Document Hierarchies from Text and Links

Qirong Ho
Machine Learning Department
School of Computer Science
Carnegie Mellon University
qho@cs.cmu.edu

Jacob Eisenstein[*]
School of Interactive Computing
Georgia Institute of Technology
jacobe@gatech.edu

Eric P. Xing
Machine Learning Department
School of Computer Science
Carnegie Mellon University
epxing@cs.cmu.edu

ABSTRACT

Hierarchical taxonomies provide a multi-level view of large document collections, allowing users to rapidly drill down to fine-grained distinctions in topics of interest. We show that automatically induced taxonomies can be made more robust by combining text with relational links. The underlying mechanism is a Bayesian generative model in which a latent hierarchical structure explains the observed data — thus, finding hierarchical groups of documents with similar word distributions and dense network connections. As a nonparametric Bayesian model, our approach does not require pre-specification of the branching factor at each non-terminal, but finds the appropriate level of detail directly from the data. Unlike many prior latent space models of network structure, the complexity of our approach does not grow quadratically in the number of documents, enabling application to networks with more than ten thousand nodes. Experimental results on hypertext and citation network corpora demonstrate the advantages of our hierarchical, multimodal approach.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; G.3 [**Probability and Statistics**]

General Terms

Algorithms, Experimentation

Keywords

hierarchical clustering, Bayesian generative models, topic models, stochastic block models

1. INTRODUCTION

As the quantity of online documents continues to increase, there is a need for organizational structures to help readers find the content that they need. Libraries have long employed *hierarchical taxonomies* such as the Library of Congress System[1] for this purpose; a similar approach was taken in the early days of the Web, with portal sites that present the user with a hierarchical organization of web pages. The key advantage of such taxonomies is that the logarithmic depth of tree structures permits fine-grained distinctions between thousands of "leaf" subtopics, while presenting the user with at most a few dozen choices at a time. The user can recursively drill down to a very fine-grained categorization in an area of interest, while quickly disregarding irrelevant topics at the coarse-grained level. A partial example of a hierarchical taxonomy for Wikipedia is shown in Figure 1.

Manual curation of taxonomies was possible when membership was restricted to books or a relatively small number of web publishers, but becomes increasingly impractical as the volume of documents grows. This has motivated research towards inducing hierarchical taxonomies automatically from data [39, 7]. However, these existing solutions rely exclusively on a single modality, usually text. This can be problematic, as content is often ambiguous — for example, the words "scale" and "chord" have very different meanings in the contexts of computer networks and music theory.

As a solution, we propose to build taxonomies that incorporate the widely available metadata of links between documents. Such links appear in many settings: hyperlinks between web pages, citations between academic articles, and social network connections between the authors of social media. Network metadata can disambiguate content by incorporating an additional view which is often orthogonal to text. For example, we can avoid conflating two documents that mention "scales" and "chords" if they exist in completely different network communities; analagously, we can group documents which share network properties, even if the text is superficially different.

We have incorporated these ideas into a system called TopicBlock, which uses both text and network data to induce a hierarchical taxonomy for a document collection. This requires meeting three technical challenges:

- **Challenge 1: Combining the disparate representations of text and network data**. Network and text content have very different underlying representations. We propose a model in which both the text and network are stochastic emissions from a latent hierarchical structure. The inference task is to find the latent structure which is likely to have emitted the observed data. On the text side we use the machinery of hierarchical latent topic models [7], a coarse-to-fine representation in which high-level content is generated from shared nodes near the root of the hierarchy, while more technical

[*] Jacob Eisenstein's contribution to this work was performed at Carnegie Mellon University.

[1] http://www.loc.gov/catdir/cpso/lcco

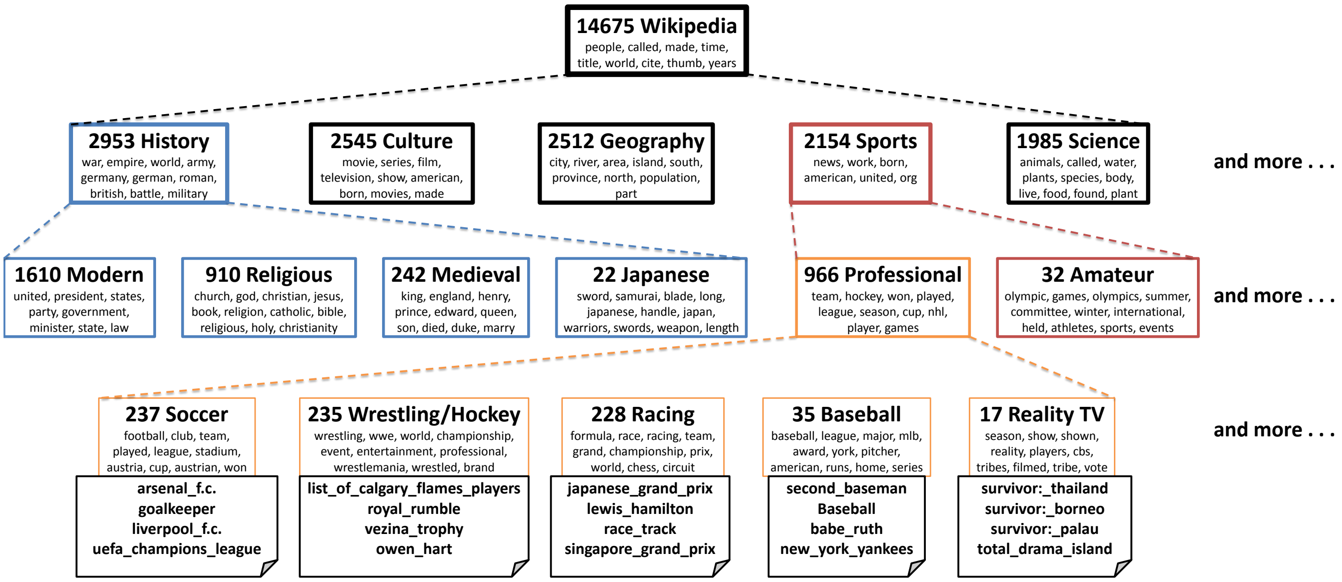
14675 Wikipedia
people, called, made, time, title, world, cite, thumb, years
2953 History
war, empire, world, army, germany, german, roman, british, battle, military
2545 Culture
movie, series, film, television, show, american, born, movies, made
2512 Geography
city, river, area, island, south, province, north, population, part
2154 Sports
news, work, born, american, united, org
1985 Science
animals, called, water, plants, species, body, live, food, found, plant
and more . . .
1610 Modern
united, president, states, party, government, minister, state, law
910 Religious
church, god, christian, jesus, book, religion, catholic, bible, religious, holy, christianity
242 Medieval
king, england, henry, prince, edward, queen, son, died, duke, marry
22 Japanese
sword, samurai, blade, long, japanese, handle, japan, warriors, swords, weapon, length
966 Professional
team, hockey, won, played, league, season, cup, nhl, player, games
32 Amateur
olympic, games, olympics, summer, committee, winter, international, held, athletes, sports, events
and more . . .
237 Soccer
football, club, team, played, league, stadium, austria, cup, austrian, won
235 Wrestling/Hockey
wrestling, wwe, world, championship, event, entertainment, professional, wrestlemania, wrestled, brand
228 Racing
formula, race, racing, team, grand, championship, prix, world, chess, circuit
35 Baseball
baseball, league, major, mlb, award, york, pitcher, american, runs, home, series
17 Reality TV
season, show, shown, reality, players, cbs, tribes, filmed, tribe, vote
and more . . .
arsenal_f.c.
goalkeeper
liverpool_f.c.
uefa_champions_league
list_of_calgary_flames_players
royal_rumble
vezina_trophy
owen_hart
japanese_grand_prix
lewis_hamilton
race_track
singapore_grand_prix
second_baseman
Baseball
babe_ruth
new_york_yankees
survivor:_thailand
survivor:_borneo
survivor:_palau
total_drama_island

Figure 1: An example 4-level topic hierarchy built from Wikipedia Simple English. We annotate each topic with its number of documents, a manually-chosen label describing the topic, and a list of highly ranked-words according to TF-IDF. The dotted lines in the hierarchy show parent and child topics (only the children of some parents are shown). For the bottom level topics, we also provide the names of some Wikipedia documents associated with them. The associated network data is shown in Figure 4.

information is generated from the detailed subtopics at the leaves. On the network side, we employ a hierarchical version of the stochastic block model [21], in which links are emissions from Bernoulli distributions associated with nodes in the hierarchy.

- **Challenge 2: Selecting the appropriate granularity**. The problem of identifying model granularity is endemic for latent structure models [38], but it is particulary vexing in the hierarchical setting. A flat mixture model or topic model requires only a single granularity parameter (the number of clusters or topics), but a hierarchy requires a granularity parameter at each non-terminal. Furthermore, the ideal granularity is not likely to be identical across the hierarchy: for example, the nuclear physics topic may demand fewer subtopics than the cephalopods topic. TopicBlock incorporates a Bayesian nonparametric prior which lets the data speak for itself, thus automatically determining the appropriate granularity at each node in the hierarchy.

- **Challenge 3: Scaling the network analysis**. In network data, the number of possible links grows quadratically with the number of nodes. This limits the scalability of many previous techniques [2, 29]. In contrast, TopicBlock's complexity scales linearly with the number of nodes and the depth of the hierarchy. This is possible due to the hierarchically-structured latent representation, which has the flexibility to model link probabilities finely where necessary (at the leaf level), while backing off to a coarse representation where possible (between nodes in disparate parts of the hierarchy).

We apply TopicBlock to two datasets. The first is Simple English Wikipedia, in which documents on a very broad array of subjects are connected by hyperlinks. The second is the ACL Anthology [5], a collection of scientific research articles, in which documents are connected by citations. TopicBlock yields hierarchies which are coherent with respect to both text and relational structure, grouping documents which share terms and also contain dense relational patterns. In the ACL Anthology data, we evaluate the capability of TopicBlock to recommend citation links from text alone. In the Wikipedia data, we evaluate TopicBlock's ability to identify the correct target of a hyperlink that is lexically ambiguous.

2. RELATED WORK

There is substantial prior work on hierarchical document clustering. Early approaches were greedy, using single-link or complete-link heuristics [39]. This yields a *dendrogram* of documents, in which a root node is decomposed in a series of binary branching decisions until every leaf contains a single document. We prefer flatter trees with fewer non-terminals, which are more similar to manually-curated hierarchies.[2] Other work on hierarchical clustering includes top-down techniques for iteratively partitioning the data [41], search-based incremental methods [36], probabilistic modeling of manually-created taxonomies [31], and interactive exploration [11].

The splitting and merging decisions that characterize most hierarchical clustering algorithms can be made on the basis of Bayesian hypothesis tests [19]. However, our work more closely relates to Bayesian *generative* models over the document content, as we focus on inducing a latent structure that provides a likely explanation for the observed text and links. Hierarchical latent Dirichlet allocation (hLDA) is a prototypical example of such an approach: each document sits on a path through a hierarchy with unbounded tree-width, and the text is generated from a mixture of multinomials along the path. We extend hLDA by incorporating network data, enabling a better understanding of the relationship between these two modalities. Adams et al. [1] present a hierarchical topic model which differs from hLDA in that documents can be located at any level, rather than exclusively at leaf nodes. Because all content for each document is generated from the hierarchy node at which it sits, the generative distributions must be formed by chaining together conjugate priors, requiring more complex inference.

In network data, clustering is often called "community discovery" [23]. Graph-based approaches such as normalized-cut [37] are fast

[2] e.g., the Open Directory Project, `http://www.dmoz.org`

and deterministic, but often require the desired number of clusters to be specified in advance, and do not easily generalize to hierarchical models. SHRINK [22] induces a hierarchical clustering that prioritizes high modularity, while tolerating hubs and outliers that violate the traditional hierarchical structure. However, our work is more closely related to probabilistic approaches, which provide a principled way to combine content and network structure. Clauset et al. show that hierarchical community discovery can be obtained using a Monte Carlo sampling algorithm; the generative model assigns a link probability at each node in the hierarchy, and the sampling moves then converge on a stationary distribution centered on a hierarchy with high likelihood of generating the observed links [9]. However, this model is restricted to dendrograms, or binary trees, which are unlike the flatter hierarchies produced by human curators.

An alternative line of work on network clustering begins with the Stochastic Block Model (SBM) [21]. The SBM is a generative model in which nodes are partitioned into communities, which in turn determine the link probabilities. This idea was extended in the mixed-membership stochastic blockmodel (MMSB) [2], where each node has a mixed-membership vector over possible "roles"; an additional pair of latent variables selects the roles that are relevant for each potential network connection. The multiscale community block model (MSCB) places this idea in a non-parametric hierarchical setting: each document is associated with a path through a hierarchy, and the roles correspond to levels on the path [20]. Both the MSCB and MMSB assign latent variables to every *potential* link, so that each sampling pass requires $\mathcal{O}(N^2)$ complexity in the number of nodes.

A key feature of TopicBlock is that we merge text and network data, with the goal of inducing a more robust hierarchy and enabling applications in which the two modalities can help to explain each other. Mei et al. combine latent topic models with network information by compiling the network into a regularizer that encourages the topic proportions of linked documents to be similar [28]. This approach encodes the network into the structure of the generative model, so it does not permit probabilistic inferences about the likelihood of additional network connections. Topic-sensitive PageRank [17] takes a different notion of "topic," seeding each topic with documents from the top-level categories of the manually-curated Open Directory Project hierarchy. This method is designed to support information retrieval, and does not permit probabilistic modeling of new content or unseen links. Unlike both of these approaches, TopicBlock is generative over both text and links.

Much of the prior work on joint generative models of text and links falls into two classes. In one class, the identity of the target and/or source of the link is encoded as a discrete random variable [10, 29, 14, 27, 35]. Such models permit probabilistic inference within the documents in the training set, but they are closed to outside documents; it is not possible to use the text of an unseen document to predict who will link to it. In the second class of models, each link is a binary random variable generated from a Bernoulli distribution that is parametrized by the topical similarity of the documents. In the Relational Topic Model (RTM), the link probability is a function of the topical similarity [8] (Liu et al. extend the RTM by incorporating a per-document "community" membership vector [25]). The RTM treats non-edges as *hidden* data, so its complexity is linear in the number of edges, and thus less than the $\mathcal{O}(N^2)$ required by the blockmodel variants. Such a model is encouraged to assign arbitrarily high likelihood to the observed links, leading to instability in the parameter estimates, which must be corrected by a regularization heuristic. In contrast, we model both edges and non-edges probabilistically, achieving sub-quadratic complexity by limiting the flexibility of the link probability model.

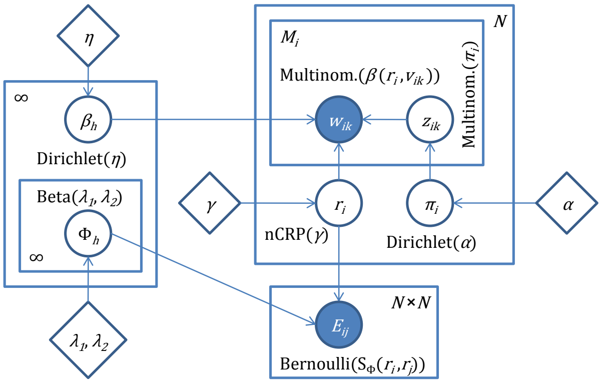

Figure 2: Graphical model illustration

3. MODEL DESCRIPTION

TopicBlock treats document text and relational links as emissions from a latent hierarchy, which has fixed depth L but a nonparametric branching factor at each non-terminal. Each document is represented as a complete *path* through the hierarchy, with words generated from a mixture across levels in the path, and links generated directly from the paths. We now present the model in detail. A summary of the hypothesized generative process is presented in Table 1, and a plate diagram is shown in Figure 2.

3.1 Latent hierarchy

Each document's position in the hierarchy is denoted by an $L \times 1$ vector of integers $r_i \in \mathbb{Z}^L$, which we call a *path*. The path is interpreted as follows: r_{i1} denotes the hierarchy branch taken by document i from level 0 (the implicit root, denoted by r_0) to level 1, r_{i2} denotes the branch taken from level 1 to level 2 *relative* to r_{i1} (the branch just taken), and so forth. Example: $r_i = (2, 1, 3, \dots)$ says that entity i is reached by taking the 2nd branch from the root, then the 1st branch at the node we just arrived at, followed by the 3rd branch at the next node, etc. The set of all paths r_i fully determines the shape of the hierarchy.

The nested Chinese Restaurant Process (nCRP) provides a suitable Bayesian prior for non-parametric hierarchies [7]. Each path is obtained by making a series of draws from standard Chinese Restaurant Processes associated with each node in the hierarchy. This prior displays the "rich-get-richer" property: at each level, a draw is likely to follow branches taken by previous documents; however, there is always a possibility of choosing a new branch which has never been taken before. Blei *et al.* [7] show that this model permits collapsed sampling in a way that follows naturally from the original Chinese Restaurant Process.

3.2 Generating words and links

We assume that each document $i \in \{1, \dots, N\}$ is associated with two kinds of observed data. The first is a collection of words w, where w_{ik} denotes the k-th word associated with document i, and M_i is the total number of word tokens in document i. The second type of observation is a collection of directed links to other documents, referred to as a *network*. This network is given as an $N \times N$ adjacency matrix E, such that $E_{ij} = 1$ denotes the presence of a (directed) link from document i to document j, while $E_{ij} = 0$ denotes its absence. We ignore self-links E_{ii}.

Every node in the hierarchy represents a distribution over words and links; documents whose path contains a hierarchy node h can draw their words and links from the distributions in h. More formally, every hierarchy node h is associated with two distributions. For the text, we define a set of vocabularies β_h which generate words w_{ik}; specifically, β_h is a V-dimensional multinomial parameter representing a distribution over words, as in LDA. For the links, we define a set of *link-density* probabilities Φ_h. Here, Φ_h is the probability of generating a link between documents whose paths both contain hierarchy node h. In cases where two document paths share multiple hierarchy nodes, we take h to be the *deepest shared node*, which may be the root of the tree.

3.2.1 Words

Document text is generated from a bag-of-words model, in which each word is produced by some node along the document's path through the hierarchy. On this view, some words will be general and could appear in any document (these words are drawn from the root) while others will be specific (these are drawn from a leaf). This encourages a hierarchy in which the most similar documents are grouped at the leaf level, while moderately similar documents are grouped at coarser levels of the hierarchy.

More formally, the words for document i are generated from a mixture of the β distributions found along the path r_i, including the implicit root. Each word w_{ik} associated with document i can be generated by *any* of the path nodes $r_{i1}, \ldots, r_{iL}$ or the root r_0. The specific path node chosen to generate w_{ik} is given by a *level indicator* $z_{ik} \in \{0, \ldots, L\}$, for example, $z_{ik} = 3$ means that w_{ik} is generated from the vocabulary β_h associated with the hierarchy node at (r_{i1}, r_{i2}, r_{i3}). These level indicators z_{ik} are drawn from $(L + 1)$-dimensional multinomial parameters π_i, which we refer to as *level distributions*. Intuitively, these represent document i's preference for shallower or deeper hierarchy levels.

3.2.2 Links

The generative model for links between documents is motivated by the intuition that the non-terminals of the hierarchy represent progressively more specific communities of documents. While one might explicitly model the link probability between, say, organic chemistry and ancient Greek history (as distinct from the likelihood of links from organic chemistry to ancient Roman history), a much simpler and more tractable model can be obtained by using the hierarchical structure to abstract this relationship. We make the simplifying assumption that relations between communities in disparate parts of the hierarchy can be summarized by their deepest common ancestor. As a result, the number of parameters grows linearly rather than quadratically with the number of non-terminals.

More formally, each nonterminal h has an associated Bernoulli parameter Φ_h, which indicates the link-likelihood between documents that share h as their deepest common ancestor. We define $\mathrm{S}(r_i, r_j)$ as a function that selects the deepest shared Φ_h between the paths r_i, r_j:

$$\begin{aligned} &\mathrm{S}_\Phi(r_i, r_j) := \Phi_h \\ &h := (r_{i1}, \ldots, r_{i\omega}), \qquad \omega := \arg\max_{k \geq 0} \mathbb{I}(r_{i,1:k} = r_{j,1:k}), \end{aligned} \tag{1}$$

so that,

$$P(\mathbf{E} \mid \mathbf{r}, \Phi) = \prod_{i, j \neq i} \mathrm{S}_\Phi(r_i, r_j)^{E_{ij}} (1 - \mathrm{S}_\Phi(r_i, r_j))^{1 - E_{ij}}.$$

- Draw the hierarchy — for each entity i:
  - Path $r_i \sim \mathrm{nCRP}(\gamma)$
  - Word level distribution $\pi_i \sim \mathrm{Dirichlet}(\alpha)$
- Draw hierarchy node parameters — for each node h:
  - Word probabilities $\beta_h \sim \mathrm{Dirichlet}(\eta_{\mathrm{depth}(h)})$
  - Link probabilities $\Phi_h \sim \mathrm{Beta}(\lambda_1, \lambda_2)$
- Draw text — for each entity i and word k:
  - Word level $z_{ik} \sim \mathrm{Multinomial}(\pi_i)$
  - Word $w_{ik} \sim \mathrm{Multinomial}(\beta_h)$, where h is the hierarchy node at $(r_{i,1}, \ldots, r_{i,z_{ik}})$
- Draw network — for each pair of entities i and $j \neq i$:
  - Link $E_{ij} \sim \mathrm{Bernoulli}(\mathrm{S}_\Phi(r_i, r_j))$, where $\mathrm{S}_\Phi()$ is defined in Section 3.2

Table 1: The generative process for TopicBlock's model of text and relational connections

The likelihood is a product over all N^2 potential links, but as we will see, it can be computed in fewer than $\mathcal{O}(N^2)$ steps. Note that $\mathrm{S}_\Phi(r_i, r_j)$ will select the root parameter Φ_{r_0} when r_i and r_j are completely different.

3.3 Parameters

TopicBlock has four parameter types: the paths r_i, level distributions π_i, word probabilities β_h, and the link probabilities Φ_h. Each parameter is drawn from a suitable prior: the paths r_i are drawn from a depth-L $\mathrm{nCRP}(\gamma)$; the level distributions π_i are drawn from $\mathrm{Dirichlet}(\alpha)$; the topics β_h are drawn from a symmetric $\mathrm{Dirichlet}(\eta_k)$ (where k is the depth of node h); and the link probabilities Φ_h are drawn from $\mathrm{Beta}(\lambda_1, \lambda_2)$. The hyperparameter $\gamma > 0$ is an $L \times 1$ vector, while $\alpha, \eta > 0$ are $(L + 1) \times 1$ vectors, and $\lambda_1, \lambda_2 > 0$ are scalars.

4. INFERENCE

Exact inference on our model is intractable, so we derive a collapsed Gibbs sampler for posterior inference [34]. We integrate out π, β and Φ for faster mixing (collapsed sampling for topic models was introduced in [13]), so we need sample only the paths $\mathbf{r}$ and word levels $\boldsymbol{z}$. We present the sampling distributions for these parameters now.

Word levels $\boldsymbol{z}$. The sampling distribution of z_{ik} is

$$\begin{aligned} &\mathbb{P}(z_{ik} \mid \mathbf{r}, \boldsymbol{z}_{-(ik)}, \mathbf{E}, \mathbf{w}) \\ &\quad \propto \mathbb{P}(w_{ik}, z_{ik} \mid \mathbf{r}, \boldsymbol{z}_{-(ik)}, \mathbf{E}, \mathbf{w}_{-(ik)}) \\ &\quad = \mathbb{P}(w_{ik} \mid \mathbf{r}, \boldsymbol{z}, \mathbf{w}_{-(ik)}) \mathbb{P}(z_{ik} \mid \boldsymbol{z}_{i,(-k)}) \end{aligned} \tag{2}$$

where $\boldsymbol{z}_{i,(-k)} = \{z_{i\cdot}\} \setminus z_{ik}$ and $\mathbf{w}_{-(ik)} = \{w_{\cdot}\} \setminus w_{ik}$. The first term represents the likelihood; for a particular value of z_{ik}, it is

$$\begin{aligned} &\mathbb{P}(w_{ik} \mid \mathbf{r}, \boldsymbol{z}, \mathbf{w}_{-(ik)}) = \frac{\eta_{z_{ik}} + a_{w_{ik}}}{V \eta_{z_{ik}} + \sum_{v=1}^{V} a_v}, \\ &a_v = \big|\{(x, y) \mid (x, y) \neq (i, k), z_{xy} = z_{ik}, \\ &\qquad (r_{x1}, \ldots, r_{x z_{xy}}) = (r_{i1}, \ldots, r_{i z_{ik}}), w_{xy} = v\}\big|. \end{aligned} \tag{3}$$

In plain English, a_v is the number of words w_{xy} equal to v (excluding w_{ik}) and coming from hierarchy position $(r_{i1}, \ldots, r_{i z_{ik}})$. Thus, we are computing the empirical frequency of emitting word v, smoothed by level z_{ik}'s symmetric Dirichlet prior $\eta_{z_{ik}}$.

The second term represents the prior on z_{ik}:

$$\mathbb{P}(z_{ik} \mid \boldsymbol{z}_{i,(-k)}) = \frac{\alpha_{z_{ik}} + \#[\boldsymbol{z}_{i,(-k)} = z_{ik}]}{\sum_{\ell=1}^{L} \alpha_\ell + \#[\boldsymbol{z}_{i,(-k)} = \ell]}. \tag{4}$$

Paths $\mathbf{r}$. The sampling distribution for the path r_i is

$$\begin{aligned}
&\mathbb{P}(r_i \mid \mathbf{r}_{-i}, \boldsymbol{z}, \mathbf{E}, \mathbf{w}) && (5)\\
&\quad \propto \mathbb{P}(r_i, \mathbf{E}_{(i\cdot),(\cdot i)}, \mathbf{w}_i \mid \mathbf{r}_{-i}, \boldsymbol{z}, \mathbf{E}_{-(i\cdot),-(\cdot i)}, \mathbf{w}_{-i})\\
&\quad = \mathbb{P}(\mathbf{E}_{(i\cdot),(\cdot i)} \mid \mathbf{r}, \mathbf{E}_{-(i\cdot),-(\cdot i)}) \mathbb{P}(\mathbf{w}_i \mid \mathbf{r}, \boldsymbol{z}, \mathbf{w}_{-i}) \mathbb{P}(r_i \mid \mathbf{r}_{-i})
\end{aligned}$$

where $\mathbf{w}_i = \{w_{i\cdot}\}$ is the set of tokens in document i, and $\mathbf{w}_{-i}$ is its complement. $\mathbf{E}_{(i,\cdot),(\cdot,i)} = \{E_{xy} \mid x = i \vee y = i\}$ is the set of all links touching document i and $\mathbf{E}_{-(i,\cdot),-(\cdot,i)}$ is its complement. In particular, the set $\mathbf{E}_{(i,\cdot),(\cdot,i)}$ is just the i-th row and i-th column of the adjacency matrix $\mathbf{E}$, sans the self-link E_{ii}.

Equation 5 decomposes into three terms, corresponding to link likelihoods, word likelihoods, and the path prior distribution respectively. The first term represents the link likelihoods for all links touching document i. These likelihoods are Bernoulli distributed, with a Beta prior; marginalizing the parameter Φ yields a Beta-Bernoulli distribution, which has an analytic closed-form:

$$\prod_{\Phi \in \mathbf{\Phi}_{(i\cdot),(\cdot i)}} \frac{\Gamma(A+B+\lambda_1+\lambda_2)}{\Gamma(A+\lambda_1)\Gamma(B+\lambda_2)} \cdot \frac{\Gamma(A+C+\lambda_1)\Gamma(B+D+\lambda_2)}{\Gamma(A+B+C+D+\lambda_1+\lambda_2)} \tag{6}$$

$$\begin{aligned}
\mathbf{\Phi}_{(i\cdot),(\cdot i)} &= \{\Phi_h \mid \exists(x,y)[E_{xy} \in \mathbf{E}_{(i\cdot),(\cdot i)}, \mathrm{S}_{\Phi}^{xy} = \Phi_h]\}\\
A &= \left|\{(x,y) \mid E_{xy} \in \mathbf{E}_{-(i\cdot),-(\cdot i)}, \mathrm{S}_{\Phi}^{xy} = \Phi, E_{xy} = 1\}\right|\\
B &= \left|\{(x,y) \mid E_{xy} \in \mathbf{E}_{-(i\cdot),-(\cdot i)}, \mathrm{S}_{\Phi}^{xy} = \Phi, E_{xy} = 0\}\right|\\
C &= \left|\{(x,y) \mid E_{xy} \in \mathbf{E}_{(i\cdot),(\cdot i)}, \mathrm{S}_{\Phi}^{xy} = \Phi, E_{xy} = 1\}\right|\\
D &= \left|\{(x,y) \mid E_{xy} \in \mathbf{E}_{(i\cdot),(\cdot i)}, \mathrm{S}_{\Phi}^{xy} = \Phi, E_{xy} = 0\}\right|
\end{aligned}$$

where $\mathbf{\Phi}_{(i\cdot),(\cdot i)}$ is the set of all link probability parameters Φ_h touched by the link set $\mathbf{E}_{(i,\cdot),(\cdot,i)}$. Observe that only those Φ_h along path r_i (or the root) can be in this set, thus it has size $|\mathbf{\Phi}_{(i\cdot),(\cdot i)}| \leq L+1$. Also, note that the terms A, B, C, D depend on Φ. The second term of Equation 5 represents the word likelihoods:

$$\prod_{\ell=1}^{L} \frac{\Gamma(V\eta_\ell + \sum_{v=1}^{V} G_{\ell,v})}{\prod_{v=1}^{V} \Gamma(G_{\ell,v}+\eta_\ell)} \cdot \frac{\prod_{v=1}^{V} \Gamma(G_{\ell,v}+H_{\ell,v}+\eta_\ell)}{\Gamma(V\eta_\ell + \sum_{v=1}^{V} G_{\ell,v}+H_{\ell,v})} \tag{7}$$

$$\begin{aligned}
G_{\ell,v} &= |\{(x,y) \mid x \neq i, z_{xy} = \ell,\\
&\qquad (r_{x1}, \ldots, r_{x\ell}) = (r_{i1}, \ldots, r_{i\ell}), w_{xy} = v\}|\\
H_{\ell,v} &= |\{y \mid z_{iy} = \ell, w_{iy} = v\}|
\end{aligned}$$

where V is the vocabulary size. $G_{\ell,v}$ is just the number of words in $\mathbf{w}_{-i}$ equal to v and coming from hierarchy position $(r_{i1}, \ldots, r_{i\ell})$. $H_{\ell,v}$ is similarly defined, but for words in $\mathbf{w}_i$.

The third term of Equation 5 represents the probability of drawing the path r_i from the nCRP, and can be computed recursively for all levels ℓ,

$$\mathrm{P}(r_{i\ell} = x \mid \mathbf{r}_{-i}, r_{i,1:(\ell-1)}) = \tag{8}$$

$$\begin{cases} \frac{\left|\{j \neq i \mid r_{j,1:(\ell-1)} = r_{i,1:(\ell-1)}, r_{j\ell} = x\}\right|}{\left|\{j \neq i \mid r_{j,1:(\ell-1)} = r_{i,1:(\ell-1)}\}\right| + \gamma_\ell} & \text{if } x \text{ is an existing branch,}\\ \frac{\gamma_\ell}{\left|\{j \neq i \mid r_{j,1:(\ell-1)} = r_{i,1:(\ell-1)}\}\right| + \gamma_\ell} & \text{if } x \text{ is a new branch} \end{cases}$$

This equation gives the probability of path r_i taking branch x at depth ℓ. At step ℓ in the path, the probability of following an existing branch is proportional to the number of documents already in that branch, while the probability of creating a new branch is proportional to γ_ℓ.

Hyperparameter Tuning. The hyperparameters $\gamma, \alpha, \eta, \lambda_1, \lambda_2$ significantly influence the size and shape of the hierarchy. We automatically choose suitable values for them by endowing γ, α, η with a symmetric Dirichlet(1) hyperprior, and λ_1, λ_2 with an $\mathrm{Exponential}(1)$ hyperprior. Using the Metropolis-Hastings algorithm with these hyperpriors as proposal distributions, we sample new values for $\gamma, \alpha, \eta, \lambda_1, \lambda_2$ after every Gibbs sampling iteration.

4.1 Linear time Gibbs sampling

To be practical on larger datasets, each Gibbs sampling sweep must have runtime linear in both the number of tokens and the number of 1-links $E_{ij} = 1$. This is problematic for standard implementations of generative network models such as ours, because we are modeling the generative probability of all 1-links *and* 0-links. The sufficient statistics for each Φ_h are the number of 1-links and 0-links, and these statistics must be updated when we resample the paths r_i. Naïvely updating these parameters would take $\mathcal{O}(N)$ time since there are $2N-2$ links touching document i. It follows that a Gibbs sampling sweep over all r_i would require $\mathcal{O}(N^2)$ quadratic runtime.

The solution is to maintain an *augmented* set of sufficient statistics for Φ_h. Define $h \subseteq r_i$ to be true if path r_i passes through node h. Then the augmented sufficient statistics are:

1. $U_{h,i} = \sum_{j \neq i} (\mathbf{E}_{ij} + \mathbf{E}_{ji}) \mathbb{I}(h \subseteq r_i, h \subseteq r_j)$, the number of 1-links touching document i and drawn from Φ_h *and* its descendants.
2. $U_h = \sum_{i,j} \mathbf{E}_{ij} \mathbb{I}(h \subseteq r_i, h \subseteq r_j)$, the number of 1-links drawn from Φ_h *and* its hierarchy descendants.
3. $u_h = \sum_{h' \in \mathrm{children}(h)} U_{h'}$, the number of 1-links drawn from Φ_h's descendants *only*.
4. $T_h = \sum_i \mathbb{I}(h \subseteq r_i)$, the number of documents at h *and* its descendants.
5. $t_h = \sum_{h' \in \mathrm{children}(h)} T_{h'}$, the number of documents at h's descendants *only*.

The number of 0- or 1-links specifically at Φ_h is given by

$$\begin{aligned}
&\#[\text{1-links at } h] = U_h - u_h && (9)\\
&\#[\text{0-links at } h] = [(T_h)(T_h - 1) - (t_h)(t_h - 1)] - (U_h - u_h)
\end{aligned}$$

Before sampling a new value for document i's path r_i, we need to remove its edge set $\mathbf{E}_{(i,\cdot),(\cdot,i)}$ from the above sufficient statistics. Once r_i has been sampled, we need to add $\mathbf{E}_{(i,\cdot),(\cdot,i)}$ back to the sufficient statistics, based on the new r_i. Algorithms 1, 2 perform these operations efficiently; observe that they run in $\mathcal{O}(P_i L)$ time where P_i is the number of 1-links touching document i. Letting P be the total number of 1-links in $\mathbf{E}$, we see that a Gibbs sampler sweep over all r_i spends $\mathcal{O}(PL)$ time updating Φ_h sufficient statistics, which is linear in P.

The remaining work for sampling r_i boils down to (1) calculating existing and new path probabilities through the hierarchy, and (2) updating sufficient statistics related to the vocabularies β. Calculating the path probabilities requires $\mathcal{O}(HLV)$ time, where H is the number of hierarchy nodes and V is the vocabulary size; updating the vocabularies requires $\mathcal{O}(M_i L)$ time where M_i is the number of tokens w_{ik} belonging to document i. Thus, the total runtime required to sweep over all r_i is $\mathcal{O}(PL + NHLV + ML)$ where M is the total number of tokens $\mathbf{w}$. Treating L, H, V as constants and noting that $N \leq M$, we see that sampling all r_i is indeed linear in the number of tokens M and number of 1-links P. We also need to sample each word level z_{ik}, which takes $\mathcal{O}(L)$ time

Algorithm 1 Removing document i from sufficient statistics of Φ_h

```
Let $h_0, \ldots, h_L$ be the hierarchy nodes along $r_i$.
Let $A$ be a temporary variable.
for $\ell = L \ldots 0$ do
    if $\ell < L$ then
        $u_{h_\ell} \leftarrow u_{h_\ell} - (A - U_{h_{\ell+1}})$
        $t_{h_\ell} \leftarrow t_{h_\ell} - 1$
    end if
    $A \leftarrow U_{h_\ell}$        (Store the original value of $U_{h_\ell}$)
    $U_{h_\ell} \leftarrow U_{h_\ell} - U_{h_\ell,i}$
    $T_{h_\ell} \leftarrow T_{h_\ell} - 1$
    for $j$ s.t. $j \in \text{Neighbors}(i)$ and $h_\ell \subseteq r_j$ do
        $U_{h_\ell,j} \leftarrow U_{h_\ell,j} - \mathbb{I}(E_{ij} = 1) - \mathbb{I}(E_{ji} = 1)$
        $U_{h_\ell,i} \leftarrow U_{h_\ell,i} - \mathbb{I}(E_{ij} = 1) - \mathbb{I}(E_{ji} = 1)$
    end for
end for
```

Algorithm 2 Adding document i to sufficient statistics of Φ_h

```
Let $h_0, \ldots, h_L$ be the hierarchy nodes along $r_i$.
Let $A$ be a temporary variable.
for $\ell = L \ldots 0$ do
    if $\ell < L$ then
        $u_{h_\ell} \leftarrow u_{h_\ell} + (U_{h_{\ell+1}} - A)$
        $t_{h_\ell} \leftarrow t_{h_\ell} + 1$
    end if
    for $j$ s.t. $j \in \text{Neighbors}(i)$ and $h_\ell \subseteq r_j$ do
        $U_{h_\ell,j} \leftarrow U_{h_\ell,j} + \mathbb{I}(E_{ij} = 1) + \mathbb{I}(E_{ji} = 1)$
        $U_{h_\ell,i} \leftarrow U_{h_\ell,i} + \mathbb{I}(E_{ij} = 1) + \mathbb{I}(E_{ji} = 1)$
    end for
    $A \leftarrow U_{h_\ell}$        (Store the original value of $U_{h_\ell}$)
    $U_{h_\ell} \leftarrow U_{h_\ell} + U_{h_\ell,i}$
    $T_{h_\ell} \leftarrow T_{h_\ell} + 1$
end for
```

(including sufficient statistic updates) for a total of $\mathcal{O}(ML)$ linear work over all $\boldsymbol{z}$. Finally, the hyperparameter tuning steps require us to compute the probability of all tokens $\mathbf{w}$ and links $\mathbf{E}$ given the paths $\mathbf{r}$ and word levels $\boldsymbol{z}$, which can be performed in at most linear $\mathcal{O}(PL + ML)$ time. Since we only update the hyperparameters once after every Gibbs sampling sweep, our total runtime per sweep remains linear.

We contrast our linear efficiency with alternative models such as the Mixed-Membership Stochastic Block Model (MMSB [2]) and Pairwise Link-LDA [29]. The published inference techniques for these models are quadratic in the number of nodes, so it would be very difficult for serial implementations to scale to the 10^4 node datasets that we handle in this paper.

5. DATA

We evaluate our system on two corpora: Wikipedia and the ACL Anthology. The Wikipedia dataset is meant to capture familiar concepts which are easily comprehended by non-experts; the ACL Anthology dataset tests the ability of our model to build reasonable taxonomies for more technical datasets. We expect different network behavior for the two datasets: a Wikipedia page can contain an arbitrary number of citations, while research articles may be space-limited, and can only cite articles which have already been published. Thus, the ACL dataset may fail to include many links which would seem to be demanded by the text, but were omitted due to space constraints or simply because the relevant article had not yet been published. The Wikipedia dataset poses its own challenges, as some links are almost completely unrelated to document topical content. For example, the article on DNA contains a link to the article on Switzerland, because DNA was first isolated by a Swiss scientist.

	Wikipedia	ACL Anthology
documents	14,675	15,032
tokens	1,467,500	2,913,665
links	134,827	41,112
vocabulary	10,013	2,505

Table 2: Basic statistics about each dataset

5.1 Simple English Wikipedia

Our first dataset is built from Wikipedia; our goal is to use the text and hyperlinks in this dataset to induce a hierarchical structure that reflects the underlying content and connections. We chose this dataset because the content is written at a non-technical level, allowing easy inspection for non-experts. The dataset supports the evaluation of *link resolution* (defined in Section 6.3).

There is previous work on modeling the topics underlying Wikipedia data [14, 32]. Gruber et al. [14] constructed a small corpus of text and links by crawling 105 pages starting from the page for the NIPS conference, capturing 799 in-collection links. Our goal was a much larger-scale evaluation; in addition, we were concerned that a crawl-based approach would bias the resulting network to implicitly reflect a hierarchical structure (centered on the seed node) and an unusually dense network of links.

Instead of building a dataset by crawling, we downloaded the entire "Simple English" Wikipedia, a set of 133,462 articles written in easy-to-read English. Many of these documents are very short, including placeholders for future articles. We limited our corpus to documents that were at least 100 tokens in length (using the LingPipe tokenizer [3]), and considered only articles (ignoring discussion pages, templates, etc.). This resulted in a corpus of 14675 documents. The link data includes all 152,674 in-collection hyperlinks; the text data consists of the first 100 tokens of each document, resulting in a total of 1,467,500 tokens. We limited the vocabulary to all words appearing at least as frequently as the 10,000th most frequent word, resulting in a total vocabulary of 10,013. We apply a standard filter to remove stopwords [24].

5.2 ACL Anthology

Our second dataset is based on the scientific literature, which contains both text and citations between documents. The ACL anthology is a curated collection of papers published in computational linguistics venues, dating back to 1965 [5]. We downloaded the 2009 release of this dataset, including papers up to that year, for a total of 15,032 documents. Taxonomy induction on research corpora can serve an important function, as manually-curated taxonomies always risk falling behind new developments which may splinter new fields or unite disparate ones. As noted above, we use the entire ACL Anthology dataset from 1965 to 2009. We limit the vocabulary to 2,500 terms, and limit each document to the first 200 tokens — roughly equivalent to the title and abstract — and remove stopwords [24].

There is substantial previous work on the ACL Anthology, including temporal and bibliometric analysis [16, 33], citation prediction [4], and recognition of latent themes [15] and influence [12, 30]. However, none of this work has considered the problem of inducing hierarchical structure of the discipline of computational linguistics.

Our quantitative evaluation addresses the citation-prediction task considered by Bethard and Jurafsky [4]. Following their methodology, we restrict our quantitative analysis to the 1,739 journal and conference papers from 2000 to 2009. Our version of the corpus is a more recent release, so our data subset is very similar but not identical to their evaluation set.

6. QUANTITATIVE ANALYSIS

We present a series of quantitative and qualitative evalutions of TopicBlock's ability to learn accurate and interpretable models of networked text. Our main evaluations (sections 6.2 and 6.3) test the ability of TopicBlock to predict and resolve ambiguous links involving heldout documents.

6.1 System Details

For all experiments, we use an $L = 2$ hierarchy (root plus two levels) unless otherwise stated. We initialize TopicBlock's document paths $\mathbf{r}$ by using a Dirichlet Process Mixture Model (essentially a one-level, text-only TopicBlock with no shared root) in a recursive clustering fashion, which provides a good starting hierarchy. From there, we ran our Gibbs sampler cum Metropolis-Hastings algorithm for 2,500 passes through the data or for 7 days, whichever came first; our slowest experiments completed at least 1,000 passes. All experiments were run with 10 repeat trials, and results were always obtained from the most recent sample. We selected the best trial according to experimentally-relevant criteria: for the qualitative analyses (Section 7), we selected according to sample log-likelihood; in the citation prediction task we employed a development set; in the link resolution task we show the results of all trials.

6.2 Citation Prediction

Our citation prediction evaluation uses the induced TopicBlock hierarchy to predict outgoing citation links from documents which were not seen during training time. For this evaluation, we use the 1,739-paper ACL subset described earlier. Citation prediction has been considered in prior research; for example, Bethard and Jurafsky present a supervised algorithm that considers a broad range of features, including both content and citation information [4]. We view our approach as complementary; our hierarchical model could provide features for such a discriminative approach. He et al. attack the related problem of recommending citations in the context of a snippet of text describing the purpose of the citation [18], focusing on concept-based relevance between citing and cited documents. Again, one might combine these approaches by mining the local context to determine which part of the induced hierarchy is most likely to contain the desired citation.

Metric. We evaluate using *mean average precision*, an information retrieval metric designed for ranking tasks [26]. The *average precision* is the mean of the precisions at the ranks of all the relevant examples; *mean average precision* takes the mean of the average precisions across all queries (heldout documents). This metric can be viewed as an approximation to the area under the precision-recall curve.

Systems. We divided the 1,739-paper ACL subset into a training set (papers from 2000-2006), a development set (2006-2007), and a heldout set (2008-2009). For each experiment we conducted 10 trials, using the following procedure:

1. build a topic hierarchy from the training set using TOPICBLOCK,
2. fit the development set text to the learnt hierarchy, and predict development links,
3. retrieve the trial with the highest mean average precision over development set links,
4. fit the heldout set text to that trial's hierarchy, and predict heldout links,
5. compute mean average precision over heldout set links.

In essence, the development set is being used to select the best-trained model with respect to the citation prediction task. The final predictions were obtained by inferring each test document's most appropriate hierarchy path r given only its text, and then using the path r to predict links to training documents according to our network model.

Baselines. To evaluate the contribution of jointly modeling text with network structure, we compare against hierarchical latent Dirichlet allocation (HLDA) [7], a closely-related model which ignores network structure. We use our own implementation, which is based on the TOPICBLOCK codebase. As HLDA does not explicitly model links, we postfit a hierarchical blockmodel to the induced hierarchy over the training data; this hierarchy is learnt only from the text. Thus, the comparison with HLDA directly tests the contribution of network information to the quality of the hierarchy, over what the text already provides. After postfitting the blockmodel, we fit the development and heldout sets as described earlier.

We can also isolate the contribution of network information to the hierarchy, by learning the shape of the hierarchy based on network contributions but not text. After learning the hierarchy's shape (which is defined by the paths $\mathbf{r}$) this way, we postfit text topics to this hierarchy by running hLDA while keeping the paths $\mathbf{r}$ fixed. Then we fit the development and heldout sets as usual. This approach can be viewed as a hierarchical stochastic blockmodel, so we name the system HSBM.

Next, we consider a simpler text-only baseline, predicting links based on the term similarity between the query and each possible target document; specifically, we use the TF-IDF measure considered by Bethard and Jurafsky [4]. For a fair comparison, we use the same text which was available to TopicBlock and hLDA, which is the first 200 words of each document.

Finally, we consider a network-only baseline, where we rank potential documents in descending order of IN-DEGREE. In other words, we simply predict highly cited documents first.

Results. As shown in Table 3, TOPICBLOCK achieves the highest MAP score of all methods, besting the hierarchies trained using only text (HLDA) or only the network (HSBM). This demonstrates that inducing hierarchies from text and network modalities jointly yields quantitatively better performance than post-hoc fitting of one modality to a hierarchy trained on the other. In addition, all hierarchy-based methods beat the TF-IDF and IN-DEGREE baselines by a strong margin, validating the use of hierarchies over simpler, non-hierarchical alternatives.

System	Text?	Network?	Hierarchical?	MAP
TopicBlock	x	x	x	**0.137**
hLDA	x		x	0.117
hSBM		x	x	0.112
In-Degree		x		0.0731
TF-IDF	x			0.0144

Table 3: Results on the citation prediction task for the ACL Anthology data. Higher scores are better. Note that hLDA is equivalent to TopicBlock without the network component, while hSBM is equivalent to TopicBlock without text.

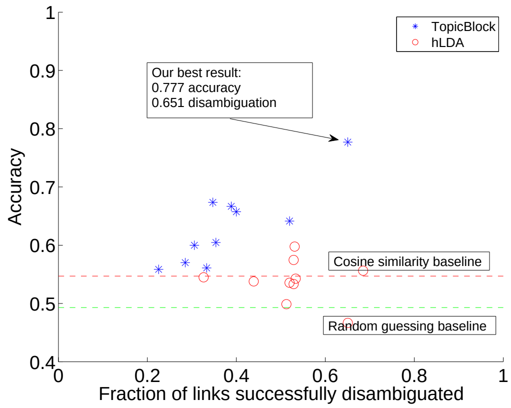

Figure 3: Wikipedia link resolution accuracy, plotted against proportion of links which could be resolved by the hierarchy.

6.3 Link Resolution

Wikipedia contains a substantial amount of name ambiguity, as multiple articles can share the same title. For example, the term “mac” may refer to the Media Access Control address, the luxury brand of personal computers, or the flagship sandwich from McDonalds. The *link resolution* task is to determine which possible reference article was intended by an ambiguous text string. In our Wikipedia data, there were 88 documents with the same base name, such as “scale_(music)" and “scale_(map)", and there were 435 references to such articles. These references were initially unambiguous, but we removed the bracketed disambiguation information in order to evaluate TopicBlock’s ability to resolve ambiguous references.

Systems. We run TopicBlock to induce a hierarchy over the training documents, and then learn the best paths r for each of the 88 ambiguous documents according to just their text. Then, for each of the 435 ambiguous references to the 88 target documents, we select the target with the highest link probability to the query document. If two targets are equally probable, we select the one with the highest text similarity according to TF-IDF. This experiment was conducted 10 times, and all results are shown in Figure 3. We also compare against hLDA, which is run in the same way as TopicBlock but trained without network information, using hierarchy path similarity instead of link probability to rank query documents. Finally, as a baseline we consider simply choosing the target with the highest text similarity.

Metric. The evaluation metric for this task is accuracy: the proportion of ambiguous links which were resolved correctly. In most cases the ambiguity set included only two documents, so more complicated ranking metrics are unnecessary.

Results. We performed ten different runs of TopicBlock and hLDA. In each run, a certain number of links could not be resolved by the hierarchy, because the target nodes were equally probable with respect to the query node — in these cases, we use the TF-IDF tie-breaker described above. Figure 3 plots the accuracy against the proportion of links which could be resolved by the hierarchy. As shown in the figure, TopicBlock is superior to the text similarity baseline on all ten runs. Moreover, the accuracy increases with the specificity of the hierarchy with regard to the ambiguous links — in other words, the added detail in these hierarchies coheres with the hidden hyperlinks. In contrast, hLDA is rarely better than the cosine similarity baseline, and does not improve in accuracy as the hierarchy specificity increases. This demonstrates that training from text alone will not yield a hierarchy that coheres with network information, while training from both modalities improves link disambiguation.

7. QUALITATIVE ANALYSIS

We perform a manual analysis to reveal the implications of our modeling decisions and inference procedure for the induced hierarchies, showcasing our model’s successes while highlighting areas for future improvement. Note that while the quantitative experiments in the previous section required holding out portions of the data, here we report topic hierarchies obtained by training on the entire dataset.

7.1 Wikipedia

Figure 1 shows a fragment of the hierarchy induced from the Simple English Wikipedia Dataset. Unlike our other experiments, we have used an $L = 3$ (root plus 3 levels) hierarchy here to capture more detail. We have provided the topic labels manually; overall we can characterize the top level as comprised of: history (W1), culture (W2), geography (W3), sports (W4), biology (W5), physical sciences (W6), technology (W7), and weapons (W8). The subcategories of the sports topic are shown in the figure, but the other subcategories are generally reasonable as well: for example biology (W5) divides into non-human and human subtopics; history (W1) divides into modern (W1.1), religious (W1.2), medieval (W1.3), and Japanese (W1.4). While a manually-created taxonomy would likely favor parallel structure and thus avoid placing a region (Japan) and a genre (religion) alongside two temporal epochs (modern and medieval), TopicBlock chooses an organization that reflects the underlying word and link distributions.

Figure 4 shows the link structure for the Wikipedia data, with the source of the link on the rows and the target on the columns. Documents are organized by their position in the induced hierarchy. Topic 1 has a very high density of incoming links, reflecting the generality of these concepts and their relation to many other documents. Overall, we see very high link density at the finest level of detail (indicated by small dark blocks directly on the diagonal), but we also see evidence of hierarchical link structure in the larger shaded blocks such as culture (W2) and physical science (W6).

7.2 ACL Anthology

The full ACL anthology hierarchy is shown in Figure 5, which gives the top words corresponding to each topic, by TF-IDF.[3] As

[3] Specifically, we multiplied the term frequency in the topic by the log of the inverse average term frequency across all topics [6].

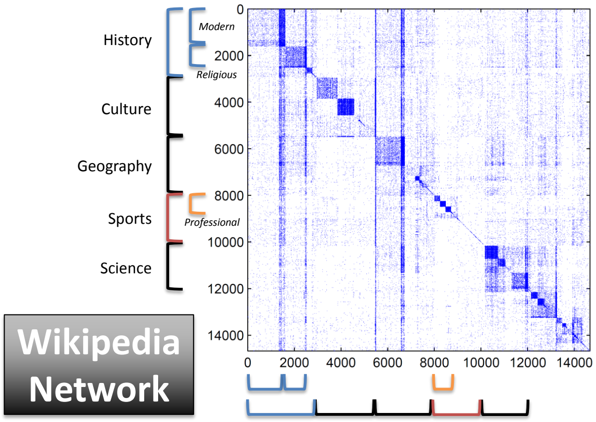

Figure 4: The network block matrix for the Simple English Wikipedia data.

before, the topic labels are provided by us; for simplicity we have chosen to focus on an $L = 2$-level hierarchy. The top-level categories include both application areas (interactive systems (A1) and information systems (A2)) as well as problem domains (discourse and semantics (A4); parsing (A6); machine translation (A8)). These areas are often close matches for the session titles of relevant conferences such as ACL.[4] At the second level, we again see coherent topical groupings: for example, the children of information systems include popular shared tasks such as named-entity recognition (A2.1), summarization (A2.3), and question answering (A2.4); the children of discourse and semantics (A4) include well-known theoretical frameworks, such as centering theory and propositional semantics (not shown here).

Occasionally, seemingly related topics are split into different parts of the tree. For example, the keywords for both topics A3 and A6.1 relate to syntactic parsing. Nonetheless, the citation links between these two topics are relatively sparse (see Figure 6), revealing a more subtle distinction: A3 focuses on representations and rule-driven approaches, while A6.1 includes data-driven and statistical approaches.

As in the Wikipedia data, the network diagram (Figure 6) reveals evidence of hierarchical block structures. For example, A2 contains 4101 links out of 4.4 million possible, a density of $9.3 * 10^{-4}$. This is substantially larger than the background density $1.8 * 10^{-4}$, but less than subtopics such as A2.1, which has a density of $6.4 * 10^{-3}$. We observe similar multilevel density for most of the high-level topics, except for interactive systems (A1), which seems to be more fractured. One of the densest topics is machine translation (A8), an area of computational linguistics which has become sufficiently distinct as to host its own conferences.[5]

One could obtain more parallel structure by imposing a domain-specific solution for research papers, such as Gupta and Manning's work on identifying the "focus, technique, and domain" of each article [15]; of course, such a solution would not necessarily generalize to Wikipedia articles or other document collections. While parallel structure is desirable, it is often lacking even in taxonomies produced by human experts. For example, a similar critique might be leveled at the sessions associated with a research conference, or even the ACM taxonomy.[6]

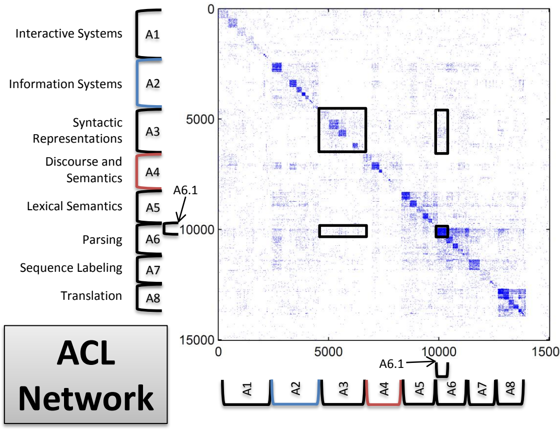

Figure 6: The network block matrix for the ACL Anthology Data. Blocks corresponding to links within/between A3 and A6.1 have been delineated by black rectangles. There are 2190 and 2331 citation links within A3 and A6.1 respectively, but only 343 links between them.

8. CONCLUSION

We have presented TopicBlock, a hierarchical nonparametric model for text and network data. By treating these two modalities jointly, we not only obtain a more robust latent representation, but are also able to better understand the relationship between the text and links. Applications such as link prediction, document clustering, and link ambiguity resolution demonstrate the strengths of our approach. In the future we plan to consider richer structures, such as multiple hierarchies which capture alternative possible decompositions of the document collection. We also plan to investigate dynamic models, in which temporal changes in the hierarchy may reveal high-level structural trends in the underlying data. Finally, in many practical settings one may obtain a partially-complete initial taxonomy from human annotators. An interesting future direction would be to apply techniques such as TopicBlock to refine existing taxonomies [40].

Acknowledgments This work was supported by NSF IIS-0713379, NSF IIS-1111142, NSF DBI-0546594 (Career), ONR N000140910758, AFOSR FA9550010247, NIH 1R01GM093156, and an Alfred P. Sloan Research Fellowship to Eric P. Xing. Qirong Ho is supported by a graduate fellowship from the Agency for Science, Technology and Research, Singapore.

9. REFERENCES

[1] R. P. Adams, Z. Ghahramani, and M. I. Jordan. Tree-Structured stick breaking processes for hierarchical data. In *Neural Information Processing Systems*, June 2010.

[2] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, 2008.

[3] Alias-i. Lingpipe 3.9.1, 2010.

[4] S. Bethard and D. Jurafsky. Who should I cite: learning literature search models from citation behavior. In *Proceedings of CIKM*, pages 609–618, 2010.

[5] S. Bird, R. Dale, B. J. Dorr, B. Gibson, M. T. Joseph, M.-y. Kan, D. Lee, B. Powley, D. R. Radev, and Y. F. Tan. The ACL anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of LREC*, 2008.

[6] D. Blei and J. Lafferty. Topic models. In *Text Mining: Theory and Applications*. Taylor and Francis, 2009.

[4] http://www.acl2011.org/program.utf8.shtml

[5] http://www.amtaweb.org/

[6] http://www.computer.org/portal/web/publications/acmtaxonomy

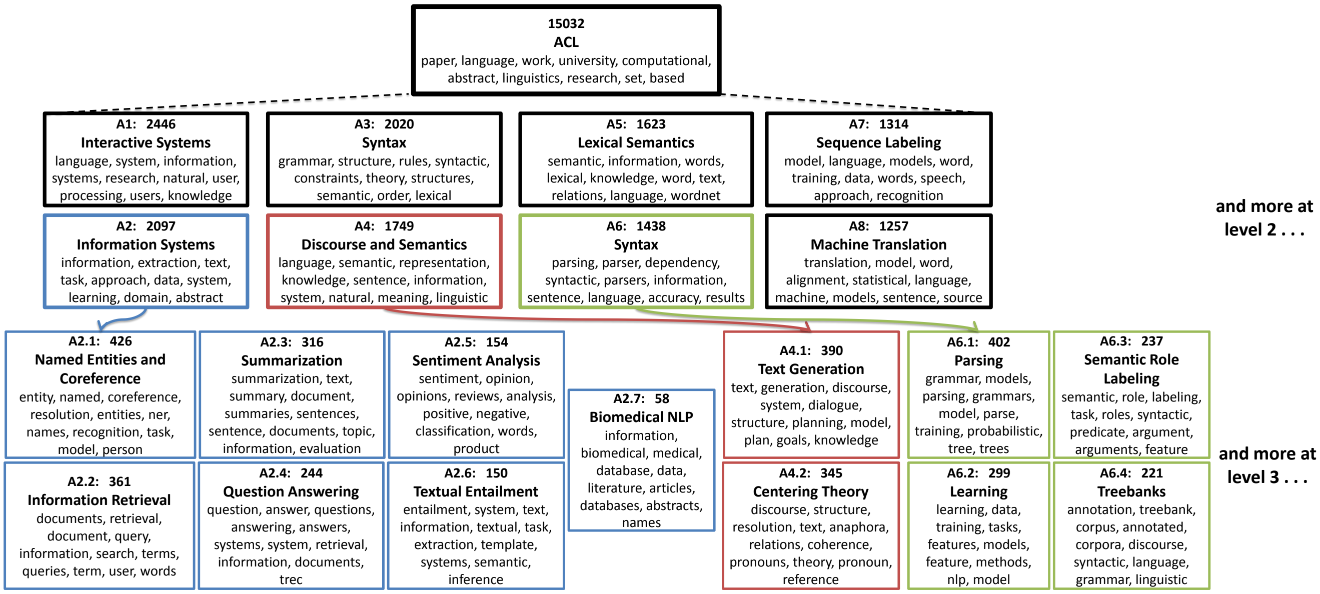
15032
ACL
paper, language, work, university, computational, abstract, linguistics, research, set, based
A1: 2446
Interactive Systems
language, system, information, systems, research, natural, user, processing, users, knowledge
A3: 2020
Syntax
grammar, structure, rules, syntactic, constraints, theory, structures, semantic, order, lexical
A5: 1623
Lexical Semantics
semantic, information, words, lexical, knowledge, word, text, relations, language, wordnet
A7: 1314
Sequence Labeling
model, language, models, word, training, data, words, speech, approach, recognition
and more at level 2 . . .
A2: 2097
Information Systems
information, extraction, text, task, approach, data, system, learning, domain, abstract
A4: 1749
Discourse and Semantics
language, semantic, representation, knowledge, sentence, information, system, natural, meaning, linguistic
A6: 1438
Syntax
parsing, parser, dependency, syntactic, parsers, information, sentence, language, accuracy, results
A8: 1257
Machine Translation
translation, model, word, alignment, statistical, language, machine, models, sentence, source
A2.1: 426
Named Entities and Coreference
entity, named, coreference, resolution, entities, ner, names, recognition, task, model, person
A2.3: 316
Summarization
summarization, text, summary, document, summaries, sentences, sentence, documents, topic, information, evaluation
A2.5: 154
Sentiment Analysis
sentiment, opinion, opinions, reviews, analysis, positive, negative, classification, words, product
A2.7: 58
Biomedical NLP
information, biomedical, medical, database, data, literature, articles, databases, abstracts, names
A4.1: 390
Text Generation
text, generation, discourse, system, dialogue, structure, planning, model, plan, goals, knowledge
A6.1: 402
Parsing
grammar, models, parsing, grammars, model, parse, training, probabilistic, tree, trees
A6.3: 237
Semantic Role Labeling
semantic, role, labeling, task, roles, syntactic, predicate, argument, arguments, feature
and more at level 3 . . .
A2.2: 361
Information Retrieval
documents, retrieval, document, query, information, search, terms, queries, term, user, words
A2.4: 244
Question Answering
question, answer, questions, answering, answers, systems, system, retrieval, information, documents, trec
A2.6: 150
Textual Entailment
entailment, system, text, information, textual, task, extraction, template, systems, semantic, inference
A4.2: 345
Centering Theory
discourse, structure, resolution, text, anaphora, relations, coherence, pronouns, theory, pronoun, reference
A6.2: 299
Learning
learning, data, training, tasks, features, models, feature, methods, nlp, model
A6.4: 221
Treebanks
annotation, treebank, corpus, annotated, corpora, discourse, syntactic, language, grammar, linguistic

Figure 5: 3-level topic hierarchy built from the ACL Anthology.

[7] D. M. Blei, T. L. Griffiths, and M. I. Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *Journal of the ACM*, 57(2):1–30, Feb. 2010.

[8] J. Chang and D. Blei. Hierarchical relational models for document networks. *Annals of Applied Statistics*, 2009.

[9] 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.

[10] D. Cohn and T. Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity. In *Neural Information Processing Systems*, 2001.

[11] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. Scatter/Gather: a cluster-based approach to browsing large document collections. In *Proceedings of SIGIR*, 1992.

[12] S. Gerrish and D. Blei. A language-based approach to measuring scholarly impact. In *Proceedings of ICML*, 2010.

[13] T. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101:5228–5235, 2004.

[14] A. Gruber, M. Rosen-zvi, and Y. Weiss. Latent topic models for hypertext. In *Proceedings of UAI*, 2008.

[15] S. Gupta and C. Manning. Analyzing the dynamics of research by extracting key aspects of scientific papers. In *Proceedings of IJCNLP*, 2011.

[16] D. Hall, D. Jurafsky, and C. D. Manning. Studying the history of ideas using topic models. In *Proceedings of EMNLP*, 2008.

[17] T. H. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE Transactions on Knowledge Data Engineering*, 15(4):784–796, 2003.

[18] Q. He, J. Pei, D. Kifer, P. Mitra, and C. L. Giles. Context-aware citation recommendation. In *Proceedings of WWW*, pages 421–430, 2010.

[19] K. A. Heller and Z. Ghahramani. Bayesian hierarchical clustering. In *Proceedings of ICML*. ACM, 2005.

[20] Q. Ho, A. P. Parkih, L. Song, and E. P. Xing. Multiscale Community Blockmodel for Network Exploration. In *Proceedings of AISTATS*, 2011.

[21] P. Holland and S. Leinhardt. Local structure in social networks. *Sociological methodology*, 7:1–45, 1976.

[22] J. Huang, H. Sun, J. Han, H. Deng, Y. Sun, and Y. Liu. Shrink: a structural clustering algorithm for detecting hierarchical communities in networks. In *Proceedings of CIKM*, pages 219–228, 2010.

[23] A. Lancichinetti and S. Fortunato. Community detection algorithms: A comparative analysis. *Physical Review E*, 80(5):056117+, Nov. 2009.

[24] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, December 2004.

[25] Y. Liu, A. Niculescu-Mizil, and W. Gryc. Topic-link lda: joint models of topic and author community. In *Proceedings of ICML*, 2009.

[26] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008.

[27] A. McCallum, A. Corrada-Emmanuel, and X. Wang. Topic and role discovery in social networks. In *Proceedings of IJCAI*, 2005.

[28] Q. Mei, D. Cai, D. Zhang, and C. Zhai. Topic modeling with network regularization. In *Proceedings of WWW*, 2008.

[29] R. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen. Joint latent topic models for text and citations. In *Proceedings of KDD*, 2008.

[30] R. Nallapati, D. McFarland, and C. Manning. Topicflow model: Unsupervised learning of topic-specific influences of hyperlinked documents. In *Proceedings of AISTATS*, 2011.

[31] Y. Petinot, K. McKeown, and K. Thadani. A hierarchical model of web summaries. In *Proceedings of ACL*, 2011.

[32] X. Phan, L. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of WWW*, 2008.

[33] D. Radev, M. Joseph, B. Gibson, and P. Muthukrishnan. A bibliometric and network analysis of the field of computational linguistics. *Journal of the American Society for Information Science and Technology*, 2009.

[34] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

[35] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Proceedings of UAI*, pages 487–494, 2004.

[36] N. Sahoo, J. Callan, R. Krishnan, G. Duncan, and R. Padman. Incremental hierarchical clustering of text documents. In *Proceedings of CIKM*, 2006.

[37] J. Shi and J. Malik. Normalized cuts and image segmentation. *Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, Aug. 2000.

[38] Y. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

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

[40] F. Wu and D. S. Weld. Automatically refining the wikipedia infobox ontology. In *Proceedings of WWW*, 2008.

[41] Y. Zhao, G. Karypis, and U. Fayyad. Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery*, 10(2):141–168, Mar. 2005.