

Research Article

Recommendation of Crowdsourcing Tasks Based on Word2vec Semantic Tags

Qingxian Pan ^{1,2}, Hongbin Dong ¹, Yingjie Wang,² Zhipeng Cai ³, and Lizong Zhang⁴

¹College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China

²School of Computer and Control Engineering, Yantai University, Yantai 264005, China

³Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA

⁴School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

Correspondence should be addressed to Hongbin Dong; donghongbin@hrbeu.edu.cn

Received 1 November 2018; Revised 18 February 2019; Accepted 3 March 2019; Published 24 March 2019

Guest Editor: Michele Nogueira

Copyright © 2019 Qingxian Pan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Crowdsourcing is the perfect show of collective intelligence, and the key of finishing perfectly the crowdsourcing task is to allocate the appropriate task to the appropriate worker. Now the most of crowdsourcing platforms select tasks through tasks search, but it is short of individual recommendation of tasks. Tag-semantic task recommendation model based on deep learning is proposed in the paper. In this paper, the similarity of word vectors is computed, and the semantic tags similar matrix database is established based on the Word2vec deep learning. The task recommending model is established based on semantic tags to achieve the individual recommendation of crowdsourcing tasks. Through computing the similarity of tags, the relevance between task and worker is obtained, which improves the robustness of task recommendation. Through conducting comparison experiments on Tianpeng web dataset, the effectiveness and applicability of the proposed model are verified.

1. Introduction

Deep learning was proposed by Geoffrey Hinton et al. in 2006. This method simulates human brain neural network to model and realize multiple level abstraction [1, 2]. In 2006, Jeff Howe of American Wired magazine reporter proposed crowdsourcing concept [3]. As a new kind of business model, crowdsourcing has been widespread concern in various fields and becomes the new hot point of computer research fields. Task requester, crowdsourcing platform, and worker make up crowdsourcing system [4]. The process of crowdsourcing includes designing task, publishing task, selecting task, sensing task, submitting solution, and integrating solution. Among them, task selection is the key phase in the process of crowdsourcing. This is the key to complete crowdsourcing task that the appropriate worker selects appropriate task in appropriate time [5].

The popular crowdsourcing platforms use task searching to get the favourite task by keyword searching [6]. However, with the rapid development of crowdsourcing, the problem of

information overload is more and more serious. In addition, it is more and more difficult to get the favourite crowdsourcing task for worker. Recommender system is an effective medium to solve the problem, which is used on many E-Commerce Platforms, such as Alibaba, Amazon, and Netflix [7]. But there are many problems which are not solved in recommender systems, such as similarity calculation, the lower recommended accuracy, data sparseness, and cold boot. In brief, improving the accuracy and reliability of recommender systems has been paid more attention by scholars.

However, individual recommendation research of the task is lesser in crowdsourcing, and task selection is relied on hobbies and expertise. Few crowdsourcing platforms can actively recommend task. This paper researches the crowdsourcing tasks recommendation model based on Word2vec semantic tags in order to achieve individual recommendation of crowdsourcing tasks [8].

The main contributions of this paper include following three contents:

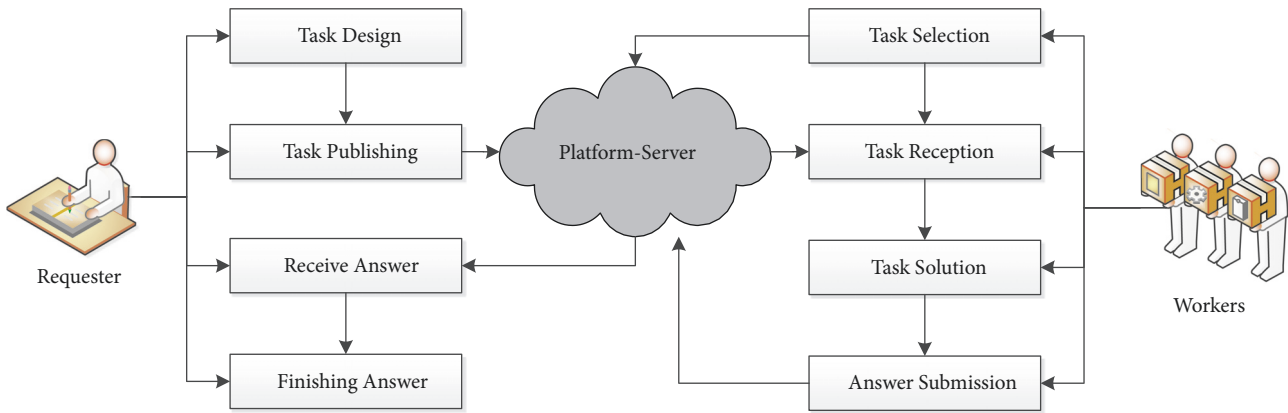


FIGURE 1: The workflow of crowdsourcing.

- (1) Compute the similarity of word vectors and build the semantic tags similar matrix database based on the Word2vec deep learning.
- (2) Research the task recommending model based on semantic tags to achieve the individual recommendation of crowdsourcing tasks. This paper computes similarity of tasks and workers based on the semantic tag similar matrix.
- (3) Utilizing the Tianpeng Web dataset, the experiments are conducted. The experimental results show that the model is feasible and effective. The model can be used in other fields according to the different semantic databases.

This paper is organized as follows. Section 2 reviews the related works. The Work2vec is discussed in Section 3. In addition, the tasks recommendation model and realization method based on semantic tags are researched in Section 4. The comparison experiments, as well as the analysis for the experimental results, are introduced in Section 5. The conclusion is presented in Section 6.

2. Related Works

In order to discuss the related works for recommendation of crowdsourcing, we, respectively, introduce the related works of crowdsourcing and recommendations.

2.1. Crowdsourcing. In 2006, Jeff Howe proposed crowdsourcing concept firstly [3]: a company or an institution outsources the tasks performed by an employee in the past to an unspecific public network in a free and voluntary manner. With the development of crowdsourcing technology, the different crowdsourcing concepts appeared. Chen et al. [9] summarized 40 different crowdsourcing definitions. Feng et al. [10] gave the definition of crowdsourcing according to the basic features of crowdsourcing. According to the definition, crowdsourcing is a distributed problem-solving mechanism opening to the Internet public, and it completes the tasks

that are difficult to complete by a computer through integrating computers and the unknown public on the Internet [11].

Crowdsourcing is successfully applied in language translation, image recognition, intelligent transportation, software development, entry interpretation, tourism photography, and other fields, which has become the perfect embodiment of group wisdom [12, 13]. Crowdsourcing is made up of the task requester, crowdsourcing platform, and workers. The crowdsourcing workflow includes designing tasks by task requester, publishing tasks, selecting tasks by workers, solving tasks, submitting answer, and arranging answer. The workflow of crowdsourcing is shown by Figure 1. The public participation is the basis of crowdsourcing. And the key to high-quality complete crowdsourcing tasks is to recommend appropriate tasks to appropriate worker in appropriate time [14].

2.2. Recommender Systems. With the arrival of big data era, the problem of information overload is more and more serious and that finding the useful and best information is more and more difficult. Recommender Systems is an effective medium to solve the above problems [15]. However, there are some inherent defects in recommendation systems, such as low accuracy, data sparseness, cold boot, the defects of the centralized system, similarity calculation, and being easy to be attacked. In addition, many recommender systems applied to business systems, whose purpose is to sell more goods and seek the maximum benefits, rather than to recommend the best commodities to users. In brief, the credibility and accuracy of recommendation systems need to be improved, which has attracted the attention of scholars. Yang et al. [16] proposed a recommender system based on transfer learning. Chen et al. [17] proposed a recommender system based on bind context. Tang et al. [18] researched recommender system based on crossing knowledge. Liu [19] and Zhou et al. [20] researched recommender systems for social recommendation. Combining Markov and social attributes of users, Wang et al. [21] proposed a probability-based recommendation model to recommend items for users.

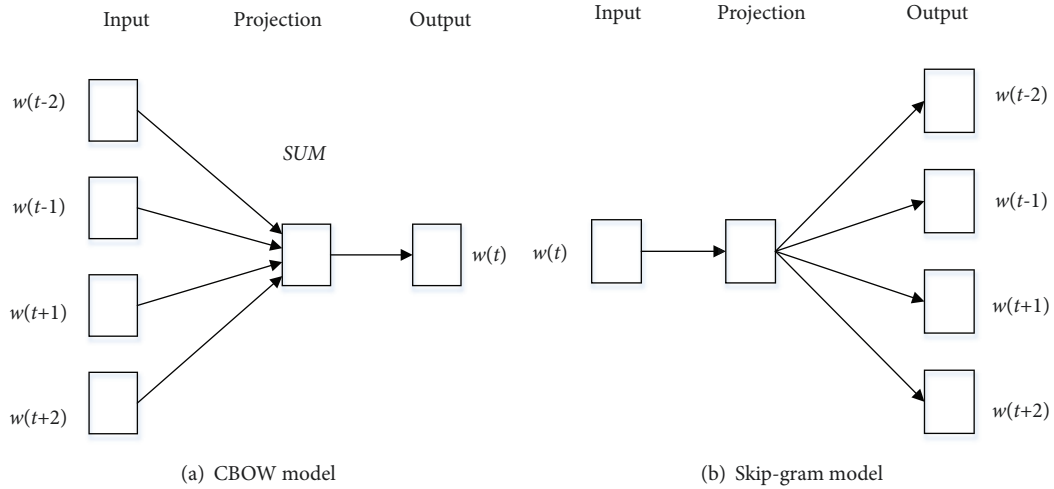


FIGURE 2: CBOw model and Skip-gram model.

Crowdsourcing task recommendation is mainly from the perspective of crowdsourcing platform. Based on the task discovery model, crowdsourcing platform recommends related tasks according to the preferences of workers [5]. The main crowdsourcing platforms basically adopt the way of task search and rarely adopt the method of task recommendation [22]. Some task recommendation methods were researched based on traditional recommendation methods, including content-based recommendation, collaborative filtering, and mixed recommendation algorithms. Ambati et al. [23] proposed the use of task and workers' historical information for task recommendation. Yuen et al. [24] proposed a worker-task recommendation model through combining the historical information of workers and browsing history. Deng et al. [25] researched the problem of maximizing task selection for spatiotemporal tasks.

3. Word2vec

In 2003, Bengio et al. [26] proposed Neural Network Language Model-NNLM based on 3 levels. NNLM is used to compute the probability $p(w_t = i \mid \text{context})$ of the next word w_t of a context, and word vector is the byproduct during training. Word2vec is a tool based on deep learning to compute the similarity of word vector which was proposed by Google company in 2013 [27]. It converts the word into word vector and computes similarity according to the cosine between word vectors. When using the tool, the texts after segmentation are input, and the output-word vector can be used to do a lot of Natural Language Processing (NLP) related work, such as clustering, looking for synonyms, and part of speech analysis.

Word2vec uses word vector presentation mode based on Distributed representation. Distributed representation is proposed by Hinton in 1986 [28]. Its basic thought is to map each word into a k -dimension real vector by training (k is a hyperparameter in the model) and to judge the semantic similarity between them according to the distance between

words (such as cosine similarity, Euclidean distance). It uses a '3 layers neural network', input layer-hidden layer-output layer. Its core technology is to use Huffman code according to word frequency, which makes the activated content basically consistent of all word frequency similar words in hidden layer. The higher the frequency of the word, the less the number of hidden layers they activate, which effectively reduces the computational complexity.

Compared with Latent Semantic Index-LSI and Latent Dirichlet Allocation-LDA, Word2vec uses the context of words and makes the semantic information richer. There are two kinds of training model-CBOw (Continuous Bag-of-Words) and Skip-gram in Word2vec, which are shown by Figure 2. Two models both include input layer, projection layer, and output layer. CBOw model predicts the current words according to the known context, and Skip-gram model predicts context according to the current words.

In this paper, the objective optimization function of CBOw is expressed by

$$p(w \mid \text{Context}(w)) = \prod_{j=2}^{l^w} p(d_j^w \mid x_w, \theta_{j-1}^w) \quad (1)$$

where x_w means the word vector of the root node in the Hoffman tree, $\text{Context}(w)$ represents the context of word w , that is, the collection of w peripheral words, l^w represents the nodes number of the path p^w , and $d_j^w \in \{0, 1\}$ represents Huffman code of the word w ; $\theta_1^w, \theta_2^w, \dots, \theta_{j-1}^w \in R^m$ represents the vectors corresponding to nonleaf nodes of the path p^w . Therefore, the logistic regression probability $p(d_j^w \mid x_w, \theta_{j-1}^w)$ that w passes a node j in the Hoffman tree is shown by (2). The corresponding parameter $\sigma(x_w^T \theta_{j-1}^w)$ is shown by (3).

$$p(d_j^w \mid x_w, \theta_{j-1}^w) = \begin{cases} \sigma(x_w^T \theta_{j-1}^w), & d_j^w = 0; \\ 1 - \sigma(x_w^T \theta_{j-1}^w), & d_j^w = 1. \end{cases} \quad (2)$$

$$\sigma(x_w^T \theta_{j-1}^w) = \frac{1}{1 + e^{-x_w^T \theta_{j-1}^w}} \quad (3)$$

In order to clearly represent the meaning of logistic regression probability $p(d_j^w | x_w, \theta_{j-1}^w)$, we combine (2) and (3) to obtain the value of $p(d_j^w | x_w, \theta_{j-1}^w)$, which is shown by

$$p(d_j^w | x_w, \theta_{j-1}^w) = [\sigma(x_w^T \theta_{j-1}^w)]^{1-d_j^w} \cdot [1 - \sigma(x_w^T \theta_{j-1}^w)]^{d_j^w} \quad (4)$$

For avoiding the value of $p(w | \text{Context}(w))$ too small, logarithm Likelihood function is used to represent the objective function; thus, (1) can be converted into

$$L = \sum_{w \in C} \log p(w | \text{Context}(w)) \quad (5)$$

Through combining (4) and (5), the objective function L is shown by

$$\begin{aligned} L &= \sum_{w \in C} \log \prod_{j=2}^w \left\{ [\sigma(x_w^T \theta_{j-1}^w)]^{1-d_j^w} \cdot [1 - \sigma(x_w^T \theta_{j-1}^w)]^{d_j^w} \right\} \\ &= \sum_{w \in C} \sum_{j=2}^w \left\{ (1-d_j^w) \cdot \log [\sigma(x_w^T \theta_{j-1}^w)] + d_j^w \cdot \log [1 - \sigma(x_w^T \theta_{j-1}^w)] \right\} \end{aligned} \quad (6)$$

Therefore, (6) is the object function of CBOW in this paper. Word2vec uses random gradient ascent method to optimize the object function of CBOW.

4. The Tasks Recommendation Model and Realization Method Based on Semantic Tags

4.1. Basic Model Frame and Mathematical Computation Model. The results and discussion may be presented separately, or in one combined section, and may optionally be divided into headed subsections.

The core of the model is the research of tag similar matrix. The model uses tag similar matrix to compute the similarity of workers and tasks, produces worker-tag similar matrix, and realizes tasks recommendation or workers recommendation. In model, tag similar matrix is obtained by Word2vec computing. Worker-tag matrix is got according to history work information of the worker, registration information, etc. And task-tag matrix is got according to task description, task classification, etc.

Define tag similar matrix $L \in R^{m \times m}$, $\begin{bmatrix} l_{11} & \dots & l_{1m} \\ \vdots & \ddots & \vdots \\ l_{m1} & \dots & l_{mm} \end{bmatrix}$, L is a symmetric matrix, that is, $l_{ij} = l_{ji}$, l_{ij} represents the similarity of tag i and tag j , $l_{ij} \in [0, 1]$, and its value is got through using Word2vec tool to compute. Define worker-tag matrix $W \in R^{n \times m}$, $\begin{bmatrix} w_{11} & \dots & w_{1m} \\ \vdots & \ddots & \vdots \\ w_{n1} & \dots & w_{nm} \end{bmatrix}$, and, among them, $w_{ij} = \{1, \text{worker } i \text{ has tag } j; 0, \text{worker } i \text{ has not tag } j\}$.

We define the task-tag matrix $T \in R^{p \times m}$, $\begin{bmatrix} t_{11} & \dots & t_{1m} \\ \vdots & \ddots & \vdots \\ t_{p1} & \dots & t_{pm} \end{bmatrix}$, and, among them, $t_{ij} = \{1, \text{task } i \text{ has tag } j; 0, \text{task } i \text{ has not tag } j\}$.

Therefore, the worker-task similar matrix W_T is obtained by (7), where W is the worker-tag matrix, L is the tag similar matrix, and T^T means the task-tag transposed matrix. Through (7), the relationship between workers and tasks can be obtained.

$$\begin{aligned} W_T &= W \times L \times T^T \\ &= \begin{bmatrix} w_{11} & \dots & w_{1m} \\ \vdots & \ddots & \vdots \\ w_{n1} & \dots & w_{nm} \end{bmatrix} \times \begin{bmatrix} l_{11} & \dots & l_{1m} \\ \vdots & \ddots & \vdots \\ l_{m1} & \dots & l_{mm} \end{bmatrix} \\ &\quad \times \begin{bmatrix} t_{11} & \dots & t_{1m} \\ \vdots & \ddots & \vdots \\ t_{p1} & \dots & t_{pm} \end{bmatrix}^T \end{aligned} \quad (7)$$

4.2. Basic Flow. The main steps of the process of the proposed recommendation model are shown as follows: (1) compute the word vectors based on Word2vec; (2) computing the similarity of word vectors; (3) generating the tag similar matrix; (4) obtaining the worker-tag matrix and task-tag matrix; (5) computing the worker-task similarity matrix; (6) L^2 standardization and normalization; (7) tasks and workers recommendation. Tag similar matrix generation uses Word2vec tool. Worker-task similarity computation uses mathematical methods introduced in the previous section. The section mainly introduces standardization and normalization method.

L^2 standardization method: the L^2 norm definition of vector $x(x_1, x_2, \dots, x_n)$ is shown as follows: $\text{norm}(x) = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$.

In order to make x normalized to the unit L^2 norm, the mapping between x and x' is established, so that the L^2 norm of x' is 1, and the proof is shown as follows:

$$\begin{aligned} 1 &= \text{norm}(x') = \frac{\sqrt{x_1^2 + x_2^2 + \dots + x_n^2}}{\text{norm}(x)} \\ &= \sqrt{x_1'^2 + x_2'^2 + \dots + x_n'^2} \\ &= \sqrt{\left(\frac{x_1}{\text{norm}(x)}\right)^2 + \left(\frac{x_2}{\text{norm}(x)}\right)^2 + \dots + \left(\frac{x_n}{\text{norm}(x)}\right)^2} \end{aligned} \quad (8)$$

where the value of x'_i is shown by

$$x'_i = \frac{x_i}{\text{norm}(x)} \quad (9)$$

In order to get the standardization and generality of data, the standardization data of L^2 is normalized, so that the data fall in the interval $[0, 1]$, the conversion formula is shown by (10), where $\min(X)$ means the minimum in X , and $\max(X)$ is the maximum in X .

$$x'_i = \frac{x_i - \min(X)}{\max(X) - \min(X)} \quad (10)$$

TABLE 1: Word2vec parameter setting.

Parameter	Value	Parameter	Value
window	8	hs	1
size	100	cbow	yes
threads	20	alpha	0.001
binary	0	negative	25

TABLE 2: Tag similar matrix L of simulation dataset.

	L1	L2	L3	L4	L5	L6	L7	...
L1	1.000	0.407	0.124	0.119	0.126	0.434	0.075	...
L2	0.407	1.000	0.766	0.917	0.993	0.642	0.546	...
L3	0.124	0.766	1.000	0.930	0.477	0.526	0.744	...
L4	0.119	0.917	0.930	1.000	0.909	0.531	0.394	...
L5	0.126	0.993	0.477	0.909	1.000	0.636	0.860	...
L6	0.434	0.642	0.526	0.531	0.636	1.000	0.166	...
L7	0.075	0.546	0.744	0.394	0.860	0.166	1.000	...
...

TABLE 3: Worker-tag matrix W.

	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	...
W1	0	0	0	1	0	0	0	0	0	0	0	...
W2	0	0	0	0	0	0	0	1	0	0	0	...
W3	0	1	0	0	1	0	1	0	0	1	0	...
W4	1	0	0	0	0	0	0	0	1	0	0	...
W5	0	0	0	0	0	1	0	0	0	0	0	...
...

5. Experiment and Simulation

In this section, we conduct the comparison experiments on the simulation dataset and real dataset, respectively. The real dataset is the dataset crawled from Tianpeng web site.

In the experiment, text8 is corpora training set, and experimental environment is Intel Core (TM) i5-337U CPU @1.8GHz dual-core, and 8GB memory.

5.1. The Experiments Conducted on Simulation Dataset. In this group of comparison experiments, the training parameters are shown in Table 1.

In addition, the tag similar matrix after training is shown in Table 2. In the matrix, the elements indicate the similarities between tags.

In this group of experiments, there are 100 workers, 50 tasks, 2000 tags in the experiment. The worker-tag matrix is generated randomly, which is shown in Table 3. The elements in Table 3 represent the similarities between workers and tags. The task-tag matrix is shown in Table 4. The elements in Table 4 indicate the similarities between tasks and tags. After computing the worker-task matrix, the standardization and normalization of worker-task matrix are shown in Table 5. The elements in Table 5 mean the similarities between workers and tasks.

TABLE 4: Task-tag matrix T.

	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	...
T1	0	0	0	0	0	0	0	1	0	0	0	...
T2	0	0	1	0	0	0	1	0	1	0	0	...
T3	0	0	0	0	0	0	0	0	0	0	0	...
T4	0	0	1	0	1	0	0	0	0	0	0	...
T5	0	0	0	0	0	0	1	0	0	0	0	...
...

TABLE 5: Worker-task similar matrix.

	T1	T2	T3	T4	T5	T6	T7	...
U1	0.754	0.410	0.369	0.438	0.365	0.420	0.396	...
U2	0.680	0.694	0.712	0.706	0.682	0.747	0.720	...
U3	0.387	0.378	0.407	0.403	0.385	0.678	0.351	...
U4	0.405	0.304	0.681	0.731	0.733	0.835	0.704	...
U5	0.279	0.278	0.696	0.284	0.294	0.265	0.324	...
...

Recall, precision, and F-measure are commonly used evaluation indexes [29]. The computing methods for the three evaluation indexes are shown by (11), (12), and (13). According to (11), (12), and (13), it can be seen that F-measure index is the comprehensive measure index through considering both recall and precision.

Recall

$$= \frac{\text{the quantity of related information retrieved}}{\text{the quantity of related information in system}} \quad (11)$$

Precision

$$= \frac{\text{the quantity of related information retrieved}}{\text{the quantity of all information retrieved}} \quad (12)$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (13)$$

The threshold values are 0.55, 0.6, and 0.65, respectively, and the recall, precision, and F-measure of the 50 tasks are obtained. The comparison experimental results on recall, precision, and F-measure indexes are shown by Figures 3, 4, and 5, respectively. In these experiments, x-coordinate indicates the Task-tag matrix T, and y-coordinates are recall rate, precision rate, and F-measure rate, respectively. From the experimental results, it can be seen that *threshold=0.6* has better performance than other two thresholds comprehensively.

In addition, we compare the proposed method with the method of tasks research. The experimental result is shown in Figure 6, where x-coordinate indicates the Task-tag matrix T and y-coordinate means the number of workers. The method used in this paper is better than the method used in tasks research, which proves the effectiveness of the method of this paper. In addition, the potential workers can be found

TABLE 6: Tag similar matrix L of Tianpeng dataset.

	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	...
L1	1.000	0.004	-0.041	0.100	0.018	0.048	0.020	-0.040	-0.049	-0.029	-0.009	0.038	-0.026	...
L2	0.004	1.000	0.803	0.261	0.882	0.225	0.493	0.610	0.391	0.315	0.817	0.666	0.601	...
L3	-0.041	0.803	1.000	0.231	0.761	0.166	0.393	0.533	0.351	0.259	0.722	0.603	0.609	...
L4	0.100	0.261	0.231	1.000	0.268	0.134	0.234	0.176	0.191	0.229	0.248	0.251	0.173	...
L5	0.018	0.882	0.761	0.268	1.000	0.218	0.475	0.571	0.352	0.228	0.753	0.659	0.583	...
L6	0.048	0.225	0.166	0.134	0.218	1.000	0.133	0.135	0.101	0.095	0.198	0.222	0.181	...
L7	0.020	0.493	0.393	0.234	0.475	0.133	1.000	0.334	0.190	0.192	0.504	0.459	0.296	...
L8	-0.040	0.610	0.533	0.176	0.571	0.135	0.334	1.000	0.295	0.258	0.556	0.480	0.515	...
L9	-0.049	0.391	0.351	0.191	0.352	0.101	0.190	0.295	1.000	0.248	0.386	0.239	0.277	...
L10	-0.029	0.315	0.259	0.229	0.228	0.095	0.192	0.258	0.248	1.000	0.238	0.288	0.236	...
L11	-0.009	0.817	0.722	0.248	0.753	0.198	0.504	0.556	0.386	0.238	1.000	0.616	0.535	...
L12	0.038	0.666	0.603	0.251	0.659	0.222	0.459	0.480	0.239	0.288	0.616	1.000	0.455	...
L13	-0.026	0.601	0.609	0.173	0.583	0.181	0.296	0.515	0.277	0.236	0.535	0.455	1.000	...
...

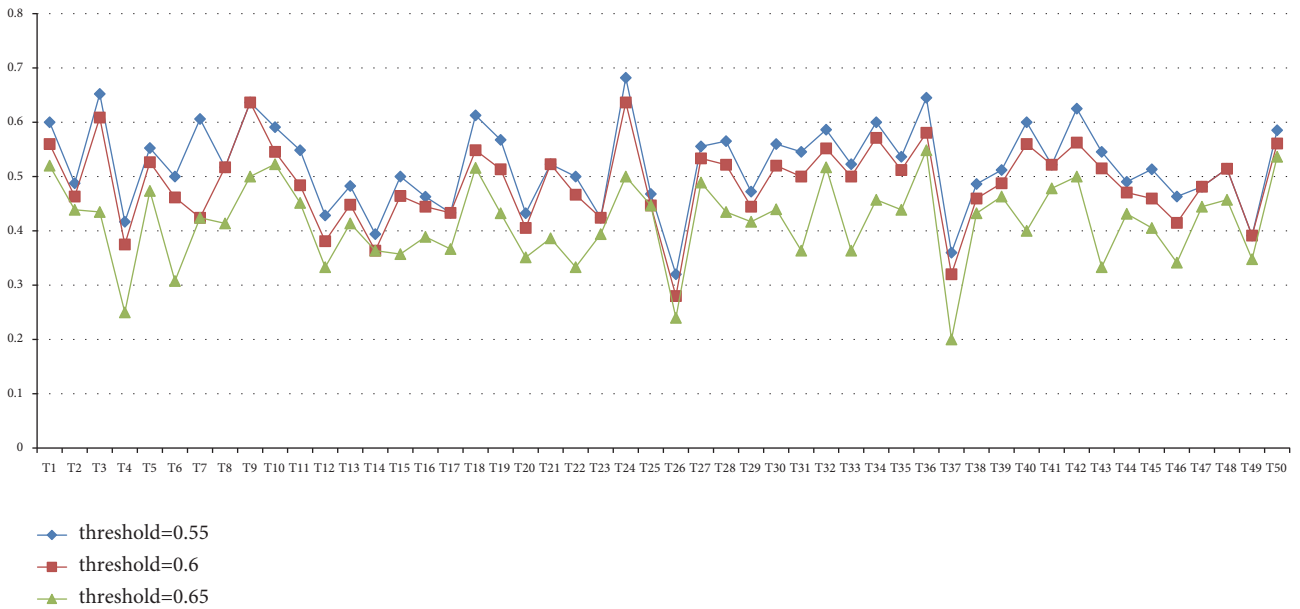


FIGURE 3: Recall of different thresholds.

by lowering the threshold, which can be used to analyze the potential users.

5.2. The Experiments Conducted on Tianpeng Dataset. The data collected from the Tianpeng web site were collected to form a corpus for training, and the tag similarity matrix was obtained as shown in the Table 6.

We select 510 workers and 371 tasks from Tianpeng dataset as experimental objects. Utilizing the dataset, we conduct the comparison experiments to verify the effectiveness of the proposed model. In the comparison experiments, 0.6 is taken as the threshold, and 20 tasks are randomly selected as recommended objects. The experimental results were compared with binary map matching and greedy algorithm

in terms of recall rate, accuracy rate, and F-value measure indexes.

According to the recall measure index, the comparison experimental result is shown by Figure 7. The x-coordinate indicates the Task-tag matrix T, and y-coordinate presents the recall rate. From the experimental result, it can be seen that the proposed recommendation model has the best performance on recall rate through compared with greedy algorithm and bipartite graph matching. In addition, the proposed recommendation model has better stability with the changing of T.

Figure 8 shows the experimental result on precision rate. Similarly, the x-coordinate indicates the Task-tag matrix T, and y-coordinate means the precision rate. In experimental

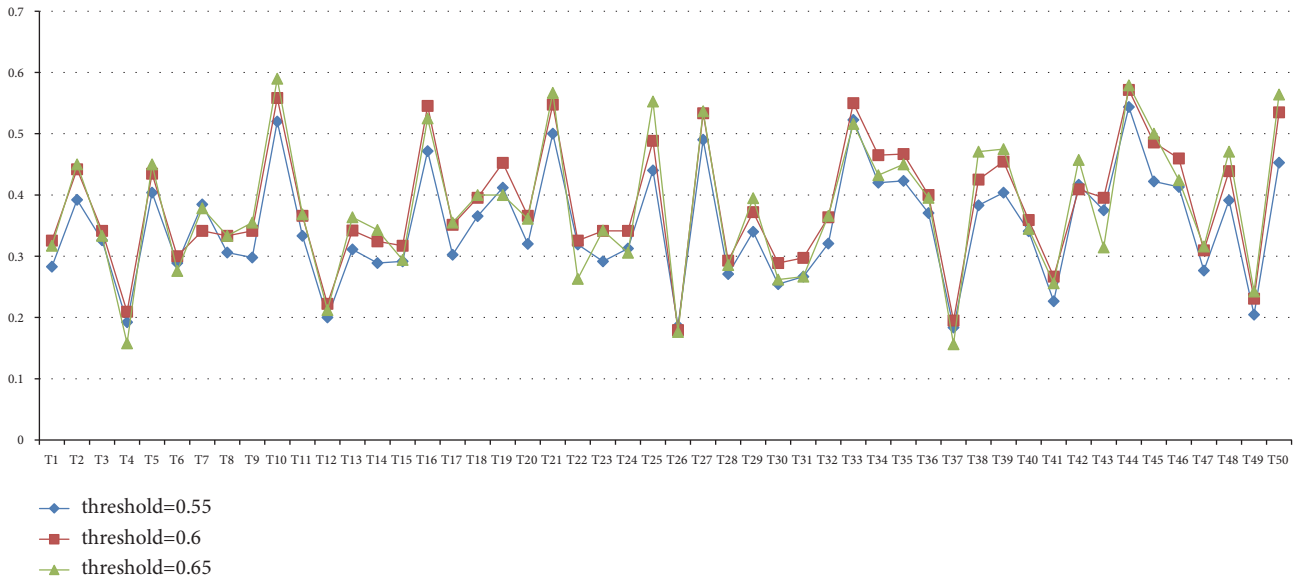


FIGURE 4: Precision of different thresholds.

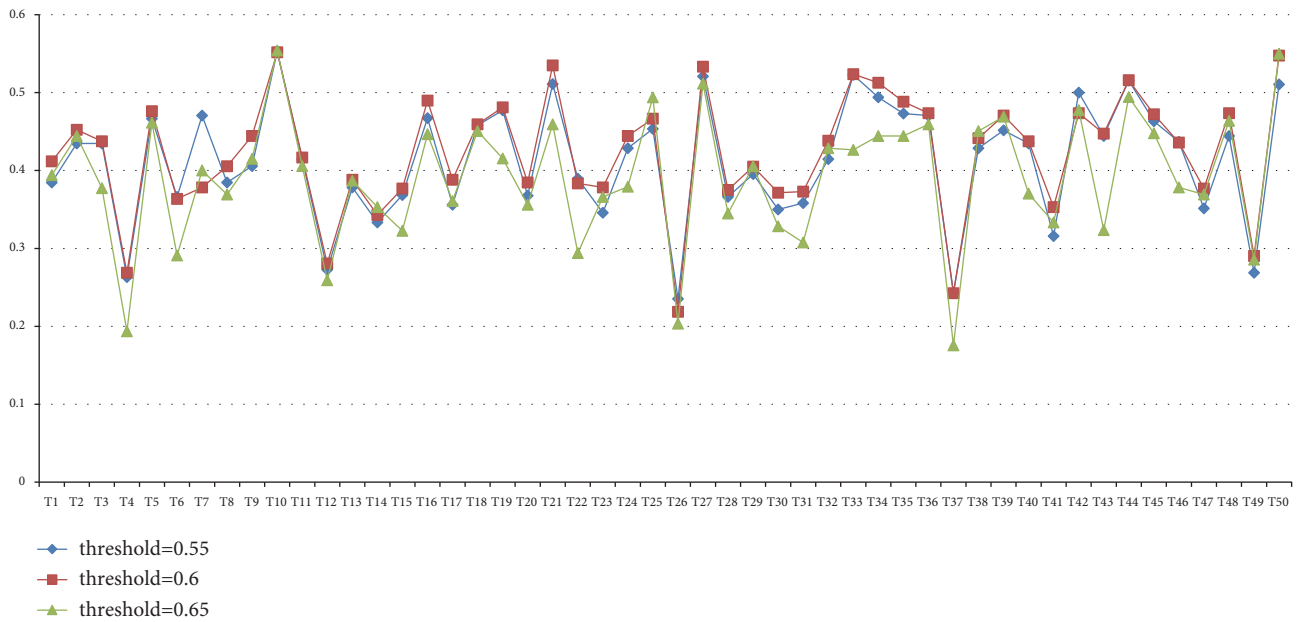


FIGURE 5: F-measure of different thresholds.

result, the average precision rate of the proposed recommendation is better than other two algorithms. From Figure 7, it can be seen that the proposed recommendation has the best performance on precision rate through compared with greedy algorithm and bipartite graph matching.

According to the experimental result on F-measure shown by Figure 9, we can see that the proposed recommendation also has the best performance on F-measure. In addition, F-measure index is the comprehensive measure index through considering both recall and precision. Therefore, we can infer that the proposed recommendation has the best

performance through compared with greedy algorithm and bipartite graph matching algorithm.

Through the comparison shows that the proposed methods than the binary map matching method, greedy algorithm in the recall, F-measure index significantly, in terms of accuracy with high and low, because to make the task would be able to complete the task of recommended for workers as much as possible, including the potential of workers, so the accuracy index can be put lower in the recommended requirements. It can be seen that the method proposed in this paper has higher practical significance and application value.

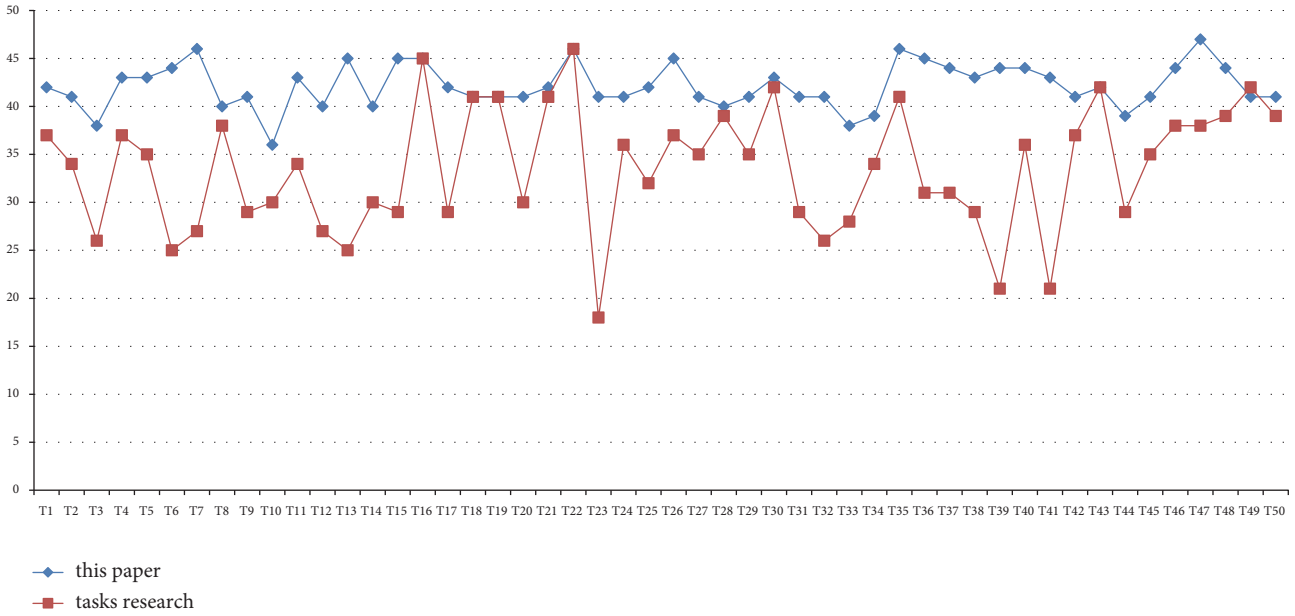


FIGURE 6: Comparison of experimental results.

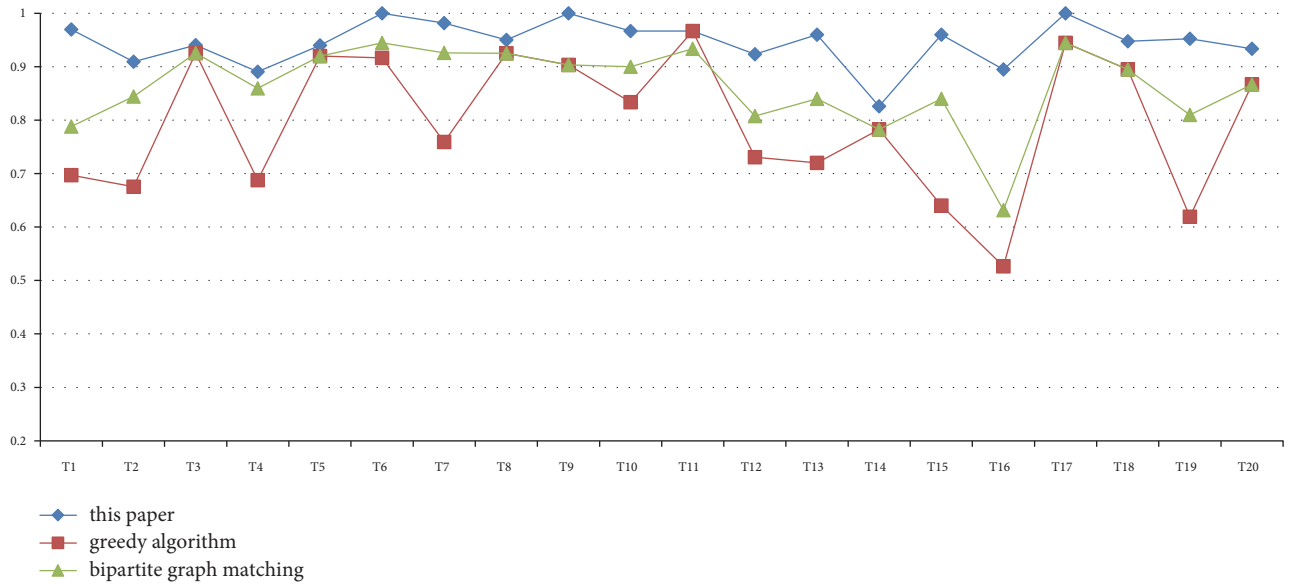


FIGURE 7: Recall of different methods.

6. Conclusion

Crowdsourcing is the perfect shown of group wisdom. It was applied in many fields as a new business model. In recent years, it has become the new hot research in computer science. The success key of crowdsourcing is to recommend task to appropriate worker. The recommendation method based on tag similar matrix is proposed in this paper. The method uses Word2vec technology to generate tag similar matrix and then computes the similarity of worker and task. According to the comparison experiments, it proves that

the method is effective and feasible. The recommendation method can be extended to other fields with the different corpora.

Because the success key of crowdsourcing is the participate rate of workers, it has become a hot topic in crowdsourcing research, such as reputation mechanism, preference evolution, and privacy protection of workers. It will be the focus of future research to improve the accuracy of recommender systems by combining recommender systems with reputation, preference evolution and historical information.

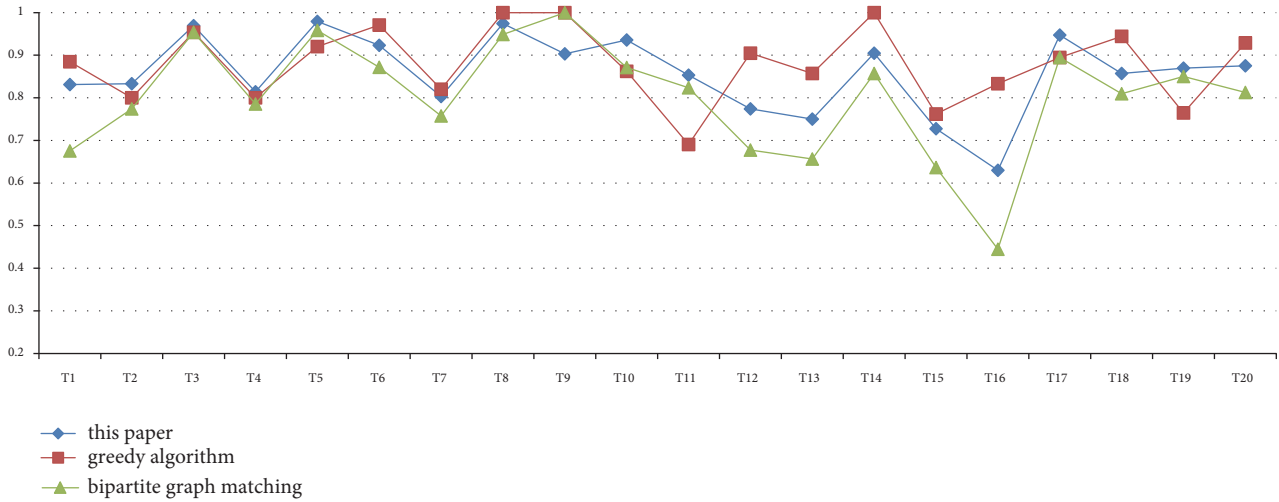


FIGURE 8: Precision of different methods.

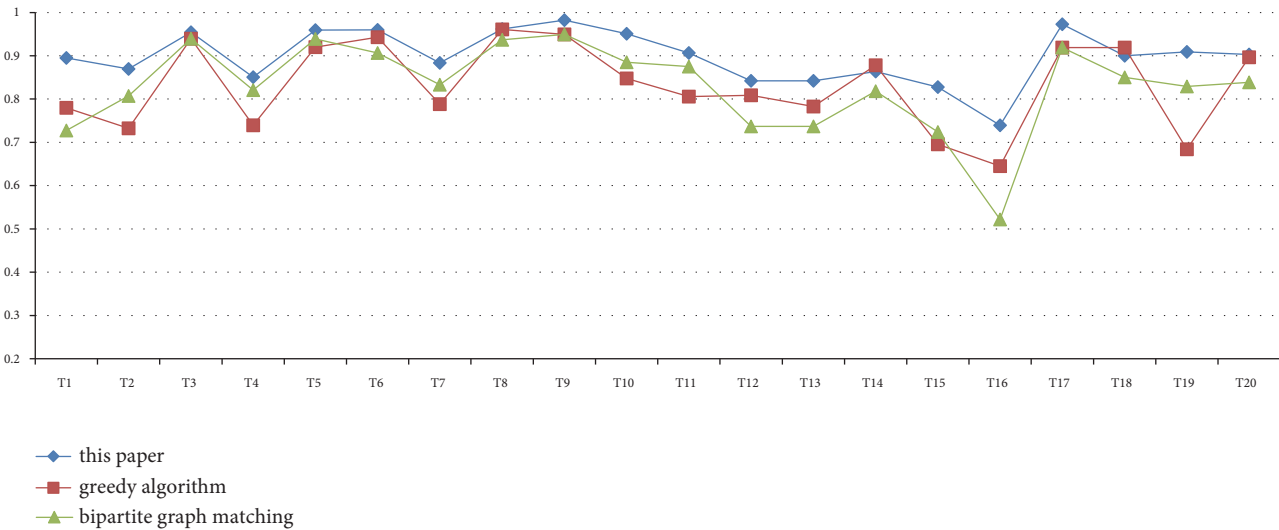


FIGURE 9: F-measure of different methods.

Data Availability

The [Tianpeng] dataset used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

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

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants No. 61472095, No. 61502410, and No. 61572418, the China Postdoctoral Science Foundation under Grant No. 2017M622691, the National Science Foundation (NSF) under Grants No. 1704287, No. 1252292,

and No. 1741277, and the Natural Science Foundation of Sichuan Province under Grant No. 2018HH0075.

References

- [1] Y. Cun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [2] Y. Wang, Z. Cai, G. Yin, Y. Gao, X. Tong, and G. Wu, “An incentive mechanism with privacy protection in mobile crowdsourcing systems,” *Computer Networks*, vol. 102, pp. 157–171, 2016.
- [3] J. Howe, “The rise of crowdsourcing,” *Wired Magazine*, vol. 14, no. 6, pp. 1–4, 2006.
- [4] Z. Cai and X. Zheng, “A private and efficient mechanism for data uploading in smart cyber-physical systems,” *IEEE Transactions on Network Science and Engineering*, p. 1, 2018.
- [5] Y. Hu, Y. Wang, Y. Li, and X. Tong, “An incentive mechanism based on multi-attribute reverse auction in mobile crowdsourcing,” *Sensors*, vol. 18, no. 10, p. 3453, 2018.

- [6] J. Li, Z. Cai, J. Wang, M. Han, and Y. Li, "Truthful incentive mechanisms for geographical position conflicting mobile crowdsensing systems," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 2, pp. 324–334, 2018.
- [7] R. Katarya and O. P. Verma, "Recent developments in affective recommender systems," *Physica A: Statistical Mechanics and its Applications*, vol. 461, pp. 182–190, 2016.
- [8] K. W. Church, "Emerging trends: Word2Vec," *Natural Language Engineering*, vol. 23, no. 1, pp. 155–162, 2017.
- [9] X. Chen, P. N. Bennett, K. Collins-Thompson, and E. Horvitz, "Pairwise ranking aggregation in a crowdsourced setting," in *Proceedings of the Sixth ACM International Conference*, pp. 193–202, Rome, Italy, February 2013.
- [10] J. Feng, G. Li, and J. Feng, "A survey on crowdsourcing," *Chinese Journal of Computers*, vol. 38, pp. 1713–1726, 2015.
- [11] Z. Duan, W. Li, and Z. Cai, "Distributed auctions for task assignment and scheduling in mobile crowdsensing systems," in *Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 635–644, GA, USA, June 2017.
- [12] Y. Wang, Z. Cai, X. Tong, Y. Gao, and G. Yin, "Truthful incentive mechanism with location privacy-preserving for mobile crowdsourcing systems," *Computer Networks*, vol. 135, pp. 32–43, 2018.
- [13] Y. Wang, Y. Li, Z. Chi, and X. Tong, "The truthful evolution and incentive for large-scale mobile crowd sensing networks," *IEEE Access*, vol. 6, pp. 51187–51199, 2018.
- [14] J. L. Cai, M. Yan, and Y. Li, "Using crowdsourced data in location-based social networks to explore influence maximization," in *Proceedings of the 35th Annual IEEE International Conference on Computer Communications*, 2016.
- [15] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997.
- [16] W. Pan and Q. Yang, "Transfer learning in heterogeneous collaborative filtering domains," *Artificial Intelligence*, vol. 197, pp. 39–55, 2013.
- [17] G. Chen and L. Chen, "Recommendation based on contextual opinions," *UMAP 2014, LNCS 8538*, pp. 61–73, 2014.
- [18] L. Liu, J. Tang, J. Han, and S. Yang, "Learning influence from heterogeneous social networks," *Data Mining and Knowledge Discovery*, vol. 25, no. 3, pp. 511–544, 2012.
- [19] J. Tang, X. Hu, and H. Liu, "Social recommendation: a review," *Social Network Analysis and Mining*, vol. 3, no. 4, pp. 1113–1133, 2013.
- [20] L. Lü, M. Medo, C. H. Yeung, Y. Zhang, Z. Zhang, and T. Zhou, "Recommender systems," *Physics Reports*, vol. 519, no. 1, pp. 1–49, 2012.
- [21] Y. Wang, G. Yin, Z. Cai, Y. Dong, and H. Dong, "A trust-based probabilistic recommendation model for social networks," *Journal of Network and Computer Applications*, vol. 55, pp. 59–67, 2015.
- [22] L. Zhang, Z. Cai, and X. Wang, "FakeMask: a novel privacy preserving approach for smartphones," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 335–348, 2016.
- [23] V. Ambati, S. Vogel, and J. Carbonell, "Towards task recommendation in micro-task markets," in *Proceedings of the 25th AAAI Workshop in Human Computation*, pp. 80–83, CA, USA, 2011.
- [24] M. C. Yuen, I. King, and K. S. Leung, "Probabilistic matrix factorization in task recommendation in crowdsourcing systems," in *Proceedings of the 19th International Conference on Neural Information Processing*, pp. 516–525, Springer, Doha, Qatar, 2012.
- [25] D. Deng, C. Shahabi, and U. Demiryurek, "Maximizing the number of worker's self-selected tasks in spatial crowdsourcing," in *Proceedings of the 21st ACM SIGSPATIAL International Conference*, pp. 1–10, FL, USA, November 2013.
- [26] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: a simple and general method for semi-supervised learning," in *Proceedings of the 8th Annual Meeting of the Association for Computational Linguistics*, pp. 384–394, Uppsala, Sweden, July 2010.
- [27] Y. Yao, X. Li, X. Liu et al., "Sensing spatial distribution of urban land use by integrating points-of-interest and Google Word2Vec model," *International Journal of Geographical Information Science*, vol. 31, no. 4, pp. 825–848, 2017.
- [28] R. Wang, H. Zhao, B.-L. Lu, M. Utiyama, and E. Sumita, "Bilingual continuous-space language model growing for statistical machine translation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 23, no. 7, pp. 1209–1220, 2015.
- [29] L. Li, G. Liu, and Q. Liu, "Advancing iterative quantization hashing using isotropic prior," in *Proceedings of the International Conference on Multimedia Modelling*, pp. 174–184, Springer International Publishing, 2016.



Hindawi

Submit your manuscripts at
www.hindawi.com

