Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 675234, 11 pages
http://dx.doi.org/10.1155/2014/675234

*Research Article*

# Constructing Topic Models of Internet of Things for Information Processing

**Jie Xin,[1,2] Zhiming Cui,[1] Shukui Zhang,[1,3] Tianxu He,[1] Chunhua Li,[1] and Haojing Huang[1]**

[1] *The Institute of Intelligent Information Processing and Application, Soochow University, Suzhou 215006, China*
[2] *Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou 215006, China*
[3] *State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China*

Correspondence should be addressed to Shukui Zhang; zhangsk2000@163.com

Received 25 May 2014; Accepted 18 June 2014; Published 9 July 2014

Academic Editor: Juncheng Jia



*Internet of Things* (IoT) is regarded as a remarkable development of the modern information technology. There is abundant digital products data on the IoT, linking with multiple types of objects/entities. Those associated entities carry rich information and usually in the form of query records. Therefore, constructing high quality topic hierarchies that can capture the term distribution of each product record enables us to better understand users' search intent and benefits tasks such as taxonomy construction, recommendation systems, and other communications solutions for the future IoT. In this paper, we propose a novel *record entity topic model* (RETM) for IoT environment that is associated with a set of entities and records and a Gibbs sampling-based algorithm is proposed to learn the model. We conduct extensive experiments on real-world datasets and compare our approach with existing methods to demonstrate the advantage of our approach.

## 1. Introduction

The *Internet of Things* (IoT) is a novel paradigm that is rapidly developing in the scenario of modern wireless telecommunications and the information age. The ubiquitous of IoT technology especially in embedded devices has led to smart systems that affect people's daily life. The concept behind IoT is that a variety of objects around us can interact and work with each other to pursue common goals [1].

There are several popular services among IoT such as online shopping or goods recommendation systems. An ideal e-commerce website should correctly understand user's query intent and return satisfactory searching results as fast as possible; a favorite recommendation system should exactly predict user's individual preference, meaning the application must be cognizant of the topics behind every piece of product. The fact is tens of thousands of digital products/records within IoT constantly update and new records are generating every day. How to manage those data, how to define right records categories, those two questions will bring fresh challenges to the IoT industry.

A variety of existing works are dedicated to constructing topic or concept hierarchies from text data in the Internet. One innovative approach is to learn a topic model based on probability statistics. Traditional topic models treat each document as a bag of words and predict the topic through the word distributions. It has been widely used in fields such as taxonomy construction [2], concept extraction [3], citation analysis [4], and social network regularization [5]. LDA [6] and PLSA [7] are the most well-known ones and several different topic models are proposed as extensions. For example, Link-LDA [4] and HTM [8] are designed to deal with documents with hyperlinks, citations, and other forms of link information; author model (AM) [9] and author topic model (ATM) [10] deal with author related academic paper search; Link-PLSA-LDA [11] and MB-LDA [12] are designed to model documents like blogs or twitters.

Although these models perform well in certain domains, they are not appropriate in our case for two reasons. First, most of the records consist of short text like tweets, so it is sometimes difficult to discover the topic through only a few

tokens. But unlike tweets and other short texts, these records exist behind the query interface of certain websites and can only be obtained through keywords querying. Records from the same keyword query have some connections between each other and together they represent common topics. In other words, the word distribution of a single record can be regarded as a mixture latent topic distribution of current record and its "neighbor" records (records extracted from the same keyword queries), and the overall query records list might finally determine the topic of one record. Second, one product record usually is associated with plenty of attributes that describe the detailed features. For example, a book record has attributes like author and publisher; a movie record contains attributes like casts and director; a research paper record is associated with authors and publication. The truth is that almost any record is associated with some set of real-world entities, which is the foundation of IoT. Despite the fact that previous works have involved one or two types of entity relations in their topic models, like author relationship in ATM, they can only deal with certain domain documents. While our records associate with multiple objects, our topic model should be more general to conduct the topic analysis in diverse domains.

To further explain our goal, we need to know the term distributions for each entity or topic-entity pair. Specifically, let $z$, $e$, $w$ denote a topic, an entity, and a word, respectively; our goal is to design a topic model that can solve the problem like $P(w \mid z)$, $P(w \mid e)$, and $P(w \mid e, z)$.

The dependencies among topic, entity, and word could help the system to gain a better understanding of the model. For example, in a collection of computer science research papers, we could know the basic concept of data mining through the word distribution of $P(w \mid \text{Data Mining})$; we could know a specific author A's research interest in data mining area by analyzing the word distribution of $P(w \mid \text{A, Data Mining})$; we can further compare his data mining related work with other researchers like B in the same field by comparing $P(w \mid \text{A, Data Mining})$ with $P(w \mid \text{B, Data Mining})$.

In addition, topic model that involves entity distributions can be more accurate since it might avoid some mistakes caused by traditional models. In most previous works, they do not care for the distribution of $P(w \mid e, z)$ but directly assume that either $P(w \mid e, z) = P(w \mid e)$ or $P(w \mid e, z) = P(w \mid z)$ by introducing different types of conditional dependency relations among topics, entities, and words. For instance, both LDA and Link-LDA believe that $P(w \mid e, z)$ equals to $P(w \mid z)$; ATM, on the other hand, assumes that $P(w \mid e, z) = P(w \mid e)$. However, such "seems reasonable" assumptions may not always be correct in many cases. Recall the previous example, if $P(w \mid e, z) = P(w \mid e)$ holds, we can obtain that $P(w \mid \text{A, Data Mining}) = P(w \mid \text{A, Machine Learning})$, but this equation does not hold since authors seldom use the same terms in papers from different topics. Meanwhile, if $P(w \mid e, z) = P(w \mid z)$ holds, means $P(w \mid \text{A, Data Mining})$ is equal to $P(w \mid \text{B, Data Mining})$, but obviously different authors use different terms due to writing habits even doing the same research topic. To this point, it is really necessary for us to model $P(w \mid e, z)$ directly to find the correlation of words between a pair of an entity and a topic.

To trickle the above challenges, we propose a novel topic model to analyze the topics of querying records in IoT environment named *record entity topic model* (RETM). This model is a promotion of LDA model and argues that each record can be identified as a word distribution over a set of entity-topic pairs. Furthermore, it points that "neighbor" records should have some impact on the topic distribution of one record, since they come from the same query list so that they have great topical similarities. In this case, these improvements make the model more flexible and specific.

The main contributions of our work are as follows. (1) We studied an important practical problem of modeling the topics for IoT industry using the latent relationships among querying records. To the best of our knowledge, this is the first work to introduce such issue. (2) We improve our model by adding entity relative dependency and emphasize the correlation of entity-term and topic-term distributions. We also adopt Gibbs-sampling-based algorithm to learn the model. (3) We conduct complex experiments on real-world dataset and compare with some state-of-the-art topic models. Results show that our method performs well in certain aspect and has some advantages.

The rest of this paper is organized as follows. We begin by reviewing the related works in the next section. After that, we introduce the overall model by giving the problem statements and describing the generation process as well as how to inference and learning the model in Section 3. The experimental results are addressed in Section 4 and conclusions are drawn in Section 5.

## 2. Related Work

Latent topic modeling has become very prevalent as a completely unsupervised technique for topic discovery. PLSA (probabilistic latent semantic analysis) [7] and LDA (latent Dirichlet allocation) [6] are two most well-known models in large document collections. The key idea of them is to represent topics as distribution of words and the document is a mixture over hidden topics so that it can be processed in this lower-dimensional space. However, the disadvantage of the two models is that they treat each document as an independent one and ignore the obverse linking relationships, so a lot of researches have been done while making use of multiple typed links in different ways.

The original purpose of topic models is to process plain text of documents. Recently, more attention has been paid in respect of topic discovery with hypertexts or citation information. Link-LDA [4] is designed specifically for scientific publication with citations, which regards one document as a combination of a bag of words and a bag of citations. It provides an efficient model to understand the distribution of links. Dietz et al. [13] propose a topic model for citation analysis, in which the content of a citing document is assumed to be generated by mixing the topic distributions of the cited documents. It is the first time to study the relations

of citing paper and cited paper to a published paper. Link-PLSA-LDA [11] combines PLSA and LDA together into a single framework, which produces a bipartite graph to model the connections of citing documents to cited ones. Another model called HTM [8] argues that the content of a document is determined by its own topic distributions as well as by its cited documents', which is fairly similar to our model where we seek the background topics from the "neighbor" records. However, all the above models are particular designed for scientific publications, which are extremely different from querying records in structure and hardly applied to other domains.

In addition to review the hyperlinks between the documents, some researchers proposed models that concentrate on the relations of author. For instance, Mccallum [9] intends to model the author interests with a one-to-one correspondence between topic and authors in multilabeled documents, so each label could be represented as an entity. The ATM [10] integrates the authorship into the topic model, which addresses the task of modeling corpora annotated with the ids of multiple people who authored the documents. ArnetMiner [14] extends the aforementioned work by adding conference relations to its model. It tries to mine the academic social networks simultaneously from the aspects of papers, authors, and publication venues. Still all these models are limited to academic publications like mentioned above; in addition, they all treat $P(w \mid e, z) = P(w \mid z)$, which is usually not the case.

Some works pay attention to model short text documents like records in our case. For example, PatentMiner [15] is proposed to mine the topic of heterogeneous patent network involving several types of objects like companies, inventors, and technical contents. However, it is uniquely designed for patent topic discovery and cannot be used for more general records of IoT, and in addition, all the patent records are gathered from databases, which is quite different from querying records in our case. Wang et al. [16] successfully discover the topics from web tables by summarizing the content of each row with columns. His dominant theme is similar to our model since we also argue there is a background topic that impacts the distribution of single record topics. However, instead of using probabilistic topic model, his work adopts a Hearst pattern-based method. MB-LDA [12] is designed for topic mining of microblogs or tweets, which takes both contact relation and document relation into consideration. Nevertheless, the structure and relations of tweets are significantly different from the records we are talking about. One model that particular points out the importance of differentiating term distribution from entity distribution in a topic model is ETM [17], who can explicitly model $P(w \mid e, z)$. However, besides the previous explanations we made to illustrate the particular characteristics of our case, ETM has paid great attentions on determining the shared asymmetric Dirichlet priors. Conversely, our method tries to balance between entity distributions with topic distribution obtained from local topic or overall topics, so we decide to adopt symmetric Dirichlet priors, which is enough in our case. In addition, the two models have entirely different generative process.

## 3. Record Entity Topic Model

In this section, we firstly give the definition of RETM and then explain the generation process and finally provide a Gibbs-sampling-based learning algorithm to infer the model.

*3.1. Overview of the Problem.* Records we mentioned are obtained from keywords querying over the Internet. They are dynamic, highly qualified, and exist in the hidden databases. Each record consists of several attributes that describe all the associated features of one unique object. Most records involve multiple entities in the attributes, such as the attributes *director* and *production company* in Table 1. A collection of entity related records is the basic input of RETM.

*Definition 1.* Each record $r$ is associated with a term vector, where each $w_{d,i}$ is selected from a vocabulary database $W$ and an entity vector $E_d$, which is chosen from a set of entity of size $E$. Let a collection of $R$ records ($r \in R$) is determined by $R = \{\langle w_1, E_1\rangle, \langle w_2, E_2\rangle, \ldots, \langle w_R, E_R\rangle\}$.

The goal of this paper is to identify word patterns for each pair of an entity and a topic, that is, to find the precise word distribution over topic $z$ and entity $e$ to get $P(w \mid e, z)$, which also satisfies a multinomial distribution with parameter $\varphi_{e,z}$. The notation of the whole paper is present in Notation section. Note that the entity we used in our model is gained from some well-known corpus such as DBpedia [18] or Probase [19]. Unlike ETM or other topic models, we assume that all the entities are following a multinomial distribution $\delta_d$, which are not generated but can be easily calculated with the help of modern knowledge base.

*3.2. Generative Process of RETM.* Figure 1 reveals the graphic representation of RETM. The core idea of this model is that firstly, different entities can be represented by different word distributions, and the words used to describe an entity can be changed with the topic. For example, Guo Jingming is known as a famous writer in the *Book* domain, but he is often identified as the director of movie *Tiny Time* in the *Movie* domain. In other words, $P(w \mid e_m, z) \neq P(w \mid e_n, z)$ if $e_m \neq e_n$ and $P(w \mid e, z_m) \neq P(w \mid e, z_n)$ if $z_m \neq z_n$. Secondly, the topics of one record could be determined based on the current record itself or together with its "neighbor" records. For example, in Table 1, judging from one record, we might say that it is a science fiction movie, but combined with the remaining two records from the same query, we can say that all these records are American movies of extraterrestrial for sure. That is the reason why we particularly studied the overall background topic in RETM.

Algorithm 1 explains the generative process of RETM. Note that each record has its own topic, and we name them *local topics*; one record and its "neighbor" records from the identical query list together produce some topics and we name them *background topics*. They all originate in the same vocabulary source and there are $K_c$ local topics and $K_b$ background topics, respectively. Therefore, for each record $d$, multinomial distributions $\theta_d$ and $\theta'_d$ over topics can be drawn from a Dirichlet prior with $\alpha_0$. Suppose that the probability

Table 1: Example of query records from IMDB (movies).

| | | | | |
|---|---|---|---|---|
| Avatar | USA\| UK | James Cameron | Twentieth Century Fox Film Corporation | January 4, 2010 (China) |
| Star trek | USA\| Germany | J.J. Abrams | Paramount Pictures | May 15, 2009 (China) |
| Men in black 3 | USA\| United Arab Emirates | Barry Sonnenfeld | Columbia Pictures | May 25, 2012 (USA) |

```
(1)  foreach topic-entity pair (e, z) do
(2)    Draw topic-entity distribution φ_{e,z} ~ Dir(β)
(3)  endfor
(4)  foreach record d do
(5)    Draw entity distribution δ_d ~ Dir(α_1, E_d)
(6)    Draw local topic distribution θ_d ~ Dir(α_0)
(7)    Draw background distribution θ_d' ~ Dir(α_0)
(8)    foreach word i ∈ {1, 2, ..., N_d} do
(9)      Draw t_i ~ Bernoulli(p_z)
(10)     Draw e_{d,i} ~ Multinomial(δ_d)
(11)     if t_i = 1 then
(12)       Draw z_{d,i} ~ Multinomial(θ_d)
(13)       Draw w_{d,i} ~ Multinomial(φ_{e_{d,i},z_{d,i}})
(14)     else Draw z_{d,i}' ~ Multinomial(θ_d')
(15)       Draw w_{d,i} ~ Multinomial(φ_{e_{d,i},z_{d,i}'})
(16)     endif
(17)   endfor
(18) endfor
```

Algorithm 1: Record entity topic model (RETM).

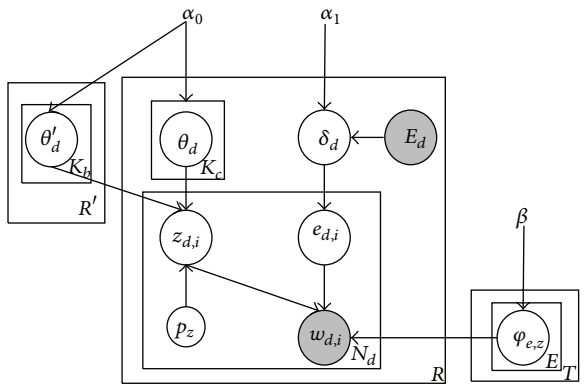


Figure 1: A graphical representation of RETM.

of choosing local topics is $p_z$ and the probability of choosing background topics has to be $1 - p_z$. By observations, we find that once the value of attributes consists of short tokens, record $d$'s topic mostly depends on the distribution of local topics. On the contrary, if the record contains long texts such as abstract information of a paper or the reviews of a product, $d$ largely relies on the background topics. For convenience, we define a flag variable $t \in \{0, 1\}$, where $t = 1$ means the multinomial distribution $\theta_d$ of a word $z_i$ is determined by local topics whereas $t = 0$ means $\theta'_d$ is selected. At the same time, multinomial distribution $\delta_d$ over entity set $E_d$ can be drawn from a Dirichlet prior with $\alpha_1$. Unlike other topic models which usually select an entity directly from the entity sets, we construct our model that depends on a record-specific multinomial distribution $\delta_d$. This stems from the fact that each entity tends to have different weights in generating a record. For example, casts and the director dedicate differently to a movie record; if a research paper has multiple authors, the lead author possibly makes more effort than the others. Next, to generate each word in records, a topic $z_{d,i}$ is drawn from the determined $\theta_d$ or $\theta'_d$, and an entity $e_{d,i}$ is drawn from $\delta_d$. After that, a word $w_{d,i}$ can be generated through Gibbs sampling based on an entity and topic specific multinomial distribution $\varphi_{e_{d,i},z_{d,i}}$ that has a Dirichlet prior $\beta$. In a word, each term is related to a pair of entity and topic and the generative process is the result of their joint actions. Here, the values of $w_{d,i}$ and $E_d$ can be derived from observations and parameters like $\alpha_0$, $\alpha_1$, and $\beta$ can also be obtained from empirical statistics. Others parameters like $\theta_d$, $\delta_d$, $z_{d,i}$, $e_{d,i}$ and $\varphi_{e_{d,i},z_{d,i}}$ are all unknown latent variables that needs to infer.

*3.3. Model Learning and Inference.* Now we can infer the topic of querying records through probability statistics based on previously known data and the generative process. Researchers usually employ approximate techniques to estimate the LDA parameters such as Variation Expectation Maximization [20], Expectation Propagation [21], or Gibbs sampling [22]. In this work, we intend to use a Gibbs-sampling-based method for the posterior distribution of the latent variables.

Gibbs sampling is a Markov chain Monte Carlo (MCMC) algorithm for obtaining a sequence of observations which are approximated from a specified multivariate probability

distribution. It generates a Markov chain of samples, each of which is correlated with adjacent samples. The output parameters can be estimated iteratively until convergence. Figure 2 gives a brief illustration of this entire process.

Depending upon this process, we could repeatedly sample the entity-topic pair for each word, in case of the assignment of all the rest words $(Z, \varepsilon)$ along with the priors parameters. At this point, the conditional posterior of assignment $(e_{d,i}, z_{d,i})$ to the $i$th word $w_{d,i}$ in record $d$ is

$$\begin{aligned} P\left(z_{d,i}, e_{d,i}, t \mid w_{d,i}, Z_{\setminus d,i}, \varepsilon, \Phi, p_z\right) &\propto P\left(w_{d,i} \mid z_{d,i}, e_{d,i}, Z_{\setminus d,i}, \varepsilon, \Phi\right) \\ &P\left(z_{d,i} \mid Z_{\setminus d,i}, \Phi\right) P\left(t \mid p_z\right) \\ &P\left(e_{d,i} \mid \varepsilon_{\setminus d,i}, \Phi\right), \end{aligned} \tag{1}$$

where subscript "$\setminus d$" denotes a quantity excluding data from position $i$ in record $d$ in order to eliminate the influence of current position word tokens.

Formula (1) is comprised of three terms that we can calculate separately. Firstly, term 3 denotes the probabilistic distribution of one entity in record $d$ given all its associated entity sets except $i$th token in record $d$. For records set with $R$ records, the prior over $\delta = (\delta_1, \ldots, \delta_R)$ is also assumed to be a symmetric Dirichlet with concentration parameter $\alpha_1$. Thus, given corresponding entity assignments $\varepsilon = \{e_d\}_{d=1}^{R}$ in $|E_d|$ entities vocabulary set, the conditional posterior of one main topic $k$ of certain record $d$ can therefore be written in the form

$$\begin{aligned} P\left(e_{d,N_d+1} = k \mid \varepsilon, \alpha_1\right) &= \int \mathrm{d}\delta_d P\left(k \mid \delta_d\right) P\left(\delta_d \mid \varepsilon, \alpha_1\right) \\ &= \frac{N_{k|d} + \alpha_1 / \left|E_d\right|}{N_d + \alpha_1}, \end{aligned} \tag{2}$$

where $N_{k|d} = n_d$ denotes frequency of occurrence of topic $k$ to all topics associated with record $d$ and $N_d = \sum_k N_{k|d}$. Note that each record can contain a few duplicated topics, but the main topics cannot be repeated.

It is clear that the probability of excluding entity $e_{d,i}$ at current position is equal to total probability divided by the probability of excluding the word at $i$th position and $e_{d,i} = k$, which we represent as $p_{-di}$, where

$$p_{-di} = \prod_{d=1, d \neq di}^{R} \frac{\Delta\left(n_{d,-i} + \alpha_1\right)}{\Delta\left(\alpha_1\right)}, \tag{3}$$

and the distribution probability $p_i$ of $e_{d,i}$ is

$$p_i = \frac{\Delta\left(n_{di} + \alpha_1\right)}{\Delta\left(\alpha_1\right)}, \tag{4}$$

so that

$$\begin{aligned} P\left(e_{d,i} \mid \varepsilon_{\setminus d,i}, \Phi\right) &= \frac{P\left(e_{d,i} \mid \varepsilon_{d,i}, \Phi\right)}{\Delta\left(n_{di,-i} + \alpha_1\right) / \Delta\left(\alpha_1\right)} \\ &\propto \frac{N_{e_{d,i}|d}^{\setminus d,i} + \left(\alpha_1 / \left|E_d\right|\right)}{N_d - 1 + \alpha_1}, \end{aligned} \tag{5}$$

where $N_{e_{d,i}|d}^{\setminus d,i}$ is the number of word tokens assigned with entity $e_{d,i}$ except $i$th token in record $d$. In addition, as defined in Dirichlet distribution, there is

$$\frac{1}{\Delta\left(\alpha\right)} = \frac{\Gamma\left(\sum_{v=1}^{V} \alpha_v\right)}{\prod_{v=1}^{V} \Gamma\left(\alpha_v\right)}, \tag{6}$$

where $V$ stands for the size of vocabulary set and meanwhile, we have $n_{di} = n_{di,-i} + 1$ according to our definition. At this point, we can infer (5).

For the second term in (1), as mentioned previously, the main topic may come from local topic or overall background topics. In the case of the Bernoulli distribution, the probability distribution over $t$ can be written in the form

$$P\left(t \mid p_z\right) = \left(p_z\right)^{n_t=1} \left(1 - p_z\right)^{n_t=0}, \tag{7}$$

where $t$ is generated from Bernoulli $(p_z)$. Since word $w_i$ is generated from local topic $z_{d,i}$ if $t = 1$, according to Figure 1, there are $K_c$ topics $Z = \{z_1, z_2, \ldots, z_{k_c}\}$, we have

$$P\left(z_{d,i} \mid Z_{\setminus d,i}, \Phi\right) P\left(t \mid p_z\right) \propto \frac{N_{z_{d,i}|d}^{\setminus d,i} + \left(\alpha_0 / K_c\right)}{N_d - 1 + \alpha_0} p_z. \tag{8}$$

And if $t = 0$, word $w_i$ is generated from background topic $z'_{d,i}$. Since there are $K_b$ topics $Z' = \{z'_1, z'_2, \ldots, z'_{k_b}\}$, we obtain

$$P\left(z_{d,i} \mid Z_{\setminus d,i}, \Phi\right) P\left(t \mid p_z\right) \propto \frac{N_{z'_{d,i}|d}^{\setminus d,i} + \left(\alpha_0 / K_b\right)}{N'_d - 1 + \alpha_0} \left(1 - p_z\right), \tag{9}$$

where $N_{z_{d,i}|d}^{\setminus d,i}$ or $N_{z'_{d,i}|d}^{\setminus d,i}$ is the number of word tokens assigned with topic $z_{d,i}$ or $z'_{d,i}$ except $i$th token in record $d$. Meanwhile, we take the same methods for determining $P(z_{d,i} \mid Z_{\setminus d,i}, \Phi)$ as term 3.

We now determine the first term in the objective formula. Note that the distribution of words in every record is regarded as a mixture of topic-entity pair distributions. Given $K$ topic entity pairs whose distribution $\varphi_{e,z}$ is a multinomial distribution with a Dirichlet prior $\beta$, the word distribution probability that not include the word at position $i$ and associated pair $< z_i, e_i >= k$ is in the form of

$$P\left(w_i \mid Z_{\setminus i}, \varepsilon_{\setminus i}, \Phi\right) = \prod_{k=1,\ k \neq \langle z_i, e_i \rangle}^{K} \frac{\Delta\left(n_{k,-i} + \beta\right)}{\Delta\left(\beta\right)}, \tag{10}$$

where $n_{k,-i}$ stands for the number of tokens assigned with the pair $\langle z_i, e_i \rangle$ other than the word $w_i$. Next, the probability of $\langle z_i, e_i \rangle = k$ can be written in the form of

$$P\left(w_i \mid z_i, e_i, \Phi\right) = \frac{\Delta\left(n_k + \beta\right)}{\Delta\left(\beta\right)}, \tag{11}$$

where $n_k$ is the number of word tokens that are assigned to the $k$th entity topic pairs.

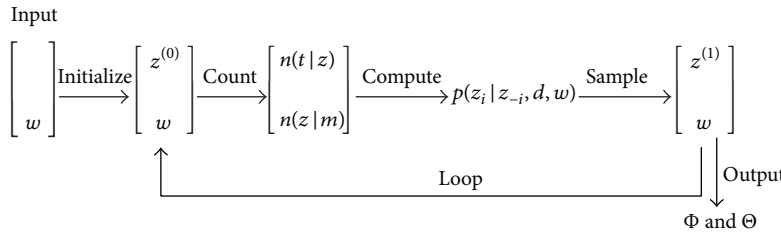


Figure 2: A brief illustration of Gibbs sampling process.

Finally, the expression of first term in (1) takes the form

$$\begin{aligned} & P\left(w_i \mid z_i, e_i, Z_{\setminus i}, \varepsilon_{\setminus i}, \Phi\right) \\ &= \prod_{k=1, k \neq \langle z_i, e_i \rangle}^{K} \frac{\Delta\left(n_{k,-i} + \beta\right)}{\Delta\left(\beta\right)} \cdot \frac{\Delta\left(n_k + \beta\right)}{\Delta\left(\beta\right)} \\ &= \prod_{k=1}^{K} \frac{\Delta\left(n_{k,-i} + \beta\right)}{\Delta\left(\beta\right)} \cdot \frac{\Delta\left(\beta\right)}{\Delta\left(n_{\langle z,e\rangle=k,-i} + \beta\right)} \cdot \frac{\Delta\left(n_k + \beta\right)}{\Delta\left(\beta\right)} \\ &\propto \frac{\Delta\left(n_k + \beta\right)}{\Delta\left(n_{\langle z,e\rangle=k,-i} + \beta\right)} \\ &= \frac{\Gamma\left(\sum_{v=1}^{V}\left(n_{k,-i}^{v} + \beta_v\right)\right)}{\prod_{v=1}^{V}\Gamma\left(n_{k,-i}^{v} + \beta_v\right)} \cdot \frac{\Gamma\left(\sum_{v=1}^{V}\left(n_{k}^{v} + \beta_v\right)\right)}{\prod_{v=1}^{V}\Gamma\left(n_{k}^{v} + \beta_v\right)} \\ &= \frac{n_{k,-i}^{v} + \beta_v}{\sum_{v=1}^{V}\left(n_{k,-i}^{v} + \beta_v\right)}, \end{aligned} \tag{12}$$

given that

$$\begin{aligned} \Gamma\left(a+1\right) &= a\Gamma\left(a\right), \\ n_k^v &= n_{k,-i}^v + 1. \end{aligned} \tag{13}$$

So far, we have obtained three terms in (1), which are the objective of this paper. With the known observed variables, once we obtain the entity topic pair assignments of each word, we can derive the distribution of word tokens over this pair, namely, $\varphi_{e,z}$. Then we can estimate the parameters like $\theta_d$, $\theta'_d$, and $\delta_d$ accordingly to get the distribution of each record over entity topic pairs to ultimately find the most possible topic for each record through analysis.

## 4. Experiments

*4.1. Datasets.* We make use of two large-scale datasets for the evaluation of our model, both of which are extracted from real-world data sources. One dataset is named *BOOK*, and all the records are crawled from Soochow University library bibliographic query system (http://lib.suda.edu.cn) through LAN. We pick up 2000 nonrepeating records of computer science books manually and each record has attributes that include *book name*, *author name*, *ISBN*, *press*, *price,* and *overview*. Owing to the demand of constructing background topics, we keep the URI information about records in the same query lists in the form of RDF. Another dataset is from ACM research paper data source, namely, *PAPER*. It contains 38469 published papers about information system retrieved in 2002–2011 years in 2012 September according to the classic category (download from: http://www.datatang.com/datares/go.aspx?dataid=619787). Each metadata consists of the corresponding *citation network*, *keywords*, *title*, *date of publication*, *journals, and abstract*. In order to facilitate the evaluation, we pick up the top 19235 records in our experiment, which is perfect to test our model, since it contains the associated keywords data. However, it does not contain the information about querying so it is not suitable for testing the background topic part.

Table 2: Summary statistics of the corpus.

| Dataset | $R$ | $E$ | $W$ | avg ($\lvert E_d \rvert$) | avg ($\lvert N_d \rvert$) |
|---|---|---|---|---|---|
| Book | 2000 | 622 | 11342 | 10.82 | 178.6 |
| Paper | 19235 | 3175 | 10889 | 1.76 | 98.61 |

To testify the impact of entity distribution on the main model, we use DBpedia Soptligh [23] to preprocess the data from two datasets. DBpedia Soptligh is a high-performance online entity extraction and disambiguation service that links all the extracted entity to Wikipedia, and in addition, with the help of DBpedia, we could get the corresponding concepts of each entity. After this procedure, we delete some infrequent words manually, which means words that appear less than 5 records. The statistics of the corpus are summarized in Table 2.

*4.2. Baseline and Evaluate Criterion.* The objective of this paper is to verify two factor distributions impacts on overall topic distribution: the entity distribution and the background topic distribution. Since the datasets are composed of discrete records, we need to choose the baseline methods in the experiments for different purposes.

Firstly, we choose LDA, Link-LDA, AM, and ATM as the baselines in our experiment of testing entity distribution impact. We claim that word distributions should depend on associated entities besides the topics. Figure 3 gives the graphical representations of these four models based on the notations of this paper.

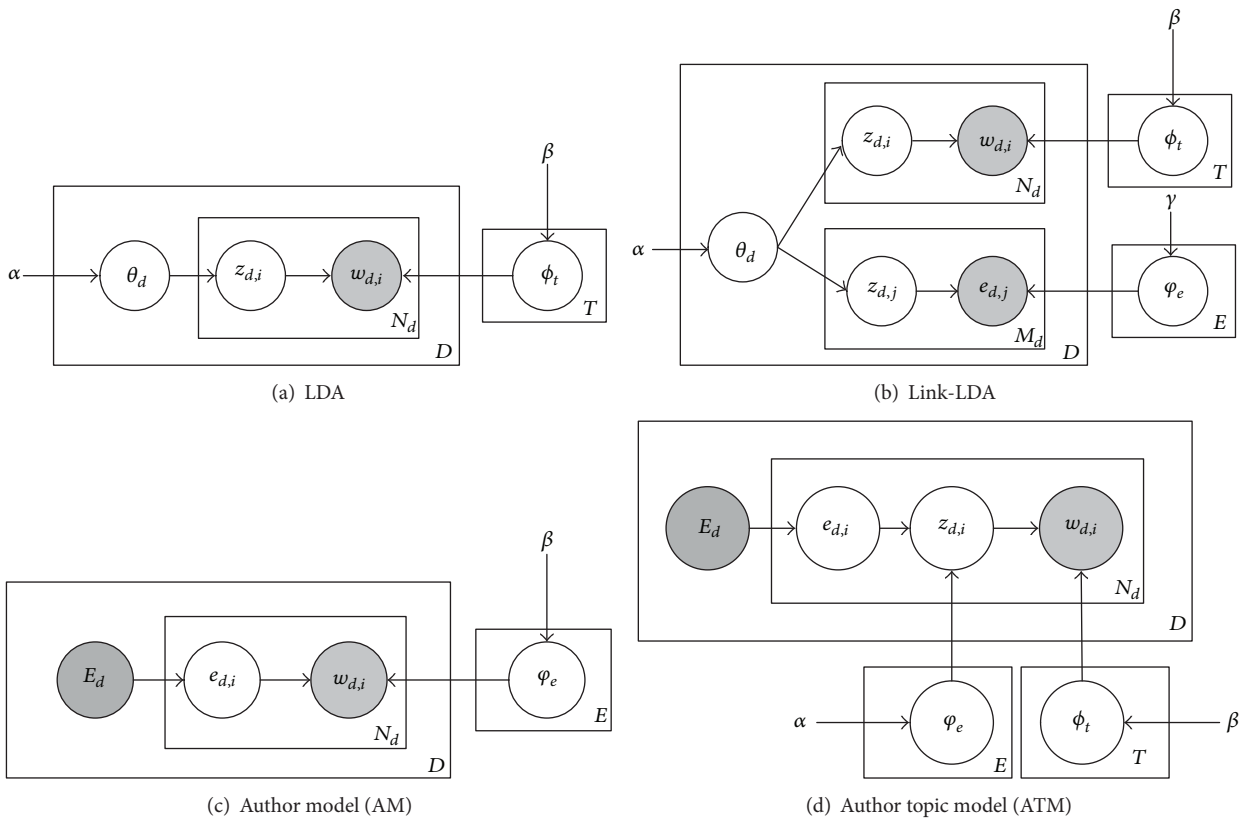


Figure 3: Four topic models associated with entities, topics, and words.

From Figure 3, we note that besides LDA, the other three all contain entities, so they are suitable for the first test. Secondly, to verify the background topics distribution's impact, we can divide RETM into several models due to different test objectives. In other words, RETM can be simplified as LDA if we do not take entity distributions and background topic distributions into consideration. If we measure the impact caused by entities other than background topic distributions, meaning all the word distribution is decided by the local topic distributions, the model is defined as RETM-self. Because RETM is the mixture of entity distribution and topic distributions, we can assess the impacts caused by each part distinctly through all kinds of comparisons.

For *BOOK*, we set $K = 20, \alpha_0 = \alpha_1 = 0.1$, and $\beta = 0.1$. The fact is that one record tends to choose the local topics once this record contains attributes like *abstract, overview, review*, and so forth, which usually involves long text. Therefore, we say that one record has long text if $|N_d| > 120$ and $p_z = 0.48$ through statistical calculations. For *PAPER*, we set $K = 50$, $\alpha_0 = \alpha_1 = 0.1$, and $\beta = 0.1$ and when $|N_d| > 60$, we define that this record has long text; in this case, $p_z = 0.65$. Above hypermeters are applicable for LDA, Link-LDA, and AM. ATM's hypermeters are set as $\alpha_0 = \alpha_1 = 50/T$ and $\beta = 0.01$ according to [10]'s suggestion.

In addition, we perform perplexity analysis to examine the performance of each model. Perplexity evaluates how well a probability distribution or probability model predicts a sample. The lower a trained model's perplexity is, the better the topic model is, the more flexibility one model will get. The definition of perplexity is hereby given as follows:

$$\text{Perplexity}(W) = \exp\left\{-\frac{\log P\left(D^{\text{test}} \mid D^{\text{train}}\right)}{\sum_{d \in D^{\text{test}}} N_d}\right\}, \quad (14)$$

where if $\Phi$ is the hypermeters set of current model, then

$$P\left(D^{\text{test}} \mid D^{\text{train}}\right) = \int P\left(D^{\text{test}} \mid \Phi\right) P\left(\Phi \mid D^{\text{train}}\right) d\Phi. \quad (15)$$

Formula (15) can be approximately calculated by averaging $P(\Phi|D^{\text{train}})$ under samples from $P(D^{\text{test}} \mid \Phi)$. Since we already known a set of entities, we can get

$$P\left(D^{\text{test}} \mid \Phi\right) = \prod_{d \in D^{\text{test}}} P\left(w_d \mid E_d, \Phi\right). \quad (16)$$

### 4.3. Results and Analysis

*4.3.1. The Impact of Entity Distribution on Topic Models.* This paper claims that word distributions should depend on associated entities as well as the topics. To prove this argument, we could examine the change of word distributions over topics with a fixed entity and then over entities with a fixed topic. We choose to conduct this experiment on *PAPER,* since it does not require any information about the background topics.

Take Professor Michael I. Jordan as an example for the first case study, who is a leading researcher in machine learning and artificial intelligence. He applies Bayesian networks into machine learning and makes outstanding contributions on optimization problems in likelihood estimation. Table 3 shows the word distributions of certain topics when picking Michael I. Jordan as the fixed entity. The first column demonstrates the top 15 words in the entity prior $\varphi_e$ of Michael. Usually the prior distribution of entities presents one person's general methodologies or research interests. Just like the word "neural learning," "intelligence" in the first column is conforming to Michael's research interest. In this way, the model can better describe the word distribution $\varphi_{e,z}$ under different research topics after combining the entity prior $\varphi_e$ and the topic prior $\varphi_z$. In the table, the remainig three columns represent top 15 words of Michael I. Jordan's three main research topics: machine learning, neural networks, and Bayesian network. From these words, we find that the top words have varied a lot with the changes of topics even concerned about the same person. Not many words are repeated or kept in the same rank. Furthermore, Table 4 displays the word distributions with one fixed research topic (machine learning) over three related authors based on $P(e \mid \text{machine learning}, \varepsilon, Z)$. It is obvious that all three authors have published many papers on machine learning area but in different research approaches. For example, Pedro Domingo is specialized in the research of Markov logic networks whereas Judea Pearl is credited for inventing Bayesian networks and several inference methods in the models; Andrew Ng is Michael I. Jordan's student, who mainly focuses on researches about deep learning recently. Although three authors study very distinct approaches under the same research topic, which causes totally different word distributions, this difference could be measured through analysis of $P(w \mid e, z)$ based on RETM. Now we can get the conclusion that topic model that is associated with entity distribution could be more detailed and distinguishable in topic category subdivision process. In addition, it provides intuitive and understandable comparisons through $P(w \mid e, z)$, which is impossible if we are just modeling $P(w \mid z)$ or $P(w \mid e)$.

We also need to analyze the perplexity value of RETM compared to the baseline models: LDA, Link-LDA, AM, and ATM. Recall that there is no background topics information in *PAPER*, so RETM and RETM-self are considered to be the same. According to (16), we randomly choose the 70% of the records as the training set $D^{\text{train}}$ and the remaining 30% records as the test set $D^{\text{rest}}$. Figure 4 shows the changes of perplexity of the tested models under diverse number of topics in *BOOK*. It is clear that RETM-self, LDA, and Link-LDA have very similar perplexity value, that is, mainly because most words used in academic papers are topic related, but in original corpus, not many words are associated with entity features. For instance, one author may use an inventive term that particular points to one object in his article (IoT, Cloud, etc.), this term will not be generally accepted unless such kind of research topics become popular or famous. From Figure 3(c) we can tell that AM does not have topics in its model, so it cannot gain any advantages from other priors. As a result, it gets the highest perplexity value and does not change with the increment of number of topics. Compared with other models, RETM-self has the lowest perplexity value when the number of topics is small; furthermore, the values decrease steadily as the topic numbers increase. However, it begins to rise when the topic numbers exceed 100. That is because the model has to estimate a large number of parameters, which causes overfitting problems. Same problem happens to ATM as well. Judged from the overall performances of the five models, we can get the conclusion that models that adopt the given associated entity sets as extra information to learn the topic distributions for records perform better than other models.

Table 3: Michael I. Jordan's entity prior ($\varphi_e$) and word distributions ($\varphi_{e,z}$) of his research topics.

| Michael I. Jordan | Machine learning | Neural networks | Bayesian network |
|---|---|---|---|
| Learning | Probabilistic | Cognitive | Causal |
| Neural | Causal | Learn | Distributions |
| Processing | Systems | Neural | Models |
| Awards | Methods | Networks | Markovian |
| Advances | Reasoning | Proceedings | dynamic |
| Machine | Algorithm | Artificial | Identification |
| AAAI | Graphs | Conference | Characterization |
| Bayesian | Model | International | Recursive |
| ACM | Kernel | System | Joint |
| Networks | Computational | NIPS | Variables |
| Statistical | Award | Graph | Data |
| Computational | Uncertainty | Control | Effects |
| Associates | Representation | Kernel | Algorithm |
| Conference | Processing | Science | Based |
| Intelligence | Networks | Probability | Clustering |

Table 4: Machine learning's topic prior ($\varphi_z$) and word distributions ($\varphi_{e,z}$) of its related entities.

| Machine learning | Judea Pearl | Pedro Domingos | Andrew Ng |
|---|---|---|---|
| Learning | Causal | Logic | Learning |
| Machine | Revisited | Markov | Stanford |
| Data | Markovian | Networks | Neural |
| Algorithm | Data | Learning | Deep |
| Representation | Counterfactual | MLNs | Networks |
| Theory | Artificial | Algorithm | Word |
| Examples | Explanations | Theory | Technology |
| Sparse | Independence | Knowledge | Model |
| Analysis | Path | Order | Google |
| Computational | Specificity | Models | Class |
| Artificial | Representation | Systems | Machine |
| Recognition | Proven | Representation | Artificial |
| Programming | Tags | Reasoning | Intelligence |
| Model | Embracing | World | System |
| Supervised | Tolerating | Structure | Data |

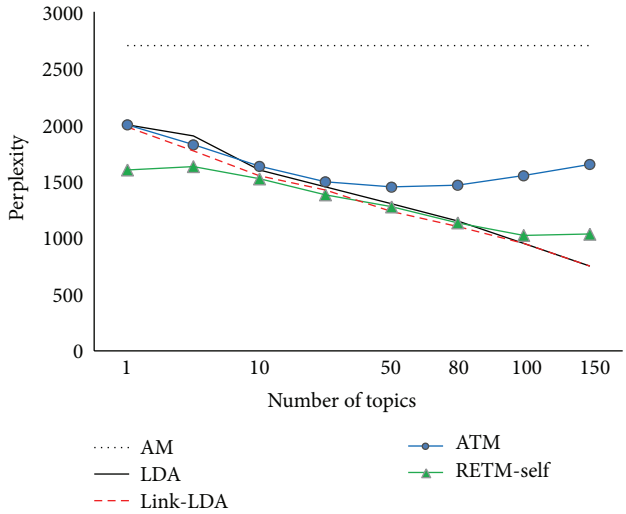


Figure 4: Perplexity values for different numbers of topics (*BOOK*).

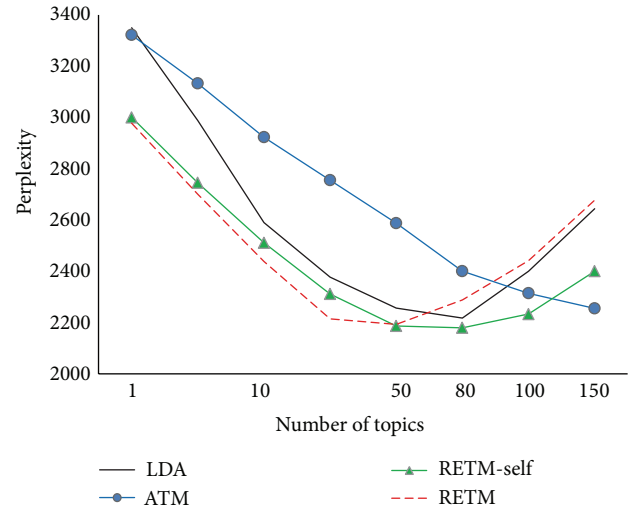


Figure 5: Perplexity values for different numbers of topics (*PAPER*).

*4.3.2. The Impact of Background Topics Distribution on Topic Models.* In this task, we prefer to use *BOOK* in that this dataset contains detailed linking information about records crawled from the data sources. In this case, we could easily summarize the background topic from one record as well as its neighbor records. This time, we compare RETM with LDA, ATM, and RETM-self. Since AM does not involve entity or background topics in its model, there is certainly no change in its perplexity values as the topic number grows; thus, it has no comparison value. Furthermore, LDA and Link-LDA perform very alike under these situations so we elect to select LDA as the representative model for simplicity. Other hyperparameters or experimental settings are kept the same as the previous one. Unlike *PAPER*, we need to preprocess the data in *BOOK* with DBpedia knowledge base. As a result, relatively more words that associated with entities are chosen for the corpus.

Figure 5 shows the changes of perplexity of the tested models under different numbers of topics for *PAPER* dataset. First of all, ATM gives almost the same performance as in *BOOK*. It only needs to estimate a few hyperparameters and can make use of the extra entity distributions prior informatio;, its perplexity values have been decreasing as more topics added. Secondly, both RETM-self and RETM's performances are better than LDA, which further indicates that with the help of entity distributions, our model can be more accurate and convenient to categorize the given topics. This effect is rather obvious when the number of topics is no less than 20. Finally, we can find that REMT gets the lowest perplexity values, comparing RETM with RETM-self, especially when the number of topics is less than 10. No doubt RETM can take advantage of the related background topics information, which is the only difference between the two models. Unfortunately, RETM has to estimate more parameters when picking the background topics distributions, which bring a large amount of calculation and lead to enormous complexity in each sampling iteration. Consequently, its perplexity values grow very fast, limiting its generalization performance, and get worse when the number of topics is larger than 100.

Table 5: Accuracy rate of topics models for records with different lengths.

| Name | Manual | Random |
|---|---|---|
| ATM | 0.341 | 0.72 |
| LDA | 0.311 | 0.695 |
| RETM | **0.822** | 0.738 |

Next, we will discuss to what extent the background topic distributions can affect the general topic. Recall that all the above experiments are conducted under the assumption that the record chooses to use the local topics when this record has long described text in its attributes, meanwhile, it decide to choose the background topics which generated from current record and neighbor records when the text is consist of short tokens. Now to prove the accuracy of hypothesis, we randomly pick up 2000 records that contain different lengths text from the *BOOK* as set 1,and then pick up 2000 records that do not have any *overview* information about this book manually as set 2. In this case, set 2 is considered to contain short text only. We now extract the topics of each record from two sets, respectively, and compare the accuracy with ATM and LDA. The results can be found in Table 5. We can see that for the random dataset 1, all three models perform fairly well except that LDA and ATM fail to get high points when dealing with set 2. ATM performs slightly better than LDA due to the linking information gathered from entities but still cannot compare with RETM, who gains almost twice accuracy rate than others. This further provides evidence that making use of the distributions of background topics can greatly improve the topic models, locate the correct position of each word, and finally extract the record concept that better expresses the content itself.

## 5. Conclusions

This paper studies a particular topic model construction method for the infrastructure of the information network for IoT. Based on the fact that almost every record extracted from the query result lists of IoT involves several entities, it proves that the entity distributions have some substantial impact on the word distributions within records. That is, once we add the entity distributions as priors into the topic model, the model can produce meticulous category as well as increase its distinguishability, which helps the users to understand the record content easily. In addition, our model particularly pays attention to the way each record is extracted. We classify the records gathered after querying the same keywords and then claim that the word distribution in every record should associate with topics that is generated from either its own topic distributions or the background topic distributions. These two arguments are accepted through the experiments on real-world datasets. From the analysis of experimental results, we find that our model is more suitable for extract topics from records compared to other traditional models. Furthermore, this approach can extract more accurate topics that can exactly describe the content of the records. So our work has some advanced and realistic significance to the further work of IoT.

## Notations

$R$: Number of query records
$K$: Number of topic-entity pairs
$W$: Number of words
$E$: Number of entities
$N_d$: Number of word tokens in record $d, d \in \{1, 2, \ldots, R\}$
$K_c$: Number of topics produced by current record
$K_b$: Number of topics produced by neighbor records
$\theta_d$: Multinomial distribution of topics specific to record $d$
$\theta'_d$: Multinomial distribution of overall topics specific to record $d$
$\delta_d$: Multinomial distribution of entities specific to record $d$
$E_d$: List of entities associated with record $d$
$e_{d,i}$: Entity associated with the $i$th token in record $d$
$z_{d,i}$: Topic associated with the $i$th token in record $d$
$w_{d,i}$: $i$th token in record $d$
$\varphi_{e,z}$: Multinomial distribution of words specific to entity $e$ and topic $t$
$P_z$: Probability of the final topics coming from record $d$ itself
$t$: Flag variables
$Z$: Set of all topic assignments $\{z_{d,i}\}$
$\varepsilon$: Set of all entities assignments $\{e_{d,i}\}$
$\Phi$: Set of all parameters in the model like $\alpha_0, \alpha_1, \beta$.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This research was partially supported by the Natural Science Foundation of China under Grants no. 61003054, no. 61170020, and no. 61070169, Natural Science Foundation of Jiangsu Province under Grant no. BK2011376, the Opening Project of Suzhou High-tech Key Laboratory of Cloud Computing & Intelligent Information Processing no. SXZ201302, Provincial Key Laboratory for Computer Information Processing Technology no. KJS1329, Science and Technology Support Program of Suzhou no. SG201257, Science and Technology Support program of Jiangsu province mo. BE2012075, and Suzhou Key Laboratory of Converged Communication no. SKLCC2013XX.

## References

[1] D. Giusto, A. Iera, G. Morabito, and L. Atzori, Eds., *The Internet of Things*, Springer, New York, NY, USA, 2010.

[2] X. Liu, Y. Song, S. Liu, and H. Wang, "Automatic taxonomy construction from keywords," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*, pp. 1433–1441, 2012.

[3] Y. Zhang, F. Yang, Z. Xiong et al., "Study on context-based domain ontology concept extraction and relation extraction," *Application Research of Computers*, vol. 1, article 023, 2010.

[4] E. Erosheva, S. Fienberg, and J. Lafferty, "Mixed-membership models of scientific publications," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 1, pp. 5220–5227, 2004.

[5] Q. Mei, D. Cai, D. Zhang et al., "Topic modeling with network regularization," in *Proceedings of the 17th International Conference on World Wide Web (WWW '08)*, pp. 101–110, 2008.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. 4-5, pp. 993–1022, 2003.

[7] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99)*, pp. 50–57, 1999.

[8] C. Sun, B. Gao, Z. Cao, and H. Li, "HTM: a topic model for hypertexts," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 514–522, October 2008.

[9] A. Mccallum, "Multi-label text classification with a mixture model trained by EM," in *Proceedings of the AAAI Workshop on Text Learning*, 1999.

[10] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents," in *Proceedings of the 20th Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI '04)*, pp. 487–494, AUAI Press, Arlington, Va, USA, 2004.

[11] R. Nallapati and W. W. Cohen, "Link-PLSA-LDA: a new unsupervised model for topics and influence of blogs," in *Proceedings of the International Conference on Weblogs and Social Media (ICWSM '08)*, 2008.

[12] C. Zhang, J. Sun, and Y. Ding, "Topic mining for microblog based on MB-LDA model," *Computer Research and Development*, vol. 48, no. 10, pp. 1795–1802, 2011.

[13] L. Dietz, S. Bickel, and T. Scheffer, "Unsupervised prediction of citation influences," in *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*, pp. 233–240, June 2007.

[14] J. Tang, S. Wu, B. Gao, and Y. Wan, "Topic-level social network search," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*, pp. 769–772, August 2011.

[15] J. Tang, B. Wang, Y. Yang et al., "PatentMiner: topic-driven patent analysis and mining," in *Proceedings of the 18th International Conference on Knowledge Discovery and Data Mining (KDD '12)*, pp. 1366–1374, August 2012.

[16] J. Wang, H. Wang, Z. Wang, and K. Q. Zhu, "Understanding tables on the web," in *Conceptual Modeling*, pp. 141–155, Springer, Berlin, Germany, 2012.

[17] H. Kim, Y. Sun, J. Hockenmaier, and J. Han, "ETM: entity topic models for mining documents associated with entities," in *Proceedings of the 12th IEEE International Conference on Data Mining (ICDM '12)*, pp. 349–358, December 2012.

[18] S. Auer, C. Bizer, G. Kobilarov et al., "Dbpedia : a nucleus for a web of open data," in *The Semantic Web*, pp. 722–735, Springer, Berlin, Germany, 2007.

[19] W. Wu, H. Li, H. Wang, and K. Q. Zhu, "Probase: a probabilistic taxonomy for text understanding," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '12)*, pp. 481–492, Scottsdale, Ariz, USA, May 2012.

[20] J. Graça, K. Ganchev, and B. Taskar, "Expectation maximization and posterior constraints," in *Advances in Neural Information Processing Systems*, vol. 20, pp. 569–576, 2007.

[21] T. Minka and J. Lafferty, "Expectation-propagation for the generative aspect model," in *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pp. 352–359, 2002.

[22] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 1, pp. 5228–5235, 2004.

[23] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer, "DBpedia spotlight: shedding light on the web of documents," in *Proceedings of the 7th International Conference on Semantic Systems (I-SEMANTICS '11)*, pp. 1–8, September 2011.