Recently, Mashup is becoming a promising software development method in the mobile service computing environment, which enables software developers to compose existing mobile services to create new or value-added composite RESTful web application. Due to the rapid increment of mobile services on the Internet, it is difficult to find the most suitable services for building user-desired Mashup application. In this paper, we integrate word embeddings enhanced hierarchical Dirichlet process and factorization machines to recommend mobile services to build high-quality Mashup application. Our method, first of all, extends the description documents of Mashup applications and mobile services by using Word2vec tool and derives latent topics from the extended description documents of Mashup and mobile services by exploiting the hierarchical Dirichlet process. Secondly, the factorization machine is applied to train these latent topics to predict the probability of mobile services invoked by Mashup and recommend mobile services with high-quality for Mashup development. Finally, the performance of the proposed method is comprehensively evaluated. The experimental results indicate that compared with the existing recommendation methods, the proposed method has significant improvements in MAE and RMSE.

1. Introduction

Service computing offers an exciting paradigm for service provision and consumption. The field is now embracing new opportunities in the mobile Internet era, which is characterized by ubiquitous wireless connectivity and powerful smart devices that let users consume or provide services anytime and anywhere [1]. As emerging techniques such as cloud and mobile computing become more prevalent, the way we provide and consume services is ever-changing [1]. Due to the rapid development and effective utilization of mobile computing technologies, services are no longer limited to traditional platforms and contexts. To some degree, traditional service computing is extended by delivering service through mobile techniques. That is to say, service can be deployed on cloud servers or mobile devices and delivered over wireless networks. Mobile devices can play the roles of provider, broker, and consumer simultaneously. Mobile service computing (i.e., the combination of service computing and mobile computing) is undoubtedly enabling us to provide and access services anytime and anywhere, which greatly facilitates our life, work, and study [1]. Recently, Mashup technology is becoming a popular software development method in the mobile service computing environment. It allows software developers to compose existing mobile services to create novel and composite RESTful services [2]. Tremendous mobile services have been released by various service providers [3]. For instance, until November 2018, there are more than 20,000 services on ProgrammableWeb. Therefore, it is a great challenge to find the most suitable service from the tremendous services to build the Mashup application that the user expects. To solve the above challenges, some researchers use service recommendation...
techniques to improve service discovery [4, 5]. Among them, some topic model technologies, such as latent Dirichlet allocation (LDA) [6], have been utilized to obtain the latent topics of Mashup and services to improve the accuracy of recommendations [4, 5]. However, LDA needs to identify the optimal number of topics in advance. To obtain the optimal topics, it is needed to repetitively perform model training, which leads to massive time-consumption. Aiming at this problem, the hierarchical Dirichlet process (HDP) model is proposed by Teh et al. [7], which can derive the optimal number of topics and save the time and cost. We use it to model and derive the topics of Mashups and service to achieve more accurate service recommendations. Moreover, the topic training and modelling of LDA usually needs large-scale corpus. However, the description documents of Mashup and service usually are short, and their corpuses are insufficient. In the field of information retrieval, some researchers exploit Word2vec [8] to expand short text into long text, in order that topic model can effectively estimate the latent topics of text for more accurate information searching. The Word2vec model is proposed by Google [8], which can process large-scale text corpus and generate word embeddings vector with high efficiency. In this paper, we exploit Word2vec to extend the description document of Mashup and mobile service to build a dense word embeddings vector representation for more accurate topic modelling.

Recently, matrix factorization is widely applied in service recommendations [9, 10]. It usually decomposes Mashup-service matrix into two matrices with lower dimension by using service invocations in historical Mashups. But matrix factorization based on service recommendation technology mainly depends on enough records in historical Mashup-service interactions [10]. To solve this problem, some additional information, such as users' social relations [11] or location similarity [12], is incorporated into matrix factorization to achieve a better accuracy of service recommendation. However, matrix factorization only is applicable to special and single input data and is not suitable for general prediction task. Therefore, the incorporation of additional information drops the performance of service recommendation. As we investigated, Rendle [13, 14] proposed a general predictor working with any real-valued feature vector FMs (factorization machines). It can be used for general prediction task and model all interactions between various input variables. Therefore, it can predict the probability of service invoked by Mashups. In this paper, we combine word embeddings enhanced HDP and FMs to recommend mobile service for Mashup development. The contributions of this paper are as follows:

(i) We use Word2vec to extend the description documents of Mashups and mobile services to build a dense word embeddings vector representation for more accurate topic modelling. Based on the word embeddings vector, we exploit HDP to derive the latent topics from the extended description document of Mashups and mobile services.

(ii) We employ the FMs to train the latent topics derived from the HDP and predict the probability of mobile services invoked by Mashups. Various valuable information, such as cooccurrence and popularity, is exploited to achieve high-quality mobile service recommendation for Mashup development.

(iii) We crawl a real Web service dataset from ProgrammableWeb and perform a series of experiments. The experimental results indicate that compared with the existing methods, the proposed method has significant improvements in MAE and RMSE.

(iv) The rest of this paper is organized as follows: Section 2 indicates the proposed approach. Section 3 presents the experimental results. Section 4 reviews related works. Section 5 provides discussion. Finally, we draw conclusions and discuss future work in Section 6.

2. Method Overview

This section consists of four subsections, respectively, describing the overall framework, description extension, topic modelling, and mobile service recommendation of the proposed method with details.

2.1. Overall Framework of Mobile Service Recommendation. The overall framework of our mobile service recommendation method is shown in Figure 1, which includes three main parts, i.e., description extension, topic modelling, and service recommendation. In the description extension part, we firstly extract the description documents of Mashup and mobile service from Web service dataset and then exploit the English Wikipedia corpus trained by Word2vec as extension source of the description documents of Mashup and mobile service to obtain their extended words. Finally, the original description documents and extended words of Mashup and mobile service are together preprocessed as the input of next step. In the topic modelling part, we use HDP topic technology to model the extended description documents of Mashups and mobile services and derive their latent topics. In the service recommendation part, when a user submits a Mashup requirement, our method applies FM model to perform the prediction and recommendation of mobile service for the given Mashup requirement.

2.2. Description Extension of Mashup and Mobile Service Based on Word2vec. The description documents of Mashup and mobile service are usually short, in which the number of contained words is relatively few and the word frequency cooccurrence is not enough. For example, according to our statistics, on an average, every service description document on ProgrammableWeb contains only 27.16 words. Due to the limited words (corpus), it is difficult to effectively derive the latent topics when using HDP topic model to train and model the description documents of Mashup and mobile
service. Therefore, it is very necessary for topic modelling to extend the description documents of Mashup and mobile service. Google develops a tool Word2vec [8] to express words as real numerical vectors in 2013. It is an open source word embeddings vector toolkit, which simplifies the document content processing into vector operations in K-dimensional vector space by using the idea of deep learning [8]. The word embeddings vector contains not only the semantic and grammatical relations of words but also extracts the context information of the document, which achieves more accurate word embeddings vector representation. As we know, Wikipedia is recognized as the most comprehensive and authoritative online encyclopedia on the Internet, which possesses rich corpus. We use Wikipedia corpus as the extension source of the description documents of Mashup and mobile service. More concretely, first of all, we use Word2vec tool to train Wikipedia corpus and obtain the word embeddings vector model of Wikipedia corpus. Then, we exploit the trained word vector model to extend the description documents of Mashup and mobile service. That is to say, for each word \( w_i \) in the preprocessed service description document \( S_{\text{ext}} \), the most similar Top-N words \( T_w = (t_1, t_2, \ldots, t_N) \) to \( w_i \) are identified from the word embeddings vector space of Wikipedia corpus and used as the extended words. The extended service description document is denoted as \( S_{\text{ext}}^{\text{ExText}} = \{(w_1, T_{w_1}), (w_2, T_{w_2}), \ldots, (w_n, T_{w_n})\} \), where \( n \) is the total number of words in service description document and \( N \) is the number of extended words. In the next section, we will perform comparative experiments to determine the optimal number of extended words.

The word embeddings vector model of Wikipedia corpus is trained by adopting the CBOW model based on the negative sampling and the Negative Sampling algorithm, in which the more close words in word meaning indicate more close distance in their word vectors space. These close words have similar semantic and grammatical relations, which can be used for service description document expansion. The input of the CBOW model is the word vector of the context-related words of a specific word, and its output is just the word vector of the specific word. Suppose \( w, C_w \) represent the extracted word and their context related words information from the service description document dataset, respectively. The probability of predicting \( w \) by the context-related words (surrounding words) can be denoted as follows:

\[
p(w|C_w) = \frac{1}{1 + e^{-W_w T w}} \tag{1}
\]

where \( W_w \) is the parameter of the hidden layer and softmax layer in the neural network and \( v_w \) is the sum of the vectors of each word in \( C_w \). The training objectives of the CBOW model are defined as follows (maximum likelihood estimation function):

\[
\text{OBJ}_{\text{CBOW}} = \arg \max_{V_w, W_1} \left( \prod_{(w,C_w) \in T} \log(\theta(w|C_w)) \right) \times \prod_{(w,C_w) \in T} \left( 1 - \log(\theta(w|C_w)) \right) \tag{2}
\]

Next, we use the HDP model to model the extended service description document \( S_{\text{ext}}^{\text{ExText}} \), to automatically extract the optimal number of topics and fully estimate the latent topics distribution.

2.3. Topic Modelling of Mashup and Mobile Service Using HDP. The hierarchical Dirichlet process (HDP) is a Dirichlet process (DP) mixture model with multilevel form, and it is also a nonparametric Bayesian approach to clustering grouped data [15]. Assume \((\Theta, B)\) is the measurable space, where \( G_0 \) is a spatial probability measure and \( a_0 \) is a positive real number. The Dirichlet process [16] is interpreted as the
distribution of the random probability measure $G$ over $(\Theta, B)$. If $G$ satisfies the Dirichlet distribution, for any finite partition $(A_1, A_2, \ldots, A_r)$ of the measure space $\Theta$, there is a random vector $(G(A_1), \ldots, G(A_r))$ distributed as a finite-dimensional Dirichlet distribution with the parameters $(a_0 G_0(A_1), \ldots, a_0 G_0(A_r))$:

$\left( G(A_1), \ldots, G(A_r) \right) \sim \text{Dir}(a_0 G_0(A_1), \ldots, a_0 G_0(A_r))$.  

(3)

HDP is used to model documents for mobile services and Mashup. Figure 2 is an HDP probability graph, which clearly indicates the documents of mobile services or Mashup and their words and potential topics. Among them, $\gamma$ and $a_0$ represent concentration parameter and $D$ represents the entire Mashup document set in which each Mashup document in $D$ is represented as $d$. At the lower part of Figure 2, $H$ represents the base probability measure and $G_0$ indicates the global random probability measure. The generated topic probability distribution of Mashup document $d$ is represented as $G_d$, and the generated topic of the $n$-th word in $d$ from $G_d$ is represented as $\psi_{d,n}$, and $w_{d,n}$ is a generated word from $\beta_{d,n}$.

The generative process of the HDP model is as follows:

1. Sample the probability distribution $G_0 \sim \text{DP}(\gamma, H)$
2. For each $d$ in $D$: sample a topic distribution $G_d \sim \text{DP}(a_0, G_0)$
3. For each word $n \in \{1, 2, \ldots, N\}$ in $d$,
   a. Sample a topic of the $n$-th word $\psi_{d,n} \sim G_d$  
   b. Sample a word from the multinomial distribution of the topic words $w_{d,n} \sim \text{multi}(\psi_{d,n})$

In order to perform HDP sampling, it is needed to devise a construction method to infer the posterior distribution of parameters. Chinese Restaurant Franchise (CRF) is a representative construction method which provides a way to construct the Dirichlet process. Assume there are $j$ Chinese restaurants containing $m_j$ tables ($\psi_{j,t} | t = 1, \ldots, m_j$), and each table sits $k_j$ customers. In the $j$ restaurant, each table shares the food $\phi_{j,k}$, where $K$ is the amount of food. Customers can choose a table at random, and each table is served a dish from a menu common to all restaurants. In this way, the customers, restaurants, and food in the Chinese restaurant correspond to the words, documents, and topics in our HDP model, respectively. Assuming $\delta$ is a probability measure, the topic distribution $\theta_j$ of the word $x_j$ is treated as the customer entering the restaurant, and the different value $\psi_{j,t}$ corresponds to the table where the customer is seated. The customer sits the table $\psi_{j,t}$ with a probability $n_{jt}/(i-1+a_0)$ or chooses with a new probability $a_{jt}/(i-1+a_0)$ to sit the new table $\psi_{j, m+1}$, sharing the food $\phi_{j,k}$. Among them, $n_{jt}$ indicates the sum of customers at the $t$-th table of the $j$-th restaurant. If the customer chooses a new table, she/he can distribute the food $\phi_{j,k}$ for the new table with a probability $m_j/(m_j + \delta)$, according to the popularity of chosen foods, or new foods $\phi_{j,m+1}$ with a probability $\delta/(m_j + \delta)$. Here, $m_j$ indicates the sum of tables providing the food $\phi_{j,k}$. We have the conditional distributions:

$$\begin{align*}
\theta_j &\sim \text{DP}(a_0, G_0) \\
\psi_{j,t} &\sim \text{Dir}(\theta_j) \\
\phi_{j,k} &\sim \text{Dir}(\psi_{j,t})
\end{align*}$$

2.4. Mobile Service Recommendation for Mashup Using FMs

2.4.1. Rating Prediction in Recommendation System and FMs.

In the traditional recommendation system, as for user set $U = \{u_1, u_2, \ldots\}$ and item set $I = \{i_1, i_2, \ldots\}$, the rating prediction function is denoted as follows:

$$y: U \times I \rightarrow R,$$

where $y$ is the rating and $y(u, i)$ represents the rating of user $u$ to item $i$.

FMs are a universal predictor that estimates reliable parameters at very high sparsity [13, 14]. It integrates the advantages of SVMs with factorization models. Different from the SVM, it not only is suitable for any real-valued feature vector but also can use decomposition parameters to model all interactions between feature variables. Therefore, it is very suitable for predicting the rating of items for users. Assume there are an input feature vector $x \in R^{m \times p}$ and an output target vector $y = (y_1, y_2, \ldots, y_n)^T$. Here, $n$ is the sum of input-output pairs, $p$ denotes the sum of input features, i.e., the $i^{th}$ row vector $x_i \in R^p$, $p$ represents $x_i$ has $p$ input feature values, and $y_i$ is the predicted target value of $x_i$. On the basis of $x$ and $y$, the 2-order FMs can be denoted as follows:

$$\hat{y}(x) = w_0 + \sum_{i=1}^{p} w_i x_i + \sum_{i=j+1}^{p} \sum_{j=1}^{p} x_j x_i + \sum_{f=1}^{k} \sum_{i=1}^{p} v_{i,f} v_{j,f},$$

(7)
where $w_0$ is the global bias and $k$ is the dimensionality of factorization. $w_0$ models the strength of the $i$-th feature, and $x_i x_j$ indicates all the pairwise variables in the training instances $x_i$ and $x_j$. The model parameters $\{w_0, w_1, \ldots, w_p, v_1, \ldots, v_p\}$ are denoted as follows:

$$
w_0 \in \mathbb{R},
\quad w \in \mathbb{R}^p,
\quad V \in \mathbb{R}^{p \times k}.
$$

2.4.2. Prediction and Recommendation of Mobile Service for Mashup Based on FMs. The prediction and recommendation of mobile service is a typical classification problem, and it is regarded as the task of ranking mobile services and recommending enough related mobile services for a given Mashup. The result of classification can be denoted as $y = \{-1, 1\}$. When $y = 1$, the relevant mobile services are recommended to the given Mashup. However, in the experiment, we can only obtain the predicted values ranging from 0 to 1 by formula (5). We firstly rank these prediction values, then label the top-$K$ results as positive (+1) and the rests as negative (−1), and finally recommend the mobile services with positive values to the given or target Mashup.

In the modelling of FMs, target Mashup and active mobile services can be considered as user and item, respectively. In addition to the two-dimensional features (target Mashup and active mobile services), we add other features, such as similar mobile services, similar Mashup, popularity, and cooccurrence of mobile services, to improve the accuracy of prediction and recommendation. The additional features can be used as input feature vectors in FM modelling. Therefore, the model in formula (6) can be extended to the below prediction model with six dimensions:

$$
y: MA \times MS \times SMS \times SMA \times CO \times POP \longrightarrow S,
$$

where MA is the target Mashup, MS is the active mobile service, SMS represents a similar Mashup, SMA represents a similar mobile service, CO represents the cooccurrence of mobile service, POP indicates the popularity of mobile service, and $S$ indicates the prediction ranking score, respectively. These similar Mashups and mobile services are derived from our HDP model in Section 2.3.

Figure 3 is an example of recommending mobile services for target Mashup using the FM model, in which the data consist of two parts. The first part is the input feature vector set $X$, and the second part is the output target set $Y$. Each row includes a feature vector $x_i$, and its corresponding target score value $y_i$. The first binary indicator matrix (i.e., Box 1) indicates the target Mashup MA. The second binary indicator matrix (i.e., Box 2) indicates the active mobile service MS. The third indicator matrix (i.e., Box 3) represents that Top-$S$ mobile services are similar to active mobile service in Box 2. For example, the similarity between $S_1$ and $S_2$ ($S_3$) is 0.3 (0.7). The fourth indicator matrix (i.e., Box 4) represents Top-$M$ similar Mashups SMA of the target Mashup in Box 1. For instance, the similarity between $M_2$ and $M_1$ ($M_3$) is 0.3 (0.7). The fifth indicator matrix (i.e., Box 5) indicates the cooccurrence CO of active mobile services composed or invoked by the same Mashup in historical records. The sixth indicator matrix (i.e., Box 6) indicates the popularity (or the frequency/times) POP of active mobile services composed or invoked by Mashup. Target $Y$ represents the output result of the model, and the prediction ranking score $S$ can be classified as a positive value (+1) or a negative value (−1) on the top of a given threshold. If $y_i > 0.5$, then $S = +1$, otherwise $S = −1$. These mobile services with positive values will be selected and recommended to the target Mashup. The case in point is, if there are two active mobile service members $S_1$ and $S_2$, the selection of the target Mashup $M_1$, $S_1$ will be selected and recommended to $M_1$. This is because it has a higher prediction value, i.e., $y_2 > 0.92$. Furthermore, we will investigate the influences of top-$S$ and top-$M$ on recommendation performance in the Experiments section.

3. Experiments

3.1. Experiment Dataset and Settings. In this experiment, we firstly crawled 3929 real Mashups, 10648 services, and 12715 invocations between these Mashups and services from ProgrammableWeb. As for each Mashup or service, a pre-processing process is performed to obtain their standard description information. Secondly, we use the Word2vec tool to expand the description document of Mashup or service from English Wikipedia corpus published on April 2017 and obtain their word embeddings vector. More concretely, the gensim module in Python is applied to train the English Wikipedia corpus and produce its word embeddings vector, and Table 1 presents the special parameters of Word2vec. Finally, the trained English Wikipedia corpus is exploited to expand the description documents of Mashup and service. The most similar Top-$N$ words to the original word are identified and used as the extended words. For instance, the top similar 10 expanded words of the two words “Earth” and “Google” are shown in Table 2. All Mashups in the dataset are uniformly divided into 5 subsets, in which 1 subset is used as the testing set and other 4 subsets are integrated as a training set. A five-fold cross validation is conducted, and the results for each fold are summed up to obtain their mean value as the reported experiment results. As for the testing set, through randomly removing some score values from the matrix of Mashup service, we change the number of score values provided by the active Mashups as 10, 20, and 30 and call them as Given 10, Given 20, and Given 30, respectively. At the same time, the removed score values are exploited as the expected values. Similarly, as for the testing set, through randomly removing some score values, the Mashup-service matrix becomes more sparser with density 10%, 20%, and 30%, respectively.

3.2. Evaluation Metrics. Mean absolute error (MAE) refers to the expected value of the square of the difference between the observed value and the true value, which can evaluate the change degree of data [16]. Root-mean-squared error (RMSE) is the square root of the ratio of the square of the deviation between the observed value and the true value to
the observed times $N$ [16]. We adopt the MAE and RMSE as the evaluation metrics of Web APIs recommendation. The smaller the MAE and RMSE mean, the better the recommendation effect:

\[
\text{MAE} = \frac{1}{N} \sum_{ij} |r_{ij} - \hat{r}_{ij}|,\]

\[
\text{RMSE} = \sqrt{\frac{1}{N} \sum_{ij} (r_{ij} - \hat{r}_{ij})^2},
\]

where $N$ is the number of predicted score, $r_{ij}$ indicates the true score of Mashup $M_i$ to service $S_j$, and $\hat{r}_{ij}$ indicates the predicted score of $M_i$ to $S_j$.

### 3.3. Baseline Methods

We choose the below methods as baseline to compare them with our proposed approach:

(i) **SPCC.** Similar to IPCC [17], service-based utilizing Pearson correlation coefficient (SPCC) approach measure the similarities between mobile services and perform recommendation.

(ii) **MPCC.** Similar to UPCC [17], Mashups-based utilizing Pearson correlation coefficient (MPCC) approach measure the similarities between Mashups and perform recommendation.

(iii) **PMF.** In the collaborative filtering, probabilistic matrix factorization (PMF) is a very popular matrix factorization model [10]. The historical invocation record between Mashups and mobile services is denoted as a matrix $R = [r_{ij}]_{nk}$. If $r_{ij} = 1$, the mobile service is invoked by a Mashup is shown, otherwise $r_{ij} = 0$. The probability of the mobile service $S_j$ invoked by the Mashup $M_i$ can be predicted and represented as $\hat{r}_{ij} = S_j^T M_i$.

(iv) **LDA-FMs.** The topic probability distributions of description documents in Mashup and mobile service firstly are derived by the LDA model, and then they are trained via FMs to predict the probability distribution of mobile service invoked by Mashup and recommend mobile service with high quality. Besides the topic information, the cooccurrence and popularity of mobile service are exploited in the FM modelling.

(v) **HDP-FMs.** The prior work [18], which integrates HDP and FMs to recommend mobile service for target Mashup. The HDP model is applied to derive the topic probability distributions of description documents in Mashup and mobile service. Similarly, the topic information and the cooccurrence and popularity of mobile service are all used in the FM modelling.

(vi) **EHDP-FMs.** The proposed method in this paper is an extended work of the prior work [18]. It firstly
uses Word2vec tool to expand the document description of mobile service and Mashup from Wikipedia corpus. Then the HDP model is applied to derive the topic probability distributions of the extended document description of mobile service and Mashup. Finally, the FM is deployed to predict and recommend high-quality mobile service for Mashup.

3.4. Experimental Results

3.4.1. Recommendation Performance Comparison. To study the performance of mobile service recommendation, we compare our method with other five baseline methods. We select the optimal number of extended words for each original word in description documents of Mashup and service, to achieve the best recommendation result in our EHDP-FMs. A detailed investigation about it will be discussed in subsequent section. Table 3 reports the MAE and RMSE comparison of multiple recommendation methods, which show our EHDP-FMs greatly outperforms WPCC and MPCC, significantly surpasses PMF and LDA-FMs, and slightly exceeds HDP-FMs consistently. The reason for this is that, in the EHDP-FMs, (a) more useful words information can be obtained from the extended description content of Mashup and service, (b) more similar Mashups and similar services in topic distribution are identified by using HDP technology, and (c) FM models and trains those useful information (including similar Mashups and similar services, the cooccurrence and popularity of service) to achieve more accurate service probability score prediction. Furthermore, when the given score values increase from 10 to 30 and the density of training matrix rises from 10% to 30%, the MAE and RMSE in the EHDP-FMs definitely drop. That is to say, more score values and training matrix with higher sparsity mean better accuracy of recommendation.

3.4.2. Effect of the Number of Extended Words on Mobile Service Recommendation. The experiments investigate the effect of the number of extended words on mobile service recommendation in our proposed method. During the experiments, we set the number of extended words to 1, 3, 5, and 7 (respectively, denoted as EHDP-FMs-1, EHDP-FMs-3, EHDP-FMs-5, and EHDP-FMs-7), when training matrix density = 10%, and obtain their values of MAE and RMSE in Figures 4 and 5. The experimental results indicate the performance of EHDP-FMs-1 is the worst in all cases. This is because the extended description documents of Mashup and service are still short and the contained, useful information is less in it when only extending a word for each original word. We can see that the MAE and RMSE of EHDP-FMs-3 are the optimal and best in all cases. However, when the number of the extended words continues to increase from 5 to 7, the recommendation performance decreases. The reason for this is that too many extended words contain more other irrelevant syntax and semantics information, which maybe makes the HDP topic model fail to mine the latent topics accurately and therefore weakens the performance of service recommendation. Therefore, we select 3 extended words for each original words of description document of Mashup and service in our EHDP-FM method. The observations indicate it is very important to choose an appropriate number of extended words for mobile service recommendation.

3.4.3. HDP-FMs Performance vs. LDA-FMs Performance with Different Topic Numbers. In this experiment, we, respectively, set the number of topics as 3, 6, 12, and 24 for LDA-FMs and denote as LDA-FMs-3/6/12/24. The experimental results in Figures 6 and 7, respectively, show the MAE and RMSE values when the training matrix density is equal to 10%. We also observe that the performance of HDP-FMs is the best. At the same time, the MAE and RMSE of LDA-FMs-12 are close to that of HDP-FMs and surpassed those of LDA-FMs-3, LDA-FMs-6, and LDA-FMs-24. The observations prove that HDP-FM is better than LDA-FMs since it can automatically derive the optimal topic numbers instead of repeatedly training like LDA.

3.4.4. Impacts of Top-S and Top-M in HDP-FMs. We investigate the effects of top-S and top-M to mobile service recommendation in order to obtain their optimal values. The optimal values of top-M (top-S) for all similar top-S (top-M) services (Mashups) are obtained, i.e., S = 5 for all top-M similar Mashups and M = 10 for all top-S similar mobile services. Under the setting of training matrix density = 10% and given number = 30, the MAEs of HDP-FMs are presented in Figures 8 and 9. We can see that from Figure 8, the MAE of HDP-FMs constantly rises when S grows from 5 to 25. Figure 9 indicates the MAE of HDP-FMs runs up to its peak value when M = 10 and then continuously rises with the increasing or decreasing of M. The observations mean that it is very important to identify suitable values of S and M for the HDP-FM method.

4. Related Works

Service recommendation is a hot topic nowadays in service-oriented computing [19]. Traditional service recommendation solves the quality problem of Mashup services in order to realize high-quality service recommendation. The quality of a single service can facilitate recommendation showed by Picozzi et al. [20]. The quality attributes of Mashup components (APIs) and information quality in Mashups [21] is analyzed by Cappiello [22]. In addition, collaborative filtering (CF) technique is widely exploited in QoS-based service recommendation [16]. We can use it to measure the similarity of services or users, predict the missing QoS values on the basis of the QoS records of similar services or similar users, and recommend services to users.

The problems of the data sparsity and long tail bring about inaccurate and imperfect search results according to the results in references [23, 24]. To attack the problem, some researchers try to use matrix factorization technology to decompose historical QoS or Mashup service interactions to obtain service recommendations [25, 26]. A collaborative QoS prediction method is proposed, in which a matrix
factorization model of neighbourhood integration is designed to predict the QoS value of personalized Web services, by Zheng et al. [26]. A social awareness service recommendation method is proposed, in which the multidimensional social relationships among potential users, Mashups, topics, and services are depicted by the coupling matrix model, by Xu et al. [9]. These methods aim to transform Mashup-service rating matrix or QoS into a feature space matrix with lower dimensions and predict the probability of services invoked by Mashups or unknown QoS.

Considering that matrix factorization depends on abundant historical interaction records, recent work incorporates additional information into matrix factorization to obtain more accurate service recommendation [5, 10–12]. Among them, Ma et al. [11] integrates matrix factorization with geographic and social influence to recommend interest points. By using location information and QoS of Web services to cluster services and users, a personalized service recommendation is proposed by Chen et al. [12]. The historical invocation relationship between Mashups and services is studied to infer the implicit functional correlation between services, and the correlation is incorporated into the matrix factorization model to facilitate service recommendation by Yao et al. [10]. Collaborative topic regression is proposed by Liu and Fullia [5], which combines probabilistic topic modelling and probabilistic matrix factorization for service recommendation.

The existing methods based on matrix factorization undoubtedly improve the performance of service recommendation. At the same time, we observed that few of them recognized the historical invocation between services and

<table>
<thead>
<tr>
<th>Method</th>
<th>Matrix density = 10%</th>
<th>Matrix density = 20%</th>
<th>Matrix density = 30%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAE</td>
<td>RMSE</td>
<td>MAE</td>
</tr>
<tr>
<td>SPCC</td>
<td>0.4258</td>
<td>0.5643</td>
<td>0.4005</td>
</tr>
<tr>
<td>MPCC</td>
<td>0.4316</td>
<td>0.5701</td>
<td>0.4108</td>
</tr>
<tr>
<td>PMF</td>
<td>0.2417</td>
<td>0.3835</td>
<td>0.2263</td>
</tr>
<tr>
<td>LDA-FMs</td>
<td>0.2091</td>
<td>0.3225</td>
<td>0.1969</td>
</tr>
<tr>
<td>HDP-FMs</td>
<td>0.1547</td>
<td>0.2874</td>
<td>0.1329</td>
</tr>
<tr>
<td>EHDP-FMs</td>
<td>0.1308</td>
<td>0.2507</td>
<td>0.1154</td>
</tr>
</tbody>
</table>

| Method   | MAE      | RMSE     | MAE      | RMSE     | MAE      | RMSE     |
|----------|----------------------|----------------------|----------------------|
| SPCC     | 0.4135   | 0.5541   | 0.3918   | 0.5158   | 0.3890   | 0.5003   |
| MPCC     | 0.4413   | 0.5712   | 0.4221   | 0.5202   | 0.4151   | 0.5109   |
| PMF      | 0.2398   | 0.3599   | 0.2137   | 0.3427   | 0.1992   | 0.3348   |
| LDA-FMs  | 0.1989   | 0.3104   | 0.1907   | 0.3018   | 0.1801   | 0.2894   |
| HDP-FMs  | 0.1486   | 0.2713   | 0.1297   | 0.2513   | 0.1185   | 0.2291   |
| EHDP-FMs | 0.1227   | 0.2419   | 0.1055   | 0.2216   | 0.0952   | 0.1904   |

| Method   | MAE      | RMSE     | MAE      | RMSE     | MAE      | RMSE     |
|----------|----------------------|----------------------|----------------------|
| SPCC     | 0.4016   | 0.5447   | 0.3907   | 0.5107   | 0.3739   | 0.5012   |
| MPCC     | 0.4518   | 0.5771   | 0.4317   | 0.5159   | 0.4239   | 0.5226   |
| PMF      | 0.2214   | 0.3319   | 0.2081   | 0.3117   | 0.1986   | 0.3052   |
| LDA-FMs  | 0.1970   | 0.3096   | 0.1865   | 0.2993   | 0.1794   | 0.2759   |
| HDP-FMs  | 0.1377   | 0.2556   | 0.1109   | 0.2461   | 0.1047   | 0.2057   |
| EHDP-FMs | 0.1113   | 0.2248   | 0.0926   | 0.2057   | 0.0804   | 0.1673   |

**Table 3: MAE and RMSE comparison of multiple recommendation approaches.**

**Figure 4: MAE values of EHDP-FMs-1/3/5/7.**

**Figure 5: RMSE values of EHDP-FMs-1/3/5/7.**

8 Mobile Information Systems
Mashups to derive potential topics, and they did not use FMs to model and train these potential topics to predict the probability of Mashup calling services to obtain more accurate service recommendation. In our previous work [8, 27, 28], we mainly address on LDA or enhanced LDA topic model for Web services clustering [27, 28] and also exploit word embedding technique to enhance the accuracy of service clustering [8]. Driven by these methods, we combine FMs and word-embedded enhanced HDP for recommending mobile services to build novel Mashup application. We apply the HDP model to export the potential topics from the description documents of mobile services and Mashups to support FM model training. We use FMs to predict the probability of Mashups calling mobile services and recommend high-quality services for building novel Mashup application.

5. Discussion

Recommending mobile service to build novel Mashup application for software developers in the mobile service computing environment is becoming a promising research topic. In our paper, the functional semantic representation of Mashup applications and mobile services is fully consolidated and mined by extending their description documents and modelling their topic probability distribution, and the quality prediction is performed by exploiting FMs to train and model multiple dimension features of mobile service. The high-quality mobile services are ranked and recommended to build Mashup by simultaneously considering their functionality representation and quality feature. The accuracy of mobile service recommendation is significantly improved as a result.

Although the above approach and solution seem very effective in the Mashup development, it will be better if a
Mobile Information Systems

prototype can be designed or implemented to validate the effectiveness and application value of the approach. As we expected, the objective of the prototype system is to rank and recommend the high-quality mobile service to software developers for building Mashup application. We can use the tools of Python 3.5, Mysql 5.6, and the technologies of Flask and Pyecharts to develop the prototype system. It should achieve four basic function parts, i.e., service data crawling and preprocessing, description extension of Mashup and mobile service, topic modelling of Mashup and mobile service, and recommendation of mobile service for the given Mashup requirement. More concretely,

(1) In the first part (service data crawling and preprocessing), the system incrementally crawls Mashups, services, and invocations between these Mashups and services from ProgrammableWeb and builds their corresponding data table to store. Because the description documents of Mashup and mobile service contain some useless or meaningless terms/words, the preprocessing is performed to normalize and standardize the description information. The preprocessing mainly includes tokenization, words are segmented by spaces and punctuation is separated from words by using NLTK (Natural Language Toolkit) in Python; removing stop words (the common short words or symbols that have no practical meaning but occur frequently, such as the, to, a, an, with, and at), the stop vocabulary table in the NLTK is applied to remove stop words; stemming, various forms of a word are usually used in the grammatical expression, such as provide, providing, provides, and provided, and their common word endings, such as ing, s, and ed, should be removed.

(2) In the second part (description extension of Mashup and mobile service), English Wikipedia corpus trained by Word2vec based on the genism module in Python is exploited as the description extension source of Mashup and mobile service. The most similar Top-N words to an original word in the description documents of Mashup and mobile service are identified and saved as the extended words in the prototype system. Word2vec uses the hierarchical softmax algorithm to speed up and train word vector for English Wikipedia corpus, whose time complexity is \(O(\log N)\). Meanwhile, Word2vec calculates the similarity between words in English Wikipedia corpus and obtain the similarity matrix, whose time and space complexity are all \(O(n^2)\). \(n\) is the total amount of words. During the process of description extension, the most similar Top-N words to an original word can be found only by the way of look-up table in the trained English Wikipedia corpus, and its time complexity is \(O(1)\) and space complexity is \(O(N^2)\).

(3) In the third part (topic modelling of Mashup and mobile service), hierarchical Dirichlet process technology is used to extract the implicit topics of Mashup and mobile service, which clusters/groups service data according to the cooccurrence of word frequency. It can automatically determine the optimal number of topics, which avoids adjusting the number of topics repeatedly and so saves the time cost. It can also accurately predict the topic distribution of Mashup and mobile service, which do not need to retrain the dataset and make the prototype system real time. A topic modelling module can be designed in the prototype system, in which Flask framework and Pyecharts visualization tool in Python are used to present the effect of topic modelling, and a download function is provided to download the transformed topic vectors of Mashup and mobile service for users.

(4) In the fourth part (recommendation of mobile service for the given Mashup requirement), when a software developer submits a Mashup requirement, a recommendation of mobile services with good quality for software developer to build novel Mashup application. During the process of recommendation, factorization machines train and model the important input features. These input features include functional features, i.e., similar Mashups of target Mashup and similar mobile services of active mobile service derived from the topic similarity based on the HDP model, and quality features, i.e., the cooccurrence and popularity of mobile services obtained from the stored data table. FMs predict the probability of mobile services invoked by Mashups and recommends the high-quality mobile service for a Mashup creation. Similarly, Flask framework and Pyechart visualization tool in Python are used to present the effect of recommendation.

6. Conclusions and Future Work

This paper proposes a mobile service recommendation method for Mashup development in mobile service computing by combining word embeddings enhanced HDP and FMs. The experimental results on the top of ProgrammableWeb dataset show that compared with the existing recommendation methods, the proposed method achieves significant improvements in the accuracy of recommendation. In the future work, we will investigate and apply fine-grained service relationship information into the proposed model for more accurate recommendation.

Data Availability

The crawled dataset from ProgrammableWeb can be accessed at http://49.123.0.60:8080/MashupNetwork2.0/dataset.jsp.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.
Acknowledgments

The work was supported by the National Natural Science Foundation of China under grant nos. 6187316, 61872139, 61572187, 61772193, 61702181, and 61572371; National Key R&D Program of China under grant no. 2017YFB1400602; Hunan Provincial Natural Science Foundation of China under grant nos. 2017J2098, 2017J4036, 2018J2139, and 2018JJ2136; and Innovation Platform Open Foundation of Hunan Provincial Education Department of China under grant no. 17K033.

References
