Incorporating Word Correlation into Tag-Topic Model for Semantic Knowledge Acquisition

Fang Li[2], Tingting He[1, 2], Xinhui Tu[1], Xiaohua Hu[1, 3]

[1]Department of Computer Science, Central China Normal University, Wuhan, China
[2]National Engineering Research Center for E-Learning, Central China Normal University, Wuhan, China
[3]College of Information Science and Technology, Drexel University, Philadelphia, PA, USA
fang__lf@163.com tthe@mail.ccnu.edu.cn tuxinhui@gmail.com xh29@drexel.edu

ABSTRACT

This paper presents a tag-topic model with Dirichlet Forest prior (TTM-DF) for semantic knowledge acquisition from blog. The TTM-DF model extends the tag-topic model (TTM) by replacing the Dirichlet prior with the Dirichlet Forest prior over the topic-word multinomial. The correlation between words are calculated to generate a set of Must-Links and Cannot-Links, then the structures of Dirichlet trees are obtained though encoding the constraints of Must-Links and Cannot-Links. Words under the same subtrees are expected to be more correlated than words under different subtrees. We conduct experiments on a synthetic and a blog dataset. Both of the experimental results show that the TTM-DF model performs much better than the TTM model. It can improve the coherence of the underlying topics and the tag-topic distributions, and capture semantic knowledge effectively.

Categories and Subject Descriptors

I.2.7 [Artificial Intelligence]: Natural Language Processing –*Text analysis.*

General Terms

Algorithms, Experimentation, Design.

Keywords

Topic model, Dirichlet Forest prior, Tag, Blog

1. INTRODUCTION

The prevalence of Web 2.0 services and applications has brought an large amount of resources, especially text resources, such as encyclopedia, blogs, and social networks. These resources contain a wealth of semantic information, which can be applied to a variety of fields of information processing to improve the service quality. How to automatically obtain semantic knowledge from online text resources and represent them in a way that can be calculated by computer programs has become a hot research area in natural language processing(NLP).

Blog is the fourth communication means in web following E-mail, BBS and instant communication, and it is an important platform for self-expression, information release and social networking. Blog can be seen as a knowledge base with wide coverage and good scalability. Tagging has recently emerged as a popular way to organize user generated content for Web 2.0 applications, such as blogs and bookmarks. In blogs, users can assign one or more tags for each log. Usually, these tags can reflect the concerned subjects of the contents. Tags can be seen as labeled meta-information about the content, and they are beneficial for knowledge acquisition from blogs.

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

This paper focuses on semantic knowledge acquisition from blogs with topic model. In order to use the tags, we use the framework of the TTM model [1], which extends the Latent Dirichlet Allocation (LDA) [2] by adding a tag layer between the document and topic layer. The model represents each document with a mixture of tags, each tag is associated with a multinomial distribution over topics and each topic is associated with a multinomial distribution over words. However, the TTM model is an unsupervised graphical model. The topics don't always make sense to users because the model only depends on the statistical strength without any additional knowledge. But, users may have additional knowledge about the word distribution. For example, we know that some words are synonyms, and they should have aligned topics. Incorporating the knowledge into topic models would be useful to improve their performance.

In this paper, we extend the tag-topic model by incorporating domain knowledge via Dirichlet Forest prior [3]. We compute the relatedness between words, and use them to generate a set of Must-Links and Cannot-Links [4], then encode them using a Dirichlet Forest prior, to replace the Dirichlet prior over the topic-word multinomial distribution *p(word|topic)*. After estimating the parameters of the model, we get the document-tag, tag-topic and topic-word distributions, and represent the semantic knowledge underlying in the topics as $c:(w_1, r_1), (w_2, r_2), \ldots (w_i, r_i) \ldots (w_n, r_n)$, where c is a tag, $w_1 \ldots w_n$ are the top words arranged to the top topics of it, r_i is the semantic relatedness between w_i and c.

The remainder of this paper is organized as follows: Section 2 reviews the related works . Section 3 presents the TTM-DF model and introduces the parameter estimation process. Section 4 shows the experimental results on synthetic documents and blog corpus. At last, we conclude the paper in Section 5.

2. RELATED WORKS

In this section, we introduce the related works of semantic knowledge acquisition and variations of LDA from topic level.

Recently, the methods of semantic knowledge acquisition based on large-scale text collection are widely used. There are mainly two kinds of text collections. One is the online encyclopedias and the other is the online or off-line unstructured plain texts. Online

encyclopedias are semi-structured texts. Various relations, links, categories between the pages are formatted, and they have been widely used to mine semantic knowledge[5-8]. For unstructured text collections, predefined patterns are applied to web pages to produce peer or sibling relationship words [9, 10], and generative models[3, 11] are used to mine the underlying topics, and they have been used to semantic analysis in several tasks[12, 13].

The status of LDA is a fully generative probabilistic model, which allows principled extensions and variations capable of incorporating rich knowledge. At the topic level, there has been an effort to inject various knowledge into topic models. For example, Word sense was seen as a hidden variable to be integrated into the topic model[14]. Topic-in-Set knowledge was defined to recover topics relevant to user interests[15]. The Dirichlet Forest prior (DF) was proposed to improve the topic-word multinomial [3]. Seed words were used to improve both topic-word and document-topic distributions[16], et al.

3. TTM-DF MODEL

3.1 Dirichlet Forest Prior

In LDA, the Dirichlet distribution has two key limitations: the multinomial topic-word distribution ϕ is drawn from a shared symmetric Dirichlet prior, and the topic assignments of words are independent except the normalization constraint. The Dirichlet-Tree distribution [17] overcomes the limitations while preserving the conjugation with multinomial distribution. It is a tree with the words as leaf nodes. Each node in the tree is represented by a Dirichlet distribution over the branches to its child nodes. The probability of a leaf is the product of branch probabilities leading to that leaf. An independence relationship is given for words in the tree. The Dirichlet Forest prior is a mixture of Dirichlet Tree with different structures, and it assigns a tree for each topic. The trees are constructed by encoding the set of Must-Links and Cannot-Links associated with the domain knowledge. Must-Link(A, B) means the two words A and B tend to be generated by the same topic. Cannot-Link(A, B) means A and B tend to be generated by different topics. The details can be seen in [3, 17].

3.2 Model Description

The TTM model extends LDA by adding an additional tag layer between the document and topic layer. Its basic idea is that before writing an article, a blogger is clear that the content will contains which main aspects, and for each aspect he will choose a tag to describe it. To generate a word, it first chooses a tag, describing the aspect that the word would convey, and then chooses a topic conditioned on that tag, and samples a word from that topic.

However, the TTM model is an unsupervised graphical model. It discovers the latent topics only depending on the statistical strength without any additional knowledge. Unrelated or loosely related words may be mixed into a topic if they co-occur frequently, or related words may be assigned to different topics if they co-occur rarely. The discovered topics don't always make sense as expected. In many applications, users may have additional knowledge about the word distribution. For example, we know that some words are synonyms, which should be assigned to a same topic no matter whether they co-occur frequently or not, and some words are not related, which should appear in different topics. We would like these preferences to guide the recovery of latent topics. But the TTM model lacks a mechanism for incorporating such domain knowledge.

We extend the TTM model by replacing the Dirichlet prior with the Dirichlet Forest prior, into which we incorporate the domain knowledge to improve the quality of topics. We express the knowledge as a set of Must-Links and Cannot-Links, and encode these constraints through the structures of Dirichlet trees. The Dirichlet Forest distribution assigns a tree for each topic. Because the structures of the trees are different, the Dirichlet Tree priors of each topic are also different. To generate ϕ, a tree is sampled from the Dirichlet Forest prior for each topic, and then the multinomial distributions ϕ are sampled conditioned on these trees. For each tree, a multinomial of branching probabilities is drawn from a Dirichlet distribution at each internal node independently, and then the probability ϕ_w of word w is computed as the product of multinomial parameters on the edges from the root node to the leaf node w. The other procedures of generating a document set are same as before. So, the generative process in the TTM-DF model can be described as follows, and the notations are explained in Table 1.

- For each of the D documents d, sample $\psi_d \sim Dirichlet(\mu)$
- For each of the L tags l, sample $\theta_l \sim Dirichlet(\alpha)$
- For each of the K topics k,
  - sample a tree $q_k \sim DirichletForest(\eta,\beta)$
  - sample $\phi_k \sim DirichletTree(q_k)$
- For each of the N_d words w_i in document d
  - sample a tag $t_i \sim Multinomial(\psi_d)$
  - sample a topic $z_i \sim Multinomial(\theta_{t_i})$
  - sample a word $w_i \sim Multinomial(\phi_{z_i})$

The joint distribution of the model is:

$$p(w,z,t,q_{1:K} \mid \alpha,\beta,\mu,\eta) = p(w \mid q_{1:K},z,\beta,\eta)p(z \mid t,\alpha)p(t/\mu)\prod_{k=1}^{K} p(q_k) \cdot$$

Where, $p(w \mid q_{1:K},z,\beta,\eta) = \prod_{k=1}^{K}\prod_{s}^{I_k}\left(\frac{\Gamma(\sum_v^{C_k(s)}\gamma_k^{(v)})}{\Gamma(\sum_v^{C_k(s)}(\gamma_k^{(v)}+n_k^{(v)}))}\prod_v^{C_k(s)}\frac{\Gamma(\gamma_k^{(v)}+n_k^{(v)})}{\Gamma(\gamma_k^{(v)})}\right) \cdot$

3.3 Parameter Estimation

The Dirichlet Forest distribution is a mixture of Dirichlet Tree distribution, which is conjugate to the multinomial distribution, so we can use the Gibbs sampling [18] for parameter inference. In the TTM-DF model, the Markov-chain Monte Carlo (MCMC) state is defined by both the **(t, z)** pair and the tree indices $\mathbf{q_{1:K}}$. The sampling scheme consists of two parts: sampling (t_i, z_i) for word w_i and sampling q_k for topic z_k.

Sampling a (t_i, z_i) pair for word w_i:

$$p(t_i = l, z_i = k \mid \mathrm{t}_{-\mathrm{i}}, \mathrm{z}_{-\mathrm{i}}, \mathrm{w}, q_{1:K}, \alpha, \beta, \mu, \eta) \propto$$
$$\frac{n_{-i,l}^{(d_i)} + \mu}{\sum_{\mathrm{l}'} n_{-i,l'}^{(d_i)} + \mathrm{L}\mu} \cdot \frac{n_{-i,l}^{(k)} + \alpha}{\sum_{\mathrm{k}'} n_{-i,l}^{(k')} + \mathrm{K}\alpha} \cdot \prod_{s}^{I_k(\uparrow i)} \frac{\gamma_k^{(C_k(s\downarrow i))} + n_{-i,k}^{C_k(s\downarrow i)}}{\sum_v^{C_k(s)}(\gamma_k^{(v)} + n_{-i,k}^{(v)})} \cdot$$

Sampling a tree q_k for topic z_k:

Since the connected components are independent, sampling q_k can be decomposed into sampling the cliques for each connected component $q_k^{(r)}$ respectively.

$$p(q_k^{(r)} = q' \mid q_{-k}, q_k^{(-r)}, \mathrm{z}, \mathrm{w},) \propto$$
$$\left(\sum_v^{M_{rq'}} \beta_v\right) \times \prod_s^{I_{k,r=q'}} \left(\frac{\Gamma(\sum_v^{C_k(s)}\gamma_k^{(v)})}{\Gamma(\sum_v^{C_k(s)}(\gamma_k^{(v)}+n_k^{(v)}))}\prod_v^{C_k(s)}\frac{\Gamma(\gamma_k^{(v)}+n_k^{(v)})}{\Gamma(\gamma_k^{(v)})}\right) \cdot$$

Table1. The explanation of notations

α, β, μ	Hyper-parameters of Dirichlet distributions.
η	The strength parameter of the domain knowledge.
Ψ, θ, Φ	Matrixes indicating the document-tag, tag-topic and topic-word distributions.
$n_l^{(d)}$	The times that tag l has been selected by a word in document d
$n_l^{(k)}$	The number of times that topic k is assigned to tag l
$n_k^{(v)}$	The number of words that are under node v in tree q_k
I_k	The set of internal nodes in tree q_k.
$C_k(s)$	The set of immediate children of node s in tree q_k.
$I_k(\uparrow i)$	The set of the ancestors of word w_i in tree q_k
$C_k(s \downarrow i)$	The node that is an immediate child of node s and an ancestor of word w_i in tree q_k
$\gamma_k^{(v)}$	The weight of the edge leading into node v in tree q_k.
$Q(r)$	The number of cliques in connected component r.
$M_{rq'}$	The q'-th clique of the connected component r.
$I_{k,r=q'}$	The set of internal nodes below the r-th branch of tree q_k

After a set of sampling processes, a sample **(t, z, $\mathbf{q_{1:K}}$)** obtained from the Markov chain can be used to estimate the parameters with the following formulas:

$$\psi_{dl} = \frac{n_l^{(d)} + \mu}{\sum_{l'} n_{l'}^{(d)} + \mathrm{L}\mu}, \quad \theta_{lk} = \frac{n_l^{(k)} + \alpha}{\sum_{k'} n_l^{(k')} + \mathrm{K}\alpha},$$

$$\phi_{kv} = \prod_{s}^{I_k(\uparrow v)} \frac{\gamma_k^{(C_k(s \downarrow v))} + n_k^{C_k(s \downarrow v)}}{\sum_{v'}^{C_k(s)} (\gamma_k^{(v')} + n_k^{(v')})}.$$

4. EXPERIMENTAL RESULTS

4.1 Experiments on Synthetic Corpus

We use three words "A, B, C" and two tags "T1, T2" to synthesize a corpus to show the property of the TTM-DF model. The dataset contains three documents: < T1, T2 # A B C C >, < T2# C C C C >, < T1 # A B A B >.

Let K=2, the topics of the TTM model are mainly three kinds: one kind is around [0.5A, 0.5B | C], which is shorthand for ϕ_1 = (0.5, 0.5, 0), ϕ_2 = (0, 0, 1), and it appears 103 times. The other two kinds are around [B | 0.33A, 0.67C] and [A | 0.33B, 0.67C], which appear 33 and 45 times respectively. They correspond to the clusters 1, 2 and 3 in Figure 1(a-1). For every kind of topics, the tag-topic distribution has mainly two kinds: around [0.7, 0.3 | 0.2, 0.8] and [0.8, 0.2 | 0.3, 0.7] on the topics [0.5A, 0.5B | C], they correspond to the clusters 1, 4 in Figure 1(a-2). For topics [B | 0.33A, 0.67C] and [A | 0.33B, 0.67C], the tag-topic distributions are same. They are around [0.5, 0.5 | 0.2, 0.8] and [0.4, 0.6 | 0.2, 0.8], and they correspond to the clusters 2, 3 in Figure 1(a-2).

We can see that A and B are almost interchangeable. We assume that they have high correlation and should be present or absent together in the topics, so we add a Must-Link(A, B). Let η = 1000, the topics are showed in Figure 1(b-1). The clusters 2 and 3 vanish because they violate the Must-Link. The cluster 1 become bigger and the topics appear about 190 times. And in Figure 1(b-2), the clusters 2 and 3 vanish. In the distributions, T1 is response to A, B much more than C and T2 is response to C much more than A, B. But the TTM model cannot capture the information. It means the TTM-DF model can not only improve the underlying topics but also the tag-topic distributions. In Figure 1(b-1), the discrete points are mainly around [0.33A, 0.33B, 0.33C | C], which accord with the Must-Link (A, B). However, we assume that AB and C are not related, and they should not appear in the same topic, so we add a Cannot-Link(B, C). The results are shown in Figure 1(c-1) and 1(c-2). The discrete points almost disappear and the clusters in Figure 1(c-2) become more concentrated.

4.2 Experiments on Blog Corpus

The dataset is a collection of 1000 blogs from the Chinese blog corpus (http://pop.clr.org.cn/). It contains 819427 words and 4620 tags. After words segmentation and part of speech tagging, stop words, extremely common words are removed. Then, Only the nouns or nominal phrases are retained and other words are filtered. These preprocessing lead to a vocabulary size of 19789 unique words and 2452 unique tags. And in the experiments, the hyperparameters of the model are set as: α =0.1, β =0.01, μ =0.05.

4.2.1 Domain Knowledge Calculation

The domain knowledge incorporated in the model is a set of Must-links and Cannot-Links. We compute the relatedness between words, then set Must-links between words that are highly related and Cannot-links between words with low relatedness. In the experiments, two methods are used to compute the relatedness. One is based on HowNet [19]. If the relatedness of two words is larger than 0.9, a Must-link is set between them, and if the relatedness is smaller than 0.09, a Cannot-link is set. Finally, we select 196 Must-links and 124 Cannot-links. The other method is based on Wikipedia [20]. The thresholds for Must-links and Cannot-links are 0.45 and 0.08 respectively. And 128 Must-links and 120 Cannot-links are selected. The strength parameters η of the domain knowledge are both set to 1000.

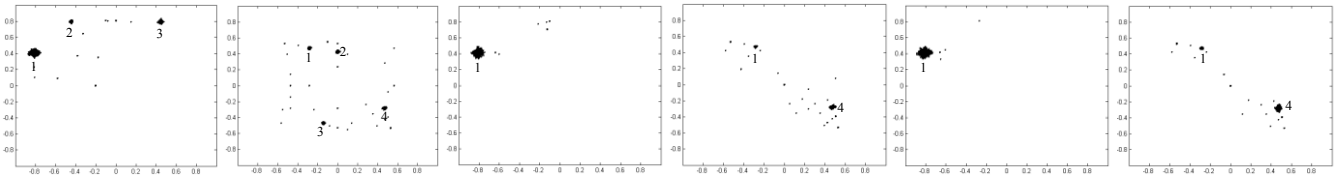
4.2.2 Results of Perplexity

Perplexity is used as the criterion of model evaluation, and lower perplexity score indicates better generalization performance. In this paper, the perplexity reflects the ability of the model in predicting words with the tags. It is equivalent to the inverse of the geometric mean of per-word likelihood. So, the perplexity can be calculated with the following formula:

$$Perplexity(D_{test}) = \exp\left\{-\sum_{d=1}^{D_{test}} \log(p(w_d \mid t_d)) \Big/ \sum_{d=1}^{D_{test}} N_d\right\}.$$

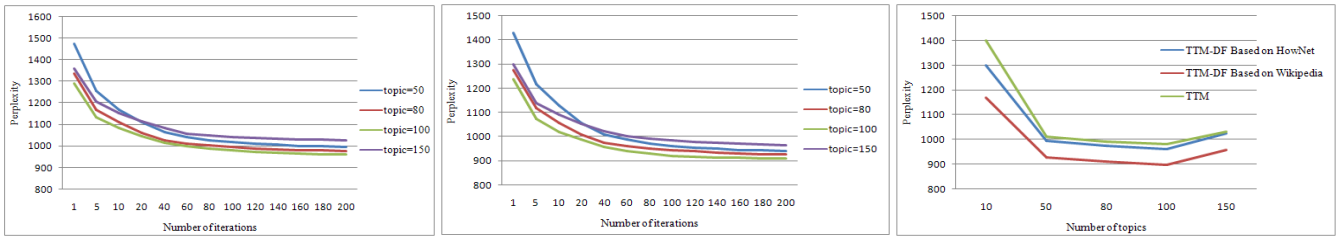
The number of topics K has great impact on the performance of the topic model. We test a serial of perplexity of the model for different number of topics. Figure 2(a) and 2(b) show the perplexities of different settings of K (K=50,80,100,150) with domain knowledge from HowNet and Wikipedia respectively. In general, while the number of iterations increases, all the perplexities with different topic numbers decrease and larger topic number always bring better perplexity from the beginning. But when the topic number is set too large (both K=150 in Figure 2(a) and 2(b)), the perplexity goes up. So we set K=100 which leads to the minimum perplexity in the following experiments.

The performance of the TTM-DF model based on HowNet (TTM-DF Based on HowNet) and Wikipedia (TTM-DF Based on Wikipedia), and the performance of the TTM model are shown in Figure 2(c). We can see that the TTM-DF Based on HowNet and TTM-DF Based on Wikipedia both perform better than the TTM with all the different numbers of K. It illustrates that they have better ability of word prediction. By incorporating the Must-links and Cannot-links, they can generate better topics, and they can also improve the tag-topic distributions, so the models can give better performance.



(a-1) (a-2) (b-1) (b-2) (c-1) (c-2)

Figure 1. PCA projections of samples



(a) (b) (c)

Figure 2. The perplexities of different models

The perplexities of the TTM-DF Based on Wikipedia are lower than that of the TTM-DF Based on HowNet, even though it adds less domain knowledge into the model. This means that the domain knowledge obtained from Wikipedia is better than that from HowNet. It is mainly because that some words are not included in the HowNet, the relatedness with other words cannot be calculated. And the relatedness between words in Hownet depend on their locations in the trees, so word pairs with different correlations will have same values if their path are similar in the trees. It is difficult to select the thresholds for Must-links and Cannot-links, and this will be influenced the quality of the knowledge. So we choose Wikipedia as the source of domain knowledge in the final experiments.

4.2.3 Examples of Tag and Topic Distributions

There is no quantitative measure for distributions of topic model, so we evaluate them by observing the top topics assigned to each tag and the top words assigned to the topics. Table 2 displays two examples. Each tag is illustrated with the top 2 topics and the top 10 words of the corresponding topics.

For the TTM-DF model, the topics 72 and 28 are the two top topics of the tag "电影(dian ying; English: movie)". By analyzing the top words, we can find that only the topic 72 is related to the tag. The theme of the topic 28 is "TV play". It can be the second top topic because its top words "导演(dao yan; English: director)" and "制片人(zhi pian ren; English: producer)" are related to the tag. For the TTM model, the top words of the topics 5 and 98 are both related to the tag "电影(dian ying; English: movie)" except some noise words such as "全国(quan guo; English: nationwide)", "教师(jiao shi; English: teacher)" and so on. These related words should be assigned to a same topic. The TTM-DF model merges the two topics together with a single Must-Link ("电影(dian ying; English: movie)", "票房(piao fang; English: box office)"). It illustrates that the Must-Link can pull other words (such as "投资方(tou zi fang; English: investor)" and "制片人(zhi pian ren; English: producer)") along even though they are not contained in the domain knowledge. And for the tag "黄岩岛(huang yan dao; English: Huangyan Island)", a Must-Link ("军舰(jun jian; English: warship)", "海军(hai jun; English: navy)") assigns words "军舰(warship)" and "国家(guo jia; English: nation)" to topic 13 together with word "海军(hai jun; English: navy)".

From Table 2, we can find that topics of the TTM-DF model are more concentrated than topics of the TTM model. It shows that the TTM-DF model can generate better topics with the constraints of Must-links and Cannot-links. Besides, the model can also improve the tag-topic distributions. The probabilities of the related topics are larger than before, so tags could be response to the related words much more than other words.

Table 2. Examples of distributions

Model	**TTM-DF**				**TTM**			
Tag	**电影(movie)**		**黄岩岛(Huangyan Island)**		**电影(movie)**		**黄岩岛(Huangyan Island)**	
Top Topics	topic 72 P=0.50640	topic 28 P=0.04392	topic 82 P= 0.53241	topic 13 P= 0.15172	topic 5 P=0.30281	topic 98 P=0.24597	topic 31 P=0.43003	topic 65 P=0.09415
Top Words	电影(movie) 票房(box office) 导演(director) 影片(film) 影院(cinema) 投资人(investor) 片子(film) 电影节(film festival) 制片人 (producer) 投资方(investor)	作品(work) 编剧(playwright) 电视剧(TV play) 纪录片 (documentary) 导演(director) 电视台 (TV station) 演员(actor) 制片人(producer) 题材(theme) 故事(story)	菲律宾(Philippines) 黄岩岛(Huangyan Island) 南海(South China Sea) 主权(sovereignty) 领土(territory) 渔民(fishermen) 菲方(Philippines) 海域(sea area) 事件(event) 中方(China)	国家(nation) 军舰(warship) 军事(military) 战争(war) 外交 (diplomacy) 航母(aircraft carrier) 战略(strategy) 问题(problem) 海军(navy) 武器(arm)	电影(movie) 导演(director) 影片(film) 影院(cinema) 电影节(film festival) 观众 (audience) 全国(nationwide) 文艺 (literature and art) 上海(Shanghai) 故事(story)	票房(box office) 市场(market) 老师(teacher) 片子(film) 作品(work) 版本(version) 投资方(investor) 类型(type) 制片人 (producer) 商业(commerce)	菲律宾(Philippines) 黄岩岛(Huangyan Island) 中国(China) 南海(South China Sea) 主权(sovereignty) 军舰(warship) 国家(nation) 领土(territory) 渔民(fishermen) 渔船(fishing boat)	问题(problem) 事件(event) 新闻(news) 态度(attitude) 利益(interest) 理由(reason) 方面(respect) 资源(resource) 行为(behavior) 事实(fact)

Table 3. Examples of semantic knowledge

Concept	Semantic knowledge
电影 (movie)	电影(0.4013, movie), 票房(0.2230, box office), 导演(0.1046, director), 影片(0.0796, film), 影院(0.0507, cinema), 投资人(0.0322, investor), 片子(0.0303, film), 电影节(0.0283, film festival), 制片人(0.0257, producer), 投资(0.0243, investor)
黄岩岛 (Huangyan Island)	菲律宾(0.2493, Philippines), 黄岩岛(0.2192, Huangyan Island), 南海(0.1163,South China Sea), 主权(0.0840, sovereignty), 领土(0.0731, territory), 渔民(0.0712,fishermen), 菲方(0.0518,Philippines), 海域(0.0499, sea area), 事件(0.0481, event), 中方(0.0371, China)
欧债危机 (European debt crisis)	经济(0.1948, economy), 欧洲(0.1832, Europe), 债务(0.1093, debt), 问题 (0.0860, problem), 危机(0.0798, crisis), 欧元(0.0743, Euro), 银行(0.0743, bank), 债券(0.0692, bond) , 财政(0.0685, finance), 希腊(0.0604, Greece)

4.2.4 Examples of Semantic Knowledge

With the tag-topic and topic-word distributions, the tags could be used to describe the topics. So, the semantic knowledge underlying in the topics can be represented explicitly and formally as $c:(w_1, r_1), (w_2, r_2), \ldots (w_i, r_i) \ldots (w_n, r_n)$. The tags were treated as concepts, and the top 10 words from the related topics were selected as related words. The probabilities of the words under the topics could be interpreted as the semantic relatedness. And we normalized the probabilities by their sum. Table 3 gives some examples of the semantic knowledge. we can find that the semantic knowledge obtained from blog is real-time, and some is the information of hot events, such as Huangyan Island, since some blogs are focused on the daily events.

The semantic knowledge is maneuverable. It can be easily utilized for relatedness computing, information retrieval, etc. Besides, the document-tag, tag-topic and topic-word distributions can be used for tag recommendation and other text mining applications.

5. CONCLUSIONS

In this paper, we expand the tag-topic model by incorporating domain knowledge via Dirichlet Forest prior. With Dirichlet Forest prior, topics have different Dirichlet Tree priors, and trees have different Dirichlet priors on words. So, word correlations can be encoded into the model. It can improve the coherence of the underlying topics and the tag-topic distribution. But the Must-links may lead to conflict as they are transitive. For example, if there are two Must-Links: Must-Link(apple, banana) and (apple, Jobs), it will generate a false transitive closure Must-Link(apple, banana, Jobs). In the future, we will use techniques of word sense disambiguation to prevent the transitive.

6. ACKNOWLEDGMENTS

This work was supported by the NSF of China (No.90920005, No.61003192), The Major Project of State Language Commission in the Twelfth Five-year Plan Period (No.ZDI125-1), Project in the National Science & Technology Pillar Program in the Twelfth Five-year Plan Period (No. 2012BAK24B01), the Program of Introducing Talents of Discipline to Universities (No.B07042), the NSF of Hubei Province (No.2011CDA034) and the self-determined research funds of CCNU from the colleges' basic research and operation of MOE (No.CCNU10A02009 and No.CCNU10C01005).

7. REFERENCES

[1] Tingting He, Fang Li. 2012. Semantic Knowledge Acquisition from Blogs with Tag-Topic Model. China Communications, 2012, 9(3): 38-48.

[2] DM Blei, AY Ng, MI Jordan. 2003. Latent Dirichlet Allocation. Journal of Machine Learning Research, 2003, 3(4-5): 993-1022.

[3] David Andrzejewski, Xiaojin Zhu, and Mark Craven. 2009. Incorporating Domain knowledge into topic modeling via Dirichlet Forest priors. In Proc. of ICML 2009, 25-32.

[4] Basu, S., Davidson, I., & Wagstaff, K. (Eds.). 2008. Constrained clustering: Advances in algorithms, theory, and applications. Chapman & Hall/CRC.

[5] S.P. Ponzetto, M. Strube. 2007. Deriving a large-scale taxonomy from Wikipedia. In Proc. of AAAI07, 1440–1445.

[6] Suchanek, F. M., G. Kasneci & G.Weikum. 2007. YAGO: A core of semantic knowledge. In Proc. of WWW-07, 2007

[7] Michael Strube and Simon Paolo Ponzetto. 2006. WikiRelate! Computing semantic relatedness using Wikipedia. In Proc. of AI-06, Boston, Massachusetts, USA, 2006.

[8] Xinhui Tu, Tingting He, Jing Luo and Long Chen. 2010. Wikipedia-based semantic smoothing for the language modeling approach to information retrieval. In Proc. of ECIR-2010, 370-381.

[9] Marius Pasca. 2004. Acquisition of Categorized Named Entities for Web Search. In Proc. of CIKM 2004. USA.

[10] Keiji Shinzato and Kentaro Torisawa. 2005. A Simple WWW-based Method for Semantic Word Class Acquisition. In Proc. of RANLP-05, 2005.

[11] Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In Proc. of SIGIR-99, 50-57.

[12] X. Wei and W.B. Croft. 2006. LDA-based document models for ad-hoc retrieval. In Proc. of SIGIR-06,178-185.

[13] R. Arora and B. Ravindran. 2008. Latent dirichlet allocation based multi-document summarization. In Proc. of AND-08, 91-97.

[14] Jordan Boyd-Graber, David M. Blei, and Xiaojin Zhu. 2007. A topic model for word sense disambiguation. In Proc. of EMNLP 2007.

[15] David Andrzejewski, Xiaojin Zhu. 2009. Latent Dirichlet Allocation with Topic-in-Set Knowledge. Proceedings of the NAACL HLT Workshop on Semi-supervised Learning for Natural Language Processing, 43-48.

[16] Jagadeesh Jagarlamudi, Hal Daume III, Raghavendra Udupa. 2010. Incorporating Lexical Priors into Topic Models. The 5th Annual Machine Learning Symposium, 2010.

[17] Minka, T. P. 1999. The Dirichlet-tree distribution (Technical report).http://research.microsoft.com/~minka/papers/dirichlet / minka-dirtree.pdf.

[18] T.L. Griffiths, and M. Steyvers. 2004, Finding scientific topics. Proceedings of National Academy of Sciences of the United States of America 101, 2004, 5228-5235.

[19] Qun LIU, Sujian LI. 2002. Word Similarity Computing Based on How-net, Computational Linguistics and Chinese Language Processing, 2002.

[20] Xinhui Tu, Tingting He, Hongchun Zhang and Kunfeng Zhou. 2012. Extracting Structured Information from Chinese Wikipedia and Measuring Relatedness between Words. Journal of Chinese Information Process, 2012.