The Pennsylvania State University
The Graduate School

TOPIC MODELS FOR LINK PREDICTION IN DOCUMENT NETWORKS

A Dissertation in
Information Sciences and Technology
by
Saurabh Kataria

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2012

The thesis of Saurabh Kataria was reviewed and approved* by the following:

Prasenjit Mitra
Chair of Committee
Associate Professor of Information Sciences and Technology
Dissertation Advisor

C. Lee Giles
Professor of Information Sciences and Technology

Dongwon Lee
Associate Professor of Information Sciences and Technology

Daniel Kifer
Assistant Professor of Computer Science and Engineering

Mary Beth Rosson
Director of Graduate Programs
College of Information Sciences and Technology

*Signatures are on file in the Graduate School.

Abstract

Recent explosive growth of interconnected document collections such as citation networks, network of web pages, content generated by crowd-sourcing in collaborative environments, etc., has posed several challenging problems for data mining and machine learning community. One central problem in the domain of document networks is that of *link prediction* among any two documents or document centric entities, such as authors, based upon already present links in a given network. The problem of link prediction in document networks is a fundamental problem. Several applications, such as recovering missing link among entities in a given network of documents, citation recommendation to research professionals, collaborator recommendations to authors, discovering influential authors or bloggers in research articles or web-logs respectively, studying ideas and opinion propagation in evolving collection of research documents or news media, disambiguating references of people mentioned in news articles, etc. can be cast as a particular flavour of link prediction problem to be solved. This thesis studies following three link prediction based research problems in document networks: (i) *Who influences other's actions in a collaborative research environment?*, (ii)*which documents get cited by a document that joins a citation network?*, and (iii) *which is the correct entity for an entity mention in free text?*.

Among various computation methods to solve domain specific link prediction problem, statistical machine learning based techniques are an increasingly acceptable method due to their capability of modeling complex relationships among documents and document centric entities and dedicated efforts from research community to make the resulting intractable inference computationally scalable. This thesis proposes two types of statistical models: (1) models that mimic the generation process of document networks e.g. citation network of scientific documents, interconnected blog articles, web pages, etc.; (2) models that are capable of incorporating a specific task oriented features as supervision. The proposed statistical models are an extension of Latent Dirichlet Allocation, also known as *topic models*. In this work, I show how topic models can be adapted for the above mentioned link prediction problems. The proposed techniques perform superior to previous approaches for these link prediction problems.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to extend my heartfelt thanks to my adviser Dr. Prasenjit Mitra for his enormous support and guidance throughout my graduate student life. He inculcated in me the virtues of patience, perseverance, and scientific rigour which are essential to a researcher's life. I am also deeply indebted to Dr. C. Lee Giles for mentoring me throughout my graduate research duration at Penn State. I am also very thankful to Dr. Dongwon Lee, and Dr. Daniel Kifer for agreeing to serve on my thesis committee. Their careful comments and feedback have enormously helped me in my research and, especially, this thesis. I am very thankful to all the faculty of the College of Information Sciences and Technology for being excellent teachers and mentors during my graduate life.

I am thankful to DTRA for supporting my research at Penn State. I am also very thankful to Dr. Frank Ritter for numerous discussion and intellectual comments throughout my research assistant duration. I also deeply value the industry exposure at Yahoo under the guidance of Dr. Prithviraj Sen and Dr. Rajeev Rastogi. Their support and intellectual contribution to this thesis is indispensable. I am also thankful to Dr. Luca Marchesotti and Dr. Florent Perronnin at Xerox Research Center Europe for the great research experience, discussion and motivation which proved critical for my overall research achievements. Research discussions with my collaborators and colleagues Dr. Ritendra Dutta, Dr. Jian Huang, Pradeep Teregowda, Sujatha Das, Sumit Bhatia, Puck Treeratpituk, William Brouwer, Cornelia Caragea and Xiao Zhang.

Dedication

To My Mother

Chapter 1

Introduction

1.1 Learning to Predict Links in Document Networks

With the advent of world-wide-web, various collections of inter-connected documents, such as web-logs, hypertext documents, encyclopedic articles, scientific documents, etc., are easy to obtain. These collection of documents can be described as relational network of entities where documents and their attributes, such as authors, words and categories, act as nodes and connections among them act as links. These relational datasets have been widely utilized for analysing various prediction tasks such as author collaboration in collaboration networks of scientific publications, citation prediction in citation networks, time sensitive prediction of citation network evolution, etc. One can cast these various prediction tasks into a framework of a fundamental link prediction problem in document networks. *Formally, link prediction can be defined as the estimation of the likelihood of a link between any two entities.*

Fundamentally, there are two types of algorithmic approaches to solve link prediction problems: (1) Topological structure analysis of graph associated with entities [33, 35], and (2) statistical machine learning techniques that learn from existing links to infer parameters that govern or explain link formulations in document network [12, 22, 42]. Analysis of graph structure for link prediction is algorithmically easy and tractable; however, often leads to inferior results compared to statistical machine learning approaches [33]. Although computationally intractable in nature, statistical machine learning approaches have proven to be effective and, also, provide an extensible framework which is both theoretically and practically appealing. This dissertation provides an in depth analysis of a particular class of statistical machine learning methods, known as *topic models*, from a link prediction perspective. Additionally, I provides analysis of three specific variants of link prediction problem: (1) citation prediction (2) influential author detection, and (3) entity disambiguation in crowed-sourced knowledge bases.

Learning in document network with machine learning approaches is a challenging task. Traditionally, machine learning for network (or graph) oriented datasets assumes a random sample of homogeneous objects with a singly type of relations to learn for any given task [18]. However, a given sample of document network pose two specific challenges contrary to this assumption: (1) multi-type relations among entities present in documents, e.g., document-word relation, document-author relation, author-author relation, etc., and (2) relations with attributes, e.g. anchor text with hyper-links, citation context for citations, etc. Generative models for documents, in the form of topic models, have emerged as a natural class of models that can accommodate features in their generative process to address both of these challenges.

1.2 Topic Models for Document Networks

Latent Dirichlet allocation (LDA) [7] was among the first topic model proposed to understand the generative process of document collections. LDA defines *topics* as a multinomial distribution over vocabulary of words in a given corpus, where the weight of each word in a topic signifies its importance in the topic. The generative process of LDA assumes that there lies unobserved (or *latent*) concepts in the corpus that imposes an inherent clustering on documents and each document probabilistically belongs to a concept. An arguably similar class of latent class models, named latent semantic analysis (LSA) [15], can be seen as predecessors of topic models as follows. LSA applies singular value decomposition (SVD) on the occurrence matrices obtained from document-word or document-citation relations and construct a lower rank matrix representation of documents, words and citations respectively. However, this representation can introduce negative concept associations which can be problematic for interpreting concept representations. Probabilistic latent semantic analysis (PLSA) [26] overcomes this drawback by formulating a probabilistic generative model that assigns probabilities to document concept association and document word association, however, PLSA is prone to over-fitting [7]. LDA overcomes the over-fitting problem with a Bayesian version of PLSA where each document has a prior knowledge about its concept proportion/representation that can be pre-specified of learnt from the corpus itself. An in-depth comparisons of these techniques follow in incoming chapters.

Topic models provide an extensible framework to incorporate attributes of documents in their generative processes. Authors and citation among documents are among the most common attributes of documents and are integral part of relational document networks dealt with in this dissertation. Author Topic Model (ATM) [47] extends topic models to let authors generate the content of documents whereas Link-LDA [42] generates links in documents depending upon document's probabilistic association with topics. However, recent advancement of web technologies have introduced increasingly sophisticated means to collaboratively generate collection of documents, e.g. crowd sourced media such as Wikipedia. These document collections are endowed with extra features such as document categories, entity cataloguing, etc., and these features can potentially help in improving link prediction quality. However, incorporating these features into topic models is not straightforward and a challenging problem. The next section introduces specific topic models for two types of document networks: (1) citation networks of scientific and web documents (2) Crowd-sourced media, i.e. Wikipedia. The main focus of the below mentioned topic-model based approaches are two-fold: (1) modeling a generative process that is informative about the features in the dataset, and (2) application of these approaches to link prediction in the dataset.

1.3 Application of Topic Models to Variants of Link Prediction problems in Document Networks

1.3.1 Citation Prediction with Context Sensitive Topic Models

A document network where one document cites another document, e.g. hyperlinked web-pages, citation network of scholarly articles, has been analysed for mining patterns present in the links structure of the graph. One of the corresponding problem is to detect latent structures like *topics*, present in a given corpus and apply these topics for citation prediction [42, 11]. These latent structures, inherently, tend to seek a clustering of *semantically* similar entities present in the collection. Probabilistic approaches such as LDA [7] and PLSA [26] model the co-occurrence patterns present in text and identify a probabilistic membership of the words and the documents in a lower dimensional space. The link structure contains meaningful information about entities, e.g., documents, authors etc.; this information has been successfully utilized in web search [9]. However, the content based *topic* models [7, 26, 6] completely ignore this information. Recently, Dietz, et al. [16], Nallapati, et al. [42] and Cheng, et al. [11] have shown that modeling the citation and the content together not only helps to better understand the latent structure present in the data, but also helps to understand certain aspects of a linked corpus such as novelty detection, influence prorogation, citation prediction etc. Although current approaches look at what other documents influenced the content of a document, they overlook how those documents influenced the content of this document. This thesis puts emphasis upon modeling the context in which a citation appear in a document.

In document networks, the context in which a citation appears provides extra information about the cited document. However, associating terms in the context to the cited document remains an open problem. We propose a novel document generation approach that statistically incorporates the context in which a document links to another document. I quantitatively show that the proposed generation scheme explains the linking phenomenon better than previous approaches. The context information along with the actual content of the document provides significant improvements over the previous approaches for various real world evaluation tasks such as link prediction and log-likelihood estimation on unseen content. As I demonstrate in chapter 2, the proposed method is more scalable to large collection of documents compared to the previous approaches.

1.3.2 Author Influence Detection with Context Sensitive Topic Models

Modeling the interest of authors given a corpus of documents has been studied to answer important queries about the authors such as who produces similar work [47], who belongs to the same research community [34], etc. These queries form the basis of several information retrieval and machine learning tasks such as expert search, community detection, etc. Recently, several generative models of document corpus have begun exploring latent structures such as topics, present in the documents. LDA [7] and PLSA [26] model the co-occurrence patterns present in text and identify a probabilistic membership of words and documents in a lower dimensional space. Rosen-Zvi, et al., [47] extended these approaches to answer queries related to interests of authors. However, these approaches are unable to answer another fundamental question about the attribution of interests: who influences the generation of new content in a particular topic of interest? In chapter 3, I propose generative models that take the linkage between authors of citing and cited documents into consideration and explore various qualitative and quantitative aspects of this question.

1.3.3 Entity Disambiguation with Hierarchical Topic Models

Disambiguating entity references by annotating them with unique ids from a catalog is a critical step in the enrichment of unstructured content. In this paper, we show that topic models, such as *Latent Dirichlet Allocation* (LDA) and its hierarchical variants, form a natural class of models for learning accurate entity disambiguation models from crowd-sourced knowledge bases such as Wikipedia. Our main contribution is a semi-supervised hierarchical model called *Wikipedia-based Pachinko Allocation Model* (WPAM) that exploits: (1) All words in the Wikipedia corpus to learn word-entity associations (while existing approaches only use words in a small fixed window around annotated entity references in Wikipedia pages), (2) Wikipedia annotations to appropriately bias the assignment of entity labels to annotated (and co-occurring unannotated) words during model learning, and (3) Wikipedia's category hierarchy to capture co-occurrence patterns among entities. We propose a new sampling algorithm to speed up model learning when topics are organized in a hierarchy, and a scheme for pruning spurious nodes from Wikipedia's crowd-sourced category hierarchy. Finally, in experiments with multiple real-life datasets, we show that WPAM outperforms state-of-the-art baselines by as much as 22% in terms of disambiguation accuracy.

1.4 The Hypotheses and the Organization of the Dissertation

In this section, I specify the hypotheses that I verify in the dissertation and describe the organization of the dissertation.

1.4.1 The Hypotheses

In this dissertation, I verify following hypotheses:

H.1. Latent Dirichlet Allocation based models, i.e. *topic models* provides a natural mechanism to explain, with a statistical process, the generation of links among documents in document networks.

H.2. Topic models can be adapted to predict citations in document network of scientific articles as well as links among hypertext documents.

H.3 Topic models can be adapted to identify research area specific influential authors in a given corpus of scientific documents.

H.4. Content surrounding mention of links, i.e., *citation context*, in document networks of scientific and hypertext documents provide important citation usage information that can improve prediction of links between recently introduced articles to the network.

H.5. Topic models can learn from citation context explicitly and the resultant models can have an improved link prediction capability.

H.6. Crowd-sourced knowledge-bases, such as Wikipedia, provide a weak form of supervision for predicting links between entity pages and their references within the knowledge-base. Hierarchical extensions of topic models provide a natural framework to incorporate crowd source induced features as weak form of supervision to learn effective entity-entity associations which, in turn, provides improvements in predicting above mentioned links.

1.4.2 The Organization of the Dissertation

The rest of the dissertation is organised as follows. In Chapter 2, I describe extensions of topic modes for citation network of scientific documents as well as hypertext documents. I also describe how citation context of cited documents can be incorporated into topic models' generation process to achieve improvements in predicting citations among documents. In chapter 3, I introduce topic models to determine topic specific influential authors. I also describe application of topic models for predicting links among citing and cited authors. In chapter 4, I describe hierarchical extensions of topic models that learn from annotations and categories present in Wikipedia. In this chapter, I also apply these extension to predicting links among entity references present in textual documents in Wikipedia and news media and the corresponding entity profiles in Wikipedia. I conclude the dissertaion in chapter 5 and lay ground for future research direction that can be pursued based upon the work in this dissertation by discussing some research questions that are yet unanswered.

Chapter 2

Context Sensitive Topic Models for Citation Networks

In a citation network of documents such as network of scholarly articles in scientific digital libraries, web-logs, etc., words surrounding a citation mention in a document, i.e., *citation context*, provides extra information for the usage of the citation in the citing document. This chapter present context sensitive models for citation network which learns from citation contexts explicitly. The models build upon Latent Dirichlet Allocation (LDA) adding a key property in the modeling approach: topics distribution in a citation context affects both the word and citation generation. I provide results on multiple real-world datasets providing evidence that the citation context helps in improving model quality compared to the context insensitive topic models in (1) explaining the unseen content, and (2) recovering missing links in the citation network.

2.1 Introduction

Large collections of interlinked documents such as the World Wide Web, digital libraries of scientific literature, weblogs have given rise to several challenging problems, e.g., detecting latent structures like *topics*, present in a given corpus. These latent structures, inherently, tend to seek a clustering of *semantically* similar entities present in the collection. Probabilistic approaches such as LDA [7] and PLSA [26] model the co-occurrence patterns present in text and identify a probabilistic membership of the words and the documents in a lower dimensional space. These *topic models* have been used to explore various aspects of the document collection, such as correlation among different topics [6], the evolution of concepts [19], etc.

In a linked corpus, the link structure contains meaningful information about entities, e.g., documents, authors etc.; this information has been successfully utilized in web search [9]. However, the content based *topic* models [7, 26, 6] completely ignore this information. Recently, Dietz, et al. influence, Nallapati, et al. plsa-lda and Cheng, et al. relation-topic have shown that modeling the citation and the content together not only helps to better understand the latent structure present in the data, but also helps to understand certain aspects of a linked corpus such as novelty detection, influence propagation, citation prediction etc. Although current approaches analyze the citation phenomenon to identify which documents influenced the content of a particular document, however, these approaches overlook how the cited document influences the content of the citing document. In other words, the process of incorporation of the citation information ignores the context in which that citation appears in the document.

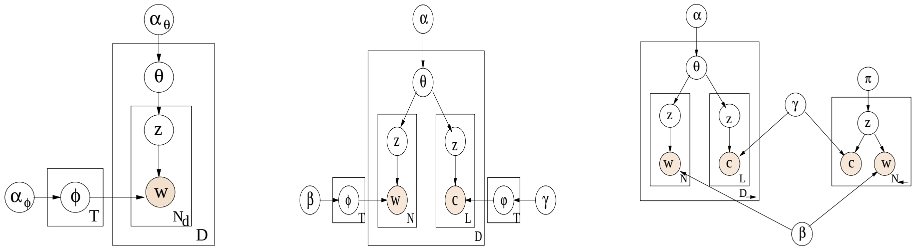
In this chapter, for the citation network, we propose a generative model of the content and citations in a document that incorporate context information while modeling content and citations jointly. We hypothesize that context information can help in improving the topic identification for words and, in turn, documents. We assume that the author of the citing document chooses a topic first, and then while writing the text of the document chooses the citation context to describe a citation. The citation context does not necessarily portray the entire content of the cited document, but, provides a description from the author's perspective in relation to the citing document's topic. The citation context contains words related to the chosen topic and these words can help identify the major topics in the cited document. On the other hand, the topic of the context words can be identified using the major topics of the cited document as well. On the world-wide-web, anchor text and words surrounding the anchor text represent the context of the hyper-linked document.

The organisation of the rest of the chapter is as follows. In section 2, we discuss the relevant related work for topic modeling based document network analysis.

In section 3, we describe our context sensitive approaches for document network modeling. In the subsequent section 4, we formulate the inference algorithms using Gibbs sampling for our modeling approaches listed in the previous section. We provide quantitative evaluation of our models on three real-world datasets with a few anecdotal evidences in section 5 and summarize in section 6.

2.1.1 Related Work

One of the earliest attempts at modeling text and citation together in a linked corpus was posed as an extension of probabilistic latent semantic analysis [26] and its analogue for citations named *missing link model* [13]. The Bayesian version of the missing link model was proposed as the *mixed membership model* [50] and *link-LDA* [42] with the Dirichlet distribution acting as a conjugate distribution to the multinomial distribution for the citation and word generation process in the missing link model. The generation process of links in link-LDA is similar to the generation process of words in LDA (shown in fig 1(a)) as in the same document-specific topic distribution, i.e. θ in fig 1(b), is used to generate words and links. The corresponding plate-diagram for link-LDA is shown in fig 2.1.2.



2.1.1 LDA

2.1.2 link-LDA

2.1.3 link-PLSA-LDA

Figure 2.1. Bayesian Network for (a) Latent Dirichlet Allocation, (b) link-LDA, (c) link-PLSA-LDA

Although missing link model and its Bayesian extensions are quantitatively successful in clustering the citations and words, the underlying generative process is too simplistic to explain various phenomenon related to linked structure of the corpus, e.g. influence propagation, associating words and links, etc. Recently, Nallapati, et al., plsa-lda proposed a more rigorous modeling of the content and links together, link-PLSA-LDA (depicted in Figure 2.1.3), where the data is partitioned into two subsets of cited and citing documents[1] and both the subsets are modeled differently with the same global parameters due to scalability concerns. The cited set of documents is modeled using PLSA and the citing set of documents is modeled using the *link-LDA* model. The underlying assumption behind the link-PLSA-LDA model is that there exist both a global topic-citations distribution according to which the citing document chooses its citations and a global word-topic distribution from which the words are generated. This bipartite representation approach was first proposed by Dietz, et al., influence to impose an explicit relation between the cited and the citing text so that the two together can augment the information provided by the citation links, while modeling a linked corpus.

Grueber, et al., htm and Guo, et al., nec-bern proposed a generative model for linked documents with a two step statistical process where first, the presence or absence of a link is decided and second, an actual link is "generated". Grueber, et al., htm uses a multinomial distribution over the links present in the document with an additional *e*mpty link whereas Guo, et al., nec-bern uses a Bernoulli random variable to decide whether to link to an external document or not. In contrast to these methods, our context sensitive approach takes a direct approach where we make an explicit use of the citation location in the citing document and assumes a definite influence of the citation over the words in a window around the citation.

2.2 Utilizing Context in Modeling Approaches

2.2.1 Context Sensitive Topic Models for Citation Network

Notations: Let V, D, N_d and C denote the size of the word vocabulary, the total number of documents, number of words in document d and the number of cited documents respectively. Let K denote the number of topics and suppose there exist a $K \times V$ topic-word distribution matrix ϕ that indexes a probabilistic distribution over words given the topic and a $K \times C$ topic-citation distribution matrix φ that indexes the probability of a document being cited given a topic. At the document level, we assume that the author chooses to mix the topics with

[1] duplication is done for those documents that are both citing and cited in the corpus

θ_d as the mixing proportion for document d. We treat the context information explicitly as follows. First, we define a citation context for a cited document as a bag of words that contains a certain number of words appearing before and after the citation's mention in the citing document. Table 2.1 shows such an example of a citation context. In case a cited document is mentioned multiple times, we assimilate all the corresponding context words.

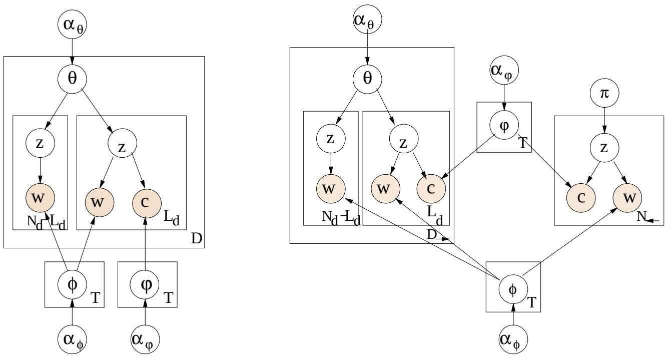
Citing paper: A Statistical Learning Model of Text Classification for Support Vector Machines → Cited paper: Latent Semantic Indexing, A Probabilistic Analysis Abstract of the citing paper: ....“Unlike conventional approaches to learning text classifiers, which rely primarily on empirical evidence, this model explains why and when SVMs perform well for text classification. In particular, it addresses the following questions: Why can support vector machines handle the large feature spaces in text classification effectively? ” Citation & its Context: ...“ Papadimitriou et. al, is most similar in spirit to the approach presented here[16]. They show that latent semantic indexing leads to a suitable lowdimensional representation ”...

Table 2.1. An example of a citation context

Following is the formal introduction to our context sensitive topic models for document network. We take a stepwise approach and first extend the basic Link-LDA [50] model and its extension, Link-PLSA-LDA, to model the citation context explicitly. Later, we propose two statistical approaches that model the influence of a cited document over the generation of the context in the citing document.

2.2.1.1 cite-LDA Model

The basic underlying assumption while incorporating the above mentioned context is that, given a topic, the choice of words in the context surrounding a cited document mention and the cited documents are independent. Suppose the author has a topic in mind (i.e., a distribution over words), and she comes across multiple documents that she can cite related to this topic. Now, if she has sufficiently narrowed down the topic, then the choice of words to describe the cited document



2.2.1 2.2.2

Figure 2.2. Bayesian Network for (a) cite-LDA, (b) cite-PLSA-LDA

depends only upon the topic instead of the document that she would cite. Based upon this simplifying statistical assumption, next we describe the model for a linked corpus. cite-LDA is a generative model with the generation process described in Algorithm 1 and the corresponding plate diagram is given in Figure 2.2.1.

Formally, given the model parameters α_θ, α_ϕ and α_φ, the joint distribution of the topic variables **z**, the document **w** and the citation context **c** can be written as:

$$
\begin{aligned}
& p(\mathbf{z},\mathbf{w},\mathbf{c}|\alpha_\theta,\alpha_\phi,\alpha_\varphi) \\
&= \int \prod_{d=1}^{D} p(\theta_d|\alpha_\theta) \prod_{n=1}^{N_d-L_d} p(z_n|\theta)p(w_n|z_n,\beta) \prod_{n=1}^{C_d} p(z_n|\theta_d)p(w_n,c_n|z_n,\alpha_\phi,\alpha_\varphi)d\theta_d \\
&= \int \prod_{d=1}^{D} p(\theta_d|\alpha_\theta) \prod_{n=1}^{N_d-L_d} p(z_n|\theta_d)p(w_n|z_n,\alpha_\phi) \prod_{n=1}^{L_d} p(z_n|\theta_d)p(w_n|z_n,\alpha_\phi)p(c_n|z_n,\alpha_\varphi)d\theta_d \qquad (2.1) \\
&= \int \prod_{d=1}^{D} p(\theta_d|\alpha_\theta) \prod_{n=1}^{N_d} p(z_n|\theta_d)p(w_n|z_n,\alpha_\phi) \prod_{n=1}^{L_d} p(c_n|z_n,\alpha_\varphi)d\theta_d \qquad (2.2)
\end{aligned}
$$

L_d is the total length of all citations contexts in the document d. The independence assumption allows us to factorize the joint distribution separately for the words in the context and the citations. Intuitively, Eq. 2.1 implies that the author first picks the words from the topic and then citations from the topic or vice versa. The product $p(z_n|\theta_d).p(w_n|z_n)$ acts as the mixing proportions for the citation generation probability over the entire citation context of the corresponding citation. Therefore, one can expect that this explicit relation between citation generation probability and the word generation probability will lead to a better association of words and citations with documents than without utilizing the citation context

Algorithm 1: The cite-LDA generation process

```
for each document $d \in (1, 2, .., D)$: do
    $\theta_d \sim Dir(.|\alpha_\theta)$.
    for each word in $w_n \in d$ that appears outside any citation context: do
        Choose a topic $z_n \sim Mult(.|\theta_d)$.
        Choose $w_n$ from word-topic distribution, i.e. $w_n \sim Mult(.|z_n, \phi_{z_n})$.
    end for
    for each word in $w_n \in d$ that appears inside of any citation context: do
        Choose a topic $z_n \sim Mult(.|\theta_d)$.
        Choose $w_n$ from topic-word distribution, i.e. $w_n \sim Mult(.|z_n, \phi_{z_n})$.
        Choose a document $c_n$ to link from topic-citation distribution i.e.
        $c_n \sim Mult(.|z_n, \varphi_{z_n})$.
    end for
end for
```

explicitly.

2.2.2 Context aware approach for modeling the influence of cited documents

In the generation processes described so far, once a topic for a word is identified, we assume that the citation that is associated with the context only depends upon the topic. However, this assumption can be restrictive in cases where the word is tightly associated with the citation. For example, words such as “LDA” and “PLSA” are associated with Blei, et al. lda and Hoffman plsa respectively and this association is irrespective of the topic with which these documents are cited. The accommodation of these associations requires that we relax the independence assumption and model the correlations among citations and words directly, which is described through following additional modeling schemes.

2.2.2.1 cite-PLSA-LDA Model

Similar to the link-PLSA-LDA model [42], this model views the data as two separate sets of citing and cited documents as explained previously. The cite-PLSA-LDA model assumes that the words and citations occurring in the citing documents generate from a smoothed (with a Dirichlet prior) topic-word and topic-citation multinomial distributions respectively. We model the generation of citation context by assuming the conditional independence of a word and a citation given the word. However, for cited documents, we assume that an empirical distribution of the topics is to be fitted that explains the generation of documents and words in the cited set. Therefore, LDA [7] and PLSA [26] become the natural choice of frameworks for modeling the citing and the cited set respectively. The generation process assumed by the cite-PLSA-LDA model is described in Algorithm 2 and the corresponding graphical depiction is given in Figure 2.2.2.

Algorithm 2: The Cite-PLSA-LDA generation process

```
for each word $w_n$ in cited set of documents: do
    Choose $z_i \sim Mult(.|\pi)$.
    Choose $w_n \sim Mult(.|z_i, \phi_{z_i})$.
    Sample $d_i \in 1, ...D_{\leftarrow} \sim Mult(.|z_i, \varphi_{z_i})$.
end for
for each citing document $d \in (1, 2, .., D_{\rightarrow})$: do
    $\theta_d \sim Dir(.|\alpha_\theta)$.
    for each word in $w_n \in d$ that appears outside any citation context: do
        Choose $z_n \sim Mult(.|\theta_d)$.
        Choose $w_n$ from word-topic distribution, i.e. $w_n \sim Mult(.|z_n, \phi_{z_n})$.
    end for
    for each word in $w_n \in d$ that appears inside of any citation context: do
        Choose a topic $z_n \sim Mult(.|\theta_d)$.
        Choose $w_n$ from topic-word distribution, i.e. $w_n \sim Mult(.|z_n, \phi_{z_n})$.
        Choose a document $c_n$ to link from topic-citation distribution i.e.
        $c_n \sim Mult(.|z_n, \varphi_{z_n})$.
    end for
end for
```

Formally, given the model parameters α_θ, α_ϕ, α_φ and π (the topic mixture for cited documents), the complete data likelihood can be obtained by marginalizing the joint distribution of a topic mixture θ for citing documents, the topic variable $\mathbf{z}$, the document $\mathbf{w}$ and the citation context $\mathbf{c}$ and can be written as:

$$\begin{aligned} p(\mathbf{w,c}|\alpha_\theta, \alpha_\phi, \alpha_\varphi, \pi) = \prod_{n=1}^{N_{\leftarrow}} (\sum_k p(z|\pi) p(d_n|z) p(w_n|z)) \\ \times \prod_{d}^{D\rightarrow} \int p(\theta_d|\alpha_\theta) (\prod_{n=1}^{N_d} \sum_{z=1}^{K} (p(z_n|\theta_d) p(w_n|z_n, \alpha_\phi)) \times \prod_{n=1}^{C} \sum_{z=1}^{K} (p(c_n|z_n, \alpha_\varphi))) d\theta_d \end{aligned} \tag{2.3}$$

Here, d_n indicates the document to which word w_n belongs to.

2.2.2.2 The Switch-Cite-LDA Model

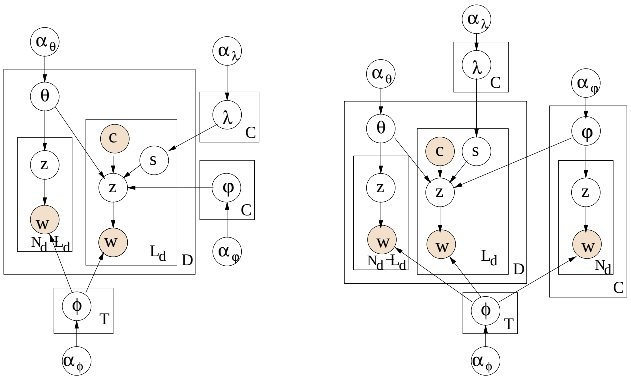
To handle the associations among citations and the words in their contexts, we let the citation or the citing document choose to “generate” the topic of a word in its context. Before generating any word in the citation context, the statistical process tosses a biased coin to decide whether the citation or the citing document shall "generate" the word. Each citation indexes a distribution over topics, φ, [2] which is used to generate topics in the context of the citation. The generative process for switch-cite-LDA is described in Algorithm 3 and its corresponding plate diagram is shown in Fig. 2.3.1.

Algorithm 3: The switch-cite-LDA generation process

```
for each topic $t \in (1,2,..,T)$: do
    Sample $\phi_t \sim Dir(.|\alpha_\phi)$
end for
for each cited article $c \in (1,2,..,C)$: do
    Sample $\lambda_c \sim Beta(.|\alpha_{\lambda_\theta}, \alpha_{\lambda_\varphi})$
    Sample $\varphi_c \sim Dir(.|\alpha_\varphi)$
end for
for each document $d \in (1,2,..,D)$: do
    $\theta_d \sim Dir(.|\alpha_\theta)$.
    for each word in $w_n \in d$ that appears outside any citation context: do
        Choose a topic $z_n \sim Mult(.|\theta_d)$.
        Choose $w_n$ from word-topic distribution, i.e. $w_n \sim Mult(.|z_n, \phi_{z_n})$.
    end for
    for each word in $w_n \in d$ that appears inside of any citation context: do
        Sample $s \sim Bern(.|\lambda_{c_n})$
        if $(s == 0)$: then
            Choose a topic $z_n \sim Mult(.|\theta_c)$.
        else
            Choose a topic $z_n \sim Mult(.|\varphi_c)$.
        end if
        Choose $w_n$ from topic-word distribution, i.e. $w_n \sim Mult(.|z_n, \phi_{z_n})$.
    end for
end for
```

In addition to the document-topic mixture parameter α_θ, topic-word distribution parameter α_ϕ and citation-topic mixture parameter α_φ, we assume that the parameter α_λ controls the biasing of the coin corresponding to each citation in its favor. Intuitively, α_λ governs the generation of those words from α_φ that tend to appear with the citation very frequently. Later, we demonstrate the qualitative importance of this switching parameter.

[2] the parameter variable φ is different from the one in previous models. However, to maintain the consistency among the notations, we choose to use the same notation



2.3.1 2.3.2

Figure 2.3. Bayesian Network for (a) Switch-Cite-LDA, (b) Bipartite-Switch-Cite-LDA

The complete log-likelihood of the model can be written as follows.

$$p(\mathbf{w},\mathbf{z},\mathbf{c},\mathbf{s}|\alpha_\theta, \alpha_\lambda, \alpha_\phi, \alpha_\varphi) = \int \prod_{d=1}^{D} \Big(p(\theta_d|\alpha_\theta) \prod_{n=1}^{N_d - L_d} p(w_n|z_n, \phi_{z_n}, \alpha_\phi) p(z_n|\theta_d) \times$$

$$\prod_{n=1}^{L_d} p(w_n|z_n, \phi_{z_n}, \alpha_\phi) p(z_n|\theta_d, c_n, s_n, \varphi_c, \alpha_\varphi) p(s_n|\lambda_{c_n}, \alpha_\lambda) \Big) d\theta_d \qquad (2.4)$$

2.2.2.3 The Bipartite-Switch-Cite-LDA Model

The model proposed in this subsection extends the Switch-Cite-LDA model by modeling the generation of the content of the cited documents as well as the citing documents. The parameter φ for a cited document c in Switch-Cite-LDA model infers its distribution over topics from the words in the citation context of c only whereas one can hope that a better estimate obtained by additional inference from the content of c. However, as any cited document can also belong to the set of citing documents, therefore, we take a bipartite view over the corpus where we divide the documents into the cited and citing set of documents. A document can belong to the cited set as well as the citing set depending upon whether the document has been cited at least once or not. The generation process of the Bipartite-Switch-Cite-LDA generates the words in the citing set of documents as well as in the cited set of documents. The bipartite view for the generation process has also been adopted by Dietz, et al. influence, however, their model has an additional statistical component in the generation process that chooses to generate the cited document for each word in the citing document. We avoid this step by associating the words in the citation context with the citation itself.

The generative process for the Bipartite-Switch-Cite-LDA is described in Algorithm 4 and its corresponding plate diagram is shown in Fig. 2.3.2.

Algorithm 4: The Bipartite-Switch-Cite-LDA model generation process

```
for each topic $t \in (1, 2, .., T)$: do
    Sample $\phi_t \sim Dir(.|\alpha_\phi)$
end for
for each cited article $c \in (1, 2, .., C)$: do
    Sample $\lambda_c \sim Beta(.|\alpha_{\lambda_\theta}, \alpha_{\lambda_\varphi})$
    Sample $\varphi_c \sim Dir(.|\alpha_\varphi)$
    for each word in $w$ in cited article $c$: do
        Choose a topic $z_n \sim Mult(.|\varphi_c)$.
        Choose $w_n$ from word-topic distribution, i.e. $w_n \sim Mult(.|z_n, \phi_{z_n})$.
    end for
end for
for each document $d \in (1, 2, .., D)$: do
    $\theta_d \sim Dir(.|\alpha_\theta)$.
    for each word in $w_n \in d$ that appears outside any citation context: do
        Choose a topic $z_n \sim Mult(.|\theta_d)$.
        Choose $w_n$ from word-topic distribution, i.e. $w_n \sim Mult(.|z_n, \phi_{z_n})$.
    end for
    for each word in $w_n \in d$ that appears inside of any citation context: do
        Sample $s \sim Bern(.|\lambda_{c_n})$
        if $(s == 0)$: then
            Choose a topic $z_n \sim Mult(.|\theta_c)$.
        else
            Choose a topic $z_n \sim Mult(.|\varphi_c)$.
        end if
        Choose $w_n$ from topic-word distribution, i.e. $w_n \sim Mult(.|z_n, \phi_{z_n})$.
    end for
end for
```

The complete log-likelihood of the model can be written as follows.

$$
\begin{aligned}
& p(\mathbf{w},\mathbf{z},\mathbf{c},\mathbf{s}|\alpha_\theta, \alpha_\lambda, \alpha_\phi, \alpha_\varphi) = \int\int \prod_{d=1}^{D} \Big(p(\theta_d|\alpha_\theta) \prod_{n=1}^{N_d - L_d} p(w_n|z_n, \phi_{z_n}) p(z_n|\theta_d) \times \\
& \prod_{n=1}^{L_d} p(w_n|z_n, \phi_{z_n}, \alpha_\phi) p(z_n|\theta_d, c_n, s_n, \varphi_c) p(s_n|\lambda_{c_n}, \alpha_\lambda) \Big) \times \\
& \prod_{c=1}^{C} p(\varphi_c|\alpha_\varphi) \Big(\prod_{n=1}^{C_d} p(w_n|z_n, \phi_{z_n}, \alpha_\phi) p(z_n|\varphi_c) \Big) d\varphi_c \; d\theta_d
\end{aligned}
\tag{2.5}
$$

2.2.3 Dynamic Selection of Length of Context Window

Since the cite-* models imposes independence assumption in the context window surrounding the citation mention, it becomes important to identify the context that refers to the cited article. Previous work on context utilization in topic models, either assumes a fixed window of 10 words radius surrounding the citation mention [28] or the whole document as the context for any citation mention [53]. However, the amount of relevant context in the vicinity of the citation anchor depends upon various factors such as the strength of the influence of cited article over the citing article, the location of the citation mention in the citing article, etc. Therefore, I propose to identify a dynamic window surrounding the citation anchor with the following method.

Let $\overleftarrow{d}$ represent the cited document for a given citation anchor c_i^d, where i ranges over all citation mentions in the citing document d. Let $S(c_i^d)$ (or simply S_i) represent the bag of words in the citation context surrounding c_i^d. The objective function that I choose to maximize is $f(\overleftarrow{d}|S_i)$ which is defined as:

$$f(\overleftarrow{d}|S_i) = \sigma(Z_{\overleftarrow{d}}.Z_{S_i}) \tag{2.6}$$

Here, Z_p is the topic vector defined as $\frac{1}{N_p}\sum_n z_{p,n}$ where n ranges over all the tokens in the bag p and N_p denotes the cardinality of p. σ represents the sigmoid function and $'.'$ represents the dot product between two vectors. Intuitively, $f(\overleftarrow{d}|S_i)$ represents the topical similarity between cited document and its corresponding context.

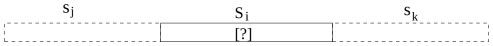


Figure 2.4. An illustrative citation context window

Next I describe our dynamic context selection procedure. I allow our window to grow over sentences beginning with the sentence that has the citation mention, although the method proposed is general enough to be applicable to any building block such as words or paragraphs. I choose to begin with the sentence that contains the citation mention as the sentence carries most of the information about the cited document. I let S_i denote the current context window and s_j and s_k are the next left and right candidates to either include in the window or to let the growth stop in either direction. I update the window as defined below and continue to grow in the direction which maximizes the objective function 3.3.

$$S_i = max\{f(\overleftarrow{d}|S_i), f(\overleftarrow{d}|\{S_i, s_j\}), f(\overleftarrow{d}|\{S_i, s_k\}), f(\overleftarrow{d}|\{S_i, s_j, s_k\})\} \tag{2.7}$$

2.3 Inference using Gibbs Sampling

The computation of the posterior distribution of the hidden variables θ and $\mathbf{z}$ is intractable for both the cite-LDA and cite-PLSA-LDA model because of the pairwise coupling between θ, β and θ, γ. Therefore, we need to utilize approximate methods e.g. variational methods [42] or sampling techniques [21] for inference. Considering that the Markov Chain Monte Carlo sampling methods such as Gibbs sampling come with a theoretical guarantee of converging to the actual posterior distribution and the recent advances that make its fast computation feasible over a large corpus [45], we utilize Gibbs sampling as a tool to approximate the posterior distribution for both the models.

2.3.1 Inference Estimation for cite-LDA:

According to Eq. 2.2, the joint probability distribution of the latent and the observed variables can be factorized as follows:

$$p(\mathbf{w}, \mathbf{c}, \mathbf{z}|\alpha_\theta, \alpha_\phi, \alpha_\varphi) = p(\mathbf{w}|\mathbf{z}, \alpha_\phi)p(\mathbf{c}|\mathbf{z}, \alpha_\varphi)p(\mathbf{z}|\alpha_\theta) \tag{2.8}$$

Let $n_a^{(b)}$ denote the number of times entity b is observed with entity a. Particularly, let $n_k^{(c)}$ denote the number of times document c is observed with topic k. According to the multinomial assumption on occurrences of citations, we obtain:

$$p(\mathbf{c}|\mathbf{z}, \varphi) \;=\; \prod_{i=1}^{C} p(c_i|z_i) \;=\; \prod_{k=1}^{K}\prod_{c=1}^{C} \varphi_{k,c}^{n_k^{(c)}}$$

$\varphi_{k,c}$ is the probability that document c to be cited with the topic k. The target posterior distribution for citation generation, i.e. $p(\mathbf{c}|\mathbf{z}, \alpha_\varphi)$, can be obtained by

$$p(z_i = k|\mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{k,-i}^{(t)}+\alpha_\phi^t}{\sum_{t=1}^{V} n_{k,-i}^{(t)}+\alpha_\phi^t} . \frac{n_{m,-i}^{(k)}+\alpha_\theta^k}{\sum_{k=1}^{K} n_{m,-i}^{(k)}+\alpha_\theta^k-1}; \text{ if } z_i \in (\mathbf{z}, \mathbf{w}). \tag{i}$$

$$p(z_i = k|\mathbf{z}_{-i}, \mathbf{w}, \mathbf{c}) \propto \frac{n_{k,-i}^{(t)}+\alpha_\phi^t}{\sum_{t=1}^{V} n_{k,-i}^{(t)}+\alpha_\phi^t} . \frac{n_{k,-i}^{(c)}+\alpha_\varphi^c}{\sum_{c=1}^{D} n_{k,-i}^{(c)}+\alpha_\varphi^c} . \frac{n_{m,-i}^{(k)}+\alpha_\theta^k}{\sum_{k=1}^{K} n_{m,-i}^{(k)}+\alpha_\theta^k-1}; \text{ if } z_i \in (\mathbf{z}, \mathbf{w}, \mathbf{c}) \tag{ii}$$

$$p(z_i = k|\mathbf{z}_{-i}, \mathbf{w}, \mathbf{c}) \propto \frac{n_{k,-i}^{(t)}+\alpha_\phi^t}{\sum_{t=1}^{V} n_{k,-i}^{(t)}+\alpha_\phi^t} . \frac{n_{k,-i}^{(c)}+\alpha_\varphi^c}{\sum_{c=1}^{D} n_{k,-i}^{(c)}+\alpha_\varphi^c} . \frac{n_k^{(.)}}{N_\leftarrow}; \text{ if } z_i \in (\mathbf{z}, \mathbf{w}^\leftarrow, \mathbf{c}) \tag{iii}$$

Table 2.2. Gibbs updates for cite-LDA(i,ii) and cite-PLSA-LDA(i,ii,iii)

integrating over all possible values of φ:

$$\begin{aligned} p(\mathbf{c}|\mathbf{z}, \alpha_\varphi) &= \int \prod_{z=1}^{K} \frac{1}{\Delta(\alpha_\varphi)} \prod_{c=1}^{D} \varphi_{z,c}^{n_z^{(c)}+\alpha_\varphi^c-1} d\varphi_z; \text{ where } \Delta(\alpha_\varphi) = \frac{\prod_{c=1}^{C} \Gamma(\alpha_\varphi^c)}{\Gamma(\sum_{c=1}^{C} \alpha_\varphi^c)} \times \\ &= \prod_{z=1}^{K} \frac{\Delta(\mathbf{n}_{\mathbf{z}\varphi} + \alpha_\varphi)}{\Delta(\alpha_\varphi)}; \text{ where } \mathbf{n}_{\mathbf{z}\varphi} = \{n_z^{(c)}\}_{c=1}^{D} \end{aligned}$$

A similar derivation holds for $p(\mathbf{w}|\mathbf{z}, \alpha_\phi)$ and $p(\mathbf{z}|\alpha_\theta)$ leading to the expression (see [21] for further details) for joint distribution:

$$p(\mathbf{w}, \mathbf{c}, \mathbf{z}|\alpha_\theta, \alpha_\phi, \alpha_\varphi) = \prod_{z=1}^{K} \frac{\Delta(\mathbf{n}_{\mathbf{z}\phi} + \alpha_\phi)}{\Delta(\alpha_\phi)} \prod_{z=1}^{K} \frac{\Delta(\mathbf{n}_{\mathbf{z}\varphi} + \alpha_\varphi)}{\Delta(\alpha_\varphi)} \prod_{d=1}^{D} \frac{\Delta(\mathbf{n}_\mathbf{m} + \alpha_\theta)}{\Delta(\alpha_\theta)}$$

For Gibbs sampler, we need to derive $p(z_i = k|\mathbf{z}_{-i}, \mathbf{w}, \mathbf{c})$ where $\mathbf{z}_{-i}$ denote the entire state space of z except the i^{th} token and i iterates over each word in the corpus. With some algebraic manipulation, the updates for cite-LDA can be shown equivalent to Eq. (i) & (ii) in Table 2.2. Here, (**z**,**w**) implies that z is sample from outside the citation context whereas (**z**,**w**,**c**) inside the citation context.

2.3.2 Inference Estimation for cite-PLSA-LDA:

The joint distribution of the hidden topic variables **z**, words **w** and the citations **c** can be written as:

$$p(\mathbf{w}, \mathbf{c}, \mathbf{z}|\alpha_\theta, \alpha_\phi, \alpha_\varphi, \pi) = p(\mathbf{w}|\mathbf{z}, \alpha_\phi)p(\mathbf{c}|\mathbf{z}, \alpha_\varphi)p(\mathbf{z}|\alpha_\theta)p(\mathbf{z}|\pi) \tag{2.9}$$

The derivation in previous section applies here, which leads to following algebraic

Algorithm 5: Gibbs sampling for cite-PLSA-LDA model

```
while Not Converged do
    while Not Converged do
        for each word token $w_n$ in citing documents: do
            if $w_n$ appears in citation context of cited document $c_n$ then
                sample $z_i$ from $p(z_i = k|\mathbf{z}_{-i}, \mathbf{w}, \mathbf{c})$ according to Eq.(ii), Table 2.2.
            else
                sample $z_i$ from $p(z_i = k|\mathbf{z}_{-i}, \mathbf{w})$ according to Eq.(i), Table 2.2.
            end if
        end for
    end while
    for each word token $w_n$ in cited documents: do
        sample $z_i$ from $p(z_i = k|\mathbf{z}_{-i}, \mathbf{w}, \mathbf{c})$ according to Eq.(iii) in Table 2.2.
    end for
end while
```

expression:

$$
\begin{aligned}
p(\mathbf{w}, \mathbf{c}, \mathbf{z}|\alpha_\theta, \alpha_\phi, \alpha_\varphi, \pi) = & \prod_{z=1}^{K} \frac{\Delta(\mathbf{n_z}_{\phi}^{\rightarrow} + \alpha_\phi)}{\Delta(\alpha_\phi)} \prod_{z=1}^{K} \frac{\Delta(\mathbf{n_z}_{\varphi}^{\rightarrow} + \alpha_\varphi)}{\Delta(\alpha_\varphi)} \prod_{d=1}^{D_{\rightarrow}} \frac{\Delta(\mathbf{n_m} + \alpha_\theta)}{\Delta(\alpha_\theta)} \\
& \times \prod_{z=1}^{K} \frac{\Delta(\mathbf{n_z}_{\phi}^{\leftarrow} + \alpha_\phi)}{\Delta(\alpha_\phi)} \prod_{z=1}^{K} \frac{\Delta(\mathbf{n_z}_{\varphi}^{\leftarrow} + \alpha_\varphi)}{\Delta(\alpha_\varphi)} \prod_{z=1}^{K} \pi_z^{n_z^{(.)}}
\end{aligned}
\tag{2.10}
$$

where (→)/(←) indicates that the corresponding token was seen in citing/cited set and $n_z^{(.)}$ indicates the number of times topic z was observed in the cited set.

The corresponding updates are obtained as given in Eq. (i), (ii) & (iii) in Table 2.2. However, as we noted in section 2.2.2.1, we intend to fit the topic distribution of words and citations learned from the citing set onto the cited set of documents. Therefore, a sequential scan over all the three partitions of the state space is inappropriate. Since the cited set of documents are not "generated" by the Cite-PLSA-LDA model, therefore, if we want to capture the conditional dependence based on topics between the citing set of documents and the cited set of documents, an iterative scheme of inference over citing documents and cited documents needs to be constructed. We begin with fitting the distribution over the cited set of documents and then generate the words in the citing set of documents. These two steps are repeated until convergence is reached. The corresponding Gibbs sampling update algorithm is depicted in Algorithm 5.

2.3.3 Inference Estimation for Switch-Cite-LDA:

Eq 2.4 can be factorized as follows:

$$p(\mathbf{w,z,c,s}|\alpha_\theta, \alpha_\lambda, \alpha_\phi, \alpha_\varphi) = p(\mathbf{w}|\mathbf{z}, \alpha_\phi)p(\mathbf{z}|\mathbf{c},\mathbf{s}, \alpha_\theta, \alpha_\lambda, \alpha_\varphi)p(\mathbf{s}|\alpha_\lambda) \tag{2.11}$$

The algebraic manipulation of the above factorization, as in section 2.3.1, lead us to following expression.

$$p(\mathbf{w}, \mathbf{z}, \mathbf{s} = \mathbf{1}, \mathbf{c}|\alpha_\theta, \alpha_\lambda, \alpha_\phi, \alpha_\varphi) = \\ \prod_{z=1}^{K} \frac{\Delta(\mathbf{n}_{\mathbf{z}\phi} + \alpha_\phi)}{\Delta(\alpha_\phi)} \times \prod_{c=1}^{K} \frac{\Delta(\mathbf{n}^{\mathbf{c}}_{\mathbf{z}\varphi} + \alpha_\varphi)}{\Delta(\alpha_\varphi)} \prod_{m=1}^{D} \frac{\Delta(\mathbf{n}^{\mathbf{m}}_{\mathbf{z}} + \alpha_\theta)}{\Delta(\alpha_\theta)} \prod_{c=1}^{C} \frac{n^c_{s_i=1} + \alpha_{\lambda_\varphi}}{n^c_{s_i} + \alpha_{\lambda_\varphi} + \alpha_{\lambda_\theta}} \tag{2.12}$$

The Gibbs sampler needs to sample from $p(z_i = k|\mathbf{z_{-i}}, \mathbf{w}, \mathbf{c}, \mathbf{s})$ where index i runs over all the words in the corpus. Depending upon the position of index i in the document, i.e. inside or outside the context, the distributions that sampler needs to sample from can be over following possible combinations of variables: $p(z_i = k|\mathbf{z_{-i}}, \mathbf{w})$, if outside the context window; $p(z_i = k|\mathbf{z_{-i}}, \mathbf{w}, \mathbf{c}, \mathbf{s})$, if inside the context window. Also, since switching variable s is unobserved, Gibbs sampler need to obtain a sample from $p(s_i = 0, 1|\mathbf{s_{-i}}, \mathbf{c})$. The algebraic forms of these above mentioned distributions is listed in Table 2.2.

2.3.4 Inference Estimation for Bipartite-Switch-Cite-LDA

The joint likelihood of hidden and observed variable in Eq 2.5 can be factorized as follows:

$$p(\mathbf{w,z,c,s}|\alpha_\theta, \alpha_\lambda, \alpha_\phi, \alpha_\varphi) = p(\mathbf{w}|\mathbf{z}, \alpha_\phi)p(\mathbf{z}|\mathbf{c,s}, \alpha_\theta, \alpha_\lambda, \alpha_\varphi)p(\mathbf{z}|\alpha_\varphi)p(\mathbf{s}|\alpha_\lambda) \tag{2.13}$$

As above, the algebraic form for the above equation can be obtained as below. The difference in Eq. 2.11 and Eq. 2.13 is that of one additional factor of cited documents' likelihood in Eq. 2.13.

$$p(\mathbf{w}, \mathbf{z}, \mathbf{s} = \mathbf{1}, \mathbf{c}|\alpha_\theta, \alpha_\lambda, \alpha_\phi, \alpha_\varphi) = \prod_{z=1}^{K} \frac{\Delta(\mathbf{n}_{\mathbf{z}\phi} + \alpha_\phi)}{\Delta(\alpha_\phi)} \prod_{c=1}^{K} \frac{\Delta(\mathbf{n}^{\mathbf{c}}_{\mathbf{z}\varphi} + \alpha_\varphi)}{\Delta(\alpha_\varphi)} \times \\ \prod_{m=1}^{D} \frac{\Delta(\mathbf{n}^{\mathbf{m}}_{\mathbf{z}} + \alpha_\theta)}{\Delta(\alpha_\theta)} \prod_{m=1}^{C} \frac{\Delta(\mathbf{n}^{\mathbf{m}}_{\mathbf{z}} + \alpha_\varphi)}{\Delta(\alpha_\varphi)} \prod_{c=1}^{C} \frac{n^c_{s_i=1} + \alpha_{\lambda_\varphi}}{n^c_{s_i} + \alpha_{\lambda_\varphi} + \alpha_{\lambda_\theta}} \tag{2.14}$$

$$p(z_i = k|\mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{k,-i}^{(t)}+\alpha_\phi^t}{\sum_{t=1}^V n_{k,-i}^{(t)}+V.\alpha_\phi^t} \cdot \frac{n_{m,-i}^{(k)}+\alpha_\theta^k}{\sum_{k=1}^K n_{m,-i}^{(k)}+K.\alpha_\theta^k} \qquad \text{(i)}$$

$$p(z_i = k|\mathbf{z}_{-i}, \vec{\mathbf{w}}) \propto \frac{n_{k,-i}^{(t)}+\alpha_\phi^t}{\sum_{t=1}^V n_{k,-i}^{(t)}+V.\alpha_\phi^t} \cdot \frac{n_{m,-i}^{(k)}+\alpha_\varphi^k}{\sum_{k=1}^K n_{m,-i}^{(k)}+K.\alpha_\varphi^k} \qquad \text{(ii)}$$

$$p(s_i = 0|\mathbf{c}, \mathbf{s_{-i}}) \propto \frac{n_{c,-i}^{(s_i=0)}+\alpha_{\lambda_\theta}-1}{\sum_{s_i} n_{c,-i}^{(s_i)}+\alpha_{\lambda_\theta}+\alpha_{\lambda_\varphi}-1} \qquad \text{(iii)}$$

$$p(s_i = 1|\mathbf{c}, \mathbf{s_{-i}}) \propto \frac{n_{c,-i}^{(s_i=1)}+\alpha_{\lambda_\varphi}-1}{\sum_{s_i} n_{c,-i}^{(s_i)}+\alpha_{\lambda_\theta}+\alpha_{\lambda_\varphi}-1} \qquad \text{(iv)}$$

$$p(z_i = k|\mathbf{z}_{-i}, \mathbf{c}, \mathbf{w}, s_i = 0) \propto \frac{n_{k,-i}^{(t)}+\alpha_\phi^t}{\sum_{t=1}^V n_{k,-i}^{(t)}+V.\alpha_\phi^t} \cdot \frac{n_{c,-i}^{(k,s=0)}+\alpha_\theta^k}{\sum_{k=1}^K n_{c,-i}^{(k,s=0)}+K.\alpha_\theta^k} \qquad \text{(v)}$$

$$p(z_i = k|\mathbf{z}_{-i}, \mathbf{c}, \mathbf{w}, s_i = 1) \propto \frac{n_{k,-i}^{(t)}+\alpha_\phi^t}{\sum_{t=1}^V n_{k,-i}^{(t)}+V.\alpha_\phi^t} \cdot \frac{n_{c,-i}^{(k,s=1)}+\alpha_\varphi^k}{\sum_{k=1}^K n_{c,-i}^{(k,s=1)}+K.\alpha_\varphi^k} \qquad \text{(vi)}$$

Table 2.3. Gibbs updates for Switch-Cite-LDA(i,iii, iv, v, vi) and Bipartite-Switch-Cite-LDA(i,ii,iii,iv,v,vi)

In contrast to the previous subsection, the Gibbs sampler needs to sample from cited documents as well. Therefore, each sampling iteration, we extend the Gibbs sampler to sample from $p(z_i = k|\mathbf{z_{-i}}, \vec{\mathbf{w}})$, where $\vec{\mathbf{w}}$ corresponds to words in the cited documents. The rest of the distributions are identical to the ones in previous subsection and their algebraic forms are listed in Table 2.3.

2.4 Experiments

2.4.1 Evaluation of Context Sensitive Topic Models for Citation Networks

We undertake two main tasks to quantitatively evaluate our proposed models:(1) comparison of the log-likelihood of unseen documents in test set, and, (2) capability of predicting outgoing links from the citing documents in a test set to the cited documents in the whole corpus.

2.4.1.1 Data Sets and Experimental Settings

We generate our citation networks from the following datasets: (1) scientific documents from the *CiteSeer* digital library, and (2) web-pages from the *WebKb* data set. These datasets have also been utilized by Nalapatti, et al. [42] for the two tasks.

CiteSeer Dataset: This labeled dataset[3] was made publicly available by Lise Getoor's research group at the University of Maryland and is derived from the CiteSeer[4] digital library. The data set contains 3312 documents belonging to six different research fields and the vocabulary consists of 3703 unique words. There are a total of 4132 links present in the data set. We supplement the data set with the context information for each link. For each link, we add 60 words in the radius of size 30 originating at the citation mention in the document. We vary the radius; that proves to be crucial for the performance of the models (described later). As pre-processing, we remove 78 common stop words and stem the words with the Porter Stemmer, which gives us 1987 unique words in the corpus. Further, we split the 1485 citing documents into 10 sets of 70-30 training and test split respectively. Since the link-PLSA-LDA and cite-PLSA-LDA model require bipartite structure for the corpus, we split the documents into two sets with duplication as suggested by Nalapatti, et al. plsa-lda.

CiteULike Dataset: For evaluations on a user selected scientific documents dataset, I also acquired dataset from CiteULike [5] for over 2 years from November 2005 to January 2008 (referred as *CiteSeer-DS2*). The dataset is available at http://citeulike.org. Overall, there are 33,456 distinct papers in CiteULike sample. I map the document ids of CiteULike documents to document ids of CiteSeer documents [6] to gain access to citation network of the sample. The resultant CiteSeer-DS2 contains 18354 documents in which 9571 documents are cited. There are a total of 29645 unique authors in CiteSeer-DS2 out of which 15967 authors are cited at least once. I follow the same preprocessing step as the CiteSeer dataset mentioned above.

WebKb Dataset: We also used the WebKb dataset that was collected by the CMU WebKb project[7]. The dataset consists of web pages from the computer science department of various US universities. It includes faculty, staff, project and course web pages. The dataset consists of 2,877 different web pages with a vocabulary size of 102,927 words. After removing the stop words and stemming, the

[3] http://www.cs.umd.edu/s̃en/lbc-proj/LBC.html

[4] http://CiteSeer.ist.psu.edu/

[5] http://citeulike.org

[6] mapping is obtained from http://citeulike.org

[7] http://www.cs.cmu.edu/ WebKB/

vocabulary size is 24,447 words. Note that the large vocabulary size as compared to CiteSeer dataset is due to the fact that a majority of pages contain unique nouns like faculty and staff names, project names etc. We found 1764 citations (hyperlinks) in the dataset with an average anchor text length of 3.02 words.

Experimental Set-up: Considerable prior work has been done on hyper-parameter tuning for topic models [54], however, we choose to fix the hyper-parameters and evaluate different models within same setting. We set the hyper-parameters to the following values: $\alpha_\theta = 50/T$, $\alpha_\phi = 0.01$, $\alpha_\varphi = 0.01$. We fix the number of topics to 100 except in cases where we indicate otherwise. We run 1000 iterations of Gibbs sampling for training and extend the chain with 100 iterations over the test set. The multinomial parameters of the model are calculated by taking expectations of the corresponding counts from 10 samples collected during test iterations.

For dynamic window selection, I collect 10 samples from the chain after every 10 iterations starting from 1000 iterations, and compute the new window with the average of the samples using Eq. 2.7. After the window update, I let the chain converge and start to update the window again. Starting with the sentence that contains the citation mention, I allow our window to grow up to a maximum of 5 sentences in either direction. The multinomial parameters of the model are calculated by taking expectations of the corresponding counts from 10 samples collected during test iterations.

2.4.1.2 Loglikelihood Estimation on Unseen Text

We estimate quantitatively the generalization capabilities of a given model over unseen data. In order to find the log-likelihood of words in the test set, we followed a similar approach taken byRosen-Zvi, et al., atm where the inference algorithm is run exclusively on the new set of documents. We achieve this by extending the state of the Gibbs sampler with the observation of the new documents. Before *sweeping* the test set, our algorithm first randomly assigns topics to the words and the citations in the test set and then loops through the test set, until convergence, using following Gibbs sampling updates:

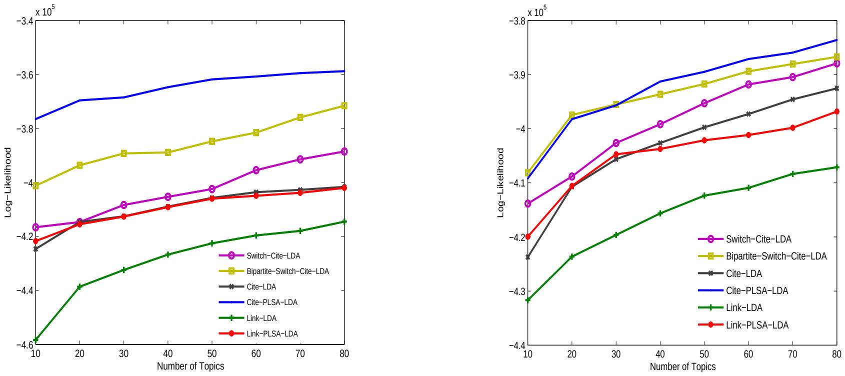
$$p(z_i^u | w_i^u = t, \mathbf{z}^{\mathbf{u}}_{-\mathbf{i}}, \mathbf{w}^{\mathbf{u}}_{-\mathbf{i}}) = \frac{n^{(t)}_{k,-i} + \beta}{\sum_{t=1}^{V} n^{(t)}_{k,-i} + V.\beta} . \frac{n^{(k)}_{m^u,-i} + \alpha}{\sum_{k=1}^{K} n^{(k)}_{m^u,-i} + K.\alpha - 1} \tag{2.15}$$

where the superscript $(.^u)$ stands for any unseen element. The sampling updates in Eq. 7 can be used to update the model parameters, $\Pi = (\theta, \phi, \varphi)$ for new documents as:

$$\theta_{m^u,k} = \frac{n_{m^u}^{(k)} + \alpha_k}{\sum_{k=1}^{K} n_{m^u}^{(k)} + \alpha_k}; \phi = \frac{{n_k^{(t)}}^u + n_k^{(t)} + \beta_t}{\sum_{t=1}^{V} {n_k^{(t)}}^u + n_k^{(t)} + \beta_t}$$

The predictive log-likelihood of a text document in the test set, i.e. $log(p(\mathbf{w^u}))$, given the model $\Pi = (\theta, \phi, \varphi)$ can be directly expressed as a function of the multinomial parameters of any given model:

$$p(\mathbf{w^u}|\mathbf{\Pi}) = \prod_{n=1}^{N_{m^u}} \sum_{k=1}^{K} p(w_n|z_n = k).p(z_n = k|d = m^u) = \prod_{t=1}^{V} (\sum_{k=1}^{K} \phi_{k,t}.\theta_{m^u,k})^{n_{m^u}^{(t)}} \qquad (2.16)$$



2.5.1 Citeseer Dataset

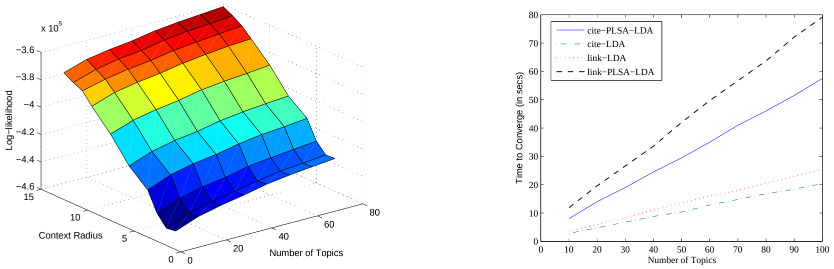
2.5.2 WebKb Dataset

Figure 2.5. Comparision of Loglikelihood (a & b) for the proposed models with link-PLSA-LDA [42] and link-LDA [50] on CiteSeer and WebKb datasets.

Fig. 3.3(a) & (b) show the comparison results on the two data sets. For both the data sets, we perform a 10-fold cross validation and report the average of the log-likelihood. Clearly, the cite-PLSA-LDA model outperforms all the other models on both of the data sets. The improvement in the performance is due to the fact that the association between a citation and the words appearing in its context helps to identify the topic of the word. Also, the performance of cite-LDA and link-PLSA-LDA is comparable because, for obtaining the topical association of words in the citing document, the information provided by the link structure of the corpus and the context of the links is as good as the content of cited document.

The improvements obtained for the CiteSeer data set is relatively larger than that obtained in the WebKb data set. Primarily because the contexts of the links in the WebKb data set are very noisy. There are very few instances where the author of the web-page discusses a scientific project or his work and inserts some links that are relevant to that discussion. In most cases, the links correspond to class projects and departmental home-pages that do not have any context information close to the position of the link. On the other hand, the citations in the CiteSeer data always appear along with a context that describes the topic of the cited document.

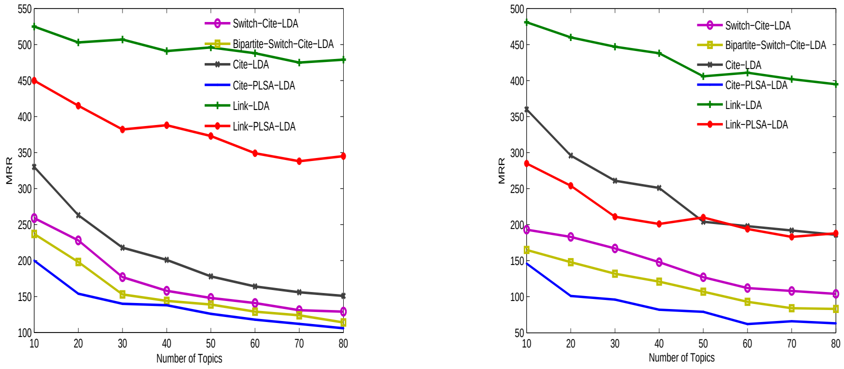
Next, we discuss the effect of varying the context radius on the performance of the *cite* models. We measure the radius from the citation mention and vary it from 3 to 15 words. We observe a rapid increase in the log-likelihood function with the radius increasing from 3 to 10 words. After 10 words, the log-likelihood starts to stabilize and does not vary much after 14 words. This is mainly because after 10 words radius, the topic of discussion, generally, broadens beyond the topics in the cited document. Also, for the WebKb data, we observed this trend to appear only after 6 words of radius. Fig. 2.6(a) shows, for cite-PLSA-LDA, the change in log-likelihood with the change in the number of topics and the context radius. The same trend was observed for *link* models as well. The automatic selection of the appropriate radius for a given corpus will be of interest in future work.



2.6.1 context radius vs. number of topics vs. loglikelihood

2.6.2 Convergence time on CiteSeer data

Figure 2.6. (a) Effect of varying context length in CiteSeer data. (b) Comparision of convergence time for the four models on CiteSeer data



2.7.1 Maximum rank at 100% recall for Citeseer Dataset

2.7.2 Maximum rank at 100% recall for WebKb Dataset

Figure 2.7. Comparision of link prediction task (a & b) for the proposed models with link-PLSA-LDA [42] and link-LDA [50] on CiteSeer and WebKb datasets.

2.4.1.3 Link Prediction

The experimental design for this task is very similar to the one in previous subsection. First, we ran the inference algorithm, described in the previous section, on the training set for each model. Then, we extended the Gibbs sampler state with the samples from the test set using the following updates:

$$
\begin{aligned}
p(z_i^u|w_i^u = t, \mathbf{z^u_{-i}}, \mathbf{w^u_{-i}}) &= \frac{n_{k,-i}^{(t)} + \beta}{\sum_{t=1}^{V} n_{k,-i}^{(t)} + V.\beta} . \frac{n_{m^u,-i}^{(k)} + \alpha}{\sum_{k=1}^{K} n_{m^u,-i}^{(k)} + K.\alpha - 1} \\
p(z_i = k|\mathbf{z}_{-i}, \mathbf{w}, \mathbf{c}) &\propto \frac{n_{k,-i}^{(t)} + \beta}{\sum_{t=1}^{V} n_{k,-i}^{(t)} + V.\beta} . \frac{n_{k,-i}^{(c)} + \gamma}{\sum_{c=1}^{D} n_{k,-i}^{(c)} + D.\gamma} \\
&\quad . \frac{n_{m^u,-i}^{(k)} + \alpha}{\sum_{k=1}^{K} n_{m^u,-i}^{(k)} + K.\alpha - 1}; \text{ if } z_i \in (\mathbf{z}, \mathbf{w}, \mathbf{c}) \qquad (2.17)
\end{aligned}
$$

The parameters ϕ and θ were obtained as in Eq. 8, and the parameter φ can be obtained as:

$$
\varphi = \frac{n_k^{(c)^u} + n_k^{(c)} + \gamma_t}{\sum_{c=1}^{D} n_k^{(c)^u} + n_k^{(c)} + \gamma_t}
$$

The probability $p(c|w_d)$, where c is the document to be cited and w_d is the citing document, can be expressed as:

$$
p(c|w_d) = \sum_z p(c|z) \int p(z|\theta_d) d\theta_d \propto \sum_k \varphi_{c,k}.\theta_{k,d}
$$

To evaluate the different models, we take an approach similar to the one taken by Nallapatti, et al. [42]. We label the actual citations of the document as the relevant set for that citing document and evaluate our models based upon what rankings are given to these actual citations. In Fig. 2.7 (a) & (b), we plot the average of the maximum rankings given to these relevant links. The plot shows the average over all the test set. Clearly, the lower the rank assigned by the model, the better it performs. Cite-PLSA-LDA outperforms all the other models and cite-LDA and link-PLSA-LDA have comparable performance. Other models outperform the link-LDA model because of its over-simplicity.

We also evaluate our models with recently published relevance framework based context sensitive citation recommendation (mentioned as *Relevance*) [24]. The setting for Relevance method used in the following evaluation is the one described as "local" recommendation in [24]. That is, for each citation context, the position of original citation is used for evaluation purposes. The normalization in this case is done based upon total number of citation contexts as opposed to total number of documents in other evaluation metrics.

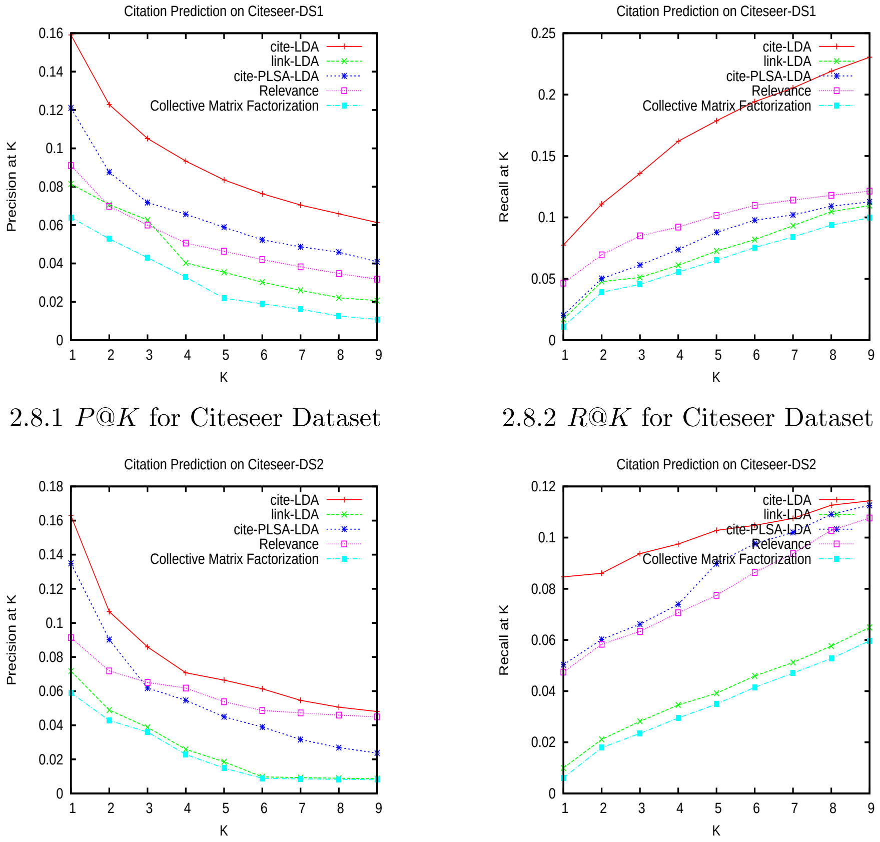
We take three evaluation metrics for this comparison: (1) Precision@ K, (2) Recall @ K, and (3) NDCG. Here, NDCG is defined as below. Here, $rel(i)$ is 0, if a document is not cited and 1 otherwise. Also, p ranges over all the missing links in a given citing document. The below mentioned NDCG metric is normalized.

$$NDCG = rel(1) + \sum_{i=2}^{p} \frac{rel(i)}{log_2 i}$$

Fig. 2.8 shows the evaluation of different models on link prediction task. Table 2.4 shows the NDCG evaluation on two CiteSeer datasets. Strangely, although we observed that introducing more complexity to model improves the log-likelihood criterion but does not improve the link prediction criterion. For link prediction, a simple context sensitive extension, i.e. cite-LDA, provides the best results.

Table 2.4. NDCG Evaluation on CiteSeer datasets

	Cite-LDA	Link-LDA	Cite-PLSA-LDA	Relevance	Collective Matrix Factorization
CiteSeer	0.3152	0.1496	0.1680	0.2403	0.1134
CiteSeer-DS1	0.3221	0.1254	0.2580	0.2703	0.08134



2.8.1 $P@K$ for Citeseer Dataset

2.8.2 $R@K$ for Citeseer Dataset

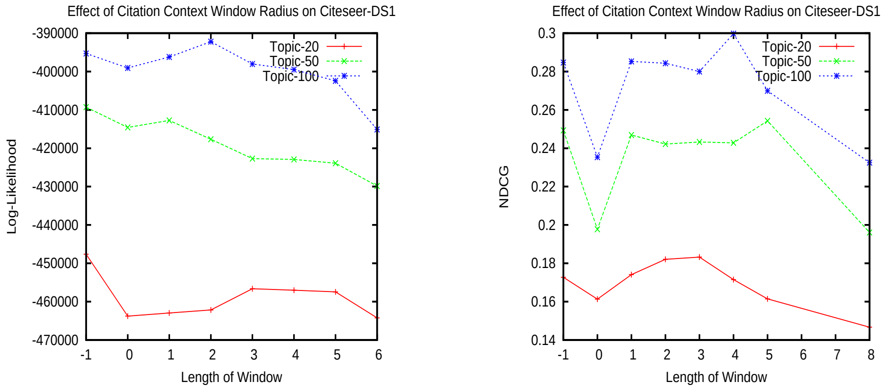
2.8.3 $P@K$ for Citeseer-DS2 Dataset

2.8.4 $R@K$ for Citeseer-DS2 Dataset

Figure 2.8. Comparision of link prediction task (a & b) for the proposed models .

2.4.1.4 Evaluating Adaptive Window Selection

Fig. 2.9 shows the results of evaluating the effect of window length on various evaluation metric. Fig. 2.9(a) shows how the log-likelihood on test set varies with various window lengths. In Fig. 2.9, -1 indicates the adaptive window length, whereas all the other lengths refer to a fixed radius surrounding the citation mention sentence. Adaptive window performs very well on the likelihood criterion; however, link prediction is only above average performance. We believe this is because the likelihood and link prediction are not correlated functions by definition, therefore, a good performance on one does not guarantee a similar performance on the other.



2.9.1 Model Fitting with Adaptive Window Selection

2.9.2 Link Prediction with Adaptive Window Selection

Figure 2.9. Experiments with varying window size and adaptive window selection. These experiments are carried out on CiteSeer Dataset. Here, -1 indicates adaptive window length.

2.4.1.5 Complexity Analysis and Runtimes

For link-LDA and link-PLSA-LDA, the time complexity of a single iteration of the Gibbs sampler grows linearly with the number of links present in the corpus. This growth can be prohibitive in the case of large corpuses such as the world-wide-web where the number of links are in the order of 10^6. For cite-LDA and cite-PLSA-LDA, the modeling of the citation variable is explicitly associated with the word variable. Therefore, sampling from the posterior distribution of the topic variable does not depend upon the number of links and only grows linearly with the number of words in the corpus. The time complexity of one sampling iteration from the citing set for cite-PLSA-LDA and cite-LDA is $O(\sum_d \sum_n d_n * K)$ where K is the number of topics, d is the iterator over the documents and n is the iterator over the words in document d, whereas it is $O(\sum_d (\sum_n d_n * K + \sum_l d_l * K)$ for link-LDA and link-PLSA-LDA, where l is iterator over citations in document d.

Runtimes: Fig. 2.6(b) shows the convergence time for the four models on the CiteSeer data with varying number of topics. For the cite-PLSA-LDA and the link-PLSA-LDA model, we compare the performance of the outer loop of Gibbs sampling until the model parameters reach convergence. The performance of cite-LDA and link-LDA is comparable and the performance of cite-PLSA-LDA and link-PLSA-LDA model is comparable. In both cases, the former perform better than the latter.

2.4.1.6 Anecdotal Evidences

Table 2.5(a) and (b) show the topics assigned to the example citation context in Table 2.1 by link-LDA [13] and cite-LDA respectively. The words belonging to the "support vector machine" topic in the document are assigned one topic whereas the "latent semantic indexing" topic words in the citation context are assigned different topics. However, the assignments by cite-LDA (Table 2.5(b)) are also coherent for the latter topic. The coherent topic assignment is due to the affect of simultaneous sampling of topics for citation as well as words in the citation context.

(a) Unlike conventional approaches to learning text classifiers, which rely primarily on empirical evidence, this model explains why and when SVMs perform well for text classification. In particular, it addresses the following questions: Why can support vector machines handle the large feature spaces in text classification effectively? ... Papadimitriou et. al, is most similar in spirit to the approach presented here[16]. They show that latent semantic indexing leads to a suitable low dimensional representation...

(b) Unlike conventional approaches to learning text classifiers, which rely primarily on empirical evidence, this model explains why and when SVMs perform well for text classification. In particular, it addresses the following questions: Why can support vector machines handle the large feature spaces in text classification effectively? ... Papadimitriou et. al, is most similar in spirit to the approach presented here[16]. They show that latent semantic indexing leads to a suitable low dimensional representation...

Table 2.5. Topic assignments recovered using (a) link-LDA [13] and (b) cite-LDA (this chapter). Different colors indicate different topics.

2.5 Summary

We presented a framework that utilizes context information of citations in documents to model the generation process of documents and citations. The context around a citation mention in a citing document has topical information regarding the relationship between citing and cited document. We show how to statistically model the citation context explicitly. Our model explains the generation process of the links and content both qualitatively and quantitatively. We utilize Gibbs sampling to perform inference on emission probabilities corresponding to citations and words given a topic and show significant improvement on various objective functions. We test our models on several real-world datasets and observe that, quantitatively, the cite-PLSA-LDA model outperforms the models which do not model the cited documents and context information explicitly. The cite-PLSA-LDA model achieves superior performance in both missing citation prediction and model fitting experiments.

In addition, we propose novel models for author-author linkage conditioned on topics latent in the content of the documents. We exploit the citations between documents to infer influence of certain authors over topics. We also propose context sensitive extensions of the proposed model that incorporates the context of the cited document and how it infers the topic of both cited and citing authors with better quality.

Chapter 3

Context Sensitive Models for Authorship Networks

In a document network such as a citation network of scientific documents, weblogs, etc., the content produced by authors exhibits their *interest* in certain *topics*. In addition some authors *influence* other authors' interests. In this work, I propose to model the influence of cited authors along with the interests of citing authors. Moreover, I hypothesize that apart from the citations present in documents, the context surrounding the citation mention provides extra topical information about the cited authors. However, associating terms in the context to the cited authors remains an open problem. I propose novel document generation schemes that incorporate the context while simultaneously modeling the interests of citing authors and influence of the cited authors. The experiments show significant improvements over baseline models for various evaluation criteria such as link prediction between document and cited author, and quantitatively explaining unseen text.

3.1 Introduction

The popularity of Web 2.0 applications has resulted in large amounts of online text data, e.g. weblogs, digital libraries of scientific literature, etc. These data require *effective* and *efficient* methods for their organization, indexing, and summarization, to facilitate delivery of content that is tailored to the interests of specific individuals or groups. Topic models such as Latent Dirichlet Allocation (LDA) [7] and Probabilistic Latent Semantic Analysis (PLSA) [26] are generative models of text documents, which successfully uncover hidden structures, i.e., *topics*, in the data. They model the co-occurrence patterns present in text and identify a probabilistic membership of words and documents into a much lower dimensional space compared to the original term space. Since their introduction, many extensions have been proposed.

One such line of research aimed at modeling the interests of authors to answer important queries about authors, e.g., who produced similar work [47], who belongs to the same research community [34] and who are the experts in a domain [53]. However, another fundamental question about the attribution of topics to authors still remains not answered: who influences the generation of new content in a particular topic of interest? In this work, I propose generative models that take the linkage between authors of citing and cited documents into consideration and explore various qualitative and quantitative aspects of this question.

Another line of research aimed at modeling topics for content and citations together to quantify the influence of citations over the newly generated content [16, 42, 11, 28]. However, these statistical methods for parameterizing the influence of a document cannot easily quantify the influence of authors because one document often has multiple authors.

In this chapter, I exploit the complementary strengths of the above lines of research to answer queries related to authors' influence on topics. Specifically, I present two different generative models for inter-linked documents, namely the author link topic (ALT) and the author cite topic (ACT) models, which simultaneously model the content of documents, and the interests as well as the influence of authors in certain topics. As in the author topic model (ATM) [47], ALT models a document as a mixture of topics, with the weights of the mixture being determined by the authors of the document. In order to capture the influence of cited authors, ALT extends ATM to let the set of cited authors in a document be represented as a mixture of topics and again the weights of the topics are determined by the authors of the document.

Moreover, I hypothesize that the context in which a cited document appears in a citing document indicates how the authors of the cited document have influenced the contributions by the citing authors. ACT extends ALT to explicitly incorporate the citation context, which could provide additional information about the cited authors. Kataria et al. kataria have previously used the citation context while jointly modeling documents and citations (without authors) and have shown that a fixed-length window around a citation mention can provide improvements over context-oblivious approaches. Unlike Kataria et al. kataria, I model the authors of the document along with the content and argue that a fixed-length window around a citation mention can provide either limited or erroneous information in cases where the context spans are larger or smaller, respectively, than the length of the window. Hence, I dynamically select an adaptive-length window around a citation that is statistically more likely to explain the cited document than a fixed-length window.

In summary, this chapter has following research contributions:

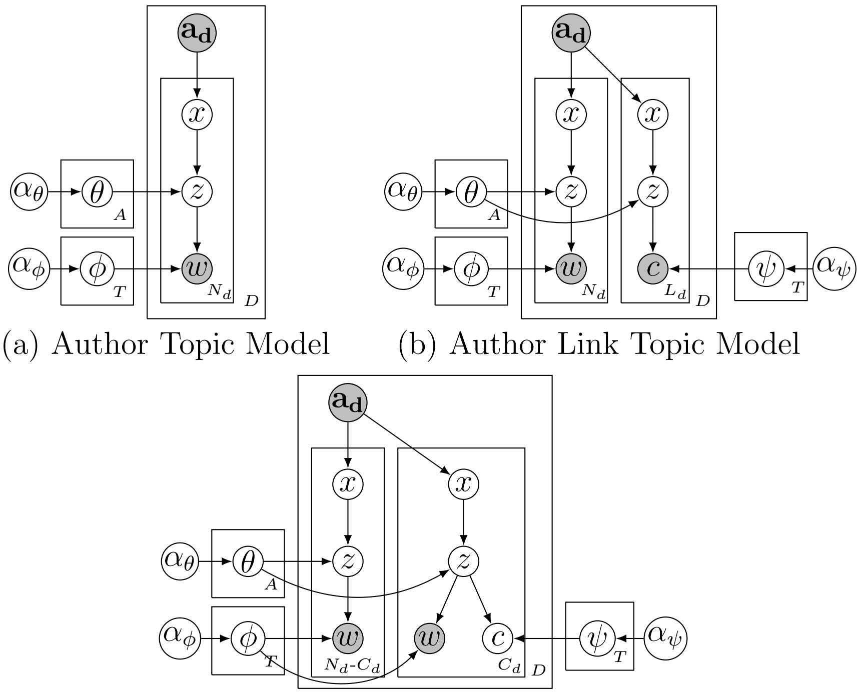
- I propose generative models for author-author linkage from linked documents conditioned on topics of interest to authors. Our models are able to distinguish between authors' interests and authors' influence on the topics.
- I utilize the context information present in the citing document explicitly while modeling the cited authors and obtain significant benefits on evaluation metrics on real world data sets. Moreover, I dynamically select the length of context surrounding the citation mention and circumvent the erroneous context inclusion by a fixed window approach.

3.2 Related work

One of the earliest attempts at modeling the interests of authors is the author topic model (ATM) [47], where the authors and the content are simultaneously modeled with coupled hyper-parameters for the interests of authors and the themes present in text (shown in Fig. 3.1(a)). The (latent) topics represent the shared dimensions among the interest of authors and the themes. Bhattacharya and Getoor entity-lda extended ATM to disambiguate incomplete or unresolved references to authors. Another stream of author centric modeling deals with expert finding [17, 1, 53] where an expert is defined as a person *knowledgeable* in the field. I define an *expert/interested* author as someone who has produced several contributions in a particular field whereas an influential author as someone who has certain key contributions in that field and gets cited more often. Therefore, given a field, an influential author is not necessarily an expert in that field, however, her key contributions have led several interested authors to contribute to that field. However our main goal is to model the influence of authors [29] along with the interest of authors.

Linking to external content or entities is an important ingredient of social content such as citation graph of academic documents, asynchronous communications such as weblogs, e-mails, etc. The *mixed membership model* [50], also referred as *linked-LDA* [42], extended LDA to model links among documents with an additional parameter that governs link generation from citing documents to cited documents. Further extensions of *linked-LDA* analyzed the association between words and hyperlinks [42, 23, 11], influence propagation [16], community of links detection [34], context-sensitive citation and text modeling [28]. To model the authors in an inter-linked corpus of documents, Tu et al. TJRH10 proposed an extension of the author topic model to inter-linked documents. In contrast to our approach, they consider the entire citing document as the context of the citation, which, as explained in 3.4.2, can easily be considered as a special case of our approach. In addition, it performs inferior to dynamically selecting the context length.

Topic models have also been extended to social networks of entities where entity-entity relationships conditioned upon topics are explored. Mccallum, et al., enron extended the basic ATM to cluster the entity pairs based upon topic of conversation in e-mail corpus. Their approach assumes that the sender and the recipient both decide the entire topic of conversation. This assumption is not applicable in our setting because only the author of the citing document decides the topic of the document and every cited authors may not share the interest in all the topics discussed in citing document. Newman et al. entity-topic and Shiozaki et al. entity-topic1 proposed other entity-entity relationship models for named-entities in news articles where documents are modeled as mixture of topics over both entities and words.



(a) Author Topic Model (b) Author Link Topic Model

(c) Author Cite Topic Model

Figure 3.1. Plate diagram for: (a) Author Topic Model; (b) Author Link Topic Model; and (c) Author Cite Topic Model.

3.3 Models

Before presenting our models, I introduce some useful notations. Let V, D, A, $\mathbf{a_d}$ and N_d denote the size of the word vocabulary, the number of documents, the number of authors, a set of authors and the number of words in document d respectively. Let T denote the number of latent topics, i.e., the latent variable z (see Fig. 3.1) can take any value between 1 and T inclusively. Suppose there exists a $T \times V$ topic-word distribution matrix ϕ that indexes a probabilistic distribution over words given a topic and a $T \times A$ topic-author distribution matrix θ that indexes the probability with which an author shows interest in a topic. The corresponding hyper-parameters for distributions ϕ and θ are α_ϕ and α_θ respectively.

3.3.1 Detecting Influential Authors with Author Link Topic Model

Citations among documents exhibit the biases of citing authors towards certain influential authors who have key contributions in the topic of discourse. I quantify the influence of an author given a topic by the probability, denoted by φ_{cz}, that the author c's work gets cited when there is a mention of the topic z in a citing document. Since the Author Topic Model (ATM) does not model the citations among the documents, it is not possible to estimate the influence of an author given a topic. In contrast, Author link topic model (ALT) generates the references to cited authors along with the words from a mixture of topics. As in ATM, a set of authors $\mathbf{a_d}$ decides to write a document. To generate each word, an author x is chosen uniformly at random from $\mathbf{a_d}$, and a topic is sampled from the chosen author's specific distribution. Then the corresponding word is generated from the chosen topic. For each author in the referenced set of authors in the document d, again an author x is chosen to generate a topic, and based upon the topic, an author c is selected from the topic specific distribution over authors. ALT model captures the intuition that given a topic and a list of relevant authors to be cited, authors from $\mathbf{a_d}$ would choose to reference those authors's work that are influential in that topic. Fig. 3.1(b) shows the plate diagram for the ALT model.

In the following subsections, I will use $\mathbf{w}$ and $\mathbf{c}$ to denote the words and observed cited authors in a document and $\mathbf{z}$ to denote the vector of topic assignments in the document. With the model hyper-parameters α_θ, α_ϕ and α_φ, the joint distribution of authors $\mathbf{x}$, the topic variables $\mathbf{z}$, the document $\mathbf{w}$ and the cited authors $\mathbf{c}$ can be written as below. Here, L_d stands for the number of cited authors in the document d.

$$p(\mathbf{x},\mathbf{c},\mathbf{z},\mathbf{w}|\mathbf{a_d}, \alpha_\theta, \alpha_\phi, \alpha_\varphi) = \tag{3.1}$$

$$\int\int\int \prod_{n=1}^{N_d} p(x|\mathbf{a_d})p(z_n|x,\theta_x)p(w_n|z_n,\phi_{z_n})p(\theta_x|\alpha_\theta)p(\phi_{z_n}|\alpha_\phi)$$

$$\prod_{l=1}^{L_d} p(x|\mathbf{a_d})p(z_l|x,\theta_x)p(c_l|z_l,\varphi_{z_l})p(\theta_x|\alpha_\theta)p(\varphi_{z_l}|\alpha_\varphi)d\theta d\phi d\varphi$$

3.3.2 Context sensitive modeling of Author-linkage : Author Cite Topic Model

ALT model does not utilize the context in which a document cites an author. Although ALT models the cited authors in the citing document, yet, because of the bag of words assumption, the topic assignment to the authors does not explicitly depend upon the topics assigned to the content in that document. To enforce this dependence, I model the cited authors along with the context of the citation. In contrast to ALT, the Author Cite Topic (ACT) model associates cited authors and the words in the citation context of the cited authors with topic assignments to the context words. This association is based upon the assumption that given a topic, the choice of words and the authors to be cited are independent (see the plate diagram in Fig 3.1(c). With this independence assumption, the topic sampled for words in the citation context window generates both a word and a reference to the cited author. Since I observe a set of authors for a cited document, I treat c as hidden similar to x. The parameters of the ACT model remain the same as those of the ALT model, however the complete data log-likelihood function is different due to a difference in the generation process. The log-likelihood function to optimize can be written as below. Here, C_d is the total length (number of words) of all citation contexts in the document d.

$$\begin{aligned}
&p(\mathbf{x},\!\mathbf{c},\!\mathbf{z},\!\mathbf{w}|\mathbf{a_d}, \alpha_\theta, \alpha_\phi, \alpha_\varphi) = \\
&\int\!\!\int\!\!\int \prod_{n=1}^{N_d - C_d} \Big(p(x|\mathbf{a_d}) p(z_n|x, \theta_x) p(w_n|z_n, \phi_{z_n}) p(\theta_x|\alpha_\theta) \\
&p(\phi_{z_n}|\alpha_\phi) \Big) \prod_{n=1}^{C_d} \Big(p(x|\mathbf{a_d}) p(z_n|x, \theta_x) p(\theta_x|\alpha_\theta) p(w_n|z_n, \phi_{z_n}) \\
&p(\phi_{z_n}|\alpha_\phi) p(c_n|z_n, \varphi_{z_n}) p(\varphi_{z_n}|\alpha_\varphi) \Big) d\theta d\phi d\varphi
\end{aligned} \tag{3.2}$$

Intuitively, Eq. 3.2 implies that the author first picks the words from the topic and then chooses to cite an author's work or vice versa. The product $p(z_n|x, \theta_x).p(w_n|z_n, \phi_{z_n})$ acts as the mixing proportions for the author "generation" probability over the entire citation context of the corresponding citation. Therefore, one can expect that this explicit relation between citation generation probability and the word generation probability will lead to a better association of words and citations, and in turn authors, with documents than without utilizing the citation context explicitly.

3.3.3 Dynamic Selection of Length of Context Window

Since the ACT model imposes independence assumption in the context window surrounding the citation mention, it becomes important to identify the context that refers to the cited article. Previous work on context utilization in topic models, either assumes a fixed window of 10 words radius surrounding the citation mention [28] or the whole document as the context for any citation mention [53]. However, the amount of relevant context in the vicinity of the citation anchor depends upon various factors such as the strength of the influence of cited article over the citing article, the location of the citation mention in the citing article, etc. Therefore, I propose to identify a dynamic window surrounding the citation anchor with the following method.

Let $\overleftarrow{d}$ represent the cited document for a given citation anchor c_i^d, where i ranges over all citation mentions in the citing document d. Let $S(c_i^d)$ (or simply S_i) represent the bag of words in the citation context surrounding c_i^d. The objective function that I choose to maximize is $f(\overleftarrow{d}|S_i)$ which is defined as:

$$f(\overleftarrow{d}|S_i) = \sigma(Z_{\overleftarrow{d}}.Z_{S_i}) \tag{3.3}$$

Here, Z_p is the topic vector defined as $\frac{1}{N_p}\sum_n z_{p,n}$ where n ranges over all the tokens in the bag p and N_p denotes the cardinality of p. σ represents the sigmoid function and $'.'$ represents the dot product between two vectors. Intuitively, $f(\overleftarrow{d}|S_i)$ represents the topical similarity between cited document and its corresponding context.

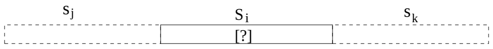


Figure 3.2. An illustrative citation context window

Next I describe our dynamic context selection procedure. I allow our window to grow over sentences beginning with the sentence that has the citation mention, although the method proposed is general enough to be applicable to any building block such as words or paragraphs. I choose to begin with the sentence that contains the citation mention as the sentence carries most of the information about the cited document. I let S_i denote the current context window and s_j and s_k are the next left and right candidates to either include in the window or to let the growth stop in either direction. I update the window as defined below and continue to grow in the direction which maximizes the objective function 3.3.

$$S_i = max\{f(\overleftarrow{d}|S_i), f(\overleftarrow{d}|\{S_i, s_j\}), f(\overleftarrow{d}|\{S_i, s_k\}), f(\overleftarrow{d}|\{S_i, s_j, s_k\})\} \tag{3.4}$$

3.3.4 Inference using Gibbs Sampling

I utilize Gibbs sampling as a tool to approximate the posterior distribution for both the models. Specifically, I want to estimate θ, ϕ and φ parameters of the multinomial distributions $Multi(.|\theta)$, $Multi(.|\phi)$ and $Multi(.|\varphi)$, respectively, in Fig. 3.1(b) and 3.1(c).

$p(z_i = k, x_i = x \vert \mathbf{z}_{-i}, \mathbf{x}_{-i}, \mathbf{w}) \propto \frac{n_{k,-i}^{(t)} + \alpha_\phi^t}{\sum_{t=1}^{V} n_{k,-i}^{(t)} + V.\alpha_\phi^t} . \frac{n_{x,-i}^{(k)} + \alpha_\theta^k}{\sum_{k=1}^{K} n_{x,-i}^{(k)} + K.\alpha_\theta^k}$	(i)
$p(z_i = k, x_i = x \vert \mathbf{z}_{-i}, \mathbf{x}_{-i}, \mathbf{c}) \propto \frac{n_{k,-i}^{(c)} + \alpha_\varphi^c}{\sum_{c=1}^{C} n_{k,-i}^{(c)} + C.\alpha_\varphi^c} . \frac{n_{x,-i}^{(k)} + \alpha_\theta^k}{\sum_{k=1}^{K} n_{x,-i}^{(k)} + K.\alpha_\theta^k}$	(ii)
$p(z_i = k, x_i = x, c_i = c \vert \mathbf{z}_{-i}, \mathbf{x}_{-i}, \mathbf{c}_{-i}, \mathbf{w}) \propto \frac{n_{k,-i}^{(t)} + \alpha_\phi^t}{\sum_{t=1}^{V} n_{k,-i}^{(t)} + V.\alpha_\phi^t} . \frac{n_{k,-i}^{(c)} + \alpha_\varphi^c}{\sum_{c=1}^{C} n_{k,-i}^{(c)} + C.\alpha_\varphi^c} . \frac{n_{x,-i}^{(k)} + \alpha_\theta^k}{\sum_{k=1}^{K} n_{x,-i}^{(k)} + K.\alpha_\theta^k}$	(iii)

Table 3.1. Gibbs updates for ALT(i,ii), ACT(i,iii)

According to Eq. 3.2, the joint probability distribution of the latent and the observed variables can be factorized as follows:

$$\begin{aligned} & p(\mathbf{x}, \mathbf{c}, \mathbf{z}, \mathbf{w}|\mathbf{a_d}, \alpha_\theta, \alpha_\phi, \alpha_\varphi) \\ & = p(\mathbf{w}|\mathbf{z}, \alpha_\phi) p(\mathbf{c}|\mathbf{z}, \alpha_\varphi) p(\mathbf{z}|\mathbf{x}, \mathbf{a_d}, \alpha_\theta) p(\mathbf{x}|\mathbf{a_d}) \end{aligned} \tag{3.5}$$

To generalize the notations, let $n_a^{(b)}$ denote the number of times entity b is observed with entity a. Particularly, if an observation of topic z is made with author x, then $n_x^{(z)}$ denotes the number of times this observation is made in the whole corpus. Similarly, I define $n_z^{(t)}$, $n_z^{(c)}$ where t and c stand for term and cited author respectively. Here, I derive $p(\mathbf{c}|\mathbf{z}, \alpha_\varphi)$. Other factors can be obtained in a similar fashion. The target posterior distribution for cited author generation, i.e., $p(\mathbf{c}|\mathbf{z}, \alpha_\varphi)$, can be obtained by integrating over all possible values of φ:

$$\begin{aligned} & p(\mathbf{c}|\mathbf{z}, \alpha_\varphi) \\ & = \int \prod_{k=1}^{K} \frac{1}{\Delta(\alpha_\varphi)} \prod_{c=1}^{A} \varphi_{z,c}^{n_z^{(c)} + \alpha_\varphi^c - 1} d\varphi_z = \prod_{z=1}^{K} \frac{\Delta(\mathbf{n}_{\mathbf{z}\varphi} + \alpha_\varphi)}{\Delta(\alpha_\varphi)} \\ & \text{where } \Delta(\alpha_\varphi) = \frac{\prod_{i=1}^{dim(\alpha_\varphi)} \Gamma(\alpha_\varphi^i)}{\Gamma(\sum_{i=1}^{dim(\alpha_\varphi)} \alpha_\varphi^i)} \quad \text{and} \quad \mathbf{n}_{\mathbf{z}\varphi} = \{n_z^{(c)}\}_{c=1}^{A} \end{aligned} \tag{3.6}$$

With the likely treatment to other factors, the joint distribution can be written as:

$$\begin{aligned} & p(\mathbf{x}, \mathbf{w}, \mathbf{c}, \mathbf{z}|\alpha_\theta, \alpha_\phi, \alpha_\varphi) \\ & = \prod_{z=1}^{K} \frac{\Delta(\mathbf{n}_{\mathbf{z}\phi} + \alpha_\phi)}{\Delta(\alpha_\phi)} \prod_{z=1}^{K} \frac{\Delta(\mathbf{n}_{\mathbf{z}\varphi} + \alpha_\varphi)}{\Delta(\alpha_\varphi)} \prod_{x=1}^{A} \frac{\Delta(\mathbf{n}_{\mathbf{x}} + \alpha_\theta)}{\Delta(\alpha_\theta)} \end{aligned} \tag{3.7}$$

$$\text{where} \quad \mathbf{n}_{\mathbf{z}\phi} = \{n_z^{(t)}\}_{t=1}^{V} \text{ and } \quad \mathbf{n}_{\mathbf{x}} = \{n_x^{(z)}\}_{z=1}^{K}$$

Starting with a random assignment of topics $\mathbf{z}$ and authors $\mathbf{x}$ from the list of co-authors in a document, Gibbs sampler iterates through each word and cited authors in a document, for all the documents in the corpus. For the ALT model, I need to sample topic assignment for each word variable and cited author variable. Since I have two unobserved random variables x and z for both types of assignments, our Gibbs sampler performs blocked sampling on these two random variables. I draw a sample from $p(z_i = k, x_i = x|\mathbf{z}_{-i}, \mathbf{x}_{-i}, \mathbf{w})$ for the word variable and from $p(z_i = k, x_i = x|\mathbf{z}_{-i}, \mathbf{x}_{-i}, \mathbf{c})$ for the cited author variable. The subscript $-i$ indicates that I leave the i^{th} token out from the otherwise complete assignment. After algebraic manipulation to Eq. 3.7, I arrive at the sampling equations as given in Eq. (i & ii) in Table 3.1.

Unlike the ALT model, Author Cite Topic (ACT) model has one additional unobserved random variable c that appears inside the citation context of a given citation in any document. I initialize c from the co-authors of the cited documents by uniformly selecting one author. The remaining initializations remains the same as above. I block x, z and c while sampling and for each word in the citation context, I sample from the conditional distribution, i.e. $p(z_i = k, x_i = x, c_i = c|\mathbf{z}_{-i}, \mathbf{x}_{-i}, \mathbf{c}_{-i}, \mathbf{w})$. The algebraic form of the conditional distribution is given in Eq.(iii) in Table 3.1.

3.4 Experiments

I describe our data set and experimental settings below and, in 3.4.2 and 3.4.3, I provide the details of evaluation tasks with corresponding results.

3.4.1 Data Sets and Experimental Settings

I use two different subsets of scientific documents for our evaluation purpose. For the first dataset (referred as *CiteSeer-DS1*), I use publicly available [1] subset of the CiteSeer [2] digital library. The data set contains 3312 documents belonging to 6 different research fields and the vocabulary size is 3703 unique words. There is a total of 4132 links present in the data set. The dataset contains 4699 unique authors[3] where 1511 authors are cited. After standard preprocessing of removing stop words, I supplement the data set with the context information for each citation.

I employ CiteSeer-DS1 because various previous studies [42], [11] have used the dataset for link prediction task, however CiteSeer-DS1 is a hand-picked dataset prepared for document classification purposes [36]. For both qualitative and quantitative evaluations on a user selected scientific documents dataset in a collaborative setting, I also acquired dataset from CiteULike [4] for over 2 years from November 2005 to January 2008 (referred as *CiteSeer-DS2*). The dataset is available at http://citeulike.org. Overall, there are 33,456 distinct papers in CiteULike

[1] http://www.cs.umd.edu/s̃en/lbc-proj/LBC.html

[2] http://CiteSeer.ist.psu.edu/

[3] I use disambiguated authors for each documents available at http://CiteSeerx.ist.psu.edu/about/metadata

[4] http://citeulike.org

sample. I map the document ids of CiteULike documents to document ids of CiteSeer documents [5] to gain access to citation network of the sample. The resultant CiteSeer-DS2 contains 18354 documents in which 9571 documents are cited. There are a total of 29645 unique authors in CiteSeer-DS2 out of which 15967 authors are cited at least once. I follow the same preprocessing step as the CiteSeer-DS1 dataset.

Experimental Set-up: I choose to fix the hyper-parameters and evaluate different models with the same setting. I set the hyper-parameters to the following values [47]: $\alpha_\theta = 50/T$, $\alpha_\phi = 0.01$, $\alpha_\varphi = 0.01$. I run 1000 iterations of Gibbs sampling for training and extend the chain with 100 iterations over test set. For dynamic window selection, I collect 10 samples from the chain after every 10 iterations starting from 1000 iterations, and compute the new window with the average of the samples using Eq. 3.4. After the window update, I let the chain converge and start to update the window again. Starting with the sentence that contains the citation mention, I allow our window to grow up to a maximum of 5 sentences in either direction. The multinomial parameters of the model are calculated by taking expectations of the corresponding counts from 10 samples collected during test iterations.

3.4.2 Model Evaluation on Unseen Content

This task quantitatively estimates the generalization capabilities of a given model on unseen data. In particular, I compute the *perplexity* on the held-out test set. I run the inference algorithm exclusively on the unseen words in the test set of documents, same as [47], to obtain the log-likelihood of test documents. Before extending the Gibbs sampling chain and *sweeping* the test set, I first initialize the topic assignment to authors and unseen words randomly and run the Gibbs iteration on the test set with following Gibbs updates:

$$p(z_i^u, x_i^u | w_i^u = t, \mathbf{z^u_{-i}}, \mathbf{w^u_{-i}}, \mathbf{x^u_{-i}})$$

[5] mapping is obtained from http://citeulike.org

$$= \frac{n_{k,-i}^{(t)} + \beta}{\sum_{t=1}^{V} n_{k,-i}^{(t)} + V.\beta} . \frac{n_{x^u,-i}^{(k)} + \alpha_\theta^k}{\sum_{k=1}^{K} n_{x^u,-i}^{(k)} + K.\alpha_\theta^k} \tag{3.8}$$

Superscript $(.^u)$ stands for any unseen element. The sampling updates in Eq. 3.8 can be used to calculate the model parameters, $\Pi = (\theta, \phi, \varphi)$ for unseen documents as:

$$\theta_{x^u,k} = \frac{n_{x^u}^{(k)} + \alpha_\theta^k}{\sum_{k=1}^{K} n_{x^u}^{(k)} + K.\alpha_\theta^k}; \phi_{k,t} = \frac{n_k^{(t)^u} + n_k^{(t)} + \beta_t}{\sum_{t=1}^{V} n_k^{(t)^u} + n_k^{(t)} + \beta_t} \tag{3.9}$$

The predictive log-likelihood of a text document in the test set, given the model $\Pi = (\theta, \phi, \varphi)$, can be directly expressed as a function of the multinomial parameters:

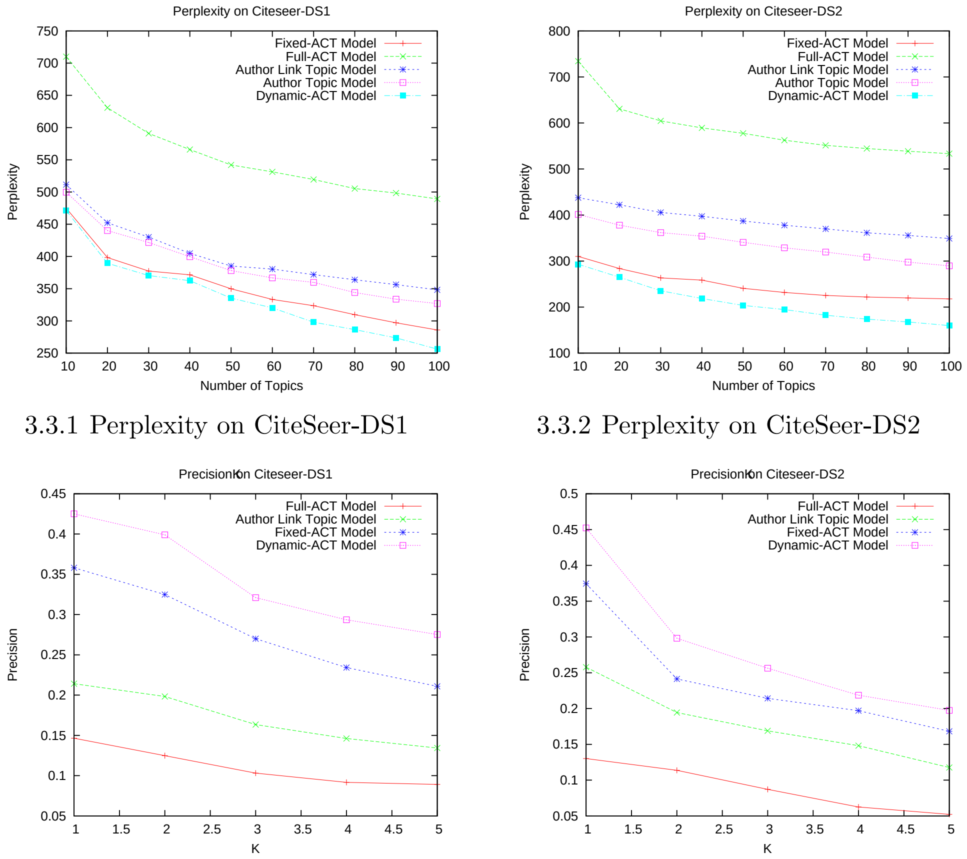
$$\begin{aligned} p(\mathbf{w}|\mathbf{\Pi}) &= \prod_{n=1}^{N_x} \sum_{k=1}^{T} \Big(\frac{1}{|a_d|} \sum_{x \in a_d} p(w_n|z_n = k).p(z_n = k|d = x) \Big) \\ &= \prod_{n=1}^{N_x} (\frac{1}{|a_d|} \sum_{k,x \in a_d} \phi_{k,t} \theta_{x,k}) \end{aligned} \tag{3.10}$$

Next, I compute the perplexity as defined below. Here, N_w is the total number of word occurrences in the test set.

$$Perplexity(\mathbf{w}) = exp(\frac{-log\ p(\mathbf{w})}{N_w}) \tag{3.11}$$

Baselines: I use following two baselines from [47] and [53], namely Author Topic Model (ATM) and Citation Author Topic Model (CAT) respectively. Since ATM does not learn from links among documents, comparison with ATM signifies the importance of learning from links along with the text of the documents. CAT model treats all the content of a citing document as context for any cited document within, therefore, comparison with CAT highlights the importance of choosing a context window surrounding the citation mention. I compare these baselines against the proposed Author Link model (ALT), fixed length window Author Cite Topic Model (Fixed-ACT) [6] and dynamically selected window based ACT model

[6] I set the radius to be 10 words from the citation mention after stop word removal, i.e., 20



3.3.1 Perplexity on CiteSeer-DS1

3.3.2 Perplexity on CiteSeer-DS2

3.3.3 P@K on CiteSeer-DS1

3.3.4 P@K on CiteSeer-DS2

Figure 3.3. Experimental results for (a) Perplexity on CiteSeer datasets DS1, (b) Perplexity on CiteSeer datasets DS2, (c) Precision @ K for cited author prediction on CiteSeer datasets DS1 and (d) Precision @ K for cited author prediction on CiteSeer datasets DS2

(Dynamic-ACT). For our experiments, the training data consists of 4 splits with 75% documents (training docs) along with the 25% words of the remaining 25% of the documents (test docs). The rest 75% words in test documents are used to calculate log-likelihood. The average value over the 4 splits are reported in the experiments.

Fig. 3.3 (a)&(b) show the comparison of perplexity on test set of CiteSeer-DS1 and CiteSeer-DS2, respectively. The ATM model performs slightly better than the ALT model. I believe that this is because the links considered separately from the content actually deteriorate the prediction capability of the models over words. In

words window

contrast, while training, links along with the content help to learn the topics better. However, when all the content is treated as context for every cited article [53] in a given citing document, the performance deteriorates significantly. Therefore, I argue that a wise selection of context window is essential when a context sensitive topic modeling approach is considered.

Dynamic-ACT outperforms all the other approaches (see Fig. 3.3 (a)&(b)). During our experiments, I observed that the length of a relatively large fraction of citation contexts was limited to a single sentence that contains the citation mention. The fraction decreases as I increase the number of topics. Specifically, for CiteSeer-DS1, 78% of the total citation contexts were composed of only one sentence when I set the number of topics to 10. This number drops to 65% with 100 topics. Also, I found the average window length on CiteSeer-DS1 to be 1.4 with 10 topics and 1.6 with 100 number of topics. I observe the similar trend with CiteSeer-DS1 where 81% of the total total citation contexts were composed of only one sentence with topic count 10 whereas the number decreases to 62% with 100 topics. Considering that the topic assignment to words is fine grained with a large number of topics, the growth outside the window is more likely to explain the finer details mentioned in the cited document.

3.4.3 Cited Author Prediction

In this task, I evaluate the capability of the models to predict the authors that this document links to. That is, given the text of a test document, which authors' work should this document cite to? The experimental design for this task is very similar to the one in the previous subsection. I again perform the Gibbs update following the sampling from conditional distribution in Eq. 3.8 and calculate the model parameters. With the model parameters for the ALT and ACT models, the probability $p(c|\mathbf{w_d})$, where c is the author to be cited given a document $\mathbf{w_d}$ is:

$$p(c|\mathbf{w_d}) = \sum_{z} p(c|z) \int_{x \in a_d} \frac{1}{|a_d|} p(z|\theta_x) d\theta_x \propto \sum_{k} \frac{1}{|a_d|} \varphi_{c,k} . \theta_{d,k} \tag{3.12}$$

Baselines: Because the ATM does not model the links, it is not possible to treat ATM as a baseline for this task. I keep all the other four comparisons intact for this task. The training data consists of 4 splits with 75% documents and their outgoing links to cited authors (training docs) and the 25% outgoing links of the remaining 25% of the documents (test docs). The rest of 75% outgoing links in the test documents are used for this task. I set the number of topic to be 100 for this task. I use Precision@K as the evaluation metric. The average value over the 4 splits are reported in the experiments.

To evaluate the prediction accuracy for the proposed models, I first label the actual authors that are cited by a test document as its relevant result set. I rank the authors in the train corpus against each test document using $p(c|\mathbf{w_d})$ and compare the models based upon the precision of the retrieved results. Fig. 3.3(c) & (d) shows the results for the three methods on CiteSeer-DS1 and CiteSeer-DS2, respectively.

3.4.4 Anecdotal Evidences

Table 3.2 shows the most likely words, interested and influential authors in 6 topics from the CiteSeer-DS2 dataset obtained using the ACT model (e.g. Griffith, Beal, etc., as interested authors and Mackay, Ghahramani and Hinton as influential authors in Bayesian learning). For each topic shown in the table, most influential authors are well known in their respective areas and their authored papers gets cited in the respective fields.

3.5 Summary

I propose novel models for author-author linkage conditioned on topics latent in the content of the documents. I exploit the citations between documents to infer influence of certain authors over topics. I also propose context sensitive extensions of the proposed model that incorporates the context of the cited document and how it infers the topic of both cited and citing authors with better quality

Topic-6					
Top Words		Top interested authors		Top influential authors	
scale	0.01663	k. mikolajczyk	0.95714	c. schmid	0.08392
shape	0.01434	j. ponce	0.95641	j. malik	0.07407
object	0.01385	t. lindeberg	0.95619	d. g. lowe	0.06075
images	0.01069	s. lazebnik	0.95412	s. belongie	0.04564
matching	0.01000	r. fergus	0.86715	k. mikolajczyk	0.03996
recognition	0.00846	a. c. berg	0.86306	j. puzicha	0.03863
features	0.00772	g. loy	0.85624	j. shi	0.02436
local	0.00751	e. rosten	0.04269	d. p. huttenlocher	0.01704
Topic-97					
Top Words		Top interested authors		Top influential authors	
retrieval	0.03634	s.-fu chang	0.04901	j. r. smith	0.05583
images	0.01635	s. mehrotra	0.04856	t. s. huang	0.04141
texture	0.01572	r. paget	0.04451	y. rui	0.03214
color	0.01184	j. z. wang	0.04399	r. jain	0.03025
features	0.01016	m. ortega	0.04214	a. efros	0.02561
content	0.00958	p. harrison	0.04118	t. leung	0.02385
search	0.00763	g. wiederhold	0.04098	w.-ying	0.01933
visual	0.00754	r. peteri	0.04058	j. malik	0.01732
Topic-45					
Top Words		Top interested authors		Top influential authors	
learning	0.02424	xiaoli li	0.04352	t. mitchell	0.09649
classification	0.02189	k. nigam	0.04203	k. nigam	0.08227
text	0.01635	t. mitchell	0.04146	a. mccallum	0.05819
training	0.01420	a. mccallum	0.04076	a. blum	0.05808
unlabeled	0.01351	yang dai	0.04031	d. d. lewis	0.04469
examples	0.01150	andrew ng	0.03843	s. thrun	0.03260
set	0.00913	r. gilleron	0.03619	ken lang	0.02693
Topic-71					
Top Words		Top interested authors		Top influential authors	
model	0.01829	t. l. griffiths	0.04683	d. j.c. mackay	0.07512
data	0.01164	m. j. beal	0.04588	z. ghahramani	0.06245
learning	0.00817	z. ghahramani	0.04376	g. e. hinton	0.04727
bayesian	0.00791	b. j. frey	0.04345	l. r. rabiner	0.03903
mixture	0.00773	d. m. blei	0.04263	t. hofmann	0.03840
inference	0.00689	d. j.c. mackay	0.04158	c. e. rasmussen	0.03226
distribution	0.00657	r. m. neal	0.04147	r. m. neal	0.02999
Topic-46					
Top Words		Top interested authors		Top influential authors	
algorithms	0.01376	e. zitzler	0.04734	d. e. goldberg	0.06921
quantum	0.01087	k. deb	0.04655	k. deb	0.06606
genetic	0.01043	k. sastry	0.04552	p. j. fleming	0.05680
optimization	0.00847	t. goel	0.04523	c. m. fonseca	0.04943
objective	0.00792	l. thiele	0.04520	n. srinivas	0.04930
pareto	0.00713	l. barbulescu	0.04503	k. l. clarkson	0.03059
population	0.00708	d. aharonov	0.04448	l. k. grover	0.02802
evolutionary	0.00658	k. svozil	0.04429	j. horn	0.02654

Table 3.2. Top words, interested authors and influential authors for 6 topics in CiteSeer-DS2

Chapter 4

Topic Models for Entity Disambiguation in Document Networks

Disambiguating entity references by annotating them with unique ids from a catalog is a critical step in the enrichment of unstructured content. In this chapter, we show that topic models, such as *Latent Dirichlet Allocation* (LDA) and its hierarchical variants, form a natural class of models for learning accurate entity disambiguation models from crowd-sourced knowledge bases such as Wikipedia. Our main contribution is a semi-supervised hierarchical model called *Wikipedia-based Pachinko Allocation Model* (WPAM) that exploits: (1) All words in the Wikipedia corpus to learn word-entity associations (while existing approaches only use words in a small fixed window around annotated entity references in Wikipedia pages), (2) Wikipedia annotations to appropriately bias the assignment of entity labels to annotated (and co-occurring unannotated) words during model learning, and (3) Wikipedia's category hierarchy to capture co-occurrence patterns among entities. We propose a new sampling algorithm to speed up model learning when topics are organized in a hierarchy, and a scheme for pruning spurious nodes from Wikipedia's crowd-sourced category hierarchy. Finally, in experiments with multiple real-life datasets, we show that WPAM outperforms state-of-the-art baselines by as much as 22% in terms of disambiguation accuracy.

4.1 Introduction

Even though the world wide web is a veritable knowledge base for everything under the sun, a large chunk of it still exists in the form of unstructured text generated with little or no curation. One important step to instilling structure into such free-form text is to collect all references to entities within it and annotate them with ids from an existing catalog of entities. Doing this will not only enable merge operations with other pieces of similarly annotated text but also promises to aid semantic search, information extraction and integration, document classification, and a host of other applications. For instance, if we know that a user is browsing a page about Michael Jordan the basketball player, then we can show the user additional articles related to only Michael Jordan the sportsperson and not Michael I. Jordan the machine learning researcher.

The crucial task in annotating entity references is to decide which entity from the catalog a particular reference is associated with. In this chapter, we refer to this problem as the *entity disambiguation* problem [27].

4.1.1 Leveraging Wikipedia for Entity Disambiguation

With over 3.4 million crowd-sourced entities, Wikipedia[1] is clearly a formidable resource, one that can serve as a comprehensive reference catalog for large-scale entity disambiguation. Each Wikipedia entity has a separate page, and a vast network of internal links annotate words in the body of pages with entities that they refer to. The copious annotations constitute valuable training data, and prior work [10, 37, 14, 39, 30] has used them to learn models for disambiguating entities at scale. Thus, Wikipedia has enabled a paradigm of *weak supervision* where freely available annotations are used to train machine-learned models.

One way to broadly categorize the various entity disambiguation approaches proposed in the literature is based on the sources of evidence that they utilize. The *local context* of a reference forms one major source of evidence used in previous work [10, 37, 14, 30]. More precisely, words that appear in the vicinity of a reference often help to decide which entity is being referred to. For instance, there exist a number of Columbus's, e.g., explorer, film director, etc., but if the words "ocean"

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

or “ship” appear in the vicinity then that should increase our belief that the reference is to the explorer. Existing approaches use local context evidence for entity disambiguation in two steps. First, they learn the context for each entity from the local context of annotated references to the entity (embedded in other Wikipedia pages). Subsequently, to resolve a reference, they compare the local context of the reference with the (learned) context for each candidate entity to determine their compatibility.

Co-occurrence patterns form yet another major source of evidence used in prior work [14, 39, 30] where the hypothesis is that certain entities often appear together and disambiguating one reference should help to decide which entities the other references within the same document are referring to. For example, there are eight Michael Jordan’s (basketball player, researcher, actor, etc.) and three Charles Barkley’s (basketball player, politician, etc.) in Wikipedia. But if a document contains mentions of both Michael Jordan and Charles Barkley, then we can be fairly certain that both references are to basketball players.

Of the existing disambiguation approaches, [30] exploits the two above-mentioned sources of evidence the most. Specifically, it attempts to annotate references with entities so that the sum of the local context compatibility for the entities and the co-occurrence between entity pairs is maximized.

4.1.2 Shortcomings of Existing Approaches

A general shortcoming of existing Wikipedia-based entity disambiguation approaches [10, 37, 14, 39, 30] is that they are somewhat ad hoc in the manner in which they combine and compute the various sources of evidence. For instance, local context compatibility and entity co-occurrence are calculated using entirely different mechanisms and may have disparate value ranges, yet they are combined in [30] by simply summing the two quantities together.

Similarly, when computing local context compatibility, prior research works have used a range of window sizes around each reference for determining the words that are a part of its local context. [10] picked an “optimum” length of 55 words centered around the entity reference. On the other hand, [37] used a window of 3 words to the left and right of the reference, set through cross-validation. And [30] used a window but do not specify its length. Clearly, there is a lack of consensus on the best setting for the window size parameter – a window size that is too small can cause important words to be excluded from the local context while irrelevant words may become part of the context with too large a window size.

Furthermore, even though freely available knowledge bases such as Wikipedia contain a large number of annotations, they are still at best, only *partially* annotated datasets. (Usually, only the first occurrence of a relevant reference on a page is annotated as advocated by Wikipedia's manual of style[2].) Existing disambiguation methods completely ignore un-annotated references to an entity in a Wikipedia page, and so the local context of these references is not included in the context for the entity. Thus, by ignoring un-annotated parts, we risk losing out on learning likely word-entity associations that might prove crucial for disambiguation performance. Note here that, many un-annotated references are indirect references like pronouns, or short forms like "JFK" for "John F. Kennedy" – these are generally difficult to resolve.

Finally, prior work [30] has also proposed using human-annotated training datasets to train classifiers for computing local context compatibility. A general problem with such (strongly) supervised approaches is the substantial amount of human effort required to create training datasets that are representative of a wide variety of document types (e.g., news articles, research reports) and domains (e.g., sports, health).

One way to broadly categorize the various entity disambiguation approaches proposed in the past is based on the sources of evidence they utilize. The *content* available in the text forms one major source of evidence used in prior work [10, 37, 14, 39, 30]. More precisely, words that appear in the vicinity of a reference often helps decide which entity is being referred to. For instance, there exist a number of Columbus's, e.g., explorer, film director etc., but if the words "ship" or "ocean" appear in its vicinity then that should increase our belief for it being a reference to the explorer. *Co-occurrence patterns* form another major source of evidence used in prior work [14, 39, 30] where the hypothesis is that certain entities often appear together and disambiguating one reference should help decide which entities the other references within the same document are referring to.

[2] http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style

Another major component aiding the development of large-scale entity disambiguation is the use of *weak supervision*. Prior work [30] has proposed using human-annotated training datasets to learn disambiguation models. Besides the standard issues involved with such an approach such as requiring substantial human effort, for applications such as entity disambiguation that may require new entities to be added to the catalog, it is entirely possible that one might need to re-train existing models and learn new models frequently. Weak supervision [10, 37, 39] on the other hand, denotes the use of prior annotations freely available in many large-scale, crowd-sourced knowledge bases such as Wikipedia[3]. Weak supervision holds the promise to be able to incorporate new entities into the catalog at almost no extra cost.

4.1.3 Problem Definition

We denote the set of all Wikipedia entity pages by $\mathcal{W}$. Consider an arbitrary set of documents $\mathcal{D}$. These can be news articles, product or restaurant reviews, blog posts, or crawled web pages. The entity disambiguation problem is to label the entity references in documents from $\mathcal{D}$ with Wikipedia entities in $\mathcal{W}$. In this chapter, we assume that entity references are given to us. Thus, we do not address the problem of identifying entity references in text which is a separate problem that can be handled using named-entity recognizers as described in [14].

4.1.4 Our Contributions

Our main contribution is a *weakly semi-supervised hierarchical topic model* for entity disambiguation called *Wikipedia-based Pachinko Allocation Model* (WPAM). WPAM is a hierarchical variant of the popular *Latent Dirichlet Allocation* (LDA) [5] topic model, and is inspired by the *Pachinko Allocation Model* (PAM) recently proposed in [32]. However, unlike PAM which is completely unsupervised, WPAM is semi-supervised. WPAM extensively leverages Wikipedia pages, annotations, and category information to provide a form of weak supervision when training models. To the best of our knowledge, our work is the first to apply topic models in conjunction with Wikipedia for large-scale entity disambiguation.

[3] `http://www.wikipedia.org`

There is a body of work [2, 8, 49] that employs topic models for entity resolution but, like PAM, these are also completely unsupervised and do not exploit Wikipedia resources.

Topic models such as LDA posit that each document is a mixture of latent dimensions or *topics*, and they achieve this by associating with each word in the document a unique topic variable. There are two key ideas underlying our use of topic models for disambiguating entities: (1) We associate with each Wikipedia entity a unique topic, and (2) We learn models on the Wikipedia corpus $\mathcal{W}$ and use the learned model to label words in $\mathcal{D}$. This way, by annotating words with topics, we in fact annotate words with entities, thus achieving disambiguation.

Topic models provide a principled approach to entity disambiguation.

Unlike existing disambiguation schemes (described earlier), topic models are oblivious to window size settings – their internal machinery naturally selects words that frequently co-occur with each entity (across the entire document corpus) to learn word-entity mappings. The selected words can be from anywhere within a document including the neighborhood of un-annotated references. Thus, topic models can learn word-entity associations of a higher quality compared to previous approaches, and this can boost disambiguation accuracy.

Even though earlier topic models like LDA relied primarily on co-occurrences at the word level for labeling, later *hierarchical* versions such as PAM are capable of capturing not only word-entity associations but also co-occurrence patterns among entities. A hierarchical topic model takes as input a topic hierarchy and annotates each word with a root-to-leaf path as opposed to topics. The model induces a clustering effect among the *topic paths* used to annotate a document's words with a preference for paths that share subpaths beginning at the root. Consequently,

with a topic hierarchy that has entities at the leaf level, the model labels a document's words with semantically related and co-occurring entities that are close by in the topic hierarchy. Thus, hierarchical topic models such as PAM and WPAM allow diverse sources of evidence like entity context and co-occurrence patterns to be combined in a single unified framework for entity disambiguation.

WPAM uses the Wikipedia category hierarchy[4] as the topic hierarchy to disambiguate entities with hierarchical topic models. Wikipedia's category hierarchy

[4] `http://en.wikipedia.org/wiki/Special:Categories`

has a DAG structure with entities forming the leaves and each entity or category assigned to (one or more) parent categories. Furthermore, semantically related entities (that are likely to occur together in documents) are grouped under one or more relevant categories. For instance, Michael Jordan and Charles Barkley are both assigned to the category "African American basketball players" (among a number of others) which is a sub-category of "American basketball players" and "African-American sportspeople".

Thus, WPAM uses a flexible, semantically rich topic hierarchy that captures entity correlations much better compared to the rigid hierarchies with a fixed number of levels used by different PAM variants (e.g., four-level PAM [32]).

Furthermore, PAM is completely unsupervised and relies on co-occurrence relationships among words and entities to assign topics to words. However, with millions of fine-grained topics, one per entity, the space of possible word-topic assignments is enormous and this can easily confound PAM. Thus, a key challenge here is to be able to guide the topic models so that they annotate words with the correct entities. To this end, we develop *weakly semi-supervised* techniques that exploit Wikipedia's crowd-sourced annotations for WPAM model learning. During learning, when selecting an entity (leaf of a topic path) to label a word, our techniques introduce a bias in favor of entities that frequently appear in annotations for either the word or the document containing it. This bias originating from annotated words also spreads to co-occurring un-annotated words, and recursively through them to more words. Thus, since un-annotated portions of documents also play a role in propagating entity labels, WPAM's learning techniques are semi-supervised.

Finally, learning the WPAM model involves sampling topic paths from the Wikipedia category hierarchy and assigning these sampled paths to words. Since the Wikipedia hierarchy is a DAG containing tens of thousands of categories and millions of entities, naive sampling strategies that enumerate each individual path are not scalable. We develop an efficient path sampling algorithm whose running time is proportional to the number of edges in the hierarchy as opposed to the number of paths which can be significantly higher.

Also, due to the lack of curation and the crowd-sourced manner in which it is produced, the Wikipedia hierarchy contains some spurious categories that can mislead our WPAM model. For example, Michael Jordan is listed under the categories “Living People” and “1963 Births”. We develop techniques to prune such irrelevant categories containing uncorrelated entities which co-occur infrequently.

To gauge the effectiveness of our topic model-based entity disambiguation approach, we conduct an extensive experimental study using two real-life datasets as the ground truth: (1) Held-out subset of annotated Wikipedia data, and (2) Pre-annotated news articles from New York Times (NYT). Our experimental results indicate that even with a semi-supervised version of LDA that leverages Wikipedia annotations, we are able to obtain an impressive disambiguation accuracy of 81% compared to 59% for the current state-of-the-art method of [30]. The accuracy gains are even higher for WPAM that exploits Wikipedia’s category hierarchy as well, with disambiguation accuracy numbers reaching 81%.

The rest of the chapter is organized as follows. In sec:lda, we develop weakly semi-supervised techniques for LDA and use Wikipedia annotations to learn accurate models of disambiguation based on entity context alone. In sec:pam, we develop weakly semi-supervised techniques for our WPAM hierarchical topic model that exploits both Wikipedia annotations and its category hierarchy to learn accurate models of disambiguation based on both entity context and co-occurrence.

In sec:experiments, we experiment with real-life Wikipedia and NYT datasets to demonstrate the efficacy of our proposed techniques. We take a closer look at relevant related work in Section 4.5, and sec:conclusion concludes the chapter.

4.2 Weakly Semi-Supervised Topic Model

Topic models represent each document as a mixture of (latent) topics, where each topic is a probability distribution over words. The document-topic and topic-word distributions are learned automatically from the data in an unsupervised manner with no human labeling or prior knowledge required.

Our entity disambiguation models build upon the popular LDA topic model which has been extensively used for classification of short text segments [44], unsupervised entity and author resolution [2, 8, 49], prediction of movie ratings from reviews [4], and extraction of ratable aspects of objects from online user reviews [52]. In this section, we first show how we can use LDA to assign entity labels to document words by mapping each entity to a separate topic. We then present techniques that leverage Wikipedia annotations to bias prior document-topic and topic-word distributions. The result is a semi-supervised version of the LDA model which is different from the unsupervised models previously used for entity resolution [2, 8, 49].

4.2.1 Latent Dirichlet Allocation

LDA is a probabilistic generative model that assumes Dirichlet priors on document-topic and topic-word distributions. Consider a collection of M documents containing words from the vocabulary of terms $\{1, \ldots, T\}$, and let $\{1, \ldots, K\}$ be a set of topics. The LDA model is defined by two parameters: (1) the multinomial distribution $\vec{\theta}_m = P(z|d = m)$ over topics for each document m, and (2) the multinomial distribution $\vec{\phi}_k = P(w|z = k)$ over words for each topic k. LDA's document generation process is as follows:

- For each topic k, sample word distribution $\vec{\phi}_k \sim \text{Dir}(\vec{\beta})$.
- For each document m
  - Sample topic distribution $\vec{\theta}_m \sim \text{Dir}(\vec{\alpha})$.
  - For each word w_i in document m
    - Sample a topic $z_i \sim \text{Mult}(\vec{\theta}_m)$.
    - Sample a word $w_i \sim \text{Mult}(\vec{\phi}_{z_i})$.

Above, $\text{Dir}(\vec{\alpha})$ and $\text{Dir}(\vec{\beta})$ are Dirichlet distributions with hyper-parameters $\vec{\alpha}$ and $\vec{\beta}$, respectively. And $\text{Mult}(\vec{\theta}_m)$ and $\text{Mult}(\vec{\phi}_{z_i})$ are multinomial distributions. In the following subsections, we will use $\overrightarrow{w}$ to denote the vector of words contained in the documents, and $\overrightarrow{z}$ to denote the corresponding topics for the words.

4.2.2 Inference Using Gibbs Sampling

The problem of statistical inference involves estimating the probability distribution $\vec{\phi}_k$ over words associated with each topic k, the distribution over topics $\vec{\theta}_m$ for each document m, and often, the topic responsible for generating each word. Instead of directly estimating $\vec{\phi}_k$ and $\vec{\theta}_m$, we use the approach of [20] that first constructs the posterior distribution $P(\vec{z}|\vec{w})$ and then estimates $\vec{\phi}_k$ and $\vec{\theta}_m$ from this posterior distribution.

To efficiently estimate the posterior distribution, [20] uses Gibbs sampling which is a simple and widely applicable *Markov chain Monte Carlo* (MCMC) algorithm for sampling from complex high-dimensional distributions. Starting with a random topic assignment $\vec{z}$, the Gibbs sampling algorithm iterates through each word in the document corpus. In each step, the algorithm samples a topic assignment for a word w_i conditioned on the topic assignments of all other words. More formally, in each Gibbs sampling step, the algorithm replaces z_i by a topic drawn from the distribution $P(z_i|\vec{z}_{-i}, \vec{w})$, where $\vec{z}_{-i}$ is $\vec{z}$ without the i^{th} component.

For a word $w_i = t$ in document m, the conditional probability that $z_i = k$ is given by (a detailed derivation of the formula below can be found in [25]):

$$\begin{aligned} & P(z_i = k|\vec{z}_{-i}, \vec{w}, \vec{\alpha}, \vec{\beta}) \\ & \propto \frac{n_{t,-i}^{(k)} + \beta_t}{\sum_{t'=1}^{T}(n_{t',-i}^{(k)} + \beta_{t'})} \cdot \frac{n_{k,-i}^{(m)} + \alpha_k}{\sum_{k'=1}^{K}(n_{k',-i}^{(m)} + \alpha_{k'})} \end{aligned} \tag{4.1}$$

Above, $n_{t,-i}^{(k)}$ is the number of times term t is assigned topic k excluding the current assignment; similarly, $n_{k,-i}^{(m)}$ is the number of words in document m that are assigned topic k excluding the current assignment. The Gibbs sampling equation (4.1) is fairly intuitive – the first term is the probability of term t under topic k and the second term is the probability of topic k in document m. Observe that due to the second term, the topic assignment for a word is heavily influenced by the topic assignments for the remaining words in the document. Therefore, co-occurring words do influence each other's topic assignments and will very likely be assigned the same topic by the Gibbs sampling algorithm.

The sequence of samples obtained from Gibbs sampling form a Markov chain that converges to the posterior distribution $P(\vec{z}|\vec{w})$. Thus, after an initial burn-in period, we can get a representative set of samples from the distribution by collecting Gibbs samples at regularly spaced intervals. These can then be used to estimate the distributions $\vec{\phi}_k$ and $\vec{\theta}_m$ as described in [20, 25].

In our implementation, we estimate the $\vec{\alpha}$ and $\vec{\beta}$ Dirichlet hyper-parameters using Minka's fixed-point iteration [41].

4.2.3 Using LDA for Entity Disambiguation

We use LDA to develop an unsupervised algorithm for disambiguating the entity references in document set $\mathcal{D}$. Essentially, our disambiguation algorithm runs Gibbs sampling with a separate topic per Wikipedia entity. Thus, Gibbs sampling assigns entity labels to all the words in the documents, effectively disambiguating all the entity references.

Our disambiguation algorithm has two phases: *training* and *labeling*. In the training phase, we run Gibbs sampling only on the collection of Wikipedia pages $\mathcal{W}$. Let $\vec{z}_{\mathcal{W}}$ denote the topic assignments for words in $\mathcal{W}$ at the end of the training phase. Next, in the labeling phase, we run an *incremental* Gibbs sampling algorithm on $\mathcal{W} \cup \mathcal{D}$ that only samples topics for words in $\mathcal{D}$ while keeping the topics for words in $\mathcal{W}$ fixed at $\vec{z}_{\mathcal{W}}$. Let $\vec{z}_{\mathcal{D}}$ be the topic assignments for words in $\mathcal{D}$. Then, in each incremental Gibbs sampling step during the labeling phase, only $\vec{z}_{\mathcal{D}}$ changes but $\vec{z}_{\mathcal{W}}$ stays constant. A topic for word w_i in $\mathcal{D}$ is sampled according to the Gibbs sampling equation (4.1). Note that the count $n_{t,-1}^{(k)}$ in the equation is the total number of times term t is assigned topic k across $\vec{z}_{\mathcal{D}}$ and $\vec{z}_{\mathcal{W}}$.

A major scalability challenge is that with a separate topic per Wikipedia entity, the number of topics K is equal to the number of entities in Wikipedia which can be quite large. As a result, in each Gibbs sampling step, the cost of sampling a topic assignment for w_i from its conditional distribution can be prohibitive since we need to compute the conditional probabilities for all K topics according to Equation (4.1). The key observation we make here is that in general only a few entities are relevant to each document. And these are typically entities with a surface form that matches a keyword in the document. (The surface forms for an entity are the anchor text appearing within links to the entity embedded in Wikipedia pages.) Thus, for each document, we identify the entities with matches, and only consider topics corresponding to these entities when sampling a topic assignment for a word in the document. This simple optimization results in a substantial speedup, enabling us to scale to lots of topics.

4.2.4 Weakly Semi-Supervised LDA

The standard LDA topic model is completely unsupervised, and determines topics for words entirely based on the co-occurrence patterns of words across documents. However, with millions of fine-grained topics, the space of topic assignments to words is vast, and this makes the task of finding a "good" topic assignment extremely challenging.

In this section, we leverage the prior Wikipedia annotations of words with entities to provide a form of weak supervision to the basic LDA model with the objective of improving its labeling accuracy. Let $\mathcal{A}$ denote the portions of Wikipedia pages containing only annotated words along with their entity labels. The key idea underlying our approach is to bias the topic-word distributions $\vec{\phi}_k$ in favor of words that are frequently annotated with (the entity corresponding to) topic k in $\mathcal{A}$, and the document-topic distributions $\vec{\theta}_m$ in favor of topics (corresponding to entities) that frequently occur in document m's annotations in $\mathcal{A}$.

We consider the Wikipedia annotations $\mathcal{A}$ as multinomial observations that bias the distributions of parameters $\vec{\theta}_m$ and $\vec{\phi}_k$. Recall that $\vec{\theta}_m$ and $\vec{\phi}_k$ have Dirichlet priors with hyper-parameters $\vec{\alpha}$ and $\vec{\beta}$, respectively. The posterior distribution of $\vec{\phi}_k$ conditioned on annotations $\mathcal{A}$ is given by:

$$
\begin{aligned}
P(\vec{\phi}_k|\vec{\beta},\mathcal{A}) &= \frac{\prod_{w_i\in\mathcal{A},z_i=k} P(w_i|\vec{\phi}_k)\cdot P(\vec{\phi}_k|\vec{\beta})}{\int\prod_{w_i\in\mathcal{A},z_i=k} P(w_i|\vec{\phi}_k)\cdot P(\vec{\phi}_k|\vec{\beta})\cdot d\vec{\phi}_k} \\
&= \frac{\prod_{t=1}^{T}\phi_{kt}^{\delta_t^{(k)}}\cdot\frac{\Gamma(\sum_{t=1}^{T}\beta_t)}{\prod_{t=1}^{T}\Gamma(\beta_t)}\cdot\prod_{t=1}^{T}\phi_{kt}^{\beta_t-1}}{\int\prod_{t=1}^{T}\phi_{kt}^{\delta_t^{(k)}}\cdot\frac{\Gamma(\sum_{t=1}^{T}\beta_t)}{\prod_{t=1}^{T}\Gamma(\beta_t)}\cdot\prod_{t=1}^{T}\phi_{kt}^{\beta_t-1}\cdot d\vec{\phi}_k} \\
&= \frac{\prod_{t=1}^{T}\phi_{kt}^{\delta_t^{(k)}+\beta_t-1}}{\int\prod_{t=1}^{T}\phi_{kt}^{\delta_t^{(k)}+\beta_t-1}\cdot d\vec{\phi}_k} \\
&= \frac{\Gamma(\sum_{t=1}^{T}(\delta_t^{(k)}+\beta_t))}{\prod_{t=1}^{T}\Gamma(\delta_t^{(k)}+\beta_t)}\cdot\prod_{t=1}^{T}\phi_{kt}^{\delta_t^{(k)}+\beta_t-1} \\
&= \mathrm{Dir}(\vec{\phi}_k/\vec{\beta}+\vec{\delta}^{(k)})
\end{aligned}
$$

Above, $\delta_t^{(k)}$ is the number of times term t is assigned topic k in $\mathcal{A}$. Thus, $\vec{\phi}_k$

has a Dirichlet posterior with hyper-parameters $\vec{\beta} + \vec{\delta}^{(k)}$. Similarly, we can show that $\vec{\theta}_m$ has a Dirichlet posterior with hyper-parameters $\vec{\alpha} + \vec{\delta}^{(m)}$. Here $\delta_k^{(m)}$ is the number of annotated words in document m that are assigned topic k in $\mathcal{A}$. Note that $\delta_k^{(m)}$ is zero for documents in $\mathcal{D}$ corresponding to non-Wikipedia pages since they do not contain any annotations.

Thus, in our weakly semi-supervised LDA model, the document generation process draws each $\vec{\phi}_k$ from a Dirichlet distribution with hyper-parameters $\vec{\beta} + \vec{\delta}^{(k)}$ (instead of $\vec{\beta}$), and each $\vec{\theta}_m$ from a Dirichlet distribution with hyper-parameters $\vec{\alpha} + \vec{\delta}^{(m)}$ (instead of $\vec{\alpha}$). Substituting the posterior distributions for $\vec{\phi}_k$ and $\vec{\theta}_m$ in the derivation of the conditional distribution equations, we finally get that the new conditional probability that $z_i = k$ (for word $w_i = t$ in document m) given the observed Wikipedia annotations $\mathcal{A}$ is:

$$\begin{aligned} & P(z_i = k | \vec{z}_{-i}, \vec{w}, \vec{\alpha}, \vec{\beta}, \mathcal{A}) \\ & \propto \frac{n_{t,-i}^{(k)} + \beta_t + \delta_t^{(k)}}{\sum_{t'=1}^{T}(n_{t',-i}^{(k)} + \beta_{t'} + \delta_{t'}^{(k)})} \cdot \frac{n_{k,-i}^{(m)} + \alpha_k + \delta_k^{(m)}}{\sum_{k'=1}^{K}(n_{k',-i}^{(m)} + \alpha_{k'} + \delta_{k'}^{(m)})} \end{aligned} \tag{4.2}$$

Thus, we incorporate the knowledge of Wikipedia annotations in the new Gibbs sampling equation (4.2) above by adding the prior counts $\delta_t^{(k)}$ and $\delta_k^{(m)}$ observed in $\mathcal{A}$ to the counts $n_{t,-i}^{(k)}$ and $n_{k,-i}^{(m)}$, respectively. This has the effect of biasing the topic assignment for a word (in each Gibbs sampling step) in favor of topics that frequently appear in annotations $\mathcal{A}$ for either the word or the document containing it. Furthermore, this topic bias spreads to other occurrences of the word in the document corpus as well as to co-occurring words, and recursively through them to more words. These words that propagate topic assignment biases can be unannotated words not contained in $\mathcal{A}$, and hence our LDA model with biased $\vec{\phi}_k$ and $\vec{\theta}_m$ distributions is semi-supervised. Note that this bias propagation can be quite powerful in practice; in fact, previous research on semi-supervised learning [38, 43] has shown that if we can avail of large amounts of unlabeled data and combine that with the labeled training data then the accuracy of machine-learned models can be drastically improved.

4.3 Weakly Semi-Supervised Hierarchical Topic Model

LDA selects topics based entirely on the correlations among words, but it does not explicitly model correlations among topics. However, real-world documents are generally topically coherent, and so a model that can capture topic correlations as well can lead to higher disambiguation accuracy. As we pointed out earlier in Section 4.1, exploiting topic correlations can help to resolve an ambiguous reference to Michael Jordan to the basketball player (as opposed to the researcher) if the document also contains a reference to Charles Barkley the basketball player. LDA may have difficulty disambiguating such references correctly because it can combine arbitrary sets of topics.

Hierarchical topic models extend LDA to capture both word and topic correlations in a single unified framework. A hierarchical model takes as input a topic hierarchy which can be an arbitrary DAG over topics with interior nodes representing a correlation among child topics. It assigns each word a root-to-leaf topic path with a preference for annotating a document's words with overlapping paths that share subpaths starting at the root. Thus, the topics for each document are close by in the topic hierarchy, and therefore highly correlated.

In Section 4.3.1, we describe our *weakly semi-supervised hierarchical topic model* which we call *Wikipedia-based Pachinko Allocation Model* (WPAM). WPAM builds on the *Pachinko Allocation Model* (PAM) proposed in [32] but differs from the work of [32] in two important respects. First, [32] focuses on a fixed four-level topic hierarchy – a problem with such a rigid structure is that it may not accurately model real-world topic correlations. In contrast, WPAM leverages the Wikipedia category hierarchy that has a flexible DAG structure with a different number of levels in different regions of the hierarchy. The Wikipedia hierarchy groups semantically related entities (and categories) under one or more relevant categories – thus, it is quite effective at capturing topic correlations. A second key difference between PAM and WPAM is that PAM is unsupervised. WPAM, on the other hand, is a weakly semi-supervised model that uses Wikipedia annotations as training data to bias topic assignments.

The Wikipedia hierarchy contains over 0.34 million categories, and so learning models with such a big hierarchy poses a serious challenge. In Section 4.3.2, we present an algorithm for efficiently sampling paths from large topic hierarchies in each Gibbs sampling step. Also, since Wikipedia's category hierarchy is generated by human volunteers, it is not perfect and contains irrelevant categories like "Living People" and "People with Year of Birth Missing". In Section 4.3.3, we describe a scheme for cleansing the hierarchy of such extraneous category nodes.

4.3.1 Wikipedia-based Pachinko Allocation Model

WPAM uses Wikipedia's category hierarchy as the topic hierarchy which we denote by H. Non-leaf topics in H are the Wikipedia categories, and leaf topics correspond to Wikipedia entities. We denote the set of children of a non-leaf topic k in H by $c(k)$. WPAM assigns a root-to-leaf path $z_i = \langle z_{i1}, z_{i2}, \ldots, z_{il_i} \rangle$ from H to each word w_i in the document corpus. Here, z_{i1} is always the root, and topic $z_{i(j+1)}$ is a child of z_{ij}. The WPAM model is defined by the following parameters: (1) For each document m and non-leaf topic k, a multinomial distribution $\vec{\theta}_{mk}$ over k's children $c(k)$, and (2) For each leaf topic k, a multinomial distribution $\vec{\phi}_k$ over words.

The model parameters $\vec{\theta}_{mk}$ have a Dirichlet prior with hyper-parameters $\vec{\alpha}_k$, and the parameters $\vec{\phi}_k$ have a Dirichlet prior with hyper-parameters $\vec{\beta}$. As shown in Section 4.2.4, with the Wikipedia annotations $\mathcal{A}$, $\vec{\phi}_k$ has a Dirichlet posterior with hyper-parameters $\vec{\beta} + \vec{\delta}^{(k)}$ which we denote by $\vec{\beta}^{(k)}$ (recall that $\delta_t^{(k)}$ is the number of times term t is assigned topic k in $\mathcal{A}$). Similarly, if $\delta_{kk'}^{(m)}$ is the number of topic paths in document m in $\mathcal{A}$ containing k' as a subtopic of k, then $\vec{\theta}_{mk}$ can be shown to have a Dirichlet posterior with hyper-parameters $\vec{\alpha}_k + \vec{\delta}_k^{(m)}$ which we denote by $\vec{\alpha}_k^{(m)}$.

A slight problem we face here is that Wikipedia annotations only specify entities and not paths in the topic hierarchy H. Furthermore, since H is a DAG, there can be multiple paths in H from the root to the leaf topic corresponding to an entity. This makes it difficult to obtain exact $\delta_{kk'}^{(m)}$ counts for topic, subtopic pair occurrences in $\mathcal{A}$. So we approximate $\delta_{kk'}^{(m)}$ as follows. First, for each annotation $a \in \mathcal{A}$ in document m, we compute the fraction of paths in H from the root to the entity specified in a that pass through topic k and its subtopic k'. The fraction essentially represents the probability that a topic path for annotation a passes through the topic, subtopic pair (k, k'). Our $\delta_{kk'}^{(m)}$ estimate is then the sum of the fraction values for all the annotations in document m. Note that we can compute the number of paths between any two nodes of a DAG efficiently using dynamic programming, and this can be used to calculate the path fraction value for each annotation.

WPAM uses the Dirichlet posterior distribution for each $\vec{\phi}_k$ and $\vec{\theta}_{mk}$ to generate documents, and so it is a semi-supervised model. The generative process for a document collection under WPAM is as follows:

- For each leaf topic k, sample word distribution $\vec{\phi}_k \sim \text{Dir}(\vec{\beta}^{(k)})$. Here, $\vec{\beta}^{(k)} = \vec{\beta} + \vec{\delta}^{(k)}$.

- For each document m

  - For each non-leaf topic k, sample topic distribution $\vec{\theta}_{mk} \sim \text{Dir}(\vec{\alpha}_k^{(m)})$. Here, $\vec{\alpha}_k^{(m)} = \vec{\alpha}_k + \vec{\delta}_k^{(m)}$.

  - For each word w_i in document m

    * Sample a root-to-leaf path $z_i = \langle z_{i1}, z_{i2}, \ldots, z_{il_i} \rangle$ of length l_i from the topic hierarchy H. For each topic z_{ij} in the topic path, sample child $z_{i(j+1)} \sim \text{Mult}(\vec{\theta}_{mz_{ij}})$.

    * Sample a word $w_i \sim \text{Mult}(\vec{\phi}_{z_{il_i}})$.

Following this process, the joint probability of generating the observed words $\vec{w}$ and the topic path assignments $\vec{z}$ can be computed by integrating out the $\vec{\phi}_k$'s and the $\vec{\theta}_{mk}$'s. Ignoring constants, we get

, and the multinomial distributions $\Phi = \{\vec{\phi}_k\}$ and $\Theta = \{\vec{\theta}_{mk}\}$ is

$$
\begin{aligned}
&P(\vec{z}, \vec{w}, \Phi, \Theta / \{\vec{\alpha_k}\}, \vec{\beta}, \mathcal{A}) \\
&\quad = \prod_{m=1}^{M} \prod_{k=1}^{K'} P(\vec{\theta}_{mk} / \vec{\alpha}_k, \mathcal{A}) \cdot \prod_{w_i \in m} \prod_{j=1}^{l_i - 1} P(z_{i(j+1)} / \vec{\theta}_{mz_{ij}}) \cdot \\
&\quad \prod_{k=K'+1}^{K} P(\vec{\phi}_k / \vec{\beta}, \mathcal{A}) \cdot \prod_{w_i} P(w_i / \vec{\phi}_{z_{l_i}})
\end{aligned}
$$

$$= \prod_{m=1}^{M}\prod_{k=1}^{K'} \frac{\Gamma(\sum_{k'\in c(k)} \alpha_{kk'} + \delta_{kk'}^{(m)})}{\prod_{k'\in c(k)} \Gamma(\alpha_{kk'} + \delta_{kk'}^{(m)})} \cdot \prod_{k'\in c(k)} (\theta_{mkk'})^{n_{kk'}^{(m)} + \alpha_{kk'} + \delta_{kk'}^{(m)} - 1} \cdot$$
$$\prod_{k=K'+1}^{K} \frac{\Gamma(\sum_{t=1}^{T} \beta_t + \delta_t^{(k)})}{\prod_{t=1}^{T} \Gamma(\beta_t + \delta_t^{(k)})} \cdot \prod_{t=1}^{T} (\phi_{kt})^{n_t^{(k)} + \beta_t + \delta_t^{(k)} - 1}$$

The joint distribution of all observed words $\vec{w}$ and hidden topics can be computed by integrating out the $\vec{\phi}_k$'s and the $\vec{\theta}_{mk}$'s.

$$P(\vec{z}, \vec{w} | \{\vec{\alpha_k}\}, \vec{\beta}, \mathcal{A}) \tag{4.3}$$
$$= \prod_{m=1}^{M}\prod_{k=1}^{K'} \frac{\prod_{k'\in c(k)} \Gamma(n_{kk'}^{(m)} + \alpha_{kk'} + \delta_{kk'}^{(m)})}{\Gamma(\sum_{k'\in c(k)} (n_{kk'}^{(m)} + \alpha_{kk'} + \delta_{kk'}^{(m)}))} \cdot$$
$$\prod_{k=K'+1}^{K} \frac{\prod_{t=1}^{T} \Gamma(n_t^{(k)} + \beta_t + \delta_t^{(k)})}{\Gamma(\sum_{t=1}^{T} (n_t^{(k)} + \beta_t + \delta_t^{(k)}))}$$

Above, $n_t^{(k)}$ is the number of occurrences of term t with leaf topic k, and $n_{kk'}^{(m)}$ is the number of times topic k' appears as a subtopic of topic k in a topic path of document m. From the above joint distribution equation, it follows that WPAM favors topic assignments that (1) assign each leaf topic to only a small number of distinct terms (this increases the numerator value in the second term), and (2) within each document, assign topic paths containing only a few subtopics for each non-leaf topic (this increases the numerator value in the first term). The second point introduces a clustering effect among the topic paths within a document with a preference for paths with common subpaths from the root. Thus, words in a document are assigned entity labels that are close by in H, and hence correlated.

As before, we can use Gibbs sampling to compute the topic path assignments $\vec{z}$ for document words in $\vec{w}$. Essentially, Gibbs sampling draws samples from the posterior distribution $P(\vec{z}|\vec{w})$ by sequentially sampling topic paths for each z_i from $P(z_i|\vec{z}_{-i}, \vec{w})$. For a word $w_i = t$ in document m, the conditional probability that $z_i = \langle k_1, k_2, \ldots, k_l \rangle$ is given by:

$$P(z_i = \langle k_1, k_2, \ldots, k_l \rangle | \vec{z}_{-i}, \vec{w}, \{\vec{\alpha}_k\}, \vec{\beta}, \mathcal{A}) \tag{4.4}$$

$$\propto \frac{n_{t,-i}^{(k_l)} + \beta_t^{(k_l)}}{\sum_{t'=1}^{T}(n_{t',-i}^{(k_l)} + \beta_{t'}^{(k_l)})} \cdot \prod_{j=1}^{l-1} \frac{n_{k_j k_{j+1},-i}^{(m)} + \alpha_{k_j k_{j+1}}^{(m)}}{\sum_{k' \in c(k_j)}(n_{k_j k',-i}^{(m)} + \alpha_{k_j k'}^{(m)})}$$

$$P(z_i = \langle k_1, k_2, \ldots, k_l \rangle | \vec{z}_{-i}, \vec{w}, \{\vec{\alpha}_k\}, \vec{\beta}, \mathcal{A}) \propto \tag{4.5}$$

$$\frac{n_{t,-i}^{(k_l)} + \beta_t + \delta_t^{(k_l)}}{\sum_{t'=1}^{T}(n_{t',-i}^{(k_l)} + \beta_{t'} + \delta_{t'}^{(k_l)})} \cdot \prod_{j=1}^{l-1} \frac{n_{k_j k_{j+1},-i}^{(m)} + \alpha_{k_j k_{j+1}} + \delta_{k_j k_{j+1}}^{(m)}}{\sum_{k' \in c(k_j)}(n_{k_j k',-i}^{(m)} + \alpha_{k_j k'} + \delta_{k_j k'}^{(m)})}$$

Above, $n_{t,-i}^{(k_l)}$ is the number of occurrences of term t with leaf topic k_l excluding the current assignment, and $n_{k_j,k',-i}^{(m)}$ is the number of times topic k' appears as a subtopic of topic k_j in a topic path of document m excluding the current assignment. In Equation (4.5) above, the first term is the probability of term t under leaf topic k_l and the second term is the probability of path $\langle k_1, k_2, \ldots, k_l \rangle$ in document m. Observe that the second term induces a clustering effect among the topic paths of a document with a preference for paths with common subpaths from the root. Thus, words in a document are assigned entity labels that are close by in H, and hence correlated.

Our entity disambiguation algorithm first runs WPAM's Gibbs sampling on the collection of Wikipedia documents $\mathcal{W}$. In the labeling phase, it disambiguates each reference in $\mathcal{D}$ by running incremental Gibbs sampling (see Section 4.2.3) on $\mathcal{W} \cup \mathcal{D}$. Each reference is labeled with the entity at the leaf of the topic path assigned it.

4.3.2 Speeding up Gibbs Sampling

Each Gibbs sampling step in WPAM takes time proportional to the number of paths in the topic hierarchy H. This is because we need to compute conditional probabilities for all paths in order to sample a topic path from the conditional distribution $P(z_i|\vec{z}_{-i}, \vec{w})$. Notice that since the hierarchy H is an arbitrary DAG, the number of paths in it can be quite large. In fact, in the worst case, for a DAG with n nodes and depth h, the number of paths in the worst case can be $O(n^h)$. Thus, a naive strategy of enumerating all paths is not scalable for the Wikipedia hierarchy containing thousands of categories and millions of entities.

Algorithm 6: GibbsSamplingStep

```
Input: Word vector $\vec{w}$ and topic path assignments $\vec{z}$, topic hierarchy $H$,
       topic path $z_i$ to be sampled for word $w_i = t$ from document $m$;
Output: Sample from $P(z_i|\vec{z}_{-i}, \vec{w})$;
foreach leaf topic $k$ do $F(k) = \frac{n^{(k)}_{t,-i}+\beta^{(k)}_t}{\sum_{t'=1}^{T}(n^{(k)}_{t',-i}+\beta^{(k)}_{t'})}$;
foreach non-leaf topic $k$ (in bottom-up order) do
$F(k) = \sum_{k' \in c(k)}(\frac{n^{(m)}_{kk',-i}+\alpha^{(m)}_{kk'}}{\sum_{k'' \in c(k)}(n^{(m)}_{kk'',-i}+\alpha^{(m)}_{kk''})} \cdot F(k'))$;
$k_1$ = root of $H$;
$j = 1$;
while $k_j$ is not a leaf do
    Sample topic $k'$ from $k_j$'s children $c(k_j)$ with probability
    $\frac{1}{F(k_j)} \cdot \frac{n^{(m)}_{k_j k',-i}+\alpha^{(m)}_{k_j k'}}{\sum_{k'' \in c(k)}(n^{(m)}_{k_j k'',-i}+\alpha^{(m)}_{k_j k''})} \cdot F(k')$;
    $j = j + 1$;
    $k_j = k'$;
return $\langle k_1, \ldots, k_j \rangle$;
```

In this subsection, we present an efficient algorithm for sampling a path from H during each Gibbs sampling step. The key idea is to incrementally construct a path sample by recursively drawing samples from the children of topics along the path. This helps to reduce the time complexity of our algorithm to be proportional to the number of edges in H. In the worst case, the number of edges in H is $O(n^2)$ which can be a lot smaller than the worst-case number of paths $O(n^h)$. Thus, our path sampling algorithm can significantly speed up each Gibbs sampling step.

Algorithm 6 describes our procedure for sampling the topic path assignment z_i during a Gibbs sampling step. The algorithm starts by recursively computing $F(k)$ for topics k in H in a bottom-up fashion starting with the leaf topics. It is easy to see that all the $F(\cdot)$ values can be computed in time proportional to the number of edges in H. Also, observe that the value of $F(\cdot)$ for the root is equal to the sum of the conditional probability values (RHS of Equation (4.5)) for all topic paths in H.

Once all the $F(k)$'s have been computed, Algorithm 6 samples a path top-down starting from the root and recursively sampling a child for each topic in the path until a leaf is reached. We now show that Algorithm 6 samples paths from H according to WPAM's Gibbs sampling equation (4.5). It is straightforward to see that the algorithm selects root-to-leaf path $\langle k_1, \ldots, k_l \rangle$ with probability $\prod_{j=1}^{l-1} \frac{1}{F(k_j)} \cdot \frac{n^{(m)}_{k_j k_{j+1},-i} + \alpha^{(m)}_{k_j k_{j+1}}}{\sum_{k' \in c(k_j)} (n^{(m)}_{k_j k',-i} + \alpha^{(m)}_{k_j k'})} \cdot F(k_{j+1})$. $\frac{\frac{n^{-i}_{mk_1k_2} + \alpha_{k_1k_2}}{\sum_{k' \in c(k_1)} n^{-i}_{mk_1k'} + \alpha_{k_1k'}} \cdot F(k_2)}{F(k_1)} \cdots \frac{\frac{n^{-i}_{mk_{l-2}k_{l-1}} + \alpha_{k_{l-2}k_{l-1}}}{\sum_{k' \in c(k_{l-2})} n^{-i}_{mk_{l-2}k'} + \alpha_{k_{l-2}k'}} \cdot F(k_{l-1})}{F(k_{l-2})}$. $\frac{\frac{n^{-i}_{mk_1k_2} + \alpha_{k_1k_2}}{\sum_{k' \in c(k_1)} n^{-i}_{mk_1k'} + \alpha_{k_1k'}} \cdot \frac{n^{-i}_{k_lt} + \beta_t}{\sum_{t'=1}^{T} n^{-i}_{k_lt'} + \beta_{t'}}}{F(k_{l-1})}$. Notice that all the intermediate $F(k_j)$'s cancel out except for $F(k_1)$ for the root topic k_1 which is shared by all the paths in H, and $F(k_l)$ for the leaf topic k_l which is equal to $\frac{n^{(k_l)}_{t,-i} + \beta^{(k_l)}_t}{\sum_{t'=1}^{T} (n^{(k_l)}_{t',-i} + \beta^{(k_l)}_{t'})}$. Thus, we get that Algorithm 6 samples path $\langle k_1, \ldots, k_l \rangle$ according to Equation (4.5).

Note that even with our efficient path sampling algorithm, drawing a path sample from the entire Wikipedia hierarchy H may be too expensive because of its huge size. A key observation here is that each document is typically only about a few topics and entities – so we can significantly reduce the overhead of our sampling algorithm by only considering the restriction of H to topics and entities mentioned in the document. Specifically, let E_m be the set of entities with a surface form that matches a keyword in document m. Then, our restricted hierarchy H_m for document m consists of all root-to-leaf paths in H to entities in E_m. Note that H_m can be computed efficiently using simple depth-first search. Furthermore, H_m only needs to be computed once at the start of the sampling procedure (and not in each step). Now, in each Gibbs sampling step, when sampling topic path assignment z_i in document m, we use the much smaller hierarchy H_m instead of H. Specifically, in Algorithm 6, for each non-leaf topic k, the children $c(k)$ are defined with respect to the restricted hierarchy H_m and not H.

4.3.3 Pruning Noisy Categories from Hierarchy

As mentioned earlier, due to the crowd-sourced manner in which it is produced, the Wikipedia hierarchy contains some spurious categories. These categories can cause WPAM to incorrectly infer topic correlations, and assign wrong entity labels to words. In this subsection, we propose a scheme to detect such spurious non-leaf topics, and prune them from the topic hierarchy H.

Good topics are the ones whose child topics are correlated and frequently co-occur in documents. Thus, our scheme for detecting spurious topics defines a goodness measure for each topic based on the co-occurrence of its children, and then deletes the topics with low goodness scores. For a non-leaf topic k, let $S(k)$ denote the set of all entities that are reachable from k in H. Furthermore, consider the set of Wikipedia documents $\mathcal{W}$, and let $\lambda_m(e)$ be 1 if a reference to entity e appears in document $m \in \mathcal{W}$. Then, the quantity $Q(k)$ defined below captures the co-occurrence counts for entities associated with the sub-topics of topic k.

$$Q(k) = \sum_{m \in \mathcal{W}} \sum_{e_1, e_2 \in S(k)} \lambda_m(e_1) \cdot \lambda_m(e_2)$$

Essentially, $Q(k)$ counts the number of distinct occurrences of all entity pairs $e_1, e_2 \in S(k)$. Clearly, a larger value of $Q(k)$ is an indicator of higher co-occurrence among k's children. Observe that more entities are reachable from topics in H that are closer to the root; consequently, we expect that topics nearer to the root will have higher $Q(k)$ values. Thus, in order to compare co-occurrence counts for topics throughout the hierarchy, we normalize the $Q(k)$ values using the number of distinct entity occurrences for each topic k. Specifically, we divide each $Q(k)$ value by

$$R(k) = \sum_{m \in \mathcal{W}} \sum_{e \in S(k)} \lambda_m(e)$$

Thus, our goodness measure that captures co-occurrence counts for topic k is given by $G(k) = Q(k)/R(k)$. A natural candidate for the goodness measure is *entropy* since it increases with the number of subtopics per document and as the co-occurrence among subtopics grows bigger. Consider the set of Wikipedia pages $\mathcal{W}$. Let $\vec{z}$ be the topic paths in H assigned to words in $\mathcal{W}$ by WPAM's Gibbs sampling algorithm when training the WPAM model on $\mathcal{W}$. Then, for a non-leaf topic k, we define the entropy $E(k)$ as:

$$E(k) = - \sum_{m \in \mathcal{W}} \sum_{k' \in c(k)} \frac{n_{kk'}^{(m)}}{\sum_{k' \in c(k)} n_{kk'}^{(m)}} \log \frac{n_{kk'}^{(m)}}{\sum_{k' \in c(k)} n_{kk'}^{(m)}} \tag{4.6}$$

(Recall that $n_{kk'}^{(m)}$ is the number of times topic k' appears as a subtopic of topic

k in a topic path of document m.) It is easy to see that if each document contains very few of topic k's children, then the value of $E(k)$ is low. In fact, in the extreme case, when only one subtopic of k appears per document, k's entropy is 0. On the other hand, if many of k's subtopics appear in each document, then the value of $E(k)$ is higher. The maximum value of course is $|\mathcal{W}| \cdot \log |c(k)|$ when all of k's subtopics are uniformly distributed within each document. Thus, the entropy $E(k)$ is effective at capturing subtopic co-occurrence counts for topic k, and so we use it as the goodness measure for topics.

Our scheme for pruning spurious topics from the Wikipedia hierarchy H iteratively deletes topics k with the lowest entropy scores $E(k)$. During each iteration, it breaks ties between topics with identical entropy scores by selecting the topic k that occurs most frequently in topic paths. The rationale here is that frequent topics have higher confidence levels. A topic k is deleted from hierarchy H by performing the following two modifications: (1) For every child k' of k, new edges are added from the parents of k to k' (thus parents of k become parents of k's children), and (2) Topic k and all the edges incident on k are deleted.

Specifically, in each iteration, it first computes the entropy scores for topics with the current hierarchy H, and selects the topic k with the smallest entropy score $E(k)$ for deletion. Let H_{-k} be the resulting hierarchy when topic k is deleted from the current hierarchy H. We learn a WPAM model using Gibbs sampling for the hierarchy H_{-k}. If the model for H_{-k} has disambiguation accuracy that is at least as high as the accuracy of the model for H, then we go ahead and delete topic k from H. Thus, we proceed to the next iteration with the new current hierarchy $H = H_{-k}$. Else if the accuracy for H_{-k} is lower compared to H, then our algorithm terminates and returns the current hierarchy H.

The stopping criterion for deleting topics from H is once the disambiguation accuracy of models trained with the new hierarchy (after a deletion) starts to decrease. In our experiments, we measure the disambiguation accuracy for a hierarchy H by first learning on a training dataset and then testing the learned model on a validation set.

each model on a held out validation set which is a subset of $\mathcal{W}_{annot}$. Note here that the annotations for the held out validation set are not considered when learning models.

In each iteration, our algorithm orders topics based on the goodness measure $G(k)$, and selects the one with the smallest $G(k)$ value for deletion. In practice, however, we have found that ordering topics according to the ratio $G(k)/G(\pi)$ is more effective – here π is the parent of k with the maximum $G(\cdot)$ value. This is because we have found that despite normalizing each $Q(k)$ value with $R(k)$, $G(k)$ values tend to be higher for topics that are closer to the root. Ordering nodes by $G(k)/G(\pi)$ instead of simply $G(k)$ helps to correct this bias, and thus is more robust.

4.4 Experimental Results

In this section, we show that our WPAM model performs better than the state-of-the-art baselines proposed in [30] on real-life datasets. More specifically, we show that WPAM improves the disambiguation accuracy by as much as 22% compared to [30]. We also show that hierarchical topic models outperform the best performing LDA model by as much as 26%. We finally show that hierarchy pruning improves disambiguation performance by 6%.

4.4.1 Experimental Setup

Datasets: We created two datasets – one from Wikipedia and the other from the New York Times corpus. The first dataset (called **WIKI**) is an extract of Wikipedia containing people belonging to seven categories – tennis, basketball, football and baseball players along with actors, musicians, and scholars. This dataset contains 58,577 people, each with an entity page, 627,370 intra-Wikipedia links/annotations and 13,035,881 tokens (after stop word removal). The other dataset is a subset of the New York Times annotated corpus[5]. The full corpus contains 1,855,658 annotated articles published in the New York Times, with 2,372,244 annotations. In some of our experiments, we train on WIKI and test on a subset of the New York Times corpus containing entities that occur in WIKI too. To do this, we first need to match the entity names that occur in the WIKI dataset with those that occur in the New York Times corpus. We use exact string matching

[5] http://www.ldc.upenn.edu/Catalog/

to identify matching entity names. The subset of the New York Times corpus that contains references to these matched entities is what we use to perform our experiments and we refer to this extract as NYT in the sequel. NYT contains 9151 unique person names, 65,012 references, 47,581 documents (containing only the references along with 300 characters of local context), and 1,130,872 tokens after stopword removal.

Metrics: In the experiments, we use precision as the primary metric of comparison: precision is the fraction of entity references that are annotated correctly. In some applications, instead of the best matching entity, a (short) ranked list of entities is used to capture the correct entity. The metric used in these cases is "Precision@k" – defined as the fraction of references for which the correct entity appears in the *top-k* candidate list for that reference. In our experiments, following prior work [30], we compare words appearing in the text with surface forms of entities in Wikipedia to identify references.

Outline of Experiments: The experiments are performed in unlabeled and held-out modes. In the "unlabeled" mode, we present the dataset to the algorithms after hiding a subset of the annotations present in the dataset. Subsequently, we estimate performance based on the predictions on the hidden references. In the held-out mode, all references in the test data are unannotated. The datasets are called WIKI-U, WIKI-H, and NYT-H.

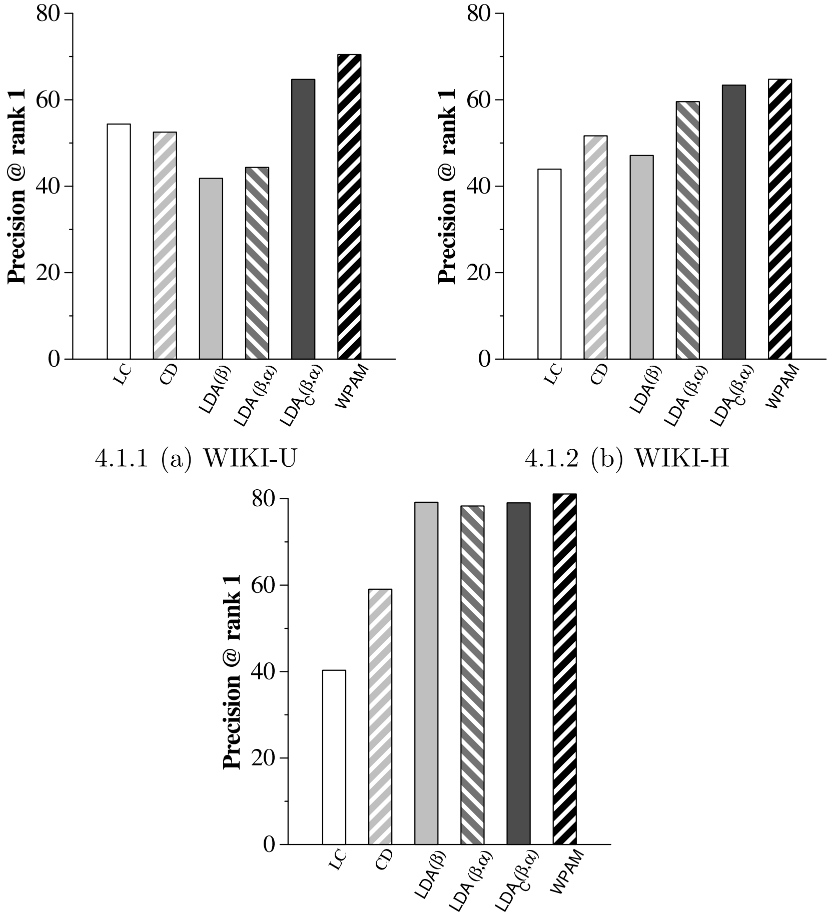
WIKI-U: In WIKI-U (WIKI data in the unlabeled mode), the training data consists of four splits each containing 75% of the pages in WIKI. In the remaining 25%, we hide 30% of the annotations. 100% of the data is used for training and precision is computed on the hidden part. We report the average precision over splits.

WIKI-H: In WIKI-H (WIKI in held-out mode), the splits are the same as WIKI-U. The hidden part of the pages is used to evaluate performance and not used for training.

NYT-H: In this case, WIKI is used for training and NYT is used for testing. We perform the following experiments.

- Baseline, LDA, and WPAM run on WIKI-U and WIKI-H.

- Baseline, LDA, and WPAM run on NYT-H.



4.1.1 (a) WIKI-U

4.1.2 (b) WIKI-H

4.1.3 (c) NYT-H

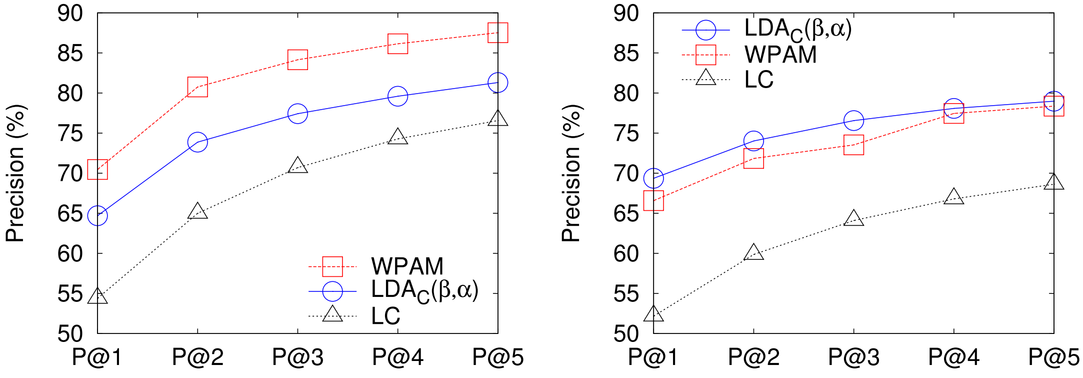
Figure 4.1. Precisions of baselines, LDA variants and WPAM.

- Hierarchy pruning.
- Train and test execution times.

Baselines: We use the following two baselines from [30].

Local Context (LC): Given a local context window around a reference and an entity, LC first builds a feature vector by comparing the window with the entity's textual metadata (extracted from Wikipedia) using string similarity metrics. The feature vector is fed into a support vector machine (SVM) and the entity with the highest SVM score is the disambiguation result.

Collective Disambiguation (CD): CD uses all local contextual windows in a document and attempts to assign an entity to each window such that the sum of the average SVM scores and the average *relatedness scores* [39] among pairs of assigned



(a) WIKI-U (b) NYT-H

Figure 4.2. Precision@k results for WPAM and LC.

entities is maximized. For this optimization, in addition to the greedy hill climbing implemented in [30], we also use more sophisticated inference algorithms such as *loopy belief propagation* (LBP) [55]. The results we report were obtained using max-product LBP (using exponentiated relatedness as entries in clique potentials).

LDA variants: Section 4.2.4 introduced weakly semi-supervised LDAs. To estimate the gains in performance achieved by biasing the distributions of parameters $\vec{\theta}_m$ and $\vec{\phi}_k$, we experiment with two variants of weakly semi-supervised LDAs. Recall that semi-supervised learning was achieved by adding precounts $\vec{\delta}^{(k)}$ and $\vec{\delta}^{(m)}$ to Dirichlet hyperparameters $\vec{\beta}$ and $\vec{\alpha}$, respectively. We refer to WLDA with only precounts in $\vec{\beta}$ as WLDAB, and with precounts in both $\vec{\alpha}$ and $\vec{\beta}$ as WLDABA.

In our WPAM implementation, we found that instead of initializing the Gibbs sampling algorithm with a random topic assignment, initializing word topics in a document with more frequently occurring candidate entities in it gives better performance. We report results with the latter initialization scheme.

4.4.2 Performance Comparison

pat1-fig shows the results of testing on WIKI-U, WIKI-H, and NYT-H. On all three datasets, WPAM outperforms the remaining models.

On the WIKI-U dataset (see pat1-fig(a)), WPAM shows a clear 16% improvement over CD. WLDAB and WLDABA show a nice progression of performance improvements demonstrating the benefits of increasing levels of supervision.

On the WIKI-H dataset (see pat1-fig(b)), WPAM improves upon LC and CD by 23% and 15%, respectively. In this held-out experiment, even WLDABA does better than the baseline methods perhaps indicating some degree of overfitting by the SVM-based methods. The precision of WLDABA is 59.64% which 7% less than WPAM.

On the NYT-H dataset (see pat1-fig(c)), even WLDAB and WLDABA outperform the baselines, again indicating some degree of over-fitting. In fact, all of WLDAB, WLDABA, and WPAM have similar precision on NYT-H, with WPAM doing 1% better than WLDAB.

A likely explanation for this is that since the test set is held out, WPAM is not able to estimate the document's mixture over entities properly. In such a scenario, we found out that biasing the ranking obtained for a reference over the set of entities using the content in the document produces better results. In this case, we sample using the WPAM sampling equation until the joint distribution stabilizes and we begin sampling from the modes of the distribution. In the end, to compute the ranking over the set of entities, we order entities that explain the words in the document better by computing: $\text{score}(k,t) = \prod_{i \in m}(n_{t,-i}^{(k)} + \beta_t^{(k)})/(\sum_{t'}(n_{t',-i}^{(k)} + \beta_{t'}^{(k)}))$, where k is an entity, m is the held-out document and t denotes the term from the vocabulary i corresponds to. Comparing with eq:hlda-cond, $\text{score}(k,t)$ corresponds to the first part of the equation. This boosts WPAM's precision from 78.53% to 81.08% on NYT.

precatn-fig shows Precision@k results, for $k = 1, \cdots, 5$, for WPAM and LC. We use SVM scores and likelihoods to rank for LC and WPAM, respectively. precatn-fig(b) shows that P@5 can reach as high as almost 90%, which is achieved by WPAM on NYT-H.

The superior performance of WPAM can be attributed to the use of the Wikipedia hierarchy to capture entity co-occurrence patterns. Our topic models also benefit significantly in many cases from the semi-supervised learning enabled by Wikipedia annotations. This is corroborated by the fact that unsupervised LDA (on WIKI-U) has a precision of only 9.34%.

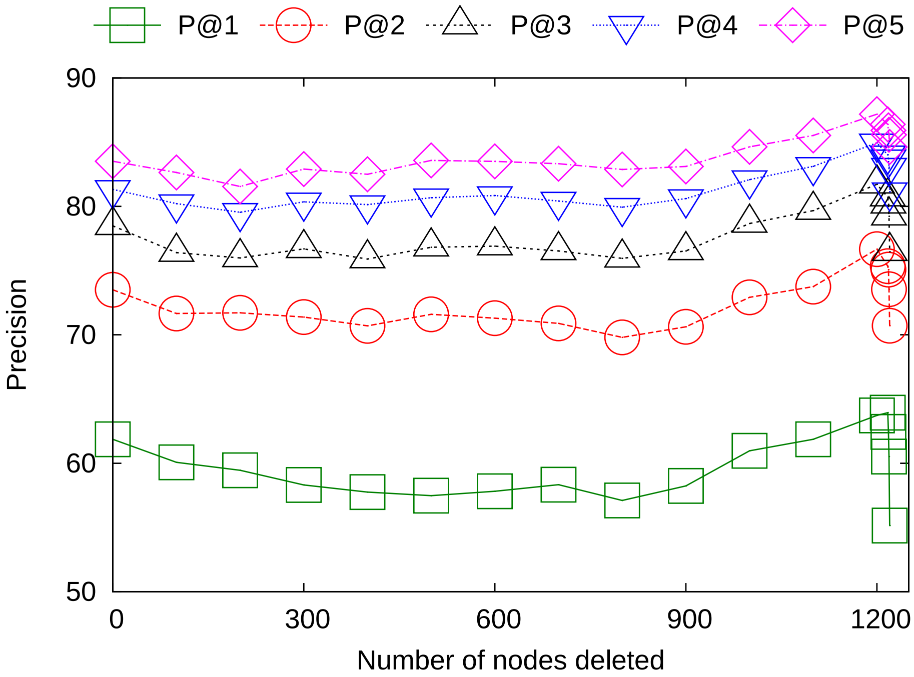


Figure 4.3. Variation of precision with pruning.

4.4.3 Hierarchy Pruning

The default Wikipedia hierarchy for all the entities in our dataset consists of 1225 non-leaf nodes, 58577 leaves, and 75910 paths. As mentioned in Section 4.3.3, not all of the nodes help WPAM, and hence it is necessary to prune the hierarchy. The pruning is done using the entropy-based method as described in sec:pam, which not only uses the topic path assignments for document words, but also the performance of the learned WPAM model on a held-out dataset.

We performed pruning with three datasets, each being a 10% random extract of the WIKI. depicts the variation of precision with number of nodes deleted from the hierarchy averaged across the splits. The performance of pruning reaches a maximum when the optimal hierarchy is reached, and then drops drastically upon further pruning. Since the performance does not increase monotonically towards the maximum, we use this drastic drop as the stopping criterion for pruning.

The reason the performance reaches a maximum at the last stages of pruning is due to the nature of the Wikipedia corpus and its corresponding hierarchy. The final pruned hierarchy consists of one root with 8 non-leaf nodes for children thus forming a three-level hierarchy (along with the entities forming the leaves).

Hence, dynamic pruning deletes most of the non-leaf nodes (more than 1200) in the hierarchy to obtain optimal performance. Observe that the final pruned hierarchy has 6% higher precision compared to the original hierarchy.

4.4.4 Training and Test Times

Topic models are known to require excessive amounts of training time. Our code on full WIKI takes two and a half days to train. For a mid-to-large scale application such as ours, this issue exists with the baseline algorithms too. For example, [30] points out that it is not difficult to generate SVM learning problems for with millions of constraints. The number of constraints is a function of the number of annotations and entities in the training set. Training an SVM on full WIKI has 6,241,263 constraints even after we restrict the number of candidates per reference to 10 and takes almost a day (using the libsvm[6] JAVA implementation).

Labeling with WPAM's Gibbs sampling algorithm, however, is much faster compared to the baseline. For our experiments, the labeling time of WPAM is roughly of the order of few tens of milliseconds per reference. In contrast, testing with the baseline models may require comparing each window with each entity in the extreme case costing around 0.2 sec of disambiguation time per reference. Hence labeling using topic models is approximately 10 times faster.

4.4.5 Anecdotal Evidence

One advantage of using topic models is that we get useful by-products as a result of training, such as, dominant words for entities, dominant entities figuring prominently in documents, etc. In anecdotal, we show some of the top-30 entities in four categories ranked by decreasing values of α_k learned by WPAM. These entities are more prominently featured in the pages of WIKI. Most of the names in anecdotal are well-known (Novak Djokovic, Peyton Manning, Jackie Chan, etc.) as expected, besides some surprises such as Kareem Abdul Jabbar who shows up in the top-30 list of actors but not in the top-30 list of basketball players (Kareem Abdul was a basketball player and an actor, as reflected in the Wikipedia hierarchy we were using).

[6] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

Tennis Players	Jelena Jankovic, Li Na, Jim Courier, Rafael Nadal, Xavier Malisse, Mary Pierce, Novak Djokovic
Football Players	Jerome Bettis, Kevin Mawae, Adrian L. Peterson, Dan Marino, Peyton Manning, Donovan McNabb
Basketball Players	Tim Duncan, Dwyane Wade, Ray Allen, Karl Malone, Chauncey Billups, Shaquille O'Neal
Actors	Gong Li, Amitabh Bachchan, Prasenjit Chatterjee, Greta Garbo, Jackie Chan, Kareem Abdul Jabbar

Table 4.1. Top entities sorted by learned α_k values.

4.5 Related Work

Due to its scale and coverage, Wikipedia has been the knowledge base of choice for most of the large-scale entity disambiguation approaches proposed in the literature [10, 14, 37, 39, 30]. The early Wikipedia-based disambiguation schemes [10, 37] resolved one reference at a time using only local context information. Subsequent works [14, 39, 30] propose to disambiguate all the references in a page collectively taking into account the relatedness of entities. Of these, [39] performs a limited form of collective disambiguation based on entity co-occurrence patterns – the entities selected to disambiguate the ambiguous references in a page are the ones that co-occur frequently with (entities for) the unambiguous references in the page.

Unlike [39], [14] performs full-fledged collective disambiguation by also including ambiguous references in each disambiguation decision. It represents each entity as a feature vector consisting of local context information and categories for the entity. It then disambiguates a reference in a page with the entity whose feature vector matches best with the aggregated feature vector for the page (over all possible entity disambiguations for all references in the page). Clearly, a problem with the scheme of [14] is that the aggregate vector for a page may include irrelevant entities.

[30] formulates the entity disambiguation problem as a joint optimization problem that seeks to collectively annotate all the references within a document so that the compatibility of each entity's context with the local context of its reference and the co-occurrence between entity pairs is maximized. The optimization problem is NP-hard, and the authors resort to a hill-climbing strategy that greedily annotates references with entities that maximize the objective function.

We pointed out the shortcomings of previous Wikipedia-based disambiguation approaches in Section 4.1.2 – these include combining the various sources of evidence like entity context and co-occurrence in a somewhat ad-hoc fashion, using a fixed-size window around a reference to define its local context (with no consensus on the best window size), and ignoring words that appear in the vicinity of un-annotated references when computing the context for an entity.

Our WPAM entity disambiguation model does not suffer from any of the above-mentioned problems since it is based on hierarchical topic models. Hierarchical models allow diverse sources of evidence like word-entity associations and entity co-occurrence patterns to be combined in a single unified framework for entity disambiguation. Moreover, topic models do not need to use difficult-to-set window parameters – their internal machinery can naturally use words from anywhere in the document, including the vicinity of un-annotated references, to learn high-quality word-entity mappings. WPAM differs from previously proposed hierarchical models like PAM [32] in two aspects: (1) WPAM uses the Wikipedia category hierarchy, and (2) Learning in WPAM is weakly semi-supervised with Wikipedia annotations serving as training data.

Word sense disambiguation and entity resolution are two related areas of research that bear close connections to entity disambiguation. Topic models have been used in both of these areas previously [2, 8, 49]. The main differentiating factor between entity disambiguation and such related fields is the assumption of the presence of a catalog of entities. Word sense disambiguation and entity resolution usually do not assume the presence of any such catalog and researchers in these fields have devoted most of their efforts to developing unsupervised algorithms. Thus, none of [2, 8] or [49] consider including prior annotations/labels for weakly semi-supervised learning the way we do in this chapter.

Recently, there have been attempts to incorporate document labels into topic models [4, 31, 46]. Since our aim is to annotate words instead of documents, these approaches are quite different from the topic models we develop in this chapter. [52] proposes the Multi-grain LDA topic model that uses local topics at the sentence level to capture ratable aspects like sound quality, battery life, etc. in user reviews. The Multi-grain LDA model is orthogonal to hierarchical topic models that instead model semantic relationships between topics typically at the document level. Similar to us, [44] runs LDA on Wikipedia pages to discover hidden topics that are then used as additional features to classify short text segments. However, [44] only considers a few hundred coarse-grained topics, and does not exploit Wikipedia's annotations or category hierarchy for topic inference.

Due to its scale and coverage, Wikipedia has been the knowledge base of choice for developing large-scale entity disambiguation approaches [10, 14, 37, 39, 30]. One aspect of Wikipedia that has been exploited widely is the fact that internal links in Wikipedia are expressed as pairs composed of the entity reference and the page of the actual entity it refers to. Thus, these internal links can be used to construct a fully labeled training set of disambiguated entities to learn disambiguation models from.

As mentioned in the Introduction, existing approaches use two key pieces of evidence from Wikipedia for entity disambiguation. First, for an unresolved reference, the local context of the reference is compared with the textual metadata for a candidate entity to obtain a local context compatibility score. Here, the textual metadata for each entity consists of words from the entity's Wikipedia page and the local context of references to the entity in other Wikipedia pages. The second piece of information is a relatedness score that captures co-occurrence patterns between entity pairs. The relatedness score between two entities is proportional to the overlap between Wikipedia pages that contain references linked to the entity.

The early Wikipedia-based disambiguation schemes [10, 37] resolved one reference at a time, essentially selecting the entity with the highest local compatibility score. [39] used the relatedness scores between entity pairs to collectively disambiguate all the references within a page. For each ambiguous reference in a page, candidate entities are scored based on their relatedness scores with respect to the entities for unambiguous references in the page, and these scores along with some additional coherence and quality parameters are used to select the final entity.

[14] also proposes to disambiguate all the references in a page simultaneously, but adopts a slightly different approach. Each entity is represented by a feature vector consisting of the textual metadata and categories for the entity. At test time, given a reference in document d, [14] proposes returning the entity whose feature vector matches best with the aggregated feature vector obtained by summing over all possible disambiguations of the remaining references in d.

Even though [14] does not begin by constructing a fully labeled training set, they too propose an approach where every entity e is represented by a feature vector. A major component of this feature vector comprises entities that co-occur with e obtained from the annotated internal links in Wikipedia. At test time, given a reference in document d, [14] proposes returning the entity whose feature vector matches best with the aggregated feature vector obtained by summing over all possible disambiguations of the remaining references in d.

As we pointed out earlier, the current Wikipedia-based approaches combine the various forms of evidence in a somewhat ad-hoc fashion, define the local context of a reference as a fixed-size window around it (different approaches use different size windows ranging from a few words to an entire para), and only take into account words appearing in the local context of annotated references. As we shall see in subsequent sections, and perhaps testament to the fact that they form a natural model for entity disambiguation, topic models do not need to use such difficult-to-set window parameters and the internal machinery can naturally use words in the vicinity of both annotated as well as un-annotated references to disambiguate.

Using only the annotated parts of Wikipedia is unlikely to produce substantial amounts of training data required to learn large-scale models of disambiguation. On the other hand, research on semi-supervised learning [38, 43] has shown that if we can avail of large amounts of unlabeled data and combine that with the labeled training data then the accuracy of machine-learned models can be drastically improved. Wikipedia certainly provides significant amounts of un-annotated references (for instance, all instances of "Karl Rove" are not expressed as internal Wikipedia links on George W. Bush's page in Wikipedia[7]). A rough count of the number of person references in Wikipedia showed that compared to the internal links pointing to a person's page there are a possible number of un-annotated per-

[7] http://en.wikipedia.org/wiki/George_W._Bush

son references. Research on semi-supervised learning has developed a number of different techniques to exploit unlabeled data along with labeled data (see [56] for a survey), we concentrate on developing on an approach that lets us incorporate content, entity co-occurrence patterns and easily includes both annotated and un-annotated references. To this end, we concentrate on using topic models.

Topic models allow one to learn a set of multinomial distributions over words called topics to describe documents [5]. As mentioned in the introduction, one of our key ideas that allows us to extend the use of topic models to disambiguate entities is to associate with each entity a unique topic. This way, for each entity, the multinomial distribution over words that we end up learning captures the content-entity associations for that entity which we subsequently use to disambiguate. The earliest topic models assumed topics were independent [5] and could not capture correlations between topics. [3] was one of the earliest proposals to represent correlations among topics but they used a set of $O(k^2)$ parameters to model correlations between k topics. [32] proposed modeling a directed acyclic graph (DAG) structure among topics to capture correlations using a restricted set of parameters. [32] allowed modeling correlations between parent-child topics and the number of extra parameters required is $O(E)$ where E is the number of edges in the DAG structured hierarchy which, in the worst case, can be $O(k^2)$ but usually is much less. Most of the discussion in [32] is restricted to a four level topic hierarchy.

In this chapter, we extend the techniques proposed in [32] to general, non-uniform, arbitrary sized DAGs such as the Wikipedia concept hierarchy to capture co-occurrence among entities. Note that, category information from the Wikipedia concept hierarchy has been used before to disambiguate entities [10, 14] but more as words from an augmented vocabulary that captures content similarity and not to capture entity co-occurrence patterns.

Recently, there have been attempts to incorporate document labels into topic models [4, 31, 46]. Since our aim is to annotate words instead of documents, these approaches are quite different from the topic models we develop in this chapter. Perhaps closer to our application, [48] considers annotating words with labels they refer to as “aspects”. However, unlike our approach of using entity-specific topics, [48] uses topics to generate both the word and its label parallely. Of course, [48] does not use concept hierarchies to incorporate co-occurrence patterns.

Word sense disambiguation and entity resolution form two related areas of research that bear close connections to entity disambiguation. Topic models have been used in both of these areas previously [2, 8, 49]. The main differentiating factor between entity disambiguation and such related fields is the assumption of the presence of a catalog of entities. Certainly, if entity references in different documents were not annotated using ids from the same catalog then some of the applications of entity disambiguation such as information integration would not be possible. Word sense disambiguation and entity resolution usually do not assume the presence of any such catalog and researchers in these fields devote most of their efforts to developing unsupervised algorithms. Thus, none of [2, 8] or [49] consider including prior annotations/labels and none of them consider developing weakly semi-supervised techniques the way we do in this chapter.

4.6 Summary

In this chapter, we proposed the weakly semi-supervised hierarchical topic model WPAM for disambiguating entities. WPAM uses all the words in a document, including those in the vicinity of un-annotated references, to learn high-quality word-entity associations. It leverages Wikipedia annotations to appropriately bias the assignment of entity labels to annotated words (and un-annotated words co-occurring with them), and the Wikipedia category hierarchy to capture entity context and co-occurrence patterns in a single unified disambiguation framework. We devised an algorithm for efficiently sampling paths from large topic hierarchies, and a scheme for pruning spurious categories with poorly correlated sub-categories in the hierarchy. In large-scale experiments with both held out subsets of Wikipedia and the NYT corpus, our WPAM model achieved 81% disambiguation accuracy compared to 59% for the existing state-of-the-art baselines. Promising directions for future work include exploring the use of other hierarchical topic models (e.g., hPAM [40]) for entity disambiguation, and learning good topic hierarchies from Wikipedia using non-parametric priors (e.g., hierarchical Dirichlet processes [51]).

Chapter 5

Conclusion and Future Work

In this thesis, I have worked on enhancing the capabilities of a particular existing class of generative machine learning models, named topic models, specific to domain of document networks. Aligned to the enhancing efforts, this thesis also proposes novel topic models applicable to link prediction problems in document network. The proposed topic models are designed for following three kind of link prediction problems in document networks: (1) citation prediction in scientific documents, (2) predicting links among the citing authors and cited authors in an author citation graph of document networks, and (3) predicting links among an entity reference and its Wikipedia page. I describe my individual research contribution next.

1. **Utilizing Citation Context in Topic Models for Citation Recommendation**

   I presented topic modeling based statistical generative models that utilizes context information of citations in documents to model the generation process of documents and citations together. Identifying the text from the context that describes the cited document and utilizing it in topic model based statistical process is a challenging task. I demonstrate how context length for each individual context can be selected effectively and how topic models can incorporate the selected citation context explicitly. For selecting the context length for each citation mention in a citing article, I greedily select only the length of context which best "matches" the concepts described in the cited article. For incorporating the selected citation context in topic modeling framework, I make a simplifying statistical independence assumption between the generation of link and generation of words in the context window. Both of these two processes happen simultaneously and I design a heuristic Gibbs sampling mechanism to infer the parameters of the designed model. The proposed models explains the generation process of the links and content both qualitatively and quantitatively. The designed Gibbs sampling to perform inference on emission probabilities corresponding to citations and words given a topic and show significant improvement on various objective functions.

2. **Detecting Topic Specific Influential Authors**

   I proposed novel models for author-author linkage in author citation networks conditioned on topics latent in the content of the documents. The proposed models exploit the citations between documents to infer influence of certain authors over topics. Drawing motivation from context sensitive topic models in chapter[], I also proposed context sensitive extensions of the topic models for author citation network. Corresponding to a given topic, I identified two kind of authors in document networks: (1) interested authors, and (2) influential authors; where I define an expert/interested author as someone who has produced several contributions in a particular field whereas an influential author as someone who has certain key contributions in that field and gets cited more often. I evaluated the quality of these identifications quantitatively by predicting which authors get cited in a given document. I also presented various anecdotal evidences to ascertain the quality of the interested and influential authors in various topics.

3. **Employing Topic Models for Learning to Link to Wikipedia**

   I proposed the weakly semi-supervised hierarchical topic model, WPAM, for disambiguating entities. WPAM has two primary sources of supervision: (1) annotated references of entities in Wikipedia pages, and (2) category information for entity pages present in Wikipedia. WPAM takes advantage of the already present supervision in Wikipedia in following ways: (1) WPAM uses all the words in a document, including those in the vicinity of un-annotated references, to learn high-quality word-entity associations. It leverages Wikipedia annotations to appropriately bias the assignment of entity labels to annotated words (and un-annotated words co-occurring with them), and (2) WPAM utilizes Wikipedia category hierarchy to capture co-occurrence patterns present among entities and learns high-quality entity-entity associations. Moreover, WPAM presents these learning capabilities in a single unified disambiguation framework. I also devised a scheme for pruning spurious categories with poorly correlated subcategories in the hierarchy. In large-scale experiments with both held out subsets of Wikipedia and the NYT corpus, the WPAM model achieved over 70% disambiguation accuracy compared to about 54% for the existing state-of-the art baselines.

5.1 Future Work

Following are few future research directions that can be explored based upon the work presented in this dissertation.

1. **Utilizing Hierarchical Topic Models for Citation Prediction**

   Digital libraries such as ACM, IEEE Xplore, etc., provides the document collection with category information. An interesting direction to pursue is to adapt hierarchical topic models, such as Pachinko Allocation Model (PAM), Chinese Restaurant Processes (CRP), for citation prediction. Intuitively, within a citing document, authors tend to cite documents from same field of study as that of the citing article. Since hierarchical topic models tend to obtain a hierarchical clustering of documents, it can be interesting to study the citation pattern with in a bottom-up hierarchy of documents.

2. **Future directions for "Topic" based Citation Recommendation System**

   Contextual Query Sensitive Citation Recommendation
   Although the training phase of Citation Recommendation learn from citations explicitly, however, in on-line phase, it suggest citations based upon a

global level topical analysis of query document. The query document may contain several topics discussed, however, usually the context of citation contains a few topics which are related to both citing and cited topics. An explicit consideration of topics in citation context while making recommedations may improve the quality of recommendations.

3. **Automatic Construction of Hierarchy in Crowd-Sourced knowledge bases** As mention in chapter 4, the category present in Wikipedia is crowd-sourced in nature and uncorrelated with topics present in Wikipedia corpus. For example, category such as "Person born in Year XXXX" are uncorrelated with field of person and provides a noisy measurement for topical clustering of entities. I proposed a method to prune the hierarchy based upon topical cohesion of entities falling in one hierarchy with entropy based method. However, the purpose of pruning is to obtain a better link prediction in Wikipedia. However, the method proposed in this thesis can also be applicable to automatic construction of hierarchy in crowd-sourced documents.

Bibliography

[1] K. Balog, L. Azzopardi, and M. de Rijke. A language modeling framework for expert finding. *Inf. Process. Manage.*, 45(1):1–19, 2009. 37

[2] I. Bhattacharya and L. Getoor. A latent dirichlet model for unsupervised entity resolution. In *SDM*, 2006. 57, 59, 60, 82, 86

[3] D. Blei and J. Lafferty. Correlated topic models. In *NIPS*, 2006. 85

[4] D. Blei and J. McAuliffe. Supervised topic models. In *NIPS*, 2007. 59, 83, 85

[5] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *JMLR*, 2003. 56, 85

[6] D. M. Blei and J. D. Lafferty. Correlated topic models. In *NIPS 18*, 2006. 4, 9

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003. 2, 3, 4, 5, 8, 9, 15, 35

[8] J. Boyd-Graber, D. Blei, and X. Zhu. A topic model for word sense disambiguation. In *EMNLP*, 2007. 57, 59, 60, 82, 86

[9] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Computer Networks and ISDN Systems*, pages 107–117, 1998. 4, 9

[10] R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *European Association for Computational Linguistics*, 2006. 53, 54, 55, 56, 81, 83, 85

[11] J. Chang and D. Blei. Relational topic models for document networks. In *Proc. of Conf. on AI and Statistics (AISTATS'09)*, 2009. 4, 36, 38, 45

[12] J. Chang and D. M. Blei. Relational topic models for document networks. *Journal of Machine Learning Research - Proceedings Track*, 5:81–88, 2009. 1

[13] D. Cohn and T. Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity. In *NIPS 13*, 2001. ix, 10, 33

[14] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP*, 2007. 53, 54, 55, 56, 81, 83, 84, 85

[15] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990. 2

[16] L. Dietz, S. Bickel, and T. Scheffer. Unsupervised prediction of citation influences. In *ICML 2007*, pages 233–240, 2007. 4, 36, 38

[17] H. Fang and C. Zhai. Probabilistic models for expert finding. In *ECIR*, pages 418–430, 2007. 37

[18] L. Getoor and C. P. Diehl. Introduction to the special issue on link mining. *SIGKDD Explorations*, 7(2):1–2, 2005. 2

[19] A. Gohr, A. Hinneburg, R. Schult, and M. Spiliopoulou. Topic evolution in a stream of documents. In *SDM*, pages 859–872, 2009. 9

[20] T. Griffiths and M. Steyvers. Finding scientific topics. In *Proceedings of the National Academy of Sciences*, 2004. 61

[21] T. L. Griffiths and M. Steyvers. Finding scientific topics. *In Proc of National Academy of Science U.S.A.*, pages 5228–5235, 2004. 20, 21

[22] A. Gruber, M. Rosen-Zvi, and Y. Weiss. Latent topic models for hypertext. In *UAI*, pages 230–239, 2008. 1

[23] A. Gruber, M. Rosen-Zvi, and Y. Weiss. Latent topic models for hypertext. In *UAI*, pages 230–239, 2008. 38

[24] Q. He, J. Pei, D. Kifer, P. Mitra, and C. L. Giles. Context-aware citation recommendation. In *WWW*, pages 421–430, 2010. 30

[25] G. Heinrich. Parameter estimation for text analysis. Technical report, 2005. 61

[26] T. Hofmann. Probabilistic latent semantic analysis. In *UAI 1999*, pages 289–296, 1999. 3, 4, 5, 8, 9, 10, 15, 36

[27] S. Kataria, K. S. Kumar, R. Rastogi, P. Sen, and S. H. Sengamedu. Entity disambiguation with hierarchical topic models. In *KDD*, pages 1037–1045, 2011. 53

[28] S. Kataria, P. Mitra, and S. Bhatia. Utilizing context in generative bayesian models for linked corpus. In *AAAI*, 2010. 19, 36, 38, 42

[29] S. Kataria, P. Mitra, C. Caragea, and C. L. Giles. Context sensitive topic models for author influence in document networks. In *IJCAI*, pages 2274–2280, 2011. 38

[30] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of wikipedia entities in web text. In *KDD*, 2009. 53, 54, 55, 56, 59, 74, 75, 76, 77, 80, 81, 82, 83

[31] S. Lacoste-Julien, F. Sha, and M. I. Jordan. DiscLDA: Discriminative learning for dimensionality reduction and classification. In *NIPS*, 2008. 83, 85

[32] W. Li and A. McCallum. Pachinko allocation: DAG-structured mixture models of topic correlations. In *ICML*, 2006. 56, 58, 65, 82, 85

[33] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proceedings of the twelfth international conference on Information and knowledge management*, CIKM '03, pages 556–559, New York, NY, USA, 2003. ACM. 1, 2

[34] Y. Liu, A. Niculescu-Mizil, and W. Gryc. Topic-link lda: joint models of topic and author community. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 665–672, New York, NY, USA, 2009. ACM. 5, 36, 38

[35] L. Lu and T. Zhou. Link prediction in complex networks: A survey. *CoRR*, abs/1010.0725, 2010. 1

[36] Q. Lu and L. Getoor. Link-based classification. In *ICML*, pages 496–503, 2003. 45

[37] R. Mihalcea and A. Csomai. Wikify! linking documents to encyclopedic knowledge. In *CIKM*, 2007. 53, 54, 55, 56, 81, 83

[38] D. J. Miller and H. S. Uyar. A mixture of experts classifier with learning based on both labelled and unlabelled data. In *NIPS*, 1997. 64, 84

[39] D. Milne and I. Witten. Learning to link with Wikipedia. In *CIKM*, 2008. 53, 54, 55, 56, 76, 81, 83

[40] D. Mimno, W. Li, and A. McCallum. Mixtures of hierarchical topics with pachinko allocation. In *ICML*, 2007. 86

[41] T. P. Minka. Estimating a dirichlet distribution. Technical report, Microsoft Research, 2003. 62

[42] R. M. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen. Joint latent topic models for text and citations. In *KDD 2008*, pages 542–550, 2008. viii, 1, 3, 4, 10, 14, 20, 24, 27, 29, 30, 36, 38, 45

[43] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Learning to classify text from labeled and unlabeled documents. In *AAAI*, 1998. 64, 84

[44] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *WWW*, 2008. 59, 83

[45] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *KDD 2008*, pages 569–577, 2008. 20

[46] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP*, 2009. 83, 85

[47] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *UAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press. 3, 5, 36, 37, 46, 47

[48] M. Sharifi. Semi-supervised extraction of entity aspects using topic models. Master's thesis, Carnegie Mellon University, 2009. 85

[49] L. Shu, B. Long, and W. Meng. A latent topic model for complete entity resolution. In *ICDE*, 2009. 57, 59, 60, 82, 86

[50] E. E. Stephen, S. Fienberg, and J. Lafferty. Mixed membership models of scientific publications. In *Proceedings of the National Academy of Sciences*, page 2004, 2004. viii, 10, 12, 27, 29, 38

[51] Y. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical dirichlet processes. *J. of the American Statistical Association*, 2005. 86

[52] I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *WWW*, 2008. 59, 83

[53] Y. Tu, N. Johri, D. Roth, and J. Hockenmaier. Citation author topic model in expert search. In *COLING (Posters)*, pages 1265–1273, 2010. 19, 36, 37, 42, 47, 49

[54] H. Wallach, D. Mimno, and A. McCallum. Rethinking lda: Why priors matter. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1973–1981. 2009. 26

[55] J. Yedidia, W. Freeman, and Y. Weiss. Generalized belief propagation. In *NIPS*, 2000. 77

[56] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005. 85

Vita

Saurabh Kataria

Research Experience

Research Intern, **Yahoo! Labs, India** Summer 2010

Research Intern, **Xerox Research Center, Europe** Summer 2009

Research Asst., **Intelligent Information Systems Lab, PSU** Fall '06-12

Refereed Publications

- **Entity Disambiguation with Hierarchical Topic Models**, Saurabh Kataria, K. Kumar, Rajeev Rastogi, Prithviraj Sen, S. Sengamedu, *17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* **(KDD-2011).**

- **Text Classification using Abstract Features**, Cornelia Caragea, Adrian Silvescu, Saurabh Kataria, Doina Caragea, Prasenjit Mitra, *Symposium on Abstraction, Reformulation, and Approximation* **(SARA-2011).**

- **Context Sensitive Topic Models for Author Influence in Document Networks**, Saurabh Kataria, Prasenjit Mitra, C. Lee Giles, *22nd International Joint Conference on Artificial Intelligence* **(IJCAI-2011).**

- **Utilizing Context in Generative Bayesian Models for Linked Corpus**, Saurabh Kataria, Prasenjit Mitra, Sumit Bhatia, *Association for the Advancement of Artificial Intelligence* **(AAAI-2010).**

- **Font Retrieval on a Large Scale: an Experimental Study**, Saurabh Kataria, Luca Marchesotti, Florent Perronnin, *International Conference on Image Processing* **(ICIP-2010).**

- **Generative Models for Authorship Networks**, Saurabh Kataria, Prasenjit Mitra, C. Lee Giles *Workshop on Machine Learning for Social Computing, Neural Information Processing Systems* **(MLSC-NIPS-2010).**

- **Automatic Extraction of Data Points and Text Blocks from 2-Dimensional Plots in Digital Documents**, Saurabh Kataria, William Browuer, Prasenjit Mitra, C. Lee Giles *Association for the Advancement of Artificial Intelligence* **(AAAI-2008).**

- **Segregating and Extracting Overlapping Data Points in Two-dimensional Plots**, William Browuer, Saurabh Kataria, Sujatha Das, Prasenjit Mitra, C. Lee Giles *Joint Conference on Digital Libraries* **(JCDL-2008).**

- **Automated Analysis of Images in Documents for Intelligent Document Search**, Xiaonan Lu, Saurabh Kataria, William Brouwer, James Z. Wang, Prasenjit Mitra, C. Lee Giles *International Journal on Document Analysis and Recognition* **(IJDAR-2008).**