Entity Matching across Heterogeneous Sources

Abstract

Given an entity in a source domain, finding its matched entities from another (target) domain is an important task in many applications. Traditionally, the problem was usually addressed by first extracting major keywords corresponding to the source entity and then query relevant entities from the target domain using those keywords. However, the method would inevitably fails if the two domains have *less or no overlapping* in the content. An extreme case is that the source domain is in English and the target domain is in Chinese.

In this paper, we formalize the problem as entity matching across heterogeneous sources and propose a probabilistic topic model to solve the problem. The model integrates the topic extraction and entity matching, two core subtasks for dealing with the problem, into a unified model. Specifically, for handling the text disjointing problem, we use a cross-sampling process in our model to extract topics with terms coming from all the sources, and leverage existing matching relations through latent topic layers instead of at text layers. Benefit from the proposed model, we can not only find the matched documents for a query document, but also explain why these documents are related by showing the common topics they share.

1 Introduction

With the rapid growth of the Web, including online digital libraries, online social and information networks, and E-commerce systems, the Web provides abundant information to describe entities from different sources. Given an entity in a source domain, finding its matched entities from another (target) domain is an important task in many applications. For example, a patent expert may be interested in finding related patents in a patent database for a product described by a Wiki article; a user may be interested in finding all the related Chinese Wiki pages for a particular English Wiki page; and a doctor may be interested in finding all related drugs for a specific disease. Similar search problems can be found in many other applications.

The problem can be generalized as an entity matching problem across corpora from heterogeneous sources. In other words, given a document describing an entity (e.g., product) in one source, the goal is to find related documents describing other types of entities (e.g., patent) from a different source. Different from traditional search tasks, one key challenge of such problem is that different sources of corpora may use rather different languages or terminologies even when describing the same topic. For example, the terms used to express the same topic about Siri, are quite different in Wikipedia and patents. As Figure 1 (a) shows, the Siri Wiki article uses more daily expressions (e.g., "voice control," "personal assistant," "iPhone," etc.) to describe Siri, in order to make it easier understood by everyone. However, more professional and technique terms are used in patents (e.g., "information retrieval," "heuristic modules," "computer-readable medium," etc.). The two relevant documents from different sources can be very dissimilar in terms of their text similarity, and thus the traditional text-based search can no longer solve the problem. In addition, for each relevant documents, it would be interesting to know on which topic the target document is relevant to the source document. For example, as shown in Figure 1 (a), the patent "Method for improving voice recognition" is talking about "voice control" and its relevance probability to the source Wiki article on this topic is 0.83, while the relevance probability of the second patent is 0.54 but on topic "ranking".

One possible solution is to map two documents into the same latent topic space. Intuitively, two documents are relevant to each other if they refer to the same topic, e.g., a Wiki article and a patent article should be relevant if they are both talking about the topic of Siri. A topic in such case should contain terms from heterogeneous sources. For example, the topic of Siri should contain both the general terms in Wiki and the special terms in the related patents. If we can extract hidden topics from heterogeneous sources, we will be able to infer the relevance score between two documents. However, for most topic modeling methods, such as PLSA (Hofmann 1999) and LDA (Blei, Ng, and Jordan 2003), they do not deal with the issue of heterogeneous sources and are not able to generate topics with terms from different sources, since these terms seldom appear in the same documents.

In this paper, we propose a novel probabilistic model, Cross-Source Topic (CST) model, to solve the entity matching problem for a two-source case, which integrates the topic extraction and entity matching into a unified model. We first ask the users to give a small portion of labels indicating the matching between documents from hetero-

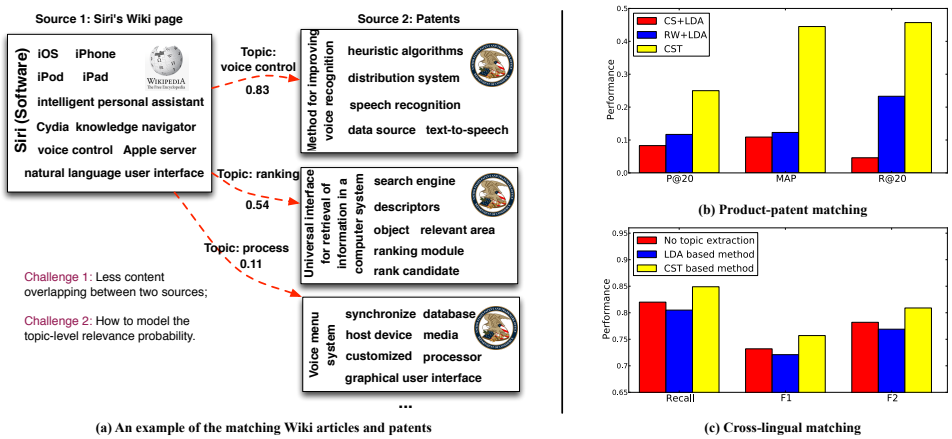

Figure 1: (a) An example of the matching between Wiki articles and patents. The rectangle on the left side represents the Wiki article which gives a general description of Siri. The rectangles on the right side denote patents reporting related technologies to Siri. Titles and high frequency phrases in the documents are shown in the rectangles. Links between the Wiki article and patents indicate their matching relations, with the topic relevance probability presented. (b) Product-patent matching performance of LDA based methods and the proposed model (CST). (c) Cross-lingual matching performance of a method not considering latent topics, a LDA based method, and a CST based method.

geneous sources. Then we model both the hidden topics and the entity matching in a unified framework, where a topic contains terms from heterogeneous sources and the entity matching is determined by the topic distributions of the two documents. By using this model, we can not only find the matched documents for a query document, but also explain why these documents are related by showing the common topics they share. It turns out that our model can successfully overcome the little-text-overlap problem across heterogeneous corpus sources, by modeling a topic with terms coming from all the sources and utilizing the matching labels for documents across different sources. A mean-field variational inference (Wainwright and Jordan 2008; Jordan et al. 1999) method is used to learn the model, which can be used to infer the matching relation between documents with no labels.

We evaluate the CST model in two real scenarios: 1) given a Wiki article describing a specific product, searching patents in the online patent database USPTO[1] that are related to the same product; 2) given an English Wikipedia article, searching the corresponding article from the Chinese Wiki knowledge base. Figure 1 (b)-(c) show the experimental results in each scenario respectively, from which we can see that the proposed model extensively improve the performance (averagely +19.8% and +7.1% in two real scenarios respectively).

Our contributions are summarized in the following.

- We identify and formalize a new problem called *entity matching across heterogeneous sources*, which is important and useful in this age of plentiful online open sources from different domains. To the best of our knowledge, no previous work has extensively studied this problem.
- We propose a novel and powerful probabilistic model, Cross-Source Topic (CST) model, to solve the entity matching problem for a two-source case, which integrates the topic extraction and entity matching into a unified model.
- We have demonstrated the power of our new method using two real-world applications, compared with the state-of-the-art baselines.

[1]http://www.uspto.gov/

2 Problem Definition

In this section, we present related definitions and formulate the problem. We first give the formal definition of heterogeneous source corpus. Generally, a heterogeneous source corpus contains documents from multiple sources. However, to make the definition and the description of the proposed model clear, we use a dual source corpus as an instance in all related definitions. We leave the source extension as future work.

Definition 1 ***Dual Source Corpus.*** *A dual source corpus C is a set of text collections $\{C_1, C_2\}$ from two sources with vocabulary $V_t = \{w_1^t, w_2^t, ..., w_{N_t}^t\}$ ($t \in \{1, 2\}$), where $C_t = \{d_1^t, d_2^t, ..., d_{D_t}^t\}$ is a collection of documents from source t, D_t is the number of documents in C_t, and N_t is the total number of words in V_t. Following the common assumption of bag-of-words representation, each document d_i^t in C_t can be represented as a bag of words $\{w_{i_1}^t, w_{i_2}^t, ..., w_{i_{N_i^t}}^t\}$, where N_i^t is the number of words in the document d_i^t.*

Given a dual source corpus, we can extract cross-source topics, which contain terms from different sources:

Definition 2 ***Cross-Source Topic.*** *A cross-source topic φ contains multiple multinomial distributions over words from different sources. For example, a 2-source topic contains two word distributions $P_1(w|\varphi)$ and $P_2(w|\varphi)$, where $P_t(w|\varphi)$ defines the probability of a word w from source t ($t \in$*

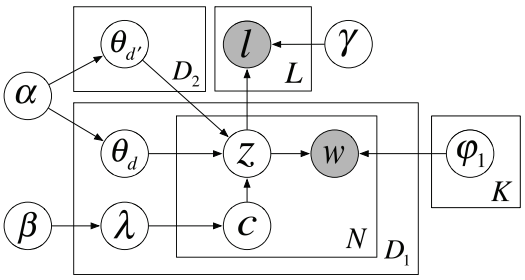

Figure 2: Proposed model. Modeling part for documents in source 2 has a symmetrical structure as source 1. For simplicity, the modeling part for the documents in source 2 is omitted.

$\{1, 2\}$*) appearing in this topic. Thus words with highest probabilities associated with each topic would suggest the semantics represented by the topic. Notice that we have* $\sum_{w \in V_t} p_t(w|\varphi) = 1$ *(*$t \in \{1, 2\}$*) for any cross-source topic* φ*.*

Next, we use a matching relation matrix to represent the correlations between documents from different sources.

Definition 3 ***Matching Relation Matrix.*** *A matching relation matrix* L *represents the matching status between documents in a dual source corpus* C*. If* d_i^1 *and* d_j^2 *is matched,* $l_{i,j} = 1$*, otherwise* $l_{i,j} = -1$*.* $l_{i,j} =?$ *denotes that the value is missing and needs to be inferred.*

Since documents from different sources may share few terms, the known values in the matching relation matrix are important guidance to extract the cross-source topics and infer the missing values in the matrix. We can finally define the main problem addressed in this paper:

Problem 1 ***Entity Matching across Heterogeneous Sources.*** *Given a heterogeneous source corpus* C*, and a matching relation matrix* L*. The goal of cross-source entity matching is to determine the missing values in* L*.*

3 Cross-Source Topic Model

3.1 Model Overview

Framework. The basic assumption of the proposed model is that, for documents from different sources, *their matching relations and hidden topics are influenced by each other*. Matching documents are similar in hidden space of topics, though the topics have different representations (e.g., word distributions) in different sources, and vice versa, documents that are similar in hidden space of topics tend to be matched. Thus the basic idea here is *to leverage the known matching relations to help the extraction of hidden topics, and use the extracted topics to infer the unknown relations*.

Figure 2 shows the plate representation of our semi-supervised model. For simplicity, we omit the modeling part for the words in source 2 as it is the same as source 1.

Cross-Sampling. We then introduce an important concept in the CST model: cross-sampling, which allows CST to leverage known relations and extract cross-source topics. The idea of cross-sampling is: when generating topics for a document d, the sampling process is not only based on the topic distribution of d, but also the topic distributions of all the matching documents of d. The intuition behind the idea is that the matched documents are similar in hidden space of topics. For example, a user would like to edit a Chinese Wikipedia article about "Barack Obama." Before he starts, he may take a look at what topics the corresponding English Wikipedia article contains, and finds out that the article contains Obama's early career as a Chicago community organizer. Thus he will edit the Chinese Wikipedia article to present Obama's experience as a community organizer but in different words. This process of cross-sampling allows us to bridge the topics in documents from different sources and model the cross-source topics.

3.2 Generative Process

The generative process consists of two parts: (1) cross-sampling-based document generation and (2) matching relation generation.

Cross-Sampling-Based Document Generation. Here, we introduce the document generation in detail. First, for each document d in source 1, we sample its topic distribution θ_d: $\theta_d \sim \text{Dir}(\alpha)$. Next, for each word w in d, we choose a topic z: $z \sim \text{Mult}(\theta_c)$, where c could be d itself or one of d's matching documents. We sample c according to $c \sim \text{Mult}(\lambda_d)$, where λ_d indicates how likely a document matched with d (including d itself) will be sampled. λ_d is sampled according to $\lambda_d \sim \text{Dir}(\beta_d)$, β_d is a $|D|$-dimensional vector, where $|D|$ is the total number of documents, and we define β_d as follows: we set $\beta_{d,d} = e_1$, where e_1 is a constant value denotes the weight of the prior to sample d's topics from its own topic distribution θ_d; for a document d' matched with d, we set $\theta_{d,d'} = e_2$, where e_2 is another constant value represents the weight of the prior to sample topics from one of d's matching documents; for other documents we set the corresponding values in β to 0.

With above definition, there is no chance to sample a document d's topics from documents not matching with d. If d has no matching relations, each z is sampled according to its own document topic distribution θ_d. Thus the generation of d is the same with LDA (Blei, Ng, and Jordan 2003).

Finally the word w is sampled according to the word distribution of topic z in source 1: $w \sim \text{Mult}(\varphi_{1,z})$. As different terminologies are used to represent the same topic in different sources, we separate the word distribution of a topic z into $\varphi_{1,z}$ and $\varphi_{2,z}$. We use source 1 as an example above and the documents in source 2 are generated in the same way.

Matching Relation Generation. In this step, each matching relation $l_{d,d'}$ is modeled as a binary variable. As documents with similar topic distributions tend to be matched with a higher probability, it is natural to model the probability of a matching relation as a function of topic distributions. There are many possibilities for the relation probability function ρ. In this paper, we consider the following form

$$\rho(l_{d,d'} = 1|\boldsymbol{z}_d, \boldsymbol{z}_{d'}, \gamma) \propto \exp[\boldsymbol{\gamma}^T(\tilde{\boldsymbol{z}}_d \circ \tilde{\boldsymbol{z}}_{d'})] \quad (1)$$

where the $\circ$ notation denotes the Hadamard product ($(\tilde{\boldsymbol{z}}_d \circ \tilde{\boldsymbol{z}}_{d'})_k = \tilde{z}_{d,k} \times \tilde{z}_{d',k}$), $\tilde{\boldsymbol{z}}_d$ is a K-dimension vector indicating the appearance of each topic in d, $\tilde{z}_{d,k} = \sum_{j=1}^{N_d} 1(z_{d,j} = k)$. The function ρ is parameterized by coefficients $\boldsymbol{\gamma}$. We define the function as an exponential one thus when $\boldsymbol{z}_d$ and $\boldsymbol{z}_{d'}$ are close, with large weighted Hadamard product, the probability increases exponentially.

A similar regression method is used in Relational Topic Model (RTM) (Chang and Blei 2009). The difference between RTM and CST is, RTM can hardly deal with the documents from multiple sources while CST bridges multiple sourced documents by learning how likely they will be influenced by each other (λ). Also, by cross-sampling, CST models a high-order dependency between matching documents and utilize the known relations more sufficiently.

3.3 Model Learning

We employ mean-field variational inference (Wainwright and Jordan 2008). Due to the space limitation, we only present the final update with each variational parameter below. See more details in our supplemental materials.

$$\eta_{d,c} = \beta_{d,c} + N_d \times \epsilon_{d,c} \quad (2)$$

$$\tau_{d,k} = \alpha_k + \sum_{n=1}^{N_d} \vartheta_{d,n,k} \quad (3)$$

$$\epsilon_{d,n,c} \propto \exp\{\Psi(\eta_{d,c}) - \Psi(\sum_{i \in R(d)} \eta_{d,i})\} \quad (4)$$

$$\vartheta_{d,n,k} \propto \sum_{d' \in \{R(d),d\}} (\exp\{\sum_{d'' \neq d'} \frac{\gamma_k \sum_{i=1}^{N_{d''}} \vartheta_{d'',i,k}}{N_{d'} N_{d''}} + \Psi(\tau_{d',k}) - \Psi(\sum_{j=1}^{K} \tau_{d',j})\} \epsilon_{d,n,d'} \times \varphi_{t,k,v}) \quad (5)$$

$$\varphi_{t,k,v} \propto \sum_{d=1}^{D_t} \sum_{n=1}^{N_d} \vartheta_{d,n,k} 1(w_{d,n}^t = v) \quad (6)$$

$$\gamma_k = \frac{\sum_{d,d'} 1}{2 \sum_{d,d'} l_{d,d'} [(\Upsilon_d - \Upsilon_{d'}) \circ (\Upsilon_d - \Upsilon_{d'})]_k} \quad (7)$$

where t is the source of document d, v is the n-th word of d, and $R(d)$ is a set of documents matched with d. Intuitively, Eq. 5 utilizes the known relations to update ϑ. The first summation in this equation is related with cross-sampling and the second one is based on the regression part of CST. These updates above are performed iteratively until convergence, since they depend on each other.

With all update equations above, we employ the variational expectation-maximization algorithm to learn the model, which yields the following iterations:

E-step: optimize the ELBO with respect to the variational parameters $\{\boldsymbol{\vartheta}, \boldsymbol{\tau}, \boldsymbol{\eta}, \boldsymbol{\epsilon}\}$. Update these variational parameters according to Eqs. 3-5.

M-step: maximize the resulting ELBO with respect to the model parameters $\{\boldsymbol{\varphi}, \boldsymbol{\gamma}\}$. Update the model parameters according to Eqs. 6-7.

Inferring Matching Relations. We finally detect the matching documents from different sources. Given a dual source corpus and a matching relation matrix with missing values, we use the learning algorithm from Section 3.3 to estimate the model's parameters by optimizing the ELBO for the observed data: words from the corpus and known relations in the matching relation matrix. After that, given two documents d and d' with an unknown relation ($l_{d,d'} = ?$), we use the fitted model's variational parameters to approximate the predictive probability:

$$P(l_{d,d'}|\boldsymbol{w}_d, \boldsymbol{w}_{d'}) \approx \mathbb{E}_q[p(l_{d,d'}|\boldsymbol{z}_d, \boldsymbol{z}_{d'})] \quad (8)$$

4 Experiments

4.1 Tasks and Data Sets

We validate the proposed model in two real scenarios: product-patent matching and cross-lingual matching. All data sets and codes used in this work are publicly available[2].

Product-patent matching. In this task, given a Wiki article describing a specific product, we aim to find relevant patents, e.g., a Wiki article and a patent should be relevant if they are both talking about the topic of Siri. We collect 13,085 Wiki articles and 15,000 patents from Wikipedia and USPTO respectively. For some Wiki article that describes a product, we use it as a query to find patents related with the same product. One Wiki article may be matched with more than one patent, e.g., a Wiki article describing iPhone corresponds to patents that claim on touch screen, camera, soft keyboard, etc.. We sample 233 Wiki articles as queries and find 1060 matching relations in total. We randomly choose 30% of the matching relations as known. The remaining relations are regarded as unknown and need to be inferred.

The ground truth data, which consists of 1060 matching relations, is labeled by four human annotators. For each of 233 Wiki articles as queries, each annotator reads all patents belonging to the same company with the corresponding product in the query. Some online systems and materials are referred when filtering the candidates and labeling the data (e.g., PatentMiner[3], news related with companies' lawsuit, official documents of the products, etc.). To see more details of how we label the data, please refer to our public web page[2]. We say a Wiki article is matched with a patent when four annotators all agree. Based on this work, we have deployed a product-patent matching function to PatentMiner. We are collecting user feedbacks to create a bigger evaluation data set for future work.

Cross-lingual matching. In this task, given an English Wiki article, we aim to find a Chinese article, which reports the same content, from a Chinese Wiki knowledge base. We collect the data set as follows: we randomly select an English article A with a cross-lingual link to a Chinese article B from Wikipedia, we then use the B's title to find another Chinese article C with the same title in Baidu Baike[4]. As

[2]We omit the URL here due to anonymity.

[3]A public patent search and analysis system: http://pminer.org

[4]A Chinese Wiki knoledge base: http://baike.baidu.com/

A is cross-lingually linked with B in Wikipedia, and B has the same main idea with C (normally a Wiki article uses its main idea as the title). It is reasonable to say there is a cross-lingual matching relation between A and C.

We totally collect 2,000 English articles from Wikipedia, and 2,000 Chinese articles from Baidu Baike. Thus in the data set, each English article corresponds to one Chinese article. The data set is from (Wang et al. 2012), which matches English / Chinese articles both from Wikipedia. We conduct 3-fold cross validation on the evaluation data set.

4.2 Evaluation

In the first experiment, for each Wiki article, we rank all patents according to the probability predicted by the proposed model and alternative methods. In the second experiment, to keep consistence with (Wang et al. 2012), we consider cross-lingual matching as a two-class classification problem: given an English Wiki article and a Chinese Wiki article, we label this pair of two documents as "matched" or "not matched".

Comparison methods. For the first experiment, we compare the following methods for product-patent matching:

• ***Content Similarity based on LDA (CS + LDA):*** It calculates the similarity between a Wiki article and a patent based on their topic distributions calculated by LDA. Specifically, we use p_{d_1} and p_{d_2} to represent the topic distribution of a Wiki article and a patent respectively. The similarity score is defined based on the Cosine similarity between p_{d_1} and p_{d_2}

$$Sim(d_1, d_2) = \frac{p_{d_1} \cdot p_{d_2}}{||p_{d_1}|| \times ||p_{d_2}||} \tag{9}$$

• ***Random Walk based on LDA (RW + LDA):*** It ranks candidates by combining the extracted topics into a random walk with restart algorithm (Tong, Faloutsos, and Pan 2006). Specifically, it creates a graph containing Wiki articles and patents as nodes. And it links a Wiki article u to a patent v with a weight

$$W_{u,v} = \begin{cases} \frac{Sim(u,v)}{\sum_w Sim(u,w)} & \text{if } Sim(u,v) \geq \mu \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where μ is a threshold value defined manually, and $Sim(u, v)$ is the Cosine similarity between u and v. Thus there is a bigger chance for a Wiki article node to reach a more similar patent node. It employs LDA to calculate the topic distributions. Besides the text contents of documents, this framework also considers the structural information. We create a link from one patent node to another if the former one cites the latter one. We also create a link from one Wiki article nodes to another if they have a hyperlink in Wikipedia. The weights of these links are defined as a constant value (in practice, we define all of them as 1). Finally, the transition probability from u to v can be defined as

$$Q_{u,v} = (1 - a)\frac{W_{u,v}}{\sum_x W_{u,w}} + a1(v = s) \tag{11}$$

where s is the start node, a is the restart probability.

• ***Relational Topic Model (RTM):*** It employs the RTM, which is generally used to model the links between documents, proposed by Blei et al. (Chang and Blei 2009). In our problem, this method regards there is a link between two matching documents. We use Blei's implementation of RTM[5].

• ***Random Walk based on CST (RW + CST):*** The difference between this method and RW + LDA is, instead of using $Sim(u, v)$ to define the weight of links from a Wiki node to a patent node, it uses $P(l_{u,v})$ (see Section 3.3 for details) calculated by CST.

• ***CST:*** It is our proposed model. We first use the training set to learn the model. Then we use the fitted model to detect unknown relations. We set $K = 50$, $\alpha = 50/K$, $e_1 = 4$, and $e_2 = 1$ in both this method and RW + CST.

For the second experiment, we compare the following methods for cross-lingual matching:

• ***Title Only:*** This method first translates the title of Chinese articles into English by Google Translation API [6], then matches the translated titles with English articles. Two articles are considered as equivalent ones if they have strictly the same English titles.

• ***SVM-S:*** It is a classifier proposed by Sorg et al. (Sorg and Cimiano 2008) to find cross-lingual links between English Wikipedia and German Wikipedia. The authors define several graph-based and text-based features. Here we train a SVM with their features on evaluation data set. For SVM, we choose LIBSVM (Chang and Lin 2011).

• ***LFG:*** It is the method proposed by Wang et al. (Wang et al. 2012), which is based on a factor graph model and mainly considers the structural information to solve the problem of cross-lingual matching.

• ***LFG + LDA:*** It adds a feature, which captures the content similarity between articles, to the feature function of LFG. It uses $Sim(u, v)$ (see Eq. 10) as the feature value.

• ***LFG + CST:*** LFG mainly considers structural information. We enhance it by bringing in content information (hidden topics extracted by CST). The difference between this method and LFG + LDA is that, instead of using $Sim(u, v)$ to define the newly added feature, it uses $P(l_{u,v})$ calculated by CST. We compare this method with LFG to see if content information can help in this problem. We compare it with Title Only and SVM-S to show the power of utilizing corss-lingual topics extracted by CST. We also compare it with LFG + LDA to show the effectiveness of the CST model compared with a traditional topic model. Here we keep values of K, α, and e_2 the same with the first task, and set $e_1 = 2$ empirically.

4.3 Quantitative Results

Product-patent matching. Table 1 lists the performance of product-patent matching problem using different methods. We first compare CST with two unsupervised methods, CS + LDA and RW + LDA. With the help of known relations

[5] http://www.cs.princeton.edu/~blei/topicmodeling.html

[6] https://developers.google.com/translate/?hl=zhcn

Table 1: Performance of product-patent matching task.

Method	P@3	P@20	MAP	R@3	R@20	MRR
CS + LDA	0.111	0.083	0.109	0.011	0.046	0.053
RW + LDA	0.111	0.117	0.123	0.033	0.233	0.429
RTM	0.501	0.233	0.416	0.057	0.141	0.171
RW + CST	**0.667**	0.167	0.341	**0.200**	0.333	0.668
CST	**0.667**	**0.250**	**0.445**	0.171	**0.457**	**0.683**

Table 2: Performance of cross-lingual matching task.

Method	Precision	Recall	F_1-Measure	F_2-Measure
Title Only	**1.000**	0.410	0.581	0.465
SVM-S	0.957	0.563	0.709	0.613
LFG	0.661	0.820	0.732	0.782
LFG + LDA	0.652	0.805	0.721	0.769
LFG + CST	0.682	**0.849**	**0.757**	**0.809**

as guidance, we can see CST clearly outperforms these two methods (+72.4%-75.5% in terms of MAP). We then compare CST with RTM, which also utilizes the known relations as guidance. With the help of the cross-sampling, CST can better extract cross-source topics. Thus it can better detect the matching relations (+74.9% in terms of MRR). To our surprise, when employing the CST model, combining content and structural information hurts the performance (RW + CST drops 23.4% in terms of MAP). By a careful investigation, we find that a Wiki article normally has lots of hyperlinks to other articles (56.4 out-links in average). Much noise is contained in these links and hurts the performance. However, the structural information does help for top results (+14.5% in terms of R@3).

Cross-lingual matching. Table 2 shows the performance of cross-lingual matching problem. Title Only and SVM-S employ the translated terminologies and perform well in terms of Prec. However, without capturing the hidden topics of documents, the translation can not be performed precisely. Thus these methods miss a number of matching relations between documents, which hurts the Recall.

LFG focuses on utilizing structural information. We enhance this method by bringing in hidden topics extracted by LDA and CST respectively. From the table, we see that LFG + CST improves the performance. It outperforms all baselines in terms of Recall, F_1, and F_2 (e.g., averagely +15.2% in terms of F_2). In fact, cross-lingual topics can hardly be extracted due to the low co-occurrence of English and Chinese terminologies. Without a precise cross-lingual topic extraction, LFG + LDA performs worse than LFG, which indicates the incorrect topics will hurt the performance. By studying some cross-lingual topics found by the CST model, we find that the top Chinese and English terminologies in the same topic are very relevant. Some Chinese terminologies are translated results of English ones.

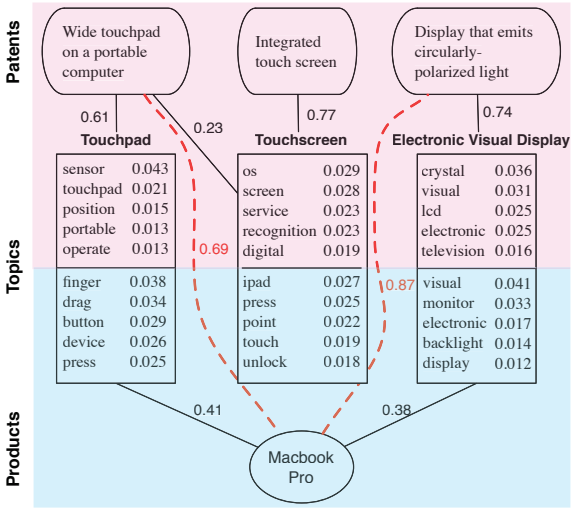

Figure 3: Examples of the correlations between topics, patents, and Wiki articles in the CST model. θ, the probability of a topic give a document, is represented on each black-solid edge. And the weight on each red-dotted edge denotes the likelihood of a matching relation. The titles of topics are hand-labeled. And for each topic, we separate the terminologies used in patents (the upper part of each topic box) and the terminologies used in Wiki articles (the lower part of each topic box). We remove some edges whose probabilities are negligible.

4.4 Qualitative Results

We further demonstrate some examples generated from our experiments to show the effectiveness of the CST model. Figure 3 shows a part of the matching results of "Macbook Pro" Wiki article. We select 3 topics extracted by the CST model and display them with top words in both two sources. We also represent the probability of a specific topic z given a document d ($\theta_{z,d}$), and the matching probability of two documents in the form of edges. As we can see from the figure, a patent mostly focus on one topic, a specific technology. And a Wiki article generally describe a number of features of a product. Thus Wiki articles have more diverse topic distributions.

When predicting a matching relation for two entities, the regression part of the CST mode is able to distinguish relevant topics from others. As the figure shows, CST mode successfully detects the Macbook Pro is matched with "Wide touchpad on a portable computer" and "Display that emits circularly-polarized light" respectively. Each of the two patents is associated with a topic relevant to Macbook Pro.

5 Conclusion

In this paper, we propose an approach to solve the problem of entity matching across heterogeneous sources. The model we proposed integrates the topic extraction and entity matching into a unified framework. We validate the model on two real scenarios. The experimental results demonstrate that our model can extensively improve the performance compared with baseline methods.

References

Airoldi, E. M.; Blei, D. M.; Fienberg, S. E.; and Xing, E. P. 2008. Mixed membership stochastic blockmodels. *JMLR* 9:1981–2014.

Barnard, K.; Duygulu, P.; Forsyth, D.; De Freitas, N.; Blei, D.; and Jordan, M. 2003. Matching words and pictures. *JMLR* 3:1107–1135.

Blei, D. M., and McAuliffe, J. D. 2010. Supervised topic models. *arXiv preprint arXiv:1003.0783*.

Blei, D.; Ng, A.; and Jordan, M. 2003. Latent dirichlet allocation. *JMLR* 3:993–1022.

Chang, J., and Blei, D. 2009. Relational topic models for document networks. In *Artificial Intelligence and Statistics*, 81–88.

Chang, C.-C., and Lin, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM TIST* 2:27:1–27:27.

Dietz, L.; Bickel, S.; and Scheffer, T. 2007. Unsupervised prediction of citation influences. In *ICML'07*, 233–240.

Erosheva, E.; Fienberg, S.; and Lafferty, J. 2004. Mixed-membership models of scientific publications. *PNAS'04* 101(Suppl 1):5220–5227.

Hofmann, T. 1999. Probabilistic latent semantic indexing. In *SIGIR'99*, 50–57.

Hofmann, D. C. T. 2000. The missing link-a probabilistic model of document content and hypertext connectivity. *NIPS'00* 13:430.

Jordan, M.; Ghahramani, Z.; Jaakkola, T.; and Saul, L. 1999. An introduction to variational methods for graphical models. *Machine learning* 37(2):183–233.

Liu, Y.; Niculescu-Mizil, A.; and Gryc, W. 2009. Topic-link lda: joint models of topic and author community. In *ICML'09*, 665–672.

Loeliger, H.-A. 2004. An introduction to factor graphs. *Signal Processing Magazine, IEEE* 21(1):28–41.

Lovász, L. 1993. Random walks on graphs: A survey. *Combinatorics* 2(1):1–46.

Mei, Q.; Cai, D.; Zhang, D.; and Zhai, C. 2008. Topic modeling with network regularization. In *WWW'08*, 101–110.

Mimno, D.; Wallach, H. M.; Naradowsky, J.; Smith, D. A.; and McCallum, A. 2009. Polylingual topic models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, 880–889.

Minka, T. 2000. Estimating a dirichlet distribution. Technical report, MIT.

Nallapati, R., and Cohen, W. 2008. Link-plsa-lda: A new unsupervised model for topics and influence of blogs. In *ICWSM'08*.

Nallapati, R.; Ahmed, A.; Xing, E.; and Cohen, W. 2008. Joint latent topic models for text and citations. In *KDD'08*, 542–550.

Sorg, P., and Cimiano, P. 2008. Enriching the crosslingual link structure of wikipedia-a classification-based approach. In *Proceedings of the AAAI 2008 Workshop on Wikipedia and Artifical Intelligence*, 49–54.

Tang, J.; Wu, S.; Sun, J.; and Su, H. 2012. Cross-domain collaboration recommendation. In *KDD'12*, 1285–1293.

Tong, H.; Faloutsos, C.; and Pan, J. 2006. Fast random walk with restart and its applications. In *ICDM'06*, 613–622.

Wainwright, M., and Jordan, M. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning* 1(1-2):1–305.

Wang, Z.; Li, J.; Wang, Z.; and Tang, J. 2012. Cross-lingual knowledge linking across wiki knowledge bases. In *WWW'12*, 459–468.

Winn, J. 2003. Variational message passing and its applications. *Unpublished doctoral dissertation, Cambridge University*.

Yakhnenko, O., and Honavar, V. 2008. Annotating images and image objects using a hierarchical dirichlet process model. In *MDM'08*, 1–7.