A Phrase Mining Framework for Recursive Construction of a Topical Hierarchy

Chi Wang, Marina Danilevsky, Nihit Desai,
Yinan Zhang, Phuong Nguyen, Thrivikrama Taula, Jiawei Han
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL, USA
{chiwang1,danilev1,nhdesai2,yzhng103,pvnguye2,thrivik2,hanj}@illinois.edu

ABSTRACT

A high quality hierarchical organization of the concepts in a dataset at different levels of granularity has many valuable applications such as search, summarization, and content browsing. In this paper we propose an algorithm for recursively constructing a hierarchy of topics from a collection of content-representative documents. We characterize each topic in the hierarchy by an integrated ranked list of mixed-length phrases. Our mining framework is based on a phrase-centric view for clustering, extracting, and ranking topical phrases. Experiments with datasets from different domains illustrate our ability to generate hierarchies of high quality topics represented by meaningful phrases.

Categories and Subject Descriptors

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

Keywords

Topic Modeling, Ontology Learning, Network Analysis,
Keyphrase Extraction, Keyphrase Ranking

1. INTRODUCTION

A high quality hierarchical organization of the concepts in a dataset at different levels of granularity has many valuable applications in the areas of summarization, search and browsing. A student could familiarize herself with a new domain by perusing such a hierarchy and quickly learning the domain's topics. Or, a researcher could discover which terminology phrases are representative of his topic of interest, assisting his search for relevant work done by other colleagues and potentially discovering subtopics to focus on. We are therefore motivated to create a robust framework for constructing high quality topical hierarchies from texts in different domains.

For this task, we work with datasets of short texts - in particular, *content-representative* documents. 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. The terms in a content-representative document (the title) can therefore be thought of as probabilistic priors for which terms are the most likely to generate phrases representative of the full document (the paper). Our goal is to represent the topics of a collection, so content-representative documents are cleaner, simpler to use, and more likely to be available than full documents, while yielding the desired result.

Our framework therefore aims to construct a hierarchy where each topic is represented by a ranked list of topical phrases, such that a child topic is a subset of its parent topic. For example, the topic of query processing and optimization may be described by the phrases {'query processing', 'query optimization',...}, while its parent topic of general problems in databases may be described by {'query processing', 'database systems', 'concurrency control',...}

Our goal has several challenges. Topical phrases that would be regarded as high quality by human users are likely to vary in length (e.g., 'support vector machines' and 'feature selection' would both be good phrases for a topic about machine learning). Existing phrase extraction and ranking approaches are term-centric and cannot directly compare such mixed-length phrases, highlighting the need for a phrase-centric approach. Globally frequent phrases are not assured to be good representations for individual topics, demonstrating the need to infer the frequency of phrases in each topic. Finally, we must be able to recursively estimate each phrase's topical frequency for subtopics, in order to construct a topical hierarchy.

In this work we present CATHY (Constructing A Topical HierarchY), a phrase-centric framework for topical hierarchy generation via recursive clustering and ranking. The main features of our framework are as follows:

• *Phrase-centric approach*: We employ topic analysis and frequent pattern mining to estimate the topical frequency for phrases. By using a phrase-centric topical frequency measure instead of a unigram-centric measure, we are able to mine and rank higher quality phrases for each topic.

• *Ranking of topical phrases*: We define a topical keyphrase ranking function which implements the four criteria that intuitively represent high quality topical phrases: coverage, purity, phraseness, and completeness. Because the input to

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

our ranking function is phrases and their topical frequencies, instead of unigrams that must somehow be combined, we can directly compare phrases of different lengths and yield an integrated ranking of mixed-length phrases.

- *Recursive clustering for hierarchy construction*: Our topic inference is based on term co-occurrence network clustering. For any topic, we can extract its representative subnetwork, and recursively apply CATHY to discover subtopics.

2. PROBLEM FORMULATION

Traditionally, a phrase is defined as a consecutive sequence of terms, or unigrams. However, as discussed in [16] 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. Our framework can work with alternative definition of phrases as well, such as traditonally defined consecutive ngrams.

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 topical hierarchy.

Definition 2 (Topical Hierarchy). *A 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 a ranked list of phrases $\{\mathcal{P}^t, r^t(\mathcal{P}^t)\}$, where $\mathcal{P}^t$ is the set of phrases for topic t, and $r^t(P^t)$ is the ranking score for the phrases in topic t. For every non-leaf topic t in the tree, all of its subtopics comprise its children set $C^t = \{z \in \mathcal{T}, par(z) = t\}$. A phrase can appear in multiple topics, though it will have a different ranking score in each topic.*

To construct a topical hierarchy, we must soft cluster phrases into a hierarchy and find representative phrases for each topic. As an example, consider the task of judging what constitutes high quality phrases for various topics in computer science. There are four criteria for judging the quality of a phrase:

- **Coverage:** A representative phrase for a topic should cover many documents within that topic. *Example: 'information retrieval' has better coverage than 'cross-language information retrieval' in the Information Retrieval topic.*
- **Purity:** A phrase is pure in a topic if it is only frequent in documents belonging to that topic and not frequent in documents within other topics. *Example: 'query processing' is more pure than 'query' in the Databases topic.*
- **Phraseness:** 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. *Example: 'active learning' is a better phrase than 'learning classification' in the Machine Learning topic.*
- **Completeness:** A phrase is not complete if it is a subset of a longer phrase, in the sense that it rarely occurs in a document without the presence of the longer phrase. *Example: 'support vector machines' is a complete phrase, whereas 'vector machines' is not because 'vector machines' is almost always accompanied by 'support' in documents.*

The measures which represent these criteria can all be characterized by an important concept: topical frequency.

Definition 3 (Topical Frequency). *The topical frequency $f_t(P)$ of a phrase is the count of the number of times the phrase is attributed to topic t. For the root node o, $f_o(P) = f(P)$. For each topic node in the hierarchy, with subtopics C^t, $f_t(P) = \sum_{z \in C^t} f_z(P)$, i.e., the topical frequency is equal to the sum of the sub-topical frequencies.*

Table 1 illustrates an example of estimating topical frequency for phrases in a computer science topic that has 4 subtopics. The phrase 'support vector machines' is estimated to belong entirely to the Machine Learning (ML) topic with high frequency, and therefore is a candidate for a high quality phrase. However, 'social networks' is fairly evenly distributed among three topics, and is thus less likely to be a high quality phrase. Section 3.3 discusses how such candidate phrases are actually ranked, using measures based on estimated topical frequency.

Table 1: Example of estimating topical frequency. The topics are assumed to be inferred as machine learning, database, data mining, and information retrieval from the collection

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

3. CATHY FRAMEWORK

In order to estimate the topical frequency for each phrase, we need to infer the dataset's topics. We perform topic inference and estimate topical frequency by analyzing our dataset's term co-occurrence network.

Formally, every topic node t in the topical hierarchy is associated with a term co-occurrence network G^t. The root node o is associated with the term co-occurrence network G^o constructed from the collection of content-representative documents. G^o 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 every non-root node $t \neq o$, we construct a subnetwork G^t by clustering the term co-occurrence network of its parent $par(t)$. G^t has all of the nodes from $G^{par(t)}$, but only those links belonging to the particular subtopic t.

We chose to use term co-occurrence network for topic analysis instead of document-term topic modeling because the it naturally supports recursive mining: the clustering result for one topic can be used as the input when further partitioning the topic into subtopics. The CATHY framework generates a topical hierarchy in a top-down, recursive way:

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

Step 2. For a topic t, cluster the term co-occurrence network G^t into subtopic subnetworks $G^z, z \in C^t$, and estimate the subtopical frequency for its subtopical phrases using a generative model.

Step 3. For each topic $z \in C^t$, extract candidate phrases based on estimated topical frequency.

Step 4. For each topic $z \in C^t$, rank the topical phrases using a unified ranking function based on topical frequency. Phrases of different lengths are directly compared, yielding an integrated ranking.

Step 5. Recursively apply Steps 2 - 5 to each subtopic $z \in C^t$ to construct the hierarchy in a top-down fashion.

3.1 Clustering: Estimating Topical Frequency

We first introduce the process of clustering for one 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 such as the Bayesian Information Criterion [27].

In the term co-occurrence network G^t, we assume every co-occurrence of two terms w_i and w_j is attributed to a topic $z \in C^t = \{1, \dots, k\}$. We represent the total link frequency $e_{i,j}$ between w_i and w_j as a summation of the number of links between w_i and w_j in each of the k topics: $e_{ij} = \sum_{z=1}^{k} e_{ij}^z$. The goal is thus to estimate e_{ij}^z for $z = 1 \dots k$, which is unlike most network analysis approaches.

We develop a generative model of the term co-occurrence network, and use it to estimate topical frequency $f_z, z \in C^t$.

3.1.1 A generative model for term co-occurrence network analysis

To generate a topic-z link, we first generate one end node w_i following a multinomial distribution $p(w_i|z) = \theta_i^z$, and then generate the other end node w_j with the same multinomial distribution $p(w_j|z) = \theta_j^z$. The probability of generating a topic-z link (w_i, w_j) is therefore $p(w_i|z)p(w_j|z) = \theta_i^z\theta_j^z$.

With this generative assumption for each individual link, we can derive the distribution of topical frequency for any two terms (w_i, w_j). If we repeat the generation of topic-z links for ρ_z iterations, then the chance of generating a particular topic-z link between w_i and w_j can be modeled as a Bernoulli trial with success probability $\theta_i^z\theta_j^z$. When ρ_z is large, the total number of successes e_{ij}^z approximately follows a Poisson distribution $Pois(\rho_z\theta_i^z\theta_j^z)$.

Now we can write down the generative model for random variables e_{ij}^z with parameters ρ_z, θ^z.

$$e_{ij}^z \sim Poisson(\rho_z\theta_i^z\theta_j^z), z = 1, \dots, k \quad (1)$$

$$\sum_{i=1}^{|W|} \theta_i^z = 1, \theta_i^z \geq 0, \rho_z \geq 0 \quad (2)$$

The constraints guarantee a probabilistic interpretation. According to the *expectation* property of the Poisson distribution, $E(e_{ij}^z) = \rho_z\theta_i^z\theta_j^z$. Also, according to the *additive* property of expectations,

$$E(\sum_{i,j} e_{ij}^z) = \sum_{i,j} \rho_z\theta_i^z\theta_j^z = \rho_z \sum_i \theta_i^z \sum_j \theta_j^z = \rho_z$$

In other words, ρ_z is the total expected number of links in topic z.

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

$$e_{ij} = \sum_{z=1}^{k} e_{ij}^z \sim Poisson(\sum_{z=1}^{k} \rho_z\theta_i^z\theta_j^z) \quad (3)$$

So given the model parameters, the probability of all observed links is

$$p(\{e_{ij}\}|\theta, \rho) = \prod_{w_i, w_j \in W} p(e_{ij}|\theta_i, \theta_j, \rho)$$

$$= \prod_{w_i, w_j \in W} \frac{(\sum_{z=1}^{k} \rho_z\theta_i^z\theta_j^z)^{e_{ij}} \exp(-\sum_{z=1}^{k} \rho_z\theta_i^z\theta_j^z)}{e_{ij}!} \quad (4)$$

In this model, the observed information is the total number of links between every pair of nodes, including zero links and self-links. The parameters which must be learned are the role of each node in each topic $\theta_i^z, w_i \in W, z = 1, \dots, k$, and the expected number of links in each topic ρ_z. The total number of free parameters to learn is therefore $k|W|$. We learn the parameters by the *Maximum Likelihood* (ML) principle: find the parameter values that maximize the likelihood in Eq. (4). We use an Expectation-Maximization (EM) algorithm that can iteratively infer the model parameters:

$$\mathrm{E-step:} \quad \hat{e}_{ij}^z = e_{ij} \frac{\rho_z\theta_i^z\theta_j^z}{\sum_{t=1}^{k} \rho_t\theta_i^t\theta_j^t} \quad (5)$$

$$\mathrm{M-step:}$$

$$\rho_z = \sum_{i,j} \hat{e}_{ij}^z \quad (6)$$

$$\theta_i^z = \frac{\sum_j \hat{e}_{ij}^z}{\rho_z} \quad (7)$$

Intuitively, the E-step calculates the expected number of links $\hat{e}_{ij}^z$ in each topic z between the terms w_i and w_j: the ratio of $\hat{e}_{ij}^z$ to e_{ij} is proportional to its Poisson parameter $\rho_z\theta_i^z\theta_j^z$. The M-step calculates the ML parameter estimates: θ_i^z is the ratio of the total number of links in topic z where one end node is w_i and ρ_z, which is the sum of the total expected number of links in topic z.

We update $\hat{e}_{ij}^z, \theta_i^z, \rho_z$ in each iteration. Note that if $e_{ij} \notin E$, we do not need to calculate $\hat{e}_{ij}^z$ because it equals 0. Therefore, the time complexity for each iteration is $O((|E| + |V|)k) = O(|E|k)$. Like other EM algorithms, the solution converges to a local maximum and the result may vary with different initializations. The EM algorithm may be run multiple times with random initializations to find the solution with the best likelihood. We empirically find that the EM algorithm generally requires hundreds of iterations to converge, although we can improve the efficiency with some acceleration tricks. For example, we do not need to update a parameter in each iteration if it converges before the whole model converges. Similar tricks are used in other generative models such as [2], and we omit the details here.

It is important to note that our method naturally supports top-down hierarchical clustering. To further discover subtopics of a topic, we can extract the subnetwork where $E^z = \{\hat{e}_{ij}^z | \hat{e}_{ij}^z \geq 1\}$ (expected number of links attributed to that topic, ignoring values less than 1) and then apply the same generative model on the subnetwork. This process can be recursively repeated until the desired hierarchy is constructed.

3.1.2 Topical frequency estimation

Using the learned model parameters, we can estimate the topical frequency for a phrase $P = \{w_{x_1} \dots w_{x_n}\}$:

$$f_z(P) = f_{par(z)}(P) \frac{\rho_z \prod_{i=1}^{n} \theta^z_{x_i}}{\sum_{t \in C^{par(z)}} \rho_t \prod_{i=1}^{n} \theta^t_{x_i}} \tag{8}$$

This estimation is based on two assumptions: i) when generating a topic-z phrase of length n, each of the n terms is generated with the multinomial distribution θ^z, and ii) the total number of topic-z phrases of length n is proportional to ρ_z. It is easy to see that when $n = 2$, $f_z(\{w_i, w_j\})$ reduces to $\hat{e}^z_{ij}$.

3.2 Topical Phrase Extraction

Since we define phrases to be sets of frequent terms, we develop an algorithm to mine frequent topical patterns. The goal is to extract patterns with topical frequency larger than some threshold *minsup* for every topic z. In contrast to traditional frequent pattern mining problem, the topical frequency of each pattern is unknown and must be estimated. The results from the clustering step in Section 3.1 are necessary for our estimation.

To extract topical frequent patterns, one can first mine all frequent patterns with a traditional pattern mining algorithm such as Apriori [1] or FP-growth [15], and then filter them using the topical frequency estimation using Eq. (8). The following two properties of topical frequency can be further exploited to speed up this step:

Property 1. *A phrase's topic-z frequency has an upper bound of the topic-z frequency of any of its subphrases.*

Property 2. *A phrase's topic-z frequency has an upper bound* $f_{par(z)}(P') \frac{\rho_z \prod_{i=1}^{n} \theta^z_{x_i}}{\sum_{t \in C^{par(z)}} \rho_t \prod_{i=1}^{n} \theta^t_{x_i}}$ *, where* $P' \subset P$ *is any subphrase of* P.

Note that for only the top level topics $z \in C^o$, the parent topical frequency $f_{par(z)}(P)$ is equal to $f(P)$ and must be counted from the text. However, for all lower levels, the parent topical frequency $f_{par(z)}(P)$ was already calculated when the parent topic was generated, and therefore never needs to be counted.

One problem with the extracted frequent term sets is that every subset of a frequent phrase is also a frequent phrase. However, some of these subphrases should be removed according to the completeness criterion described in Section 2 (e.g., 'vector machines'). To remove incomplete phrases, we adapt the notions of 'closed patterns' and 'maximal patterns' [14]. A maximal pattern has no frequent supersets, and a closed pattern has no supersets with the same frequency. For our task, retaining only maximal phrases is too aggressive since a long frequent phrase will override all of its subphrases. However, retaining all closed phrases only removes a phrase if it has a superphrase with exactly the same frequency, which is rare.

We therefore try to find a middle ground by unifying the definitions of maximal patterns and closed patterns together with a tunable parameter. For each topic, we remove a phrase P if there exists a frequent phrase P', such that $P \subset P', f_z(P') \geq \gamma f_z(P)$. The remaining patterns are referred to as γ-*maximal patterns* $(0 \leq \gamma \leq 1)$. When $\gamma = 1$, this is equivalent to a closed pattern, and when $\gamma = 0$, this is a maximal pattern. We empirically set $\gamma \approx 0.5$, which removes a phrase if its topical frequency is no more than twice of some superphrase. In other words, if a phrase co-occurs with a superphrase more than half the time, we consider that it is subsumed by the superphrase, and should be removed. According to Property 1, pruning can be performed by comparing the frequency of a length-n phrase with all of its length-$(n + 1)$ superphrases. The collection of all of the γ-maximal phrases of a topic z forms the candidate phrase set $\mathcal{P}^z$.

3.3 Ranking

As discussed in Section 2, topical phrases in $\mathcal{P}^z$ are ranked according to four criteria: coverage, purity, phraseness, and completeness. The last criterion is already employed as a filter for the phrase extraction step, parameterized by γ. So we now combine the remaining three criteria into a ranking function using a probabilistic modeling approach.

The ranking function should be able to directly compare keyphrases of mixed lengths, which we refer to as having the **comparability property**. For example, the keyphrases 'classification,' 'decision trees,' and 'support vector machines' should all be ranked highly in the integrated list of keyphrases for the Machine Learning topic, in spite of varying in length. Traditional probabilistic modeling approaches, such as language models or topic models do not exhibit the comparability property. These approaches simply find that longer n-grams have a much lower probability than shorter ones, because the probabilities of seeing every possible unigram sum to 1, and so do the probabilities of seeing every possible bigram, trigram, etc. However, the total number of possible n-grams grows following a power law $(O(|V|^n))$. While previous work has used various heuristics to correct for this bias during post-processing steps by, for example, using a penalization term with respect to the phrase length [29, 34], our approach is cleaner and more principled.

We propose a different ranking model that exhibits the comparability property. The key idea is to consider the *occurrence probability* of 'seeing a phrase p in a random document with topic t.' With this definition, the events of seeing n-grams of various lengths in a document are no longer mutually exclusive, and therefore the probabilities no longer need to sum to 1.

We construct estimations for occurrence probability and two *contrastive probabilities* that will be used to compare against the occurrence probablity. We use m_z to denote the number of documents that contain at least one frequent topic-z phrase. Similarly, we use m_Z to denote the number of documents that contain at least one frequent topic-z phrase for some topic $z \in Z$. We can then calculate the occurrence probability of a phrase P conditioned on topic z:

$$p(P|z) = \frac{f_z(P)}{m_z} \tag{9}$$

The *independent contrastive probability* is the probability of independently seeing every term in phrase $P = \{w_{x_1}, \dots w_{x_n}\}$ conditioned on topic z:

$$p_{indep}(P|z) = \prod_{i=1}^{n} p(w_{x_i}|z) = \prod_{i=1}^{n} \frac{f_z(w_{x_i})}{m_z} \tag{10}$$

and the *mixture contrastive probability* is the probability of a phrase P conditioned on a mixture of multiple sibling topics $Z \subset C^{par(z)}, Z \supsetneqq \{z\}$:

$$p(P|Z) = \frac{\sum_{t \in Z} f_t(P)}{m_Z} \tag{11}$$

We can now define the three remaining ranking criteria: coverage, purity, and phraseness. The coverage of a phrase is directly quantified by $p(P|z)$. The phraseness can be measured by the log ratio of the occurrence probability to the independent contrastive probability $\log \frac{p(P|z)}{p_{indep}(P|z)}$. The purity can be measured by the log ratio of the occurrence probability and the mixture contrastive probability $\log \frac{p(P|z)}{p(P|Z)}$. The definition of purity is configurable by altering the makeup of the topic mixture Z. For example, using the mixture of all the sibling topics $C^{par(z)}$ 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:

$$r^z(P) = p(P|z) \left(\log \frac{p(P|z)}{p(P|Z)} + \omega \log \frac{p(P|z)}{p_{indep}(P|z)} \right) \tag{12}$$

where ω controls the importance of the phraseness criterion. This formulation of the ranking function has several desirable characteristics:

- 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 more topic-independent but common phrases to be ranked higher. We therefore restrict $\omega \in [0, 1]$ because our task requires topic-related phrases to be highly ranked.
- The ranking function can also be nicely represented as a pointwise Kullback-Leibler (KL) divergence in an information theoretic framework. Pointwise KL divergence is a distance measure between two probabilities. It is more robust than pointwise mutual information because the former also considers absolute probability. In pointwise KL divergence, the relative difference between probabilities must be supported by a sufficiently high absolute probability. The product $p(P|z) \log \frac{p(P|z)}{p(P|Z)}$ is equivalent to the pointwise KL divergence between the probabilities of $p(P|z)$ and $p(P|Z)$. Likewise, $p(P|z) \log \frac{p(P|z)}{p_{indep}(P|z)}$ is equivalent to the pointwise KL divergence between the probabilities of $p(P|z)$ under different independence assumptions. Therefore, Eq. (12) can also be interpreted as a weighted summation of two pointwise KL divergence metrics.

4. RELATED WORK

4.1 Ontology learning

With respect to the goal of our framework, our work is broadly related to ontology learning. Topical hierarchies, concept hierarchies, ontologies, *etc.*, provide a hierarchical organization of data at different levels of granularity, and have many important applications, *e.g.*, in web search and browsing [11]. There has been a substantial amount of research on ontology learning from text, though it remains a challenging problem (see [32] for a recent survey). The techniques can be broadly categorized as statistics-based or linguistic-based. Most studies aim to mine subsumption ('is-a') relationships [17], either by using lexico-syntactic patterns (*e.g.*, 'x is a y') [28, 25] or statistics-based approaches [33, 9]. Our definition of a topical hierarchy is clearly distinct from a subsumption hierarchy. Chuang and Chien [7] and Liu *et al.* [20] generate taxonomy of given keyword phrases by hierarchical clustering techniques, with the help of knowledge bases and search engine. If our work were to be broadly viewed as an ontology-learning approach, we use statistics-based techniques, without resorting to external knowledge resources such as WordNet or Wikipedia.

4.2 Topical keyphrase extraction and ranking

The nature of our technique is related to topical keyphrase extraction and ranking. Keyphrases are traditionally extracted as ngrams using statistical modeling [31], or as noun phrases using natural language processing techniques [3]. We mainly review the related work in extracting topical keyphrases from document collections rather than keyphrase extraction from single documents. The state-of-the-art approaches to unsupervised keyphrase extraction have generally been graph-based, unigram-centric ranking methods, which first extract unigrams and rank them for each topic, and finally combine them into keyphrases [21, 34]. Some previous methods have used clustering techniques on word graphs with the help of external knowledge bases such as Wikipedia for keyphrase extraction [13]. Tomokiyo and Hurst [29] require a document collection with known topics as input and train a language model to define their ranking criteria. Mei *et al.* [22] use keyphrase extraction techniques to discover labels for topics.

In contrast to all of these studies, we relax the restriction that a phrase must be a consecutive n-gram, and instead use document colocation - which is effective due to the nature of the short, content-representative document collections which our framework expects. We also do not employ any NLP techniques to parse the text of our datasets.

4.3 Topic modeling

Our study is also related to topic modeling. Topic modeling techniques such as Latent Dirichlet Allocation [5] take documents as input, model them as mixtures of different topics, and output word distributions for each topic. Some extensions have been developed to discover topical phrases comprised of consecutive words [30, 31, 19, 4]. They cannot find hierarchical topics, and their definition of phrases is more restrictive. Several other extensions can model the hierarchical dependency of unigram-based topics [12, 18, 24]. It is challenging to apply these techniques to our scenario because: i) since our text is sparse, the distribution estimates are quite brittle [10] when calculating multiple topic levels, and ii) these methods compute the entire hierarchy simultaneously and do not support recursive discovery of subtopics from a topic.

5. EXPERIMENTS

In this section we first introduce the datasets and methods we used for comparison. We then describe our 3-part evaluation: i) we conduct a user study with 'intruder detection' tasks to evaluate hierarchy quality; ii) we use category-labeled data to evaluate the mutual information between phrase-represented topics and known topical divisions; and iii) we present several case studies.

5.1 Datasets

We analyze our performance on two datasets:[1]

- **DBLP**. We collected a set of titles of recently published computer science papers in the areas related to Databases, Data Mining, Information Retrieval, Machine Learning, and Natural Language Processing. These titles come from DBLP[2], a bibliography website for computer science publications. We minimally pre-processed the dataset by removing all stopwords from the titles, resulting in a collection of 33,313 titles consisting of 18,598 unique terms.
- **Library.** We obtain titles of books from the *University of Illinois Library* catalogue database in 6 general categories: Architecture, Literature, Mass Media, Motion Pictures, Music, and Theater. We pre-processed the titles by removing all stopwords and terms with frequency < 5 in the dataset. We also remove titles over 10 words in length, and titles not in English. The resulting dataset contains 33,372 titles consisting of 3,556 unique terms.

5.2 Methods for Comparison

As the topical hierarchy construction problem setting that we study is new, there are no directly comparable algorithms. We implement several methods:

SpecClus: As one baseline, we implement a common framework of clustering-based ontology construction, which first extracts all concepts from the text and then hierarchically clusters them. We adapt this to our setting by first mining all frequent phrases using FP-growth [15], a typical pattern mining algorithm. We then implement spectral clustering [26] for the clustering step, where the similarity metric between two phrases is their co-occurrence count in the dataset. This approach uses K-means to perform hard clustering after computing a spectral embedding of the similarity graph. Finally, we rank phrases in each cluster based on their distance from the cluster center. In order to go down in the hierarchy we recursively perform the same clustering and ranking on each cluster of phrases.

hPAM: As a second baseline, we use a state-of-the-art hierarchical topic modeling approach: the hierarchical Pachinko Allocation Model [24]. hPAM takes documents as input and outputs a specified number of supertopics and subtopics, as well as the associations between them. However, it builds a hierarchy for 3 levels simultaneously, not recursively, so we only generate a hierarchy with 3 levels.[3]

hPAM$_{\mathbf{rr}}$: For each topic, hPAM outputs a multinomial distribution over unigrams. These distributions can be used to calculate the coverage and purity measures in our ranking function (phraseness and completeness do not matter when all candidate phrases are unigrams). We therefore also implement a method that reranks the unigrams in each topic generated by hPAM, with our ranking function adopting the distribution learned by hPAM. We refer to the result as hPAM$_{\mathrm{rerank}}$, or hPAM$_{\mathrm{rr}}$. Note that we cannot rerank SpecClus because it does not generate a probability distribution that can be input into our ranking function.

CATHY$_{\mathbf{cp}}$: In this version of CATHY the ranking function only considers the coverage and purity criteria, and not phraseness or completeness (γ=1, ω=0). This allows us to more closely compare the performance of our clustering and mining step with hPAM, using hPAM$_{\mathrm{rr}}$.

CATHY: For evaluation we set *minsup*=5, γ=ω=0.5 for phraseness and completeness criteria, and we use the strong definition of purity, as discussed in Section 3.3 (Refer to [8] for more thorough studies of the effects of the ranking function's parameters.)

5.3 Topical Hierarchy of DBLP Paper Titles

Our first evaluation assesses the ability of our method to construct topical phrases that appear to be high quality to human judges, via a user study. We construct hierarchies with 4 levels from the DBLP dataset. For simplicity we set the number of subtopics for the root node to be 5, for all other non-leaves to be 4, for all of the methods. Since hPAM and hPAM$_{\mathrm{rr}}$ only construct 3 levels of the hierarchy, we compare the 3-level hierarchies across all methods, and the full hierarchies for the 3 methods which constructed them.

In the following subsections, we present a sample of the hierarchies actually generated by these methods and encountered by participants in the user study. We then explain the details of our user study, and present quantitative results.

5.3.1 Qualitative Results

Figure 1 shows a subset of hierarchies constructed by CATHY and the two baselines, SpecClus and hPAM. In general, CATHY constructs high quality phrases, representing the areas and subareas on both levels. hPAM outputs unigrams that are fair at conveying the top-level topics when considered jointly, but independently are topic-ambiguous (e.g., 'services' for IR). hPAM's second level subtopics are generally more difficult to interpret, and some parent-child relationships are not clearly observed. SpecClus tends to generate phrases with good purity but unsatisfactory coverage and phraseness (e.g., 'querying spatial' for DB).

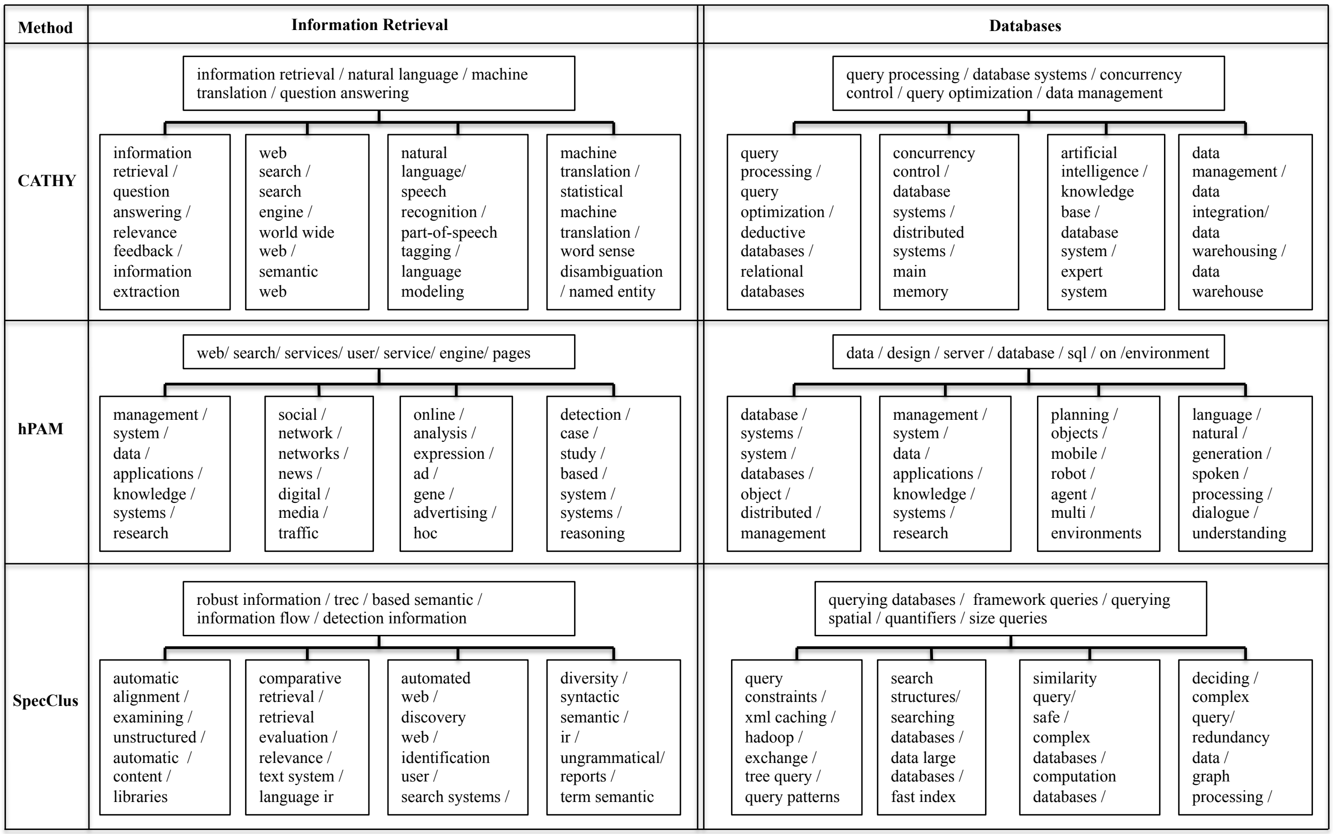
5.3.2 Word and Topic Intrusion User Study

To quantitatively measure topical phrase quality, we invited people to judge the topical phrases generated by the different methods. Since the DBLP dataset generates topics in computer science, we recruited 9 computer science graduate students - who could thus be considered to be very knowledgeable judges - for a user study. We first describe

[1] The datasets are available at http://illimine.cs.illinois.edu/cathy

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

[3] hPAM has several parameters. Our tuning shows that the optimal values of mixture prior between supertopic and subtopic are 1.5 and 1.0 respectively. The mixture prior over topics in the same level is optimal at 1.0 for both levels. The optimal prior for topic distribution over terms is 0.01.

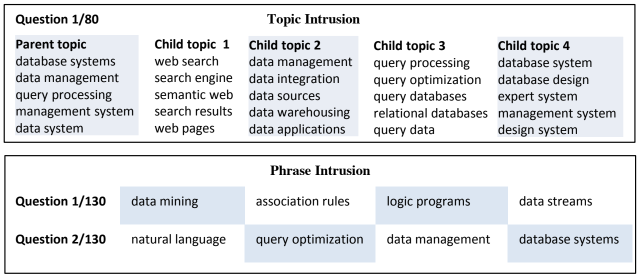


Method	Information Retrieval	Databases
CATHY	information retrieval / natural language / machine translation / question answering information retrieval / question answering / relevance feedback / information extraction web search / search engine / world wide web / semantic web natural language/ speech recognition / part-of-speech tagging / language modeling machine translation / statistical machine translation / word sense disambiguation / named entity	query processing / database systems / concurrency control / query optimization / data management query processing / query optimization / deductive databases / relational databases concurrency control / database systems / distributed systems / main memory artificial intelligence / knowledge base / database system / expert system data management / data integration/ data warehousing / data warehouse
hPAM	web/ search/ services/ user/ service/ engine/ pages management / system / data / applications / knowledge / systems / research social / network / networks / news / digital / media / traffic online / analysis / expression / ad / gene / advertising / hoc detection / case / study / based / system / systems / reasoning	data / design / server / database / sql / on /environment database / systems / system / databases / object / distributed / management management / system / data / applications / knowledge / systems / research planning / objects / mobile / robot / agent / multi / environments language / natural / generation / spoken / processing / dialogue / understanding
SpecClus	robust information / trec / based semantic / information flow / detection information automatic alignment / examining / unstructured / automatic / content / libraries comparative retrieval / retrieval evaluation / relevance / text system / language ir automated web / discovery web / identification user / search systems / diversity / syntactic semantic / ir / ungrammatical/ reports / term semantic	querying databases / framework queries / querying spatial / quantifiers / size queries query constraints / xml caching / hadoop / exchange / tree query / query patterns search structures/ searching databases / data large databases / fast index similarity query/ safe / complex databases / computation databases / deciding / complex query/ redundancy data / graph processing /

Figure 1: Each method generates a hierarchy. For each method, we show the subtrees rooted at Level 2 that are the most likely to represent the topics of Information Retrieval and Databases. The ordering of words in each phrase are determined by the most frequent ordering in the documents, and two phrases only differeing in plural/single forms are shown only once.

the two tasks administered in the user study, and then discuss the obtained results.

In order to evaluate the quality of the generated topical phrases, we adapt two tasks from Chang et al. [6], who were the first to explore human evaluation of topic models. Our first task is Topic Intrusion, which tests the quality of the parent-child relationships in the generated hierarchies. Our second task is Phrase Intrusion, which evaluates how well the hierarchies are able to separate phrases in different topics. Both tasks are depicted in Figure 2.



Question 1/80 — Topic Intrusion

Parent topic	Child topic 1	Child topic 2	Child topic 3	Child topic 4
database systems	web search	data management	query processing	database system
data management	search engine	data integration	query optimization	database design
query processing	semantic web	data sources	query databases	expert system
management system	search results	data warehousing	relational databases	management system
data system	web pages	data applications	query data	design system

Phrase Intrusion

Question 1/130	data mining	association rules	logic programs	data streams
Question 2/130	natural language	query optimization	data management	database systems

Figure 2: Examples of user study questions. In the Topic Intrusion task (left), participants are asked to select which child topic does not belong to the given parent topic (Child topic 1). In the Phrase Intrusion task (right), participants are asked to select which phrase does not belong with the others (Question 1: 'logic programs'; Question 2: 'natural language')

Topic Intrusion Task: Participants are shown a parent topic t and T candidate child topics. $T-1$ 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 phrases. Participants are asked to select the intruder child topic, or to indicate that they are unable to make a choice.

Phrase Intrusion Task: Participants are shown T phrases. $T-1$ of the phrases come from the same topic and the remaining phrase is from a sibling topic. Each phrase is a top-5 ranked phrase in the topic which it represents. Participants are asked to select the intruder phrase, or to indicate that they are unable to make a choice.

For the user study we set $T = 4$, and asked participants 80 Topic Intrusion questions and 130 Phrase Intrusion questions. Questions are generated from the hierarchies constructed by each of the methods. We sample questions from each hierarchy in a uniform way, drawing equally from all topics in each level.

We then calculate the agreement of the user choices with the actual hierarchical structure constructed by the various methods. We consider a higher match between a given hierarchy and user judgment to imply a higher quality hierarchy. For each method, we report the average percent of questions answered 'correctly' (matching the method), as well as the average percent of questions that users were able to answer.

Since the hPAM and hPAM$_{\mathrm{rr}}$ hierarchies had one fewer level than other methods, we present two analyses. Table 2 presents the results from the full set of questions, except the questions generated by hPAM and hPAM$_{\mathrm{rr}}$ (4 Levels), and the results from only those questions taken from the shared levels of every method's hierarchies (3 Levels).

Table 2: User study results, for 3 level and 4 level hierarchies. Higher values indicate a higher quality constructed hierarchy

	Topic Intrusion		**Phrase Intrusion**	
3 Levels	Correct	Answered	Correct	Answered
hPAM	34.4%	75.6%	38.8%	78.9%
hPAM$_{\mathbf{rr}}$	32.2%	72.2%	47.8%	77.2%
SpecClus	38.9%	65.6%	36.1%	77.2%
CATHY$_{\mathbf{cp}}$	78.9%	**97.8%**	**57.8%**	**90.0%**
CATHY	**82.2%**	**98.8%**	**57.2%**	**88.9%**
4 Levels	Correct	Answered	Correct	Answered
SpecClus	34.4%	68.3%	32.9%	77.4%
CATHY$_{\mathbf{cp}}$	61.7 %	**96.7%**	**56.7%**	**88.5%**
CATHY	**78.3%**	**97.8%**	**54.1%**	**89.3%**

For the Topic Intrusion task, CATHY outperforms all non-CATHY methods by a large margin. CATHY does slightly better than CATHY$_{\rm cp}$ in the 3 level hierarchy, and is significantly better in the 4 level hierarchy, suggesting that participants found the phraseness and completeness criteria to be helpful. SpecClus slightly outperforms hPAM, because hPAM generates broad unigrams with good coverage, which makes the parent-child relationships difficult to identify, while SpecClus yields phrases with better purity which, when considered jointly, represent a topic more successfully. Participants answered more questions generated by the hPAM variations than by SpecClus, which, combined with the resulting accuracies, suggests that hPAM generated the least well-separated hierarchy (even with reranking).

CATHY and CATHY$_{\rm cp}$ outperform other methods comparably in the Phrase Intrusion task. As hPAM favors high coverage phrases which are often topic-ambiguous, it posted a low performance. hPAM$_{\rm rr}$ considers phrase purity as well as topical coverage, and therefore performs much better on this task. SpecClus favors purity, and thus is more likely to generate seemingly unrelated high ranked phrases in the same topic, which is reflected here by its poor performance. Once again, participants were more likely to answer questions generated by the CATHY variations than by any of the other three methods.

5.4 Topical Hierarchy of Book Titles

In this section, we work with the Library dataset. Since the book titles are labeled with their subjects, we examine how well a high quality topical phrase can predict its category, and vice versa. For this, we construct a hierarchy and measure the *coverage-conscious mutual information at* K ($CCMI_K$) of the labels with the top level branches. Our evaluation is based on [24] but we modify their definition of mutual information to also depend on coverage because we represent topics with phrases.

As we saw in Section 5.3 that CATHY generally performs equal to or better than CATHY$_{\rm cp}$, and hPAM$_{\rm rr}$ similarly outperforms hPAM, we simply compare the performances of CATHY, hPAM$_{\rm rr}$, and SpecClus, with 6 topics ($k = 6$). For each method, we do multiple runs for various values of K (the number of top-ranked phrases per topic considered). To calculate $CCMI_K$, we label each of the top K phrases per topic with the topic in which it is ranked highest. We then check if each title contains any of these top phrases. If so, we update the number of events "seeing a topic t and category c" for $t = 1 \dots k$, with the averaged count for all those labeled phrases in the title; otherwise we update the number of events "seeing a topic t and category c" for $t = 1 \dots k$ uniformly, where c is the category label for the title. Finally, we compute coverage-conscious mutual information at K:

$$CCMI_K = \sum_{t,c} p(t,c)\ log_2 \frac{p(t,c)}{p(t)p(c)}$$

Figure 3 shows $CCMI_K$ for each method, K$\in [1, 100]$. Since $CCMI_K$ considers the coverage of a phrase as well as its mutual information with a category, its value generally grows with K. Both CATHY and SpecClus demonstrate this by slowly improving over time, although CATHY is consistently much better at differentiating the categories as K increases. hPAM$_{\rm rr}$ prefers unigrams with high coverage, and thus hits an asymptote almost immediately because it is unable to improve on the performance of the first few phrases.

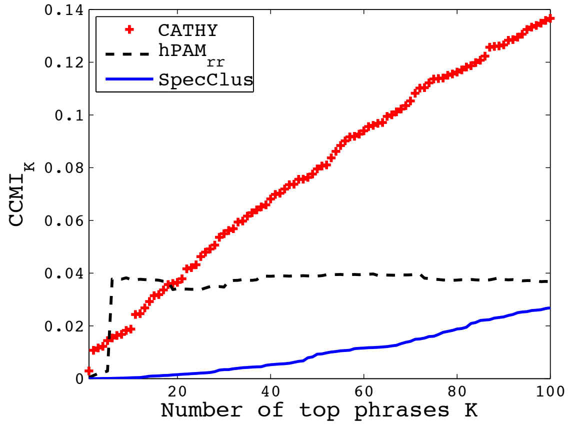


Figure 3: CCMI$_{\rm K}$ values for various methods (methods in legend are ordered by performance, high to low)

5.5 On Defining Term Co-occurrence

Term co-occurrence can be defined in many ways: two term may be said to co-occur if they appear in the same sentence, same paragraph, or in a window of N unigrams of each other [23]. Because we worked with collections of short texts, we consistently defined term co-occurrence for our framework to mean co-occurring in the same document. However, most traditional methods of keyphrase extraction only consider phrases to be sequences of terms which explicitly occur in the text. We ran a variation of CATHY which emulates this behavior by defining two terms to co-occur only if they are actually adjacent in the same title, and constructed a hierarchy on the DBLP dataset.

Using adjacency co-occurrence results in a sparser network and lowers the estimated phrase topical frequencies at every level. As can be seen in Figure 4, we observe lower quality phrases in the adjacency-based hierarchy (e.g., the topics which are supposed to be represented by the two rightmost children are very difficult to identify.)

6. CONCLUSION

In this work, we address the problem of constructing a topical hierarchy from short, content-representative texts, where topics are represented by ranked lists of phrases. We design a novel phrase mining framework to cluster, extract

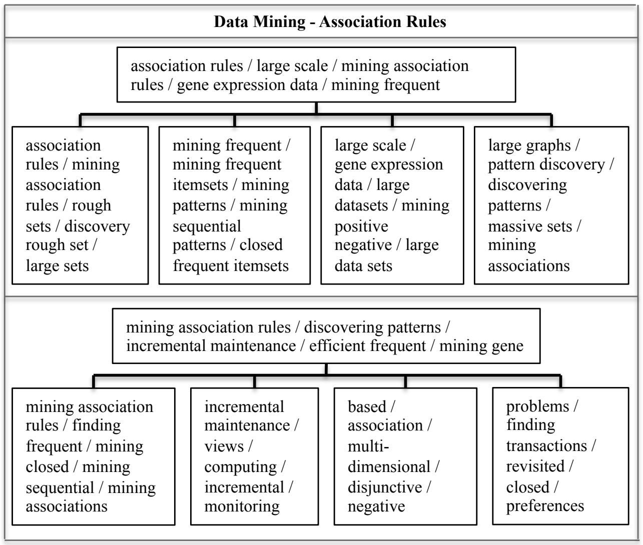

Data Mining - Association Rules
association rules / large scale / mining association rules / gene expression data / mining frequent
association rules / mining association rules / rough sets / discovery rough set / large sets
mining frequent / mining frequent itemsets / mining patterns / mining sequential patterns / closed frequent itemsets
large scale / gene expression data / large datasets / mining positive negative / large data sets
large graphs / pattern discovery / discovering patterns / massive sets / mining associations
mining association rules / discovering patterns / incremental maintenance / efficient frequent / mining gene
mining association rules / finding frequent / mining closed / mining sequential / mining associations
incremental maintenance / views / computing / incremental / monitoring
based / association / multi-dimensional / disjunctive / negative
problems / finding transactions / revisited / closed / preferences

Figure 4: A level 3 topic and its level 4 subtopics, from hierarchies constructed by CATHY on two different DBLP term co-occurrence networks: document-based co-occurrence (top) and adjacency-based co-occurrence (bottom). Document-based co-occurrence yields better quality phrases, especially at lower levels

and rank phrases which recursively discovers specific topics from more general ones, thus constructing a top-down topical hierarchy. A key aspect of our approach involves shifting from a unigram-centric to a phrase-centric view in order to consistently generate high caliber topics over multiple levels. By evaluating our approach on two datasets from different domains, we validate our ability to generate high quality, human-interpretable topic hierarchies.

We would like to extend our framework to incorporate supervised knowledge, either from user guidance or external knowledge bases. We would also like to explore integrating advanced text mining and natural language processing techniques, which would help us work with longer texts.

7. 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 Research PhD Fellowship. Marina Danilevsky was supported by a National Science Foundation Graduate Research Fellowship grant NSF DGE 07-15088. The authors wish to acknowledge the University of Illinois at Urbana-Champaign Library (`http://www.library.illinois.edu`), which provided support for this research.

8. REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB*, 1994.

[2] B. Ball, B. Karrer, and M. E. J. Newman. Efficient and principled method for detecting communities in networks. *Phys. Rev. E*, 84:036103, 2011.

[3] K. Barker and N. Cornacchia. Using noun phrase heads to extract document keyphrases. In *Proc. 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence*, 2000.

[4] D. M. Blei and J. D. Lafferty. Visualizing Topics with Multi-Word Expressions. *arXiv: 0907.1013*, July 2009.

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

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

[7] S.-L. Chuang and L.-F. Chien. A practical web-based approach to generating topic hierarchy for text segments. In *CIKM*, 2004.

[8] M. Danilevsky, C. Wang, N. Desai, J. Guo, and J. Han. Kert: Automatic extraction and ranking of topical keyphrases from content-representative document titles. arxiv: 1306.0271.

[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] J. Eisenstein, A. Ahmed, and E. P. Xing. Sparse additive generative models of text. In *ICML*, 2011.

[11] S. Gauch, J. Chaffee, and A. Pretschner. Ontology-based personalized search and browsing. *Web Intelligence and Agent Systems*, 1(3/4):219–234, 2003.

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

[13] M. Grineva, M. Grinev, and D. Lizorkin. Extracting key terms from noisy and multitheme documents. In *WWW*, 2009.

[14] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2011.

[15] J. Han, J. Pei, Y. Yin, and R. Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Min. Knowl. Discov.*, 8(1):53–87, jan 2004.

[16] 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, 2009.

[17] D. Lawrie and W. B. Croft. Discovering and comparing topic hierarchies. In *Proc. RIAO*, 2000.

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

[19] R. V. Lindsey, W. P. Headden, III, and M. J. Stipicevic. A phrase-discovering topic model using hierarchical pitman-yor processes. In *EMNLP-CoNLL*, 2012.

[20] X. Liu, Y. Song, S. Liu, and H. Wang. Automatic taxonomy construction from keywords. In *KDD*, 2012.

[21] Z. Liu, W. Huang, Y. Zheng, and M. Sun. Automatic keyphrase extraction via topic decomposition. In *EMNLP*, 2010.

[22] Q. Mei, X. Shen, and C. Zhai. Automatic labeling of multinomial topic models. In *KDD*, 2007.

[23] R. Mihalcea and D. Radev. *Graph-based natural language processing and information retrieval.* Cambridge University Press, 2011.

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

[25] R. Navigli, P. Velardi, and S. Faralli. A graph-based algorithm for inducing lexical taxonomies from scratch. In *IJCAI*, 2011.

[26] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2001.

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

[28] R. Snow, D. Jurafsky, and A. Y. Ng. Learning syntactic patterns for automatic hypernym discovery. *NIPS*, 2004.

[29] T. Tomokiyo and M. Hurst. A language model approach to keyphrase extraction. In *Proc. ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment - Volume 18*, 2003.

[30] H. M. Wallach. Topic modeling: beyond bag-of-words. In *ICML*, 2006.

[31] X. Wang, A. McCallum, and X. Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *ICDM*, 2007.

[32] W. Wong, W. Liu, and M. Bennamoun. Ontology learning from text: A look back and into the future. *ACM Computing Surveys (CSUR)*, 44(4):20, 2012.

[33] E. Zavitsanos, G. Paliouras, G. A. Vouros, and S. Petridis. Discovering subsumption hierarchies of ontology concepts from text corpora. In *Proc. IEEE/WIC/ACM International Conference on Web Intelligence*, WI, 2007.

[34] W. X. Zhao, J. Jiang, J. He, Y. Song, P. Achananuparp, E.-P. Lim, and X. Li. Topical keyphrase extraction from twitter. In *ACL-HLT*, 2011.