Transferring auxiliary knowledge to enhance heterogeneous web service clustering

Gang Tian*
College of Information and Science Engineering, Shandong University of Science and Technology, Shandong Qingdao 266590, China and State Key Laboratory of Software Engineering, School of Computer, Wuhan University, Hubei Wuhan 430072, China Email: tiangang@whu.edu.cn *Corresponding author

Chengai Sun
College of Information and Science Engineering, Shandong University of Science and Technology, Shandong Qingdao 266590, China Email: sun910213@163.com

Ke-qing He
State Key Laboratory of Software Engineering, School of Computer, Wuhan University, Hubei Wuhan 430072, China Email: hekeqing@whu.edu.cn

Xiang-min Ji
College of Computer and Information science, Fujian Agriculture and Forestry University, Fujian Fuzhou 350002, China Email: jixm168@126.com

Abstract: The growing number of web services puts forward higher requirements for searching desired web services, and clustering web services can greatly enhance the discovery of web service. Most existing clustering approaches are designed to handle long text documents. However, the descriptions of most services are in the form of short text, which impairs the quality of clustering owing to the lack of statistical information. To solve this problem, we propose a new service clustering approach based on transfer learning from auxiliary long text data obtained from Wikipedia. To handle the inconsistencies in semantics between service descriptions and auxiliary data, we introduce a novel topic model – dual tag aided latent Dirichlet allocation (DT-LDA), which jointly learns two sets of topics on the two datasets. Experimental results show the proposed approach achieves better performance than several existing approaches.

Keywords: web service clustering; auxiliary knowledge; topic model; transfer learning.


Biographical notes: Gang Tian is currently working toward his PhD degree in the State Key Lab of Software Engineering, Wuhan University. He is a student member of the IEEE. His research interests include web service discovery, transfer learning, feature engineering, knowledge extraction.

Chengai Sun is an Associate Professor at Shandong university of Science and Technology. Her research interests include requirement engineering and workflow.
1 Introduction

Along with the development of the service-oriented computing (SOC), web services are increasing continuously on the internet (Papazoglou et al., 2007). For example, Programmableweb (http://www.programmableweb.com) (PWeb) has published over 11,339 web services till 2014-4-24. The numbers of web services in PWeb increase 196% from 2011-9-2 to 2014-4-24. Consequently, how to find suitable web services for users becomes a challenging problem. Existing web service discovery methods tend to use web service search engines instead of universal description discovery and integration (UDDI) as many UDDI registries are no longer available. However, web service searching engines which primarily focus on keyword-based matching may suffer from a lack of key words in description or use of synonyms or variations of keywords (Elgazzar et al., 2009). Therefore, service clustering has been recently exploited to deal with these drawbacks and improve the search quality (Elgazzar et al., 2009; Yu, 2011; Yu and Rege, 2010; Platzer et al., 2009).

In fact, many studies have illustrated that clustering web services greatly boosts the ability of web service discovery. For example, Elgazzar et al. (2009) propose to extract features from web service description language (WSDL) documents to cluster web services. However, existing approaches of web service clustering still need further address some problems.

Firstly, web services are various and are described by kinds of different description languages. For example, PWeb provides 17 protocols for different web services and 14 of them have available services. However, existing methods of web service clustering mainly focus on a certain type of web service description file. Such as works (Elgazzar et al., 2009; Yu, 2011; Yu and Rege, 2010; Platzer et al., 2009) are mainly about WSDL and works (Pop et al., 2010; Cassar et al., 2010) are on OWL-S. As a result, these approaches fail to work when facing heterogeneous web services while it is important to interact with heterogeneous network environment (Chen et al., 2015; Chen and Chuang, 2015).

Secondly, keyword-based searching on web service descriptions provided by some web service registries (i.e., PWeb) suffers from insufficient statistic information of short text description (Tang et al., 2012). As shown in Table 1, some web services may not have descriptions (21/10050 in PWeb) such as Notify.Me in the first row, some descriptions may be too short to contain useful feature words such as OnlyWire in the second row, and others may not be functional description such as Microsoft MapPoint in the third row.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notify.Me</td>
<td>One method API. Allows submission of new URL.</td>
</tr>
<tr>
<td>OnlyWire</td>
<td>Note that this is a commercial, fee-based service but that 45 trial accounts are available.</td>
</tr>
<tr>
<td>Microsoft MapPoint</td>
<td>Note that this is a commercial, fee-based service but that 45 trial accounts are available.</td>
</tr>
</tbody>
</table>

In this paper, we handle heterogeneous web service clustering by using web service short text description and propose a topic model, dual tag aided latent Dirichlet allocation (DT-LDA) based on latent Dirichlet allocation (LDA) (Rosen-Zvi et al., 2004, 2010), to enhance web service clustering by incorporating auxiliary long texts and seamlessly leveraging the tagging information. Though heterogeneous web services may be described by WSDL or OWL-S or other languages, most of them (99.79% in PWeb) have short text descriptions which can be used to handle heterogeneous problems. There are few services with neither web service descriptions nor tags according to our statistics in PWeb and leveraging the tagging information can help improve the quality of service clustering (Chen et al., 2013), therefore, we use long texts from Wikipedia (http://www.wikipedia.org/) corresponding to tags to enhance web service clustering. The DT-LDA jointly learns a set of target topics on the web service description and another set of auxiliary topics on the long texts from Wikipedia corresponding to the tags while carefully modelling the attribution to each type of topics when generating documents. The contributions of this paper are as follows:
1 We propose a novel web service clustering approach DT-LDA based on a topic model, which transfers topical information generated from long texts from Wikipedia to enhance short text web service descriptions clustering.

2 DT-LDA utilises the short text descriptions of web services to handle the problem of heterogeneous web services clustering and can integrate tagging information seamlessly to improve the clustering performance.

3 We crawl 2,761 real web services and 1,065 Wikipedia entries concerning the tags of web services to evaluate the performance of DT-LDA.

The rest of this paper is organised as follows. Section 2 discusses the related work. Section 3 introduces the proposed DT-LDA model in detail. Section 4 reports our empirical experiments. Section 5 concludes this paper.

2 Related work

Recently, web service clustering has been demonstrated as an effective method to boost the ability of web service discovery (Elgazzar et al., 2009; Yu, 2011; Yu and Rege, 2010; Platzer et al., 2009). Based on description language, web services can be divided into two categories: semantic-based and non-semantic-based. For instance, ontology web language for services (OWL-S) and web service modelling ontology (WSMO) are two commonly used description languages for semantic-based approaches while WSDL and web application description language (WADL) are widely adapted for non-semantic-based approaches. Due to the difference, approaches for clustering web services are also classified into two categories: semantic and non-semantic web services. In our approach, we target the clustering of non-semantic web services.

The semantic-based approaches often use predefined ontologies to compute the similarity between web services in many existing works (Pop et al., 2010; Cassar et al., 2010; Dasgupta et al., 2010; Sun and Jiang, 2008). Especially, Pop et al. (2010) propose an ant-based method to cluster web services based on semantic similarity. Cassar et al. (2010) used PLSA and LDA to create an intermediate layer for service clustering. Sun and Jiang (2008) adopt Petri net for the specification of a service process model and cluster services based on functional similarity and process similarity. However, semantic-based approaches are not very popular because it is too costly to construct exact semantic representation for each web service.

Non-semantic web services clustering approaches use information retrieval techniques. Elgazzar et al. (2009) apply text mining techniques to extract content, types, messages, ports, and service name from WSDL documents to cluster web services. Liu and Wong (2009) extract four different features from the WSDL document to cluster services. Approaches based on matrix factorisation are also adopted to find relationship between services and operations by co-clustering them since the duality relationship can help achieve better quality (Yu, 2011; Yu and Rege, 2010). Integrating both WSDL documents and auxiliary data such as tags of the web service to cluster Web services can achieve better performance (Chen et al., 2013; Wu et al., 2012). Chen et al. (2013) use an augmented LDA model to blend in tags to improve clustering WSDL and propose three tag preprocessing strategies. Do and Hussain (2013) propose a personalisation algorithm for cloud computing which is a combination of the TOPSIS and Pearson correlation coefficient methods.

There are several limitations for above mentioned approaches: Firstly, all these approaches are based on single web service description. This may result in poor recall facing heterogeneous context. Secondly, all these approaches are dealing with machine readable description but sometimes the machine readable description documents are hard to obtain. For instance, WADL for RESTful web service is often registered in the service provider’s server and is hard to collect. Finally, although the service description in natural language is easy to collect, few works have paid attention to it which may due to the sparsity of representation.

Our work leverages the idea of transfer learning which studies transferring knowledge from auxiliary domain to help learning task in the target domain (Jin et al., 2011; Hu et al., 2009; Phan et al., 2011). Many transfer learning approaches have been proposed to handle short text clustering. Hu et al. (2009) introduce a short text clustering method by using world knowledge. Phan et al. (2011) convert auxiliary information to topics to improve the representation of the short texts. Both works (Hu et al., 2009; Phan et al., 2011) generally make the implicit assumption that the auxiliary data are semantically related to the short texts, which may have difficulties in practice due to the noisy nature of the auxiliary information. Jin et al. (2011) offer a transfer learning approach which can facilitate short texts clustering by using auxiliary long texts. However, their approach cannot integrate tag information which highly improves the performance of clustering in web services (Chen et al., 2013; Wu et al., 2012). To the best of our knowledge, there is still no reported approach on leveraging transfer learning techniques combined with tagging information in web service clustering.

In this paper, we focus on the clustering of non-semantic web services described by using short text in natural language. To solve the problems described above, we propose a novel web service clustering approach in which long texts from Wikipedia are used to transfer topical information to enhance short text web service descriptions clustering.
A framework of auxiliary knowledge aided web service clustering

In this section, we describe the proposed approach of web service clustering in detail.

3.1 Web service clustering framework

As shown in Figure 1, the clustering process in the framework is divided into three steps:

1. data preprocessing
2. generating hidden topical representation
3. service clustering with hidden topics.

In step 1, both the short text descriptions and tags of web services are crawled from the PWeb. Then each stemmed tag is used to crawl its entry in Wikipedia. The entries of tags will be used to construct auxiliary documents. At the end of step 1, the raw descriptions of web services and tags are converted to feature words which will be the input of DT-LDA. In step 2, the DT-LDA which is composed of two topic models is used to leverage auxiliary information and integrate tags information to create hidden topical representation of the web services. In step 3, traditional clustering methods can be utilised to cluster web services on the hidden topics which are generated in step 2.

The major purpose of service clustering is effective service discovery. Since data preprocessing and service clustering are conducted offline, we do not need to consider the efficiency of service discovery. Hence, the focus will be placed on accuracy.

3.2 Data preprocessing

In this step, both the short text description and the auxiliary long description of web services are crawled from the internet and are preprocessed by using feature extraction technology. Firstly, we crawl short text descriptions and tags of web services from PWeb. Secondly, we perform tokenisation over short text web service descriptions and stem the tags by using NLTK (http://www.nltk.org/). Thirdly, the stemmed tags are used to generate auxiliary long text descriptions. For instance, service *OnlyWire* has two tags: *bookmarks* and *deadpool*. Firstly, the Wikipedia entry of each tag (such as bookmark) (http://en.wikipedia.org/wiki/Bookmark) is crawled. Secondly, the source code of the entry is then parsed and the information in the markup of ‘content’ is collected to form the description of the tag. Lastly, descriptions of *bookmarks* and *deadpool* are merged to form a long text description for *OnlyWire*. Finally, we get two descriptions for *OnlyWire*: one is the short text description, and the other is the auxiliary long text. Then both the short descriptions and long auxiliary descriptions are preprocessed to generate feature vector by using the following steps.

Firstly, we perform tokenisation over both short and long descriptions of all web services to produce the original content vector. Secondly, we stem all words that have the same stem by using the Porter stemmer. Thirdly, function words such as a, the, etc. that have little or no contribution to the meanings of contents are removed.

Figure 1  A framework of web service clustering (see online version for colours)
3.3 Generating hidden topical representation web services clustering

In this step, the DT-LDA is proposed to transfer the topical knowledge from the auxiliary long texts to help generate the hidden topical representation of all web services. We first introduce the LDA model which DT-LDA is generated from and then we illustrate DT-LDA in detail.

3.3.1 Author topic model

The author topic model (ATM) (Rosen-Zvi et al., 2004; 2010) is an augmented LDA model whose probabilistic graphical model is shown in Figure 2(a). In ATM, a document is generated by the following process: firstly, an author is chosen uniformly for each word in the document. Secondly, a topic is sampled for each word from the distribution over topics associated with the author of that word. Finally, the words are sampled from the distribution over words associated with each topic (Rosen-Zvi et al., 2004; 2010).

![Figure 2](image)

(a) probabilistic graphical model of ATM (b) matrix factorisation interpretation of ATM

In the context of clustering web services with tags, tags in a web services are observed and chosen uniformly. They play the same role as authors in an ATM. Therefore, we can simply modify ATM such that the authors in a document are regarded as the tags in a web service. Based on this explanation, web services can be interpreted as a form of matrix factorisation by three layers after using ATM as shown in Figure 2(b) (Chen et al., 2013).

\[ P = \Phi \times \Theta \times T \]

In DT-LDA, \( W^{tar} = \{w^{tar}_m\}_{m=1}^{M^{tar}} \) denotes the set of short texts from the target domain (i.e., domain of web service), which are to be clustered, and \( W^{aux} = \{w^{aux}_m\}_{m=1}^{M^{aux}} \) denotes the auxiliary documents (i.e., domain of long auxiliary texts).

To capture the major themes within the two datasets, we can simply split the topics in a DT-LDA model into two groups. In particular, we assume that there are \( K^{tar} \) topics with parameters \( \phi^{tar}_k \) in the target domain and \( K^{aux} \) topics with parameters \( \phi^{aux}_k \) in the auxiliary domain. To control the model generating a document by using the topics belonging to their domain, we simply tune the hyper-parameter vector \( \alpha \) which determines the prior distribution for probabilities of topics given tag \( \Theta \). Traditionally, without any prior knowledge, the entries of \( \alpha \) are set to equal which means there is no preference on particular topics. To control the DT-LDA prefer target topics to generate documents, we can set the target entries to be greater than the auxiliary entries. This leads to having two separate asymmetric ATMs with parameters

\[
\alpha^{tar} = [\alpha^{tar}_{1}, \ldots, \alpha^{tar}_{K^{tar}}, \alpha^{aux}_{K^{tar}+1}, \ldots, \alpha^{aux}_{K^{aux}}]
\]

and

\[
\alpha^{aux} = [\alpha^{aux}_{1}, \ldots, \alpha^{aux}_{K^{aux}}, \alpha^{aux}_{K^{aux}+1}, \ldots, \alpha^{aux}_{K^{aux}}]
\]

whose generation process is shown in Algorithm 1.
Algorithm 1  The generation of DT-LDA

1. For each target topic $z \in \{1, \ldots, K_{\text{tar}}\}$, draw a multinomial distribution over terms, $\phi^\text{tar}_z \sim \text{Dir}(\beta^\text{tar})$.
2. For each tag $t \in \{1, \ldots, T\}$, draw a multinomial distribution over terms, $\theta^\text{tar}_t \sim \text{Dir}(\alpha)$.
3. For each auxiliary topic $z \in \{1, \ldots, K_{\text{aux}}\}$, draw a multinomial distribution over terms, $\phi^\text{aux}_z \sim \text{Dir}(\beta^\text{aux})$.
4. For each corpus $c \in \{\text{aux}, \text{tar}\}$:
   a. For each web service description $d \in \{1, \ldots, M^c\}$:
      i. Draw a multinomial distribution over auxiliary topics, $\theta^\text{aux}_d \sim \text{Dir}(\alpha)$.
      ii. Draw a multinomial distribution over terms, $\phi^\text{aux}_d \sim \text{Dir}(\beta^\text{aux})$.
      iii. If $c = \text{tar}$:
          a. Given the vector of Tags $T_d$
             1. For each word $i = 1, \ldots, N_d$
                i. Conditioned on $T_d$, choose a tag $x_{di} \sim \text{unifom}(T_d)$
                ii. Conditioned on $x_{di}$, choose a topic $z_{di} \sim \text{Multinomial}(\phi^\text{tar}_{z_{di}})$
                iii. Conditioned on a word $x_{di}$, choose a topic $z_{di} \sim \text{Multinomial}(\phi^\text{aux}_{z_{di}})$
       iv. If $c = \text{aux}$:
         2. For each word $i = 1, \ldots, N_d$
             i. Draw a multinomial distribution over auxiliary topics, $\theta^\text{aux}_d \sim \text{Multinomial}(\theta^\text{aux}_d)$
             ii. Draw a multinomial distribution over terms, $\phi^\text{aux}_d \sim \text{Dir}(\beta^\text{aux})$

All symbols and their description are displayed in Table 2.

From the generative graphical model, we can write the joint distribution of all known and hidden variables given the hyper parameters:

$$P(\tilde{W}_{ij}, z, \bar{x}, \tilde{\theta}, \Phi | \alpha, \beta, T)$$

$$= \prod_{i=1}^{N_d} P\left(\tilde{w}_{di} | \tilde{\phi}_{z_{di}}\right)P\left(z_{di} | \bar{\theta}_{di}\right)p(x_{di} | t_d)$$

$$\cdot P(\bar{\theta}_{di} | \tilde{\alpha}) \cdot P(\Phi | \tilde{\beta}) \cdot P(\bar{x} | t_d)$$

(2)

The likelihood of the whole dataset $W = \{\tilde{w}_{di} | d=1\}^{D_{\text{tar}}}_{d=1} \cup \{\tilde{w}_{dx} | d=1\}^{D_{\text{aux}}}_{d=1}$ under the model is:

$$P(W | \alpha, \beta, T) = \prod_{d=1}^{D_{\text{tar}}} P\left(\tilde{w}_{dx} | \tilde{\alpha}, \tilde{\beta}, T\right)$$

$$\prod_{d=1}^{D_{\text{aux}}} P\left(\tilde{w}_{dx} | \tilde{\alpha}, \tilde{\beta}, T\right)$$

(3)

To estimate the parameters, we need to estimate the latent variables conditioned on the observed variables. We perform approximate inference using Gibbs sampling with the following updating formulas. For target topics $z_{di} \in \{1, \ldots, K_{\text{tar}}\}$:

$$P(z_{di} = k, x_{di} = k, y_{di} = y | w_{di} = w, z_{-di}, x_{-di}, w_{-di}, \alpha, \beta, t_d)$$

$$\propto C_{K_{\text{tar}}, w_{-di}}^w + \beta C_{K_{\text{tar}}, y_{-di}}^y + \alpha C_{K_{\text{tar}}, z_{-di}}^z$$

(4)
For auxiliary topics $z_{di} \in \{1, \ldots, K_{aux}\}$

$$P(z_{di} = k, x_{di} = k, w_{di} = w, z_{-di}, x_{-di}, w_{-di}, \alpha, \beta, t_d) \propto C_{w, aux}^\Phi + \beta \sum_{x_{-di}} C_{k, aux}^\Theta + \alpha \sum_{z_{-di}} C_{w, K}^{\Phi_{aux}} + \beta_n \sum_{k_{aux}} C_{w, aux}^{\Theta_{aux}} + \alpha_k$$

(5)

Table 2 Symbols and descriptions used in this paper

<table>
<thead>
<tr>
<th>Notations</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>Tags of the corpus</td>
</tr>
<tr>
<td>$t_d$</td>
<td>Tags of the $d$th document</td>
</tr>
<tr>
<td>$T_d$</td>
<td>Number of tags of $d$th document</td>
</tr>
<tr>
<td>$C_{K, T}$</td>
<td>Number of words assigned to tag and topic, $K \times T$ matrix</td>
</tr>
<tr>
<td>$C_{W, K}$</td>
<td>Number of words assigned to topic and word, $W \times K$ matrix</td>
</tr>
<tr>
<td>$D_{train}$</td>
<td>Set of tags and words of training data</td>
</tr>
<tr>
<td>$K$</td>
<td>Number of topics</td>
</tr>
<tr>
<td>$D$</td>
<td>Number of documents</td>
</tr>
<tr>
<td>$N_d$</td>
<td>Number of words in the $d$th document</td>
</tr>
<tr>
<td>$W$</td>
<td>Vocabulary size</td>
</tr>
<tr>
<td>$w_d$</td>
<td>Words in the $d$th document</td>
</tr>
<tr>
<td>$w_{di}$</td>
<td>$i$th word in $d$th document</td>
</tr>
<tr>
<td>$X$</td>
<td>Tag assignment</td>
</tr>
<tr>
<td>$x_{di}$</td>
<td>Tag assignment for $w_{di}$</td>
</tr>
<tr>
<td>$Z$</td>
<td>Topic assignment</td>
</tr>
<tr>
<td>$z_{di}$</td>
<td>Topic assignment for $w_{di}$</td>
</tr>
<tr>
<td>$\Phi$</td>
<td>Probabilities of words given topics, $W \times K$ matrix</td>
</tr>
<tr>
<td>$\Theta$</td>
<td>Probabilities of topics given tag, $K \times T$ matrix</td>
</tr>
<tr>
<td>$\phi_k$</td>
<td>Probabilities of words given topic $k$, $W$-dimensional vector</td>
</tr>
<tr>
<td>$\theta_{k,t}$</td>
<td>Probabilities of topics given tag $t$, $W$-dimensional vector</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Dirichlet prior</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Dirichlet prior</td>
</tr>
<tr>
<td>$\rho_{tar/aux}$</td>
<td>Parameter of target or auxiliary</td>
</tr>
</tbody>
</table>

3.4 Services clustering with hidden topics

After learning and inferring the parameters for new documents, $[topics] \times [services]$ matrix $\Theta'$ which is distribution over topics for each service document is achieved and we can represent each web service $WS_d$ by $\Theta'$ as:

$$WS_d = \left[ \sum_{t_1}^{\theta_{aux, t_1}} \ldots \sum_{t_1}^{\theta_{aux, t_1}} \sum_{k_{aux}}^{\theta_{aux, k}} \ldots \sum_{k_{aux}}^{\theta_{aux, k}} \right]$$

(6)

$$\sum_{x_{di}}^{\theta_{i, x}} x \in \{aux, tar\} \text{ is used to normalise the scale of each feature to reduce the importance of some topics that are overly general. Such topics tend to occur in most documents, but they lack the discriminative power.}$$

After having the hidden topical representations for each web service, the traditional clustering (in our experiments, $K-means$ is adopted) approaches can be utilised on them. Since these new features of web services have already involved the knowledge from the auxiliary data, clustering on the new representation may achieve better results, as we will demonstrate in the experiments.

4 Experiment and evaluation

4.1 Experimental preparation

To evaluate the performance of the proposed approach, we select the top five categories in PWeb that have the largest numbers of services as our test bed. We firstly crawled all the 2,761 web services of these 5 categories from PWeb. These 2,761 services have 9,791 tags. Since many tags may exist in multiple services, totally 975 distinct tags are included in these services. The numbers of services and tags in these categories are shown in Table 3. Since some tags may not have descriptions in Wikipedia, we totally get 1,065 descriptions of all 975 tags from Wikipedia. The average length of each short text description is 68, that is, each description contains about 68 words, while the average length of each long text description is 845.

Table 3 The statistical information about the dataset

<table>
<thead>
<tr>
<th>Service category</th>
<th>Number of services</th>
<th>Number of tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tools</td>
<td>761</td>
<td>2,879</td>
</tr>
<tr>
<td>Internet</td>
<td>600</td>
<td>1,943</td>
</tr>
<tr>
<td>Social</td>
<td>491</td>
<td>1,660</td>
</tr>
<tr>
<td>Financial</td>
<td>465</td>
<td>1,662</td>
</tr>
<tr>
<td>Enterprise</td>
<td>444</td>
<td>1,647</td>
</tr>
</tbody>
</table>

4.2 Evaluation metrics

In our experiments, we use two criteria to evaluate the performance of our approach, namely entropy and purity which are two commonly used parameters for the evaluation of document clustering (Zhao et al., 2005). Entropy measures how the various semantic classes are distributed within each cluster. Purity is a simple and transparent evaluation measure, which can help us understand the quality of the clustering result directly. The larger the purity is, the better the performance of the clustering is. Conversely, the smaller entropy means better performance of the clustering.

Given a particular cluster $c_i$ of size $n_i$, the entropy of this cluster is defined as:

$$E(c_i) = - \frac{1}{\log(q)} \sum_{j=1}^{n_i} \sum_{j=1}^{n_i} \log \left( \frac{n_j}{n_i} \right)$$

(7)
where \( q \) is the number of classes and \( n_i \) is the number of web services of the \( i \)th class that were assigned to the \( i \)th cluster. The entropy of the entire clustering is then the sum of the entropies of individual clusters weighted according to the cluster sizes:

\[
\text{Entropy} = \sum_{i=1}^{k} \frac{n_i}{n} E(c_i) \tag{8}
\]

Using the same mathematical definitions, the purity of a cluster and the purity of the entire clustering are defined as:

\[
P(c_i) = \frac{1}{n_i} \times \max_j \left( n'_j \right), \quad \text{Purity} = \sum_{i=1}^{k} \frac{n_i}{n} P(c_i) \tag{9}
\]

### 4.3 Performance of clustering approaches

To assess the performance of our method, we compare the proposed DT-LDA with other four service clustering approaches: KMeans (Jin et al., 2011), agglomerative (ALT) (Platzer et al., 2009), LDA-L and LDA-S. In Jin et al. (2011), web services are clustered according to the WSDL similarity between web services by using KMeans. ALT is a bottom-up clustering method where clusters have sub-clusters, which in turn may also have sub-clusters. As discussed in Section 3.2, each web service has two descriptions: long text description and short text descriptions. LDA-L learns an ATM from long text descriptions and then clusters the services, while LDA-S learns an ATM from short text descriptions of services and then clusters the services.

For LDA-L, LDA-S and DT-LDA model, we first build the topical representations of the web services following the procedures described in Section 3.3, and then use KMeans to cluster these web services on the hidden topics. All experiments with uncertainties are conducted 100 times and the box and whisker plot is drawn to illustrate the results as shown in Figure 4. We tune each algorithm to its best parameter setting and the parameters influence will be discussed in Section 4.4.

According to these experiments, we have several observations. Firstly, the purity and entropy performance of DT-LDA outperforms other approaches as shown in Figure 4, which demonstrates the effectiveness of the proposed approach. Secondly, three approaches including KMeans, ALT and LDA-S have poor performances in clustering short text descriptions. This quite reveals that the traditional methods suffer from high dimensionality and sparseness of representation in the short text. Thirdly, it is interesting that the LDA-L model, which ignores all the short texts, performs better than LDA-S. We believe that this is because learning topic models on short texts is inherently much more difficult and long text description constructed by using tag description may have more dense topical representation. Finally, compared with LDA-S, the proposed approach has significant purity and entropy improvement by 32.1% and 14.7%. This strongly supports the conclusion that leveraging auxiliary data can significantly improve the clustering performance.

### 4.4 Influence of parameters and auxiliary data

#### 4.4.1 Hyperparameters

Traditionally, hyper-parameters in a topic model are often set as \( \alpha = 50/K \) and \( \beta = 0.01 \) (Rosen-Zvi et al., 2004, 2010). In our work, hyper-parameter \( \alpha \) is applied to control selecting topics. \( \alpha_{\text{aux}}^{\text{tar}}, \alpha_{\text{aux}}^{\text{tar}}, \alpha_{\text{aux}}^{\text{aux}}, \alpha_{\text{aux}}^{\text{aux}} \) are need to be carefully set. We set and vary other three in \( \{0.05, 0.1, 50/K\} \).

#### 4.4.2 Topic numbers

For DT-LDA, \( K = K_{\text{aux}} + K_{\text{tar}} \) illustrates the overall complexity of the model \( K_{\text{aux}} \) and \( K_{\text{tar}} \) indicate the topic numbers of auxiliary information and target information, respectively.

Figure 5(a) shows the purity performance of the four approaches on different topic numbers. From these results, we can discover that the performances are close when those topic models with small-sized topics, but they perform much different when the number of topics is large. Figure 5(a) also shows that LDA-L, LDA-S and DT-LDA achieve best performance at \( (60, 100) \), respectively. In conclusion, DT-LDA outperforms the other methods on different topic numbers when the topic number is large.

Figure 5(b) shows the influence of \( K_{\text{aux}} \) for DT-LDA. \( K_{\text{aux}} \) is applied to control different degrees of topics shared between the target and auxiliary data. From the graph, we can see that a proper \( K_{\text{aux}} \) is needed to achieve the best performance when \( K \) is fixed. For instance, when \( K \) equals to 100 and the \( K_{\text{aux}}/K \) equals to 0.6, that is, \( K_{\text{aux}} = 60 \), the DT-LDA achieves the best performance.

#### 4.4.3 Influence of auxiliary data

The amount of auxiliary data involved in the learning process influences the result. Figure 6 shows an influence by varying the amount of the long text descriptions of web services used for training from 10% to 100% of all auxiliary data. As shown in Figure 6, when only a little auxiliary data
is used, the performances of LDA-S and DT-LDA have slightly difference. When the ratio of auxiliary data used for training grows bigger, the performance of DT-LDA is much better than LDA-S. This also shows that transferring auxiliary knowledge can help improve the clustering performance.

4.4.4 Influence of irrelevant data

We also conducted experiments to assess the influence of irrelevant data. In this experiment, the auxiliary data is mixed with some documents that are randomly chosen from a general knowledge base that contains irrelevant data. These documents are not strongly related to the target data and can be considered as irrelevant data in the auxiliary data. We control the amount of irrelevant data by inserting different number of irrelevant documents into the auxiliary data. Figure 7 presents the results obtained from the preliminary analysis of influence of irrelevant data. As can be seen from the figure, the proportion of irrelevant data blending in the auxiliary data affects the clustering performance of DT-LDA. When the proportion of irrelevant data is less than 0.4, the purity of DT-LDA is slightly different from the DT-LDA without irrelevant data. However, the performance of clustering falls down when the proportion is more than 0.5. The results show that irrelevant data can highly affect the performance of the clustering. Therefore, we should spend more human effort to do more manually adjusting of these priors.

5 Conclusions

In this paper, we presented a novel model DT-LDA which transfers the topical knowledge from the auxiliary long texts to help with the unsupervised clustering on target short texts. The DT-LDA jointly learns a set of target topics from the short text service description and another set of topics on auxiliary long texts corresponding to the tags. The experimental result shows that DT-LDA can outperform many existing approaches. This helps validate that, by leveraging the auxiliary knowledge, the clustering quality on short text can be improved.

DT-LDA uses asymmetric Dirichlet prior to control the relative importance of target versus auxiliary topics when generating a document, which may suffer from inappropriate parameter selection. Therefore, in the future, we plan to consider topic models which can choose topics automatically between the two types of topics when
generating the document. Further research also includes how to support effective web service clustering even when the tags of a service are very few.

Acknowledgements

The work is supported by the National Basic Research Program of China under grant no. 2014CB340404, the National Natural Science Foundation of China under grant nos. 61202031, 61373037, the State Key Laboratory of Software Engineering Foundation under the grant no. SKLSE 2014-10-07.

References


