SeCo-LDA: Mining Service Co-occurrence Topics for Recommendation

Zhenfeng Gao, Yushun Fan*, Cheng Wu
Tsinghua National Laboratory for Information Science and Technology
Department of Automation
Tsinghua University
Beijing 100084, China
gzf13@mails.tsinghua.edu.cn
fanyus/wuc@tsinghua.edu.cn

Wei Tan
IBM Thomas J. Watson Research Center
Yorktown Heights, NY 10598, USA
wtan@us.ibm.com

Jia Zhang
Carnegie Mellon University
Silicon Valley
jia.zhang@sv.cmu.edu

Yayu Ni, Bing Bai, Shuhui Chen
Tsinghua National Laboratory for Information Science and Technology
Department of Automation
Tsinghua University
Beijing 100084, China
nyy07/bb13/chensh13@mails.tsinghua.edu.cn

***Abstract*—Service ecosystem consists of all kinds of services, and some of them may be composed by developers to create new mashups. Existing work on service recommendation and composition mine either frequent patterns from mashup-service usage records, or latent topics from service metadata. In this paper, we propose Service Co-occurrence LDA (SeCo-LDA), a novel approach that mines latent topic models over service co-occurrence patterns. The key idea is to treat each service as a document, and its bag of co-occurring services as the bag of words in that document. Using this model, we can analyze such *service co-occurrence documents* with a probabilistic topic model. We show how to derive service co-occurrence topics, and then validate our model on the real-world ProgrammableWeb.com dataset. We illustrate that SeCo-LDA can discover meaningful latent service composition patterns including their temporal strength and services' impacts, which conventional Apriori can not reveal. Comparing with Apriori, content matching based on service description and LDA directly using mashup-service usage records, we have demonstrated that SeCo-LDA can recommend service composition more effectively, 5% better in terms of MAP than the baseline approach.**

Keywords-topic model; Service Co-occurrence LDA; service composition; service composition recommendation

I. INTRODUCTION

With the wide adoption of Service-Oriented Architecture (SOA) and Cloud Computing, the quantity of published web services on the Internet has been rapidly growing [1]. By reusing existing services (i.e., APIs), software developers are able to quickly create service compositions (i.e., mashups) to meet complex function needs and offer additional business values [2]. However, the overwhelming amount of services makes it challenging for mashup developers to manually select proper candidates to meet specific functional requirements. Meanwhile, it's difficult for developers to identify the current trend of mashup creation within certain domains. Such challenges call for new techniques to help developers gain a better understanding of latent composition patterns of services in service ecosystem, and to help select services intelligently and automatically.

In the research of automatic service composition, service association learning is a commonly-used technique. Many existing models [3, 4, 5, 6] use the Apriori algorithm, leveraging service usage records, to mine frequent patterns and instruct mashup creation. Although association rules can be mined, we could not tell what the connotative meaning of a specific service association pattern is and what exactly the impact of each containing service is. Furthermore, the popularity of these patterns might change over time, while conventional approaches could not reveal such time-dependent property.

Few have considered mining frequent patterns using latent models such as the topic model. In this paper, we intend to identify such latent patterns as "service co-occurrence topics," which differ from semantic or functional topics in [7, 8] but do make sense. During a long-term collaboration and competition in service ecosystem, service co-occurrence topics may emerge gradually. Services composited in one mashup are more likely to belong to a same service co-occurrence topic. For example, location-aware APIs (e.g., *Google Maps*) and social network APIs (e.g., *Facebook*) are often combined to realize location-aware social network mashups. Such a fact indicates that there do exist latent composition patterns between them. Therefore, service co-occurrence topics are not equal to functional topics, which are built on services' word descriptions. Service co-occurrence topics reveal the latent patterns about service compositing and are represented as the distributions over services in the service ecosystem, instead of the distribution over words in semantic functional topics.

To our best knowledge, no algorithm exists to mine the latent service co-occurrence topics and recommend service composition. In this paper, we propose a novel model "Service Co-occurrence LDA" (SeCo-LDA) by extending conventional Latent Dirichlet Allocation (LDA) to identify latent composition patterns of services in service ecosystem. We create "service co-occurrences" with the mashup-service usage records to solve the sparsity problem caused by directly using mashup-service usage records. In our model, each service is treated as a document with co-occurring

* Corresponding Author

978-1-5090-2675-3/16 $31.00 © 2016 IEEE
DOI 10.1109/ICWS.2016.13

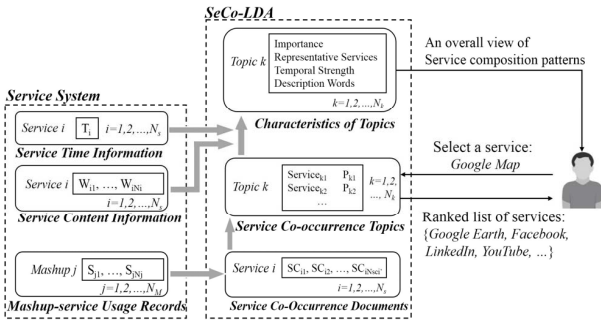

Figure 1. Framework of Service Recommendation based on SeCo-LDA. Leveraging mashup-service usage records, we construct *service co-occurrence documents*, based on which we apply SeCo-LDA to reveal service co-occurrence topics. Developer selects the first service, the system would return a ranked list of recommended services. Leveraging services' time and content information, characteristics of topics are calculated, giving an overall view of service composition patterns. Definitions about notations are presented in Section II detailly.

services as words, that is, we use co-occurring services to describe a specific service. For example, if service s_1 has co-occurred with s_2 once and with s_3 for 2 times among all the mashups, the *service co-occurrence document* of s_1 would be $d_1 = \{s_2, s_3, s_3\}$. We rely more on the co-occurrence information, which is a good indicator of service co-occurrence topics and more effective than relying on text description. Every mashup created by developers is a real API on the website, making it much less noisy than text. We directly model the generation of "service co-occurrences," and then make recommendation for service composition. The framework described above is shown in Figure 1. The main contributions of this paper are summarized as follows:

1) We propose a novel method Service Co-occurrence LDA, whose foundation is the construction of *service co-occurrence documents*. Our model is capable of discovering latent service co-occurrence topics in service ecosystem and extracting services' "impact" on each topic;

2) We discover the characteristics of service co-occurrence topics, and illustrate the concrete meaning of different service association patterns along with their temporal strength information;

3) Comprehensive experiments over real-world dataset ProgrammableWeb.com show that the topics revealed by our model is of high quality with the overall higher MAP value than baselines when recommending related services for a selected service, 5% better in terms of MAP than LDA directly using Mashup-service Usage Records.

The rest of this paper is organized as follows. Section II provides the definition of the problems and our SeCo-LDA model. Parameter learning and the calculation of the characteristics of service co-occurrence topics is illustrated in Section III. Section IV shows experimental results on a real-world dataset from ProgrammableWeb.com, including discovering characteristics of topics and recommending for service composition. Section V summarizes the related work and then Section VI concludes the paper.

II. THE SECO-LDA TOPIC MODEL

In contrast to conventional content-based topic models, our model treats each service in the ecosystem as a document, and its bag of co-occurring services as the bag of words in that document. We model these reconstructed *service co-occurrence documents* with a probabilistic generative model, which is analogous to the application of a probabilistic topic model to discover topics in text documents, but with the core difference that the discovered topics with our model would be characterized by a multinomial distribution over services, rather than over words, and they could reveal the latent composition patterns of services. So we can discover services' impact on a topic according to the topic's distribution over services, which can be easily revealed by our model, while it cannot be directly revealed by conventional content-based topic models or association learning algorithms.

In this section, we first describe the definition about background, then describe the construction of *service co-occurrence documents* and the topic model based on them.

A. Background

Definition 1: Topology of service ecosystem. The topology of a service ecosystem containing mashup-service citation records is modeled as an undirected graph $G = (M \bigcup S, E)$ in which: $M = \{m_1, m_2, ..., m_{N_M}\}$ is the set of mashups and $S = \{s_1, s_2, ..., s_{N_S}\}$ is the set of services; N_M is the number of mashups and N_S is the number of services; $E \subseteq M \times S$ is the historical usage records between mashups and services, i.e., if a mashup invokes a service, there exists a citation between them.

Definition 2: Service Content. In the service ecosystem, every service $s \in S$ comprises a collection of words $SW(s) = \{w_1, w_2, ..., w_{n_s}\}$ to describe its functions.

Definition 3: Service Co-occurrence Topics. Some certain services might have been composited and cooperated repeatedly to realize complex function when creating mashups, revealing some kind of latent topics about service composition. Different from functional topics revealed from service content, service co-occurrence topics describe latent composition patterns between topics and are represented by the distribution over services in the system. For example, *topic k* is described by $\{\phi_{k,j}, j = 1, ..., N_S\}$, in which $\phi_{k,j}$ describes the impact of *service j* on *topic k* when making service composition, and $\sum_j \phi_{k,j} = 1$. The characteristics of service co-occurrence topics contain:

1) Topic Importance*:* different service co-occurrence topics may reveal different composition patterns and have different importance in the service ecosystem. The most important topics illustrate the popular trend of service composition.

2) Topic Representative Services*:* services that are the most popular in a service co-occurrence topic, from which we could tell the meaning of composition patterns.

*3) **Topic Temporal Strength**:* an distribution over time, reflecting a service co-occurrence topic's lifecycle and revealing the change of popularity of certain composition patterns.

*4) **Topic Description Words**:* words that could describe a topic funcitionally.

Problem Definition: Recommendation for Service Composition. In this paper, we consider such a situation as in [3]: assuming a mashup-developer selects the first API, now he needs to find other APIs to create a new mashup. At some time, he might not know exactly what kind of mashup he wants to make, and just hope to find other related services to make significant compositions. Here we refer to the selected service as s_l, and the result of recommendation for its service composition is represented as a ranked list $R_l = \{s_{l1}, s_{l2}, ...\}$.

B. Reconstruction of Service Co-occurrence Documents

Definition 4: Service Co-occurrences. For each service $s_i \in S$, if service $s_j \in S$ is composited together with s_i by mashups in the service ecosystem for $c_{i,j}$ times in total, we define the co-occurrence factor $c(i, j) = c_{i,j}$, which means s_i has co-occurred with s_j for $c_{i,j}$ times. Co-occurrence with the service itself is not counted. Thus, according to our definition, we have $c_{i,j} = c_{j,i}$ and $c_{i,i} = 0$.

Definition 5: Service Co-occurrence Documents. For each service $s_i \in S$, using its co-occurred services as word tokens, we represent s_i as a "bag of service co-occurrences" $d_i = \{\#(sc_j) = c_{i,j} \mid j \in S\}$ in which: sc_j represents service co-occurrence of s_j, and $\#(sc_j) = c_{i,j}$ means service co-occurrence on s_j appears $c_{i,j}$ times in the description document of s_i. All service co-occurrence documents are referred to as D.

C. Service Co-occurrence LDA

Based on the service occurrence documents, we extend the basic Latent Dirichlet Allocation (LDA) [9] model and name it "Service Co-occurrence LDA" (SeCo-LDA). The key intuition is that if some services are composited by a mashup, they are more likely to belong to a same service co-occurrence topic, indicating latent composition patterns between them.

So we have a corpus with N_S service co-occurrence documents and K composition topics expressed over N_S unique services in the service co-occurrence vocabulary. Let $z \in [1, K]$ be the topic indicator variable, the popularity of topics in the corpus (i.e., $P(z)$) can be represented by a K-dimensional multinomial distribution $\boldsymbol{\theta} = \{\theta_k\}_{k=1}^{K}$ with $\theta_k = P(z = k)$ and $\sum_{k=1}^{K} \theta_k = 1$. The service-citation distribution for topics (i.e., $P(s \mid z)$) can by represented by a $N_S \times K$ matrix $\boldsymbol{\Phi}$. In $\boldsymbol{\Phi}$, each column ϕ_k is a N_S-dimensional multinomial distribution with $\phi_{k,s} = P(s \mid z = k)$ and $\sum_{s=1}^{N_S} \phi_k(s) = 1$.

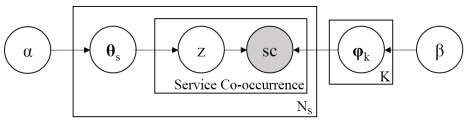

Figure 2. Graphic Model for Service Co-occurrence LDA

Following the convention of LDA [9], in Service Co-occurrence LDA, we use symmetric Dirichlet priors for $\boldsymbol{\theta}$ and ϕ_k with hyper-parameters α and β, respectively. Its graphical model is shown in Figure 2. The generative process of Service Co-occurrence LDA based on service-co-occurrence documents could be described as follows.

For each *service i*,

1. Draw topic proportions $\theta_i \sim Dirichlet(\alpha)$.
2. For each co-occurring service sc_{in} of *service i*,
   a) Draw topic assignment $z_{in} \sim Multi(\theta_i)$.
   b) Draw co-occurring service $sc_{in} \sim Multi(\phi_{z_{in}})$.

III. Learn the SeCo-LDA Model

In this section, we introduce the parameter learning of Service Co-occurrence LDA, and provide algorithm to extract characteristics of service co-occurrence topics.

A. Parameter Learning

Like [9], we use the collapsed Gibbs sampling to make inferences with the service co-occurrence model. The sampling is initialized by assigning random topic labels {z} and then updates them iteratively until reaching the setting time of iteration. In particular, for the *t*-th co-occurring service s_j in s_i's document, the topic assignment is updated according to:

$$\begin{aligned} &\Pr(z = k \mid sc_{i,t} = s_j, Z_{\neg(i,t)}, D_{\neg(i,t)}) \\ &\propto (\alpha_k + \#_{\neg(i,t)}(z = k, d = i)) \times \\ &\frac{\beta_j + \#_{\neg(i,t)}(z = k, sc = s_j)}{\sum_j \beta_j + \#_{\neg(i,t)}(z = k, sc = s_j)} \end{aligned} \quad (1)$$

After the burn-in stage, the sampling converges to the true posterior distribution. Posterior expectation of $\theta_{i,k}$ and $\phi_{k,j}$ is describes as:

$$\theta_{i,k} = \frac{\#(d = i, z = k) + \alpha_k}{\sum_k \#(d = i, z = k) + \alpha_k} \quad (2)$$

$$\phi_{k,j} = \frac{\#(z = k, sc = j) + \beta_j}{\sum_k \#(z = k, sc = j) + \beta_j} \quad (3)$$

Intuitively, $\boldsymbol{\theta}$ describes the topic distribution of services in the system (*service-topic* distribution), while $\boldsymbol{\Phi}$ indicates the distribution on co-occurring services of topics (*topic-service* distribution). Both of them will help us to calculate the topics' characteristics we mentioned before.

Further, the empirical posterior distribution of topics, which reflects the importance or popularity of topics in the service system, is given by:

$$\Pr(z = k \mid D) = \frac{\#(z = k)}{\sum_k \#(z = k)} \tag{4}$$

B. Discovery of Service Co-occurrence Topics

From results obtained from formulae (2) – (4), along with the information of service content, we can discover knowledge about individual service co-occurrence topics, which can be used to help people to have an overall view of composition patterns in the service ecosystem.

1) Topic Importance

Given by (4), the distribution $\{\Pr(z = k)\}$ indicates the probability that services in one topic getting composited. We could refer to the value of the distribution as the importance of different topics in the service ecosystem. Topics with higher importance would be more likely to be chosen by developers when creating mashups, and vice versa. The top topics reflect the major composition theme or trend in composing services into a mashup.

2) Topic Representative Services

When creating mashups, developers usually prefer the most popular services of certain service co-occurrence topics to realize some significant composition. The more popular a service is in the topic, the more representative it is. The ranking value of services based on *topic-service* distribution $\{\phi_{k,j}\}$ reflects services' popularity, or impact, on the topic z_k. Specifically, we could choose the top 80% services in probability as representative ones of the topic.

Directly ranking of mashup-service usage records to find representative services are usually not accurate because there may be situation that in some areas developers tend to composite more services than developers from other areas. With the help of *topic-service* distribution, we can solve this problem and get more significant representative services.

3) Topic Temporal Strength

A service co-occurrence topic's lifecycle usually includes *beginning period*, *boom period* and *fading period*, roughly paralleling the shape of a normal distribution. During *beginning period*, a few services of the service co-occurrence topic appear and developers begin to composite these services to create mashups. During *boom period*, the most representative services of the topic emerge and a large number of mashups containing these services are created. During *fading period*, developers' interest on this topic gradually decrease.

Different topics may have different lifecycles. Some may be long lasting while some topics' *booming period* may be very short and fade quickly. Some topics may have several *boom periods* while some may have only one.

To sufficiently describe topics' lifecycle, topic temporal distribution is studied. For a topic z_k in the service ecosystem, we use its distribution over services to describe its temporal characteristics. The proportion of accumulated probability for published services until time t forms the cumulative distribution function (CDF) of topic z_k:

$$\begin{aligned}\Pr(time \le t \mid z = k) &= \sum_{s,time(s)\le t} \Pr(service_s \mid z = k) \\ &= \sum_{s,time(s)\le t} \phi_{k,s}\end{aligned} \tag{5}$$

where $time(s)$ means the published time of *service s*.

In our case, the unit of time is *day*, which is discrete. The probability mass function (PMF) for temporal distribution of z_k is:

$$\Pr(time_t \mid z = k) = \sum_{s,time(s)=t} \phi_{k,s} \tag{6}$$

With these information, the expectation time of topic z_k, with which we could distinguish new topics with old ones, can be calculated as follows:

$$E_{s|z=k}[time(s)] = \sum_s time(service_s) \cdot \phi_{k,s} \tag{7}$$

4) Topic Description Words

From the original results of Service Co-occurrence LDA, only the *topic-service* distribution is revealed, without an intuitive description of individual topics with words. However, it is desirable to extract words to describe a topic properly or summarize it with a few words [10]. Along with the information about representative services, it would enable developers to have a brief and intuitive idea about this service co-occurrence topic. With the help of service content and *topic-service* distribution, we can extract words with high *occurrence expectation* for a topic. The *occurrence expectation* of word *w* in topic z_k is calculated as:

$$\begin{aligned}&E_{z=k}(\#word = w) \\ &= \sum_s \#(word = w \mid service_s) \Pr(service_s \mid z = k) \\ &= \sum_s \#(word = w \mid service_s) \cdot \phi_{k,s}\end{aligned} \tag{8}$$

We will show in the next section that words with high *occurrence expectation* are intuitive and useful for developers to recognize and distinguish different topics.

C. Recommendation for Service Composition

With the results above, for a selected service s_l in the ecosystem, we could calculate its expected co-occurrences with another service s_k:

$$\begin{aligned}&c^*(l,k) \\ &= \sum_z \Pr(sc = k \mid topic = z) \cdot \Pr(topic = z \mid s_l) \\ &= \sum_z \phi_{z,k} \cdot \theta_{l,z}\end{aligned} \tag{9}$$

Services with high expected co-occurrences should be recommended to developers. Finally, we give the results of recommendation for service composition as a service list $R_l = \{s_{l1}, s_{l2}, ... \mid c^*(l, s_{l1}) \ge c^*(l, s_{l2}) \ge ...\}$.

IV. EXPERIMENTS

In this section, firstly, we introduce the Programmable-Web.com dataset on which we apply SeCo-LDA and conduct related experiments. Then, we discuss the results of discovering individual service co-occurrence topics in the system. At last, we show that SeCo-LDA over-performs baselines when making service composition recommendation.

A. Dataset

ProgrammableWeb has been accumulating a variety of services and their mashups since established in 2005 [11,12]. To evaluate our methodology, we crawled the information of all service APIs and mashups from its inception (September 2005) to November 2015, including their descriptions and mashup-service usage records. Based on the dataset, we constructed "service co-occurrence documents," removing mashups that contain less than two services and services that have no co-occurrence with any other service. Detail information of the dataset received is presented in Table I.

TABLE I. DATASET OF PROGRAMMMABLEWEB.COM

Total # of services	12,711
Total # of services having co-occurrence with other services	975
Total # of mashups	6,239
Total # of mashups containing more than one service	2,513
Total # of vocabulary	21,328

In our experiments, we perform the SeCo-LDA with 35 latent topics, setting $\alpha = 50/T$ and $\beta = 0.01$.

B. Results on Service Co-occurrence Topics

1) Recognizing Important Topics

Calculated by (4), topics with higher $\Pr(z = k)$ are considered more important in service ecosystem, and would be more likely to be selected by developers when composing services.

The top five important topics are Topics 5(4.1%), 6(3.9%), 35(3.8%), 19(3.6%) and 4(3.5%). We could roughly identify that the most popular service composition topics are those about location-aware service composition, online multi-media community and social network.

TABLE II. TOP 5 REPRESENTATIVE SERVICES FOR TOPICS 6,19 & 35

Topic 6		**Topic 19**		**Topic 35**	
$\phi_{k,j}$	***Service Name***	$\phi_{k,j}$	***Service Name***	$\phi_{k,j}$	***Service Name***
0.72705	YouTube	0.13145	Google Geocoding	0.28517	Last.fm
0.20321	Amazon Product Advertising	0.08644	Google Earth	0.09795	Amazon Product Advertising
0.01666	Yahoo Query Language	0.08374	Google App Engine	0.09708	MusicBrainz
0.01333	WebThumb	0.08194	Panoramio	0.05201	Spotify Echo Nest
0.00667	Mobypicture	0.06213	Google Visualization	0.05115	LyricWiki
Mashup Example	Find Best Three	***Mashup Example***	youbeQ - Maps with Life	***Mashup Example***	Musikki

2) Finding Representative Services

The top 5 representative services for Topics 6, 19 and 35, which are topics of great importance in service ecosystem, are identified in Table II. The *topic-service* probability and mashup examples are also listed in the table. Obviously, the representative services are sufficient to represent the topics and provide an overall view about the service co-occurrence topics' major functions.

In some topics, several representative services may occupy a dominant position. For example, in Topic 6, service ***YouTube*** and ***Amazon Product Advertising*** have more than 93% impact on the topic among all the services. We can regard such topics as *core-service-oriented* service co-occurrence topics. Topic 6 is about ***YouTube***-centered video-based product advertising, in which ***Amazon Product Advertising*** allows developers to leverage the Amazon Product Discovery features that Amazon uses to advertise products and power their own business. An example mashup of this topic is ***Find Best Three***, which is a buying suggestion and amazon product comparison tool presenting the three best goods with the help of video reviews. In other topics, however, top representative services are not so dominant, such as Topics 19 and 35. Topic 19 is a service composition group based on all kinds of Google services. An example mashup of this topic is ***youbeQ-Maps with Life***, which allows people to explore the Earth, meet new people, discover new places and share information. Topic 35 is a music service composition group, in which ***Last.fm*** provides world's largest online music catalogue, ***Amazon Product Advertising*** allows developers to leverage the Amazon Product Discovery features, ***MusicBrainz*** contains a database with a huge amount of music metadata, ***Spotify Echo Nest*** allows developers to analyze tracks and to add rich artist and song metadata to their applications, and ***LyricWiki*** provides users with lyrics. An example mashup of this topic is ***Musikki***, a music search engine that gives users all the information in a single page with a single search.

3) Calculating Topic's Temporal Distribution over Services

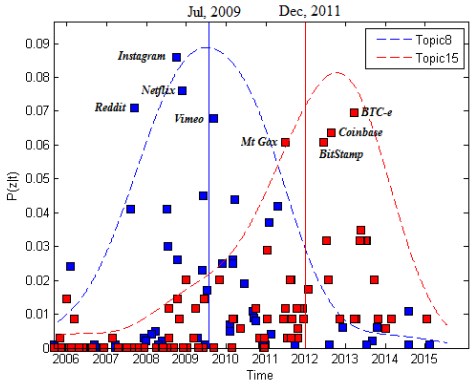

Figure 3. Distribution Over Service for Topics 8 & 15. Blue stands for Topic 8, and red for Topic 15. Squares symbolize representative services, and the most 4 representative services' names are provided. The vertical lines show the expectation time of each topic. Dotted lines describe the fitted curves of topics' temporal strength.

The services' published times range from September 4, 2005 to July 14, 2015 in our dataset. We take a *day* as the unit when calculating topics' temporal distribution over services.

As shown in Figure 3, we present the topic temporal distribution over services of Topics 8 and 15. Topic 8, which contains representative services like ***Instagram***, ***Netflix***, ***Reddit*** and ***Vimeo***, is about social multi-media sharing and discussing. While Topic 15, whose representative services are ***BTC-e***, ***Coinbase***, ***BitStamp*** and ***Mt Gox***, is service composition topic about the trade of BitCoin. From the results of Figure. 3, we can identify that Topic 15 is a newly emerged topic around 2012, while Topic 8 is an older one. In fact, most dominant social multi-media services sprang up around 2010, yet BitCoin is firstly proposed by Satoshi Nakamoto in concept in 2009 [13], and more and more people began to trade it and popular trading platform appeared around 2012.

Above all, information provided by topics' distribution over services consists with external empirical knowledge basically.

4) Extracting Topic Description Words

With occurrence expectation calculated by (8), we can extract a collection of words with high occurrence expectation for each topic. We use the Porter stemming algorithm [18] and get the word stems when doing experiments.

TABLE III. DESCRIPTION WORDS FOR TOPICS 29 & 33

Topic	Representative Services	Part of Extracted Description Words
29	LinkedIn, Last.fm, Tumblr	easy available simple text webpage javascript design block triporia hotel search date location expedia movie suggest
33	Yahoo Weather, Blogger, Instapaper	market data property chart search rate content list trend people include sale network turn level fresh category price

Similar topics may have similar description words, but difference usually does exist between them. Results about Topics 29 and 33 are shown in Table 3. From representative services, we could identify that they are similar topics about social life on the Internet, while there are also differences between them. Topic 29 concentrates more on people's daily life on communicating with others, sharing multi-media resources ("text," "webpage," "movie") and travelling places ("triporia," "hotel," "expedia"), and information is usually presented with short text ("easy," "simple"). However, Topic 33 emphasizes using a blog essay ("list," "category") to record something ("content," "list") or discuss about some problems ("property," "trend"). With representative services and description words together, we could have an overall view about the concrete meaning of a service co-occurrence topic, and describe it based on the extracted words.

C. Comparison Results of Service Composition

To demonstrate that topics revealed by our model reflect the composition relationship between services more exactly, we compare the results of recommendation for single service composition using our model and baselines. For each service, we recommend the most related *N* services for it, and compare the result with the original relationship revealed from mashup-service usage information.

1) Baselines

Baseline 1: Apriori Algorithm. (AA)

In this approach [5], each mashup is represented as the union of annotated tags of its component services. Apriori [14] mines positive rules of tags from the transactions of mashups. Then the probability of composition of any two services s_1 and s_2 is estimated as:

$$F(s_1,s_2)=\sum_{r\in R(s_1,s_2)}\beta\cdot suppprort(r)+(1-\beta)\cdot confidence(r) \quad (10)$$

where $R(s_1,s_2)$ denotes the rules satisfied by $\langle s_1,s_2\rangle$, and β is a weighting coefficient. For a selected service s_l, its recommended service list is:

$$RL_{AA}=\{s_{l1},s_{l2},...\mid F(s_l,s_{l1})\geq F(s_l,s_{l2})\geq ...\}.$$

Baseline 2: Content Matching based on Service Description. (CMSD)

Similar with [7, 15], based on service descriptions, we apply the LDA model to calculate the semantic similarities between the selected services and others. We run Gibbs sampling to get probability distribution of services over topics $p_2(k\mid s)$ and topics over words $p_2(w\mid k)$.

When a selected service s_l comes up with description as a collection of words $SW_l=\{w_{l1},w_{l2},...,w_{lN_l}\}$, CMSD calculates the semantic similarity between s_l and another service s_i, which is described as $SW_i=\{w_{i1},w_{i2},...,w_{iN_i}\}$, as follows:

$$p_{CM}(s_i\mid s_l)=\sum_{w\in SW_i}\sum_{k=1}^{T}p_2(w\mid k)\cdot p_2(k\mid s_l) \quad (11)$$

Services with higher semantic similarities are preferred to be recommended. So we could get the service recommendation list as:

$$RL_{CMSD}=\{s_{l1},s_{l2},...\mid p_{CM}(s_{l1}\mid s_l)\geq F(s_{l2}\mid s_l)\geq ...\}$$

Baseline 3: LDA directly using Mashup-service Usage Records. (MUR-LDA)

We use mashups as documents composed of services as word tokens and model the generation of mashups' containing services with a probabilistic topic model:

For each *mashup j*,

1. Draw topic proportions $\delta_j\sim Dirichlet(\alpha)$.
2. For each service s_{jn},
   a) Draw topic assignment $y_{jn}\sim Multi(\delta_j)$.
   b) Draw service $s_{jn}\sim Multi(\gamma_{y_{jn}})$.

Using the collapsed Gibbs sampling, we can get the probability distribution of mashups over topics $p_3(k\mid m)$ and topics over services $p_3(s\mid k)$.

With results above, for a selected service s_l, we could calculate its expected co-occurrences with another service s_i:

$$c_3(l,i)=\sum_m\sum_z p_3(s=i\mid k)\cdot p_3(k\mid m)\cdot p(m\mid s_l) \quad (12)$$

where $p(m \mid s_l)$ stands for the probability of mashup m given service s_l. It could be calculated using Bayes theorem. The recommendation list is ranked by the value of $c_3(l,i)$.

2) Evaluation Metric

MAP (Mean Average Precision) [16] is used as evaluation metric in this part, which is a widely used measure in recommendation system:

$$MAP = \frac{1}{|S|}\sum_{i \in S}\frac{1}{N}\sum_{s \in SC_i}\frac{n(s)}{r(s)} \tag{13}$$

where S denotes the set of testing services; N represents the recommended number of services to the given one; SC_i denotes the co-occurring services of *service i*; for each $s \in SC_i$, $r(s)$ refers to the ranking position of s in service-composition recommendation list and $n(s)$ represents the number of co-occurring services in SC_i that rank higher than or equal to s in recommendation list.

MAP is a real number between 0 and 1. The higher MAP indicates a better accuracy of the recommendation method.

3) Results of Recommending for Service Composition

The result of MAP of our Service Co-occurrence LDA and baselines with different number of N is shown in Figure 4.

CMSD only uses the word description of services to reveal latent semantic topics and make recommendation for service composition. Experiments show that the MAP value of CMSD is the lowest among the four methods, indicating that the description of services might not be a good indicator of service composition patterns. The other three methods take the usage history of services into consideration. The Apriori algorithm is the most popular model to draw association rules, which has a higher MAP value than CMSD as shown in Figure 4. Using AA we could get the probability of any popular service association rule, but couldn't know what kind of a specific service association is or the connotative meaning of it. MUR-LDA and SeCo-LDA both use topic model to analyze the association relationship of services in the system. MUR-LDA may come up with the sparsity problem for that most mashups usually contain less than five services. As a result of which, SeCo-LDA gets the highest MAP value with different number of N, approximately 5% better than MUR-LDA.

From the experiments shown above, we can conclude that by constructing *service co-occurrence documents* and modeling the service co-occurrences with a probabilistic topic model, SeCo-LDA could reveal meaningful service co-occurrence topics along with their characteristics, and perform better than the baselines when recommending for service composition.

D. Discussion about Complexity

We can divide the complexity of SeCo-LDA into two parts: constructing the service co-occurrence documents and modeling them with a probabilistic topic model. The complexity of constructing the service co-occurrence documents with mashup-service usage records is bounded by $O(N_{MAXS}^2 \cdot N_M)$, in which N_{MAXS} stands for the maximum number of services in a mashup. The complexity of the Gibbs sampling when inferring parameters is $O(N_G \cdot K \cdot N_S)$, where N_G stands for the number of iterations and K is the number of topics. So the complexity of our approach is bounded by $O(N_{MAXS}^2 \cdot N_M + N_G \cdot K \cdot N_S)$, which is a linear time complexity with N_M and N_S. So when dealing with datasets containing more services and mashups, the approach's time-cost is usually tolerable.

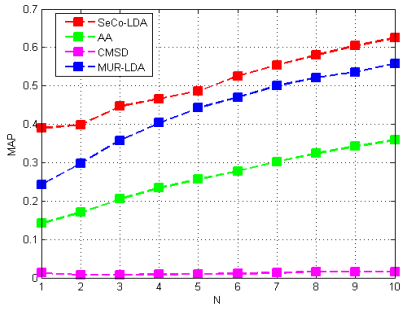

Figure 4. The MAP for SeCo-LDA, AA, CMSD and MUR-LDA with different number of N. AA is the convention method in association rule learning. CMSD and MUR-LDA use different information to apply topic model and make recommendation. β is set to 0.5 in AA. In other three probabilistic topic models, we set the number of topics $K = 35$, $\alpha = 50/T$ and $\beta = 0.01$.

V. Related Work

A. Automatic Service Composition

In the research of automatic service composition, most existing studies about service association learning are on the basis of Apriori algorithm. A global Co-utilization service ranking (gCAR) strategy is proposed in [3] using Apriori to infer association rules to recommend the best services for new mashup. A three-level model of service usage data is defined in [4] to model service usage patterns. Apriori is also used in [5] to find association rules of social tags and predict future mashups based on mined rules. Service logs is taken into consideration in [6] to mine service association rules from service logs with Apriori to build a knowledge repository to instruct mashup creation.

Different from them, we construct a probabilistic topic model based on service usage data to reveal the association rules between services, which we call "service co-occurrence topics." Leveraging the service content and time information, our approach could also calculate the topics' characteristics.

B. Topic Models

In order to calculate topics' temporal strength, we take services' time information into consideration. Existing probabilistic topic modeling considering time information has been thoroughly studied. Topic Over Time (TOT) model [17] explicitly assumes time as generated from topics. Dynamic Topic Model [18] models topics changing over time. Correlated Topic Model [19] calculates topic similarity and draw the topic graph. Through models mentioned above,

we can only compute the text similarity between document and topic, which is quite different form the document "impacts" on a topic.

When considering the issue of web services, [7], [8] and [20] also use topic models to reveal latent topics, but they are built on word co-occurrence basically. [21] proposed a Citation-LDA, which uses paper citation as word tokens apply LDA. But using method in [21] with mashup-service usage records would cause data sparsity problem [22], for most mashups only invoke less than five services.

VI. CONCLUSIONS

In this paper, we propose a novel approach for identifying the latent service composition patterns in service ecosystem. The key idea is to represent each service as a document with its co-occurring services as words. Our work includes three parts: 1) reconstruction of the *service co-occurrence documents*, modeling them with a probabilistic topic model, which we call SeCo-LDA; 2) discovering characteristics of service co-occurrence topics, including topic importance, representative services, temporal strength and description words; and 3) making recommendation of service composition based on the ProgrammableWeb dataset.

In our experiments, we show that with proposed SeCo-LDA, we could discover meaningful latent service composition patterns (service co-occurrence topics) and make recommendation for a selected service. Characteristics of the topics would help people have an overall view about service composition patterns in the system. Comparison with baselines also demonstrate our SeCo-LDA performs 5% better in terms of MAP when recommending for service composition.

Our future work would focus on two aspects: 1) adding other information (like social network information) into our model to improve the quality of revealed service co-occurrence topics; 2) utilizing information about service co-occurrence topics (e.g., *topic-service* distribution, topic temporal strength and topic description words) to design a more effective recommendation framework for automatic mashup creation based on this paper's work.

ACKNOWLEDGMENT

This research has been partially supported by the National Natural Science Foundation of China (No.61174169), and the Specialized Research Fund for the Doctoral Program of Higher Education (No. 20120002110034).

REFERENCES

[1] V. Andrikopoulos, S. Benbernou and M. P. Papazoglou, "On the Evolution of Services," IEEE Transactions on Software Engineering, Vol. 38, pp. 609-628, 2012.

[2] X. Liu, Y. Hui, W. Sun, and H. Liang, "Towards Service Composition based on Mashup," in Proceedings of IEEE World Congress on Services (SERVICES), 2007, pp. 332-339.

[3] B. Tapia, R. Torres, H. Astudillo, and P. Ortega, "Recommending APIs for Mashup Completion Using Association Rules Mined from Real Usage Data," in Proceedings of the 2011 30th International Conference of the Chilean Computer Science Society, 2011, pp. 83-89.

[4] Q.A. Liang, J.-Y. Chuna, S. Miller, and Y. Ouyang, "Service Pattern Discovery of Web Service Mining in Web Service Registry-Repository," in Proceedings of International Conference on e-Business Engineering (ICEBE), 2006, pp. 286-293.

[5] K. Goarany, G. Kulczycki, and M.B. Blake, "Mining Social Tags to Predict Mashup Patterns," in Proceedings of the 2nd International Workshop on Search and Mining User-generated Contents (SMUC), 2010, pp. 71-78.

[6] S. Bayati, A.F. Nejad, S. Kharazmi, and A. Bahreininejad, "Using Association Rule Mining to Improve Semantic Web Services Composition Performance," in Proceedings of The 2nd International Conference on Computer, Control and Communication, 2009, pp. 1-5.

[7] Y. Zhong, Y. Fan, K. Huang, W. Tan, and J. Zhang, "Time-Aware Service Recommendation for Mashup Creation in an Evolving Service Ecosystem," in Proceedings of IEEE International Conference on Web Services (ICWS), 2014, pp. 25-32.

[8] B. Bai, Y. Fan, K. Huang, W. Tan, B. Xia, and S. Chen, "Service Recommendation for Mashup Creation Based on Time-Aware Collaborative Domain Regression," in Proceedings of International Conference on Web Service (ICWS), 2015, pp. 209-216.

[9] D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent Dirichlet allocation, " Journal of Machine Learning Research, Mar. 2003, Vol. 3, pp. 993-1022.

[10] J. Chang, S. Gerrish, C. Wang, J.L. Boyd-graber, and D.M. Blei, "Reading Tea Leaves: How Humans Interpret Topic Models," Advances in Neural Information Processing Systems 32(2009), pp. 288-296.

[11] A.P. Barros and M. Dumas, "The Rise of Web Service Ecosystems," IT professional, 2006, Vol. 8, pp. 31-37.

[12] E. Al-Masri and Q. H. Mahmoud, "Investigating Web Services on the World Wide Web," in Proceedings of the 17th IEEE/ACM International Conference on World Wide Web (WWW), 2008, pp. 795-804.

[13] D. Ron and A. Shamir, "Quantitative Analysis of the Full Bitcoin Transaction Graph," Financial Cryptography and Data Security, Springer Berlin Heidelberg, 2013, pp. 6-24.

[14] R. Agrawal and R. Srikant. "Fast Algorithms for Mining Association Rules in Large Databases," in Proceedings of the 20th Internationa Conference on Very Large Data Bases (VLDB), 1994, pp. 21-30.

[15] C. Li, R. Zhang, J. Huai, X. Guo, and H. Sun, "A Probabilistic Approach for Web Service Discovery," in Proceedings of the IEEE International Conference on Services Computing, 2013, pp. 49-56.

[16] K. Huang, J. Yao, Y. Fan, W. Tan, S. Nepal, Y. Ni, and S. Chen, "Mirror, Mirror, on the Web, Which is the Most Reputable Service of Them All?" in Service-Oriented Computing: Springer, 2013, pp. 343-357.

[17] X. Wang and A. Mccallum, "Topics Over Time: a Non-Markov Continuous-time Model of Topical Trends," in Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2006, pp. 424-433.

[18] D.M. Blei and J.D. Lafferty, "Dynamic Topic Models," in Proceedings of the 23rd International Conference on Machine Learning, 2006, pp. 113-120.

[19] D.M. Blei and J.D. Lafferty, "A Correlated Topic Model of Science," Annals of Applied Statistics 1.1(2007), pp. 17-35.

[20] B. Xia , Y. Fan , W. Tan , K. Huang, J. Zhang, and C. Wu, "Category-aware API Clustering and Distributed Recommendationfor Automatic Mashup Creation," IEEE Transactions on Services Computing, 8(5), 2015, 674-687.

[21] X, Wang, C. Zhai, and D. Roth, "Understanding Evolution of Research Themes: a Probabilistic Generative Model for Citations," in Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2013, pp. 1115-1123.

[22] L. Hong and B. Davison, "Empirical Study of Topic Modeling in Twitter," in Proceedings of the First ACM Workshop on Social Media Analytics, 2010, pp. 80–88.