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 fail 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-
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.

## 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 $V_t = \{w_1, w_2, \ldots, w_N\}$ ($t \in \{1, 2\}$), where $C_t = \{d_{t1}, d_{t2}, \ldots, d_{tD_t}\}$ 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 $d_{ti}$ in $C_t$ can be represented as a bag of words $\{w_{ti1}, w_{ti2}, \ldots, w_{tin_t}\}$, where $N_{ti}$ is the number of words in the document $d_{ti}$.

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 $\varphi$ 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 $t$ ($t \in \{1, 2\}$).
lemmas in a dual source corpus. 

Matching Relation Matrix. ϕ

in the CST model: cross-sampling, which allows CST to

extract topics to infer the unknown relations

relations to help the extraction of hidden topics, and use the

Thus the basic idea here is

that are similar in hidden space of topics tend to be matched.

though the topics have different representations (e.g., word

Matching documents are similar in hidden space of topics,

ing relations and hidden topics are influenced by each other

Given a heterogeneous source corpus

Sources.

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 En-

glish Wikipedia article contains, and finds out that the arti-
cle 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.

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 rela-
tion 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

θd; θd ∼ Dir(α). Next, for each word w in d, we choose

a topic z: z ∼ Mult(θd), where c could be d itself or one of d’s matching documents. We sample c according to

c ∼ Mult(λd), where λd indicates how likely a document

matched with d (including d itself) will be sampled. λd is

sampled according to λd ∼ Dir(βd), βd is a |D|-dimensional

vector, where |D| is the total number of documents, and we

define βd as follows: we set βd,d = e1, where e1 is a con-

stant value denotes the weight of the prior to sample d’s
topics from its own topic distribution θd; for a document
d’ matched with d, we set θd,d’ = e2, where e2 is another

constant value represents the weight of the prior to sample
topics from one of d’s matching documents; for other docu-

ments we set the corresponding values in β to 0.

With above definition, there is no chance to sample a doc-

ument 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 θd. 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 dis-

tribution of topic z in source 1: w ∼ Mult(φ1,z). As different
terminologies are used to represent the same topic in differ-

ent sources, we separate the word distribution of a topic

into φ1,z and φ2,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 match-

ing relation l,d,d’ is modeled as a binary variable. As doc-

uments with similar topic distributions tend to be matched

with a higher probability, it is natural to model the proba-

bility of a matching relation as a function of topic distribu-
tions. There are many possibilities for the relation probability

function ρ. In this paper, we consider the following form

\( \rho(d, d’) \).

\( \rho(d, d’) \) 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

\( \sum_{w \in \mathcal{V}_t} p_t(w | \varphi) = 1 \) (t ∈ \{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 rela-

tion matrix L represents the matching status between docu-

ments in a dual source corpus C. If d1 and d2 is matched,

l12 = 1, otherwise l12 = -1. l12 = ? 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 in-
fer 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.
\[
\rho(l_{d,d'} = 1 | z_d, z_{d'}, \gamma) \propto \exp[\gamma^T(\tilde{z}_d \circ \tilde{z}_{d'})]
\]

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

\[
\eta_{d,c} = \beta_{d,c} + N_d \times \epsilon_{d,c}
\]

\[
\tau_{d,k} = \alpha_k + \sum_{n=1}^{N_d} \theta_{d,n,k}
\]

\[
\epsilon_{d,n,c} \propto \exp\{\Psi(\eta_{d,c}) - \Psi(\sum_{i \in R(d)} \eta_{d,i})\}
\]

\[
\theta_{d,n,k} \propto \sum_{d' \in \{R(d), d\}} \exp\{\sum_{d'' \neq d'} \frac{\gamma_{k} \sum_{i=1}^{N_{d''}} \varphi_{d''} - \varphi_{d',k}}{N_{d'} N_{d''}} + \Psi(\tau_{d',k}) - \Psi(\sum_{j=1}^{K} \tau_{d',j})\} \epsilon_{d,n,d'} \times \varphi_{k,\tau,\nu}
\]

\[
\varphi_{d',v} \propto \sum_{d=1}^{D_d} \sum_{n=1}^{N_d} \theta_{d,n,k} 1(w_{d,n} = v)
\]

\[
\gamma_{d} = \frac{2}{\sum_{d,d'} l_{d,d'}(\tilde{Y}_{d} - \hat{Y}_{d}) \circ (\tilde{Y}_{d'} - \hat{Y}_{d'})_{k}}
\]

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 \( \theta \). The first sumation 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 \( \{\theta, \tau, \eta, \epsilon\} \). Update these variational parameters according to Eqs. 3-5.

**M-step:** maximize the resulting ELBO with respect to the model parameters \( \{\varphi, \gamma\} \). 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:

\[
P(l_{d,d'} | w_d, w_{d'}) \approx \mathbb{E}_q[p(l_{d,d'} | z_d, z_{d'})]
\]

### 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 available2.

**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., PatentMiner3, 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 page2. 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 Baike4. As

2We omit the URL here due to anonymity.
3A public patent search and analysis system: http://pminer.org
4A Chinese Wiki knowledge 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 \( p_{d_2} \)

\[
\text{Sim}(d_1, d_2) = \frac{p_{d_1} \cdot p_{d_2}}{||p_{d_1}|| \times ||p_{d_2}||}
\]

- **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

\[
W_{u,v} = \begin{cases} \text{Sim}(u,v) & \text{if Sim}(u,v) \geq \mu \\ 0 & \text{otherwise} \end{cases}
\]

where \( \mu \) is a threshold value defined manually, and \( \text{Sim}(u,v) \) 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 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

\[
Q_{u,v} = (1 - a) \frac{W_{u,v}}{\sum_{v} W_{u,v}} + a1(v = s)
\]

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.

- **Random Walk based on CST (RW + CST):** The difference between this method and RW + LDA is, instead of using \( \text{Sim}(u,v) \) 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 \(^5\), 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 \( \text{Sim}(u,v) \) (see Eq. 10) as the feature value.

- **LFG + CST:** LFG mostly 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 \( \text{Sim}(u,v) \) 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 cross-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 \( K, \alpha, \) 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>P@3</th>
<th>P@20</th>
<th>MAP</th>
<th>R@3</th>
<th>R@20</th>
<th>MRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CST</td>
<td>0.667</td>
<td>0.250</td>
<td>0.445</td>
<td>0.171</td>
<td>0.457</td>
<td>0.683</td>
</tr>
<tr>
<td>LFG</td>
<td>0.652</td>
<td>0.805</td>
<td>0.721</td>
<td>0.769</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LFG + LDA</td>
<td>0.661</td>
<td>0.820</td>
<td>0.732</td>
<td>0.782</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTM</td>
<td>0.501</td>
<td>0.233</td>
<td>0.416</td>
<td>0.057</td>
<td>0.141</td>
<td>0.171</td>
</tr>
<tr>
<td>CS + LDA</td>
<td>0.111</td>
<td>0.083</td>
<td>0.109</td>
<td>0.011</td>
<td>0.046</td>
<td>0.053</td>
</tr>
<tr>
<td>RW + LDA</td>
<td>0.111</td>
<td>0.117</td>
<td>0.123</td>
<td>0.033</td>
<td>0.233</td>
<td>0.429</td>
</tr>
</tbody>
</table>

Table 2: Performance of cross-lingual matching task.

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
<th>F₁-Measure</th>
<th>F₂-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title Only</td>
<td><strong>1.000</strong></td>
<td>0.410</td>
<td>0.581</td>
<td>0.465</td>
</tr>
<tr>
<td>SVM-S</td>
<td>0.957</td>
<td>0.563</td>
<td>0.709</td>
<td>0.613</td>
</tr>
<tr>
<td>LFG</td>
<td>0.661</td>
<td>0.820</td>
<td>0.732</td>
<td>0.782</td>
</tr>
<tr>
<td>LFG + LDA</td>
<td>0.652</td>
<td>0.805</td>
<td>0.721</td>
<td>0.769</td>
</tr>
<tr>
<td>LFG + CST</td>
<td>0.682</td>
<td>0.849</td>
<td>0.757</td>
<td>0.809</td>
</tr>
</tbody>
</table>

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 performs all baselines in terms of Recall, F₁, and F₂ (e.g., averagely +15.2% in terms of F₂). 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.

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 θ given a document d (θd), 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 model is able to distinguish relevant topics from others. As the figure shows, CST model 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


