bi-directional Bayesian probabilistic model based hybrid grained semantic matchmaking for Web service discovery

Shuangyin Li¹ · Haoyu Luo¹ · Gansen Zhao¹ · Mingdong Tang² · Xiao Liu³

Received: 30 August 2021 / Revised: 29 November 2021 / Accepted: 3 January 2022 /

Published online: 17 February 2022

© The Author(s) 2022

Abstract

Web service discovery is a fundamental task in service-oriented architectures which searches for suitable web services based on users' goals and preferences. In this paper, we present a novel service discovery approach that can support user queries with various-size-grained text elements. Compared with existing approaches that only support semantics matchmaking in single texture granularity (either word level or paragraph level), our approach enables the requester to search for services with any type of query content with high performance, including word, phrase, sentence, or paragraph. Specifically, we present an unsupervised Bayesian probabilistic model, bi-Directional Sentence-Word Topic Model (bi-SWTM), to achieve semantic matchmaking between possible textual types of queries (word, phrase, sentence, paragraph) and the texts in web service descriptions, by mapping words and sentences in the same semantic space. The bi-SWTM captures textual semantics of the words and sentences in a probabilistic simplex, which provides a flexible method to build the semantic links from user queries to service descriptions. The novel approach is validated using a collection of comprehensive experiments on ProgrammableWeb data. The results demonstrate that the bi-SWTM outperforms state-of-the-art methods on service discovery and classification. The visualization of the nearest-neighbored queries and descriptions shows the capability of our model on capturing the latent semantics of web services.

Keywords Semantic Web service · Service discovery · Semantic matchmaking · Service classification · Service retrieval

1 Introduction

Along with the rapid advancement of World Wide Web technologies, web services have become the *de facto* standard for consumers to access software systems and resources over

✉ Haoyu Luo
hluo@m.scnu.edu.cn

✉ Gansen Zhao
gzhao@m.scnu.edu.cn

the Internet [4]. In recent years, the quantity of published web services on the Internet has been rapidly growing; they offer developers and users more resources to customize the IT services based on their goals and preferences [5]. For example, over 23,000 web services have been published at ProgrammableWeb by June, 2020, almost increased to five times since 2013. The overwhelming amount of services available makes it a critical challenge for developers to precisely select service candidates that meet specific requirements [5]. To precisely and efficiently search for services from large-scale repositories, many approaches on service discovery have been investigated as well as the related research tracks including service classification [21], clustering [43], selection [32, 40] and recommendation [1, 28] have been proposed.

In general, syntactic-based discovery and semantic-based discovery are the two major patterns that exist for web service discovery. Syntactic approaches mainly discover services by matching the keywords of services with user queries using information retrieval techniques. However, these approaches usually suffer from poor retrieval performance due to the insufficient understanding of the semantic meaning of web service descriptions and user queries. Semantic-aware service discovery approaches search for semantically similar services of queries; they can achieve better performance [41]. Existing semantic-aware service discovery research fall into two categories: logical and non-logical discovery approaches. Logical approaches use an ontology to formalize web service description. These require well-defined ontologies and semantic annotations of services and user queries, which make them difficult to apply [38]. Alternatively, some non-logic semantics-aware service discovery approaches based on latent factor models (namely topic models) have been proposed for better performance [24, 35].

From the perspective of the service requester, he/she may not be aware of all the specialized knowledge that constitutes the domain. The interface exposed by discovery engines is expected to be more intelligent and “user-friendly”. One can search for a service using natural language by entering any possible types of content including words, phrases, or sentences (e.g., a paragraph). For example, to suggest possible services for an engineer or designer, people tend to input some keywords, several sentences, or a long description of a service into a system and expect to obtain the most relevant candidate services with highly interpretable semantics. In such a scenario, the system needs to first extract the available features (e.g., semantics) of the services in the repository. Secondly, the system handles the various types of queries with the same computable features before matchmaking. Thus, a well-designed system has the capacity of matchmaking the services in a computable semantic space for various grained texts.

Existing semantic-based service discovery approaches can only support semantic matchmaking in a single granularity (either at the word-level or paragraph level). Specifically, in the case that searches web services by words, phrases, or one sentence, the service retrieval and discovery process is mostly based on keyword matching. For the query text with several sentences (e.g., a paragraph), some approaches based on a topic model [1, 30] and a deep neural network [39, 43] have been proposed to learn the semantics of a query text and a service description. Topic models, such as Latent Dirichlet Allocation (LDA), have been applied to web service discovery, where the semantics are extracted from rich textual information of services. This type of approach is designed to build the topic space learned by texts of services and aim to match the services with their latent topic features. With the development of topic models, the semantics are captured from various texts, such as sentences [16, 17]. The semantics among the words and the paragraphs are highly interpretable

and have been proven to be effective in many works in the area of information retrieval and text modeling [16, 19].

To achieve effective matching between the user query and service description in the scenario where a query may consist of words, short phrases, or sentences, the semantics need to be jointly measured at both the word and sentence levels. Thus, to specify a mapping between the user query and satisfactory web services, semantic matching for the text contents with various granularities (words, phrases, or sentences) in the query and the sentences in the service description is necessary. However, this remains an open research challenge: how to measure the high-level semantics among different textual elements for service discovery. Achieving this requires the mapping of various-size-grained text elements in a user query and a service description onto the same computable semantic space.

To address the challenge, we propose an unsupervised Bayesian probabilistic model, bi-Directional Sentence-Word Topic Model (bi-SWTM), to achieve semantic matchmaking by jointly learning the semantics of the words and sentences in the same topic space. Topics extracted by the topic model can be used to represent the semantics of the words and sentences. Different from the conventional topic models, the bi-SWTM takes advantage of the two-directional sequences of sentences and the words in each sentence to learn the latent topics. Words and sentences are defined in the same topic space, which builds a bridge to capture the similarities of the words and sentences from the semantic level. Thus, different textual types of user queries (keywords, phrases, and sentences) can be represented in a probabilistic simplex, which provides a flexible approach to extract the high-level semantics of the queries for service matching.

Specifically, the mechanism of modeling the sequences of text from two directions, called bi-directional modeling, is derived from the operation of deep learning for text modeling. Here, the forward and backward sequential information of the text are considered when capturing the correlations of the textual elements. We take advantage of the bi-directional sequences of sentences to extract the semantics among the words (phrases), sentences, and paragraphs. The sentences are the middle-grained textual elements, which are treated as the semantic bridge between words and long texts. With the bi-directional modeling mechanism, the concurrence of words and the coherence of sentences can be effectively captured, which is the key internal factor of the proposed model on semantic extraction.

Particularly, the main contributions of this paper are summarized as follows:

1. We propose a novel bi-Directional Sentence-Word Topic Model (bi-SWTM) to achieve semantic matchmaking between various textual types of queries and service descriptions. This model is capable of discovering the latent semantics of complex queries and service descriptions.
2. The semantics learned from service descriptions are highly interpretable, where the hybrid grained textual queries and service descriptions can be embedded into the same semantic space by the proposed model. It provides an effective way to understand the service better in matching user queries and service descriptions.
3. Comprehensive experiments on ProgrammableWeb demonstrate that the semantics revealed by our model are of high quality. It achieves at most 8% improvements on service classification in comparison with state-of-the-art comparisons, and reaches 0.97 accuracy@5 on service discovery. The visualization of the nearest-neighbored queries and descriptions shows the insight of our model on capturing the latent semantics of services.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 describes the proposed bi-SWTM and the model inference. Section 4 gives the details of web services discovery with bi-SWTM and discussions. Section 5 presents the experiments and evaluations. Section 6 concludes this work.

2 Related works

Web service discovery refers to finding services that satisfy functional requirements specified by a user query. As a fundamental task in Service-Oriented Architecture, various web service discovery approaches have been proposed in the last two decades.

As mentioned above, most existing work on service discovery can be classified into two main categories: syntactic-based and semantic-based approaches. Syntactic-based service discovery approaches are often based on keyword matching. However, the keyword matching fails to understand the semantic meanings of web service descriptions and user queries.

Semantics-aware approaches attempt to overcome the drawback of syntactic approaches by searching for semantically similar services for queries. Some of the semantic-aware approaches use an ontology to formalize a web service description [7]. Specifically, web services are described by a specific semantic tagged language, e.g., SAWSDL (Semantic Annotations for Web services description language) and OWL-S (Web Ontology Language for Services). The approaches based on a topic model are another kind of semantics-aware service discovery approaches, and many related approaches have been proposed in recent years [3, 17, 18]. LDA- SVM [22] is proposed to handle the issue of labeling a large number of services when a service classifier is trained. Cao et al. [6] propose a mashup service clustering method that exploits a two-level topic model to mine the latent useful and novel topics. Hafida et al. [25] propose a new content-based topic model to capture the maximal common semantic of sets of services. Samanta et al. [29] use the Hierarchical Dirichlet Process to intelligently discover the functionally relevant services. Gao et al. [12] develop a method of service co-occurrence LDA to extract latent service co-occurrence topics. Shi et al. [30] propose an augmented LDA model (named WE-LDA) to improve the performance of web services clustering. The derived knowledge of topics extracted by topic models does help to reveal the trend of service composition, understand the latent concepts of the services, and lead to better service recommendation.

Except for the topic model, some other traditional machine learning methods have also been employed to analyze the semantics within the descriptions or graphs of web services [20, 31, 37, 42]. Cheng et al. [8, 9] propose a conceptual services description model with the path for the interaction interface and the traditional text description. Hao et al. [13] propose a method for a specific mashup query, relying on the valuable information hidden in mashup descriptions. Gao et al. [11] propose a novel recommendation framework to improve the recommending accuracy of individual services. Rupasingha et al. [28] propose a CF-based recommendation approach for ontology generation. Surianarayanan et al. [33] propose a hierarchical agglomerative clustering-based approach for service discovery. Rodriguez-Mier et al. [27] propose a composition framework that enables the generation of a graph-based composition.

The recent research uses the approaches based on deep learning to learn the features of web services. For example, Yang et al. [39] present a deep neural network, named ServeNet, to the abstract low-level representation of service description to high-level features. Zou et al. [43, 44] propose DeepWSC to cluster services through automatic feature extraction.

Bai et al. [1] build a deep learning framework to perform accurate long-tail service recommendations. The main limitations of the works based on a deep learning framework include the non-interpretability of features and the requirements of large-scale data for model fitting.

3 bi-directional sentence-word topic model for service discovery

The most intuitive and efficient way for service discovery is that related services can be retrieved directly by different kinds of user queries including words, phrases, or short descriptions. Compared with the keywords and network structures of services, textual semantics lying within the queries and descriptions are considered to be more efficient in service discovery.

Figure 1 depicts the process of service discovery with semantic matchmaking. Users can submit a query represented by words, phrases, or sentences in the service discovery interface. With a semantic extractor, user queries and service descriptions in the repository are represented with semantic components; then, they are matched in the same semantic space. The dashed arrow in Figure 1 denotes the traditional syntactic-based searching process by keyword matching without analyzing the semantics of queries and services. These methods search the descriptions mainly by keyword matching which have low accuracy. They do not consider the latent semantics which can show many latent features of the words and various texts. Instead, the semantic-based approach, as shown as with the solid arrow in Figure 1, is the better way to analyze the queries and services by extracting the semantics of the service descriptions.

Thus, in our work, we seek to learn the semantics of both the service descriptions and the various user queries in order to compute the similarities of services and queries effectively. As shown with the solid arrow in Figure 1, we aim to extract the semantics of the queries from word-level, phrase-level, and sentence-level, and match them with the semantics of service descriptions.

To build the semantic bridge between user queries and service descriptions, it is essential to bridge the gap between the semantic matching of words and sentences in the service descriptions. In this work, a Bayesian probabilistic model named by bi-Directional Sentence-Word Topic Model (bi-SWTM) is proposed to learn the semantics of the words and sentences in an interpretable topic space. The bi-SWTM provides a novel perspective

Figure 1 Service discovery with the semantics of user queries and service descriptions. The dashed arrow denotes the traditional syntactic-based approaches with keyword matching. The solid arrow represents the process of semantic-based approaches, which is the roadmap of the proposed bi-SWTM

on modeling the multi-semantics of words and sentences with topic modeling. In this model, three metrics are jointly learned: the topic distributions of the sentences, the topic distributions of the words in the dictionary, and the word probabilities over the latent topics. The latent topics are extracted using the sequences of sentences; fine-grained topics are captured with local textual information such as sentences and paragraphs.

Different from the conventional methods on service discovery or clustering with topic modeling, the bi-SWTM assumes that the topic distribution of each sentence in service descriptions is not only determined by the concepts of the involved words but is also influenced by its preceding and subsequent sentences. This is the basic assumption in many text mining tasks, such as topic modeling [17] and neural language modeling [10]. In traditional topic modeling, the topics of a text follow a special distribution a priori such as a Dirichlet [3] or Normal [2] distributions. This assumption with a single prior is easily constrained by the data. Thus, the bi-SWTM holds an assumption of hybrid priors to generate the topics of the sentences. That is, the topics of the targeted sentence are generated by the mixture of the topics of the words in the sentence and its neighboring sentences. The related topics involved are the hybrid priors for the targeted sentences.

In addition, an attention mechanism is considered to leverage the weights of the hybrid priors to improve the effectiveness. When generating the topic distribution of a sentence, the attention signals can regulate the topic components from the involved words and the neighboring sentences. With this special mechanism, the more elaborate topics lying within the sentence level can be extracted for the semantic understanding step in service discovery.

3.1 Model definition

Let $R = \{ \mathbf{d}^1, \mathbf{d}^2, \dots, \mathbf{d}^M \}$ denote a service repository which contains M service descriptions, where $\mathbf{d}^i$, $i \in \{1, \dots, M\}$ indicates the i -th description in this repository. Each description $\mathbf{d}^i$ in the repository is defined as a sequence of sentences which is denoted by $(\mathbf{s}_1^i, \mathbf{s}_2^i, \dots, \mathbf{s}_{S^i}^i)$, where S^i is the number of sentences in $\mathbf{d}^i$. Similarly, each $\mathbf{s}_j^i$ is denoted by $(w_{j1}^i, w_{j2}^i, \dots, w_{jN_j^i}^i)$, where N_j^i is the number of words in $\mathbf{s}_j^i$. Here, we adopt the assumption of a bag-of-words (BoW). A dictionary v is made up by the words in the repository R , where the word is indexed by $\{1, 2, \dots, V\}$.

The topic space is denoted by $T = (z_1, z_2, \dots, z_K)$, which can be referred to by LDA [3]. The representations of the sentences in the repository and words in the dictionary are defined by the probability distributions in the same topic space T . We define ϑ as the topic probability distribution of a sentence in a service description. Let ϕ_{w_n} denote the topic distribution of the n -th word in sentence s . A new Bayesian process for generating the semantic probability distribution of the j -th sentence $\mathbf{s}_j^i$ in the service description $\mathbf{d}^i$ is defined as

$$\begin{aligned} \vartheta^{s_j^i} | \vartheta^{s_{j-C:j-1}}, \vartheta^{s_{j+1:j+C}}, \phi_{w_{j1}^i}, \dots, \phi_{w_{jn}^i}, \epsilon \sim \sum_p^C \epsilon_p \cdot \vartheta^{s_{j-p}} + \sum_q^C \epsilon_{q+C} \cdot \vartheta^{s_{j+q}} \\ + \epsilon_{2C+1} \cdot \phi_{w_{j1}^i} + \dots + \epsilon_{2C+n} \cdot \phi_{w_{jn}^i} + \dots + \epsilon_{2C+N_j} \cdot \phi_{w_{jN_j}^i}, \end{aligned} \quad (1)$$

where $\vartheta^{s_{j-C:j-1}}$ and $\vartheta^{s_{j+1:j+C}}$ denote the topic distribution matrices of the preceding and subsequent sentences, respectively. R is the contextual window size. $\phi_{w_{j1}^i}, \dots, \phi_{w_{jN_j}^i}$ indicate the topic distribution matrices of the words appearing in sentence $\mathbf{s}_j^i$. ϵ is an $(N + 2C) \times 1$ -dimensional weight vector, which follows a Dirichlet distribution with a hyperparameter π .

The prior of $\vartheta^{s_j^i}$ is hybrid generated by the distributions of neighboring sentences and the involved words with corresponding weight values. Equation (1) presents the generation of the topic distribution of sentence s_j^i , where the topic components are determined by both the bi-directional contextual sentences and the involved words. The preceding and subsequent sentences make up the linguistic contexts, which is a kind of long contextual information. The inner words make up the conceptual contexts, which is a kind of local contextual information. The graphical representation of the Bayesian process is shown in Figure 2.

Since we have $\epsilon \sim \text{Dir}(\pi)$, ϵ satisfies

$$\sum_i \epsilon_i = 1, \quad 0 \leq \epsilon_i \leq 1,$$

according to the property of Dirichlet distribution. Meanwhile, $\phi_{w_{jn}^i} \sim \text{Dir}(\alpha)$; thus, ϕ also satisfies

$$\sum_k \phi_{w_{jn}^i}^k = 1, \quad 0 \leq \phi_{w_{jn}^i}^k \leq 1.$$

With the properties of ϵ and ϕ , the form of $\vartheta^{s_j^i}$ follows a recursive definition, where the topic distributions of the preceding and subsequent sentences satisfy

$$\begin{aligned} \sum_k \theta_k^{s_{j-}} &= 1, \quad 0 \leq \theta_k^{s_{j-}} \leq 1, \\ \sum_k \theta_k^{s_{j+}} &= 1, \quad 0 \leq \theta_k^{s_{j+}} \leq 1. \end{aligned}$$

As all of the variables are unitized, $\vartheta^{s_j^i}$ satisfies

$$\sum_k \vartheta_k^{s_j^i} = 1, \quad 0 \leq \vartheta_k^{s_j^i} \leq 1.$$

Therefore, the random variable of $\vartheta^{s_j^i}$ with the hybrid prior also can be treated as following a special Dirichlet distribution. This enables the modeling of the sentences and words in the same topic simplex space [14, 26]. Note that the dimension of ϵ is specified by the host

Figure 2 The illustration of the Bayesian process for generating the topic distribution of sentence s_j^i by the preceding sentences, subsequent sentences, and the associated words with the corresponding weight values in a service description

sentence. The hyperparameter ϵ of the Dirichlet distribution is selected from a global vector $\pi \in \mathbb{R}^{1 \times (2C+V)}$ corresponding to the word index, where V is the size of the dictionary.

Based on the definition of the topic distribution for a sentence, we can describe the generation process of the bi-SWTM in detail. First, for each hidden topic $k \in \{1, \dots, K\}$, we draw $\beta_k \sim \text{Dir}(\mu)$, where μ is a V -dimensional parameter vector. For each word $w_v \in \{1, \dots, V\}$ in the dictionary, we draw $\phi_{w_v} \sim \text{Dir}(\alpha)$, where α is a K -dimensional parameter vector. $\beta \in \mathbb{R}^{K \times V}$ denotes the word probabilities of the hidden topics over the dictionary in the repository, and each row in $\phi \in \mathbb{R}^{V \times K}$ is the topic distribution of one word in the dictionary. Second, for the current sentence s_j^i , $j \in \{1, \dots, S_i\}$ in the description of a service d^i , we draw the topic distribution $\vartheta^{s_j^i}$ following (1). For each word w_n , $n \in \{1, \dots, N_j\}$ in sentence s_j^i , we draw $z_n \sim \text{Mult}(\vartheta^{s_j^i})$ and draw $w_n \sim \text{Mult}(\beta_{z_n})$, where $\text{Mult}(\cdot)$ is a multinomial distribution. The topic distribution of each sentence is determined by its bi-directional contexts and the words in it. After learning this model, we can obtain β , ϕ and the topic distribution of each sentence. Figure 3a shows the graphical representation of the bi-SWTM.

3.2 Model inference

The priors used in the proposed model provide the capacity to model the semantics among the various-size grained text elements as well as a challenge for the posterior inference procedure. This is because the topic generation process utilizing the hybrid priors leads to the non-conjugate relation between the topic assignment and the prior over the topic distribution of sentences. The traditional sampling-based methods, such as Gibb sampling, suffer from the computational complexity of the posterior distribution.

Thus, we adapt the variational Bayesian inference with a variational expectation-maximization (EM) algorithm [36]. In the variational EM algorithm, the E-step approximates the posterior by minimizing the Kullback-Leibler (KL) divergence between the variational distribution and the true posterior distribution. The M-step estimates the model parameters with the variational parameters learned by the E-step. This method casts the inference problem as an optimization problem to approximate the posterior distribution of

Figure 3 a Graphical representation of the proposed bi-SWTM. b The graphical representation of the fully-factorized variational distributions for $q(\epsilon_j^i, \{z_{w_n}\}_{n=1}^{N_j^i})$ in a service description d^i

a latent model. Minimizing the KL divergence is equivalent to maximizing the evidence lower bound (ELBO).

The latent variable of sentence $\mathbf{s}_j^i$ is the topic assignment z_{w_n} for each word and the weight vector ϵ_j^i . We define $\{\gamma_n\}_j^i$ as a group of variational parameters of multinomial distributions for $\{z_{w_n}\}_j^i$ and let $\xi_j^i \in \mathbb{R}^{(N+2C) \times 1}$ denote the variational parameter of a Dirichlet distribution for ϵ_j^i . Thus, for sentence $\mathbf{s}_j^i$, we use the following fully-factorized variational distribution:

$$q(\epsilon_j^i, \{z_{w_n}\}_j^i) = q(\epsilon_j^i | \xi_j^i) \prod_{n=1}^{N_j^i} q(z_{w_n} | \gamma_n). \quad (2)$$

The fully-factorized variational distributions are defined the approximation of the true posterior distributions, where $q(\cdot)$ denotes the distribution of variations with different parameters. The graphical representation of the fully-factorized variational distributions for $q(\epsilon_j^i, \{z_{w_n}\}_j^i)$ in document d^i is shown in Figure 3b. Then, based on Jensen's inequality, the ELBO on the log probability of the sentence $\mathbf{s}_j^i$ given the model parameters $\{\beta, \phi, \pi\}$ can be computed as:

$$\begin{aligned} \mathcal{L}^s(\epsilon, \gamma; \beta, \phi, \pi) &= \mathbb{E}_q[\log p(\epsilon | \pi)] \\ &+ \sum_{n=1}^{N_j^i} \mathbb{E}_q[\log p(z_n | \epsilon, \theta^{s_{j-C:j-1}}, \theta^{s_{j+1:j+C}}, \phi_{w_1}, \dots, \phi_{w_N})] \\ &+ \sum_{n=1}^{N_j^i} \mathbb{E}_q[\log p(w_n | z_n, \beta)] - \mathbb{E}_q[\log p(\epsilon)] - \mathbb{E}_q[\log p(z_n)], \end{aligned} \quad (3)$$

where the last two terms indicate the entropy of the variational distributions.

Based on the fully factorized variational distribution in (2), we can maximize the ELBO (3) to find the solution of the variational parameters via the variational expectation-maximization procedure as described above. For the variational parameter ξ , the corresponding objective function minimizes the following function:

$$\begin{aligned} \mathcal{L}_{[\xi]}^s &= \sum_l^{2C+N_j^i} (\pi_l - 1)(\Psi(\xi_l) - \Psi(\sum_{l'=1}^{2C+N_j^i} \xi_{l'})) - \log \Gamma(\sum_{l=1}^{2C+N_j^i} \xi_l) + \sum_{l=1}^{2C+N_j^i} \log \Gamma(\xi_l) \\ &- \sum_{l=1}^{2C+N_j^i} (\xi_l - 1) \cdot \left(\Psi(\xi_l) - \Psi\left(\sum_{l'=1}^{2C+N_j^i} \xi_{l'}\right) \right) \\ &+ \sum_{n=1}^{N_j^i} \sum_{k=1}^K \gamma_{nk} \cdot \left(\sum_{l=1}^{2C} \log \theta_{lk}^{j-C:j+C} \frac{\xi_l}{\sum \xi} + \sum_{l=2C}^{2C+N_j^i} \phi_{w_n} \frac{\xi_l}{\sum \xi} \right), \end{aligned} \quad (4)$$

where $\Psi(\cdot)$ indicates the digamma function that is the first derivative of the log of the Gamma function.

Also, after setting the derivatives of the ELBO to zero, we obtain the update equations for $\{\gamma_n\}_j^i$ as:

$$\gamma_{nk} \propto \beta_{k,v^{w_n}} \exp \left(\sum_{l=1}^{2C} \log \theta_{lk}^{j-C:j+C} \frac{\xi_l}{\sum \xi} + \sum_{l=2C}^{2C+N_j^i} \phi_{w_n} \frac{\xi_l}{\sum \xi} \right), \quad (5)$$

where v^{w_n} denotes the index of the word w_n in the dictionary. Thus, we can optimize the variational parameters, i.e., ξ and $\{\gamma_n\}_j^i$, for each sentence in the E-step.

In M-step, we update the model parameters $\{\beta, \phi, \pi\}$ by maximizing the lower bound after fitting ξ and $\{\gamma_n\}_j^i$. The update equations for ϕ and β are as follows:

$$\phi_{vk} \propto \sum_i^M \sum_j^{S_j^i} \gamma_{v^{w_n}k} \frac{\xi_{v^{w_n}}}{\sum_{l=2C}^{2C+N_j^i} \xi_l}, \quad (6)$$

and

$$\beta_{kv} = \sum_{i=1}^M \sum_{j=1}^{S_j^i} \sum_{n=1}^{N_j^i} \gamma_{v^{w_n}k} (w_n)^v. \quad (7)$$

We use the linear-time Newton-Raphson algorithm to estimate π , whose objective function minimizes the following function:

$$\mathcal{L}_{[\pi]} = \sum_{i=1}^M \sum_{j=1}^{S_j^i} \left(\log \Gamma \left(\sum_{l=1}^{2C+N_j^i} \pi_l \right) - \sum_{l'=1}^{2C+N_j^i} \log \Gamma(\pi_{l'}) + \sum_{l=1}^{2C+N_j^i} (\pi_l - 1) (\Psi(\xi_l) - \Psi \left(\sum_{l'=1}^{2C+N_j^i} \xi_{l'} \right)) \right). \quad (8)$$

The detailed algorithm is shown in Algorithm 1.

Algorithm 1 Variational EM algorithm for bi-SWTM.

- 1: INPUT: Sentences $(s_1^i, s_2^i, \dots, s_{S_i}^i)$ in repository R .
 - 2: OUTPUT: The topic distributions of the sentences and words, model parameters.
 - 3: Initialize the model parameters and the variational parameters.
 - 4: **repeat**
 - 5: (*E-Step*.)
 - 6: **for** each sentence s_j^i of each descriptions $\mathbf{d}^i$ in $\mathbf{D}$ **do**
 - 7: Update ξ_j^i via (4);
 - 8: Update γ_{nk} via (5).
 - 9: **end for**
 - 10: (*M-Step*.)
 - 11: update ϕ_{vk} via (6);
 - 12: update β via (7);
 - 13: update π (8) with Newton-Raphson algorithm.
 - 14: **until** convergence
-

4 Web services discovery with bi-SWTM

In the applications of services discovery, it is challenging to extract the semantics from short service descriptions comprised of limited terms coupled with the diverse naming conventions used by service providers [4]. The proposed bi-SWTM provides an effective and flexible method to address this challenge. It improves the capacity of extracting latent semantics for Bayesian topic modeling and offers a novel approach to extract the concepts of short service descriptions and consider the diverse naming conventions used by service providers.

Firstly, we generate the latent semantics using different textual elements in service descriptions, such as words (named entities), phrases, and sentences. All of the services can be embedded into an interpretable topic space with the bi-SWTM, which can be treated as feature engineering. The fine-grained learning mechanism can obtain more accurate topics from service descriptions.

Secondly, the topic compositions can be built using the bi-SWTM for complex user queries. For example, in a service retrieval system, all the services are embedded into the interpretable semantic space including the category of service, tags, keywords, and short descriptions. When a user retrieves a targeted service by keywords, phrases, or a general short description, the input can be identified by the proposed model. The model determines the exact semantics by linking the various-size-grained textual elements together.

In summary, for the retrieved services, the bi-SWTM is used to obtain the topic distributions of the service descriptions following Algorithm 1. The topic distributions of the service descriptions are the latent features that reveal the semantics of each service. Meanwhile, the bi-SWTM can also extract the semantics of the service descriptions and complicated user queries such as keywords, phrases, sentences, and paragraphs. With the bi-SWTM, each user query is assigned the latent topics before retrieving. The process of the topic assignment for user queries via the bi-SWTM is shown in more detail as follows.

4.1 Word level

In the case that a user inputs a word to search the services, the topics of the query word can be obtained from the bi-SWTM directly, where the ϕ_{vk} denotes the topic distribution of each word in the dictionary. After training the bi-SWTM, we can determine ϕ_{vk} from the entire corpus by (6); the latent topics denote the semantic components of the words. With the assumption of the bi-SWTM, the topic distribution of each word is embedded within the same topic space as the sentences and the paragraph. Thus, for each query word, we can compute the semantic similarities with all of the candidate service descriptions and retrieve the most relevant service for the target query by the matrix of cosine distance.

4.2 Phrase level

In the case that a user inputs a phrase to search the services, the topic distribution of the phrase query is generated by the inner words in the phrase. With the operation of hybrid priors, the inner words can be embedded into the same topic space. With the assumption of probability simplex space over the latent topics [3, 14], the topic distribution of several single words can be computed by the average of all the components in each topic. Given a

phrase $\mathbf{p}$ with a set of words, $(w_1, \dots, w_N)$, where N is the word number in $\mathbf{p}$, the topic proportion of z_k for $\mathbf{p}$ can be generated by:

$$p(z_k | w_1, \dots, w_N) \propto \frac{\sum_l^N \phi_{w_l, k}}{K}. \quad (9)$$

We can compute each of the topic proportions and obtain the topic distribution of $\mathbf{p}$ with the normalized operation. After the topic distribution of the phrase query is obtained with (9), we can retrieve all of the services using the approach for the keyword level described above.

4.3 Sentence level

In the case that a user inputs a sentence in the service discovery interface, the topics of the query sentence can be learned by the inference process of the bi-SWTM through (4) and (5). From Algorithm 1, the inference process is from Step 7 to Step 8 for one iteration. For all of the candidate service descriptions, we can compute the inference offline by the bi-SWTM to generate the topic distributions of all of the service descriptions. For the single query sentence without any contextual sentences, we set $C = 0$ in the bi-SWTM. However, a better approach is that we treat the query sentence as the contextual sentence itself. In this case, the query sentence can be better modeled by the bi-SWTM, where $C = 1$.

When the user retrieves the services by inputting a sentence, the topic distribution of the sentence is extracted by the inference process of the bi-SWTM; then, the relevant services can be ranked by the semantic distances of the topic distributions. Note that the phrases and a group of keywords are also treated as special sentences. With the assumption of BoW for the topic model, the sequential information of the words in a sentence is ignored. Thus, it is flexible for different scenarios of complicated user queries, where all the types of sentences can be used as the queries.

4.4 Paragraph level

The proposed bi-SWTM can also retrieve the services by inputting a long text consisting of several sentences that describes a target service. Different from the other topic models available in the literature, the proposed bi-SWTM can obtain the topic distributions of the query texts as follows.

When the user inputs a sequence of sentences, named paragraph $\mathcal{P}$, we can infer each sentence in the query by Algorithm 1 from Step 6 to Step 8 to obtain the topic components of each sentence. Following the operation described in [17], the topic distribution of the entire paragraph can be computed by summing all of the topic components of each sentence as:

$$(z_1, \dots, z_k) = \left(\frac{\sum_j^{S^{\mathcal{P}}} \sum_n^{N_j^{\mathcal{P}}} \{\gamma_{n1}\}_j^{\mathcal{P}}}{Z}, \frac{\sum_j^{S^{\mathcal{P}}} \sum_n^{N_j^{\mathcal{P}}} \{\gamma_{n2}\}_j^{\mathcal{P}}}{Z}, \dots, \frac{\sum_j^{S^{\mathcal{P}}} \sum_n^{N_j^{\mathcal{P}}} \{\gamma_{nk}\}_j^{\mathcal{P}}}{Z} \right), \quad (10)$$

where

$$Z = \sum_{\hat{k}}^K \sum_j^{S^{\mathcal{P}}} \sum_n^{N_j^{\mathcal{P}}} \{\gamma_{n\hat{k}}\}_j^{\mathcal{P}}. \quad (11)$$

Figure 4 The inference process of the web services discovery with bi-SWTM for word, phrase, sentence, and paragraph. The different types of lines and arrows are used to show the flows of each type of query

The topic distribution with K dimensions is defined by (10), where each component is summed by all the values of γ in each sentence. When the user inputs a description of a service, where the description is grouped by words, not a sequence of sentences, we can treat the description as a long sentence. Thus, the topic distribution of the description can be generated using the sentence level approach.

4.5 Discussion

The process of the web services discovery with bi-SWTM of hybrid grained textual queries is demonstrated in Figure 4. It is worthy to note that traditional topic models for the tasks of service discovery can only handle the long texts as the queries. They are unable to search for the relevant services by comparing the latent semantics of the word level queries and the candidate service descriptions. Thus, most of the previous works are based on keyword matching. The state-of-the-art models based on neural networks are also designed to extract the features of long texts (or service descriptions); however, they fail to learn the features of different level textual elements in a united framework. This leads to the low accuracy performance when the queries are complicated. Thus, the main contribution of the proposed bi-SWTM is that it learns the latent features of different level textual elements in the same computation semantic space, which brings the benefits in handling the complex user queries in service discovery.

5 Experiments

5.1 Dataset and training setting

To perform a standard and comparable evaluation, our experiments are conducted on the ProgrammableWeb.¹ ProgrammableWeb is a real-world web services repository that consists of over 23,000 web services. Each web service in the repository is described with a short unstructured text. We use the released repository and remove the web services with less than five words in the description. A subset#1 is obtained with 12,912 web services.

¹www.programmableweb.com

Table 1 Summary statistics of the subset#1 from ProgrammableWeb

#	Statistics
# of descriptions	12,912
# of sens.	68,766
# of words	980,421
length of the dic.	6,381
mean # of words per sen.	14.3
mean # of sens per doc.	5.3

Table 1 highlights the summary statistics of the dataset. We remove the stop words and the words that occur less than five times in the repository; this results in a data set with 6,381 words. Please refer to our webpage² for the code and data.

The proposed bi-SWTM is evaluated on language modeling, services classification, and service discovery. Also, the visualizations of the nearest-neighbored queries and descriptions are presented which shows the insights of our model on capturing the latent semantics of web services. We first train our model on the given repository taking the sequences of sentences as the input. For all of the following experiments, the number of topics K is set to 100, which means the dimension of the sentence representation is 100. We set $C = 1$, which means that one preceding sentence and one subsequent sentence are considered as the contexts of the current targeted sentence. When the targeted sentence is located at the head or tail of the descriptions, the sentence itself is treated as its neighbored contexts. The features of the selected sentences are extracted by the proposed model with the involved words and the neighbored sentences.

5.2 Comparison results of services classification

This section shows the comparison results of several representative methods for service classification. Different from the methods that focus on the syntactic aspects of services, we are more interested in assessing how well the representations capture the semantics of service descriptions. Therefore, we construct three types of service classification tasks with different grained texts, namely phrase (word) level, sentence level, and paragraph level.

5.2.1 Service classification on sentence level

Since the bi-SWTM can capture the concepts of sentences, service classification is first conducted on the sentence level. We reconstruct a subset#2 from subset#1, which contains 8,233 services belonging to 30 different categories (e.g., tools, financial, enterprise, e-commerce).

Each sentence in the service's description belongs to the same category of the host service, and the proposed method determines which category the target sentence belongs to. Thus, we randomly select one sentence from each description and let the service category be its label.

We compare our model bi-SWTM with the following methods. Word2Vec and FastText are selected to evaluate the BoW approaches, which are the most famous methods to obtain the word embeddings. Paragraph2Vec is the traditional method to learn paragraph embeddings based on the neural network. LDA is the traditional approach based on topic models,

²<http://shuangyin.li/biswtm/>

Table 2 The service classification results on sentence-level with accuracy and F1-score

Models	ProgrammableWeb	
	Accuracy	F1
Word2Vec	0.668	0.643
FastText	0.591	0.572
AutoEncoder	0.439	0.408
LDA	0.602	0.583
Paragraph2Vec	0.411	0.389
bi-SWTM	0.685	0.669

which is treated as the comparison of the Bayesian method. Meanwhile, we also compare the proposed model with the framework of the neural network by training an AutoEncoder. All of the comparisons are based on the unsupervised framework, where we evaluate the proposed model in creating reasonable text representations and demonstrate the capacity of the suggested hybrid grained semantic matchmaking for web service discovery.

- (1) Word2Vec (BoW). We train the Word2Vec(Skip-gram) by the implementation of Gensim.³ We train all of the words in the training set, including the words that occur less than five times in the repository. The dimension of each word vector is 100. A traditional BoW averaging was employed to produce the sentence embedding.
- (2) FastText (BoW). The code is obtained from the website.⁴ We train the FastText to obtain 100-dimensional word vectors. A traditional BoW averaging is employed to produce the sentence embedding.
- (3) Paragraph2Vec. We use the implementation from Gensim. The Paragraph2Vec learns paragraph and document embeddings via the distributed memory and distributed bag of words models [15].
- (4) LDA. We train the LDA⁵ model at the sentence level by setting the number of topics as 100. Since many works on the framework of web service classification or clustering are based on the traditional LDA, the conventional LDA is chosen as one of the comparisons.
- (5) AutoEncoder. We train the AutoEncoder with the 100 hidden units, which embeds the sentences into 100-dimension vectors. The AutoEncoder is well trained with the framework of PyTorch with a dropout rate of 0.5.

All of the above models are trained from scratch on ProgrammableWeb, respectively, and inference on the selected sentences to obtain the embeddings for a classifier. We use an SVM with Gaussian kernel as the classifier; the accuracy and F1-score are tested for all of the models. 80% sentences are used to train the classifier, and the 20% left are used for testing. Table 2 presents the results.

As one type of Bayesian generating model, the bi-SWTM provides better performance than the LDA. As described in [34], the length of the document plays a crucial role: poor performance of the LDA is expected when documents are too short, for example, taking the

³<https://radimrehurek.com/gensim/>

⁴<https://fasttext.cc>

⁵<http://www.cs.columbia.edu/~blei/lda-c/index.html>

sentences as the input. The proposed bi-SWTM overcomes this drawback by considering the context information for sentence-level modeling.

Table 2 demonstrates that the bi-SWTM outperforms the other approaches. It is worth noting that deep neural network-based methods, such as the AutoEncoder in our experiment, cannot obtain satisfactory performance on both accuracy and F1-score. The main reason is that this kind of method needs enough data for model fitting, while the web service repository is small. In contrast, the proposed bi-SWTM is based on the Bayesian model, which can handle small corpora in some closed domains, especially for the domains that need to be trained from scratch.

In addition, the bi-SWTM is compared with state-of-the-art methods on service classification, such as the WE-LDA [30] and the ServeNet [39]. Actually, the WE-LDA is tested on service classification with the services from ProgrammableWeb, where the services belong to the top 20 categories. Similarly, the web services in the top 50 categories are selected in the ServeNet. Even though the number of class labels used in the two methods is different as well as the scales of datasets, we can still compare them with the proposed model based on the same criteria.

In the experiment, we test our model on four different subsets from subset#2, in which the services are from the top of 20, 30, 40, and 50 categories, respectively. Note that the dimension of the features in the ServeNet is $K = 200$ and it is $K = 20$ in the WE-LDA. Thus, the bi-SWTM with $K = 20$ and $K = 200$ is trained accordingly.

Table 3 demonstrates the results of the three methods. From Table 3, our model significantly outperforms the WE-LDA with the same top 20 categories and $K = 20$. Meanwhile, the performance of the bi-SWTM is better than that of the ServeNet with the same category number and $K = 200$. Note that, $K = 20$, *category* = 20 for the WE-LDA, and $K = 200$, *category* = 50 for the ServeNet are the best parameters reported in the original papers. The methods based on deep neural networks may be limited by the scale of the data set, and it may be difficult to fit the model well with a small training set (See in Table 2). The proposed bi-SWTM is effective and competent in such scenarios.

In detail, since the settings used in the two representative baselines WE-LDA, and ServeNet, are different, to ensure a fair comparison, we conducted the numerical validations of bi-SWTM with different settings. Specifically, the bi-SWTM is tested with different settings, i.e., ($K = 20$, *category* = 20 as same as WE-LDA) and ($K = 200$, *category* = 50 as same as ServeNet) to show the performances with a fair assessment. The experimental results also demonstrate that our method outperforms the other two baselines, by 3%–8% increase in accuracy, which is a significant improvement in the evaluation system of machine learning.

Figure 5 presents the results of $K = 100$ with different classify difficulties, *category* = 20, 30, 40, 50. In the case that $K = 100$ and *category* = 20, the bi-SWTM achieves the best result. From Table 3, we find that the topic number is an important parameter in the web semantics extracting, where the topic number in the Bayesian models are

Table 3 Performances of web service classification on the sentence level, compared with WE-LDA and ServeNet

Models	Accuracy
WE-LDA, K=20, category=20	0.522
bi-SWTM, K=20, category=20	0.602
ServeNet, K=200, category=50	0.633
bi-SWTM, K=200, category=50	0.669

Figure 5 Performances of web service classification on sentence level with *category* = 20, 30, 40, 50 for $K = 100$

determined by the real data. From Figure 5, we find that we need to select the suitable K for different classification tasks. For *category* = 20, the bi-SWTM achieves the best result when $K = 100$, since it is easy to classify the 20 categories of web services. For *category* = 50, the best result of the proposed bi-SWTM occurs with $K = 200$, as more topics can represent the more detailed semantics to fit the categories of each service description. Thus, in practice, the best parameter of K is determined by the tasks of the service classification.

5.2.2 Service classification on phrase (word) level

Based on the results of service classification at the sentence level, we further test the performance of the proposed bi-SWTM on subset#2. In this part of the experiments, the tags of each service are used to represent the features of the service for the task of service classification. The average number of tags in each service is four (a small number of words are considered), and we treat them as a phrase.

Since the LDA cannot learn the word vectors directly, we only report the results of the Word2Vec and the bi-SWTM. For the Word2Vec, we also use the BoW approach to compute the feature of the phrase by summing the corresponding vectors of words in it. We retrain the Word2Vec model on all the descriptions of the services, after pre-training it on a large-scale Wikipedia. For the bi-SWTM, it learns the word features spontaneously, and we can simply add the vectors of the tag words to obtain the features of the phrases. Figure 6a and b show the performance of service classification on the phrase-level with 5-fold cross-validations. We can see that, with pretraining on the large-scale Wikipedia, the Word2Vec is slightly better than the bi-SWTM on the task of phrase classification. However, the bi-SWTM still achieves competitive results, even though the Word2Vec has the natural advantage on short text classification.

Figure 6 The 5-fold cross-validation classification performance on phrase and paragraph levels of web services from ProgrammableWeb

5.2.3 Service classification on paragraph level

To test the performance of service classification at the paragraph level, we use 12,912 descriptions of services (subset#1) in this part of the experiment. The features of the descriptions are obtained by LDA, Word2Vec, and bi-SWTM, respectively. Figure 6c and d represent the results with 5-fold cross-validations. Note that the bi-SWTM outperforms significantly both the Word2Vec and LDA. Meanwhile, the results also demonstrate that the Word2Vec has a limited capability on long text classification problems.

To further explore the impact on the text length of Word2Vec and bi-SWTM, we design the following experiments. We select the services for which the length of the descriptions are in the ranges of (20,30), (30,40), (40,50), (50,+). For each range, 1,500 descriptions are used. Figure 7 shows the 5-fold cross-validation classification results of the service descriptions with different text lengths. It is seen that the bi-SWTM always show a very competitive performance comparing with Word2Vec on both accuracy and F1 score.

Figure 7 The 5-fold cross-validation classification results of the web descriptions with different text lengths

5.2.4 Discussion

To prove the capacity of the proposed framework on eliminating the gap of different grained textual elements, we design a series of service classification tasks at the phrase (word), sentence, and paragraph levels. Experimental results demonstrate that the bi-SWTM achieves desirable performance on these service classification tasks.

Word2Vec, designed to learn word embeddings, can only handle word-level texts. Thus, to obtain the sentence embeddings with Word2Vec, a traditional BoW averaging is often employed. This operation is normally used in the scenario of short text classification. According to Le and Mikolov [15], this approach performs poorly for sentiment analysis tasks for two main reasons. Firstly, it loses the word order in the same way as the standard BoW models. Secondly, it fails to recognize many sophisticated linguistic phenomena, for instance, sarcasm. Therefore, if we try to obtain the sentence embeddings through Word2Vec with BoW averaging, the length of the sentence cannot be too long. Otherwise, Word2Vec computes on insignificant words. To address this limitation, some words are removed to reduce the length of the sentence, which limits the practicality of the approach, especially when the text is long and difficult to reduce.

In ProgrammableWeb, the mean number of words per sentence is 14.3, (as shown in Table 1), this is a great benefit to Word2Vec, especially when the sentences or phrases are used to evaluate the performance of service classification. Still, the proposed model bi-SWTM also outperforms Word2Vec by a small amount. However, the advantage of the bi-SWTM is its ability to handle long texts, such as the descriptions of web services. From Table 1, the average word number for each description reaches 76; this is where the proposed method exhibits strong accuracy results.

In summary, the Word2Vec and its variations excel in short texts, i.e., words or phrases. LDA and its variations perform well in the scenario of long texts. Our proposed model bi-SWTM combines the advantages of both, and specifies the hybrid grained textual elements in semantic matchmaking for service discovery. In practice, we can decide which method to choose simply by evaluating the length of texts in the specific application scenario. In the case with hybrid grained texts, our method is a better choice.

Figure 8 The accuracies of web classification tasks with top k , where $k=3$ and 5

Moreover, to validate the practicability of our approach, we test the accuracies of the top k in the tasks of service classifications at the paragraph level as shown as in Figure 8. Compared with LDA and Word2Vec, the bi-SWTM can achieve 0.97 accuracy values on the top five, which is sufficient to apply our model in real-world service discovery.

5.3 Comparison results of services discovery

In this section, we conduct experiments on information retrieval (IR) to evaluate the performance of service discovery in terms of F-Measure and precision. F-Measure and precision are standard evaluation measurements used in IR to indicate the performance of the search and matchmaking mechanism. When searching for services, a user may input keywords, phrases, or short sentences into the search engine. The search engine needs to analyze the concepts of the input and return the most related services. For example, “Graphics, Manufacturing, Monitoring, Video” is the set of tags for the web service of “NGRAIN”,⁶ that they can be used to search for related web services about “interactive 3D platform”. People may input one of the keywords, all of them as one phrase, or a sentence, like “Something to monitor equipment during manufacturing or in the field and improve efficiency”. Thus, to simulate the process of people searching services, a test query dataset based on ProgrammableWeb is constructed by the following steps.

- (1) One word from the tags or sub-categories of each service is extracted as a one-word query. The label of the word is the category of the responding service.
- (2) The keywords, e.g., tags, sub-categories, are extracted for each service from ProgrammableWeb. The category of the service is treated as the label of these keywords. These keywords are combined as a query phrase.
- (3) One sentence is randomly extracted from the description of each service. The category of the service is also treated as the label of the selected sentence.

The scale of the test query dataset is 1,000, which includes 100 word-level queries, 200 phrase-level queries, and 700 sentence-level queries. The subset#1 is used as the search

⁶<http://www.programmableweb.com/api/ngrain>

database. To determine whether a retrieved web service is relevant to the query text, we check whether they have the same class label. The embeddings of the query words, the query phrases, and the query sentences are obtained by the proposed bi-SWTM. The embedding of each query word is from ϕ , which is learned by the bi-SWTM. For each query phrase, the embedding of the phrase is generated following (9). The sentence embedding is also learned by the proposed bi-SWTM.

In the experiments, LDA is selected as the comparison method. Even though LDA is designed to handle long texts, we also apply it to extract the semantics of the sentences and keywords for comparison. Here, keywords and sentences are treated as short texts, then the LDA model is applied to learn the topic distributions. For each query, the web services in the database are ranked using the cosine distance as the similarity metric. We average the results of all the queries by using the F-Measure (F1-score) and precision values.

As shown in Figure 9a, the proposed bi-SWTM achieves 0.8@1 on precision, which means the related web services can be retrieved at the top of the ranked list. In the experiments, the words, phrases, and sentences are embedded into the same topic space with the retrieved services which provides an effective way to understand the queries from different level textual elements. Moreover, the bi-SWTM can capture the high-level semantics of the text, by which more reasonable services can be discovered in the process of retrieval.

Meanwhile, we also show the performance of different models on web service retrieval by drawing the precision-recall curves. Precision-recall curves are used when there is a moderate to large class imbalance. In service discovery, there are many different types of services that are handled for service matchmaking, sometimes long-tailed service matching. The precision-recall curves in Figure 9b present that the bi-SWTM works effectively for multi-type service searching, compared with the LDA.

Precision and recall are not particularly useful metrics when used in isolation. In the task of service matchmaking, we need to balance the precision and recall and an F1-score might be a better measure to use. Figure 9c reports the performance of service retrieval on the F1-score. The results show that the bi-SWTM outperforms the LDA from @1 to @1000.

According to the above experimental results of service classification, we can conclude that the bi-SWTM works well both on short texts and long paragraphs. When the user searches the desired services by natural language, the bi-SWTM provides an effective solution on the hybrid grained semantic matchmaking of service.

5.4 Comparison results of language modeling

As one of the Bayesian language models, we test the bi-SWTM on the language model by the held-out perplexity. In computational linguistics, the measure of held-out perplexity has been proposed to assess the generalizability of text models. A lower perplexity score

Figure 9 The results of web services retrieval on ProgrammableWeb. **a** Precision. **b** Precision-Recall curves. **c** F1-Score

Table 4 Held out perplexities of the different methods on ProgrammableWeb

Models	LDA	RATM	bi-SWTM
ProgrammableWeb	991.52	902.12	856.75

indicates better generalization performance. We define the perplexity on sentence-level as described in [16]. Considering M descriptions in the test set R_{test} , the perplexity is defined as:

$$Perplexity(R_{test}) = \exp \left(- \frac{\sum_{i=0}^M \sum_{j=0}^{S^i} \log p(s_j^i)}{\sum_{i=0}^M \sum_{j=0}^{S^i} N_j^i} \right), \quad (12)$$

where $\sum_{j=0}^{S^i} N_j^i$ indicates the total number of the words in $\mathbf{d}^i$.

We consider LDA and RATM [16] as the comparison models, and Table 4 shows the results of the held-out perplexities on subset#1, when the number of latent topics takes 100 for all the comparisons. We train on 80% service descriptions and test on the left 20%.

Our proposed model takes advantage of more information than LDA on sentence modeling, that is why our model achieves better results as shown as in Table 4. For example, LDA-based models depend on long texts, which lead to worse performance on

Table 5 Some random cases of query sentences and the corresponding best-neighbored sentence

S1: This API provides a platform for interactive rendering of 3D maps and allows you to embed them into your website.

Bestneighbor: The eeGeo 3D Map API allows developers to embed a 3D map into other applications and services.

S2: Layar is augmented reality application on your mobile phone which shows what is around you by displaying real time digital information on top of reality though the camera of your mobile phone.

Bestneighbor: NakdReality is a new location search service which allows anyone to quickly create their own location-based or augmented reality search, without having to hire specialist developers.

S3: By providing a valid phone number, users can transform any telephone into a temporary microphone and send recorded audio directly to the system.

Bestneighbor: The Knurld Call API provides developers an optional tool that can be used record audio for enrollments and verifications.

S4: Car Registrations in Portugal is an API that allows users look up car number plates in Portugal.

Bestneighbor: For instance, users of the application would enter a car registration plate like “2590 MA 64”, and it would return that the number plate belongs to a 2011 Ford Taurus.

S5: Strongteam is an AppStore that offers artificial intelligence and data mining APIs to let users pull information out of images, video and audio.

Bestneighbor: The API is useful to embed articles, images, video and audio directly into a developer’s applications.

perplexity [34] when handling short texts, such as sentences. In contrast, the proposed model is designed on the sentence level with a bi-Directional mechanism, and can more effectively capture the latent topics among the sentences than LDA and RATM.

5.5 Performance on nearest services

To demonstrate the capacity of the proposed model on learning the latent semantics of words and sentences, we design a nearest-neighbor retrieval experiment on subset#1 to test the capacity of our model on capturing the sentence semantics. The experiments examine properties of the embedding spaces to better understand how the model learns semantics. For a given query, we rank the candidate sentences in all of the services by cosine distance in the embedding space, and the best neighbor is retrieved. Table 5 shows some random cases of query sentences from the dataset and the corresponding retrieved sentences. The results show that the query sentences are highly related to the retrieved sentences on semantics.

Meanwhile, one of the benefits of our model is to embed the words and the sentences into the same topic space, which means that we can compute the similarity of a query sentence and a word by the representations learned by bi-SWTM. Thus, to demonstrate the internal behavior of our model on semantic extracting for sentences and words, we show the top 10 nearest words for the selected sentences and visualize them by t-SNE [23] on Figure 10. The points represent the words, and the stars denote the selected sentences.

Figure 10 The plots of the selected sentences and the nearest words from ProgrammableWeb. Each plot denotes a word, and a star denotes a sentence described in Table 5

6 Conclusion

In this paper, we propose bi-SWTM, a novel approach to achieve the semantic matchmaking for the words and sentences in the retrieval tasks of service discovery. The proposed model embeds the words and sentences from service descriptions into an interpretable topic space, by which we can build a bridge in matching the semantics of complex queries and service descriptions. The proposed model is examined using ProgrammableWeb data with service classification and retrieval, and the experiments show that our model can significantly outperform state-of-the-art methods for service discovery. Meanwhile, the operation of representing the words and sentences with interpretable semantic distributions provides a new perspective to understanding the queries and descriptions; this can benefit the investigation of other methods for service discovery when designing service recommendation or composition systems. Future works include automatic learning for the selection of topic numbers and improvement of the model efficiency.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (No. 62006083, No. 62002123), Science and Technology Projects in Guangzhou (No. 202102020654) and National Key Research and Development Program of China (No. 2018YFB1404402 and No. 2020YFA0712500). Haoyu Luo is the corresponding author, and Gansen Zhao is the co-corresponding author.

Declarations

Conflict of Interests We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Bai, B., Fan, Y., Tan, W., Zhang, J.: Dltsr: a deep learning framework for recommendations of long-tail web services. *IEEE Trans Serv Comput* **13**(1), 73–85 (2020)
2. Blei, D., Lafferty, J.: Correlated topic models. *Adv Neur Inform Process Syst* **18**, 147 (2006)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. *J Mach Learn Res* **3**, 993–1022 (2003)
4. Bouguettaya, A., Singh, M., Huhns, M., Sheng, Q.Z., Dong, H., Yu, Q., Neiat, A.G., Mistry, S., Benattallah, B., Medjahed, B., et al: A service computing manifesto: the next 10 years. *Commun ACM* **60**(4), 64–72 (2017)
5. Bukhari, A., Liu, X.: A web service search engine for large-scale web service discovery based on the probabilistic topic modeling and clustering. *SOCA* **12**(2), 169–182 (2018)
6. Cao, B., Liu, X.F., Rahman, M.M., Li, B., Liu, J., Tang, M.: Integrated content and network-based service clustering and web apis recommendation for mashup development. *IEEE Trans Serv Comput* **13**(1), 99–113 (2020)
7. Chen, F., Li, M., Wu, H., Xie, L.: Web service discovery among large service pools utilising semantic similarity and clustering. *Enterprise Inform Syst* **11**(3), 452–469 (2017)

8. Cheng, B., Zhao, S., Li, C., Chen, J.: Misda: web services discovery approach based on mining interface semantics. In: 2016 IEEE International conference on Web services (ICWS), pp. 332–339. IEEE (2016)
9. Cheng, B., Li, C., Zhao, S., Chen, J.: Semantics mining & indexing-based rapid web services discovery framework. *IEEE Transactions on Services Computing* (2018)
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805 (2018)
11. Gao, W., Wu, J.: A novel framework for service set recommendation in mashup creation. In: 2017 IEEE International conference on Web services (ICWS), pp. 65–72. IEEE (2017)
12. Gao, Z., Fan, Y., Wu, C., Tan, W., Zhang, J., Ni, Y., Bai, B., Chen, S.: Seco-lda: mining service co-occurrence topics for composition recommendation. *IEEE Trans Serv Comput* **12**(3), 446–459 (2019)
13. Hao, Y., Fan, Y., Tan, W., Zhang, J.: Service recommendation based on targeted reconstruction of service descriptions. In: 2017 IEEE International conference on Web services (ICWS), pp. 285–292. IEEE (2017)
14. Hofmann, T.: Probabilistic latent semantic analysis. In: Proceedings of the fifteenth conference on uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., pp. 289–296 (1999)
15. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International conference on machine learning, pp. 1188–1196 (2014)
16. Li, S., Zhang, Y., Pan, R., Mao, M., Yang, Y.: Recurrent attentional topic model. In: 31th AAAI Conference on artificial intelligence (AAAI-17), association for the advancement of artificial intelligence (AAAI), pp. 3223–3229 (2017)
17. Li, S., Zhang, Y., Pan, R.: Bi-directional recurrent attentional topic model. *ACM Trans Knowl Discov Data* **14**, 6 (2020)
18. Li, S., Zhang, Y., Pan, R., Mo, K.: Adaptive probabilistic word embedding. In: Proceedings of The Web conference 2020, association for computing machinery, New York, NY, USA, WWW '20, p. 651–661 (2020). <https://doi.org/10.1145/3366423.3380147>
19. Li, S., Pan, R., Luo, H., Liu, X., Zhao, G.: Adaptive cross-contextual word embedding for word polysemy with unsupervised topic modeling. *Knowledge-Based Systems*, 106827. <https://doi.org/10.1016/j.knosys.2021.106827>. <https://www.sciencedirect.com/science/article/pii/S0950705121000903> (2021)
20. Liang, T., Chen, L., Wu, J., Bouguettaya, A.: Exploiting heterogeneous information for tag recommendation in api management. In: 2016 IEEE International conference on Web services (ICWS), pp. 436–443. IEEE (2016)
21. Liu, X., Agarwal, S., Ding, C., Yu, Q.: An lda-svm active learning framework for web service classification. In: 2016 IEEE International conference on Web services (ICWS), pp. 49–56 (2016)
22. Liu, X., Agarwal, S., Ding, C., Yu, Q.: An lda-svm active learning framework for web service classification. In: 2016 IEEE International conference on Web services (ICWS), pp. 49–56. IEEE (2016)
23. Lvd, M., Hinton, G.: Visualizing data using t-sne. *J Mach Learn Res* **9**, 2579–2605 (2008)
24. Nabli, H., Djemaa, R.B., Amor, I.A.B.: Efficient cloud service discovery approach based on lda topic modeling. *J Syst Softw* **146**, 233–248 (2018)
25. Naim, H., Aznag, M., Quafafou, M., Durand, N.: Probabilistic approach for diversifying web services discovery and composition. In: 2016 IEEE International conference on Web services (ICWS), pp. 73–80. IEEE (2016)
26. Patterson, S., Teh, Y.W.: Stochastic gradient Riemannian Langevin dynamics on the probability simplex. In: Advances in neural information processing systems, pp. 3102–3110 (2013)
27. Rodriguez-Mier, P., Pedrinaci, C., Lama, M., Mucientes, M.: An integrated semantic web service discovery and composition framework. *IEEE Trans Serv Comput* **9**(4), 537–550 (2015)
28. Rupasingha, R.A.H.M., Paik, I.: Improving service recommendation by alleviating the sparsity with a novel ontology-based clustering. In: 2018 IEEE International conference on Web services (ICWS), pp. 351–354 (2018)
29. Samanta, P., Liu, X.: Recommending services for new mashups through service factors and top-k neighbors. In: 2017 IEEE International conference on Web services (ICWS), pp. 381–388 (2017)
30. Shi, M., Liu, J., Zhou, D., Tang, M., Cao, B.: We-lda: a word embeddings augmented lda model for web services clustering. In: 2017 IEEE International conference on Web services (ICWS), pp. 9–16. IEEE (2017)
31. Shi, W., Liu, X., Yu, Q.: Correlation-aware multi-label active learning for web service tag recommendation. In: 2017 IEEE International conference on web services (ICWS), pp. 229–236. IEEE (2017)
32. Su, X., Zhang, M., Mu, Y.: Trust-based group services selection in web-based service-oriented environments. *World Wide Web* **19**(5), 807–832 (2016)
33. Surianarayanan, C., Ganapathy, G.: An approach to computation of similarity, inter-cluster distance and selection of threshold for service discovery using clusters. *IEEE Trans Serv Comput* **9**(4), 524–536 (2015)

34. Tang, J., Meng, Z., Nguyen, X., Mei, Q., Zhang, M.: Understanding the limiting factors of topic modeling via posterior contraction analysis. In: International conference on machine learning, pp 190–198 (2014)
35. Tian, G., Liu, P., Peng, Y., Sun, C.: Tagging augmented neural topic model for semantic sparse web service discovery. *Concurr Comput: Pract Exp* **30**(16), e4448 (2018)
36. Wainwright, M.J., Jordan, M.I.: Graphical models, exponential families, and variational inference. *Found Trends Mach Learn* **1**(1-2), 1–305 (2008)
37. Wang, H., Shi, Y., Zhou, X., Zhou, Q., Shao, S., Bouguettaya, A.: Web service classification using support vector machine. In: 2010 22nd IEEE International conference on tools with artificial intelligence, vol. 1, pp. 3–6. IEEE (2010)
38. Xie, F., Wang, J., Xiong, R., Zhang, N., Ma, Y., He, K.: An integrated service recommendation approach for service-based system development. *Exp Syst Applic* **123**, 178–194 (2019)
39. Yang, Y., Ke, W., Wang, W., Zhao, Y.: Deep learning for web services classification. In: 2019 IEEE International conference on Web services (ICWS), pp 440–442 (2019)
40. Yu, Q.: Qos-aware service selection via collaborative qos evaluation. *World Wide Web* **17**(1), 33–57 (2014)
41. Zhang, N., Wang, J., Ma, Y., He, K., Li, Z., Liu, X.F.: Web service discovery based on goal-oriented query expansion. *J Syst Softw* **142**, 73–91 (2018)
42. Zhang, P., Shu, S., Zhou, M.: An online fault detection model and strategies based on svm-grid in clouds. *IEEE/CAA J Automatica Sinica* **5**(2), 445–456 (2018)
43. Zou, G., Qin, Z., He, Q., Wang, P., Zhang, B., Gan, Y.: Deepwsc: a novel framework with deep neural network for web service clustering. In: 2019 IEEE International conference on Web services (ICWS), pp 434–436 (2019)
44. Zou, G., Qin, Z., He, Q., Wang, P., Zhang, B., Gan, Y.: Deepwsc: clustering web services via integrating service composability into deep semantic features. *IEEE Trans Serv Comput*, 1–1 (2020)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Shuangyin Li¹ · Haoyu Luo¹ · Gansen Zhao¹ · Mingdong Tang² · Xiao Liu³

Shuangyin Li
shuangyinli@scnu.edu.cn

Mingdong Tang
mdtang@gdufs.edu.cn

Xiao Liu
xiao.liu@deakin.edu.au

¹ School of Computer Science, South China Normal University, Guangzhou, China

² School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou, China

³ School of Information Technology, Deakin University, Geelong, Australia