A Hierarchical Statistical Framework for the Extraction of Semantically Related Words in Textual Documents

Weijia Su

A Thesis

in

The Concordia Institute

for

Information Systems Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science (Quality System Engineering) at

Concordia University

Montréal, Québec, Canada

June 2013

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Weijia Su

Entitled: A Hierarchical Statistical Framework for the Extraction of Semantically Related Words in Textual Documents

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Quality System Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. Ben Hamza Chair

Dr. Mohammad Mannan Examiner

Dr. Abdelwahab Hamou-Lhadj Examiner

Dr. Nizar Bouguila and Dr. Djemel Ziou Supervisor

Approved by ______________________
Chair of Department or Graduate Program Director

____
Dean of Faculty

Date June 25, 2013

Abstract

A Hierarchical Statistical Framework for the Extraction of Semantically Related Words in Textual Documents

Weijia Su

Nowadays there exist a lot of documents in electronic format on the Internet, such as daily news, blog articles, messages posted online, even books and magazines. The information that can be extracted from these documents is of particular importance to several agencies and companies (e.g. security agencies, insurance companies, advertising and marketing companies, etc.). In the case of security, for instance, recent studies have shown that cyber criminals generally exchange their experiences and knowledge via media such as forums and blogs. These exchanged data, if well extracted and modeled, can provide significant clues to agencies operating in the security field. However, managing and processing the huge quantity of multimodal (i.e. image, video, text, audio) information present on the Web is a challenging task. In this thesis, we focus on textual data for which many statistical language modeling frameworks have been developed to facilitate the management of digitized texts. Many of these approaches have achieved great performances on various applications. However, most of them have focused on modeling documents individually, while in real world most documents are related, organized and archived into categories according to their themes. The main goal of this thesis is to propose a hierarchical statistical model to analyze documents collections, characterized by a hierarchical structure, to find hidden information and detect potential threats according to them. The proposed model is part of a large cyber security forensics system that we are designing to discover and capture potential security threats by retrieving and analyzing data gathered from the Web. Our approach models each node in a given textual collection using advanced statistical techniques and allows capturing the semantic information hidden inside it. In particular, a log-bilinear model is adopted to describe words in vector space in such a way that their correlations can be discovered and derived, from their representations, at each level of the hierarchical structure. Experimental results on real world data illustrate the merits of our model and its efficiency in extracting hidden semantic information from documents collections.

Acknowledgements

First and foremost, I would like to express my greatest gratitude to my supervisor Dr. Nizar Bouguila and co-supervisor Dr. Djemel Ziou. Many thanks to my colleagues in the lab, for their helpful suggestions during my two years study. And finally, I would like to thank my family for unconditional support throughout my studies, your endless love and care always encourage me all the time.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

1.1 Background

The rapid growth of electronic texts such as news, blogs, web pages, and even books on the Internet brings us both great opportunities and challenges. On one hand, Internet users can more easily get access to a large quantity of information which are available on web sites. On the other hand, people can get confused and overwhelmed by this huge amount of information. Thus, many researches have focused on language modeling using statistical methods. By describing texts in mathematical ways, hidden structures and properties within texts and correlations between them can be discovered, which can help practitioners to explore, organize and manage them more easily. It is noteworthy that several studies [9, 13, 15] have shown that cyber criminals tend to exchange their thoughts, knowledge, and make attacking announcements through media such as specific web sites, forums and blogs on the Internet. This provides a significant meaning in language modeling in the field of cyber security. Cyber criminal activities can be predicted, detected, and captured in advance by analyzing semantic information in texts posted online.

According to the method used to represent words in documents, there are two main language modeling categories: probabilistic topic models and vector space models:

Probabilistic topic models such as Probabilistic Latent Semantic Indexing (PLSI) [18] and Latent Dirichlet Allocatioin (LDA) [5], model a text document as a finite mixture of specific distributions over topics. Each topic is represented as a distribution of words in a given defined vocabulary set. It is important to mention that this type of models is based on the ”bag-of-words” assumption. That is the orders of the appearance of words are irrelevant, each word is independently selected from topics and they are exchangeable in sequence. Since probabilistic topic models catch the semantic properties within a document, they have been applied in various applications such as the discovery of scientific topics [17], the discovery of mixed memberships on scientific publications [12], the discovery of semantic communities as done in [52], the discovery of citations relations to a given paper [10], and the discovery of authors' influences [29]. Other examples include ad-hoc retrieval using LDA [48] which was used also in conjunction with WordNet in [6] to resolve word ambiguity. The measurement of the importance of a document in a collection as proposed in [14] is another example. In the field of cyber security, text mining using probabilistic topic models was introduced in [23].

Unlike probabilistic topic models that regard a document as words which are individually and independently drawn from certain distributions, a vector space model (VSM) [47] represents documents as vectors where each vector can be viewed as a point in a multi-dimensional space. The basic idea behind VSM is that in space, the closer the two points are, the more semantic similarity they are sharing and vice versa. Due to its properties, VSM approach has shown excellent performance in many real world tasks related to the measurement of semantic similarities between documents, sentences, and words. For instance, the distance between queries and documents are calculated using VSMs in most of the well-known search engines [27]. Other examples include, the development of semantic relatedness measurements as done in [35, 46], measuring the similarity of semantic relations [24, 31, 45]. In the field of measurement of words similarities, the authors in [38] developed a vector-based representation of words, tested it on ”Test of English as a Foreign Language (TOEFL)” synonyms multiple-choice questions and achieved an accuracy of 92.5% on it.

All the methods mentioned above have focused on capturing characteristics and properties between words, sentences, and documents at a document level or at a single documents collection level. However, more and more textual data such as news, discussions on forums, blog articles, and cyber crime related texts on the Internet are automatically or manually classified into hierarchical categories and available to people. Thus, developing models that are able to discover patterns and properties within a hierarchical structure representing a given collection is crucial and urgent [11, 40, 50]. The main goal is to achieve a better understanding of the information and relations conveyed by the data. This will allow us, for instance, to analyze and detect potential security threats in advance.

1.2 Objective

More and more textual data are digitized and stored online. Within these data, there are large quantities of user-generated texts such as forum discussions, and blog articles, which provide a great treasure of information, especially in the field of cyber crime forensics. Indeed, cyber criminals and victims tent to exchange their ideas, experiences and information using media on the Internet. Many data are categorized online into well-organized documents collections according to their themes and contents. For example, discussions on an anti-phishing forum can be classified into categories such as ”advanced-fee scam”, ”lottery scam”, ”job scam”, ”date scam” and so on. It is crucial to explore these data and to find possible hidden information. An important issue in this case is to preserve the documents collection structure, which can help us to have a better and clear understanding of the information conveyed by the documents, and then facilitate the process of cyber crimes investigation. The objective of this thesis is to develop a statistical framework in order to analyze texts having a hierarchical structure. These texts can be extracted from the Web to detect and identify potential cyber threats. Our framework is part of a large cyber crime forensics system, which is designed to explore, analyze, and discover hidden information and their correlations, from the data it gathers in the Web, and then generates alerts and warnings about potential threats. To summarize, our objective is to develop a statistical approach capable to describe a given textual collection that has a hierarchical structure and to catch relevant information such as words correlations. At each level of the structure, nodes are interpreted as statistical probabilities and words in the collection are represented in a vector space.

1.3 Contributions

The contributions of this thesis are summarized as follows:

☞ **Develop a hierarchical statistical model for describing documents collection** :

> We propose a hierarchical statistical method based on log-bilinear document model to represent a collection of documents which has a hierarchical structure. A hierarchical documents collection is a collection which can be represented as a tree structure. The model provides a probabilistic description for each node at the collection tree. Words are modeled by log-bilinear model, and a word representation vector is derived from the term-document information for each individual word in model learning process. Thus, our model can determine semantic information and word correlations at each node.

☞ **Verify the hierarchical statistical document model with real world data collections** :

> We examine the performance of our proposed method with various real data collected from different Internet web sites. The verification is divided into two tasks: Word learning at each node of the tree and word classification. Experimental results have shown that the proposed model achieves good results for both tasks.

1.4 Thesis Overview

The rest of this thesis is organized as follows:

❑ In Chapter 2, we present and explore in details some language modeling approaches. These approaches are categorized into two categories, namely probabilistic topic models and vector space models, according to the way data are presented.

❑ In Chapter 3, we propose a hierarchical statistical model for representing a documents collection. An approach to learn this model is also developed.

❑ In Chapter 4, we examine our proposed hierarchical statistical document model by conducting experiments on 3 real world data sets collected from different Web sites. The model is tested by 2 tasks, word learning and word classification. Experimental processes and obtained results are provided in this chapter. Our experimental results show that the proposed model can successfully learn words in a hierarchical documents collection then extract semantically related words.

❑ In the last chapter, we summarize our contributions and present some potential future works.

Chapter 2

Literature Review

2.1 Introduction

Nowadays, people are facing tons of information in various domains on the Internet, such as Wikipedia articles, news, blogs, discussions and messages posted online. Thus, a lot of research efforts have been done in language modeling field in order to provide tools to understand, organize, and manage these data. In this chapter we summarize and discuss previous works on language modeling in details.

2.2 Representation Methods for Documents

Document representation is a critical part in language modeling, since it provides a way to efficiently organize and index document content. There are two widely used methods in document representation, Bag-of-Words (BoW) [3, 5, 28] and N-Gram [16, 26, 44]. BoW method focuses on capturing semantic information of documents. In this method, words order is ignored and a document is represented by a vector using the frequency of each term in the document. Both probabilistic topic and vector space models are in this category. The limitation of this method is that it fails to catch words order which plays an important role in the document structure. N-Gram method focuses on modeling local linguistic structure of a document which is contained in words order. In

N-Gram method, a word is conditionally generated on a short sequence of previous words. Therefore, a string such as a sentence or a document is denoted by a product of probabilities of words given some previous words. The limitation of this method is that sometimes words can be highly independent. For instance, ”I am going to -” may be followed by ”the office” in one document, while in another document, it may be followed by ”drinking”.

2.3 Probabilistic Topic Models

Probabilistic topic models discover and capture hidden structures, known as topics in a collection of documents. In probabilistic topic model, a document is modeled as a finite mixture of specific distributions over topics. The first topic model was introduced in [18], where the author developed the Probabilistic Latent Semantic Indexing (PLSI) model to describe the co-occurrence of words and documents using a mixture of multinomial distributions. The limitation of this method is that the generation of topic proportions is not clear. The Latent Dirichlet Allocation (LDA) [5] solved the limitation of PLSI model by providing a dirichlet prior at the level of document. Thus, LDA results in more reasonable mixtures of topics in a document as compared to PLSI. It has become the most popular method in probabilistic topic modeling, and many extensions have been developed based on it. All probabilistic topic models are based on the bag-of-words assumption which assumes that words are drawn from topics independently and they are exchangeable in order.

2.3.1 Unigram Mixture Model

Unigram model considers that each document is determined by a single distribution, usually the multinomial distribution. For a document d that consists of a bag of N words, it is denoted as:

$$p(d) = \prod_{n=1}^{N} p(w_n) \tag{1}$$

In [34], an unigram mixture model was proposed by adding the topic mixture component z. The word generation process described in this model is: first select a topic z; then independently generate each word in the document by taking z as a condition. Thus, a document can be modeled as follows:

$$p(d) = \sum_{z} p(z) \prod_{n=1}^{N} p(w_n|z) \tag{2}$$

According to the previous equation, a document can have only one topic which largely limits the capability of modeling documents, since in real world, usually several different topics are covered in a single document.

2.3.2 Probabilistic Latent Semantic Indexing (PLSI)

The Probabilistic Latent Semantic Indexing (PLSI) model [18] frees the unigram mixture model from the constraint that each document can have only one topic. It allows words that appear in a document d to have different latent topics. The model first uses $p(d)$ to choose a document; then chooses the latent topic variables z over d and words are selected according to z. The probability of a given word w_n observed in document d is:

$$p(w_n|d) = \sum_{z} p(w_n|z)p(z|d) \tag{3}$$

And the joint probability of all the words in d is:

$$p(d, \mathbf{w}) = p(d) \prod_{n} \sum_{z} p(w_n|z)p(z|d) \tag{4}$$

The capability of modeling a document which consists of multiple topics is shown in $p(z|d)$, which is the distribution of latent topic variable z over document d. However, there are limitations in this model. First, the model does not include the prior distribution of document $p(d)$ for an unseen document, since it is calculated from the training data set. Second, due to the property of this model, the number of parameter grows linearly with the size of document which generally leads to overfitting problems.

2.3.3 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) [5] improves the PLSI model by adding a Dirichlet prior at the document level to determine the topic proportion. It presents documents in a corpus as mixtures of distributions over latent topic variables. And each topic is modeled as a distribution over words. The process of document generation in LDA model is as follows: First, a set of multinomial distributions over words are chosen for topics. Second, a topic distribution is determined to work as the mixture component. Then, a topic is randomly chosen according to that topic mixture distribution, and a word is generated from the corresponding topic multinomial distribution.

The joint probability $p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta)$ describes exactly how LDA generates each word in documents. Assume that a document $\mathbf{w} = \{w_1, w_2, ...w_n\}$ is from a corpus D, where w_n represents nth word in the document. The vocabulary size V and topic number T are fixed. First, the model gets the topic mixture proportion θ for the document from a Dirichlet distribution over free parameter α. Then, it chooses a topic z_n for w_n by a multinomial distribution. And after that, w_n is generated from a multinomial distribution conditioned on z_n and the word probability matrix β. Therefore, the joint probability $p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta)$ is denoted as:

$$p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta) \tag{5}$$

The dimension of word probability matrix β is $T \times V$ and $\beta_{ij} = p(w^j = 1|z^i = 1)$. Each row is a topic distribution of the whole vocabulary. And since $p(z_n|\theta)$ is multinomial distribution, we have $p(z_n|\theta) = \theta_i$.

The topic proportion variable θ has a dimension T and it is determined by a Dirichlet distribution with free parameter α:

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{T} \alpha_i)}{\prod_{i=1}^{T} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} ... \theta_T^{\alpha_T - 1} \tag{6}$$

Here α is a T-dimensional positive vector, and $0 \leq \theta \leq 1$, $\sum_{i=1}^{T} \theta_i = 1$.

The marginal distribution of the document can be obtained by summing up over corresponding topic assignments $\mathbf{z}$ of words $\mathbf{w}$ and integrating over topic proportion θ:

$$p(\mathbf{w}|\alpha,\beta) = \int p(\theta|\alpha)(\prod_{n=1}^{N}\sum_{n=1}^{N} p(z_n|\theta)p(w_n)|z_n,\beta))d\theta \tag{7}$$

Then the probability of the whole corpus collection can be found as:

$$p(D|\alpha,\beta) = \prod_{d}\int p(\theta_d|\alpha)(\prod_{n=1}^{N_d}\sum_{n=1}^{N_d} p(z_{dn}|\theta_d)p(w_n)|z_{dn},\beta)d\theta_d \tag{8}$$

The model can be learned using variational inference and Gibbs sampling method. Variational inference is a deterministic approximation scheme which is used to formulate the computation of a marginal or conditional probability in terms of an optimization problem. It first chooses hyper parameters to simplify the original probabilities. Then, it applies the variational EM algorithm to iteratively optimize parameters and maximize the lower bound of the real posterior probability. The Gibbs sampling applied in LDA model maximizes the posterior probability given the observation of documents by generating a Markov chain of the hidden variable $\mathbf{z}$. After iterations, it will converge to the target posterior according to the sampled $\mathbf{z}$, thus, parameters of the LDA model can be estimated.

Since LDA is the most popular probabilistic topic model, many extensions have been developed based on it. For example, extra information such as document label [4, 22, 36, 37], authorship [39] and domain knowledge [1] are imported into the basic model to have better descriptions for documents in specific fields. Other examples include taking words co-occurrences in sentences and documents into consideration [53] and extending the model to support multi-languages [7, 20, 33, 51]. In the field of model learning, the author in [30] developed a new way which determines the parameter in Dirichlet distribution by other related elements such as author and date. Some algorithms and methods to improve the efficiency of LDA have been developed in [2, 32, 43, 49].

2.4 Vector Space Model

Vector space model (VSM), represents documents as vectors where each vector can be viewed as a point in a multi-dimensional space. And it uses frequency to discover semantic information within a collection of documents. The basic idea in VSM is that in space the closer the two points are, the more semantic similarity they are sharing and vice versa.

VSM was first developed in the SMART Information Retrieval System [41]. It was applied to find correlations between queries and documents and sorting the results according to their distances to the original queries. Due to its excellent performance, it has been extended to many other semantic applications in natural language processing field, for example, words correlations are derived from vector-based word meaning representations. And word relation vectors are devised to discover the similarities between words.

There are 3 main categories of VSM models according to their matrix types, term-document matrix, word-context matrix, and pair-pattern matrix [47]. In term-document matrix, a document collection is represented as a matrix where rows correspond to terms and columns correspond to documents. Values in the matrix are frequencies of terms appearing in documents. Vector similarities in the matrix indicate correlations of documents. The word-content matrix is based on the idea that words that appear in the same context tend to have similar meanings. The context can be in various forms, such as words, phrases, sentences, chapters or documents. The semantic similarity of two words is denoted by the similarity of corresponding word vectors in the matrix. The pair-pattern matrix is designed to discover the similarities of patterns. In the matrix, rows are pairs of words and columns are patterns associated with the appearances of word pairs. The word pair relations can be calculated from the columns of the matrix.

2.4.1 Weighting Methods and Similarity Measurement

Many methods have been developed to weight values in a VSM to improve its performance. Among these methods, the most popular one is tf-idf (term frequency-inverse document frequency) [21, 42]. In this method, high weights are given to terms that frequently appear in one document, at the mean time rarely occur in other documents in a collection. This helps to control terms that are used more frequently in general. In a document collection c with total of N documents, the tf (term frequency) is the number of occurrences of term t in document d. And idf (inverse document frequency) for t is defined as follows:

$$idf_t = log\frac{N}{tf_t} \tag{9}$$

From the equation we can notice that for terms that appear frequently, the idf value will be low, while for rare terms, the idf will be high. Combine term frequency and inverse document frequency, the weighting score for t in d given by tf-idf is:

$$tf - idf_{t,d} = tf \times idf \tag{10}$$

The similarity between two points in space is often measured by the cosine value of their frequency vectors. Assume two points in a n-dimensional space $\mathbf{a} = a_1, a_2, ..., a_n$ and $\mathbf{b} = b_1, b_2, ..., b_n$, the cosine similarity is:

$$sim(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{||\mathbf{a}||||\mathbf{b}||} \tag{11}$$

Thus, the similarity between two vectors is their inner product divided by their Euclidean norms.

2.4.2 Latent Semantic Analysis (LSA)

Latent Semantic Analysis (LSA) is a widely known VSM model which was introduced in 1990 in [8]. It applies singular value decomposition (SVD) to the weighted term-document matrix to obtain representations for documents in a low-dimensional space. Thus, this approach can capture most correlations in a collection while saves the storage space.

The process of SVD is as follows: suppose a document d has a term-document representation matrix D. Then the SVD transformation for D in LSA is:

$$D = USV^T \tag{12}$$

where U and V contain orthonormal vectors and S is diagonal. Then the document-document matrix $A = D^T D$ and the word-word matrix $B = DD^T$ after SVD are represented as:

$$D^T D = (USV^T)^T USV^T = VS^T SV^T \tag{13}$$

$$DD^T = USV^T (USV^T)^T = USS^T U^T \tag{14}$$

Matrices $S^T S$, SS^T are diagonal, therefore, V contains the eigenvectors of $D^T D$, while U contains the eigenvectors of DD^T. The values in S are the eigenvalues square roots which denote the contributions made by the eigenvectors. Since some of them are too small, they can be ignored. Thus, in LSA, only the first k important values are kept. Therefore, S, U, and V are transformed to S_k, U_k, V_k by keeping the first k important values. Now the lower space approximation for term-document matrix D is:

$$D_k = U_k S_k V_k^T \tag{15}$$

Therefore, in LSA, the similarities between words, queries and document can be derived by first performing the SVD on them, then calculating cosine values between their angles.

CHAPTER 3

Hierarchical Statistical Document Model

In the past few years, the improvement of the state of the art concerning document modeling has been based on three main groups of approaches [19]. The first group of approaches has been concerned with the improvement of current learning techniques. The second one has been based on the development of better features. The third one focused on the integration of prior information about the relationship between document classes. The technique that we shall propose in this chapter belongs to the third group, since our main goal here is to take advantage of the hierarchical relationship usually present between classes. Indeed, the automatic extraction of a given document topic and semantic information about a given word meaning generally involves a hierarchy of a large number of classes. These classes can be viewed as document categories. The hierarchy encodes crucial information that should be exploited when learning a given model. Thus, we propose here the extension of the log-bilinear model to incorporate the fact that document classes are generally hierarchical. In this chapter, we start by reviewing the basic log-bilinear model and then we generalize it to encode hierarchies.

3.1 Log-bilinear Document Model

In [25], the authors have introduced a log-bilinear document model which learns the semantic word vectors from term-document data. In this model, a document is represented as a distribution of conditionally independent words given a parameter θ. Then, the probability of a document is given by:

$$p(d) = \int p(d, \theta) = \int p(\theta) \prod_{i=1}^{N} p(w_i|\theta) d\theta \tag{1}$$

where d is a document, N is the total number of words in d and w_i represents each word in d. A Gaussian prior is considered for θ.

The model uses bag-of-words representation to describe a document in which words sequences appear in an exchangeable way. The fixed vocabulary set is denoted as V and has a size of $|V|$. Each word is represented by a $|V|$- dimensional vector where only one element is equal to 1 and all the others are equal to 0 (i.e. one-hot vector). The word conditional distribution $p(w|\theta)$ in the document is defined by a log-linear model with parameters R and b. The word representation matrix is $R \in \Re^{\beta \times |V|}$ and contains the β-dimensional vector representation $\phi_w = Rw$ of each word in the vocabulary set. Therefore, the representation, ϕ_w, of each word is the corresponding column in R. Also, θ is a β-dimensional vector which works as a weighting component for the word vector representation. Moreover, the word frequency differences are captured via a parameter b_w. Given all these parameters, the log-bilinear energy assigned to each word is:

$$E(w; \theta, \phi_w, b_w) = -\theta^T \phi_w - b_w \tag{2}$$

Therefore, the word distribution using softmax is given by:

$$p(w|\theta; R, b) = \frac{\exp(-E(w; \theta, \phi_w, b_w))}{\sum_{w' \in V} \exp(-E(w'; \theta, \phi_{w'}, b_{w'}))} = \frac{\exp(\theta^T \phi_w + b_w)}{\sum_{w' \in V} \exp(\theta^T \phi_{w'} + b_{w'})} \tag{3}$$

It is noteworthy that the previous model can only find semantic information at the document level.

3.2 Hierarchical Statistical Document Model

In real-world applications, online texts are often classified into categories with respect to their themes. Thus, these texts usually have a hierarchical structure. Moreover, words are hierarchical by nature, since they may relate to different other words at different categories. For example, word “attack” may relate to “bomb” in crime category, while it may relate to “virus” in cyber crime category. In this subsection, we extend the log-bilinear document model to take hierarchical structures into account. The main goal is to discover semantic information such as word relations at each level of the hierarchical structure.

3.2.1 Model Specification

Modeling a collection of documents into different levels can be achieved by building a probabilistic model for each node in the hierarchical structure. Suppose that we have a node m, which has a total number of N_k children denoted as m_k. Each child node is considered to be a documents collection composed of N_{tk} documents which are supposed to be conditionally independent given a variable θ_{jk}. Thus, the probability of node m can be written as the following:

$$p(m) = \prod_{k=1}^{N_k} \prod_{j=1}^{N_{tk}} \int p(\theta_{jk}) p(d_{jk}|\theta_{jk}) d\theta_{jk} \tag{4}$$

where d_{jk} denotes the jth document in the child node m_k, θ_{jk} is a mixing variable corresponding to document d_{jk}, and $p(\theta_{jk})$ is a Gaussian prior. Each document consists of conditionally independent distributed words:

$$p(d_{jk}|\theta_{jk}) = \prod_{i=1}^{N_{wtk}} p(w_{ijk}|\theta_{jk}) \tag{5}$$

where N_{wtk} is the total number of words in document d_{jk}, which actually belongs to m_k, and w_{ijk} denotes the words inside the document. By combining equations 4 and 5, we obtain the distribution of the node m:

$$p(m) = \prod_{k=1}^{N_k} \prod_{j=1}^{N_{tk}} \int p(\theta_{jk}) \prod_{i=1}^{N_{wtk}} p(w_{ijk}|\theta_{jk}) d\theta_{jk} \tag{6}$$

In the equation above, the p.d.f for each word, $p(w_{ijk}|\theta_{jk})$, is defined by Equation 3 in the previous section. It is worth mentioning that the model can also be applied to classify nodes which are at the same level of the hierarchical collection. This can be achieved by treating each node as an individual document containing words from all the documents it consists of. Therefore, the model can be trained to use parameter θ to distinguish each node from the others which are its siblings.

3.2.2 Model Learning

The model can be learned by maximizing the probability of observed data at each node. The parameters are learned by iteratively maximizing $p(m)$ with respect to θ, word representation R and word frequency bias b:

$$\hat{\theta}, \hat{R}, \hat{b} = \max_{\theta, R, b} \prod_{k=1}^{N_k} \prod_{j=1}^{N_{tk}} \int p(\theta_{jk}) \prod_{i=1}^{N_{wtk}} p(w_{ijk}|\theta_{jk}) d\theta_{jk} \tag{7}$$

Therefore, the log-likelihoods for θ_{jk}, and for R and b are:

$$L(\theta_{jk}) = \sum_{j=1}^{N_{tk}} (\sum_{i=1}^{N_{wtk}} \log(p(w_{ijk}|\theta_{jk})) - \lambda \theta_{jk}^2) \tag{8}$$

$$L(R, b) = \sum_{k=1}^{N_k} \sum_{j=1}^{N_{tk}} \log(p(\theta_{jk})) \sum_{i=1}^{N_{wtk}} \log(p(w_{ijk}|\theta_{jk})) \tag{9}$$

where λ is a scale parameter of the Gaussian. We take partial derivative with respect to θ_{jk} in Equation 8, to get the gradient:

$$\nabla_{\theta_{jk}} = \frac{\partial L(\theta_{jk})}{\partial \theta_{jk}} = \sum_{i=1}^{N_{wtk}} (\phi_{w_{ijk}} - \sum_{w' \in V} p(w'|\theta_{jk}) \phi_{w'}) - 2\lambda \theta_{jk} \tag{10}$$

Then, we take partial derivative with respect to R and b in Equation 9. For each column R_v of the representation matrix, the gradient ∇_{R_v} is:

$$\nabla_{R_v} = \frac{\partial L(R, b)}{\partial R_v} = \sum_{k=1}^{N_k} \sum_{j=1}^{N_{tk}} \sum_{i=1}^{N_{wtk}} (N_{w_v} \theta_{jk} - N_{wtk} p(w_v|\theta_{jk}) \theta_{jk}) \tag{11}$$

And the gradient for b is:

$$\nabla_{b_v} = \frac{\partial L(R, b)}{\partial b_v} = \sum_{k=1}^{N_k} \sum_{j=1}^{N_{tk}} \sum_{i=1}^{N_{wtk}} (N_{w_v} - N_{wtk} p(w_v|\theta_{jk})) \tag{12}$$

Therefore, at each step of the iteration, θ, R and b are updated as:

$$\theta_{jk}^{t+1} = \theta_{jk}^{t} + \alpha \nabla_{\theta_{jk}} \tag{13}$$

$$R_v^{t+1} = R_v^t + \alpha \nabla_{R_v} \qquad b_v^{t+1} = b_v^t + \alpha \nabla_{b_v} \tag{14}$$

Thus, the parameters are optimized by moving in the direction of the gradient. The step size of the movement is indicated by α. The procedure of estimating the model's parameters is based on alternatively optimizing the values of θ, R, and b using Newton's method. It first optimizes θ for each collection child with R and b fixed. Afterwards, we optimize word representation R and bias b with θ fixed. We repeat these two steps until convergence. The complete learning procedure is shown in Algorithm 1.

The classification of words using our model can be performed by considering each node at the same level of the hierarchical structure as a document with a parameter θ. Suppose that we have N_c nodes at the same level, and each node is denoted as c which has parameter θ_c. Then the probability of the occurrence of word w given the parameter of our model is:

Algorithm 1 Model Learning Algorithm

```
1: Initialize the values of parameters $\theta$ , $R$, and $b$ with randomly generated numbers, set the step
   size ($\alpha = 1e - 4$) and iteration convergence criteria (maximum iteration number $MaxIter =$
   $1000$ and evaluation termination value $TermVal = 1e - 7$).
2: Repeat
3: Estimate $\theta_{jk}$ at each node using Eq. 13.
4: Optimize $R$ and $b$ using Equations in  14.
5: Until one of the convergence criteria is reached (The iteration exceeds $MaxIter$ or the change
   of the parameters values is less than $TermVal$)
```

$$p(w|\theta_c; R, b) = \frac{\exp(\theta_c^T \phi_w + b_w)}{\sum_{w' \in V} \exp(\theta_c^T \phi_{w'} + b_{w'})} \tag{15}$$

Thus, by learning the model parameters from the learning steps, we can derive the occurrence probability of word w given the N_c nodes which are at the same level of the hierarchical structure.

CHAPTER 4

Experimental Results

In this section, we verify our proposed hierarchical statistical document model with two challenging tasks. The first one is to find semantically related words for a query word at each level of a collection of documents characterized by a hierarchical structure. We validate our model using two real world data sets collected from different Internet web sites. The second test is to show the model's performance on word classification. Our experiments have been performed on a 2.70GHz Intel i7 machine (4GB RAM, 64-bit operating system) using Matlab version R2010b.

4.1 Finding Semantically Related Words

4.1.1 Data Sets and Similarity Measurement

In this experiment, data sets are collections of web pages gathered from the Internet. First, data are retrieved from corresponding web sites. Then the plain text of each web page is extracted. Afterwards, data are pre-processed by consulting each word property with WordNet to filter stop words (e.g. ”the”, ”and”, and ”or” etc.) and non English words. Only nouns, verbs, adjectives and adverbs are kept. Furthermore, the nouns and verbs are converted to their roots, for example ate is changed to eat, and cats is transformed to cat. This can help us to eliminate the redundancy of a root word presented in multiple formats. In our proposed model, the related words are found by calculating the cosine similarities between words from the word representation vectors ϕ, which are derived from the representation matrix R. Therefore, for words w_1 and w_2, with representation vectors ϕ_1 and ϕ_2, the similarity is:

$$Similarity(w_1, w_2) = \frac{Rw_1 \cdot Rw_2}{||Rw_1||||Rw_2||} = \frac{\phi_1 \cdot \phi_2}{||\phi_1||||\phi_2||} \tag{1}$$

4.1.2 Experiment on Fraud Articles

In this subsection, we examine our model with a collection of documents which has a 3-levels hierarchical structure. The data are collected from two of the categories in an anti-fraud forum, where articles are classified according to their themes. The two categories are "advanced-fee spam" and "lottery spam". The structure of this collection of documents is demonstrated in Figure 4.1. As we can see from this figure, the root node is "Fraud" which contains 597 articles. It has two categories "Lottery" and "Advanced-fee" as its children. Category "Lottery" contains 296 documents while "Advanced-fee" contains 301 documents. In this experiment, the first 2500 frequently appeared words in nodes are selected to construct the vocabulary set.

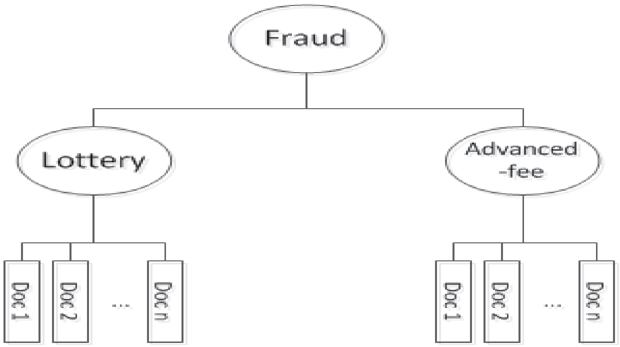

Figure 4.1: The hierarchical structure of "Fraud" category.

Tables 4.1, 4.2 and 4.3 show part of the most frequent words as well as their semantically related words when using cosine similarity scores computed at these three nodes. The results in these tables show clearly that our model has a good performance in finding different related words in this 3-levels hierarchical documents collection.

Table 4.1: Semantically related words at node “Fraud”

Word	account	score	prize	score	lottery	score	bank	score
Similar Words	transfer	0.866	winners	0.937	congratulations	0.961	fund	0.776
	foreign	0.848	lottery	0.935	prize	0.935	account	0.771
	bank	0.771	lucky	0.878	winner	0.915	transfer	0.729
	fund	0.724	ticket	0.868	lucky	0.900	beneficiary	0.703
			bonus	0.713				
Word	transfer	score	win	score	transaction	score	telephone	score
Similar Words	account	0.866	congratulations	0.923	expenses	0.884	fax	0.829
	foreign	0.865	prize	0.861	private	0.835	name	0.757
	bank	0.729	award	0.804	business	0.820	address	0.727
	commission	0.702	lottery	0.913	risk	0.798	number	0.701
			ticket	0.856				

Table 4.2: Semantically related words at node “Lottery”

Word	prize	score	lottery	score	inform	score	ticket	score
Similar Words	total	0.734	win	0.897	congratulations	0.847	number	0.797
	requirements	0.723	number	0.875	award	0.821	draw	0.784
	winner	0.705	draw	0.850	address	0.798	lottery	0.784
	conduct	0.704	hold	0.844	tel	0.779	win	0.754
			ticket	0.784	date	0.753	serial	0.750
Word	phone	score	selection	score	british	score	sell	score
Similar Words	occupation	0.794	random	0.860	great	0.948	ticket	0.842
	country	0.893	database	0.856	pounds	0.930	exclusive	0.799
	age	0.761	computerized	0.826	kingdom	0.887	pick	0.767
	sex	0.743	exclusive	0.730	London	0.756	automated	0.721
	name	0.734	list	0.711	uk	0.753	advanced	0.714

4.1.3 Experiment on Wikipedia Dataset

In this subsection, The data set is composed of web pages gathered from Wikipedia. The data is obtained via “Wikipedia Export”, which allows to retrieve web pages from the database in specific categories. Afterwards, data are pre-processed using method in Section 4.1.1. Here, we report our experimental results on words learned under the “Crime” category in Wikipedia. The structure of this collection of documents is displayed in Figure 4.2. As we can see from this figure, the root node is “Crime”, which contains 5372 documents. It has two children, namely “Fraud” and

Table 4.3: Semantically related words at node “Advanced-fee”

Word	account	score	transaction	score	pay	score	telephone	score
Similar Words	bank	0.820	account	0.745	payment	0.763	information	0.815
	transaction	0.745	have	0.706	contract	0.752	fax	0.792
	transfer	0.737	require	0.705	bankers	0.731	number	0.764
					ministry	0.722	share	0.751
					approval	0.708		
Word	approval	score	investigation	score	properties	score	reminder	score
Similar Words	application	0.813	discover	0.765	buy	0.749	notification	0.990
	execute	0.759	documents	0.711	undergo	0.747	calendar	0.937
	supply	0.763			issues	0.725	fowarding	0.792
	ministry	0.713			house	0.709	expiration	0.761
	pay	0.708				0.753	proposal	0.703

Table 4.4: Semantically related words at node “Crime”.

Word	shoot	score	attack	score	murder	score	bury	score
Similar Words	kill	0.881	wound	0.738	kill	0.830	cremate	0.820
	ambush	0.810	bomb	0.724	mutilate	0.757	die	0.760
	gun	0.762	overpower	0.706	confess	0.739	burn	0.712
	fire	0.761			assassinate	0.732	exhume	0.708
	wound	0.750			stab	0.732	survive	0.706
Word	invest	score	disappear	score	marry	score	lie	score
Similar Words	trade	0.818	vanish	0.840	move	0.848	hear	0.727
	promise	0.759	miss	0.704	bear	0.789	tell	0.718
	buy	0.742			die	0.781		
					emigrate	0.759		
					divorce	0.725		

“Murder”. The node “Fraud” contains 4341 documents and the node “Murder” contains 1391 documents. Both of them have 14 nodes as children. We report the results found by our model at these three nodes. The most frequently used 2500 words in our nodes are selected to build the vocabulary set. Tables 4.4, 4.5 and 4.6 show part of the most frequent words as well as their semantically related words when using cosine similarity scores computed at these three nodes. The results in these tables demonstrate that our model performs well on finding different related words.

Table 4.5: Semantically related words at node “Murder”.

Word	shoot	score	attack	score	murder	score	disappear	score
Similar Words	kill	0.906	injure	0.870	kill	0.906	force	0.813
	die	0.878	wound	0.843	try	0.863	kidnap	0.806
	fire	0.820	stop	0.765	commit	0.750	detain	0.774
	attempt	0.807	coordinate	0.708	die	0.845	miss	0.770
	murder	0.737			shoot	0.737	confirm	0.711
Word	assassinate	score	fire	score	injure	score	investigate	score
Similar Words	condemn	0.814	wound	0.838	wound	0.904	conclude	0.767
	oppose	0.807	shoot	0.821	attack	0.870	solve	0.763
	execute	0.712	injure	0.773	fire	0.773	examine	0.709
	fail	0.710	occur	0.748	occur	0.752	indicate	0.705
	escape	0.704	surrender	0.723	explode	0.708	file	0.704

Table 4.6: Semantically related words at node “Fraud”.

Word	invest	score	disappear	score	marry	score	lie	score
Similar Words	resell	0.782	convince	0.738	divorce	0.981	reveal	0.812
	own	0.773	vanish	0.734	bear	0.933	discover	0.809
	collapse	0.762	pose	0.731	widow	0.847	admit	0.745
	trade	0.739	notice	0.723	inherit	0.756	tell	0.724
	promise	0.718	try	0.718	emigrate	0.726	confess	0.701
Word	bury	score	identify	score	examine	score	divorce	score
Similar Words	marry	0.774	indicate	0.815	prove	0.745	widow	0.842
	burn	0.728	provide	0.758	conclude	0.734	marry	0.826
	die	0.715	employ	0.752	verify	0.731	graduate	0.772
	inherit	0.713	report	0.732			remarry	0.742
	survive	0.702	demonstrate	0.709				

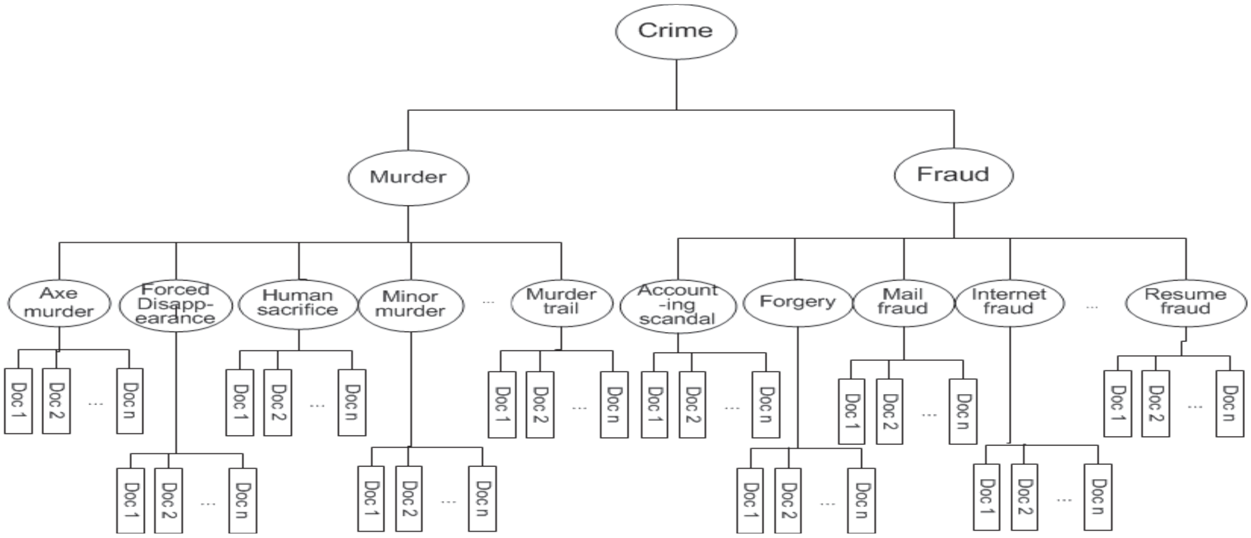

Figure 4.2: The hierarchical structure of “Crime” category.

4.2 Word Classification

In this subsection, we investigate the performance of our model on word classification problem. The data set used in this experiment is collected from “The Thesaurus” web site. In this web site, words are classified into 6 categories:

1. Words expressing “Abstract” relations
2. Words related to Space
3. Words related to Matter
4. Words related to the Intellectual Faculties; Formation and Communication of Ideas
5. Words related to the Voluntary Powers; Individual and Intersocial Volition
6. Words related to the Sentiment and Moral Powers

Each category has many sub classes. The data that we use in our experiment here are from the second and third categories which contain 137 and 136 documents, respectively. The hierarchical structures of the data in both categories are shown in Figures 4.3 and 4.4. 10-fold cross validation is performed on these two categories by randomly splitting into 10 groups the 9749 and the 7617 words in the vocabularies of the second and third categories, respectively. For each word, we try to find the correct corresponding document to which it belongs using the parameter θ. The

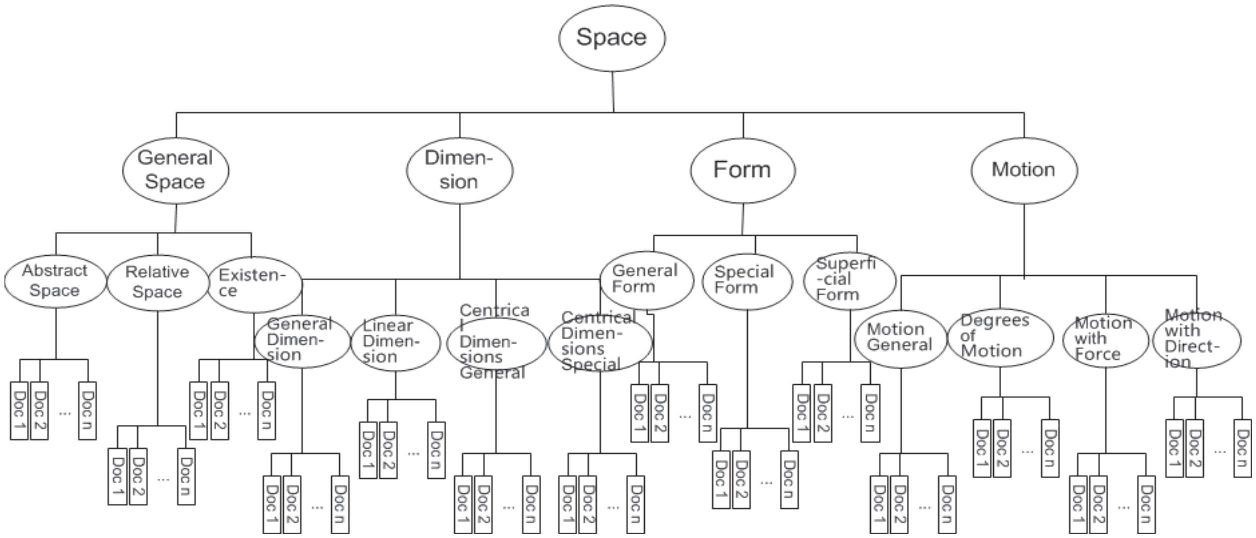

Figure 4.3: Hierarchical structure of the "Words related to space" category.

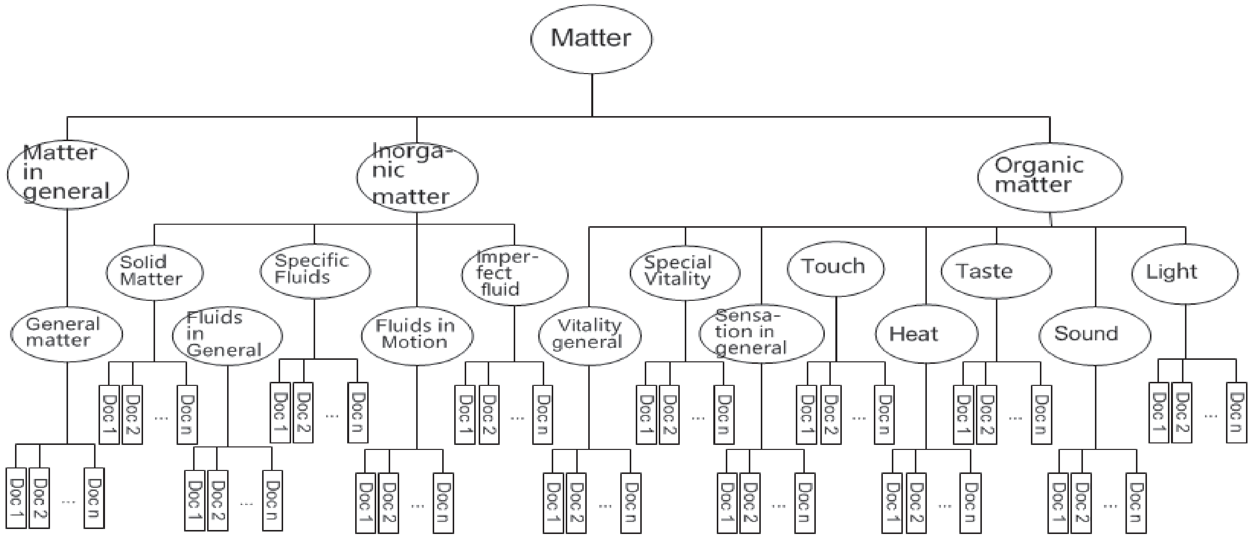

Figure 4.4: Hierarchical structure of the "Words related to matter" category.

Table 4.7: Results obtained for the word classification task.

	Accuracy (%)	
Data	category “words related to space”	category “words related to matter”
Flat Model	77.23	78.04
Our Model	81.76	82.17

Table 4.8: Statistical significance tests on the accuracy scores.

	flat model		our model		t-value	critical t-value ($\alpha = 0.05$)
data	mean	σ	mean	σ		
“words related to space”	77.23	3.67	81.76	2.10	3.39	2.1009
“words related to matter”	78.04	3.28	82.17	2.57	3.13	2.1009

probability threshold is set to 0.7. The classification results in terms of accuracy, of our model and the original flat model, are shown in Table 4.7. From this table, we can see that the accuracy scores of the original flat model are 77.23 and 78.04 while for our model, they are 81.76 and 82.17. The improvement is due to the property of our hierarchical model, since we describe the data as a tree structure where the number of classes is reduced at each estimation. Moreover, we performed a significance student t-test, with a confidence level of 95% , on the obtained scores at each cross validation. The results are shown in Table 4.8. According to these results, we can say that the difference in accuracy between our model and the flat one is statistically significant.

CHAPTER 5

Conclusion And Future Work

In this thesis, we have presented a statistical document model to analyze collections of documents having hierarchical structures. The goal of our work is to build a model which takes the hierarchical nature of documents collection into account to describe textual data, so that hidden semantic information can be found by exploring and analyzing them. Our model can be viewed as an extension of the flat log-bilinear approach. In our approach, each node is described using statistical probabilities, and a word representation matrix which contains multi-dimensional word representation vectors is derived from term-document information for each node. Thus, semantic relation can be calculated from the cosine similarity of word representation vectors. Our approach has been validated by conducting experiments involving real data gathered from different Internet web sites. Results, which have concerned the discovery of semantically related words, showed that our model has a good performance in extracting different semantically related words at each node of the collection of documents. The ability of word classification was investigated by comparing the performances of flat log-bilinear model and our model in terms of accuracy. Results have shown that our approach has a better score than the flat model. Future potential research works could be devoted to the extension of the model to online settings (e.g. adding, fusing, or deleting a node) to take into account the dynamic nature of the Web (i.e. new documents are added and others are deleted regularly on the Web). Another promising future work could be dedicated to the consideration of other languages (e.g. French, Arabic, Spanish, Chinese, etc.) for validation purposes.

List of References

[1] Andrzejewski, D., Zhu, X., Craven, M.: Incorporating domain knowledge into topic modeling via dirichlet forest priors. In: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 25–32. ICML '09, ACM, New York, NY, USA (2009)

[2] Asuncion, A., Smyth, P., Welling, M.: Asynchronous distributed learning of topic models. Graphical Models 21(1), 1–8 (2008)

[3] Blei, D.M., Lafferty, J.D.: A correlated topic model of Science. Annals of Applied Statistics 1(1), 17–35 (Aug 2007)

[4] Blei, D.M., McAuliffe, J.D.: Supervised topic models. Advances in Neural Information Processing Systems 20(2), 1–8 (2010)

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

[6] Boyd-Graber, J., Blei, D., Zhu, X.: A topic model for word sense disambiguation. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). pp. 1024–1033 (2007)

[7] Boyd-Graber, J., Blei, D.M.: Multilingual topic models for unaligned text. In: Proceedings

of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. pp. 75–82. UAI '09, AUAI Press, Arlington, Virginia, United States (2009)

[8] Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science 41(6), 391–407 (1990)

[9] Denning, P.J., Denning, D.E.: Discussing cyber attack. Communications of the ACM 53(9), 29–31 (Sep 2010)

[10] Dietz, L., Bickel, S., Scheffer, T.: Unsupervised prediction of citation influences. In: Proceedings of the 24th international conference on Machine learning. pp. 233–240. ICML '07, ACM (2007)

[11] Dumais, S., Chen, H.: Hierarchical classification of web content. In: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval. pp. 256–263. SIGIR '00, ACM (2000)

[12] Erosheva, E., Fienberg, S., Lafferty, J.: Mixed-membership models of scientific publications. Proceedings of the National Academy of Sciences of the United States of America 101(Suppl 1), 5220–5227 (2004)

[13] Franklin, J., Paxson, V., Perrig, A., Savage, S.: An inquiry into the nature and causes of the wealth of internet miscreants. In: Proceedings of the 14th ACM conference on Computer and communications security. pp. 375–388. ACM CCS '07, ACM, New York, NY, USA (2007)

[14] Gerrish, S.M., Blei, D.M.: A language-based approach to measuring scholarly impact. In: International Conference on Machine Learning. pp. 375–382 (2010)

[15] Goel, S.: Cyberwarfare: connecting the dots in cyber intelligence. Commun. ACM 54(8), 132–140 (Aug 2011)

[16] Goldwater, S., Griffiths, T.L., Johnson, M.: Interpolating between types and tokens by estimating power-law generators. In: Neural Information Processing Systems. pp. 459–466 (2005)

[17] Griffiths, T.L., Steyvers, M.: Finding scientific topics. Proceedings of the National Academy of Sciences of the United States of America 101, 5228–5235 (Apr 2004)

[18] Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. pp. 50–57. SIGIR '99 (1999)

[19] Hofmann, T., Cai, L., Ciaramita, M.: Learning with taxonomies: Classifying documents and words. In: Proceedings of Synatx, Semantics and Statistics NIPS Workshop (2003)

[20] Jagarlamudi, J., Daumé, H.: Extracting multilingual topics from unaligned comparable corpora. In: Proceedings of the 32nd European conference on Advances in Information Retrieval. pp. 444–456. ECIR '10, Springer-Verlag, Berlin, Heidelberg (2010)

[21] Jones, K.S.: A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation 28(1), 11–21 (1972)

[22] Lacoste-Julien, S., Sha, F., Jordan, M.I.: Disclda: Discriminative learning for dimensionality reduction and classification. Advances in Neural Information Processing Systems (NIPS) 21 (2008)

[23] Lau, R., Xia, Y.: Latent text mining for cybercrime forensics. International Journal of Future Computer and Communication 2(4), 368–371 (2013)

[24] Lin, D., Pantel, P.: Dirt - discovery of inference rules from text. In: In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 323–328 (2001)

[25] Maas, A., Ng, A.: A Probabilistic Model for Semantic Word Vectors. In: Deep Learning and Unsupervised Feature Learning Workshop NIPS 2010. vol. 10 (2010)

[26] Mackay, D., Peto, L.: A hierarchical dirichlet language model. Natural Language Engineering 1(3), 289–308 (1995)

[27] Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval, vol. 1. Cambridge University Press (2008)

[28] Mimno, D., Li, W., McCallum, A.: Mixtures of hierarchical topics with pachinko allocation. In: Proceedings of the 24th international conference on Machine learning. pp. 633–640. ICML '07, ACM, New York, NY, USA (2007)

[29] Mimno, D., McCallum, A.: Mining a digital library for influential authors. In: Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries. pp. 105–106. JCDL '07, ACM (2007)

[30] Mimno, D.M., McCallum, A.: Topic models conditioned on arbitrary features with dirichlet-multinomial regression. Computing Research Repository abs/1206.3278 (2012)

[31] Nakov, P.I., Hearst, M.A.: Ucb: System description for semeval task 4. In: In Proceedings of the Fourth International Workshop on Semantic Evaluations. pp. 366–369 (2007)

[32] Nallapati, R., Cohen, W., Lafferty, J.: Parallelized variational em for latent dirichlet allocation: An experimental evaluation of speed and scalability. In: Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on. pp. 349–354 (Oct)

[33] Ni, X., Sun, J., Hu, J., Chen, Z.: Cross lingual text classification by mining multilingual topics from wikipedia. In: Proceedings of the fourth ACM international conference on Web search and data mining. pp. 375–384. WSDM '11, ACM, New York, NY, USA (2011)

[34] Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using em. Journal of Machine Learning Research 39(2-3), 103–134 (May 2000)

[35] Pantel, P., Lin, D.: Discovering word senses from text. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 613–619. KDD '02, ACM (2002)

[36] Ramage, D., Dumais, S., Liebling, D.: Characterizing microblogs with topic models. In: In Proceedings of the 4th International AAAI Conference on Weblogs and Social Media (ICWSM 2010. pp. 130–137 (2010)

[37] Ramage, D., Hall, D., Nallapati, R., Manning, C.D.: Labeled lda: a supervised topic model for credit attribution in multi-labeled corpora. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1. pp. 248–256. EMNLP-CoNLL '09, Association for Computational Linguistics, Stroudsburg, PA, USA (2009)

[38] Rapp, R.: Word sense discovery based on sense descriptor dissimilarity. In: Proceedings of the Ninth Machine Translation Summit. pp. 315–322 (2003)

[39] Rosen-Zvi, M., Griffiths, T., Steyvers, M., Smyth, P.: The author-topic model for authors and documents. In: Proceedings of the 20th conference on Uncertainty in artificial intelligence. pp. 487–494. UAI '04, AUAI Press, Arlington, Virginia, United States (2004)

[40] Ruiz, M.E., Srinivasan, P.: Hierarchical text categorization using neural networks. Information Retrieval 5(1), 87–118 (2002)

[41] Salton, G.: The SMART Retrieval System; Experiments in Automatic Document Processing. Prentice-Hall, Incorporation, Upper Saddle River, NJ, USA (1971)

[42] Shannon, C.E.: A mathematical theory of communication. SIGMOBILE Mobile Computing and Communications Review 5(1), 3–55 (Jan 2001)

[43] Smola, A., Narayanamurthy, S.: An architecture for parallel topic models. Proceedings of the VLDB Endowment 3(1-2), 703–710 (Sep 2010)

[44] Teh, Y.W.: A hierarchical bayesian language model based on pitman-yor processes. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics. pp. 985–992. ACL-44, Association for Computational Linguistics, Stroudsburg, PA, USA (2006)

[45] Turney, P.D.: Similarity of semantic relations. Computational Linguistics 32(3), 379–416 (Sep 2006)

[46] Turney, P.D., Littman, M.L., Bigham, J., Shnayder, V.: Combining independent modules to solve multiple-choice synonym and analogy problems. Computing Research Repository cs.CL/0309035 (2003)

[47] Turney, P.D., Pantel, P.: From frequency to meaning : Vector space models of semantics. Journal of Artificial Intelligence Research 37(1), 141–188 (2010)

[48] Xing, W., Croft, W.B.: Lda-based document models for ad-hoc retrieval. In: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 178–185. SIGIR ’06, ACM (2006)

[49] Yan, F., Xu, N., Qi, Y.: Parallel inference for latent dirichlet allocation on graphics processing units. Advances in Neural Information Processing Systems 22, 2134–2142 (2009)

[50] Zhang, D., Lee, W.S.: Web taxonomy integration using support vector machines. In: WWW. pp. 472–481 (2004)

[51] Zhao, B., Xing, E.P.: Bitam: bilingual topic admixture models for word alignment. In: Proceedings of the COLING/ACL on Main conference poster sessions. pp. 969–976. COLING-ACL ’06, Association for Computational Linguistics, Stroudsburg, PA, USA (2006)

[52] Zhou, D., Manavoglu, E., Li, J., Giles, C.L., Zha, H.: Probabilistic models for discovering e-communities. In: Proceedings of the 15th international conference on World Wide Web. pp. 173–182. WWW ’06, ACM, New York, NY, USA (2006)

[53] Zhu, J., Xing, E.P.: Conditional topic random fields. International Conference on Machine Learning ACM (2010)