Constructing Topical Hierarchies in Heterogeneous Information Networks

Chi Wang[‡], Marina Danilevsky[‡], Jialu Liu[‡], Nihit Desai[‡], Heng Ji[†], Jiawei Han[‡]

[‡]University of Illinois at Urbana-Champaign, USA [†]Rensselaer Polytechnic Institute

{chiwang1,danilev1,jliu64,nhdesai2,hanj}@illinois.edu, {jih}@rpi.edu

***Abstract*—A digital data collection (*e.g.*, scientific publications, enterprise reports, news, and social media) can often be modeled as a heterogeneous information network, linking text with multiple types of entities. Constructing high-quality concept hierarchies that can represent topics at multiple granularities benefits tasks such as search, information browsing, and pattern mining. In this work we present an algorithm for recursively constructing multi-typed topical hierarchies. Contrary to traditional text-based topic modeling, our approach handles both textual phrases and multiple types of entities by a newly designed clustering and ranking algorithm for heterogeneous network data, as well as mining and ranking topical patterns of different types. Our experiments on datasets from two different domains demonstrate that our algorithm yields high quality, multi-typed topical hierarchies.**

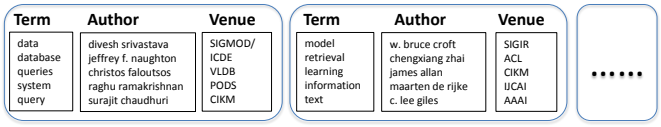
I. Introduction

In the real world there are many examples of collections of interconnected multi-typed objects, which form heterogeneous information networks (HINs). In order to facilitate tasks such as efficient search, mining and summarization of heterogeneous networked data, it is very valuable to discover and organize the concepts present in a dataset into a multi-typed topical hierarchy. Such a construction allows a user to perform more meaningful analysis of the terminology, people, places, and other network entities, which are organized into topics and subtopics at different levels of granularity.

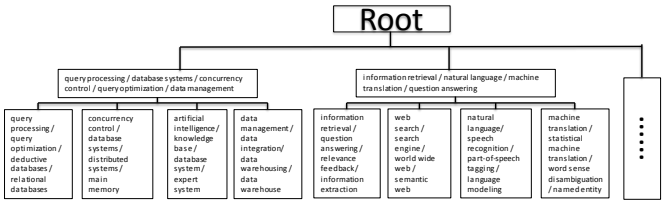
A variety of existing work is devoted to constructing topical or concept hierarchies from text data. However, few approaches utilize link information from heterogeneous entities that may be present in the data. Conversely, existing methods for heterogeneous network analysis and topic modeling have demonstrated that multiple types of linked entities improve the quality of topic discovery (*e.g.*, NetClus [1]), but these methods are not designed for finding hierarchical structures (See Figure 1a for an example output of NetClus). Therefore, there is no existing method that is able to construct a multi-typed topical hierarchy from a heteregeneous network.

In this study, we develop a method that makes use of both textual information and heterogeneous linked entities to automatically construct multi-typed topical hierarchies. The main contributions of this work are:

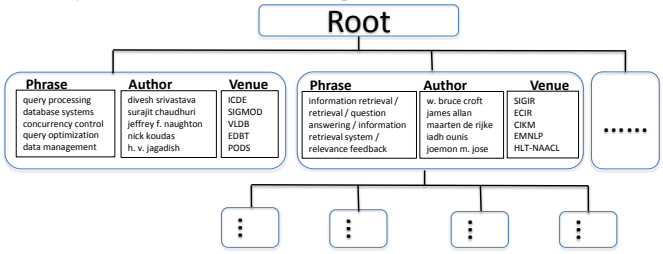
• We recursively construct a topical hierarchy where each topic is represented by ranked lists of phrases and entities of different types. We go beyond the topical hierarchies that are constructed by analyzing textual information alone (*e.g.*, Figure 1b), and enrich the topic representation with ranked lists of heterogeneous entities, which provides additional informative context for each topic in the hierarchy (shown in Figure 1c).



(a) NetClus [1] – clusters of heterogeneous entities. Each rounded rectangle represents one cluster, containing a ranked list of unigrams and two ranked lists of entities



(b) CATHY [2] – topical hierarchy of text only. Each node in the hierarchy contains a ranked list of phrases



(c) CATHYHIN – topical hierarchy of heterogeneous entities. Each node has a ranked list of phrases and two ranked entity pattern lists

Fig. 1: Sample output from three methods run on a computer science publication network with term, author, and venue attributes

• We propose a mutually enhancing clustering and ranking method for recursively generating subtopics from each topic in the hierarchy. Our approach retains the benefits of NetClus, a recently developed technique for analyzing heterogeneous networks, but is far more robust and well-suited to the task of topical hierarchy construction. Our unified general model is not confined to a particular network schema, and incorporates an inference algorithm that is guaranteed to converge.

• We develop an extension to our method which is able to automatically determine the importance of different types of entity links. We allow the importance of links to vary at differ-

ent levels of the topical hierarchy, since different information may be more or less useful at a particular granularity.

II. Related Work

A. Topical hierarchy construction

Topical hierarchies, concept hierarchies, ontologies, *etc.*, provide a hierarchical organization of data at different levels of granularity, and have many important applications, such as in web search and browsing tasks [3]. Although there has been a substantial amount of research on ontology learning from text, it remains a challenging problem (see [4] for a recent survey). The learning techniques can be broadly categorized as statistics-based or linguistic-based. Many studies are devoted to mining subsumption ('is-a') relationships [5], either by using lexico-syntactic patterns (*e.g.*, 'x is a y') [6], [7] or statistics-based approaches [8], [9]. Chuang and Chien [10] and Liu *et al.* [11] generate taxonomies of given keyword phrases by supplementing hierarchical clustering techniques with knowledge bases and search engine results.

With respect to input and output, our definition of the construction of topical hierarchy largely follows our previous work Wang *et al.* [2]. We proposed CATHY, a statistics-based technique which constructs a topical hierarchy without resorting to external knowledge resources such as WordNet or Wikipedia. However, CATHY hierarchy is constructed using only text information, while our CATHYHIN approach works with a heterogeneous network and discovers multi-typed topical entities.

B. Mining topics in heterogeneous networks

Basic topic modeling techniques such as PLSA (probabilistic latent semantic analysis) [12] and LDA (latent dirichlet allocation) [13] take documents as input, and output word distributions for each topic. Recently, researchers have studied how to mine topics when documents have additional links to multiple typed entities [1], [14], [15], [16], [17], [18]. These approaches make use of multiple typed links in different ways. iTopicModel [14] and TMBP–Regu [15] use links to regularize the topic distribution so that linked documents or entities have similar topic distributions. Chen *et al.* [16] and Kim *et al.* [17] extend LDA to use entities as additional sources of topic choices for each document. Tang *et al.* [18] argue that this kind of extension has a problem of 'competition for words' among different sources when the text is sparse. They propose to aggregate documents linked to a common entity as a pseudo document, and regularize the topic distributions inferred from different aggregation views to reach a consensus.

Nearly all of these studies still model topics as distributions over words, though they use linked entity information to help with topic inference in various ways. NetClus [1] takes a different approach by simultaneously clustering and ranking terms as well as linked entities in a heterogeneous network. It is therefore the only aforementioned approach which may be used to recursively construct heterogeneous topical hierarchies (with some slight modification). We therefore examine adapting NetClus to this task, and describe the limitations of this construciton, which are overcome by our method.

C. NetClus

As illustrated in Figure 2, the input to the NetClus algorithm is a heterogeneous network of star-schema. The example network has one central object type—the star object—and four types of attribute objects (where the type of an object is denoted by its shape and color family). Only links between a star object and an attribute object are allowed. For example, a collection of papers may be transformed into a star schema where each paper is a star object, and attributes such as authors, venues, and terms are attribute objects.

NetClus performs hard clustering on the star objects, and the induced network clusters consist of star objects and their linked attribute objects. Thus, an attribute object may belong to multiple clusters, but each star object is assigned to precisely one cluster. Next, the attribute objects within each subnetwork cluster are ranked via a PageRank-like algorithm, which is based on the structure of the cluster. A generative model then uses the ranking information to infer a cluster distribution for each star object. The cluster memberships of the star objects are then adjusted using a k-means algorithm, and the ranks of attribute objects are re-calculated. Thus, the NetClus algorithm iterates over clustering the star objects based on their inferred membership distribution (as calculated by a generative model based on the existing ranking information), and re-ranking the attribute objects within each newly defined network cluster. The heterogeneous nature of the attribute objects is respected during the ranking step, as only objects of the same type are ranked together, as shown in Figure 2.

The iterative clustering and ranking steps of NetClus thus mutually enhance each other. The clustering step provides a context for the ranking calculations, since the ranks of the attribute objects should vary among different clusters (*e.g.*, different areas of computer science). The ranking step in turn improves the quality of found clusters, since highly ranked objects should serve as stronger indicators of cluster membership for their linked star objects.

NetClus can be naturally extended for topical hierarchy construction: after each network is clustered, each of the induced subnetworks are then used as new input, and may thus be recursively clustered and ranked. However, several properties of NetClus render it undesirable for the task of topical hierarchy construction:

1. Topics are represented by ranked lists of terms, and other individual attribute objects. For topics of fine granularity in the hierarchy, this representation may be hard to interpret because single terms and entities may be ambiguous.

2. NetClus assumes a star schema, which hinders its application to more general information networks.

3. NetClus hard clusters star objects, which are usually documents. However, a document is often related to a mixture of topics, especially in the lower levels of a hierarchy. The forced hard clustering can thus result in lost information, as

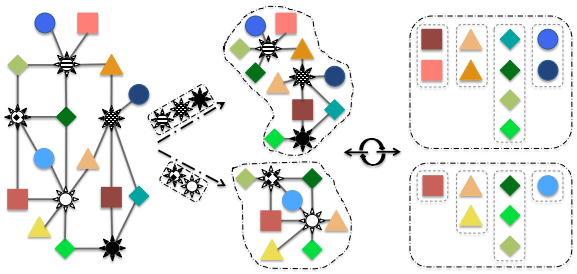


Fig. 2: An illustration of the NetClus framework. **(L)** NetClus analyses a star schema network, where every link is between a central object and an attribute object of some type (central objects are denoted by stars; attribute objects of the same type are represented by the same shape and color family, with individual objects differentiated by hue). **(M)** The star objects are partitioned into clusters so that each star appears in exactly one cluster. **(R)** Attribute objects (which may appear in multiple clusters) are ranked within each cluster, grouped by type. NetClus iterates over these clustering **(M)** and ranking **(R)** steps, as denoted by the two-way circular arrow symbol

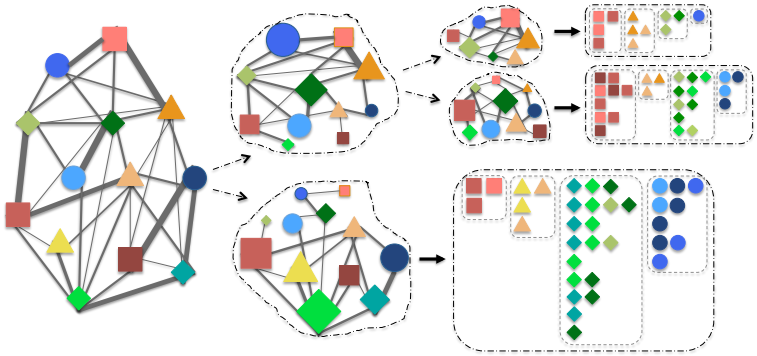


Fig. 3: An illustration of the CATHYHIN framework. **(L)** Step 1: CATHYHIN analyses a node-typed and edge-weighted network, with no central star objects. **(M)** Step 2: A unified generative model is used to partition the edge weights into clusters and rank single nodes in each cluster (here, node rank within each node type is represented by variations in node size). **(R bottom)** Step 3: Patterns of nodes are ranked within each cluster, grouped by type. **(R top)** Step 4: Each cluster is also an edge-weighted network, and is therefore recursively analyzed. The final output is a hierarchy, where the patterns of nodes of each cluster have a ranking within that cluster, grouped by type.

relevant documents fail to appear in relevant subtopics, further decreasing the hierarchy's quality.

4. The iterative algorithm used by NetClus is not guaranteed to converge. The deeper into the hierarchy, the more severe this problem becomes because the output of one level will be input of the next level of the constructed hierarchy.

III. CATHYHIN Framework

This section describes our framework CATHYHIN (shown in Figure 3), which incorporates the two positive characteristics of NetClus: the utilizing of heterogeneous link types, and the mutually enhancing clustering and ranking steps, while overcoming the disadvantages discussed in Section II-C.

Definition 1 (Heterogeneous Topical Hierarchy): A heterogeneous topical hierarchy is defined as a tree $\mathcal{T}$ in which each node is a topic. The root topic is denoted as o. Every non-root topic t with parent topic $Par(t)$ is represented by m ranked lists of patterns $L_1, \ldots, L_m$ where $L_x = \{P_i^{x,t}\}$ is the sequence of patterns for type x in topic t. The subtopics of every non-leaf topic t in the tree are its children $C^t = \{z \in \mathcal{T}, Par(z) = t\}$. A pattern can appear in multiple topics, though it will have a different ranking score in each topic.

This definition of the heterogeneous topical hierarchy addresses the first aforementioned criticism of NetClus by representing each topic as multiple lists of ranked patterns, where each list contains *patterns* of objects, rather than individual objects (*e.g.*, phrases rather than unigrams). For instance, the topics in Figure 1c each contain 3 lists of patterns.

Our approach does not restrict the network schema, and does not perform hard clustering for any objects. We discover topics by hierarchically soft clustering the links, so that any node may be assigned to multiple topics and subtopics. This removes the restrictions outlined in criticisms 2 and 3 of NetClus. We only require a collection of some kind of information chunks, such as documents, so that each chunk contains multiple objects and we can mine frequent patterns from these chunks.

Formally, every topic node t in the topical hierarchy is associated with an edge-weighted network $G^t = (\{V_x^t\}, \{E_{x,y}^t\})$, where V_x^t is the set of type-x nodes in topic t, and $E_{x,y}^t$ is the set of link weights between type x and type y nodes (x and y may be identical) in topic t. $e_{i,j}^{x,y,t} \in E_{x,y}^t$ represents the weight of the link between node v_i^x of type x and node v_j^y of type y. For every non-root node $t \neq o$, we construct a subnetwork G^t by clustering the network $G^{Par(t)}$ of its parent $Par(t)$. G^t inherits the nodes from $G^{Par(t)}$, but contains only the fraction of the original link weights that belongs to the particular subtopic t. Figure 3 visualizes the weight of each link in each network and subnetwork by line thickness (disconnected nodes and links with weight 0 are omitted).

If the original network naturally has a star schema, but the star type is not included in the final topic representation (*e.g.*, the document), we can construct a 'collapsed' network by connecting every pair of attribute objects which are linked to the same star object. In the derived network, the link weight $e_{i,j}^{x,y,t}$ between two nodes v_i^x and v_j^y is therefore equal to the number of common neighbors they share in the original star-schema network.

Our framework employs a unified generative model for recursive network clustering and subtopic discovery. The model seamlessly integrates mutually enhanced ranking and clustering while guaranteeing convergence for the inference algorithm, thus addressing the final critique of NetClus.

Our framework generates a heterogeneous topical hierarchy in a top-down, recursive way:

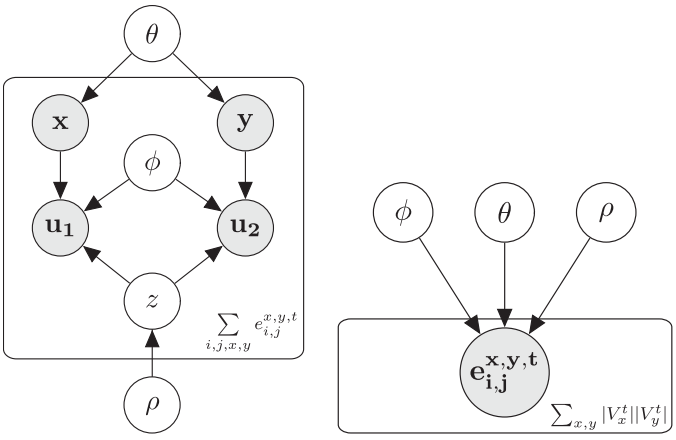
Step 1. Construct the edge-weighted network G^o. Set $t = o$.

Step 2. For a topic t, cluster the network G^t into subtopic subnetworks $G^z, z \in C^t$ using a generative model.

Step 3. For each subtopic $z \in C^t$, extract candidate topical patterns within each topic, and rank the patterns using a unified

TABLE I: Notations used in our model

Symbol	Description
G^t	the HIN associated with topic t
V_x^t	the set of nodes of type x in topic t
$E_{x,y}^t$	the set of non-zero link weights of type (x, y) in topic t
$Par(t)$	the parent topic of topic t
C^t	the set of child topics of topic t
z	child topic index of topic t
m	the number of node types
$n_{x,y}$	the total number of links between type-x and type-y nodes
v_i^x	the i-th node of type x
$e_{i,j}^{x,y,z}$	the link weight between v_i^x and v_j^y in topic z
$\phi^{x,z}$	the distribution over type-x nodes in topic z
ϕ^x	the overall distribution over type-x nodes
ρ_z	the total link weight in topic z
θ	the distribution over link type (x, y)
$\alpha_{x,y}$	the importance of link type (x, y)



(a) The generative process of the 'unit-weight' links

(b) The 'collapsed' generative process of the link weights

Fig. 4: Two graphical representation of our generative model for links in a topic t. The models are asymptotically equivalent.

ranking function. Patterns of different lengths are directly compared, yielding an integrated ranking.

Step 4. Recursively apply Steps 2 - 3 to each subtopic $z \in C^t$ to construct the hierarchy in a top-down fashion.

We describe steps 2 and 3 in the following subsections.

A. Topic Discovery in Heterogeneous Information Networks

Given a topic t and the associated network G^t, we discover subtopics by performing clustering and ranking with the network. We now describe our unified generative model and present an inference algorithm with a convergence guarantee. We further extend our approach to allow different link types to play different degrees of importance in the model (allowing the model to, for example, decide to rely more on term co-occurrence information than on co-author links).

The generative model

We first introduce the basic generative model, which considers all link types to be equally important. For a given topic t, we assume C^t contains k child topics, denoted by $z = 1 \dots k$. The value of k can be either specified by users or chosen using a model selection criterion.

In general, the network G^t contains m node types and $\frac{m(m+1)}{2}$ link types.[1] Similar to NetClus, we assume each node type x has a ranking distribution $\phi^{x,z}$ in each subtopic $z \in C^t$, such that $\phi_i^{x,z}$ is the importance of node v_i^x in topic z, subject to $\sum_i \phi_i^{x,z} = 1$. Each node type x also has a ranking distribution $\phi^{x,0}$ for the background topic, as well as an overall distribution ϕ^x, where ϕ_i^x is proportional to the degree of node v_i^x. In contrast to NetClus, we softly partition the link weights in G^t into subtopics. We model the generation of links so that we can simultaneously infer the partition of link weights (clustering) and the node distribution (ranking) for each topic.

[1]We assume the network is undirected, although our model can be easily extended to directed cases.

To derive our model, we first assume the links between any two nodes can be decomposed into one or multiple unit-weight links (*e.g.*, a link with weight 2 can be seen as a summation of two unit-weight links). Later we will discuss the case where the link weight is not an integer. Each unit-weight link has a topic label, which is either a subtopic $z \in C^t$, or a dummy label 0, implying the link is generated by a background topic and should not be attributed to any topic in C^t.

The generative process for a topic-z link, $z \in C^t$ (or background topic link, resp.) with unit weight is as follows:

1) Generate the link type (x, y) according to a multinomial distribution θ.
2) Generate the first end node u_1 from the type-x ranking distribution $\phi^{x,z}$ (or $\phi^{x,0}$, resp.).
3) Generate the second end node u_2 from the type-y ranking distribution $\phi^{y,z}$ (or ϕ^y, resp.).

Note that when generating a background topic link, the two nodes i and j are not symmetric, so that we attribute half of it to $i \to j$ and the other half to $j \to i$. The first end node is a background node, and can have a background topic link with any other nodes based simply on node degree, irrespective of any topic. Highly ranked nodes in the background topic tend to have a link distribution over all nodes that is similar to their overall degree distribution. See Figure 4a for a graphical representation of the model.

With these generative assumptions for each unit-weight link, we can derive the distribution of link weight for any two nodes (v_i^x, v_j^y). If we repeat the generation of topic-z unit-weight links for ρ_z iterations, then the process of generating a unit-weight topic-z link between v_i^x and v_j^y can be modeled as a Bernoulli trial with success probability $\theta_{x,y}\phi_i^{x,z}\phi_j^{y,z}$. When ρ_z is large, the total number of successes $e_{i,j}^{x,y,z}$ asymptotically follows a Poisson distribution $Pois\left(\rho_z\theta_{x,y}\phi_i^{x,z}\phi_j^{y,z}\right)$. Similarly, the total number of background topic links $e_{i,j}^{x,y,0}$ asymptotically follows a Poisson

distribution $Pois\left(\rho_0\theta_{x,y}\frac{\phi_i^{x,0}\phi_j^y+\phi_i^x\phi_j^{y,0}}{2}\right)$.

One important implication due to the *additive* property of Poisson distribution is:

$$e_{i,j}^{x,y,t}=\sum_{z=0}^{k}e_{i,j}^{x,y,z}\sim Poisson\left(\theta_{x,y}s_{i,j}^{x,y,t}\right) \quad (1)$$

where $s_{i,j}^{x,y,t}=\sum_{z=1}^{k}\rho_z\phi_i^{x,z}\phi_j^{y,z}+\rho_0\frac{\phi_i^{x,0}\phi_j^y+\phi_i^x\phi_j^{y,0}}{2}$.

This leads to a 'collapsed' model as depicted in Figure 4b. Though we have so far assumed the link weight to be an integer, this collapsed model remains valid with non-integer link weights (due to Property 1, discussed later).

Given the model parameters, the probability of all observed links is:

$$p\left(\{e_{i,j}^{x,y,t}\}|\theta,\rho,\phi\right)=\prod_{v_i^x,v_j^y}\frac{\left(\theta_{x,y}s_{i,j}^{x,y,t}\right)^{e_{i,j}^{x,y,t}}\exp\left(-\theta_{x,y}s_{i,j}^{x,y,t}\right)}{e_{i,j}^{x,y,t}!} \quad (2)$$

We learn the parameters by the *Maximum Likelihood* (ML) principle: find the parameter values that maximize the likelihood in Eq. (2). We use an Expectation-Maximization (EM) algorithm that can iteratively infer the model parameters.

E-step:

$$\hat{e}_{i,j}^{x,y,z} = \frac{e_{i,j}^{x,y,t}\rho_z\phi_i^{x,z}\phi_j^{y,z}}{\sum_{c=1}^{k}\rho_c\phi_i^{x,c}\phi_j^{y,c}+\frac{\rho_0}{2}(\phi_i^{x,0}\phi_j^y+\phi_i^x\phi_j^{y,0})} \quad (3)$$

$$\hat{e}_{i\to j}^{x,y,0} = \frac{e_{i,j}^{x,y,t}\rho_0\phi_i^{x,0}\phi_j^y}{2\sum_{c=1}^{k}\rho_c\phi_i^{x,c}\phi_j^{y,c}+\rho_0(\phi_i^{x,0}\phi_j^y+\phi_i^x\phi_j^{y,0})} \quad (4)$$

M-step:

$$\rho_z=\textstyle\sum_{i,j,x,y}\hat{e}_{i,j}^{x,y,z},\quad \theta_{x,y}=\frac{\sum_{i,j}e_{i,j}^{x,y,t}}{\sum_{i,j,x,y}e_{i,j}^{x,y,t}} \quad (5)$$

$$\phi_i^{x,z}=\frac{\sum_{j,y}\hat{e}_{i,j}^{x,y,z}}{\sum_{u,j,y}\hat{e}_{u,j}^{x,y,z}},\quad \phi_i^{x,0}=\frac{\sum_{j,y}\hat{e}_{i\to j}^{x,y,0}}{\sum_{u,j,y}\hat{e}_{u\to j}^{x,y,0}} \quad (6)$$

We update $\hat{e},\phi,\rho$ in each iteration ($\theta_{x,y}$ is a constant). In the E-step, we perform the clustering by estimating $\hat{e}$. In the M-step, we estimate the ranking distribution ϕ. Like other EM algorithms, the solution converges to a local maximum and the result may vary with different initializations. The EM algorithm can be run multiple times with random initializations to find the solution with the best likelihood.

The subnetwork for topic z is naturally extracted from the estimated $\hat{e}$ (expected link weight attributed to each topic). For efficiency purposes, we remove links whose weight is less than 1, and then filter out all resulting isolated nodes. We can then recursively apply the same generative model to the constructed subnetworks until the desired hierarchy is constructed.

Learning link type weights

The generative model described above does not differentiate between the importance of different link types. However, we may wish to discover topics that are biased towards certain types of links, and the bias may vary at different levels of the hierarchy. For example, in the computer science domain, the links between venues and other entities may be more important indicators than other link types in the top level of the hierarchy; however, these same links may be less useful for discovering subareas in the lower levels (*e.g.*, authors working in different subareas may publish in the same venue).

We therefore extend our model to capture the importance of different link types. We introduce a *link type weight* $\alpha_{x,y}>0$ for each link type (x,y). We use these weights to scale a link's observed weight up or down, so that a unit-weight link of type (x,y) in the original network will have a *scaled* weight $\alpha_{x,y}$. Thus, a link of type (x,y) is valued more when $\alpha_{x,y}>1$, less when $0<\alpha_{x,y}<1$, and becomes negligible as $\alpha_{x,y}$ approaches 0.

When the link type weights $\alpha_{x,y}$ are specified for our model, the EM inference algorithm is unchanged, with the exception that all the $e_{i,j}^{x,y,t}$ in E-step should be replaced by $\alpha_{x,y}e_{i,j}^{x,y,t}$. When all $\alpha_{x,y}$'s are equal, the weight-learning model reduces to the basic model. Most of the time, the weights of the link types will not be specified explicitly by users, and must therefore be learned from the data.

We first note an important property of our model, justifying our previous claim that link weights need not be integers.

Property 1 (Scale-invariant): The EM solution is invariant to a constant scaleup of all the link weights.

Due to the scale-invariant property of the link weights, we can assume that *w.l.o.g.*, the product of all the non-zero link weights remains invariant before and after scaling:

$$\prod_{e_{i,j}^{x,y,t}>0}e_{i,j}^{x,y,t}=\prod_{e_{i,j}^{x,y,t}>0}\alpha_{x,y}e_{i,j}^{x,y,t} \quad (7)$$

which reduces to $\prod_{x,y}\alpha_{x,y}^{n_{x,y}}=1$, where $n_{x,y}=|E_{x,y}^t|$ is the number of non-zero links with type (x,y). With this contraint, we maximize the likelihood $p(\{e_{i,j}^{x,y,t}\}|\theta,\rho,\phi,\alpha)$:

$$\max\prod_{v_i^x,v_j^y}\frac{(\theta_{x,y}s_{i,j}^{x,y,t})^{\alpha_{x,y}e_{i,j}^{x,y,t}}\exp(-\theta_{x,y}s_{i,j}^{x,y,t})}{(\alpha_{x,y}e_{i,j}^{x,y,t})!} \quad (8)$$

$$s.t.\prod_{x,y}\alpha_{x,y}^{n_{x,y}}=1,\alpha_{x,y}>0 \quad (9)$$

With Stirling's approximation $n!\sim(\frac{n}{e})^n\sqrt{2\pi n}$, we transform the log likelihood:

$$\max\sum_{v_i^x,v_j^y}\Big(\alpha_{x,y}e_{i,j}^{x,y,t}\log(\theta_{x,y}s_{i,j}^{x,y,t})-\theta_{x,y}s_{i,j}^{x,y,t} \quad (10)$$

$$-\alpha_{x,y}e_{i,j}^{x,y,t}[\log(\alpha_{x,y}e_{i,j}^{x,y,t})-1]-\frac{1}{2}\log(\alpha_{x,y}e_{i,j}^{x,y,t})\Big)$$

$$s.t.\sum_{x,y}n_{x,y}\log\alpha_{x,y}=0 \quad (11)$$

Using the Langrange multiplier method, we can find the

optimal value for α when the other parameters are fixed:

$$\alpha_{x,y} = \frac{\left[\prod_{x,y}\left(\frac{1}{n_{x,y}}\sum_{i,j} e_{i,j}^{x,y,t}\log\frac{e_{i,j}^{x,y,t}}{s_{i,j}^{x,y,t}}\right)^{n_{x,y}}\right]^{\frac{1}{\sum_{x,y} n_{x,y}}}}{\frac{1}{n_{x,y}}\sum_{i,j} e_{i,j}^{x,y,t}\log\frac{e_{i,j}^{x,y,t}}{s_{i,j}^{x,y,t}}} \tag{12}$$

With some transformation of the denominator:

$$\begin{aligned}\beta_{x,y} &= n_{x,y}\sum_{i,j} e_{i,j}^{x,y,t}\log\frac{e_{i,j}^{x,y,t}}{s_{i,j}^{x,y,t}} \qquad (13)\\ &= \frac{\sum_{i,j} e_{i,j}^{x,y,t}}{n_{x,y}}\sum_{i,j}\frac{e_{i,j}^{x,y,t}}{\sum_{i,j} e_{i,j}^{x,y,t}}\log\frac{e_{i,j}^{x,y,t}/\sum_{i,j} e_{i,j}^{x,y,t}}{s_{i,j}^{x,y,t}/\sum_{i,j} e_{i,j}^{x,y,t}}\end{aligned}$$

we can see more clearly that the link type weight is negatively correlated with two factors: the average link weight and the KL-divergence of the expected link weight distribution to the observed link weight distribution. The first factor is used to balance the scale of link weights of different types (*e.g.*, a type-1 link always has X times greater weight than a type-2 link). The second factor measures the importance of a link type in the model. The more the prediction diverges from the observation, the worse the quality of a link type.

So we have the following iterative algorithm for optimizing the joint likelihood:

1) Initialize all the parameters.
2) Fixing α, update ρ, θ, ϕ using EM equations (3)-(6).
3) Fixing ρ, θ, ϕ, update α using Eq. (12).
4) Repeat steps 2) and 3) until the likelihood converges.

In each iteration, the time complexity is $O(\sum_{x,y} n_{x,y})$, *i.e.*, linear to the total number of non-zero links. The likelihood is guaranteed to converge to a local optimum. Once again, a random initialization strategy can be employed to choose a solution with the best local optimum.

B. Topical Pattern Mining and Ranking

Having discovered the topics using our generative model, we can now identify the most representative topical patterns for each topic. This is done in two stages: topical pattern mining and ranking the mined patterns.

Pattern mining in each topic

A pattern P^x of type x is a set of type-x nodes: $P^x = \{v_i^x\}$. For example, a pattern of a 'term' type is a set of unigrams that make up a phrase, such as $\{support, vector, machine\}$ (or 'support vector machine' for simpler notation). A more general definition of a pattern can involve mixed node types within one pattern, but is beyond the scope of this paper.

A pattern P that is regarded to be representative for a topic t must first and foremost be frequent in the topic. The frequency of a pattern $f(P)$ is the number of documents (or other meaningful information chunks) that contain all the nodes in the pattern (or the number of star objects that are linked to all the nodes). The pattern must also have sufficiently high *topical frequency* in topic t.

Definition 2 (Topical Frequency): The topical frequency $f_t(P)$ of a pattern is the number of times the pattern is attributed to topic t. For the root node o, $f_o(P) = f(P)$. For each topic node with subtopics C^t, $f_t(P) = \sum_{z\in C^t} f_z(P)$ (i.e., topical frequency is the sum of sub-topical frequencies.)

TABLE II: Hypothetical example of estimating topical frequency. The topics are assumed to be inferred as machine learning, database, data mining, and information retrieval from the data

Pattern	ML	DB	DM	IR	Total
support vector machines	85	0	0	0	85
query processing	0	212	27	12	251
Hui Xiong	0	0	66	6	72
SIGIR conference	444	378	303	1,117	2,242

Table II illustrates a hypothetical example of estimating topical frequency for patterns of various types (term, author, and venue) in a computer science topic that has 4 subtopics.

We estimate the topical frequency of a pattern based on two assumptions: i) For a type-x topic-t pattern of length n, each of the n nodes is generated with the distribution $\phi^{x,t}$, and ii) the total number of topic-t phrases of length n is proportional to ρ_t.

$$f_t(P^x) = f_{Par(t)}(P^x)\frac{\rho_t\prod_{v_i^x\in P^x}\phi_i^{x,t}}{\sum_{z\in C^{Par(t)}}\rho_z\prod_{v_i^x\in P^x}\phi_i^{x,z}} \tag{14}$$

Both ϕ and ρ are learned from the generative model as described in Section III-A.

To extract topical frequent patterns, all frequent patterns can first be mined using a pattern mining algorithm such as FP-growth [19], and then filtered given some minimal topical frequency threshold $minsup$.

Pattern ranking in each topic

There are four criteria for judging the quality of a pattern (similar criteria are proposed for ranking phrases in [2], and also apply to other types of patterns).

• **Frequency** – A representative pattern for a topic should have sufficently high topical frequency.

• **Exclusiveness** – A pattern is exclusive to a topic if it is only frequent in this topic and not frequent in other topics. *Example: 'query processing' is more exclusive than 'query' in the Databases topic.*

• **Cohesiveness** – A group of entities should be combined together as a pattern if they co-occur significantly more often than the expected co-occurrence frequency given the chances of occurring independently. *Example: 'active learning' is a more cohesive pattern than 'learning classification' in the Machine Learning topic.*

• **Completeness** – A pattern is not complete if it rarely occurs without the presence of a longer pattern. *Example: 'support vector machines' is a complete pattern, whereas 'vector machines' is not because 'vector machines' is almost always accompanied by 'support' in occurrence.*

The pattern ranking function should take these criteria into consideration. The ranking function must also be able to directly compare patterns of mixed lengths, such as 'classification,' 'decision trees,' and 'support vector machines.'

Let N_t be the number of documents that contain at least one frequent topic-t pattern, T a subset of $C^{Par(t)}$ that contains t, and N_T the number of documents that contain at least one frequent topic-z pattern for some topic $z \in T$. We use the following ranking function that satisfies all these requirements [2]:

$$r^t(P) = \begin{cases} 0, \text{if } \exists P' \supsetneq P, f_t(P') \geq \gamma f_t(P) \\ p(P|t)\left(\log \frac{p(P|t)}{\max_T p(P|T)} + \omega \log \frac{p(P|t)}{p_{indep}(P|t)}\right) & \text{o.w.} \end{cases} \quad (15)$$

where $p(P|t) = \frac{f_t(P)}{N_t}$ is the occurrence probability of a pattern P, measuring frequency; $p_{indep}(P|t) = \prod_{v \in P} \frac{f_t(v)}{N_t}$ is the probability of independently seeing every node in pattern P, measuring exclusiveness; and $p(P|T) = \frac{\sum_{t \in T} f_t(P)}{N_T}$ is the probability of phrase P conditioned on a mixture T of t and other sibling topics, measuring cohesiveness. Incomplete patterns are filtered if there exists a superpattern P' that has sufficiently high topical frequency compared to P. $\gamma \in [0, 1]$ is a parameter that controls the strictness of the completness criterion, where a larger value of γ deems more phrases to be complete. Complete phrases are ranked according to a combination of the other three criteria. Frequency plays the most important role. The weight between exclusiveness and cohesiveness is controled by a parameter $\omega \in [0, +\infty)$, with larger values of ω biasing the ranking more heavily towards cohesiveness. Due to space limitation, we refer to [2] for more detailed discussion of this ranking function.

IV. Experiments

We evaluate the performance of our proposed method on two datasets (see Table V for summary statistics of the constructed networks):

• **DBLP**. We collected 33,313 recently published computer science papers from DBLP[2]. We constructed a heterogeneous network with three node types: term (from paper title), author and venue, and 5 link types: term-term, term-author, term-venue, author-author and author-venue.[3]

• **NEWS**. We crawled 43,168 news articles on 16 top stories from Google News,[4] and ran an information extraction algorithm [20] to extract entities. We constructed a heterogeneous network with three node types: term (from article title), person and location, and 6 link types: term-term, term-person, term-location, person-person, person-location and location-location.

Our recursive framework relies on two key steps: subtopic discovery and topical pattern mining. The major contribution of this paper is the subtopic discovery step. Hence, our evaluation is twofold: i) we evaluate the efficacy of subtopic discovery given a topic and its associated heterogeneous network; and ii) we perform several 'intruder detection' tasks to evaluate the quality of the constructed hierarchy based on human judgment.

A. Efficacy of Subtopic Discovery

We first present a set of experiments designed to evaluate just the subtopic discovery step (Step 2 in Section III).

Evaluation Measure. We extend the pointwise mutual information (PMI) metric in order to measure the quality of our multi-typed topics. The metric of pointwise mutual information PMI has been proposed in [21] as a way of measuring the semantic coherence of topics. It is generally preferred over other quantitative metrics such as perplexity or the likelihood of held-out data [18]. In order to measure the quality of our multi-typed topics, we extend the definition of PMI as follows:

For each topic, PMI calculates the average relatedness of each pair of the words ranked at top-K:

$$PMI(\mathbf{w}, \mathbf{w}) = \frac{2}{K(K-1)} \sum_{1 \leq i < j \leq K} \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} \quad (16)$$

where $PMI \in [-\infty, \infty]$, and $\mathbf{w}$ are the top K most probable words of the topic. $PMI = 0$ implies that these words are independent; $PMI > 0$ (< 0) implies they are overall positively (negatively) correlated.

However, our multi-typed topic contains not only words, but also other types of entities. So we define *heterogeneous* pointwise mutual information as:

$$HPMI(\mathbf{v^x}, \mathbf{v^y}) = \begin{cases} \frac{2}{K(K-1)} \sum_{1 \leq i < j \leq K} \log \frac{p(v_i^x, v_j^y)}{p(v_i^x)p(v_j^y)} & x = y \\ \frac{1}{K^2} \sum_{1 \leq i,j \leq K} \log \frac{p(v_i^x, v_j^y)}{p(v_i^x)p(v_j^y)} & x \neq y \end{cases} \quad (17)$$

where $\mathbf{v^x}$ are the top K most probable type-x nodes in the given topic. When $x = y$, HPMI reduces to PMI. The HPMI-score for every link type (x, y) is calculated and averaged to obtain an overall score. We set $K = 20$ for all node types.[5]

Methods for Comparison:

• **CATHYHIN (equal weight)** – The weight for every link type is set to be 1.

• **CATHYHIN (learn weight)** – The weight of each link type is learned, as described in Section III-A. No parameters need hand tuning.

• **CATHYHIN (norm weight)** – The weight of each link type is explicitly set as: $\alpha_{x,y} = \frac{1}{\sum_{i,j} e_{i,j}^{x,y}}$. This is a heuristic normalization which forces the total weight of the links for each link type to be equal.

• **NetClus** – The current state-of-the-art clustering and ranking method for heterogeneous networks. We use the implementation in Deng *et al.* [15]. The smoothing parameter λ_S is tuned by a grid search in $[0, 1]$. Note that the link type weight learning method for CATHYHIN does not apply to NetClus because NetClus is not a single unified model.

• **TopK** – Select the top K nodes from each type to form a

[2]We chose papers published in 20 conferences related to the areas of Artificial Intelligence, Databases, Data Mining, Information Retrieval, Machine Learning, and Natural Language Processing from http://www.dblp.org/

[3]As a paper is always published in exactly one venue, there can naturally be no venue-venue links.

[4]The 16 topics chosen were: Bill Clinton, Boston Marathon, Earthquake, Egypt, Gaza, Iran, Israel, Joe Biden, Microsoft, Mitt Romney, Nuclear power, Steve Jobs, Sudan, Syria, Unemployment, US Crime.

[5]The one exception is venues, as there are only 20 venues in the DBLP dataset, so we set $K = 3$ in this case.

TABLE III: Heterogeneous pointwise mutual information in DBLP (20 Conferences and Database area)

DBLP (Database Area)	Term-Term	Term-Author	Author-Author	Term-Venue	Author-Venue	Overall
TopK	-0.5228	-0.1069	0.4545	0.0348	-0.3650	-0.0761
NetClus	-0.3962	0.0479	0.4337	0.0368	-0.2857	0.0260
CATHYHIN (equal weight)	0.0561	0.4799	0.6496	0.0722	-0.0033	0.3994
CATHYHIN (norm weight)	-0.1514	0.3816	**0.6971**	0.0408	**0.2464**	0.3196
CATHYHIN (learn weight)	**0.3027**	**0.6435**	0.5574	**0.1165**	0.1805	**0.5205**
DBLP (20 Conferences)	Term-Term	Term-Author	Author-Author	Term-Venue	Author-Venue	Overall
TopK	-0.4825	-0.0204	0.5466	-1.0051	-0.4208	-0.0903
NetClus	-0.1995	0.5186	0.5404	0.2851	1.2659	0.4045
CATHYHIN (equal weight)	0.2936	0.8812	0.6595	0.5191	1.0466	0.6949
CATHYHIN (norm weight)	0.1825	0.8674	**0.9476**	0.7472	1.3307	0.7601
CATHYHIN (learn weight)	**0.4964**	**1.0618**	0.7161	**1.1283**	**1.7511**	**0.9168**

TABLE IV: Heterogeneous pointwise mutual information in NEWS (16 topics collection and 4 topics subset)

NEWS (4 topics subset)	Term-Term	Term-Person	Person-Person	Term-Location	Person-Location	Location-Location	Overall
TopK	-0.2479	0.1671	0.0716	0.0787	0.2483	0.3632	0.1317
NetClus	0.1279	0.3835	0.2909	0.3240	0.4728	0.4271	0.3575
CATHYHIN (equal weight)	**1.0471**	0.7917	0.4902	0.8506	0.6821	0.6586	0.7610
CATHYHIN (norm weight)	0.7975	0.8825	**0.5553**	0.8682	**0.8077**	0.7346	0.8023
CATHYHIN (learn weight)	0.9935	**0.9354**	0.5142	**0.9784**	0.7389	**0.7645**	**0.8434**
NEWS (16 topics)	Term-Term	Term-Person	Person-Person	Term-Location	Person-Location	Location-Location	Overall
TopK	-1.7060	-0.8663	-0.8462	-1.0238	-0.5665	-0.4578	-0.8783
NetClus	-0.3847	0.0943	0.0313	-0.1114	0.1291	0.1376	-0.0274
CATHYHIN (equal weight)	0.7804	1.0170	0.8393	0.8354	0.9467	0.6382	0.8749
CATHYHIN (norm weight)	0.8579	**1.1143**	**0.9086**	0.8530	0.9624	**0.7143**	0.9284
CATHYHIN (learn weight)	**0.9234**	1.1109	0.7966	**0.9731**	**0.9718**	0.6965	**0.9500**

pseudo topic. This method serves as a baseline value for the proposed HPMI metric.

Experiment Setup. We discover the subtopics of four datasets:

• DBLP (20 conferences) – Aforementioned DBLP dataset.

• DBLP (database area) – A subset of the DBLP dataset consisting only of papers published in 5 Database conferences. By using this dataset, which roughly repesents a subtopic of the full DBLP dataset, we analyze the quality of discovered subtopics in a lower level of the hierarchy.

• NEWS (16 topics) – Aforementioned NEWS dataset.

• NEWS (4 topic subset) – A subset of the NEWS dataset limited to 4 topics, which center around different types of entities: Bill Clinton, Boston Marathon, Earthquake, Egypt.

Experiment Results. All the methods finish in 1.5 hours for these datasets. As seen in Tables III and IV, our generative model consistently posts a higher HPMI score than NetClus (and TopK) across all links types in every dataset. Although NetClus HPMI values are better than the TopK baseline, the improvement of our best performing method - CATHYHIN (learn weight) - over the TopK baseline are better than the improvement posted by NetClus by factors ranging from 2 to 5.8. Even the improvement over the TopK baseline of CATHYHIN (equal weight), which considers uniform link type weights, is better than the improvement posted by NetClus by factors ranging from 1.6 to 4.6.

CATHYHIN with learned link type weights consistently yields the highest overall HPMI scores, although CATHYHIN with normalized link type weights sometimes shows a slightly higher score for particular link types (*e.g.*, Author-Author for both DBLP datasets, and Person-Person for both NEWS datasets). CATHYHIN (norm weight) assigns a high weight to a link type whose total link weights were low in the originally constructed network, pushing the discovered subtopics to be more dependent on that link type. Normalizing the link type weights does improve CATHYHIN performance in many cases, as compared to using uniform link type weights. However, this heuristic determines the link type weight based solely on their link density. It can severely deteriorate the coherence of desne but valuable link types, such as Term-Term in both DBLP datasets, and rely too heavily on sparse but uninformative entities, such as Venues in the Database subtopic of the DBLP dataset.

We may conclude from these experiments that CATHYHIN's unified generative model consistently outperforms the state-of-the-art heterogeneous network analysis technique NetClus. In order to generate coherent, multi-typed topics at each level of a topical hierarchy, it is important to learn the optimal weights of different entity types, which depends on the link type density, the granularity of the topic to be partitioned, and the specific domain.

B. Topical Hierarchy Quality

Our second set of evaluations assesses the ability of our method to construct a hierarchy of multi-typed topics that human judgement deems to be high quality. We generate and analyze multi-typed topical hierarchies using the DBLP dataset

TABLE V: # Links in our datasets

DBLP *(# Nodes)*	**Term** (6,998)	**Author** (12,886)	**Venue** (20)
Term	693,132	900,201	104,577
Author	–	156,255	99,249
NEWS *(# Nodes)*	**Term** (13,129)	**Person** (4,555)	**Location** (3,845)
Term	686,007	386,565	506,526
Person	–	53,094	129,945
Location	–	–	85,047

TABLE VI: Results of Intruder Detection tasks (% correct intruders identified)

	DBLP				**NEWS**			
	Phrase	Venue	Author	Topic	Phrase	Location	Person	Topic
CATHYHIN	**0.83**	**0.83**	**1.0**	**1.0**	**0.65**	**0.70**	**0.80**	**0.90**
CATHYHIN$_1$	0.64	–	–	0.92	0.40	0.55	0.50	0.70
CATHY	0.72	–	–	0.92	0.58	–	–	0.65
CATHY$_1$	0.61	–	–	0.92	0.23	–	–	0.50
CATHY$_{\text{heur_HIN}}$	–	0.78	0.94	0.92	–	0.65	0.45	0.70
NetClus$_{\text{pattern}}$	0.33	0.78	0.89	0.58	0.23	0.20	0.55	0.45
NetClus$_{\text{pattern_1}}$	0.53	–	–	0.58	0.20	0.45	0.30	0.40
NetClus	0.19	0.78	0.83	0.83	0.15	0.35	0.25	0.45

(20 conferences) and the NEWS dataset (16 topics collection).

Experiment Setup. We adapt two tasks from Chang et al. [22], who were the first to explore human evaluation of topic models. Each task involves a set of questions asking humans to discover the 'intruder' object from several options. Three annotators manually completed each task, and their evaluations scores were pooled.

The first task is Phrase Intrusion, which evaluates how well the hierarchies are able to separate phrases in different topics. Each question consists of X ($X = 5$ in our experiments) phrases; $X - 1$ of them are randomly chosen from the top phrases of the same topic and the remaining phrase is randomly chosen from a sibling topic. The second task is Entity Intrusion, a variation that evaluates how well the hierarchies are able to separate entities present in the dataset in different topics. For each entity type, each question consists of X entity patterns; $X - 1$ of them are randomly chosen from the top patterns of the same topic and the remaining entity pattern is randomly chosen from a sibling topic. This task is constructed for each entity type in each dataset (Author and Venue in DBLP; Person and Location in NEWS). The third task is Topic Intrusion, which tests the quality of the parent-child relationships in the generated hierarchies. Each question consists of a parent topic t and X candidate child topics. $X-1$ of the child topics are actual children of t in the generated hierarchy, and the remaining child topic is not. Each topic is represented by its top 5 ranked patterns of each type - *e.g.*, for the NEWS dataset, the top 5 phrases, people, and locations are shown for each topic.

For each question, human annotators select the intruder phrase, entity, or subtopic. If they are unable to make a choice, or choose incorrectly, the question is marked as a failure.

Methods for Comparison:

• **CATHYHIN** – As defined in Section III

• **CATHYHIN$_1$** – The pattern length of text and every entity type is restricted to 1.

• **CATHY** – As defined in [2], the hierarchy is constructed only from textual information.

• **CATHY$_1$** – The phrase length is restricted to 1.

• **CATHY$_{\text{heuristic_HIN}}$** – Since neither CATHY nor CATHY$_1$ provides topical ranks for entities, we construct this method to have a comparison for the Entity Intrusion task. We use a heuristic entity ranking method based on the textual hierarchy generated by CATHY, and the original links in the network. An entity's rank for a given topic is a function of its frequency in the topic (estimated as the number of documents in that topic which are linked to the entity in the original network), and its exclusivity.

• **NetClus$_{\text{pattern}}$** – NetClus is used for subtopic discovery, followed by the topical mining and ranking method of CATHYHIN, as described in Section III-B (this can also be thought of CATHYHIN, where Step 2 is replaced by NetClus).

• **NetClus$_{\text{pattern_1}}$** – Equivalent to NetClus$_{\text{pattern}}$ with the pattern length of text and every entity type restricted to 1.

• **NetClus** – As defined in [1].

The pattern mining and ranking parameters for both CATHY and CATHYHIN are set to be $minsup = 5, \omega = \gamma = 0.5$ according to [2]. The optimal smoothing parameter for NetClus is $\lambda_S = 0.3$ and 0.7 in DBLP and NEWS respectively.

Table VI displays the results of the intruder detection tasks. For the Entity Intrusion task on the DBLP dataset, we restricted the entity pattern length to 1 in order to generate meaningful questions. This renders the methods CATHYHIN$_1$ and NetClus$_{\text{pattern_1}}$ equivalent to CATHYHIN and NetClus$_{\text{pattern}}$ respectively, so we omit the former methods from reporting.

Experiment Results. The Phrase Intrusion task performs much better when phrases are used rather than unigrams, for both CATHYHIN and CATHY, on both datasets. The NEWS dataset exhibits a stronger preference for phrases, as opposed to the DBLP dataset, which may be due to the fact that the terms in the NEWS dataset are more likely to be noisy and uninformative outside of their context, whereas the DBLP terms are more technical and therefore easier to interpret. This characteristic may also help explain why the performance of every method on DBLP data is consistently higher than on NEWS data. However, neither phrase mining and ranking nor unigram ranking can make up for poor performance during the topic discovery step, as seen in the three NetClus variations. Therefore, both phrase representation and high quality topics are necessary for good topic interpretability.

For the Entity Intrusion task, all of the relevant methods show comparable performance in identifying Author and Venue intruders in the DBLP dataset (though CATHYHIN is still constistently the highest). Since the DBLP dataset is well structured, and the entity links are highly trustworthy, identifying entities by topic is likely easier. However, the entities in the NEWS dataset were automatically discovered from the data, and the link data is therefore noisy and im-

TABLE VII: The 'Egypt' topic and the least sensible subtopic, as generated by three methods (only Phrases and Locations are shown)

CATHYHIN	$\text{CATHY}_{\text{heuristic\_HIN}}$	$\text{NetClus}_{\text{pattern}}$
{egypt; egypts; death toll; morsi} / {Egypt; Egypt Cairo; Egypt Israel; Egypt Gaza}	{egypt; egypts morsi; egypt imf loan; egypts president} / {Egypt; Cairo; Tahrir Square; Port Said}	{bill clinton; power nuclear; rate unemployment; south sudan} / {Egypt Cairo; Egypt Coptic; Israel Jerusalem; Libya Egypt}
↓	↓	↓
{death toll; egyptian; sexual harassment; egypt soccer} / {Egypt Cairo; Egypt Gaza; Egypt Israel}	{supreme leader; army general sex; court; supreme court} / {US; Sudan; Iran; Washington}	{egypts coptic pope; egypts christians; obama romney; romney campaign} / {Egypt Cairo; Egypt Coptic; Israel Jerusalem; Egypt}

perfect. CATHYHIN is the most effective in identifying both Location and Person intruders. Once again, both better topic discovery and improved pattern representations are responsible for CATHYHIN's good results, and simply enhancing the pattern representations, whether for CATHY or NetClus, cannot achieve competitive performance.

CATHYHIN performs very well in the Topic Intrusion task on both datasets. Similar to the Phrase Intrusion task, both CATHYHIN and CATHY yield equally good or better result when phrases and entity patterns are mined, rather than just terms and single entities. The fact that CATHYHIN always outperforms CATHY demonstrates that utilizing entity link information is indeed helpful for improving topical hierarchy quality. As a worst-case study, Table VII illustrates three representations of the topic 'Egypt' (one of the 16 top stories in NEWS dataset), each with its least comprehensible subtopic. The locations found within the CATHYHIN subtopic are sensible. However, $\text{CATHY}_{\text{heuristic\_HIN}}$ first constructs phrase-represented topics from text, and then uses entity link information to rank entities in each topic. Thus the entities are not assured to fit well into the constructed topic, and indeed, the $\text{CATHY}_{\text{heuristic\_HIN}}$ subtopic's locations are not reasonable given the parent topic. Finally, $\text{NetClus}_{\text{pattern}}$ conflates 'Egypt' with several other topics, and the pattern representations can do little to improve the topic interpretability.

In all three intruder detection tasks on both datasets, CATHYHIN consistently outperforms all other methods, showing that an integrated heterogeneous model consistently produces a more robust hierarchy which is more easily interpreted by human judgement.

V. Conclusion

In this work, we address the problem of constructing a multi-typed topical hierarchy from heterogeneous information networks. We develop a novel clustering and ranking method which can be recursively applied to hierarchically discover multi-typed subtopics from heterogeneous network data. Our approach mines each discovered topic for topical patterns, yielding a comprehensive representation of each topic comprising lists of ranked patterns with different types (phrases, authors, *etc.*). Our experiments on the science and news domains demonstrate the significant advantage of our unified generative model for the task of hierarchical topic discovery, as compared to the state-of-the-art heterogeneous network analysis technique. We also show our constructed topical hierarchies have high quality based on human judgement.

We hope to further improve our multi-typed topical hierarchy construction method to be able to accomodate user preference for the particular hierarchical organization of a dataset. We are also interested in constructing evolving topical hierarchies that would be able to work with the constantly changing information found in data streams.

Acknowledgments: The work was supported in part by the U.S. Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053 (NS-CTA) and W911NF-11-2-0086 (CyberSecurity), the U.S. Army Research Office under Cooperative Agreement No. W911NF-13-1-0193, and U.S. National Science Foundation grants CNS-0931975, IIS-1017362, IIS-1320617. Chi Wang was supported by a Microsoft Research PhD Fellowship. Marina Danilevsky was supported by a National Science Foundation Graduate Research Fellowship grant NSF DGE 07-15088.

References

[1] Y. Sun, Y. Yu, and J. Han, "Ranking-based clustering of heterogeneous information networks with star network schema," in *KDD*, 2009.
[2] C. Wang, M. Danilevsky, N. Desai, Y. Zhang, P. Nguyen, T. Taula, and J. Han, "A phrase mining framework for recursive construction of a topical hierarchy," in *KDD*, 2013.
[3] S. Gauch, J. Chaffee, and A. Pretschner, "Ontology-based personalized search and browsing," *Web Intelligence and Agent Systems*, vol. 1, no. 3/4, pp. 219–234, 2003.
[4] W. Wong, W. Liu, and M. Bennamoun, "Ontology learning from text: A look back and into the future," *ACM Computing Surveys (CSUR)*, vol. 44, no. 4, p. 20, 2012.
[5] D. Lawrie and W. B. Croft, "Discovering and comparing topic hierarchies," in *Proc. RIAO*, 2000.
[6] R. Snow, D. Jurafsky, and A. Y. Ng, "Learning syntactic patterns for automatic hypernym discovery," *NIPS*, 2004.
[7] R. Navigli, P. Velardi, and S. Faralli, "A graph-based algorithm for inducing lexical taxonomies from scratch," in *IJCAI*, 2011.
[8] E. Zavitsanos, G. Paliouras, G. A. Vouros, and S. Petridis, "Discovering subsumption hierarchies of ontology concepts from text corpora," in *Proc. IEEE/WIC/ACM Intl. Conf. Web Intelligence*, 2007.
[9] L. Di Caro, K. S. Candan, and M. L. Sapino, "Using tagflake for condensing navigable tag hierarchies from tag clouds," in *KDD*, 2008.
[10] S.-L. Chuang and L.-F. Chien, "A practical web-based approach to generating topic hierarchy for text segments," in *CIKM*, 2004.
[11] X. Liu, Y. Song, S. Liu, and H. Wang, "Automatic taxonomy construction from keywords," in *KDD*, 2012.
[12] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Mach. Learn.*, vol. 42, no. 1-2, pp. 177–196, Jan. 2001.
[13] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.
[14] Y. Sun, J. Han, J. Gao, and Y. Yu, "itopicmodel: Information network-integrated topic modeling," in *ICDM*, 2009.
[15] H. Deng, J. Han, B. Zhao, Y. Yu, and C. X. Lin, "Probabilistic topic models with biased propagation on heterogeneous information networks," in *KDD*, 2011.
[16] X. Chen, M. Zhou, and L. Carin, "The contextual focused topic model," in *KDD*, 2012.
[17] H. Kim, Y. Sun, J. Hockenmaier, and J. Han, "Etm: Entity topic models for mining documents associated with entities," *ICDM*, 2012.
[18] J. Tang, M. Zhang, and Q. Mei, "One theme in all views: modeling consensus topics in multiple contexts," in *KDD*, 2013.
[19] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach," *Data Min. Knowl. Discov.*, vol. 8, no. 1, pp. 53–87, jan 2004.
[20] Q. Li, H. Ji, and L. Huang, "Joint event extraction via structured prediction with global features," in *ACL*, 2013.
[21] D. Newman, J. H. Lau, K. Grieser, and T. Baldwin, "Automatic evaluation of topic coherence," in *NAACL-HLT*, 2010.
[22] J. Chang, J. Boyd-Graber, C. Wang, S. Gerrish, and D. M. Blei, "Reading tea leaves: How humans interpret topic models," in *NIPS*, 2009.