Document-Topic Hierarchies from Document Graphs

Tim Weninger Yonatan Bisk Jiawei Han
University of Illinois at Urbana-Champaign
{weninge1, bisk1, hanj}@illinois.edu

ABSTRACT

Topic taxonomies present a multi-level view of a document collection, where general topics live towards the top of the taxonomy and more specific topics live towards the bottom. Topic taxonomies allow users to quickly drill down into their topic of interest to find documents. We show that hierarchies of documents, where documents live at the inner nodes of the hierarchy-tree can also be inferred by combining document text with inter-document links. We present a Bayesian generative model by which an explicit hierarchy of documents is created. Experiments on three document-graph data sets shows that the generated document hierarchies are able to fit the observed data, and that the levels in the constructed document hierarchy represent practical groupings.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; G.3 [**Probability and Statistics**]: Probabilistic algorithms, Nonparametric statistics

General Terms

Algorithms, Experimentation

Keywords

hierarchical clustering, Bayesian generative models, topic models, model evaluation

1. INTRODUCTION

As the number of online resources and Web documents continues to increase, the need for better organizational structures that guide readers towards the information they seek increases. Hierarchies and taxonomies are invaluable tools for this purpose. Taxonomies are widely used in libraries via the Library of Congress System or the Dewey Decimal System, and hierarchies were a fixture of the early World Wide Web; perhaps the most famous example being the Yahoo search engine, which is actually an acronym for Yet Another Hierarchical Officious Oracle. These hierarchical systems were developed because their effectiveness at topical organization and their logarithmic depth allowed users to quickly find the relevant documents they were searching for.

Unfortunately, taxonomy curation of documents, articles, books, etc. is mostly a manual process, which is only possible when the number of curated documents is relatively small. This process becomes increasingly impractical when the number of documents grows to Web-scale. This has motivated research towards automatically inducing document taxonomies from the data [1, 4, 14, 5, 6, 19]. Most of the existing techniques rely on a single type of data – usually text. The problem with text-only hierarchy induction is that words often have multiple meanings. For example, the words "worm" and "bug" have very different meanings in the contexts of biology and computer science. Therefore without proper context proper taxonomy induction can be difficult.

Most document repositories contain linkages between the document creating a *document-graph*. These links provide proper context to the terms in each document. Document-graphs are especially common in nonfiction and scientific literature, where citations are viewed as inter-document links. The World Wide Web can be considered to be a single, very large document-graph, where Web pages represent documents and hyperlinks link documents.

Web sites, in particular, are a collection of documents with a very specific and purposeful organizational structure. Web sites are often specifically designed to guide the user from the entry page, *i.e.*, *homepage*, to progressively more specific Web pages. Similarly, scientific literature can be categorized into a hierarchy of increasingly specific scientific topics by their citation links, and encyclopedia articles can be categorized into a hierarchy of increasingly specific articles by their cross references. Thus, we assert that most document-graphs contain a hidden document hierarchy.

In this paper we draw a specific distinction between a *hierarchy* and a *taxonomy*. We define a taxonomy to be a classification of objects into increasingly finer granularities, where each non-leaf node is a conceptual combination its children. A biological taxonomy is a great example of this definition because a classified species, say homo sapiens (*i.e.*, humans), can only be placed at a leaf in the taxonomy; the inner nodes, *e.g.*, primate, mammal, animal, do not declare new species, rather they are conceptual agglomerations of species. Furthermore, each species is described by its path through the taxonomy. For example, homo sapiens, can be described as primates, as mammals and as animals (among others). A hierarchy, on the other hand, is an arrangement of objects where some objects are considered to be *above*, *below* or *at the same level as* others. This necessarily means that objects of a hierarchy live at the internal nodes.

Strictly speaking, most existing models infer taxonomies. The goal of this paper is to construct document hierarchies from a document-graph using document text and inter-document links. For

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.
Copyright 2012 ACM 978-1-4503-1156-4/12/10 ...$15.00.

these purposes *above*, *below* or *at the same level as* refers to the topical granularity of the documents. In other words, given a document graph with an explicitly identified root, such as a Web site homepage, we aim to learn a document-hierarchy which best captures the conceptual hierarchy of the document-graph. This problem poses three technical challenges:

1. **Inducing document topic mixtures**. We propose learning a document hierarchy where the internal (non-leaf) nodes of the hierarchy are documents. In such a hierarchy, parent documents consist of topics that are more general than their children. This requires that we view a parent document as a mixture of the topics contained within its children, and children documents should topically fit underneath their selected parent. We present the *Hierarchical Document-Topic Model* (HDTM) which generates a course-to-fine representation of the text information, wherein high-level documents live near the top of the hierarchy, and low-level, more specific documents live at the leaves.

2. **Selecting document placement**. Placement of a document within the hierarchy drives the topic mixing. Because links between edges hint at the context of and relationship between documents, we constrain the document placement in the induced hierarchy by their edges within the original document-graph. In other words, if an edge exists in the induced hierarchy, then it must also exist in original document-graph. Unlike existing models, such as hLDA [3], that select topic paths using the nested Chinese Restaurant Process (nCRP), we perform document placement based on a stochastic process resembling random walks with restart (RWR) over the original document-graph. The use of a stochastic process over the document-graph frees the algorithm from rigid parameters. Furthermore, the adoption of RWR stochastic process over nCRP allows documents to live at non-leaf nodes, and frees the algorithm from the depth parameter of hLDA.

3. **Analysis at Web site-scale**. In most document-graph collections, the number of edges grows quadratically with the number of nodes. This limits the scalability of many topic diffusion algorithms [20, 10]. Fortunately, document hierarchies are represented as trees, wherein the number of edges scales linearly with the number of documents.

The remainder of this paper is organized as follows. In Section 2 we review the related literature with specific attention paid to distinctions between similar generative models. In Section 3 we discuss the intuition behind hierarchy induction from document-graphs. Our proposed model is described in Section 4. In Section 5 we perform quantitative experiments on three data sets, as well as a large qualitative exploration based on thousands of human judgments. We find that our model constructs document hierarchies that are coherent with respect to their textual topics and conform to the graphical representation of the underlying document-graph.

2. RELATED WORK

There has been a substantial amount of previous work on hierarchical clustering of documents. The first approaches were called agglomerative clustering, which used greedy heuristics such as single-link or complete-link [29]. Dendrograms are often the output of such clustering techniques, in which the root node is split into a series of branches that terminate with a single document at each leaf. Ho, *et al.*, point out that manually-curated hierarchies like the Open Directory Project[1] are typically flatter and contain fewer inner nodes than agglomerative clustering techniques produce [14]. Other hierarchical clustering algorithms include top-down processes which iteratively partition the data [31], incremental methods like COBWEB [9], CLASSIT [11], and other algorithms optimized for hierarchical text clustering.

The processes that typically define most hierarchical clustering algorithms can be made to fit in a probabilistic setting that build bottom-up hierarchies based on Bayesian hypothesis testing [13]. On the other hand, most recent work uses Bayesian generative models to find the most likely explanation of observed text and links. The first of these hierarchical generative models was hierarchical latent Dirichlet allocation (hLDA). In hLDA each document sits at a leaf in a tree of fixed depth as illustrated in Figure 1(a). The document is represented by a mixture of multinomials along the path through the tree from the document to the root. Documents are placed at their respective leaf nodes by the nested Chinese restaurant process (nCRP).

NCRP is a recursive version of the standard Chinese Restaurant Process (CRP), which progresses according to the following analogy: An empty Chinese restaurant has an infinite number of tables, and each table has an infinite number of chairs. When the first customer arrives he sits in the first chair at the first table with probability of 1. The second customer can then chose to sit at an occupied table with probability of $\frac{n_i}{\gamma+n-1}$ or sit at a new, unoccupied table with probability of $\frac{\gamma}{\gamma+n-1}$, where n is the current customer, n_i is the number of customers currently sitting at table i, and γ is a parameter that defines the affinity to sit at a previously occupied table following a rich get richer scheme.

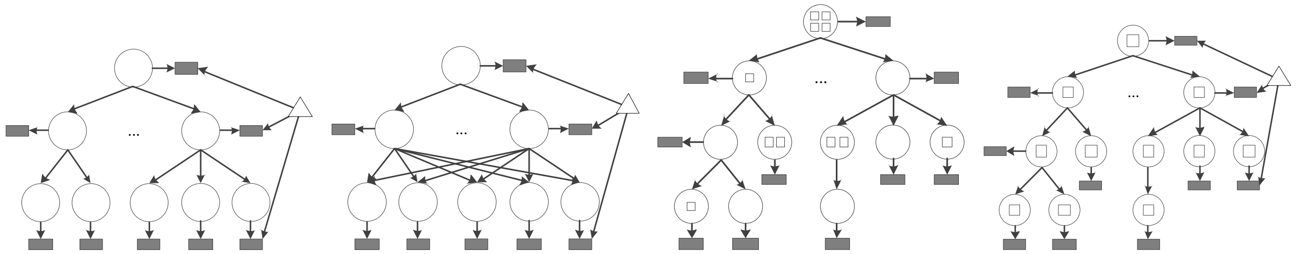
The nested version of the CRP extends the original analogy as follows: At each table in the Chinese restaurant are cards with the name of another Chinese restaurant. When a customer sits at a given table, he reads the card, gets up and goes to that restaurant, where he is reseated according to the CRP. Each customer visits L restaurants until he is finally seated and is able to each. This process creates a stochastic tree with a width determined by the γ parameter of a fixed depth L. This process has also been called the Chinese Restaurant Franchise because of this analogy [4].

Adams, *et al.* proposed a hierarchical topic model called tree structured stick breaking (TSSB), illustrated in Figure 1(c), wherein documents can live at internal nodes, rather than exclusively at leaf nodes. However, this process involves chaining together conjugate priors which makes inference more complicated, and it also does not make use of link data.

Other work along this line include hierarchical labeled LDA (hLLDA) by Petinot *et al.* [22] hLLDA, as well as fixed structure LDA (fsLDA) by Reisinger and Pasca [25] which modify hLDA by fixing the hierarchical structure and learning hierarchical topic distributions. The hierarchical pachinko allocation model(hPAM), shown in Figure 1(b), produces a directed acyclic graph (DAG) of a fixed depth allowing for each internal (non-document) node to be represented a mixture of more abstract, *i.e.*, higher level, topics [19].

In network-only data, community discovery is the process of finding self-similar group, or clusters. The SHRINK algorithm creates hierarchical clusters by identifying tightly-knit communities and by finding disparate clusters by looking for hubs and other heuristics [16]. The focus of this paper is more on probabilistic models to generate hierarchies, rather than heuristic approaches. Clauset, *et al*, discover dendrograms by Monte Carlo sampling;

[1] `http://www.dmoz.org`



(a) hLDA, TopicBlock (b) hPAM (c) TSSB (d) HDTM, fsLDA

Figure 1: Generative structures in related work. Circles and squares represent topics and documents respectively. Each topic has a multinomial over words (gray boxes), and a separate distribution over levels for each path (white triangles).

however, dendrograms poorly represent the manually curated hierarchies and taxonomies that we are pursuing [7].

In this paper we merge document text and inter-document links into a single model. This assumes that the words and their latent topics fit within the link structure of the graph, and that the graph structure explains topical relationships between interlinked documents. Topic Modeling with Network Structure (TMN), regularizes a statistical topic model with a harmonic regularizer based on the graph structure in the data; the result is that topic proportions of linked documents are similar to each other [18]. However, hierarchical information is not discovered nor can be easily inferred from this model.

Other work on generative models that combine text and links include: a probabilistic model for document connectivity [8], the Link-PLSA-LDA and Pairwise-Link-LDA methods [21], the Latent Topic Model for Hypertext (LTHM) method [12], role discovery in social networks [17], the author-topic-model [26], and others. The above models operate by encoding link probability as a discrete random variable or a Bernoulli trial that is parameterized by the topics of the documents. The relational topic model (RTM) builds links between topics, where observed links are given a very high likelihood [6]; although the paper is titled *hierarchical* relational models for topical networks, the RTM model does not build a topic or document hierarchy. The TopicBlock model combines the non-parametric hLDA and stochastic block models [15] to generate document taxonomies from text and links [14]; however, TopicBlock does not permit documents to reside at non-leaf nodes of the tree hierarchy.

In contrast to the previous work, our model builds a hierarchy of documents from text and inter-document links. In our model, each node in the hierarchy contains a single document, and the hierarchy's width and depth is not fixed.

3. HIERARCHIES OF DOCUMENTS

In the previous work, document hierarchies were not actually hierarchies of documents in the literal sense. Instead, leaf nodes of the hierarchy contains the actual, literal documents, and internal nodes contained increasingly more general topics about the ancestor documents. See Figure 1 for a brief comparison of model outputs. In this paper we require inner nodes, which in previous work are made of word multinomial distributions, to be literal documents. This requires an assertion that *some documents are more general than others*. This section explores this assertion through examples and a review of similar assertions made in previous research.

3.1 Web sites as document hierarchies

A Web site G can be viewed as a directed graph with Web pages as vertices V and hyperlinks as directed edges E between Web pages $v_x \rightarrow v_y$ – excluding inter-site hyperlinks. In most cases, designating Web site entry page as the root r allows for a Web site to be viewed as a rooted directed graph. Web site creators and curators purposefully organize the hyperlinks between documents in a topically meaningful manner. As a result, Web documents further away from the root document typically contain more specific topics than Web documents graphically close to the root document.

For example, the Web site at the University of Illinois in Urbana-Champaign, shown in Figure 2 contains a root Web document (the entry page), and dozens of children Web documents. Even with a very small subset of documents and edges, the corresponding Web graph is quite complicated and messy. A breadth first traversal of the Web graph starting with the root node is a simple way to distill a document hierarchy from the Web graph. Unfortunately, we shall see that a fixed breadth-first hierarchy cannot account for many of the intricacies of real world Web graphs.

For explanation purposes we define four types of hyperlink edges in a Web site: (1) parent-to-child links, (2) upward links, (3) shortcuts, and (4) cross-topic links. Parent-to-child links direct the user from one Web page to a more topically specific Web page; *e.g.*, a hyperlink from `../engineering` to `cs.illinois.edu` is a parent-to-child hyperlink because computer science is topically more specific than engineering. Upward links are hyperlinks that reference a more general document; *e.g.*, there may exist a hyperlink from `cs.illinois.edu` to `illinois.edu` because the computer science department would like to reference the fact that it belongs to the university. Shortcut links are hyperlinks that skip from very general Web documents to very specific Web documents as a way of featuring some specific topic; *e.g.*, if a computer science professor wins a prestigious award or grant, his Web page may be linked to from the news section of the root Web page. Cross topic links are hyperlinks that move across topical subtrees; *e.g.*, the college of media may reference some working relationship with the athletic department by creating a hyperlink between the two Web pages.

Because our goal is to infer the document hierarchy, we are, in a sense, trying to find parent-to-child links. In the event that there is more than one parent-to-child link to a particular Web page, our goal is to find the best topical fit for each Web document in the inferred hierarchy.

Web researchers and practitioners have used the hyperlink structure to organize Web documents for many years. The PageRank and HITS algorithms are two of the most famous examples of information propagation through links. Specifically, PageRank uses the model of a random Web surfer (*i.e.* random walker), who randomly follows hyperlinks over the Web. A current measure of a Web page's authority corresponds to the probability that a random surfer lands upon that Web page. In our model, we assert that

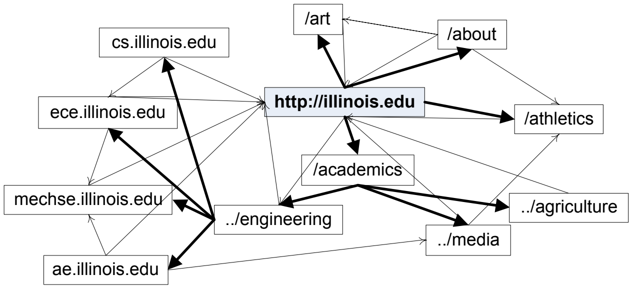


Figure 2: Truncated Web site graph of the University of Illinois Urbana-Champaign. Bold lines represent edges stochastically selected by the RWRH process

PageRank's notion of authority corresponds to topical generality. That is, Web pages with a high random surfer probability are likely to be topically more general than others.

3.1.1 *Term propagation in Web sites*

Plenty of previous works in the information retrieval domain use document-graph structure to enrich document features for improved retrieval performance. We find that some of the intuition behind these previous works are helpful in framing our generative model.

A limitation of the random walker model is that it only looks at the graphical structure of the Web. The word distributions found in each document are clearly an important factor to consider when generating Web hierarchies. Previous work by Song, *et al.* [27] and Qin, *et al.* [24] show that a given Web page can be enriched by propagating information from its children. Their relevance propagation model modifies the language distribution of a Web page to be a mixture of itself and its children according to the formula:

$$f'(w;d) = (1+\alpha)f(w;d) + \frac{(1-\alpha)}{|Child(d)|}\sum_{c\in Child(d)} f(w;c),$$

where $f(w;d)$ is the frequency of term w in Web page d before propagation, $f'(w;d)$ is frequency of term w in Web page d after propagation, c is a child page of d in the sitemap $\mathcal{T}$, and α is a parameter to control the mixing factor of the children. This propagation algorithm assumes that the sitemap, $\mathcal{T}$, is constructed ahead of time using URL features of the Web pages in a particular Web site.

Note that $f'(w;d)$ is a pseudo frequency count that is unsmoothed. The goal of previous works was to perform Web information retrieval, wherein they used BM25-type functions to normalize and smooth the language distribution. For illustration purposes, lets smooth the term distribution using Dirichlet prior smoothing [30]. The $f'(w;d)$ from above is used in place of the usual $c(w;d)$.

$$p_\mu(w;d) = \frac{f'(w;d) + \mu p(w|C)}{|d|' + \mu},$$

where C is the distribution over all terms in V, μ is the smoothing parameter, and the length is modified by the propagation algorithm to be $|d|' = (1+\alpha)|d|$.

As a result of the upward propagation p_μ of the root document (Web site entry page) contains all of the words from all of the Web pages in the Web site. The most probable words are those that occur most frequently and most generally across all documents, and are thus propagated the most.

In traditional topic hierarchy models (hLDA, TopicBlock, etc.), the root topic contains a distribution of all of the most general topic words in the document collection.

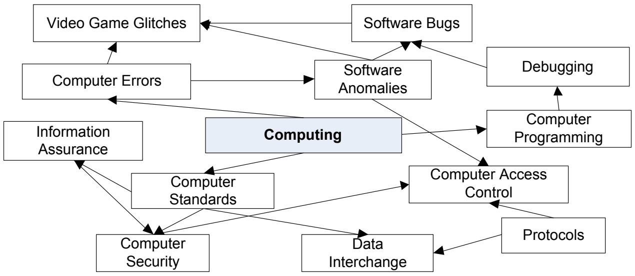


Figure 3: Truncated portion of the Wikipedia category subgraph rooted at the node COMPUTING.

p_μ $\alpha = .5$	hLDA $\gamma = 1$	HDTM $\gamma = 0.95$
illinois	illinois	illinois
computer	computer	computer
science	science	science
university	graduate	graduate
department	research	research
urbana	university	university

Table 1: Comparison of most probable words in top document (in p_μ), and in root topic of hLDA and HDTM

As a small, preliminary example, Table 1 shows the top six most probable words in the top document (via text propagation) and in root topics of hLDA and HDTM of the computer science department's Web site at the University of Illinois at Urbana-Champaign[2]. We see that the most probable words from the sitemap based Web document hierarchy is very similar to the most probable words in the most general topic of hLDA. This small example reinforces our intuition that certain Web sites have a hidden hierarchical topical structure.

In the previous term propagation work, the sitemaps were constructed ahead of time using URL heuristics. Our goal is to learn the document hierarchy automatically and in conjunction with the topical hierarchy.

3.2 Other document hierarchies

Documents from many different collections exist in hidden hierarchies. While technically a Web site, Wikipedia documents and categories form a unique document graph. Wikipedia categories are especially interesting because they provide a type of ontology wherein categories have more specific sub-categories and more general parent-categories. Most Wikipedia articles are are represented by at least one category description; this allows for users to drill down to relevant articles in a very few number of clicks by browsing the category graph. A partial example of the Wikipedia category graph is shown in Figure 3.

Bibliographical networks may also be hierarchically structured. In a bibliographic network, papers or authors (wherein each author could be a collection of documents) are represented by nodes and each citation is represented by an edge in the graph.

4. MODEL DESCRIPTION

We treat the problem of inferring the document hierarchy as a learning problem akin to finding the single, best parent for each

[2] http://cs.illinois.edu

document-node. Unlike previous algorithms, which discover latent topic taxonomies, the hierarchical document-topic model (HDTM) finds hidden hierarchies by selecting edges from a set of possible edges in the document graph. This section presents a detailed description of the model. A plate diagram of the generative process is shown in Figure 5.

We begin with a document graph $G = \{D, E\}$ of documents D and edges E. Each document is a collection of words, where a word is an item in a vocabulary. The basic assumption of HDTM and similar models is that each document can be generated by randomly mixing words from among topics. Distributions over topics are represented by z, which is a multinomial variable with an associated set of distributions over words $p(w|z, \beta)$, where β is a Dirichlet hyper-parameter. Document-specific mixing proportions are denoted by the vector θ. Parametric-Bayes topic models also include a K parameter that denote the number of topics, wherein z is one of K possible values and θ is a K-D vector. HDTM does not require a K parameter as input. Instead, in HDTM there exist $|G|$ topics, one for each graph node, and each document is a mixture of the topics on the path between itself and the root document.

In basic LDA, a single document mixture distribution is $p(w|\theta) = \sum_{i=1}^{K} \theta_i p(w|z = i, \beta_i)$. The process for generating a document is (1) choose a θ of topic proportions from a distribution $p(\theta|\alpha)$, where $p(\theta|\alpha)$ is a Dirichlet distribution; (2) sample words from the mixture distribution $p(w|\theta)$ for the θ chosen in step 1.

HLDA is an extension of LDA in which the topics are situated in a hierarchy T of fixed depth L. The hierarchy is generated by the nested Chinese restaurant process (nCRP) which essentially represents θ as an L-dimensional vector, defining an L-level path through T from root to document. Because of the nCRP process, every document lives at a leaf and the words in each document are a mixture of the topic-words on the path from it to the root.

4.1 Random Walks with Restart at Home

Because the nCRP process forces documents to the leaves in the hierarchy T, HDTM replaces nCRP with a slightly modified version of random walk with restart (RWR) called random walk with restart at home (RWRH). In traditional RWR, a walker begins by selecting a random starting point. With probability $(1 - \gamma)$ the walker randomly walks to a new, connected location or chooses to restart his walk at a random location with probability γ, where γ is called the restart probability[3].

In HDTM, the root node is fixed, either as the entry page of a Web site, or by some other heuristic. Therefore, for the purposes of hierarchy inference, we force the random walker to start and restart at the root node *i.e., at home*. Forcing the random walker to restart at the root is similar to, but not the same as, finding the personalized PageRank score [2] between the root node and every document-node in the hierarchy.

Let $deg(u)$ be the outdegree of document u in G. Consider a random walker visiting document d at time t. In the next time step, the walker chooses a document v_i from among u's outgoing neighbors $\{v|u \rightarrow_T v\}$ in the hierarchy T uniformly at random. In other words, at time $t + 1$, the surfer lands at node $v_i \in \{v|u \rightarrow_T v\}$ with probability $1/deg(u)$. If at any time, there exists an edge $k \in \{v|u \rightarrow_G v\}$, *i.e,* an edge between the current node u and the target node k in the original graph G, then we record the probability of that new path possibility for later sampling. Alg. 1 describes this process algorithmically. This procedure allows for new paths from the root $r \rightsquigarrow k$ to be probabilistically generated based on the current hierarchy effectively allowing for documents to migrate up, down and through the hierarchy during sampling. The bold edges in Figure 2 show an example of edges stochastically selected by the RWRH process.

Algorithm 1: Random Walk with Restart at Home

```
input   : Path Probs. $P$, Current Node $u$, Target $k$, Weight $w$
globals : Graph $\mathtt{G}$, Hierarchy $\mathtt{T}$, Restart Prob. $\gamma$
output  : $P$

foreach $v_i \in \mathtt{T.Ch}(u)$ do                /* child of $u$ in T */
    if $v_i \neq k$ then
        $w \leftarrow w + \log\left(\frac{1-\gamma}{\mathrm{len}(\mathtt{T.Ch}(u))}\right)$;
        RWRH ($P$, $v_i$, $k$, $w$);                   /* Recurse */
if $u \rightarrow_{\mathtt{G}} k$ then               /* Edge $u$ to $k$ exists in G */
    $P$.Put($u$, $w$);
```

4.2 Generating document paths

Because a document hierarchy is a tree, each document-node can only have a one parent. Selecting a path for a document d in the graph G is akin to selecting a parent $u = Pa(d)$ (and grandparents, etc.) from $\{d|u \rightarrow_G d\}$ in the document graph G. HDTM creates and samples from a probability distribution over each documents' parent, where the probability of document u being the parent of d is defined as:

$$\prod_{k=0}^{\mathrm{dep}_T(d)-1} \frac{1-\gamma}{\mathrm{deg}_T(d_k)},$$

where d_k is the walkers current position at time k, $\mathrm{dep}_T(d)$ is the depth of d in T, and $\mathrm{deg}_T(d_k)$ is the outdegree of d_k in T. In other words, the probability of landing at d is the product of the emission probabilities from each document in the path through T from r to d.

The modified random walker function assigns higher probabilities to parents that are at a shallower depth than those at deeper positions. This is in line with the intuition that flatter hierarchies are easier for human understanding than deep hierarchies [14]. Simply put, the restart probability γ controls how much resistance there is to placing a document at successive depths.

Algorithmically, we infer document hierarchies by drawing paths $\mathbf{c}_d$ from the r to the document d. Thus, the documents are drawn from the following generative process:

1. Each document $d \in G$ is assigned a topic $\beta_d \sim$ Dir(η):
2. For each document $d \in G$:
   (a) Draw a path $\mathbf{c}_d \sim$ RWRH(γ)
   (b) Draw an L-dimensional topic proportion vector θ from Dir(α), where $L =$len($\mathbf{c}_d$).
   (c) For each word $n \in \{1, \ldots, N\}$:
      i. Choose topic $z_{d,n}|\theta \sim$ Mult(θ_d).
      ii. Choose word $w_{d,n}|\{z_{d,n}, \mathbf{c}_d, \boldsymbol{\beta}\} \sim$ Mult($\beta_{\mathbf{c}_d, z_{d,n}}$), where $\beta_{\mathbf{c}_d, z_{d,n}}$ is the topic in the zth position in $\mathbf{c}_d$.

In this generative process hierarchical nodes represent documents *and* topics, where internal nodes contain the shared terminology of its descendants.

Like in earlier models, there is statistical pressure in the posterior to have more general terms in topics towards the root of the hierarchy. This is because every path in the hierarchy includes the

[3] Most related works denote the restart probability as α, however, this would be ambiguous with the Dirichlet hyper-parameter α.

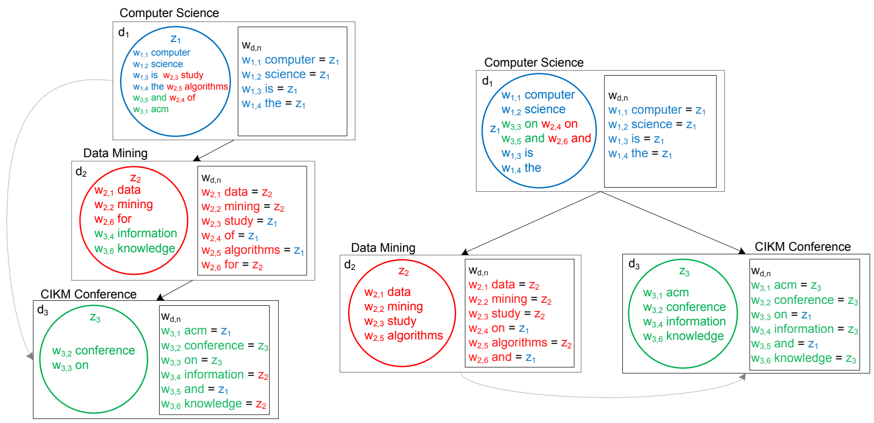


Figure 4: Illustration of two HDTM samples of the same data. Each node in the hierarchy contains a document and an associated topic. During the generative process, general terms are more likely to be found in topics near the root and vice versa.

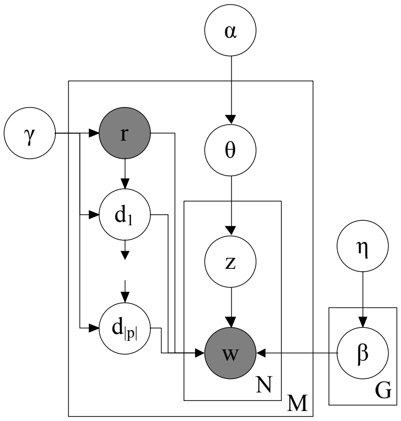


Figure 5: Graphical model illustration

root node and there are more paths through nodes at higher levels than through nodes at lower levels. As we move down the tree, the topics, and therefore the documents, become more specific.

Hyperparameters also play an important role in the shape and character of the hierarchy. The α parameter affects the smoothing on topic distributions, and the η parameter affects the smoothing on word distributions. The γ parameter is perhaps the most important parameter because it affects the depth of the hierarchy. Specifically, if γ is set to be large (*e.g.*, $\gamma = 0.95$) then resulting hierarchy shallow. Low values (*e.g.*, $\gamma = 0.05$) may result in deep hierarchies, because there is a smaller probabilistic penalty for each step that the random walker takes.

4.3 Inference

Exact inference on this model is intractable, so we must use a approximation technique for posterior inference. The Gibbs sampling algorithm is ideal in this situation because it simultaneously allows exploration of topic distributions and potential graphical hierarchies.

The variables needed by the Gibbs sampler are: $w_{d,n}$, the nth word in document d; $z_{d,n}$, the assignment of the nth word in document d; and $c_{d,z}$, the topic corresponding to document at the zth level. The θ and β variables are integrated out forming a collapsed Gibbs sampler.

The sampling is performed in two parts: (1) given the current level allocations of each word $z_{d,n}$ we sample the path $c_{d,z}$, (2) given the current state of the hierarchy, we sample $z_{d,n}$.

4.3.1 Sampling document paths

The first Gibbs sampling step is to draw a path from each document to the root through the graph. The sampling distribution for a path c_d is

$$\begin{aligned} &p(c_d|\mathbf{c}_{-d}, \mathbf{z}, \mathbf{w}, \eta, \gamma) \\ &\quad \propto p(c_d, \mathbf{w}_d|\mathbf{c}_{-d}, \mathbf{z}, \mathbf{w}_{-d}, \gamma, \eta) \\ &\quad = p(\mathbf{w}_d|\mathbf{c}, \mathbf{z}, \mathbf{w}_{-}d, \eta) p(c_d|\mathbf{c}_{-d}) \end{aligned} \tag{1}$$

where $\mathbf{w}$ is the count of terms in document d, and $\mathbf{w}_{-d}$ are the words without document d.

The second term represents the probability of drawing the path $c_{d,k}$ to document d at depth k from the RWRH process. Recall that each node has an emission probability of $1/\deg_T(d)$, and a restart probability of γ. We define the probability recursively:

$$\begin{aligned} &p(c_{d,k}|\mathbf{c}_{-d}, c_{d,1:(k-1)}) \\ &\quad = \prod_{k=0} \frac{1-\gamma}{\deg_T(d_k)} \end{aligned} \tag{2}$$

In other words, the probability of reaching d is equal to the probability of a random walker with restart probability γ being at document d at time k.

The first term represents the word distribution:

$$\begin{aligned} &p(\mathbf{w}_d|\mathbf{c}, \mathbf{w}_{-d}, \mathbf{z}, \eta) \\ &= \prod_{k=1}^{\max(\mathbf{z}_d)} \frac{\Gamma(\sum_w \#[\mathbf{c}_{-d,k} = c_{d,k}, \mathbf{w}_{-d} = w] + V\eta)}{\prod_w \Gamma(\#[\mathbf{c}_{-d,k} = c_{d,k}, \mathbf{w}_{-d} = w] + \eta)} \\ &\qquad \frac{\prod_w \Gamma(\#[\mathbf{z} = k, \mathbf{c}_k = c_{d,k}, \mathbf{w} = w] + \eta)}{\Gamma(\sum \#[\mathbf{z} = k, \mathbf{c}_k = c_{d,k}, \mathbf{w} = w] + V\eta)} \end{aligned} \tag{3}$$

where $\max(\mathbf{z}_d)$ is the maximum depth of the current hierarchy state.

4.3.2 Sampling word levels

Given the current state of all the variables, the sampler must first pick an assignment z for word n in document d. The sampling distribution of $z_{d,n}$ is

$$\begin{aligned} &p(z_{d,n}|\mathbf{c}, \mathbf{z}, \mathbf{w}, \eta, \gamma) \\ &\quad \propto p(w_{d,n}, z_{d,n}|\mathbf{c}, \mathbf{z}_{-(d,n)}, \mathbf{w}_{-(d,n)}, \eta, \gamma) \\ &\quad = p(w_{d,n}|\mathbf{c}, \mathbf{z}, \mathbf{w}_{-(d,n)}, \eta) p(z_{d,n}|\mathbf{z}_{d,-n}, \mathbf{c}, \gamma) \end{aligned} \tag{4}$$

where $\mathbf{z}_{d,-n} = \{z_{d,\cdot}\} \setminus z_{d,n}$ and $\mathbf{w}_{-(d,n)} = \{w\} \setminus w_{d,n}$. The first term is a distribution over word assignments:

$$\begin{aligned} &p(w_{d,n}|\mathbf{c}, \mathbf{z}, \mathbf{w}_{-(d,n)}, \eta) \\ &\quad \propto \#[\mathbf{z}_{-(d,n)} = z_{d,n}, \mathbf{c}_{z_{d,n}} = c_{d,z_{d,n}}, \mathbf{w}_{-(d,n)} = w_{d,n}] + \eta \end{aligned} \tag{5}$$

which is the η-smoothed frequency of seeing word $w_{d,n}$ in the topic at level $z_{d,n}$ in the path c_d.

The second term is the distribution over levels

$$p(z_{d,n} = k|\mathbf{z}_{d,-n}, \mathbf{c}, \gamma) = \left(\prod_{j=1}^{k-1} \frac{1-\gamma}{\deg_T(d_{j-1})} \frac{\#[\mathbf{z}_{d,-n} > j]}{\#[\mathbf{z}_{d,-n} \geq j]}\right) \times \frac{1-\gamma}{\deg_T(d_{k-1})} \frac{\#[\mathbf{z}_{d,-n} = k]}{\#[\mathbf{z}_{d,-n} \geq k]}, \qquad (6)$$

where we denote $\#[\cdot]$ as the number of elements in the vector which satisfy the given condition. We abuse notation in Eq. 6 so that the product from $j = 1$ to $k - 1$ combines terms representing nodes at the jth level in the path $\mathbf{c}$ down to the parent of d_k, and the second set of terms represents document d_k at level k. The $>$ symbol in Eq. 6 refers to terms representing all ancestors of a particular node, and $\geq$ refers to the ancestors of a node including itself.

5. EXPERIMENTS

This section describes the method and results for evaluating our model. We show quantitative and qualitative analysis of the hierarchical document-topic model's ability to learn accurate and interpretable hierarchies of document graphs. Our main evaluations explore the empirical likelihood of the data and a very large case study wherein human judges are asked to evaluate the constructed hierarchies.

5.1 Data

We evaluate HDTM on three corpora: the Wikipedia category graph, the Computer Science Department Web site at the University of Illinois, and a bibliographic network.

	Wikipedia	CompSci Web site	Bib. Network
root	Computing	`cs.illinois.edu`	Ponte, SIGIR [23]
documents	609	1,078	4,713
tokens	5,570,868	771,309	43,345
links	2,014	63,052	8,485
vocabulary	146,624	15,101	3,908

Table 2: Comparison of most probable words in top document (in p_μ) and in root topic (in hLDA)

The Wikipedia dataset has been used several times in the past for topic modeling purposes [12, 14]. Gruber *et al.,* crawled 105 pages starting with the article on the NIPS conference finding 799 links. Ho *et al.* performed a much larger evaluation of their TopicBlock model using 14,675 document with 152,674 links; however, they truncated each article to only the first 100 terms and limited the vocabulary to the 10,000 most popular words. Our Wikipedia dataset is a crawl of the *category* graph of Wikipedia, beginning at the category COMPUTING. In Wikipedia each category has a collection of articles and a set of links to other categories; however, categories don't typically have text associated with them, so we considered the text of each article associated with a particular category as the category's text. For example, the category INTERNET includes articles, INTERNET, HYPERLINK, WORLD WIDE WEB, ETC. In total we constructed a graph of 609 categories from 6,745 articles. The category graph is rather sparse with only 2,014 edges between categories, but has vocabulary size of 146,624 with 5,570,868 total tokens. We did not perform any stopword removal or stemming.

We chose a computer science department Web site as the second data set because it a rooted Web graph with familiar topics. By inferring the document hierarchy, we aim to find the organizational structure of the computer science department. Our intuition is that Web sites reflect the business organization of the underlying entity; thus we expect to find a subtrees consisting of courses, faculty, news, research areas, etc. at high levels, and specific Web pages at lower levels in the hierarchy. We crawled the Web site starting at the entry page and captured 1,078 Web pages and 63,052 hyperlinks. In total there were 15,101 unique terms from 771,309 tokens.

The bibliographic network consists of documents and titles from 4,713 articles from the SIGIR and CIKM conferences. There exist 3,908 terms across 43,345 tokens in the document collection. In this collection, links include citations between papers within the CIKM and SIGIR conferences. Citations between documents were provided by the authors of the ArnetMiner project [28], and is not complete. A SIGIR 1998 paper by Ponte and Croft [23] was chosen to be the root document because, in our records, it had the most in-collection citations.

5.2 Quantitative Analysis

HDTM has some distinct qualities that make apples to apples comparison difficult. Because HDTM is the first model to generate document hierarchies based on graphs, there is nothing to *directly* compare against. However, some of the models in the related work perform similar tasks, and so we perform comparisons when we are able.

The related works typically perform quantitative evaluation by measuring the log likelihood on held out data or by performing some other task like link prediction, etc. Log likelihood analysis looks at the goodness of fit on held out data. Unfortunately, we are not able to "hold out" any of our documents for testing, because each document, especially a first or second level document, is very important to the resulting hierarchy. Removing certain documents might even cause the graph to separate, which would make hierarchy inference impossible. For quantitative evaluation, we borrow the setup from [3] by comparing the states of each models' Gibbs sampler with the highest log complete likelihood.

We perform quantitative experiments on HLDA [3], TopicBlock [14], TSSB [1], and fsLDA [25]. The fixed structure in fsLDA is determined by a breadth first iteration over the document graph because URL heuristics were found to be unreliable. Hyper-parameters are the default unless otherwise specified. The depth of HLDA and TopicBlock is 4.

In all cases, a Gibbs sampler was run for 5,000 iterations; 2000 iterations were discarded as burn-in. Figure 5(a) shows the log complete likelihood for each sample. We ran the the Gibbs sampling algorithm on HDTM for various values of γ, and Figure 6 shows the best cumulative log complete likelihood for each of the tested values of γ.

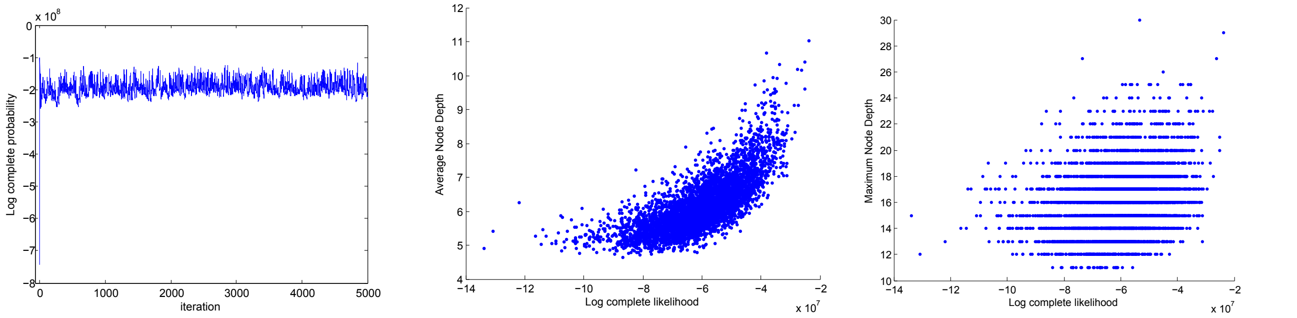
Interestingly, Figure 5(b) shows that higher likelihood values are strongly correlated with hierarchies of deeper average depth; Figure 5(c) finds that the same is true for hierarchies of deeper maximum depth.

Figure 6 shows that HDTM with $\gamma = 0.05$ achieved the best likelihood score, and HDTM with $\gamma = 0.95$ achieved the worst likelihood score.

Table 3 shows the results of the different algorithms on the three data sets. The TopicBlock and TSSB clearly infer models with the best likelihood. The remaining algorithms, including HDTM, have mixed results.

5.2.1 Discussion

In order to properly understand the results captured in Table 3, recall that log probability is a metric on the *fit* of the observations on the configuration of the model. The original work on LDA [4] found that likelihood increases as the number of topics



(a) The complete log likelihood

(b) Log complete likelihood and corresponding average node depth. High likelihood corresponds with hierarchies of high average depth.

(c) Log complete likelihood and corresponding maximum node depth. High likelihood corresponds with hierarchies of high maximum depth.

Figure 5: Analysis of Likelihood Scores for 5,000 iterations of the Gibbs sampler run on the CompSci collection.

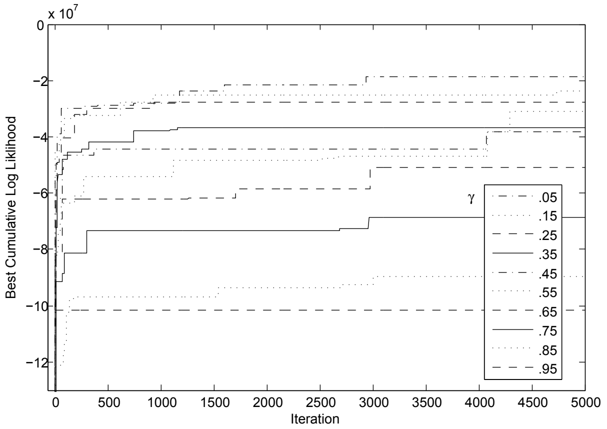


Figure 6: Best cumulative log complete likelihood for each tested γ value. Lower γ values result in deeper hierarchies.

	CompSci Web site	Wikipedia	Bib. Network
HDTM $\gamma = 0.05$	-1.8570	-148.071	-0.4758
HDTM $\gamma = 0.95$	-9.2412	-148.166	-0.5183
HLDA $\gamma = 1.0$	-8.5306	-50.6732	-8.5448
TopicBlock $\gamma = 1.0$	**-0.2404**	-2.9827	**-0.4192**
TSSB $k = 10$	-0.5689	**-0.0336**	-0.4655
fsLDA	-48.9149	-149.622	-0.6602

Table 3: Log complete likelihood results of the best sample from among 5,000 Gibbs iterations. Values are $\times 10^6$. Higher values are better. Best results are in bold.

increases. Along those lines, Chang, *et al.* demonstrated that more fine grained topics, which appear in models with a larger number of topics have a lower interpretability, despite having higher likelihood scores. Simply put, there exists a negative correlation between likelihood scores and human interpretability.

Applying these lessons to our experiments recall that HDTM has as many topics as there are documents, and non-root document topics are mixtures of the topics on the path to the root. Also recall that HLDA, TopicBlock and TSSB all generate a large number of latent topics. In HLDA and TopicBlock, there are infinitely many topics/tables in the nCRP. Practically speaking, the number of topics in the final model is much larger than the number of documents (conditioned on the γ parameter). In TSSB, the topic generation is said to be an interleaving of two stick breaking processes; practically, this generates even larger topic hierarchies. The fsLDA algorithm has as many topics as there are in hLDA, however, the fsLDA hierarchy is not redrawn during Gibbs iterations to fit the word distributions resulting in a lower likelihood.

Similarly, Figures 5(b) and 5(c) show that deeper hierarchies have higher likelihood scores. This is because long document-to-root paths, found in deep hierarchies, are able to provide a more fine grained fit for the words in the document resulting in a higher likelihood.

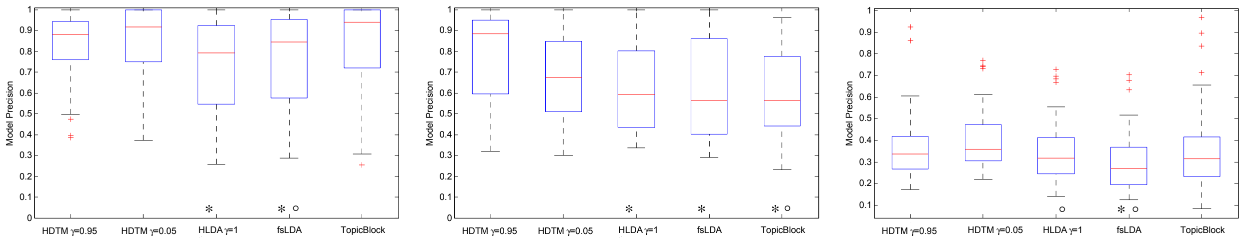
Therefore, we contend that the better likelihood values of HLDA, TopicBlock and TSSB are due to the larger number of topics that these models infer. A better way to evaluate model accuracy is by some external task or by manually judging the coherence of the topics.

5.3 Qualitative Analysis

To measure the coherence of the groupings, we modify the *word intrusion* task developed by Chang *et al* [6] to create the *document intrusion* task. In this task, a human subject is presented with a randomly ordered set of eight document titles. The task for the human judge is to find the intruder, that is, which document is out of place or does not belong. If the set of documents without the intruder document all make sense together, then the human judge should easily be able to find the intruder. For example, given a set of computer science documents with titles {`systems`, `networking`, `databases`, `graphics`, `Alan Turing`}, most people, even non-computer scientists, would pick `Alan Turing` as the intruder because the remaining words make sense together – they are all computer science disciplines.

For the set {`systems`, `networking`, `RAM`, `Minesweeper`, `Alan Turing`}, identifying a single intruder is more difficult. Human judges, when forced to make a choice, will choose an intruder at random, indicating that the grouping has poor coherence.

To construct a set of document titles to present to the human judge, we first select a grouping from the hierarchy at random (discussed in Sec. 5.3.1), and select 7 documents at random from the grouping. If the there are fewer than 7 documents available in the selected grouping, then we select all of the documents available; groupings of size less than 4 are thrown out. In addition to these documents, an intruder document is selected at random from among the entire collection of documents minus the documents in the test group. Titles are then shuffled and presented to the human judges.



(a) CompSci Web site (b) Wikipedia (c) Bib. Network

Figure 7: The model precision for five models on three document-graph collections. Higher is better. $*$ and $\circ$ represents statistical significance from HDTM $\gamma = 0.95$ and $\gamma = 0.05$ respectively.

1 / 5	LANGUAGE REVIVAL	BILINGUAL EDUCATION	LINGUISTIC PURISM	CLASSICAL PHILOLOGISTS
2 / 5	ANDROID SOFTWARE	MOBILE BUSINESS SOFTWARE	VIDEO GAME JOURNALISM	FREE MOBILE SOFTWARE
3 / 5	LIBRARY AUTOMATION	POINTING-DEVICE TEXT INPUT	MULTI-TOUCH	AUDITORY DISPLAYS
4 / 5	SPREADSHEET SOFTWARE	INDIC COMPUTING	ONLINE SPREADSHEETS	SPREADSHEET FORMATS
5 / 5	NETWORK ACCELERATION	COMPUTER HARDWARE TUNING	COPY PROTECTION	COMPUTER OPTIMIZATION

Figure 8: Illustration of the intruder detection task from the Wikipedia collection, wherein human judges are presented with a set of document titles and asked to select the document that does not belong.

5.3.1 *Comparison Models*

In preparation for human judgments, we construct a hierarchy from the mode of sampled hierarchies. Specifically, at every 20th sample (*i.e.* Gibbs lag = 20), the parent of each document is recorded. After the Gibbs iterations are complete, each document is endowed by the parent that it saw in the most samples.

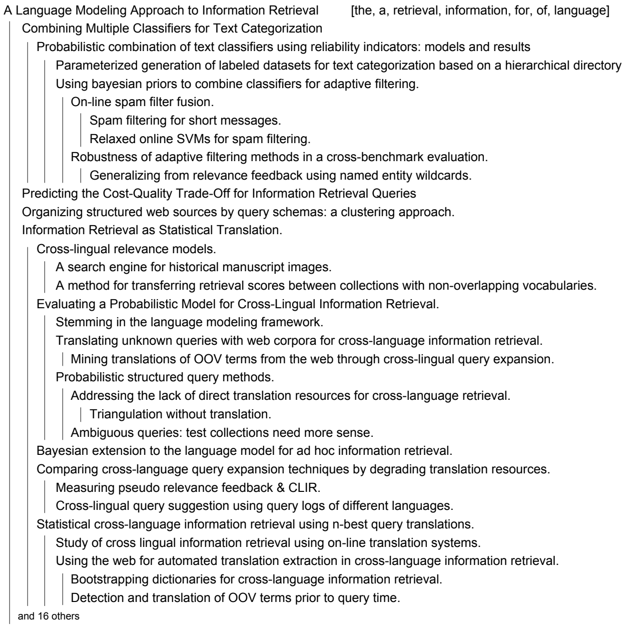
Extracting document groupings for evaluation is slightly different for each model. HDTM and fsLDA store a document at each node in the hierarchy. We select a grouping by first picking a document at random, and then choosing its siblings. TopicBlock and HLDA store documents at the leaves of the taxonomy, which often include several documents. We select a grouping from these models by first picking a document at random, and then choosing the other documents in the leaf-topic.

The hierarchies that the TSSB model constructed allowed multiple documents to live at inner nodes, We were unsuccessful in our attempts to evaluate groupings on inner nodes with more than 4 documents. We also tried to find nodes with 4 or more siblings, however, the hierarchies that were generated were too sparse to find practical groupings. Thus we were unable to provide human judges with TSSB groupings.

Each document-graph collection had different types of labels presented to the judges. The CompSci web site collection was labeled by the Web Page title and URL; the Wikipedia collection was labeled by the category title as shown in Figure 8; the bibliography network was labeled by the title of the paper.

5.3.2 *Analyzing human judgments*

The intruder detection tasks described above were offered on Amazon Mechanical Turk. No specialized training is expected of the judges. 50 tasks were created for each dataset and model combination; each user was presented with 5 tasks at a time at a cost of $0.07 per task. Each task was evaluated by 15 separate judges. In order to measure the trustworthiness of a judge, we selected 5 easy tasks, *i.e.*, groupings with clear intruders, and created gold answers.



- A Language Modeling Approach to Information Retrieval [the, a, retrieval, information, for, of, language]
  - Combining Multiple Classifiers for Text Categorization
    - Probabilistic combination of text classifiers using reliability indicators: models and results
      - Parameterized generation of labeled datasets for text categorization based on a hierarchical directory
      - Using bayesian priors to combine classifiers for adaptive filtering.
        - On-line spam filter fusion.
          - Spam filtering for short messages.
          - Relaxed online SVMs for spam filtering.
        - Robustness of adaptive filtering methods in a cross-benchmark evaluation.
          - Generalizing from relevance feedback using named entity wildcards.
  - Predicting the Cost-Quality Trade-Off for Information Retrieval Queries
  - Organizing structured web sources by query schemas: a clustering approach.
  - Information Retrieval as Statistical Translation.
    - Cross-lingual relevance models.
      - A search engine for historical manuscript images.
      - A method for transferring retrieval scores between collections with non-overlapping vocabularies.
    - Evaluating a Probabilistic Model for Cross-Lingual Information Retrieval.
      - Stemming in the language modeling framework.
      - Translating unknown queries with web corpora for cross-language information retrieval.
        - Mining translations of OOV terms from the web through cross-lingual query expansion.
      - Probabilistic structured query methods.
        - Addressing the lack of direct translation resources for cross-language retrieval.
          - Triangulation without translation.
        - Ambiguous queries: test collections need more sense.
    - Bayesian extension to the language model for ad hoc information retrieval.
    - Comparing cross-language query expansion techniques by degrading translation resources.
      - Measuring pseudo relevance feedback & CLIR.
      - Cross-lingual query suggestion using query logs of different languages.
    - Statistical cross-language information retrieval using n-best query translations.
      - Study of cross lingual information retrieval using on-line translation systems.
      - Using the web for automated translation extraction in cross-language information retrieval.
        - Bootstrapping dictionaries for cross-language information retrieval.
        - Detection and translation of OOV terms prior to query time.
  - and 16 others

Figure 9: Constructed hierarchy of bibliographic network with HDTM $\gamma = .95$. Words at the root document represent the most probable words in the root topic. Most probable words for other documents are not shown due to space constraints.

Judges who did not answer 80% of the gold answers correctly are thrown out and not paid. In total our solicitation attracted 31,494 judgments, across 14 models of 50 tasks each. Of these, 13,165 judgments were found to be from trustworthy judges.

We measure the *model precision* based on how well the intruders were detected by the judges. Specifically, if the intruder word w_k^m is from model m and task k, and $i_{k,j}^m$ is the intruder selected by the human judge j on task k in model m then

$$\text{MP}_k^m = \sum_J \mathbb{1}(i_{k,j}^m = w_k^m)/J.$$

where $\mathbb{1}(\cdot)$ is the indicator function and J is the number of judges. The model precision is basically the fraction of judges agreeing with the model.

Figure 7 shows boxplots of the precision for the four models on three corpora. In most cases, HDTM performs the best. As in [6], the likelihood scores do not necessarily correspond to human judgments. This is probably because the `RWRH` function essentially constrains the flexibility of the word sampler to operate only over explicit paths in the rooted graph. Paired, two-tailed t-tests of statistical significants ($p < 0.05$) performed between HDTM $\gamma = 0.95$ and $\gamma = 0.05$ and the other models are represented by $*$ and $\circ$ in Figure 7 respectively.

The bibliography network data had relatively low precision scores. This is probably because it was more difficult for the judges, who were probably not computer scientists, to differentiate between the topics in research paper titles. Figure 9 shows a small portion of the document hierarchy for the bibliographic network dataset constructed with HDTM $\gamma = .95$. The root document has 20 children in the hierarchy despite having 145 in-collection links. The remaining 120 documents live deeper in the hierarchy because HDTM has determined that they are too specific to warrant a first level position, and have a better fit in one of the subtrees.

Recall that each document is associated with the topics from itself to the root, where the root is a single, general topic. The seven most probable terms at the root level are also shown adjacent to the root's title in Figure 9. We see that these terms, like in HLDA and TopicBlock, are terms that are general to the entire collection.

6. CONCLUSIONS

We have presented hierarchical document-topic model (HDTM), a Bayesian generative model that creates document and topic hierarchies from rooted document graphs. We hypothesized that document graphs, such as Web sites, Wikipedia and bibliographic networks contain a hidden hierarchy, and we show corollaries to this intuition in language model propagation literature. Unlike most previous work, HDTM allows documents to live at non-leaf nodes in the hierarchy, which requires a new path sampling technique we call Random Walk with Restart at Home. An interesting side-effect of the random walker adaptation is that the path sampling step, Eq. 1, is much faster than the nCRP because RWRH only creates a sampling distribution for the parents of a document, whereas the nCRP process creates a sampling distribution over all possible paths in the taxonomy.

We performed several quantitative experiments comparing HDTM with related models. We conclude, as others have before us, that likelihood scores are a poor indicator of hierarchy interpretability, especially when the number of topics are different between comparison models. We performed a large qualitative case study which showed that the cohesiveness of the document groupings generated by HDTM were statistically better than many of the comparison models despite the poor likelihood scores.

7. ACKNOWLEDGMENTS

We thank Qirong Ho for his helpful discussion, the developers of the Mallet toolkit, as well as Rob Rutenbar and Abner Guzmán Rivera for their initial guidance. This work is funded by an NDSEG Fellowship to the first author. The third author is supported by NSF IIS-09-05215, U.S. Air Force Office of Scientific Research MURI award FA9550-08-1-0265, and by the U.S. Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 (NS-CTA).

8. REFERENCES

[1] R. P. Adams, Z. Ghahramani, and M. I. Jordan. Tree-Structured Stick Breaking for Hierarchical Data. In *NIPS*, pages 19–27, 2010.

[2] B. Bahmani, A. Chowdhury, and A. Goel. Fast Incremental and Personalized PageRank. *PVLDB*, 4(3):173–184, 2010.

[3] D. M. Blei, T. L. Griffiths, and M. I. Jordan. Hierarchical Topic Models and the Nested Chinese Restaurant Process. *Journal of the ACM*, 57(2):1–30, 2010.

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

[5] A. Chambers, P. Smyth, and M. Steyvers. Learning concept graphs from text with stick-breaking priors. In *NIPS*, pages 334–342, 2010.

[6] J. Chang and D. M. Blei. Relational Topic Models for Document Networks. *Annals of Applied Statistics*, 4(1):121–150, 2010.

[7] A. Clauset, C. Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101, May 2008.

[8] D. A. Cohn and T. Hofmann. The Missing Link - A Probabilistic Model of Document Content and Hypertext Connectivity. In *NIPS*, pages 430–436, 2000.

[9] D. H. Fisher. Knowledge Acquisition Via Incremental Conceptual Clustering. *Machine Learning*, 2(2):139–172, Sept. 1987.

[10] T. Furukawa, Y. Matsuo, I. Ohmukai, K. Uchiyama, and M. Ishizuka. Extracting Topics and Innovators Using Topic Diffusion Process in Weblogs. In *ICWSM*, 2008.

[11] J. H. Gennari, P. Langley, and D. Fisher. Models of incremental concept formation. *Artificial Intelligence*, 40(1-3):11–61, Sept. 1989.

[12] A. Gruber, M. Rosen-Zvi, and Y. Weiss. Latent Topic Models for Hypertext. In *UAI*, pages 230–239, 2008.

[13] K. A. Heller and Z. Ghahramani. Bayesian hierarchical clustering. In *ICML*, pages 297–304, New York, New York, USA, Aug. 2005.

[14] Q. Ho, J. Eisenstein, and E. P. Xing. Document hierarchies from text and links. In *WWW*, page 739, New York, New York, USA, Apr. 2012.

[15] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

[16] J. Huang, H. Sun, J. Han, H. Deng, Y. Sun, and Y. Liu. SHRINK. In *CIKM*, page 219, New York, New York, USA, Oct. 2010.

[17] A. McCallum, A. Corrada-Emmanuel, and X. Wang. Topic and Role Discovery in Social Networks. In *IJCAI*, pages 786–791, 2005.

[18] Q. Mei, D. Cai, D. Zhang, and C. Zhai. Topic modeling with network regularization. In *WWW*, pages 101–110, New York, New York, USA, Apr. 2008.

[19] D. Mimno, W. Li, and A. McCallum. Mixtures of hierarchical topics with Pachinko allocation. In *ICML*, pages 633–640, New York, New York, USA, June 2007.

[20] R. Nallapati, D. A. McFarland, and C. D. Manning. TopicFlow Model: Unsupervised Learning of Topic-specific Influences of Hyperlinked Documents. In *AISTATS*, volume 15, pages 543–551, 2011.

[21] R. M. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen. Joint latent topic models for text and citations. In *SIGKDD*, pages 542–550, New York, New York, USA, Aug. 2008.

[22] Y. Petinot, K. McKeown, and K. Thadani. A hierarchical model of web summaries. In *ACL*, pages 670–675, June 2011.

[23] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR*, pages 275–281, New York, New York, USA, Aug. 1998.

[24] T. Qin, T.-Y. Liu, X.-D. Zhang, Z. Chen, and W.-Y. Ma. A study of relevance propagation for web search. In *SIGIR*, page 408, New York, New York, USA, Aug. 2005.

[25] J. Reisinger and M. PaÅ§ca. Latent variable models of concept-attribute attachment. *ACL*, pages 620–628, Aug. 2009.

[26] M. Rosen-Zvi, T. L. Griffiths, M. Steyvers, and P. Smyth. The Author-Topic Model for Authors and Documents. In *UAI*, pages 487–494, 2004.

[27] R. Song, J.-R. Wen, S. Shi, G. Xin, T.-Y. Liu, T. Qin, X. Zheng, J. Zhang, G.-R. Xue, and W.-Y. Ma. Microsoft Research Asia at Web Track and Terabyte Track of TREC 2004. In *TREC*, 2004.

[28] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. ArnetMiner. In *SIGKDD*, page 990, New York, New York, USA, Aug. 2008.

[29] P. Willett. Recent trends in hierarchic document clustering: A critical review. *Information Processing & Management*, 24(5):577–597, Jan. 1988.

[30] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM TOIS*, 22(2):179–214, Apr. 2004.

[31] Y. Zhao, G. Karypis, and U. Fayyad. Hierarchical Clustering Algorithms for Document Datasets. *Data Mining and Knowledge Discovery*, 10(2):141–168, Mar. 2005.