Entity Role Discovery in Hierarchical Topical Communities

Marina Danilevsky, Chi Wang, Nihit Desai, Jiawei Han
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL, USA
{danilev1,chiwang1,nhdesai2,hanj}@illinois.edu

ABSTRACT

People and social communities are often characterized by the topics and themes they are working on, or communicating about. Discovering the roles played by different entities in these communities are of great interest in many real-world contexts in social network analysis. We are also often interested in discovering such roles at different levels of granularity. In this paper we study a new problem of mining entity roles in hierarchical topical communities. We first detect topical communities from the text component of a social or information network. Since we mine phrases from the network, and represent topical communities by ranked lists of mixed-length phrases, the communities have a good interpretation at multiple levels of the hierarchy. We are therefore able to discover topical roles of different types of entities in both large communities encompassing more general topics, and small, focused subcommunities. We demonstrate our method on a bibliographic information network dataset, which we use to discover the roles of authors and publication venues in the context of the hierarchical topical communities.

Categories and Subject Descriptors

I.7 [**Computing Methodologies**]: Document and Text Processing; H.2.8 [**Database Applications**]: Data Mining

Keywords

Role Discovery, Entity Roles, Hierarchical Communities, Network Analysis

1. INTRODUCTION

People and social communities are often characterized by the topics and themes they are working on, or communicating about. The roles played by different entities in these communities are of great interest in many contexts of social network analysis. We may be interested in discovering the role of an author in a research community, or the contribution of a user to a social network community organized around similar interests. These types of role discovery tasks center around topical communities mined from social or information networks.

We are also often interested in analyzing such roles at different levels of granularity. In the real world, topical communities - communities built around shared topics - are naturally hierarchical. People participate in large communities, encompassing many interests, as well as small, focused subcommunities. Therefore, in order to analyze the various roles that an entity plays in such different contexts, we must also be able to work with topical communities and subcommunities.

In this paper we study a new problem of mining entity roles in hierarchical topical communities. We first detect topical communities from the text component of a social or information network. In order to clearly represent the topics associated with all the communities throughout the hierarchy, we mine phrases of various lengths from the text, which can best summarize each of the topics. The topics of a child community represent subtopics found in the parent community. For example, in the context of computer science topics, the community centered around topics on query processing and optimization may be described by the phrases {'query processing', 'query optimization',...}, while its parent community on general database topics may be described by {'query processing', 'database systems', 'concurrency control',...}. We can then discover the roles of authors who publish in these communities. The hierarchical structure of the topical communities allows us to distinguish between, e.g., authors who publish on a diverse range of database topics, and authors who are particular experts in query processing.

Our approach works with heterogeneous social or information networks with a text component attribute in the form of a *content-representative* document. A document is content-representative if it may serve as a concise description of its accompanying full document. For example, the title of a scientific paper is usually a content-representative document, because it is a good representation of the topics found in the paper itself. However, the same is rarely true of e.g. fiction books. A bibliographic information network, consisting of paper titles, authors, and venues, or a social network, consisting of blog titles and users, are two examples of datasets that could be analyzed with our approach.

The main contributions of our work are twofold:

• We construct a hierarchy of high quality topical communities from a network dataset. Community topics are represented by ranked lists of phrases, so that the topics of

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.
Copyright 2013 ACM 978-1-4503-2174-7/13/08 ...$10.00.

a child community are subtopics found in the parent community.

- We infer roles of entities in the contexts of the topical communities via their links to the constructed hierarchy. We illustrate our role mining techniques with multiple examples on a real world dataset.

2. RELATED WORK

2.1 Hierarchical community detection

Link-based community detection has been studied extensively in the past decade. A multitude of methodologies have been proposed for the community detection, or graph clustering problem in network analysis (see [20] and [8] for comprehensive surveys). For hierarchical community discovery, both deterministic [27] and probabilistic methods [5] have been proposed. Lancichinetti *et al.* [15] presents the first algorithm that finds both overlapping communities and the hierarchical structure. Agglomerative, or bottom-up approaches are the most common in these studies.

Models for discovering topics from text documents have also been studied extensively. One such representative model is Latent Dirichlet Allocation [1], which takes documents as input, models them as mixtures of different topics, and outputs word distributions for each topic. Several extensions can model the hierarchical dependency of topics [9, 16, 19], but the results do not define hierarchical communities easily. Other extensions can use n-grams instead of unigrams to represent each topic with better interpretability [26, 23], but still cannot find hierarchical topics.

Some recent work shows that topic modeling can be used to augment community discovery [28, 17, 2, 29]. These approaches can find mixed membership for entities in various topical communities. However, they also do not study hierarchical communities. We perform top-down discovery of hierarchical topical communities, represented by phrases mined from the text component of a heterogeneous information network, and then infer communities of entities via their links to the text in the network.

2.2 Role discovery

Role analysis has its root in sociology. Sociologists use a notion of equivalence to assign nodes to different roles. For example, *automorphic equivalence* requires nodes in the same equivalence class (role) to have equivalent neighbors [7]. Network analysts use various notions of 'centrality' to find roles such as authorities and hubs [14]. Most works in computer science present a very similar flavor of role definition. As a result, the techniques of role discovery are all essentially either clustering [18, 11, 10], or ranking [12, 25, 3]. However, community knowledge has been shown to be useful in role analysis [22, 4], and Costa and Ortale [6] recently developed a Bayesian approach to jointly model the links generated by both communities and roles. They do not analyze the roles in communities defined by a textual hierarchy.

Our work makes use of the hierarchical topical community to perform contextual role discovery and analysis, in contrast to previous work on role discovery. In this study, we focus on simple types of roles, characterized by entities' contribution to different topical communities and subcommunities. However, more complicated roles can also be analyzed within this context.

3. HIERARCHICAL TOPICAL COMMUNITY DETECTION

3.1 Preliminaries and Definitions

Traditionally, a phrase is defined as a consecutive sequence of terms, or unigrams. However, as discussed in [13] this definition can be quite limiting as it is too sensitive to natural variations in the term order, or the morphological structure of a phrase. For instance, consider that two computer science paper titles, one containing 'mining frequent patterns' and the other containing 'frequent pattern mining,' are clearly discussing the same topic, and should be treated as such. A phrase may also be separated by other terms: '**mining** top-k **frequent** closed **patterns**' also belongs to the topic of frequent pattern mining, in addition to incorporating secondary topics of top-k frequent patterns, and closed patterns. Therefore, we define a phrase to be an order-free set of terms appearing in the same document.

Definition 1 (Phrase). *A phrase P with length n is an unordered set of n terms: $P = \{w_{x_1}, \ldots, w_{x_n} | w_{x_i} \in W\}$, where W is the set of all unique terms in a content-representative document collection. The frequency $f(P)$ of a phrase is the number of documents in the collection that contain all of the n terms.*

We use phrases as the basic units for constructing a hierarchy of topical communities. A *topical community hierarchy* is defined as a tree of topical communities. Every non-root *topical community c* is represented by a ranked list of phrases $\{\mathcal{P}^c, r^c(\mathcal{P}^c)\}$, where $\mathcal{P}^C$ is the set of phrases for community c, and $r^c(P^c)$ is the ranking score for the phrases in community c. For every non-leaf topical community c in the tree, its children Ch^c are its subcommunities. A phrase can appear in multiple topical communities, though it will have a different ranking score in each one. We use the terms 'community' and 'topical community' interchangeably.

Definition 2 (Community Frequency). *The community frequency $f_c(P)$ of a phrase is the count of the number of times the phrase is attributed to community c. For the root node o, $f_o(P) = f(P)$. For each community in the hierarchy, with subcommunities Ch^c, $f_c(P) = \sum_{z \in Ch^c} f_z(P)$, i.e., the community frequency is equal to the sum of the subcommunity frequencies.*

Table 1 illustrates an example of estimating community frequency of phrases for a community built around the topic of computer science, with 4 subcommunities. The phrase 'support vector machines' is estimated to belong entirely to the Machine Learning (ML) community with a community frequency of 85. However, 'social networks' is fairly evenly distributed among three communities.

Phrase	ML	DB	DM	IR	Total
support vector machines	85	0	0	0	85
query processing	0	212	27	12	251
world wide web	0	7	1	26	34
social networks	39	1	31	33	104

Table 1: Example of estimating topical frequencies of phrases in four communities (inferred to be machine learning, database, data mining, and information retrieval.)

In order to estimate the community frequency for each phrase, we need to also infer the communities present in the entire dataset. We do community inference and community frequency estimation by analyzing our dataset's term co-occurrence network.

A *term co-occurrence network* $G = (W, E)$ is constructed from a collection of content-representative documents, and consists of a set of nodes W and a set of links E. A node $w_i \in W$ represents a term, and a link (w_i, w_j) between two nodes represents a co-occurrence of the two terms in a document. The number of links $e_{ij} \in E$ between two nodes w_i and w_j is equal to the number of documents containing both terms. For each node w_i, we also create a self-link e_{ii} for every document where w_i appears.

The term co-occurrence network reflects the structure of the entire collection of content-representative documents. However, we wish to unearth the communities and subcommunities contained within the network. To accomplish this, it is beneficial to discover the term co-occurrence subnetworks associated with each subcommunity, which have all of the nodes from their parent networks, but only those links which belong to the particular subnetwork.

Formally, every community node c in the topical community hierarchy is associated with a term co-occurrence network G^c. The root node o is associated with the original term co-occurrence network, G^o, constructed from the full document collection. For every non-root node, we construct a subnetwork by clustering the term co-occurrence network of its parent.

Our approach detects topical communities in a top-down, recursive way:

Step 1. Construct the term co-occurrence network $G = (W, E)$ from the document collection. Set $c = o$, $G^c = G$.

Step 2. For a community c, cluster the term co-occurrence network G^c into subcommunity subnetworks $G^z, z \in Ch^c$, and extract the phrases in each subcommunity z based on estimated community frequency.

Step 3. Construct a representation of each subcommunity $z \in Ch^c$ as a list of the community's phrases ranked by phrase quality. The quality of a phrase in a particular community is estimated using a unified function based on community frequency.

Step 4. Recursively apply Steps 2 and 3 to each subcommunity $z \in Ch^c$ to construct the hierarchy top-down.

3.2 Topical Community Phrases

For a given community c, assume c has k child communities, denoted by $z = 1 \dots k$. The value of k can be either specified by users or chosen using a model selection criterion such as the Bayesian Information Criterion [21]. We follow the approach in [24], which develops a generative model of the term co-occurrence network, uses it to estimate topical community frequency $f_z(p), z \in Ch^c$, and then discovers phrases by mining frequent topical community patterns with minimal support *minsup*. As we follow [24] directly for this step, we omit the algorithm details here.

We are therefore able to discover phrases (previously defined to be an unordered set of terms) by mining all of the frequent term sets up to length N in each community. Note that for only the top level communities $z \in Ch^o$, the parent community frequency $f_{Parent(z)}(P)$ is equal to $f(P)$ and must be counted from the text. However, for all lower levels, the parent community frequency $f_{Parent(z)}(P)$ was already calculated when the parent community was generated, and therefore never needs to be counted.

3.3 Topical Community Representation

Finally, we construct a representation of each community so that a human judge may be able to understand a community's topics by glancing at its top ranked phrases. Therefore, we must determine and quantify the characteristics which define a high quality phrase within a community. We adapt the phrase-ranking approach in [24] to our goal of representing a topical community.

As an example, consider the task of judging what constitutes high quality phrases for various topics in computer science. The most straightforward criterion is that a representative phrase for a community should have good *coverage* of many documents within that community. For example: 'information retrieval' has better coverage than 'cross-language information retrieval' in the Information Retrieval topic, and is therefore better at representing that community.

However, a high quality phrase should also be a pure indicator of its specific community, rather than a broad indicator of many communities.We can identify a phrase which has high *purity* in a community if it is only frequent in documents belonging to that community and not frequent in documents within other communities. For instance, in the community on Databases, the phrase 'query processing' is more pure than 'query', since observing the phrase 'query' just as easily suggests other communities, such as Information Retrieval.

Finally, a phrase should be a real phrase, rather than a combination of frequent terms, which we refer to as possessing the *phraseness* characteristic. A group of terms should be combined together as a phrase if they co-occur significantly more often than the expected chance co-occurrence frequency, given that each term in the phrase occurs independently. For example, 'active learning' is a better phrase than 'learning classification' in the Machine Learning topic.

We combine the 3 criteria of coverage, purity, and phraseness into a phrase community quality ranking function using a probabilistic modeling approach. For each criterion, we estimate a ranking measure based on occurrence probability. The *occurrence probability* of a phrase is the likelihood of observing all the terms in the phrase in the same document.

Denote $|D_z|$ to be the estimated number of documents in a community z ($|D_o|$ represents the number of documents in the root community, and is therefore equal to the size of the document collection). We can then calculate the occurrence probability of a phrase P conditioned on community z:

$$p(P|z) = \frac{f_z(P)}{|D_z|} \qquad (1)$$

the probability of independently seeing every term in phrase $P = \{w_{x_1}, \dots w_{x_n}\}$ conditioned on community z:

$$p_{indep}(P|z) = \prod_{i=1}^{n} p(w_{x_i}|z) = \prod_{i=1}^{n} \frac{f_z(w_{x_i})}{D_z} \qquad (2)$$

and the probability of a phrase P conditioned on a mixture of multiple communities $Z \subset Ch^{Parent(z)}, Z \supsetneqq \{z\}$:

$$p(P|Z) = \frac{\sum_{c \in Z} f_c(P)}{\sum_{c \in Z} D_c} \qquad (3)$$

The coverage of a phrase is quantified directly by $p(P|z)$.

The phraseness can be measured by comparing the probability of seeing a phrase to the *contrastive probability* defined by (Eq. 2) - the probability of seeing the individual terms in the phrase independently, conditioned on community z. The purity can be measured by comparing the probability of seeing the phrase conditioned on community z, and the contrastive probability defined by (Eq. 3) - the probability of seeing the phrase conditioned on a mixture of multiple communities Z. The definition of purity is configurable by altering the makeup of the community mixture Z. For example, using the mixture of all the sibling communities $Ch^{Parent(z)}$ as the topic mixture results in a weaker purity criterion. However, deliberately choosing the subset Z so that the contrastive probability $p(P|Z)$ is maximized, results in a stronger purity criterion.

The three criteria are unified by the ranking function:

$$r^z(P) = p(P|z)\left(\log\frac{p(P|z)}{p(P|Z)} + \omega\log\frac{p(P|z)}{p_{indep}(P|z)}\right) \qquad (4)$$

where ω controls the importance of the phraseness criterion.

The coverage measure $p(P|z)$ is the most influential, since the other two criteria are represented by log ratios of $p(P|z)$ and a contrastive probability, and the effect of contrastive probability on the ranking score is smaller than the influence of $p(P|z)$. This is a desirable property because when a phrase P has low support, the estimates of purity and phraseness are unreliable; but their effect is small since the value of $p(P|z)$ would be correspondingly low. Therefore, a phrase with low coverage would inevitably be ranked low, as should be the case for representative phrases.

The relative importance of the purity and phraseness measures is controlled by ω. Both measures are log ratios on comparable scales, and can thus be balanced by weighted summation. As ω increases, we expect phrases which are common across all communities to be ranked higher. We empirically set $\omega = 0.5$, which ensures that community-specific phrases will be highly ranked.

4. ROLE DISCOVERY

We present role discovery results on a real-world heterogeneous bibliographic information network. We collected a set of recently published computer science papers in the areas related to Databases, Data Mining, Information Retrieval, Machine Learning, and Natural Language Processing. These papers come from DBLP[1], a bibliography website for computer science publications, and contain title, author, and publication venue information. We minimally pre-processed the dataset by removing all stopwords from the titles, and removing all authors who only have one paper, resulting in a collection of 12,886 authors, 20 venues, and 33,313 titles consisting of 18,598 unique terms.

In this section we illustrate two types of role discovery that can be performed using the topical community hierarchy constructed from the DBLP dataset. First, given a topical community and an entity type, which entities play the most important roles in the community? For example, an author's contribution to the topics of a community (by way of published papers) represents the author's role in that community. Second, for a given topical community and specific entity, what is that entity's role in the community? For instance, which topics within the community get published in a particular conference? Or, which specific topics within the community does an author contribute to? The topical community hierarchy allows for more nuanced role discovery for a given entity, presenting detailed information at different levels of granularity.

4.1 Ranking Community Entities

The role of an entity in a topical community can be interpreted as that entity's contribution to the community. For example, the role of an author is represented by the work the author does on the community's topics; the role of a venue is represented by the topics in the community which get published in the venue. Therefore, a natural question to ask is which entities play the roles of top contributors to a particular topical community.

If we consider the role of an entity E in a community z to be that of a contributor of documents (e.g., the role of an author is defined by how many papers he has published on the community's topics), we can represent the entity's contribution by estimating the number of documents connected to E which belong to z.

Denote the estimated number of documents in a community z as $|D_z|$, and the set of all documents connected to E as D_E. For example, in the DBLP dataset, the subset D_A is the set of papers authored by A, and D_V is the set of papers published in V. Then, we need to estimate $|D_{E,z}|$, the number of documents attributed to E in community z.

We must first estimate the community frequency of every document $d_E \in D_E$. In Section 3 we described how to estimate $f_z(P)$, the community frequency of phrase P. We proceed in a similar top-down recursive fashion in order to estimate the *document* community frequency, $DF_z(d_E)$.

For each document d_E we first perform the intermediate step of calculating the *total phrase frequency* of d_E in community z by adding up the normalized community-z frequencies of all the phrases in d_E:

$$TPF_z(d_E) = \sum_{P\in d_E}\frac{f_z(P)}{\sum_{c\in Ch^{Parent(z)}} f_c(P)} \qquad (5)$$

The next step is to calculate the normalized *document frequency* of d_E in community z:

$$DF_z(d_E) = \frac{TPF_z(d_E)}{\sum_{c\in Ch^{Parent(z)}} TPF_c(d_E)} DF_{Parent(z)}(d_E) \qquad (6)$$

The community frequency of a document is distributed among that community's children, so that the document frequency in a given community is the sum of the document frequencies in the community's children, $\sum_{c\in Ch^z} DF_c(d) = DF_z(d)$. One exception is that a few documents may contain no frequent topical phrases in any subcommunities because we filter out infrequent topical phrases. For such documents we do not count their contribution to any subcommunities.

Figure 1 shows a hypothetical distribution of document frequency for some document. The document frequency values for every set of subcommunities sum up to the document frequency in the parent community (where the frequency at the root is necessarily 1 for any document).

Finally, we calculate the entity community frequency $EF_z(E)$ by summing up the contributions of all the documents $d_E \in D_E$ to community z:

$$EF_z(E) = \Sigma_{d_E\in D_E} DF_z(d_E) \qquad (7)$$

[1] http://www.dblp.org/

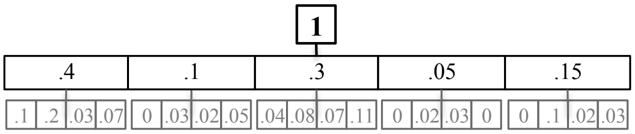


Figure 1: A hypothetical distribution of document frequency values for a document, in a hierarchy with 3 levels, beginning at the root.

Since some documents may not contribute to any of the subcommunities, the entity frequency in a given community should be equal to or larger than the sum of the entity frequencies in the community's children, $\sum_{c \in Ch^z} EF_c(E) \leq EF_z(E)$. It is clear now that $EF_z(E)$ is precisely our estimate for $|D_{E,z}|$.

4.1.1 Normalizing Phrase Community Frequency

Eq. 5 normalizes a phrase's contribution to a document in a given community. Why do we not use the unnormalized $f_z(P)$ which would ensure that a phrase that is more frequent in z influences the document more?

We argue that normalization is better. We would like to fit the document community frequency with the phrase community frequency, *i.e.*, $f_c(P) \approx \sum_{P \in d} DF_c(d)$. Consider a phrase P in document d. The total contribution of $DF_c(d)$ to $f_c(P)$ for all children c of one community z is $\sum_c DF_c(d) = DF_z(d)$. If $DF_c(d) = \frac{f_c(P)}{\sum_c f_c(P)}$ (*i.e.*, the normalized community frequency) holds for all $d \ni P$, then we have exactly $f_c(P) = \sum_{P \in d} DF_c(d)$. However, this is impossible when there are multiple phrases in a document and they have different normalized community frequency. Instead, we can try to minimize the square error $\sum_{P \in d} \sum_c [\frac{f_c(P)}{\sum_c f_c(P)} - DF_c(d)]^2$ with the constraint $\sum_c DF_c(d) = DF_z(d)$. Solving this constrained optimization problem yields the solution in Equation 6.

We also evaluated the accuracy of using normalized and unnormalized phrase community frequency. We labeled each document in our collection with the community in which it was most frequent, according to both estimates. We found that nearly $\frac{1}{3}$ of the documents ended up with different community labels. We sampled a random 1% of these papers, and manually labeled them. We found that the labeling accuracy dropped by 20% from normalizing the phrase contribution to not normalizing. Therefore, normalizing phrase contribution actually does perform better.

4.1.2 Variations in Entity Ranking

As defined in Section 3.3, let $|D_z|$ denote the estimated number of documents in a community z. Let D_E denote the set of all documents connected to E where $DF_{Parent(z)}(d_E) \neq 0$, $d_E \in D_E$. Then, $|D_{E,z}|$ denotes the estimated number of documents attributed to entity E in community z (and is precisely equal to $EF_z(E)$, the entity community frequency of E as defined in Equation 7). Ranking entities by the value of $|D_{E,z}|$ means only taking into account how much of the topic an entity covers. This ranking would find, for example, the authors who have published the most number of papers on the topics of a particular community. We refer to ranking entities by $EF_z(E)$ as $ERank_{Cov}$.

However, this entity ranking function is not able to discover authors who are more dedicated to their role in a given community than to sibling communities. In order to take this into account, we adapt the notion of purity, as introduced in Section 3.3, to apply to entities.

We can calculate the occurrence probability of entity E in community z:

$$p(E|z) = \frac{|D_{E,z}|}{|D_z|} \tag{8}$$

and the contrastive probability of seeing E conditioned on a mixture of multiple communities $Z \subset Ch^{Parent(z)}, Z \supsetneqq \{z\}$ (which is analogous to Eq. 3):

$$p(E|Z) = \frac{\sum_{c \in Z} |D_{E,c}|}{\sum_{c \in Z} |D_c|} \tag{9}$$

As noted in Section 3.3, we choose the subset of Z to maximize this probability and strengthen the purity criterion.

We evaluate the purity of entity E in z by comparing the probability of seeing a document E conditioned on community z and the contrastive probability defined by Eq. 9.

The criteria of entity purity and coverage can then be unified in an analogous way to Eq. 4, with the exception that the notion of phraseness is not applicable to entities. We refer to ranking entities by this value as $ERank_{Cov+Pur}$:

$$ERank_{Cov+Pur}(E, z) = p(E|z) \log \frac{p(E|z)}{p(E|Z)}$$

{sensor networks, selectivity estimation, large databases, pattern matching,spatio-temporal moving objects, large collections}	*{time series, nearest neighbor, moving objects, time series data, nearest neighbor queries}*	*{association rules, large scale, mining association rules, privacy preserving, frequent itemsets}*	*{high dimensional, data streams, data mining, high dimensional data, outlier detection}*
divesh srivasta	eamonn j. keogh	jiawei han	philip s. yu
nick koudas	philip s. yu	philip s. yu	jiawei han
jiawei han	christos faloutsos	jian pei	charu c. aggarwal
philip s. yu	hans-peter kriegel	christos faloutsos	jian pei
christos faloutsos	jiawei han	ke wang	christos faloutsos

(a) $ERank_{Cov}$

{sensor networks, selectivity estimation, large databases, pattern matching,spatio-temporal moving objects, large collections}	*{time series, nearest neighbor, moving objects, time series data, nearest neighbor queries}*	*{association rules, large scale, mining association rules, privacy preserving, frequent itemsets}*	*{high dimensional, data streams, data mining, high dimensional data, outlier detection}*
divesh srivasta	eamonn j. keogh	jiawei han	charu c. aggarwal
surat chaudhiri	jessica lin	ke wang	graham cormode
nick koudas	michail vlachos	xifeng yan	s. muthukrishnan
jeffrey f. naughton	michael j. passani	bing liu	philip s. yu
yannis papakonstantinou	matthias renz	mohammed j. zaki	xiaolei li

(b) $ERank_{Cov+Pur}$

Table 2: Top ranked authors in the four subcommunities of Data Mining, based on $ERank_{Cov}$ and $ERank_{Cov+Pur}$.

Table 2 shows the top ranked authors in the four subcommunities of Data Mining, based on $ERank_{Cov}$ and $ERank_{Cov+Pur}$. When only coverage is used for ranking, many authors are highly ranked in all communities (e.g. Philip Yu, Jiawei Han, and Christos Faloutsos are top-5 authors in every community). When both coverage and purity criteria are taken into account, only those authors who are significantly more dedicated to one community are highly ranked, resulting in no overlap between communities. Some prolific authors, such as Christos Faloutsos, are no longer highly ranked anywhere, because their contributions are fairly equal among the communities. We are able to easily discover both of these roles.

Table 3 shows further examples of ranking authors (using $ERank_{Cov+Pur}$) within two subcommunities of the Database community. By showing the top ranked phrases for each author in a community (we discuss how these are generated in the following section) we are able to see both which authors play the most important roles, and what part of the community each author contributes to.

{*query processing / query optimization / deductive databases / materialized views / microsoft sql server / relational databases*}	
elke a. rundensteiner	query processing / query optimization / materialized views / stream processing / object-oriented databases
hamid pirahesh	query processing / query optimization / materialized views / relational data / relational xml
surajit chaudhuri	query optimization / relational databases / microsoft sql server / materialized views / relational data
jeffrey f. naughton	materialized views / xml query / query processing / relational xml / maintenance view
per-åke larson	materialized views / microsoft sql server / query optimization / materialized maintenance views / relational data
vivek r. narasayya	microsoft sql server / materialized views / relational databases / query management / sql data
serge abiteboul	materialized views / xml data / schemas / query evaluation / materialized maintenance views

(a) A Database subcommunity

{*concurrency control / database systems / main memory / load shedding / database concurrency control / load balancing*}	
avi silberschatz	concurrency control / main memory / locking / database systems / transaction management
david b. lomet	recovery / systems recovery / b-trees / transactions recovery / performance access
henry k. korth	concurrency control / database systems / main memory / protocol / transaction systems
bharat k. bhargava	concurrency control / distributed systems / distributed database / recovery / distributed database systems
c. mohan	concurrency control / recovery / locking / data systems / transaction systems
ahmed k. elmagarmid	database systems / concurrency control / distributed database / distributed systems / access control
nancy lynch	concurrency control / locking / nested transactions / control transactions / concurrency transactions

(b) A Database subcommunity

Table 3: The top ranked authors (using $ERank_{Cov+Pur}$) in two subcommunities of the Database community, along with each author's top ranked phrases in each community. Each subcommunity is represented by its top-ranked phrases, shown in the first row of each table.

4.2 Entity Role in Community

The second type of role discovery we illustrate is finding out a specific entity's role in a given topical community. In order to represent an entity's role in a community, we want to highlight the subset of the communities which illustrates the contribution of the entity. We now therefore introduce a phrase *entity community contribution* ranking function:

$$Cont(P|z,E) = -p(P|z)log(\frac{p(P|z)}{p(P|z,E)}) \qquad (10)$$

where $p(P|z) = \frac{f_z(P)}{|D_z|}$ and $p(P|z,E) = \frac{f_{z,D_E}(P)}{|D_{E,z}|}$.

$f_{z,D_E}(p)$ represents the frequency of phrase P in community z for the document subset D_E. We estimate it as $\sum_{d \in D_E, d \ni P} DF_z(d)$. $Cont(P|z,E)$ has a nice information theoretic interpretation as the pointwise Kullback-Leibler (KL) divergence between the likelihood of seeing phrase P in the documents in community z, and the likelihood of seeing phrase P specifically in the documents linked to entity E, in z. Pointwise KL divergence is a distance measure between two probabilities. Therefore, $Cont(P|z,E)$ upranks P if its frequency in the community in conjunction with the entity E is higher than would be expected, based on its overall topical community frequency.

However, using only the Contribution ranking does not give ideal results. Table 4 shows the roles of two authors, Philip S. Yu and Christos Faloutsos, in one of the subcommunities of Data Mining subtopics. Using only the contribution ranking function defined in Eq. 10 results in poor quality phrases such as 'fast large.' On the other hand, using only the phrase quality ranking function defined in Eq. 4 - which we refer to here as $Qual(P|z)$ - is also insufficient, as it only evaluates the quality of a phrase, regardless of any entity information. Therefore, we define a *Combined* ranking function for a phrase P which incorporates both the relationship between the entity E and the phrase, as well as phrase quality:

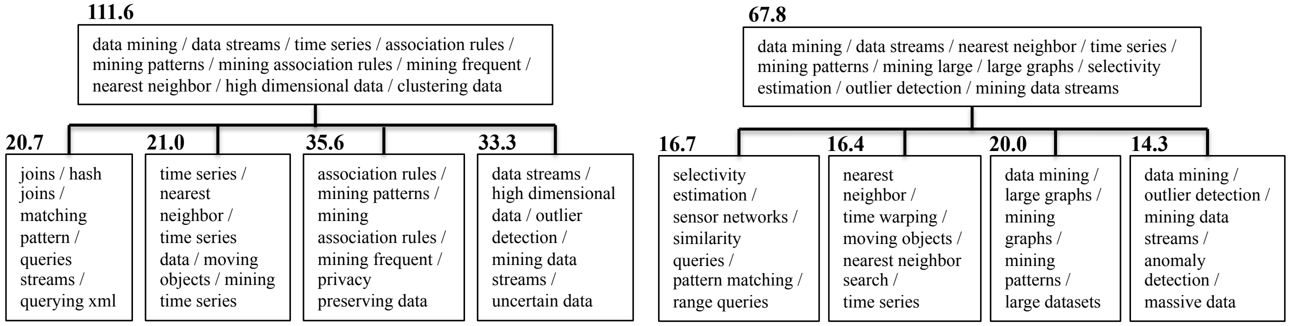
$$Comb(P|z,E) = \alpha Cont(P|z,E) + (1-\alpha)Qual(P|z) \quad (11)$$

The value of $\alpha \in [0,1]$ can vary. In our experiments, we empirically set $\alpha = 0.5$. Table 4 illustrates that the Combined ranking function yields a better list of phrases to represent the roles of the authors.

We can therefore use the Combined ranking function to discover the role of an entity in different topical communities in the hierarchy. As an example, Figure 2 shows the roles of Christos Faloutsos and Philip S. Yu in the Data Mining community, and its subcommunities. We also show the entity frequency for each community ($EF_z(E)$), which represents the estimate for the number of papers written by that author in the community.[2] The sum of the entity frequencies in the subcommunities do not quite add up to the entity frequency of the parent community because, as discussed in Section 4.1, a document does not contribute to the child subcommunities if all of its phrases have become too infrequent in them.

While both authors are prominent in the Data Mining community, Figure 2 illustrates how their roles are contrasted in that community, and even more strongly in the subcommunities. For instance, in the third (from left) sub-

[2] It so happens that our dataset contains more papers written by Philip Yu than by Christos Faloutsos, and so the entity topical community frequencies are higher for Philip Yu.


111.6
data mining / data streams / time series / association rules / mining patterns / mining association rules / mining frequent / nearest neighbor / high dimensional data / clustering data
20.7
joins / hash joins / matching pattern / queries streams / querying xml
21.0
time series / nearest neighbor / time series data / moving objects / mining time series
35.6
association rules / mining patterns / mining association rules / mining frequent / privacy preserving data
33.3
data streams / high dimensional data / outlier detection / mining data streams / uncertain data
67.8
data mining / data streams / nearest neighbor / time series / mining patterns / mining large / large graphs / selectivity estimation / outlier detection / mining data streams
16.7
selectivity estimation / sensor networks / similarity queries / pattern matching / range queries
16.4
nearest neighbor / time warping / moving objects / nearest neighbor search / time series
20.0
data mining / large graphs / mining graphs / mining patterns / large datasets
14.3
data mining / outlier detection / mining data streams / anomaly detection / massive data

(a) The roles of Philip S. Yu in Data Mining

(b) The roles of Christos Faloutsos in Data Mining

Figure 2: Contrasting the roles of two authors, Philip S. Yu and Christos Faloutsos, in the Data Mining community and subcommunities. The estimate for the number of papers the author contributes to each community is also shown.

Phrase Quality	**P. S. Yu** (Contribution)	**C. Faloutsos** (Contribution)	**P. S. Yu** (Combined)	**C. Faloutsos** (Combined)
time series	data indexing	time warping	time series	nearest neighbor
nearest neighbor	data similarity	distance	nearest neighbor	time warping
moving objects	distance	fast time	time series data	moving objects
time series data	fast large	time similarity	moving objects	nearest neighbor search
nearest neighbor queries	similarity indexing	fast large	time series mining	time series
mining time series	time series patterns	fast similarity	time series patterns	distance

Table 4: Using phrase quality, phrase entity community contribution, and a combination of both to represent the roles of Philip S. Yu and Christos Faloutsos in a Data Mining subcommunity

community, Philip Yu contributes work on the topics of mining frequent patterns and association rules, whereas the contribution of Christos Faloutsos is more geared towards the topics of mining large datasets and large graphs.

As another example of role discovery, Figure 3 shows the role of the SIGIR venue in all 5 top level communities, as well as the subcommunities of Machine Learning and Information Retrieval. The role of a venue in a community is represented by those topics within the community that are published in the venue. Thus, we can see that the Machine Learning topics that get published in SIGIR are techniques related to IR tasks such as feature selection methods that may be used for filtering, and approaches to text categorization and classification problems.

By examining the roles of different venues within a single community, we can also gain some insight to the flavor of each venue. As an example, Table 5 compares the roles of three venues - SIGIR, WWW, and ECML - in the general IR community. While both SIGIR and WWW are usually characterized as IR venues, we can clearly see that SIGIR plays a more broad role, publishing most of the topics present in the community, whereas WWW focuses only on those topics that are directly related to the web. On the other hand, ECML is considered to be an ML venue, and its contribution to the IR community is the publishing of papers on topics that use machine learning techniques. Note that all three venues share some high-ranked phrases, illustrating how the roles of all three venues overlap in this community. If we were to strictly label venues, and therefore the papers they publish, as belonging exclusively to one or another community, we would not be able to discover these interesting roles.

Table 5: The roles of three venues - SIGIR, WWW, and ECML - in the general Information Retrieval community

SIGIR	**WWW**	**ECML**
information retrieval	web search	word sense disambiguation
question answering	semantic web	world wide web
web search	search engine	information extraction
natural language	question answering	semantic role labeling
document retrieval	web pages	knowledge discovery
relevance feedback	world wide web	query expansion
query expansion	web services	machine translation

5. CONCLUSION

In this paper we study a new problem of mining entity roles in hierarchical topical communities. Our method detects topical communities from the text component of a social or information network. Since we mine phrases from the network, and represent topical communities by ranked lists of mixed-length phrases, the communities have a good interpretation at multiple levels of the hierarchy. We are able to discover topical roles of different types of entities in both large communities that encompass more general topics, and small, focused subcommunities. We demonstrate our method on a bibliographic dataset, which we use to discover the roles of authors and publication venues in the context of the hierarchical topical communities. It would be interesting to apply our method to other social and information networks, which would offer the opportunity to discover other compelling roles.

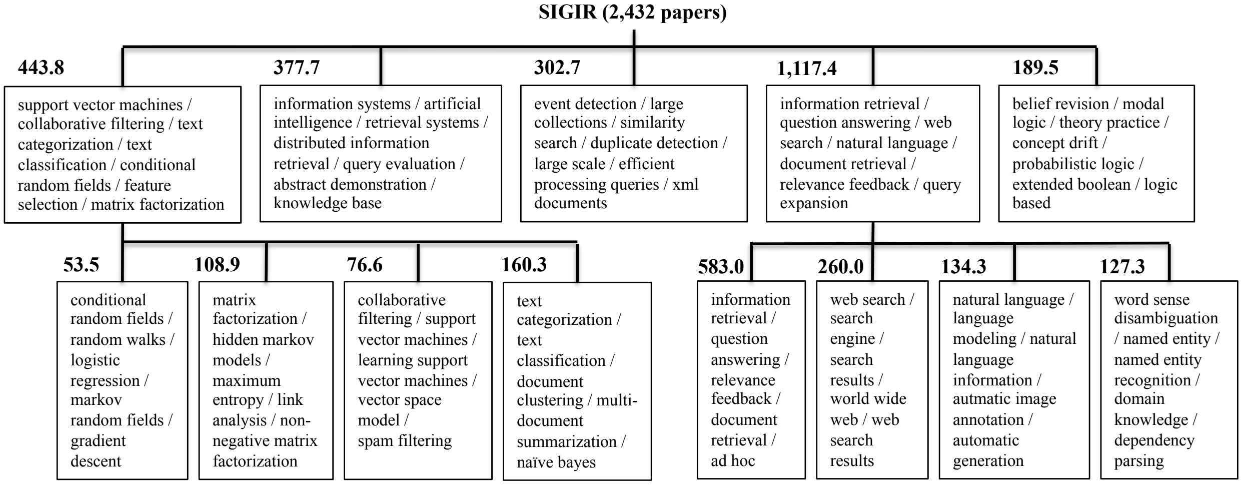

SIGIR (2,432 papers)
443.8
support vector machines / collaborative filtering / text categorization / text classification / conditional random fields / feature selection / matrix factorization
377.7
information systems / artificial intelligence / retrieval systems / distributed information retrieval / query evaluation / abstract demonstration / knowledge base
302.7
event detection / large collections / similarity search / duplicate detection / large scale / efficient processing queries / xml documents
1,117.4
information retrieval / question answering / web search / natural language / document retrieval / relevance feedback / query expansion
189.5
belief revision / modal logic / theory practice / concept drift / probabilistic logic / extended boolean / logic based
53.5
conditional random fields / random walks / logistic regression / markov random fields / gradient descent
108.9
matrix factorization / hidden markov models / maximum entropy / link analysis / non-negative matrix factorization
76.6
collaborative filtering / support vector machines / learning support vector machines / vector space model / spam filtering
160.3
text categorization / text classification / document clustering / multi-document summarization / naïve bayes
583.0
information retrieval / question answering / relevance feedback / document retrieval / ad hoc
260.0
web search / search engine / search results / world wide web / web search results
134.3
natural language / language modeling / natural language information / autmatic image annotation / automatic generation
127.3
word sense disambiguation / named entity / named entity recognition / domain knowledge / dependency parsing

Figure 3: The role of the venue SIGIR in several communities and subcommunities. The estimated number of papers published in SIGIR within each community is also shown.

Acknowledgments

This work was supported in part by the U.S. National Science Foundation grants IIS–0905215, U.S. Army Research Laboratory under Cooperative Agreement No. W911NF–09–2–0053 (NS-CTA). Chi Wang was supported by a Microsoft Graduate Fellowship. Marina Danilevsky was supported by a National Science Foundation Graduate Research Fellowship grant NSF DGE 07-15088.

6. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[2] J. Chang, J. Boyd-Graber, C. Wang, S. Gerrish, and D. M. Blei. Reading tea leaves: How humans interpret topic models. In *NIPS*, 2009.

[3] D. H. Chau, C. Nachenberg, J. Wilhelm, A. Wright, and C. Faloutsos. Large scale graph mining and inference for malware detection. In *SDM*, 2011.

[4] B.-H. Chou and E. Suzuki. Discovering community-oriented roles of nodes in a social network. In *Proceedings of the 12th international conference on Data warehousing and knowledge discovery*, DaWaK'10, 2010.

[5] A. Clauset, C. Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453:98–101, May 2008.

[6] G. Costa and R. Ortale. A bayesian hierarchical approach for exploratory analysis of communities and roles in social networks. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, ASONAM '12, 2012.

[7] M. G. Everett and S. P. Borgatti. Regular equivalence - general-theory. *Journal of Mathematical Sociology*, 19(1):29–52, 1994.

[8] S. Fortunato. Community detection in graphs. *Physics Reports*, 486:3-5, 2010.

[9] D. M. B. T. L. Griffiths and M. I. J. J. B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *NIPS*, volume 16, page 17, 2004.

[10] K. Henderson, B. Gallagher, T. Eliassi-Rad, H. Tong, S. Basu, L. Akoglu, D. Koutra, C. Faloutsos, and L. Li. Rolx: structural role extraction & mining in large graphs. In *KDD*, 2012.

[11] R. Jin, V. E. Lee, and H. Hong. Axiomatic ranking of network role similarity. In *KDD '11*, 2011.

[12] X. Jin, S. Spangler, R. Ma, and J. Han. Topic initiator detection on the world wide web. In *WWW '10*, 2010.

[13] S. N. Kim and M.-Y. Kan. Re-examining automatic keyphrase extraction approaches in scientific articles. In *Proc. Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, MWE '09, 2009.

[14] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[15] A. Lancichinetti, S. Fortunato, and J. Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015, 2009.

[16] W. Li and A. McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *ICML '06*, 2006.

[17] Y. Liu, A. Niculescu-Mizil, and W. Gryc. Topic-link lda: joint models of topic and author community. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, 2009.

[18] A. McCallum, X. Wang, and A. Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *Journal of Artificial Intelligence Research (JAIR)*, 30:249–272, 2007.

[19] D. Mimno, W. Li, and A. McCallum. Mixtures of hierarchical topics with pachinko allocation. In *ICML '07*, 2007.

[20] S. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[21] G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[22] J. Scripps, P.-N. Tan, and A.-H. Esfahanian. Exploration of link structure and community-based node roles in network analysis. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ICDM '07, 2007.

[23] H. M. Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 977–984, New York, NY, USA, 2006. ACM.

[24] 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.

[25] C. Wang, J. Han, Y. Jia, J. Tang, D. Zhang, Y. Yu, and J. Guo. Mining advisor-advisee relationships from research publication networks. In *KDD'10*, 2010.

[26] X. Wang, A. McCallum, and X. Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ICDM '07, pages 697–702, Washington, DC, USA, 2007.

[27] N. Yuruk, M. Mete, X. Xu, and T. A. J. Schweiger. Ahscan: Agglomerative hierarchical structural clustering algorithm for networks. In *Proceedings of the 2009 International Conference on Advances in Social Network Analysis and Mining*, ASONAM '09, 2009.

[28] D. Zhou, E. Manavoglu, J. Li, C. L. Giles, and H. Zha. Probabilistic models for discovering e-communities. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, 2006.

[29] W. Zhou, H. Jin, and Y. Liu. Community discovery and profiling with social messages. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, 2012.