a characteristic. Furthermore, in this work, we simply treat the textual part of a text-rich HIN as a corpus collection regardless of which node a particular piece of text is associated with. Since a node and the text associated with it are likely to be relevant, we expect that building a more unified model by leveraging such signal could introduce additional performance boost to the task of hypernymy discovery.

ACKNOWLEDGEMENTS

This research is sponsored in part by U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), DARPA under Agreement No. W911NF-17-C-0099, National Science Foundation IIS 16-18481, IIS 17-04532, and IIS-17-41317, DTRA HDTRA11810026, grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative, and LinkedIn Economic Graph Research Program.

REFERENCES

[1] Tuan Luu Anh, Jung-jae Kim, and See Kiong Ng. 2014. Taxonomy construction using syntactic contextual evidence. In *EMNLP*.
[2] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *ISWC/ASWC*.
[3] Marco Baroni and Alessandro Lenci. 2010. Distributional Memory: A General Framework for Corpus-Based Semantics. *Computational Linguistics* (2010).
[4] Georgeta Bordea, Paul Buitelaar, Stefano Faralli, and Roberto Navigli. 2015. SemEval-2015 Task 17: Taxonomy Extraction Evaluation (TExEval). In *SemEval@NAACL-HLT*.
[5] Georgeta Bordea, Els Lefever, and Paul Buitelaar. 2016. SemEval-2016 Task 13: Taxonomy Extraction Evaluation (TExEval-2). In *SemEval@NAACL-HLT*.
[6] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature Verification Using a Siamese Time Delay Neural Network. In *IJPRAI*.

[7] José Camacho-Collados, Claudio Delli Bovi, Luis Espinosa Anke, Sergio Oramas, Tommaso Pasini, Enrico Santus, Vered Shwartz, Roberto Navigli, and Horacio Saggion. 2018. SemEval-2018 Task 9: Hypernym Discovery. In *SemEval@NAACL-HLT*.

[8] Daoud Clarke. 2009. Context-theoretic semantics for natural language: an overview. In *ACL-GEMS*.

[9] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. 2018. A survey on network embedding. *TKDE* (2018).

[10] Hongbo Deng, Jiawei Han, Bo Zhao, Yintao Yu, and Cindy Xide Lin. 2011. Probabilistic topic models with biased propagation on heterogeneous information networks. In *KDD*.

[11] Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. 2004. Web-scale information extraction in knowitall. In *WWW*.

[12] Maayan Geffet and Ido Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *ACL*.

[13] Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Computational linguistics*.

[14] Wen Hua, Zhongyuan Wang, Haixun Wang, Kai Zheng, and Xiaofang Zhou. 2017. Understand Short Texts by Harvesting and Analyzing Semantic Knowledge. *TKDE* (2017).

[15] Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. Sensembed: Learning sense embeddings for word and relational similarity. In *ACL*.

[16] Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering* 16, 4 (2010), 359–389.

[17] Zornitsa Kozareva and Eduard Hovy. 2010. A semi-supervised method to learn and construct taxonomies using the web. In *EMNLP*.

[18] Alessandro Lenci and Giulia Benotto. 2012. Identifying hypernyms in distributional semantic spaces. In *ACL-SEM*.

[19] Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do supervised distributional methods really learn lexical inference relations?. In *NAACL*.

[20] Dekang Lin and others. 1998. An information-theoretic definition of similarity.. In *ICML*.

[21] Jialu Liu, Xiang Ren, Jingbo Shang, Taylor Cassidy, Clare R Voss, and Jiawei Han. 2016. Representing documents via latent keyphrase inference. In *WWW*.

[22] Yuning Mao, Xiang Ren, Jiaming Shen, Xiaotao Gu, and Jiawei Han. 2018. End-to-End Reinforcement Learning for Automatic Taxonomy Induction. *ACL* (2018).

[23] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

[24] Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. 2012. PATTY: a taxonomy of relational patterns with semantic types. In *EMNLP*.

[25] Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *NIPS*.

[26] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

[27] Dragomir R. Radev, Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara. 2009. The ACL anthology network corpus. *LREC* (2009).

[28] Marek Rei and Ted Briscoe. 2014. Looking for hyponyms in vector space. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*. 68–77.

[29] Marek Rei, Daniela Gerz, and Ivan Vulić. 2018. Scoring Lexical Entailment with a Supervised Directional Similarity Network. *arXiv preprint arXiv:1805.09355* (2018).

[30] Laura Rimell. 2014. Distributional lexical entailment by topic coherence. In *EACL*.

[31] Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *COLING*.

[32] Stephen Roller, Douwe Kiela, and Maximilian Nickel. 2018. Hearst Patterns Revisited: Automatic Hypernym Detection from Large Text Corpora. In *ACL*.

[33] Mark Sanderson and W. Bruce Croft. 1999. Deriving Concept Hierarchies from Text. In *SIGIR*.

[34] Enrico Santus, Alessandro Lenci, Qin Lu, and S Schulte im Walde. 2014. Chasing hypernyms in vector spaces with entropy. In *EACL*.

[35] Jiaming Shen, Zhenyu Song, Shitao Li, Zhaowei Tan, Yuning Mao, Luoyi Fu, Li Song, and Xinbing Wang. 2016. Modeling Topic-Level Academic Influence in Scientific Literatures. In *AAAI SBD*.

[36] Jiaming Shen, Zeqiu Wu, Dongming Lei, Jingbo Shang, Xiang Ren, and Jiawei Han. 2017. SetExpan: Corpus-Based Set Expansion via Context Feature Selection and Rank Ensemble. In *ECML/PKDD*.

[37] Jiaming Shen, Zeqiu Wu, Dongming Lei, Chao Zhang, Xiang Ren, Michelle T. Vanni, Brian M. Sadler, and Jiawei Han. 2018. HiExpan: Task-Guided Taxonomy Construction by Hierarchical Tree Expansion. In *KDD*.

[38] Jiaming Shen, Jinfeng Xiao, Xinwei He, Jingbo Shang, Saurabh Sinha, and Jiawei Han. 2018. Entity Set Search of Scientific Literature: An Unsupervised Ranking Approach. In *SIGIR*.

[39] Zhihong Shen, Hao Ma, and Kuansan Wang. 2018. A Web-scale system for scientific knowledge exploration. In *ACL*.

[40] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and S Yu Philip. 2017. A survey of heterogeneous information network analysis. *TKDE* 29, 1 (2017), 17–37.

[41] Yu Shi, Qi Zhu, Fang Guo, Chao Zhang, and Jiawei Han. 2018. Easing Embedding Learning by Comprehensive Transcription of Heterogeneous Information Networks. In *KDD*.

[42] Yu Shi, Qi Zhu, Fang Guo, Chao Zhang, and Jiawei Han. 2018. Easing Embedding Learning by Comprehensive Transcription of Heterogeneous Information Networks. In *KDD*.

[43] Vered Shwartz and Ido Dagan. 2016. CogALex-V Shared Task: LexNET - Integrated Path-based and Distributional Method for the Identification of Semantic Relations. In *CogALex@COLING*.

[44] Vered Shwartz and Ido Dagan. 2016. Path-based vs. distributional information in recognizing lexical semantic relations. *arXiv preprint arXiv:1608.05014* (2016).

[45] Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. *arXiv preprint arXiv:1603.06076* (2016).

[46] Vered Shwartz, Enrico Santus, and Dominik Schlechtweg. 2017. Hypernyms under Siege: Linguistically-motivated Artillery for Hypernymy Detection. In *EACL*.

[47] Rion Snow, Daniel Jurafsky, and Andrew Y Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In *NIPS*.

[48] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *WWW*.

[49] Yizhou Sun and Jiawei Han. 2013. Mining heterogeneous information networks: a structural analysis approach. *SIGKDD Explorations* 14, 2 (2013), 20–28.

[50] Peter D Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research* 37 (2010), 141–188.

[51] Paola Velardi, Stefano Faralli, and Roberto Navigli. 2013. Ontolearn reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics* 39, 3 (2013), 665–707.

[52] Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57 (2014), 78–85.

[53] Chengyu Wang, Xiaofeng He, and Aoying Zhou. 2017. A Short Survey on Taxonomy Learning from Text Corpora: Issues, Resources and Recent Advances. In *EMNLP*.

[54] Chenguang Wang, Yangqiu Song, Ahmed El-Kishky, Dan Roth, Ming Zhang, and Jiawei Han. 2015. Incorporating world knowledge to document clustering via heterogeneous information networks. In *KDD*.

[55] Chenguang Wang, Yangqiu Song, Haoran Li, Yizhou Sun, Ming Zhang, and Jiawei Han. Distant meta-path similarities for text-based heterogeneous information networks. In *CIKM*.

[56] Julie Weeds, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, 1015.

[57] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. 2012. Probase: A probabilistic taxonomy for text understanding. In *SIGMOD*.

[58] Josuke Yamane, Tomoya Takatani, Hitoshi Yamada, Makoto Miwa, and Yutaka Sasaki. 2016. Distributional hypernym generation by jointly learning clusters and projections. In *COLING*.

[59] Carl Yang, Yichen Feng, Pan Li, Yu Shi, and Jiawei Han. 2018. Meta-graph based hin spectral embedding: Methods, analyses, and insights. In *ICDM*. 657–666.

[60] Carl Yang, Mengxiong Liu, Frank He, Xikun Zhang, Jian Peng, and Jiawei Han. 2018. Similarity Modeling on Heterogeneous Networks via Automatic Path Discovery. In *ECML-PKDD*. 37–54.

[61] Wenpeng Yin and Dan Roth. 2018. Term Definitions Help Hypernymy Detection. In **SEM@NAACL-HLT*.

[62] Chao Zhang, Fangbo Tao, Xiusi Chen, Jiaming Shen, Meng Jiang, Brian M. Sadler, Michelle T. Vanni, and Jiawei Han. 2018. TaxoGen: Constructing Topical Concept Taxonomy by Adaptive Term Embedding and Clustering. In *KDD*.

[63] Chao Zhang, Guangyu Zhou, Quan Yuan, Honglei Zhuang, Yu Zheng, Lance M. Kaplan, Shaowen Wang, and Jiawei Han. 2016. GeoBurst: Real-Time Local Event Detection in Geo-Tagged Tweet Streams. In *SIGIR*.

[64] Maayan Zhitomirsky-Geffet and Ido Dagan. 2005. The Distributional Inclusion Hypotheses and Lexical Entailment. In *ACL*.

[65] Maayan Zhitomirsky-Geffet and Ido Dagan. 2009. Bootstrapping distributional feature vector quality. *Computational linguistics* 35, 3 (2009), 435–461.

[66] Honglei Zhuang, Jing Zhang, George Brova, Jie Tang, Hasan Cam, Xifeng Yan, and Jiawei Han. 2014. Mining query-based subnetwork outliers in heterogeneous information networks. In *ICDM*.