TRM – Learning Dependencies between Text and Structure with Topical Relational Models

Veli Bicer[1], Thanh Tran[2], Yongtao Ma[2], and Rudi Studer[2]

[1] IBM Research, Smarter Cities Technology Centre
Damastown Industrial Estate, Dublin, Ireland
velibice@ie.ibm.com
[2] Institute AIFB, Karlsruhe Institute of Technology
Geb. 11.40, KIT-Campus Süd, Karlsruhe, Germany
{duc.tran,yongtao.ma,rudi.studer}@kit.edu

Abstract. Text-rich structured data become more and more ubiquitous on the Web and on the enterprise databases by encoding heterogeneous structural information between entities such as people, locations, or organizations and the associated textual information. For analyzing this type of data, existing topic modeling approaches, which are highly tailored toward document collections, require manually-defined regularization terms to exploit and to bias the topic learning towards structure information. We propose an approach, called *Topical Relational Model*, as a principled approach for automatically learning topics from both textual and structure information. Using a topic model, we can show that our approach is effective in exploiting heterogeneous structure information, outperforming a state-of-the-art approach that requires manually-tuned regularization.

1 Introduction

We study the problem of learning on *text-rich structured data* that shares these main characteristics: it describes interconnected objects (*relational*) of different types (*heterogeneous*) that are associated with textual attributes (*text-rich*). Examples include graph-structured RDF data forming connected resource descriptions and relational database records containing textual values that are connected via foreign keys.

Topic modeling (TM) approaches have shown to be effective for dealing with text, which recently, have also been extended to deal with the combination of textual and structured data [1–9]. However, when dealing with the text-rich structured data as we consider as an input in this paper (as shown Fig. 1), two problems mainly arise: First, such data mostly consists of a heterogeneous structure such as many classes and relations each of which has varying effects on different topics. Thus entities having different structure information in the data need to have different topic distributions. For example, any `Company` entity having `product` relation is more related to a topic about *manufacturing* than another `Company` entity of being a movie distributor. Usually such correlations

H. Alani et al. (Eds.): ISWC 2013, Part I, LNCS 8218, pp. 1–16, 2013.

between the topics and these structural elements (i.e. classes and relations) are not one-to-one, but the latter can be correlated to one or more topics with different proportions. Previous TM approaches are not well suited to handle such complex correlations between the structure and text since they either consider a homogeneous structure (e.g. social networks [5], citation networks [7] or Web links [8]) or networks with a few types of relations [9, 7]. An alternative solution to handle such complex correlations can also be applying *Statistical Relational Learning* (SRL) techniques [10] which are naturally formulated as instances of learning a probabilistic model (e.g. Markov Logic Network (MLN) [11] or (non-)linear link functions [12]) from the data. However the main problem of directly applying the SRL techniques into our setting is about handling *textual data* since they consider the input data to be available in a sort of structured form (e.g. a user-movie matrix indicating whether a user likes a movie). Only few works exist that utilize SRL to learn from text-rich structured data [13, 12] but mainly suffer from a large and complex structured network required for textual data. At this point, using the latent topics as low-dimensional representation of textual data provides a better way to incorporate textual information into the SRL models.

Yet another problem also arises due to *the sparsity of these correlations* between the topics and the structure. For example, in Fig. 2 the class `City` is highly correlated to the topic t_4 but not to the other three topics. Such sparsity is not well addressed by the previous work, especially the ones focusing on heterogeneous networks (e.g. [9, 7]) which aims the topic smoothness instead of sparsity. One exception to this is the focused topic model [14] which utilizes *latent feature models* to control sparsity but it is an unsupervised approach and does not take the structure information into account.

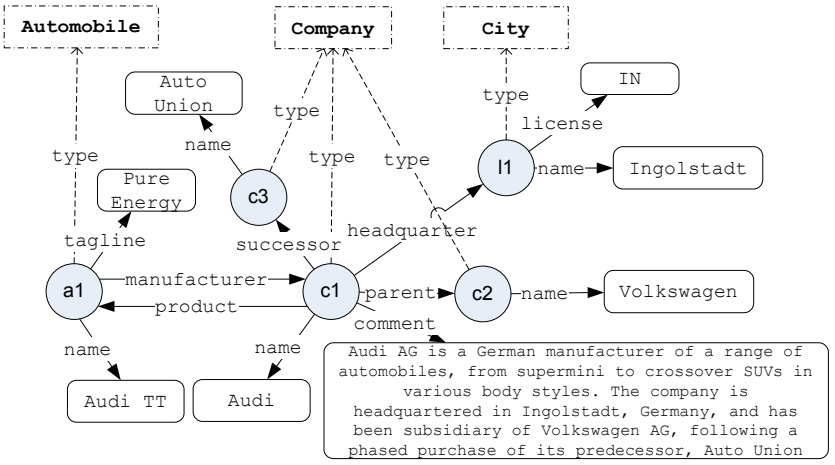

Fig. 1. An example data graph

In this paper, we propose *Topical Relational Model* (TRM) that uses relational information in structured data for topic modeling (text analysis tasks), and also allows the learned topics to be employed as a low-dimensional representation of the text to capture dependencies between structured data objects. The main novel aspects of this model are: (1) compared to previous SRL works, TRM employs hidden topic variables to reduce model complexity. TRM uses latent topics to represent textual information, targeting the specific case of text-rich structured data. In this way, it *provides a systematic way to learn low-dimensional features from text* (i.e., topics) that can be used for SRL-related tasks. We show in experiments that topics learned by TRM outperform the features manually specified in previous SRL works [12, 11]. (2) Compared to existing TM approaches, TRM is able to exploit the richness in relational information available in structured data. With TRM, the learning of topics recognizes the heterogeneity of classes and relations associated with entities. While some related TM work [9] requires manually defined regularization terms to exploit this heterogeneous structure information, TRM captures correlations between structure and topics through dedicated latent feature model parameter in order to better handle the sparsity of the correlations. In experiments, we show that leveraging relational information this way, TRM improves the performance of state-of-the-art TM approaches [9, 15]. (3) TRM is unique in its hybrid nature: it is a *topic model* that incorporates structure in addition to textual information; at the same time, it is also a *Bayesian network capturing relational dependencies* between text-rich objects that can be employed for SRL tasks.

Structure. We present the main ideas behind TRM in Sec. 2.1, TRM variables in Sec. 2.2 and their dependencies in Sec. 2.3. Sec. 2.4 describes the generative process, which is reversed in Sec. 2.5 for learning TRM. Experimental results are presented in Sec. 3, followed by conclusions in Sec. 4.

2 Topical Relational Models

TRM supports different types of *graph-structured data* including relational, XML and RDF data. The focus lies on text-rich data describing objects through textual attributes. More formally, the data is a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{R})$ (see Fig.1), where $\mathcal{V}$ is the disjoint union $\mathcal{V} = \mathcal{V}_C \uplus \mathcal{V}_E$ representing *classes* and *entities*, respectively, and $\mathcal{R}$ stands for binary *relations* between entities. The set of classes an entity e belongs to is denoted $C(e) = \{c \mid type(e, c)\}$ (*type* is a special relation that connects an entity node with a class node), while the relations e is involved in is $R(e) = \{r \mid r(e, e'), r(e', e) \in \mathcal{R} \wedge e, e' \in \mathcal{V}_E\}$. In our text-rich data setting, every entity also has some textual attribute values such as `name`, `comment` and `description`. To incorporate this, every entity node $e \in \mathcal{V}_E$ is treated as a document, i.e., modeled as a bag of words $e = \{w_1, \ldots, w_{|e|}\}$, which contains all words in the textual values of e.

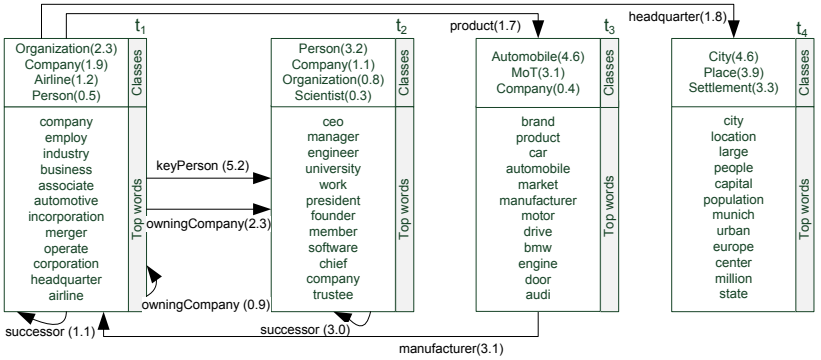

Fig. 2. An excerpt of the TRM result for DBpedia. It captures four TRM topics and and their top ranked words. Further, classes that are highly correlated with individual topics and relations that are highly correlated with pairs of topics are shown (strength of correlation shown in brackets).

2.1 TRM

TRM is a *topic model* representing $\mathcal{G}$ through a set of topics $T = \{t_i, \ldots, t_K\}$. Each $t \in T$ is a probabilistic distribution $\{p(w \mid t)\}_{w \in V}$, where $\sum_{w \in V} p(w \mid t) = 1$ and V is the vocabulary of words. However, the context from which topics are derived is not made up of plain words but also includes entities, classes and relations in $\mathcal{G}$. This is reflected in the TRM's output, which includes topics as well as their strength of correlation w.r.t. words, classes and relations. Fig. 2 illustrates this. For instance, we can see that the related words `employ` and `merger` form the topic t_1. Further, t_1 is not only drawn from words but also, correlates with entities of the types `Organization` and `Company`. This topic (and its associated words) correlates with other topics (words) such as t_3 (`brand` and `engine`) in the context of the `manufacturer` relation.

Because TRM also captures dependencies between structure elements (classes and relations), it can also be seen as a *SRL approach*. However, it captures them only indirectly using topics, which act as a low-dimensional abstraction of the text. We now present the TRM's Bayesian network representation to show these relational dependencies conditioned on topics.

2.2 Template-Based Representation of TRM

A template-based representation of Bayesian networks [16] is used to define TRM as a set of *template attributes* and *template factors*.

Let $\mathbf{X}$ denotes some of the random variables, $\mathbf{X} = \{X_1, ..., X_n\}$, where each $X_i \in \mathbf{X}$ can be assigned a value from the range $Val(X_i)$. Then, a *template attribute* (or attribute hereafter) is a function $A(\alpha_1, ..., \alpha_k)$, whose range is $Val(A)$, and each argument α_i is a placeholder to be instantiated with values of a particular type. For example, there are the template attributes $Company(\alpha_1)$ and $headquarter(\alpha_1, \alpha_2)$. The values used to instantiate α_i are drawn from the data,

called the *object skeleton*, $\mathcal{O}(A)$. Given $\mathcal{O}(A)$, the variables instantiating A is $\mathbf{X}_{\mathcal{O}(A)} = \{A(o) \mid o \in \mathcal{O}(A)\}$, where $Val(X_i) = Val(A)$ for each $X_i \in \mathbf{X}_{\mathcal{O}(A)}$. For example, from $Company(\alpha_1)$ and $\mathcal{O}(Company) = \{c1, c2, c3\}$, we obtain the random variables $\mathbf{X}_{\mathcal{O}(Company)} = \{Company(c1), Company(c2), Company(c3)\}$.

Template factors are used to define probability distributions over random variables. They are templates because instead of ground variables, template attributes are taken as arguments. They return real numbers for assignments of variables instantiated from template attributes, i.e., given a tuple of attributes $A_1, ..., A_l$, a template factor f is a function from $Val(A_1) \times ... \times Val(A_l)$ to $\mathbb{R}$. A special template factor is the *conditional probability distribution*, which splits the attributes into two groups, $\mathbf{A}_c, \mathbf{A}_{Pa} \subseteq \{A_1, ..., A_l\}$, called child and parent attributes.

Observed Variables. Elements in $\mathcal{G}$ are used to instantiate three template attributes defined for observed variables, namely the *class* c, the *relation* r and the *entity-word assignment* w. Their object skeletons are entities belonging to the type c, relations of the type r, and words in the entities' bag-of-words representation, i.e., $\mathcal{O}(c) = \{e|c \in C(e)\}$, $\mathcal{O}(r) = \{(e, e')|r(e, e') \in \mathcal{R}\}$ and $\mathcal{O}(w) = \{(e, v) \mid e \in \mathcal{V}_E \wedge v \in e\}$. These templates are binary-valued functions, indicating whether an entity, a relation instance or an entity-word assignment exists or not. For example, the variables $Company(c1)$, $product(c1, p1)$ and $w(c1, car)$ obtained for these templates model whether there is an entity `c1` that is a company, `p1` is a product of `c1` and `car` is a word associated with `c1`, respectively.

Observe that some entity-word assignments are dependent on a particular relation. For instance, the probability of observing the assignment $w(e, munich)$ is very high, given $headquarter(e, e')$ indicating "BMW, the company e, has its headquarter in Germany, the country e'". Further, such dependencies may exist only for some particular entities – e.g. not every entity that has its headquarter in Germany contains the word `munich` but some other words representing other cities in Germany. Instead of modeling dependencies between all observable variables (words and structure elements) directly, TRM models variables as being dependent on hidden topic-related variables.

Hidden Topic-Related Variables. The hidden topic variables are captured by the template attributes *topic indicator* b, *topic proportion* θ and *topic-word assignment* z. For these templates, we need object skeletons that also range over the topics T. In particular, b is instantiated with entities and topics in the skeletons $\mathcal{O}(b) = \{(e, t)|e \in \mathcal{V}_E, t \in T\}$. It is a binary-valued function such that for an entity e, $b_t(e) = 1$ indicates that t is a topic of e. The vector of all topic indicator variables of e is $\mathbf{b}(e) = \langle b_{t_1}(e), \ldots, b_{t_K}(e)\rangle$.

While $\mathbf{b}(e)$ is useful to determine which topics are present for an entity e, it does not capture sufficient information to model the probabilities of entity words. Following the tradition of topic modeling, θ is introduced for modeling entity words through a distribution of topics. While θ is defined over the same skeletons, it is different to b in that it is real-valued: for an entity e, $\theta_t(e)$ returns a

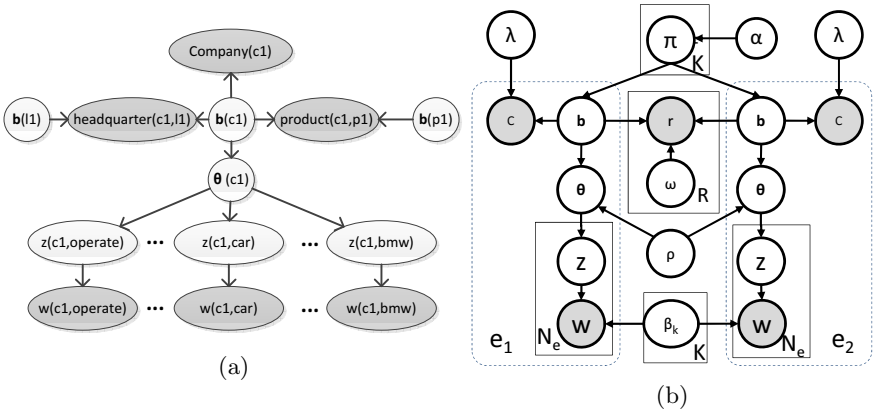

Fig. 3. (a) Template-based representation around the entity $c1$ with observed variables (dark) and hidden topic-related variables (light). (b) The generative process for two entities shown in plate notation.

real number and $\boldsymbol{\theta}(e) = \langle \theta_{t_1}(e), \ldots, \theta_{t_K}(e) \rangle$ defines a per-entity *topic distribution* such that $\sum_{t \in T} \theta_t(e) = 1$.

The semantics of the per-entity *topic-word assignment* is the same as in LDA. To capture this, we define a template attribute z that has the same skeleton as w. However, instead of a binary value, it returns a topic for an entity-word pair, i.e., $z(e, v) = t$ indicates that the word v associated with e belongs to topic t.

Fig. 3-a depicts variables obtained by instantiating the templates with information about the entity `c1`.

2.3 Probabilistic Dependencies in TRM

Central to TRM is the assumption that *given the topic indicator vector* of an entity, all its random variables derived from class and relation attributes are *conditionally independent*. That is, instead of capturing dependencies between these structure variables directly, we propose to use hidden topics. We want to capture that when entities exhibit structural resemblances, their topics shall also be similar. Vice versa, given some topics, some structure elements are more likely to be observed than others. We introduce the model parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$ to capture the quantity of how much a particular structure variable depends on a topic (pair of topics).

First, we consider that the probability of observing an entity e belonging to a class c depends on its topic indicator vector $\mathbf{b}(e)$. We model this as a template factor captured by a logistic sigmoid function defined over a linear combination of topic indicators, i.e.,

$$p(c(e) \mid \mathbf{b}(e)) = \sigma(\boldsymbol{\lambda}_c^T \mathbf{b}(e)) = \sigma\left(\sum_{t_i \in T} b_{t_i}(e)\ \lambda_{ci}\right) \tag{1}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the logistic sigmoid mapping values from $[-\infty, +\infty]$ to $[0, 1]$ and $\boldsymbol{\lambda}$ is a global parameter represented as a $|\mathcal{V}_C| \times K$ matrix. Each element λ_{ci} in the vector $\boldsymbol{\lambda}_c$ represents the strength of dependency between the class c and topic t_i.

Similarly, the probability of observing a relation $r(e_1, e_2)$ is modeled via logistic regression over the topic indicator vectors $\mathbf{b}(e_1)$ and $\mathbf{b}(e_2)$. A template factor over $r(e_1, e_2)$, $\mathbf{b}(e_1)$ and $\mathbf{b}(e_2)$ is defined as

$$p(r(e_1, e_2) \mid \mathbf{b}(e_1), \mathbf{b}(e_2)) = \sigma(\mathbf{b}(e_1)^T \boldsymbol{\omega}_r \mathbf{b}(e_2)) \tag{2}$$

where $\mathbf{b}(e_1)^T \boldsymbol{\omega}_r \mathbf{b}(e_2) = \sum_{t_k, t_l \in T} b_{t_k}(e_1) b_{t_l}(e_2) \omega_{rkl}$ and $\boldsymbol{\omega}_r$ is a $K \times K$ matrix. For any given two entities e_1 and e_2, where e_1 has the topic indicator t_k and e_2 has t_l, the weight of observing a relation r between these two entities is given as the value of the cell (k, l) of the matrix $\boldsymbol{\omega}_r$ denoted as ω_{rkl}.

Further, we employ the topic parameters θ and z to bring words into this picture. We want to capture that given some topics, some words are more likely than others. This part essentially follows the idea of topic modeling behind LDA. The only difference is that while LDA defines the topic proportion $\boldsymbol{\theta}(e)$ over all topics, $\boldsymbol{\theta}(e)$ here is defined only over the topics captured by the corresponding topic indicator vector $\mathbf{b}(e)$, i.e., topics not in $\mathbf{b}(e)$ have no density in $\boldsymbol{\theta}(e)$ (this creates a sparsity of topics similar to focused topic modeling [14]). To capture this, we introduce the template factor $p(\boldsymbol{\theta}(e) \mid \mathbf{b}(e))$.

This is an important design decision: On one hand, in order to handle the sparsity of dependencies that occur between the structure variables and the topics, the topics indicated in $\mathbf{b}(e)$ determines the probability of observing structure for the entity. On the other hand, the topic proportions of the entity in $\boldsymbol{\theta}(e)$ is governed by the topic indicator vector $\mathbf{b}(e)$ which in turn is determined by the structure around the entity.

2.4 Generative Process

We specify the full joint distribution for TRM and the generative process so that we can infer hidden variables from observed data in $\mathcal{G}$. First, we start with the vector $\mathbf{b}$ that specifies binary topic indicators for each entity. We use a prior distribution over the possible values in $\mathbf{b}$ in order to capture our initial uncertainty about the parameters. Obtaining such a prior is possible with the *Indian Buffet Process (IBP)*, which is a non-parametric Bayesian process used to generate latent features via a Beta-Bernoulli distribution [17]. IBP assumes that each entity e possesses a topic t with probability π_t, and that topic indicators in $\mathbf{b}(e)$ are then generated independently. Under this model, the probabilities of the topics are given as $\pi = \{\pi_1, ..., \pi_K\}$, and each π_t follows a Beta distribution with hyperparameter α, i.e., $p(\pi_t \mid \alpha) = Beta(\alpha/K, 1)$. Then, for an entity e, each topic indicator value is sampled from a Bernoulli distribution as $p(b_t(e) \mid \pi_t) = Bernoulli(\pi_t)$. IBP can be utilized for both finite and infinite number of topics [18]. Using infinite number of topics dynamically has great benefits in the case of an unsupervised topic learning so that number of topics gets larger whenever the need arises. However, in our case we are mostly interested in the

words of the entity to be distributed only to those topics selected for the classes and relations that entity has (no matter how many words the entity has, because we want to bias the topics according to structure). That's why, for the purpose of this work, we set the number of topics to a fixed K so that variational inference can be applied to learn the model in the exponential family [18]. For $\boldsymbol{\theta}(e)$, we set a Dirichlet prior over the topics just like in LDA. However, instead of using a uniform hyperparameter, we parametrize the Dirichlet with topic indicators $\mathbf{b}(e)$: $p(\boldsymbol{\theta}(e) \mid \mathbf{b}(e), \rho) = Dirichlet(\rho \mathbf{b}(e))$. As the number of selected topics varies according to $\mathbf{b}(e)$, each entity will have a different density of topic proportions. For the topic index z and word attribute w, the same process as defined for LDA involving the hyperparameter β is used. We arrive at the following generative process (see Fig. 3-b):

1. For each topic $t = 1, 2, .., K$:
   (a) Draw $\pi_t \mid \alpha \sim Beta(\alpha/K, 1)$.
2. For each entity e:
   (a) For each topic t:
      i. Draw $b_t(e) \mid \pi_t \sim Bernoulli(\pi_t)$
   (b) For each class c of entity e:
      i. Draw $c(e)$ using Eq. 1
   (c) Draw $\boldsymbol{\theta}(e) \mid \rho \sim Dir(\rho \mathbf{b}(e))$
   (d) For each word v of entity e:
      i. Draw topic index $z(e, v) \mid \boldsymbol{\theta}(e) \sim Mult(\boldsymbol{\theta}(e))$
      ii. Draw word $w(e, v) \mid z(e, v), \beta_{1:K} \sim Mult(\beta_{z(e,v)})$
3. For each pair of entities e, e':
   (a) For each relation $r \in \{r \mid r(e, e'), r(e', e) \in \mathcal{R}\}$:
      i. Draw $r(e, e')$ or $r(e', e)$ using Eq. 2

2.5 Learning

We propose to learn the posterior distribution of the hidden topic-related variables $\mathbf{b}, \boldsymbol{\theta}, \boldsymbol{\pi}$ and $\mathbf{z}$ conditioned on the observed variables $\mathbf{w}, \mathbf{c}$ and $\mathbf{r}$ via *variational Bayesian learning* [19]. Intuitively, the variational method approximates the posterior distribution of p by another simpler distribution q. In particular, through *mean field approximation* [19], we have q as a distribution that is fully-factorized over the hidden variables indexed by the free variational parameters $\boldsymbol{\nu}, \boldsymbol{\gamma}, \boldsymbol{\phi}$ and $\boldsymbol{\tau}$ for Bernoulli, Dirichlet, Multinomial and Beta distribution, respectively:

$$q(\mathbf{b}, \boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{z} \mid \boldsymbol{\nu}, \boldsymbol{\gamma}, \boldsymbol{\phi}, \boldsymbol{\tau}) = \prod_{t=1}^{K} \left[q(\pi_t \mid \tau_{t1}, \tau_{t2}) \prod_{e \in \mathcal{V}_E} q(b_t(e) \mid \nu_t(e)) \right]$$
$$\prod_{e \in \mathcal{V}_E} \left[q(\boldsymbol{\theta}(e) \mid \boldsymbol{\gamma}(e)) \prod_{(e,v) \in \mathcal{O}(w)} q(z(e, v) \mid \boldsymbol{\phi}(e, v)) \right] \tag{3}$$

These variational parameters are then fit such that q is close to the true posterior of p, where closeness is measured by the KL-divergence, $KL(q \parallel p)$. Because the decomposition $\log p(\mathbf{c}, \mathbf{r}, \mathbf{w}) = \mathrm{KL}(q \parallel p) + \mathcal{L}(q)$ and $\mathrm{KL}(q \parallel p) \geq 0$

hold [19], minimizing the KL-divergence is equivalent to maximizing the term $\mathcal{L}(q)$, the variational lower bound on the log marginal likelihood. The learning problem can then be expressed as optimizing

$$\{\tau, \nu, \gamma, \phi\} = \arg\max_{\{\tau,\nu,\gamma,\phi\}} \mathcal{L}(q) \tag{4}$$

For this, we use the 2-steps *variational Bayesian EM algorithm*. It takes the fixed hyperparameters α and ρ and an initial choice of the model parameters β, λ and ω as inputs. Then, it iteratively updates the variational parameters τ, ν, γ and ϕ until convergence in the E-step. Then, for fixed values of the variational parameters, the model parameters β, λ and ω are iteratively computed in the M-step. Thus, parameters are updated until convergence within the two steps, and both steps are run until convergence in the outer loop of the EM.

Variational E-Step. Update equations for this step can be obtained by setting the derivative of $\mathcal{L}(q)$ equal to zero. For each topic $t \in K$, we compute τ_{t1} and τ_{t2} of the Beta distribution as

$$\tau_{t1} = \frac{\alpha}{K} + \sum_{(e,t)\in\mathcal{O}(\boldsymbol{b})} \nu_t(e) \tag{5}$$

$$\tau_{t2} = 1 + |\mathcal{O}(\boldsymbol{b})| - \sum_{(e,t)\in\mathcal{O}(\boldsymbol{b})} \nu_t(e) \tag{6}$$

The update of $\nu_t(e)$ is given as $\nu_t(e) = \frac{1}{1+e^{\vartheta_t(e)}}$ where

$$\vartheta_t(e) = \vartheta_\tau + \sum_{c\in C(e)} \vartheta_c + \sum_{r(e,e')\in\mathcal{R}} \vartheta_{r1} + \sum_{r(e',e)\in\mathcal{R}} \vartheta_{r2} + \vartheta_\gamma \tag{7}$$

The update in Eq. 7 has five different parts. The contribution from the Beta prior can be computed by $\vartheta_\tau = \Psi(\tau_{t1}) - \Psi(\tau_{t2})$ where $\Psi(\cdot)$ is the digamma function. For each class $c \in C(e)$ the contribution to the update is given by

$$\vartheta_c = (1 - \sigma(\boldsymbol{\lambda}_c^T \boldsymbol{\nu}(e)))\lambda_{ct} \tag{8}$$

If the entity is the source of a relation, i.e., we have $r(e, e')$, the contribution is

$$\vartheta_{r1} = (1 - \sigma(\boldsymbol{\nu}(e)^T \boldsymbol{\omega}_r \boldsymbol{\nu}(e')))\omega_{rt.}\boldsymbol{\nu}(e') \tag{9}$$

or if it is the target, i.e. $r(e', e)$, the contribution is

$$\vartheta_{r2} = (1 - \sigma(\boldsymbol{\nu}(e')^T \boldsymbol{\omega}_r \boldsymbol{\nu}(e)))\omega_{r.t}\boldsymbol{\nu}(e') \tag{10}$$

and ϑ_γ is updated by

$$\vartheta_\gamma = \rho\ (\Psi(\gamma_t(e)) - \Psi(\sum_{t'} \gamma_{t'}(e))) \tag{11}$$

The variational Dirichlet parameter $\gamma_t(e)$ is

$$\gamma_t(e) = \rho\nu_t(e) + \sum_{(e,v)\in\mathcal{O}(w)} \phi_t(e, v) \tag{12}$$

The contribution to the update in Eq. 12 includes the variational multinomial $\phi_t(e, v)$ and also the variational parameter ν_t of the corresponding topic indicator b_t, i.e., $q(b_t(e) \mid \nu_t(e)$. This is the direct result of parameterizing the Dirichlet distribution with the topic indicator vector of each entity instead of using a non-informative prior α as in LDA.

The updates for the variational multinomial $\phi_t(e, v)$ is identical to that in variational inference for LDA [15]:

$$\phi_t(e, v) \propto \exp\{\log\beta_{tv} + \Psi(\gamma_t(e)) - \Psi(\sum_{t'} \gamma_{t'}(e))\} \tag{13}$$

where $\phi_t(e, v) = q(z(e, v) = t)$.

Variational M-Step. The update for the topic parameter β is the same as in LDA because also here, the words are conditionally dependent on β and z.

In order to fit the parameters λ and ω of the logistic regression defined by Eq. 1 and 2, respectively, we employ gradient-based optimization. At each iteration, we perform updates using the gradient

$$\nabla_{\lambda_{ct}} = \sum_{e \in \mathcal{V}_E} (1 - \sigma(\boldsymbol{\lambda}_c^T \boldsymbol{\nu}(e)))\nu_t(e) \tag{14}$$

for each class c and topic t and

$$\nabla_{\omega_{rtt'}} = \sum_{r(e,e') \in \in \mathcal{R}} (1 - \sigma(\boldsymbol{\nu}(e)^T \boldsymbol{\omega}_r \boldsymbol{\nu}(e')))\omega_{rtt'}\nu_{t'}(e') \tag{15}$$

for each relation r and topics t and t'.

These gradients cannot be used directly since they are only calculated for positive observations of classes and relations. For the unobserved cases ($r(e, e) = 0$) a regularization penalty is applied so the updates decreases ω and λ in each iteration for the topics controlled by the Beta-Bernoulli prior of b. This also introduces sparsity of the weights in ω and λ according to the topics selected in b. In particular, let ε be the number of observations (e.g. entities) for which the class membership is unknown such that $c(e) = 0$ and $\overline{\boldsymbol{\pi}}$ be a topic-indicator vector set to be the mean of the Beta-distributed variable π, $\overline{\boldsymbol{\pi}} = \frac{\boldsymbol{\tau}_1}{\boldsymbol{\tau}_1 + \boldsymbol{\tau}_2}$. Then, the regularization for $\boldsymbol{\lambda}_c$ is $\mathfrak{R}_{\boldsymbol{\lambda}_c} = -\varepsilon(\sigma(\boldsymbol{\lambda}_c^T \overline{\boldsymbol{\pi}}))\overline{\boldsymbol{\pi}}$. Similarly, let ζ be the number of observations where a particular relation is unknown (i.e. $r(e, e') = 0$). Then, the regularization term for $\boldsymbol{\omega}_r$ is $\mathfrak{R}_{\boldsymbol{\omega}_r} = -\zeta(\sigma(\overline{\boldsymbol{\pi}}^T \boldsymbol{\omega}_r \overline{\boldsymbol{\pi}})\overline{\boldsymbol{\pi}}$.

3 Experiments

First, we aim to obtain an initial understanding of the (1) *quality of the topic model* produced by TRM. Then, we provide a quantitative analysis of TRM by comparing its performance to state-of-the-art TM and SRL approaches w.r.t. the (2) *object clustering* and (3) *link predication* tasks, respectively.

Datasets. We use a subset of *DBpedia* containing 20,094 entities described by 112 distinct classes and 49 different types of relations. All attribute values are

treated as textual information and put into bags of words. The resulting vocabulary comprises 26,109 unique words after stop word removal. We also employ the *DBLP*[1] dataset. The abstract and title of the papers are treated as textual data. In addition, authors and conferences and their relations to papers are taken into account. We use a subset of papers that belong to the fields of database, data mining, information retrieval and artificial intelligence. In total, there are 28,569 paper, 28,702 author and 20 conference entities, and a vocabulary comprising 11,771 unique words.

3.1 Topic Analysis

A useful application of TRM is to understand the data. Fig. 2 displays the top words of four selected topics using the learned $\boldsymbol{\beta}$ parameter. Words are ranked by $score(v,t) = \beta_{tv}\left(\log\beta_{tv} - \frac{1}{K}\sum_{t'}\log\beta_{t'v}\right)$, which intuitively, assigns high scores to those words that are characteristic for a topic, relative to all other topics. We can clearly observe that structure elements (classes and relations) have an influence on the topics and the words that are ranked high for these topics. For example, t_1 has top words from entities of the type `organization` whereas t_2 captures words related to `person`. In particular, t_1 and t_2 have top words from those organization and person entities that are involved in the `keyPerson` relation. In fact, TRM not only exploits structure information for topic modeling but also explicitly models the strength of dependencies between topics and structure elements through the ω and λ parameters.

3.2 Link Prediction

Note that TRM captures the joint distribution over variables representing topics and structure elements. Thus, not only are topics dependent on structure elements but also vice versa, the existence of certain topics (and their words) can be used to infer that some structure elements are more likely than others. Here, we evaluate the effectiveness of using TRM topics for link prediction – based on the design and implementation used for the previous C^3 experiment [12]. We created training data using the `author` relations between papers and authors in *DBLP* and the `starring` relations between movies and actors in *DBpedia*. Then, this data is divided into a training and test set, with test data set to be 2, 4 and $\frac{3}{4}$ times the amount of training data.

However, it should be noted that the task of link prediction in SRL is different from LP for documents (e.g. [4, 5, 8]) in which topic similarity of documents is only distinctive feature. In the former each relation (e.g.starring,author) is characterized differently by its weights to features in a linear model like SVM. Thats why supervised TMs are not directly applicable on this task. Thus, in this experiment we show how the topic features based on ω of a relation are distinctive for link prediction beyond some base features such as the ones employed in C^3. Also note that, unlike other supervised TMs, TRM distinguishes different types of relations and a separate ω matrix is used for every relation, which assigns

[1] `http://www.informatik.uni-trier.de/ley/db/`

cross-topic weights. Link direction is also considered as the matrix omega is asymmetric.

Methods. We compare TRM against MLN [11] and C^3 [12]. To train the MLN, we use the open source Alchemy[2] implementation and adopt the rules as described in the Alchemy's tutorial for link predication. In order to predict links between two entities, C^3 employs SVM along with a set of features including Jaccard similarity computed from textual values of the two entities, words shared by the entities and adjacent nodes connected to the entities. LibSVM[3] is used to train a nu-SVM with a RBF kernel. The value of nu is set experimentally between 0.1 and 0.5. To use TRM for link prediction, we consider the combination of C^3 and the topics inferred by TRM. Namely, TRM provides two additional features that are then used by C^3. The first feature is a topic-based similarity score defined as $sim_r(e_1, e_2) = \sum_{t,t'} \sum_{v \in V_M} p(v \mid \beta_t)\, \omega_{rtt'}\, p(v \mid \beta_{t'})$ where V_M is the set of words e_1 and e_2 have in common. Instead of using these shared words only, we also use all the words in e_1 and e_2 to calculate a second feature using the formula above.

Table 1. Precision, recall and accuracy results for link prediction on DBLP and DBpedia

	DBLP(1-4)			DBLP(1-2)			DBLP(3-4)		
	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.
MLN	50.46	74.75	50.05	49.53	71.33	49.23	51.9	76.14	52.31
C^3	55.51	76.92	57.69	56.09	71.87	55.14	58.13	78.12	59.80
TRM	**66.03**	**84.84**	**67.34**	**65.98**	**83.87**	**69.53**	**68.83**	**85.54**	**71.25**
	DBpedia(1-4)			DBpedia(1-2)			DBpedia(3-4)		
	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.
MLN	50.72	**81.7**	51.08	50.4	**83.05**	50.5	51.45	**79.61**	52.03
C3	57.14	66.67	54.95	56.52	66.10	54.04	55.88	65.51	54.95
TRM	**72.71**	78.43	**69.04**	**70.58**	74.38	**67.84**	**71.68**	74.20	**67.28**

Results. We present the overall performance in Table 1. Also considering true negatives as depicted in Fig.4, we observe that while MLN can provide high recall for positive labeled data (achieves best recall for DBpedia in one setting), it does not perform well for negative labeled data. C^3 performs better than MLN in terms of precision and accuracy and also, achieves higher true negative rate. However, C^3's performance could clearly be improved when using TRM features in additional: TRM outperforms both baselines in terms of precision and accuracy and achieves average recall comparable to MLN. Also, it is more superior than these baselines in handling negative labeled data. The second feature provided by TRM captures the topical correlation between all words of the entities. It was particularly helpful in eliminating false negatives, i.e., two entities not correlated topic-wise w.r.t. ω_r are mostly not linked. The topic-based similarity calculated from matching words further helps to find true positives.

[2] `http://alchemy.cs.washington.edu/`
[3] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

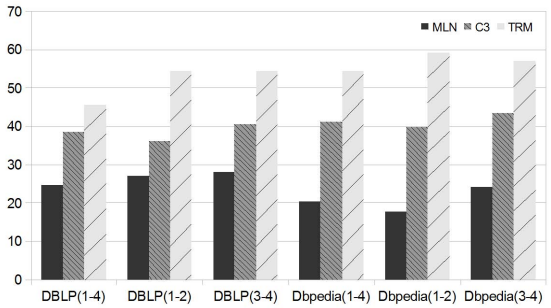

Fig. 4. True negative rate for DBLP and DBpedia

Table 2. Precision and normalized mutual information (NMI) results for object clustering on DBLP and DBpedia

Method/	Paper(%)		Author(%)		Venue(%)	
Metric	*Acc.*	*NMI*	*Acc.*	*NMI*	*Acc.*	*NMI*
LDA	47.22	15.97	–	–	–	–
TMBP-RW	69.11	45.24	74.67	65.61	65.79	67.48
TMBP-Reg	78.21	58.42	88.55	71.17	75.02	64.01
TRM	**89.35**	**65.51**	**93.44**	**78.16**	**87.73**	**75.11**

(a) DBLP

Method/	Movie(%)	
Metric	*Acc.*	*NMI*
LDA	58.71	22.29
TMBP-RW	57.10	27.65
TMBP-Reg	62.33	31.84
TRM	**71.57**	**44.28**

(b) DBpedia

3.3 Object Clustering

For DBPedia, we use entities of the type `movie`. Following the experiment performed previously [9], we use the six labels in DBLP representing various computer science fields as clusters. The clustering result is evaluated by comparing the label of each paper with the topic learned from the data.

Methods. We compare TRM to three TM approaches. As the most relevant baselines, we use two methods for learning topics from heterogeneous networks [9]: one model is learned with biased random walk (TMBP-RW) and the other results from biased regularization (TMBP-Reg). TMBP-RW propagates topic probabilities through the network via random walk, while TMBP-Reg achieves topic propagation through regularizing a statistical topic model with two generic terms. A previous experiment [9] has already shown that incorporating heterogeneous structure information as performed by these baselines helps to outperform clustering results of several existing (TM) approaches. For brevity, we thus include only the results of the standard LDA model [15]. Since LDA cannot be directly applied to heterogeneous information networks, we only use the bag-of-words representation of entities and ignore structure information.

Results. Table 2 shows the average results obtained from 10 test runs. TMBP-RW outperforms LDA on DBLP and is comparable to LDA on DBPedia. TMBP-Reg slightly outperforms TMBP-RW on both datasets. This suggests that exploiting structure information as supported by TMBP-RW and TMBP-Reg, leads to better results than LDA, which only considers word co-occurrences.

TRM leads to further improvements on both datasets by incorporating the effects of specific classes and relations on topics. For DBpedia for instance, TRM automatically infers that the structure elements `distributor` and `country` have a strong discriminative effect on assigning objects to the correct clusters.

4 Related Work

Related to our work, there are TM approaches proposed for homogeneous networks such as NetPLSA [2], Pairwise-Link-LDA [3], Nubbi [5], author-topic models [1], latent topic models for hypertext [6], citation networks [7] and relational topic models [8]. The major distinction between these models and TRM is that they consider a homogeneous network structure with only few types of entities and relations. In addition, more related to TRM, there are approaches over heterogeneous networks [9, 7] which utilize specific regularization functions to fit the topics to the underlying network structure. In general, as the network becomes more heterogeneous (i.e. more than two types of relations), more complex topic models are needed to capture complex correlations between the topic and structural variables. TRM mainly addresses this in a principled way by introducing sparsity of topics via topic indicators to create specific bias of topics towards structure information, i.e., classes and relations. In fact these approaches can be regarded as the extension of previous supervised topic models (e.g. [20–23]) to the networks in which observed variables are the relations instead of some tags or annotations. Also one similar work to ours is Type-LDA [24] which aims to discover the clusters of observed relation tuples and their associated textual expressions. However, that work only has a narrow focus on relation extraction in NLP and does not address the discovery of the correlations occurring in the whole data graph.

SRL works such as probabilistic relational models (PRM) [10] and MLN [11] learn graphical models using relational information. As discussed, this training is costly when the dependency structure is complex and the number of variables is high – which is particularly the case when a large amount of text is involved. We propose the use of hidden topic variables to reduce this complexity. To combine SRL with topic models, FoldAll [25] uses constraints given as first-order rules in a MLN to bias the topics by training a MRF. Although biasing the topics according to structure information can be accomplished through MLN, this approach does not capture correlations between topics and structure elements (e.g. predicate of rules). In addition, the number of groundings in the MLN rules poses a problem for FoldAll, since each grounding is represented as an indicator function in the corresponding topic model. TRM is unique in terms of using the topics as a low-dimensional abstraction to capture the correlations between the topics and classes/relations (i.e $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$).

5 Conclusion

We presented TRM, a novel combination of TM and SRL to learn topics from text-rich structured data. It captures dependencies between words in textual

and structured data through hidden topic variables in a *template-based model* constructed according to the underlying data structure. It represents a novel approach for automatically using heterogeneous structure information for learning topics as well as using topics to perform SRL tasks. In experiments, we show that compared to existing TM approaches, TRM is more effective in exploiting structure information. It reveals and exploits varying level of dependencies between topics and specific classes and relations, resulting in higher performance for both object clustering and link prediction.

As future work we plan to explore the extension of TRM to even richer generative models, such as time-varying and hierarchical topic models. In addition, potential application areas of TRM in the field of text-rich databases are manyfold. In particular, for selectivity estimation of structural queries comprising string predicates, TRM provides *a synopsis of the database* by capturing the topics and their correlations with classes and relations. This way, any structural query can be interpreted as a probability distribution, from which the query result size is estimated. We also consider TRM as being useful for *keyword search on structured data*. In existing work, keywords are mapped to database elements and connections between these keyword elements are discovered based on the relations given in the schema to compute structured results. The ranking of these results is separated from that computation. Instead of using the schema for discovering connections and a separate model for ranking, TRM can serve as a "probabilistic schema", capturing connections that are most probable. Hence, it can be used as a holistic model both for result computation and ranking based on their probability.

Acknowledgments. This research is partially supported by Siemens / DAAD Postgraduate Program under grant number A/10/94300.

References

1. Rosen-Zvi, M., Griffiths, T.L., Steyvers, M., Smyth, P.: The author-topic model for authors and documents. In: UAI, pp. 487–494 (2004)
2. Mei, Q., Cai, D., Zhang, D., Zhai, C.: Topic modeling with network regularization. In: WWW, pp. 101–110 (2008)
3. Nallapati, R., Ahmed, A., Xing, E.P., Cohen, W.W.: Joint latent topic models for text and citations. In: KDD, pp. 542–550 (2008)
4. Liu, Y., Niculescu-Mizil, A., Gryc, W.: Topic-link lda: joint models of topic and author community. In: ICML, p. 84 (2009)
5. Chang, J., Boyd-Graber, J.L., Blei, D.M.: Connections between the lines: augmenting social networks with text. In: KDD, pp. 169–178 (2009)
6. Gruber, A., Weiss, Y., Rosen-Zvi, M.: Hidden topic markov models. Journal of Machine Learning Research - Proceedings Track 2, 163–170 (2007)
7. Zeng, J., Cheung, W.K., Hung Li, C., Liu, J.: Multirelational topic models. In: ICDM, pp. 1070–1075 (2009)
8. Chang, J., Blei, D.M.: Relational topic models for document networks. Journal of Machine Learning Research - Proceedings Track 5, 81–88 (2009)

9. Deng, H., Han, J., Zhao, B., Yu, Y., Lin, C.X.: Probabilistic topic models with biased propagation on heterogeneous information networks. In: KDD, pp. 1271–1279 (2011)
10. Getoor, L., Taskar, B.: Introduction to statistical relational learning. MIT Press (2007)
11. Richardson, M., Domingos, P.: Markov logic networks. Machine Learning 62(1-2), 107–136 (2006)
12. Namata, G., Kok, S., Getoor, L.: Collective graph identification. In: KDD, pp. 87–95 (2011)
13. Singla, P., Domingos, P.: Entity resolution with markov logic. In: ICDM, pp. 572–582 (2006)
14. Williamson, S., Wang, C., Heller, K.A., Blei, D.M.: The ibp compound dirichlet process and its application to focused topic modeling. In: ICML, pp. 1151–1158 (2010)
15. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
16. Koller, D., Friedman, N.: Probabilistic graphical models. MIT Press (2009)
17. Ghahramani, Z., Griffiths, T., Sollich, P.: Bayesian nonparametric latent feature models. Bayesian Statistics 8, 1–25 (2007)
18. Doshi-Velez, F., Miller, K., Gael, J.V., Teh, Y.W.: Variational inference for the indian buffet process. Journal of Machine Learning Research - Proceedings Track, 137–144 (2009)
19. MacKay, D.: Information theory, inference, and learning algorithms. Cambridge Univ. Pr. (2003)
20. Blei, D.M., McAuliffe, J.D.: Supervised topic models. In: NIPS (2007)
21. Rubin, T.N., Chambers, A., Smyth, P., Steyvers, M.: Statistical topic models for multi-label document classification. Machine Learning 88(1-2), 157–208 (2012)
22. Blei, D.M., Jordan, M.I.: Modeling annotated data. In: SIGIR, pp. 127–134 (2003)
23. Yakhnenko, O., Honavar, V.: Multi-modal hierarchical dirichlet process model for predicting image annotation and image-object label correspondence. In: SDM, pp. 281–294 (2009)
24. Yao, L., Haghighi, A., Riedel, S., McCallum, A.: Structured relation discovery using generative models. In: EMNLP, pp. 1456–1466 (2011)
25. Andrzejewski, D., Zhu, X., Craven, M., Recht, B.: A framework for incorporating general domain knowledge into latent dirichlet allocation using first-order logic. In: IJCAI, pp. 1171–1177 (2011)