Hindawi
Complexity
Volume 2020, Article ID 1670483, 10 pages
https://doi.org/10.1155/2020/1670483

Research Article

Labelling Training Samples Using Crowdsourcing Annotation for Recommendation

Qingren Wang[1], **Min Zhang**[1], **Tao Tao**[2] **and Victor S. Sheng**[3]

[1]*Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230601, China*
[2]*School of Computer Science and Technology, Anhui University of Technology, Ma'anshan 243002, China*
[3]*Department of Computer Science, Texas Tech University, Lubbock 79409, USA*

Correspondence should be addressed to Tao Tao; taotao@ahut.edu.cn

Received 15 January 2020; Revised 5 March 2020; Accepted 10 March 2020; Published 5 May 2020

Guest Editor: Xuyun Zhang

The supervised learning-based recommendation models, whose infrastructures are sufficient training samples with high quality, have been widely applied in many domains. In the era of big data with the explosive growth of data volume, training samples should be labelled timely and accurately to guarantee the excellent recommendation performance of supervised learning-based models. Machine annotation cannot complete the tasks of labelling training samples with high quality because of limited machine intelligence. Although expert annotation can achieve a high accuracy, it requires a long time as well as more resources. As a new way of human intelligence to participate in machine computing, crowdsourcing annotation makes up for shortages of machine annotation and expert annotation. Therefore, in this paper, we utilize crowdsourcing annotation to label training samples. First, a suitable crowdsourcing mechanism is designed to create crowdsourcing annotation-based tasks for training sample labelling, and then two entropy-based ground truth inference algorithms (i.e., HILED and HILI) are proposed to achieve quality improvement of noise labels provided by the crowd. In addition, the descending and random order manners in crowdsourcing annotation-based tasks are also explored. The experimental results demonstrate that crowdsourcing annotation significantly improves the performance of machine annotation. Among the ground truth inference algorithms, both HILED and HILI improve the performance of baselines; meanwhile, HILED performs better than HILI.

1. Introduction

tRecommendation systems have increasingly attracted attention, since they can significantly alleviate the problem of information overload on the Internet and help people find items of interest or make better decisions in their daily life. Among the recommendation models, the supervised learning-based ones have been widely applied in many domains, such as cloud/edge computing [1], complex systems [2, 3], and Quality of Service (QoS) prediction [4, 5]. It is no doubt that sufficient training samples with high quality guarantee the excellent recommendation performance of supervised learning-based recommendation systems. Thus, it is necessary to study how to timely and accurately label sufficient training samples in the era of big data with the explosive growth of data volume. Although machine annotation can label enough training samples timely, they do not meet the requirement of high quality because of limited machine intelligence. So, it is natural to think of utilizing the intelligence of human beings.

Indeed, expert annotation (i.e., hiring domain experts to label training samples) can achieve a high accuracy. However, it requires a long time as well as more resources. Research studies [6, 7] demonstrated that crowdsourcing brings machine learning (and its related research fields) great opportunities because crowdsourcing can easily access the crowd via public or personal platforms [8, 9], such as MTurk [10], and efficiently deal with intelligent and computer-hard tasks by employing thousands of workers at a relatively low price. Therefore, as a new way of human

intelligence to participate in machine computing, crowdsourcing annotation makes up for the shortages of machine annotation and expert annotation. Crowdsourcing annotation has five steps: (a) the requesters select a public or personal crowdsourcing platform and design crowdsourcing annotation tasks, including price setting, time constraints, and required responding number of each annotation task. (b) The requesters publish crowdsourcing annotation tasks on the selected crowdsourcing platform. (c) The crowd logged in the platform (also known as workers) selects tasks that are suitable for themselves and complete tasks (i.e., providing labels). Note that the requester does not know any information (such as expertise and credit standing) of the workers completing annotation tasks in this step. (d) The requesters download the labels provided by workers and few additional information of workers (i.e., the completing times and the number of accepted tasks) from the crowdsourcing platform. (e) The requesters utilize existing ground truth inference algorithms or propose novel one(s) to infer truth value(s) from all labels provided by workers. In this paper, we focus on labelling training samples to keyphrase extraction by utilizing crowdsourcing annotation, since extracting keyphrases from a text (especially a short text) is a complex process that requires abundant auxiliary information, such as background of entities discussed and the events involved. Machine annotation and expert annotation cannot effectively handle keyphrase extraction because of their shortages. For convenience, our entire approach is denoted as Crowdsourced Keyphrase Extraction (CKE) hereafter; meanwhile, a single task of crowdsourcing annotation generated by CKE is named L-HIT.

Extracting keyphrases from training samples in CKE includes *labelling* and *ranking* operations, and each single L-HIT contains three task types [9, 11]:*multiple-choice*, *fill-in-blank*, and *rating*. The first two are used to collect proper keyphrases for a training sample, and the last one is used for importance ranking assignment of the proper keyphrases collected. This is different from binary labelling and most of multiclass labelling tasks, which usually have one single type. Besides, there are three important problems (i.e., quality control, cost control, and latency control) which are also required to be balanced in CKE [9]. Quality control focuses on labelling and ranking high-quality keyphrases, cost control aims to reduce the costs in terms of labour and money while keeping high-quality ground truth, and latency control studies how to cut down cycle of a single task [11]. We utilize four ways to handle trade-off among the three problems stated above in CKE.

In this paper, a pruning-based technique [9] is first adopted to prune the candidates provided by a machine-based algorithm; meanwhile, a complementary option is added to supplement the proper keyphrases that are lost because of various reasons. The pruning-based technique and the complementary option can efficiently reduce labour cost and time cost. Then, for each single L-HIT there is a time constraint set, since time constraints can significantly reduce the latency of a single worker [11]. Thirdly, each individual worker is asked to select an importance ranking for each keyphrase labelled by himself instead of sorting them. Finally, in order to conquer the possible low quality of some workers for keyphrase labelling and ranking, the designed crowdsourcing mechanism allows multiple workers [6] to complete a single L-HIT. The main contributions of this paper are summarized as follows:

(1) A suitable crowdsourcing mechanism is designed to create crowdsourcing annotation-based tasks for training sample labelling. In addition, four optimization methods (i.e., a pruning-based technique, a complementary option, time constraint set, and repeated labelling) are used to balance the quality, the cost, and the latency controls in CKE.

(2) Two entropy-based inference algorithms (i.e., HILED and HILI) are proposed to infer the ground truth based on labels collected by crowdsourcing annotation. In addition, two different order manners in L-HITs, which are the descending one and random one, are also explored.

(3) We conduct multiple experiments on MTurk to verify the performance improvement of crowdsourcing annotation. The experimental results demonstrate that crowdsourcing annotation performs well. Among the inference algorithms, both HILED and HILI improved the performance of the baselines.

The remainder of the paper is organized as follows. Section 2 will introduce the details of CKE, Section 3 will report the experimental results, the related works will be discussed in Section 4, and then we will reach a conclusion in Section 5.

2. Crowdsourced Keyphrase Extraction

In this section, we will first introduce the compositions of a single L-HIT, and then we will present the two proposed inference algorithms.

2.1. A Single L-HIT. Our multiple experiments are conducted on MTurk, which is a welcome crowdsourcing marketplace supporting crowdsourced execution of Human Intelligence Tasks (HITs) [12]. Since the structure of a single task published by our experiments is essentially inherited from a single HIT supported by MTurk, the ones published by us are called Labelling Human Intelligence Tasks (L-HITs). A single L-HIT, which corresponds to a single training sample, consists of five parts: guidance, content, candidate option, candidate supplement, and submission. As shown in Figure 1, the part of guidance (surrounded by a blue rectangle) helps workers complete the current task conveniently and efficiently. The part of content (surrounded by a black rectangle) shows workers the content of a single training sample. The part of submission (surrounded by a blue ellipse) is utilized to submit the completed L-HIT. These three parts are basic elements of the current task.

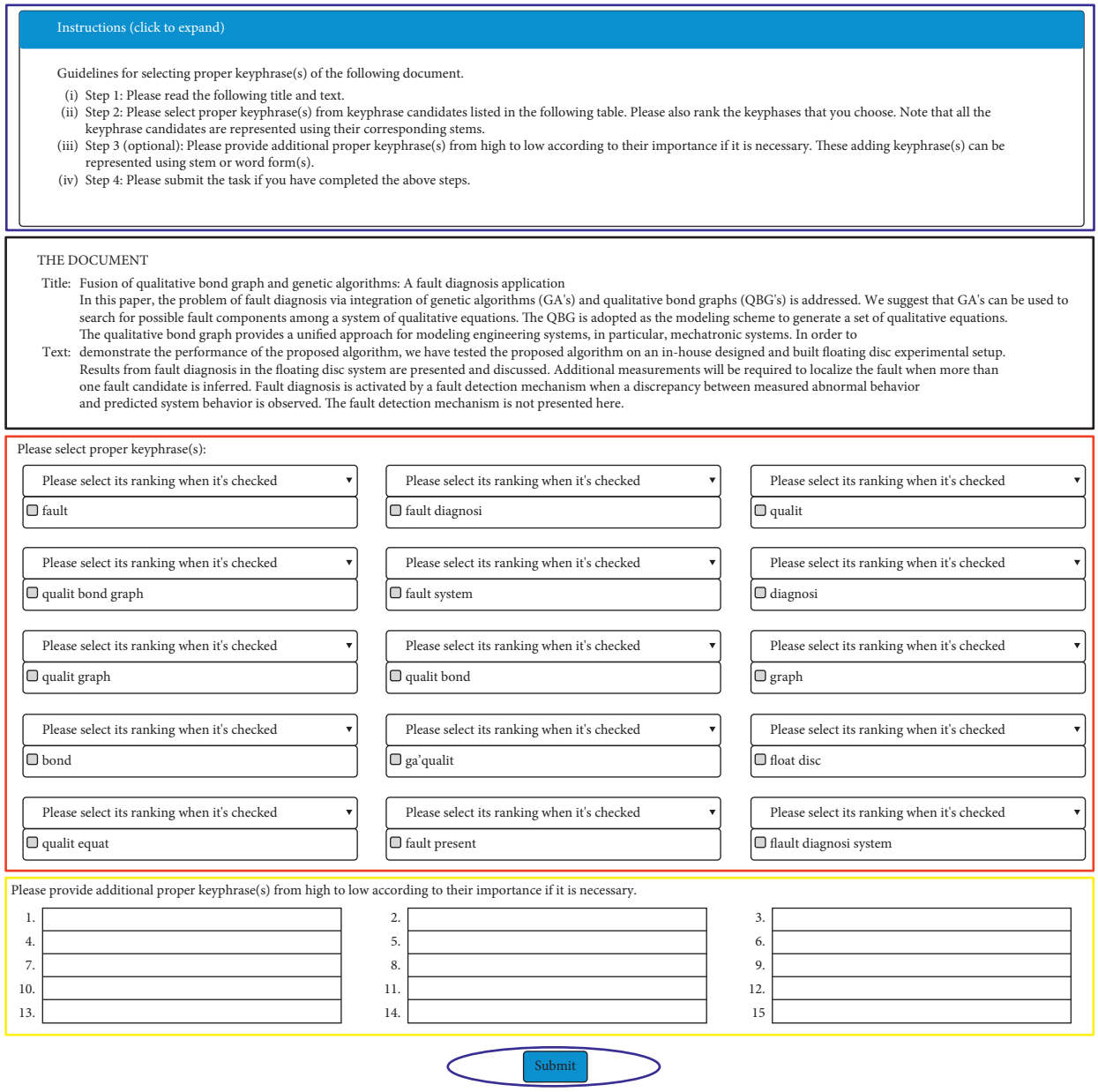

Figure 1: The main user interface of a single L-HIT.

(1) *Multiple-Choice*. When a worker has read the content of the training sample, he/she can directly select the proper option(s) from this part as the final keyphrase(s).

(2) *Rating*. Once an option is selected as a final keyphrase, the worker needs to select an importance ranking from the corresponding drop-down box. Our *rating* job is different from that in tasks of pairwise comparison (or rating) that ask workers to compare the selected items with each other [9]. It converts a comparison operation into an assignment one. That is, workers do not need to consider other selected options while assigning an importance ranking to a selected one based on their understanding of the current training sample. Such conversion can reduce latency while obtaining an ordered keyphrase list.

The part of candidate option (surrounded by a red rectangle) shows worker candidates. The candidates are keyphrases labelled by machine annotation. Note that this part only holds 15 options at most. If a training sample has more than 15 keyphrases labelled by machine annotation, this part only shows the top 15 ones with the highest scores. In addition, for each candidate, there is an independent drop-down box (providing importance rankings) above it. The importance ranking denotes how important the option is to the current training sample. It varies from −2 to 2, where 2 denotes the importance with the highest level and −2 denotes the importance with the least level. The part of candidate option has two task types as follows.

Some proper keyphrases may not be listed in the part of candidate option because of various reasons, for instance, phrases with low appearing frequencies or ones with low scores assigned by machine annotation. Therefore, for each single L-HIT, there is a candidate supplement part that lets workers supplement lost keyphrases as well as the corresponding importance rankings (surrounded by a yellow rectangle). The part of candidate supplement also has two task types, which are *fill-in-blank* (i.e., supplementing lost keyphrase(s)) and *rating* (i.e., selecting importance rankings), respectively. Note that supplementing the lost keyphrase(s) is an optional job for workers.

2.2. Inference Algorithms. In this paper, inferring a truth keyphrase list is still viewed as a process of first-integrating last-grading phrases. Although algorithms IMLK, IMLK-I, and IMLK-ED [13] are suitable for inferring a truth keyphrase list from multiple lists of keyphrases, they neglect to calculate three inherent attributes of a keyphrase capturing a topic delivered by the training samples, which are meaningfulness, uncertainty, and uselessness [14]. Study [15] shows that calculating the information entropy [16] of a keyphrase is a significant way to measure these three inherent attributes of a keyphrase. Therefore, we utilize the information entropy and corresponding equations in [15] to measure the three inherent properties of a keyphrase capturing a topic. The symbols used for ground truth inference algorithms are shown in Table 1.

The attribute meaningfulness of k in T denotes the k's positive probability of capturing a topic expressed by T. Normally, it is measured by the distribution of k as an independent keyphrase, since the more times k^{indie} occurs, the bigger positive probability the topic is delivered by k. The attribute meaningfulness is defined as follows:

$$P_{\text{pos}} = \begin{cases} \#\text{NKI}/\#\text{TN}, & 0 < \#\text{NKI} < \#\text{TN}, \\ 0, & \#\text{NKI} = 0, \end{cases} \tag{1}$$

where $P_{\text{pos}} = 0$ for the case that k does not exist in the corpus.

As the name implies, the attribute uncertainty of k in T denotes the k's unsteadiness of capturing a topic expressed by T, which is usually measured by the distribution of T as a sub-keyphrase. A sub-keyphrase means it can be extended into another keyphrase with other words. Note that (a) different keyphrases express a same point with different expression depth and (b) different keyphrases express totally different points. For example, although keyphrase "*topic model*" is a sub-keyphrase of "*topic aware propagation model*," they express different points. Intuitively, the more times k^{sub} occurs, the more unsteady the topic is delivered by k. The attribute uncertainty is defined as follows:

$$P_{\text{sub}} = \begin{cases} \#\text{NKS}/\#\text{TN}, & 0 < \#\text{NKS} < \#\text{TN}, \\ 0, & \#\text{NKS} = 0. \end{cases} \tag{2}$$

The attribute uselessness of k in T denotes the k's negative probability of capturing a topic expressed by T, which is defined as follows:

Table 1: Symbols.

No.	Symbols	Presentation
1	K	A keyphrase
2	T	A training sample
3	k^{indie}	An independent keyphrase
4	P_{pos}	The attribute meaningfulness
5	#NKI	The number of k^{indie} occurs
6	k^{sub}	A sub-keyphrase
7	P_{sub}	The attribute uncertainty
8	#NKS	The number of k^{sub} occurs
9	#TN	The total number of keyphrases in the corpus
10	P_{neg}	The attribute uselessness
11	$H(k)$	The information entropy of k

$$P_{\text{neg}} = 1 - P_{\text{pos}} - P_{\text{sub}}. \tag{3}$$

In conclusion, the information entropy of k can completely measure its three inherent attributes using equations (4) or (5) (when the situation $P_{\text{sub}} = 0$ occurs).

$$H(k) = P_{\text{pos}}\log\left(\frac{1}{P_{\text{pos}}}\right) + P_{\text{sub}}\log\left(\frac{1}{P_{\text{sub}}}\right) + P_{\text{neg}}\log\left(\frac{1}{P_{\text{neg}}}\right), \tag{4}$$

$$H(k) = P_{\text{pos}}\log\left(\frac{1}{P_{\text{pos}}}\right) + P_{\text{neg}}\log\left(\frac{1}{P_{\text{neg}}}\right). \tag{5}$$

Finally, by combining the information entropy, algorithms HILED and HILI are proposed based on algorithms IMLK-ED and IMLK-I stated in [13], respectively, and the corresponding equations recalculating the keyphrases' grades are modified as follows:

$$G_{ij} = \frac{\sum_{j=1}^{m} Q_j^{ED} \times \left(\text{RS}_{ij}\right)^2 \times H\left(k_{ij}\right)}{m}, \qquad G_{ij} = \frac{\sum_{j=1}^{m} Q_j^{I} \times \left(\text{RS}_{ij}\right)^2 \times H\left(k_{ij}\right)}{m}, \tag{6}$$

where $H(k_{ij})$ denotes the information entropy of the i^{th} keyphrase in the j^{th} keyphrase list, RS_{ij} denotes the importance scores provided by workers, Q_j^{ED} denotes the quality of a worker who provides the j^{th} keyphrase list in the algorithm HILED, Q_j^I denotes the quality of a worker who provides the j^{th} keyphrase list in the algorithm HILI, and m denotes the total number of keyphrase lists provided by a worker.

3. Experiments and Discussion

In this section, we will first introduce experiments with different order manners, which are the descending and the random ones, and then we will discuss the factors of influencing performance improvement of crowdsourcing annotation.

3.1. Crowdsourcing Experiment with Descending Ranking. Since IMLK, IMLK-I, and IMLK-ED proposed in [13] and KeyRank proposed in [15] perform very well, we employed

them as baselines. KeyRank is one of the machine annotation methods, and its performance is evaluated on dataset INSPEC [17] containing 2000 abstracts (1000 for training, 500 for development, and 500 for testing) in [15]. Considering the cost and latency of workers, we chose 100 abstracts from the 500 test ones in dataset INSPEC, where KeyRank performs the best, as the data for our multiple crowdsourcing experiments. In addition, the gold standards of these 100 test abstracts are treated as labelled ones from expert annotation. As we said before, each single abstract corresponds to a single L-HIT. That is, we have 100 corresponding L-HITs. The part of candidate option in each L-HIT lists 15 (or fewer) candidates with descending ranking. These candidates are keyphrases labelled and weighted by KeyRank. Again, in order to overcome the shortage that the quality of an individual worker for keyphrase extraction is sometimes rather low, we request 10 responses for each L-HIT from 10 different workers. That is, the whole experiment has 1000 published L-HITs since each one has to be published ten times on MTurk. Each L-HIT costs 5 cents, and the whole experiment costs 50 dollars totally. According to feedback from crowdsourcing platform MTurk, more than four out of five workers completed the optional "candidate supplement" tasks. The minimum time that a single crowdsourcing task required is 50 seconds, and the maximum time is 5 minutes. The time required for most of the crowdsourcing tasks was between 90 and 200 seconds.

The precision (P), recall (R), and F_1 score are employed as performance metrics. P, R, and F_1 score are defined as follows:

$$P = \frac{\#\text{correct}}{\#\text{labelled}},$$
$$R = \frac{\#\text{correct}}{\#\text{expert}}, \tag{7}$$
$$F_1 = 2 \times P \times \frac{R}{(P+R)},$$

where #correct denotes the number of correct keyphrases obtained from crowdsourcing annotation, #labelled denotes the number of keyphrases obtained from crowdsourcing annotation, and #expert denotes the number of keyphrases obtained from expert annotation. Normally, #expert for most abstracts varies from 3 to 5, so that the value of #labelled in our experiment varies from 3 to 5.

After 10 responses of each L-HIT are obtained from 10 different workers, algorithms IMLK, IMLK-I, IMLK-ED, HILED, and HILI are applied to infer a truth keyphrase list from these responses. The inferred results of IMLK, IMLK-I, IMLK-ED, HILED, and HILI are compared with those of KeyRank in terms of P, R, and F_1 score. Besides, in order to evaluate the performance of KeyRank, IMLK, IMLK-I, IMLK-ED, HILED, and HILI clearly, the comparisons are divided into three different groups, i.e., Group-3, Group-4, and Group-5. For example, Group-4 is named as such because the number of #labelled is 4, when it reports the comparisons among KeyRank, IMLK, IMLK-I, IMLK-ED, HILED, and HILI in terms of P, R, and F_1 score, respectively. In addition, the relations between the workers' numbers (denoted as #WorkerNum) and the inferred results are also explored by, respectively, conducting another seven comparisons in all groups. The values of #WorkerNum are set to 3, 4, 5, 6, 7, 8, and 9, respectively. Since each abstract has 10 keyphrase lists provided by 10 different workers, respectively, in order to get rid of the impact of workers' order, each algorithm on each abstract is run ten times under a certain #WorkerNum, and the corresponding number of keyphrase lists are randomly selected from its 10 keyphrase lists at each time. For example, when the #WorkerNum is 5, we randomly select 5 keyphrase lists from the 10 keyphrase lists. All comparisons of all groups among KeyRank, IMLK, IMLK-I, IMLK-ED, HILED, and HILI are shown in Figure 2.

From Figure 2, we notice that IMLK, IMLK-I, and IMLK-ED significantly perform better than KeyRank in all groups in terms of P, R, and F_1 score. We also notice that both HILED and HILI significantly perform better than KeyRank, IMLK, IMLK-I, and IMLK-ED in all groups in terms of P, R, and F_1 score. Between HILED and HILI, except the comparisons in Group-3, Group-4, and Group-5, when the values of #WorkerNum are 5, 6, and 7 (the situation of #WorkerNum = 7 only occurs in Group-3) in terms of P, R, and F_1 score, HILED always performs better than HILI. Moreover, we notice that with the increment of #WorkerNum, the performance of IMLK, IMLK-I, IMLK-ED, HILI, and HILED has a rising trend. Therefore, we can conclude that (1) both HILED and HILI perform better than IMLK, IMLK-I, and IMLK-ED; (2) HILED performs a little better than HILI; (3) #WorkerNum does influence the inferred results; and (4) employing crowdsourcing annotation is a feasible and effective way for training sample labelling.

3.2. Crowdsourcing Experiment with Random Ranking. For each published L-HIT in the Crowdsourcing experiment with Descending Ranking (denoted as CDR) in Section 3.1, the 15 (or fewer) candidates listed in the part of candidate option are ordered according to their scores assigned by KeyRank from high to low. Is there any relevancy between the order manners of the listed candidates and the improvement performance of crowdsourcing annotation?

In order to explore whether there is such a relevancy between them, we create another 100 L-HITs using the selected 100 representative abstracts mentioned in Section 3.1. Meanwhile, we also request 10 responses for each L-HIT from 10 different workers. For each L-HIT, the 15 (or fewer) candidates are randomly listed in the part of candidate option. We named the experiments conducted in this section Crowdsourcing experiment with Random Ranking (denoted as CRR). To make a fair evaluation, all experimental parameters of CRR follow those of CDR. All comparisons among KeyRank, IMLK, HILED, and HILI in terms of P, R, and F_1 scores are shown in Figure 3.

From Figure 3, we can see that IMLK, HILED, and HILI in CRR always significantly perform better than KeyRank in terms of P, R, and F_1 score. It proves once again that employing crowdsourcing annotation is a feasible and

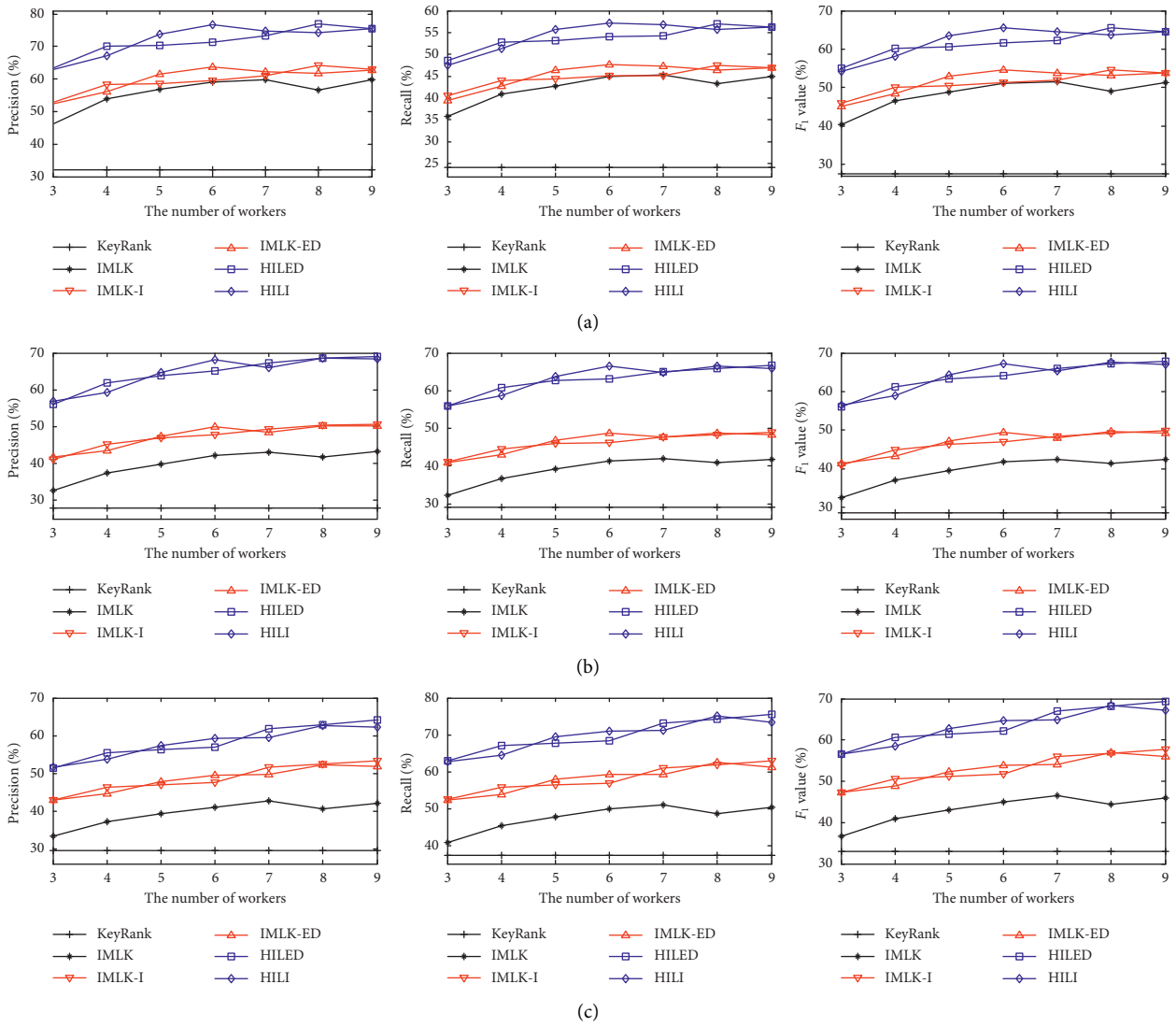

Figure 2: Comparisons among KeyRank, IMLK, IMKL-I, IMLK-ED, HILED, and HILI in all groups. (a) Group-3. (b) Group-4. (c) Group-5.

effective way for training sample labelling. However, we notice that the performance of IMLK, HILED, and HILI in CRR is worse than that of these algorithms in CDR, which proves that the order manners of the listed candidates do influence the improvement performance of crowdsourcing annotation, and the descending order manner is more effective than the random one.

3.3. Discussion

The Proper Number of Workers. Either CDR or CRR shows us that with an increment of #WorkerNum, the improvement performance of crowdsourcing annotation has a rising trend. However, more workers do not mean more suitability. On the one hand, more workers may result in more latency. For instance, workers may be distracted or tasks may not be appealing to enough workers. On the other hand, more workers mean more monetary cost since crowdsourcing annotation is not free. It is just a cheaper way to label sufficient training samples timely. Hence, the trade-off among quality, latency, and cost controls needs to be considered and balanced. The experimental results show that the proper number of workers varies from 6 to 8 because the improvement performance of crowdsourcing annotation at these stages is relatively stable and the quantity is appropriate to avoid high latency and cost.

The Descending and Random Ranking Manners. The experimental results demonstrate that the descending ranking manner performs better than the random one. The reason may be that workers have limited patience since they are not trained. Normally, workers just focus on the top 5 (or less 5) candidates listed in the part of candidate option. If they do not find any proper one(s) from the top few candidates, they may lose patience to read the remaining ones, so that they would select randomly or supplement option(s) in the part of candidate supplement for completing the current L-HIT.

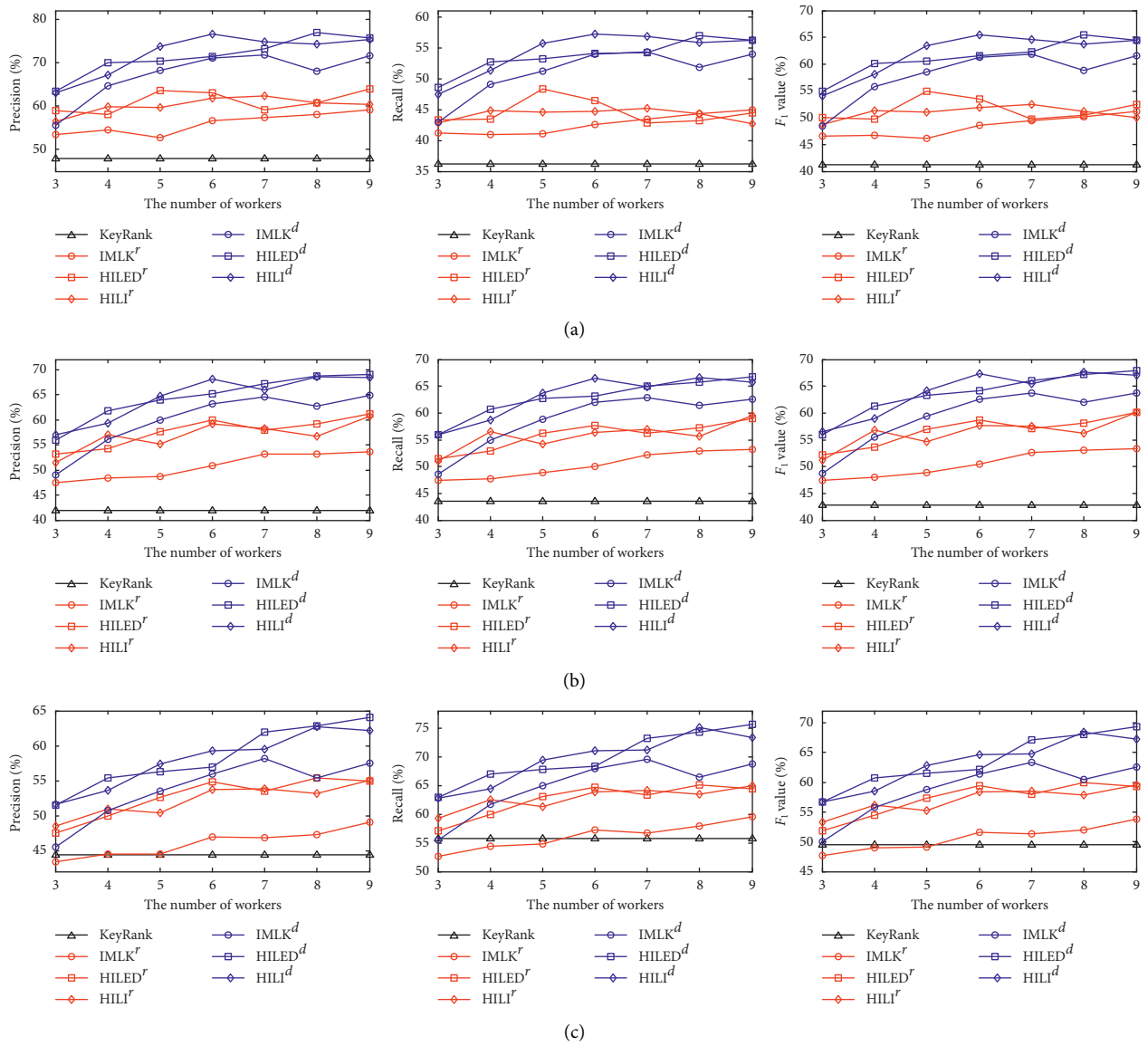

Figure 3: Comparisons among KeyRank, IMLK, HILED, and HILI in CRR and CDR. IMLK^r,HILED^r, and HILI^r denote the performance of algorithms IMLK, HILED, and HILI in CRR. IMLK^d, HILED^d, and HILI^d denote the performance of algorithms IMLK, HILED, and HILI in CDR. (a) Group-3. (b) Group-4. (c) Group-5.

However, the random selected one(s) may not be proper, and the supplementary one(s) may be repeated with the candidates listed in the part of candidate option. Therefore, the loss of accuracy happens.

4. Related Work

Recommendation models [18] have been widely applied in many domains, such as complex systems [19, 20], Quality of Service (QoS) prediction [21, 22], reliability detection for real-time systems [23], social networks [24–26], and others [27–29]. Among existing recommendation models, the supervised learning-based ones have increasingly attracted attention because of effectiveness. However, it is well known that the supervised learning-based recommendation models suffer from the quality of training samples. Therefore, labelling sufficient training samples timely and accurately in the era of big data becomes an important foundation to the supervised learning-based recommendation. Since this paper focuses on labelling training samples to keyphrase extraction by utilizing crowdsourcing annotation, the related work will be introduced in terms of keyphrase extraction and crowdsourcing annotation.

Most original works labelling keyphrases simply selected single or contiguous words with a high frequency, such as KEA [14]. Yet these single or contiguous words do not always deliver main points discussed in a text. Study [30] demonstrated that semantic relations in context can help extract high-quality keyphrases. Hence, some research studies employed knowledge bases and ontologies to obtain

semantic relations in context to improve qualities of extracting keyphrases [31]. It is obvious that the semantic relations obtained by these methods are restricted by the corresponding knowledge bases and ontologies. Studies [32, 33] utilized graph-based ranking methods to label keyphrases, in which a keyphrase's importance is determined by its semantic relatedness to others. As they just aggregate keyphrases from one single document, the corresponding semantic relatedness is not stable and could not accurately reveal the "relatedness" between keyphrases in general. Studies [34, 35] applied sequential pattern mining with wildcards to label keyphrases, since wildcards provide gap constraints with flexibility for capturing semantic relations in context. However, most of them are computationally expensive as they need to repeatedly scan the whole document. In addition, they require users to explicitly specify appropriate gap constraints beforehand, which is time-consuming and not realistic. According to the common sense that words do not repeatedly appear in an effective keyphrase, KeyRank [15] converted the repeated scanning operation into a calculating model and significantly reduced time consumption. However, it is also frequency-based algorithm that may lose important entities with low frequencies. To sum up, machine annotation can label enough training samples timely, and they do not meet the requirement of high quality because of limited machine intelligence. Hiring domain experts can achieve a high accuracy. However, it requires a long time as well more high resources. Therefore, it is natural to think of utilizing crowdsourcing annotation, which is a new way of human intelligence to participate in machine computing at a relatively low price, to label sufficient training samples timely and accurately.

Studies [6–8] showed that crowdsourcing brings great opportunities to machine learning as well as its related research fields. With the appearance of crowdsourcing platforms, such as MTurk [10] and CrowdFlower [36], crowdsourcing has taken off in a wide range of applications, for example, entity resolution [37] and sentiment analysis [38]. Despite the diversity of applications, they all employ crowdsourcing annotation at low cost to collect data (labels of training samples) to resolve corresponding intelligent problems. In addition, many crowdsourcing annotation-based systems (frameworks) are proposed to resolve computer-hard and intelligent tasks. By utilizing crowdsourcing annotation-based methods, CrowdCleaner [39] can detect and repair errors that usually cannot be solved by traditional data integration and cleaning techniques. CrowdPlanner [40] recommends the best route with respect to the knowledge of experienced drivers. AggNet [12] is a novel crowdsourcing annotation-based aggregation framework, which asks workers to detect the mitosis in breast cancer histology images after training the crowd with a few examples.

Since some individuals in the crowd may yield relatively low-quality answers or even noise, many researches focus on how to infer the ground truth according to labels provided by workers [9]. Zheng et al. [41] employed a domain-sensitive worker model to accurately infer the ground truth based on two principles: (1) a label provided by a worker is trusted, if the worker is a domain expert on the corresponding tasks; and (2) a worker is a domain expert if he often correctly completes tasks related to the specific domain. Zheng et al. [42] provided a detailed survey on ground truth inference on crowdsourcing annotation and performed an in-depth analysis of 17 existing methods. Zhang et al. tried to utilize active learning and label noise correction to improve the quality of truth inference [43–45]. One of our preliminary works [13] treated the ground truth inference of labelling keyphrases as an integrating and ranking process and proposed three novel algorithms IMLK, IMLK-I, and IMLK-ED. However, these three algorithms ignore three inherent properties of a keyphrase capturing a point expressed by the text, which are meaningfulness, uncertainty, and uselessness.

5. Conclusions

This paper focuses on labelling training samples to keyphrase extraction by utilizing crowdsourcing annotation. We designed novel crowdsourcing mechanisms to create corresponding crowdsourcing annotation-based tasks for training samples labelling and proposed two entropy-based inference algorithms (HILED and HILI) to improve the quality of labelled training samples. The experimental results showed that crowdsourcing annotation can achieve more effective improvement performance than the approach of machine annotation (i.e., KeyRank) does. In addition, we demonstrated that the ranking manners of candidates, which are listed in the part of candidate option, do influence the improvement performance of crowdsourcing annotation, and the descending ranking manner is more effective than the random one. In the future, we will keep focusing on inference algorithms, improving qualities of labelled training samples.

Data Availability

The data used in this study can be accessed via https://github.com/snkim/AutomaticKeyphraseExtraction.

Conflicts of Interest

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

Acknowledgments

This study was partially supported by the National Key R&D Program of China (grant no. 2019YFB1704101), the National Natural Science Foundation of China (grant nos. U1936220 and 31771679), the Anhui Foundation for Science and Technology Major Project (grant nos. 18030901034 and 201904e01020006), the Key Laboratory of Agricultural Electronic Commerce, Ministry of Agriculture of China (grant nos. AEC2018003 and AEC2018006), the 2019 Anhui University Collaborative Innovation Project (GXXT-2019-013), and the Hefei Major Research Project of Key Technology (J2018G14).

References

[1] X. Xu, Q. Liu, Y. Luo et al., "A computation offloading method over big data for IoT-enabled cloud-edge computing," *Future Generation Computer Systems*, vol. 95, pp. 522–533, 2019.

[2] J. Zhou, J. Sun, P. Cong et al., "Security-critical energy-aware task scheduling for heterogeneous real-time MPSoCs in IoT," *IEEE Transactions on Services Computing*, 2019, In press.

[3] J. Zhou, X. S. Hu, Y. Ma, J. Sun, T. Wei, and S. Hu, "Improving availability of multicore real-time systems suffering both permanent and transient faults," *IEEE Transactions on Computers*, vol. 68, no. 12, pp. 1785–1801, 2019.

[4] Y. Zhang, K. Wang, Q. He et al., "Covering-based web service quality prediction via neighborhood-aware matrix factorization," *IEEE Transactions on Services Computing*, 2019, In press.

[5] Y. Zhang, G. Cui, S. Deng et al., "Efficient query of quality correlation for service composition," *IEEE Transactions on Services Computing*, 2019, In press.

[6] M. Lease, "On quality control and machine learning in crowdsourcing," in *Proceedings of the Workshops at the 25th AAAI Conference on Artificial Intelligence*, pp. 97–102, San Francisco, CA, USA, January 2011.

[7] J. Zhang, X. Wu, and V. S. Sheng, "Learning from crowdsourced labeled data: a survey," *Artificial Intelligence Review*, vol. 46, no. 4, pp. 543–576, 2016.

[8] V. S. Sheng, F. Provost, and P. G. Ipeirotis, "Get another label? Improving data quality and data mining using multiple, noisy labelers," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 614–622, August 2008.

[9] G. Li, J. Wang, Y. Zheng, and M. J. Franklin, "Crowdsourced data management: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2296–2319, 2016.

[10] Mturk, 2020, https://www.mturk.com.

[11] G. Li, Y. Zheng, J. Fan, J. Wang, and R. Cheng, "Crowdsourced data management: overview and challenges," *Proceedings of the 2017 ACM International Conference on Management of Data*, pp. 1711–1716, Association for Computing Machinery, New York, NY, USA, 2017.

[12] S. Albarqouni, C. Baur, F. Achilles, V. Belagiannis, S. Demirci, and N. Navab, "AggNet: deep learning from crowds for mitosis detection in breast cancer histology images," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1313–1321, 2016.

[13] Q. Wang, V. S. Sheng, and Z. Liu, "Exploring methods of assessing influence relevance of news articles," in *Cloud Computing and Security*, pp. 525–536, Springer, Berlin, Germany, 2018.

[14] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning, "KEA: Practical automatic keyphrase extraction," in *Proceedings of the 4th ACM Conference on Digital Libraries*, pp. 1–23, Berkeley, CA, USA, August 1999.

[15] Q. Wang, V. S. Sheng, and X. Wu, "Document-specific keyphrase candidate search and ranking," *Expert Systems with Applications*, vol. 97, pp. 163–176, 2018.

[16] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[17] INSPEC, 2020, https://github.com/snkim/AutomaticKeyphrase Extraction.

[18] A. Ramlatchan, M. Yang, Q. Liu, M. Li, J. Wang, and Y. Li, "A survey of matrix completion methods for recommendation systems," *Big Data Mining and Analytics*, vol. 1, no. 4, pp. 308–323, 2018.

[19] X. Xu, R. Mo, F. Dai, W. Lin, S. Wan, and W. Dou, "Dynamic resource provisioning with fault tolerance for data-intensive meteorological workflows in cloud," *IEEE Transactions on Industrial Informatics*, 2019.

[20] L. Qi, Y. Chen, Y. Yuan, S. Fu, X. Zhang, and X. Xu, "A QoS-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems," *World Wide Web*, vol. 23, no. 2, pp. 1275–1297, 2019.

[21] Y. Zhang, C. Yin, Q. Wu et al., "Location-aware deep collaborative filtering for service recommendation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019.

[22] L. Qi, Q. He, F. Chen et al., "Finding all you need: web APIs recommendation in web of things through keywords search," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 5, pp. 1063–1072, 2019.

[23] J. Zhou, J. Sun, X. Zhou et al., "Resource management for improving soft-error and lifetime reliability of real-time MPSoCs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 12, pp. 2215–2228, 2019.

[24] G. Liu, Y. Wang, M. A. Orgun et al., "Finding the optimal social trust path for the selection of trustworthy service providers in complex social networks," *IEEE Transactions on Services Computing*, vol. 6, no. 2, pp. 152–167, 2011.

[25] G. Liu, Y. Wang, and M. A. Orgun, "Optimal social trust path selection in complex social networks," in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, Atlanta, GA, USA, July 2010.

[26] G. Liu, K. Zheng, Y. Wang et al., "Multi-constrained graph pattern matching in large-scale contextual social graphs," in *Proceedings of the 2015 IEEE 31st International Conference on Data Engineering*, pp. 351–362, IEEE, Seoul, South Korea, April 2015.

[27] C. Zhang, M. Yang, J. Lv, and W. Yang, "An improved hybrid collaborative filtering algorithm based on tags and time factor," *Big Data Mining and Analytics*, vol. 1, no. 2, pp. 128–136, 2018.

[28] Y. Liu, S. Wang, M. S. Khan, and J. He, "A novel deep hybrid recommender system based on auto-encoder with neural collaborative filtering," *Big Data Mining and Analytics*, vol. 1, no. 3, pp. 211–221, 2018.

[29] H. Liu, H. Kou, C. Yan, and L. Qi, "Link prediction in paper citation network to construct paper correlation graph," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, 2019.

[30] G. Ercan and I. Cicekli, "Using lexical chains for keyword extraction," *Information Processing and Management*, vol. 43, no. 6, pp. 1705–1714, 2007.

[31] S. Xu, S. Yang, and C. M. Lau, "Keyword extraction and headline generation using novel word feature," in *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pp. 1461–1466, Atlanta, GA, USA, 2010.

[32] R. Mihalcea and P. Tarau, "TextRank: bringing order into texts," *UNT Scholarly Works*, vol. 43, no. 6, pp. 404–411, 2004.

[33] K. S. Hasan and V. Ng, "Automatic keyphrase extraction: a survey of the state of the art," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 1262–1273, Baltimore, MD, USA, June 2014.

[34] F. Xie, X. Wu, and X. Zhu, "Document-specific keyphrase extraction using sequential patterns with wildcards," in *Proceedings of the 2014 IEEE International Conference on Data Mining*, pp. 1055–1060, Shenzhen, China, December 2014.

[35] J. Feng, F. Xie, X. Hu, P. Li, J. Cao, and X. Wu, "Keyword extraction based on sequential pattern mining," in *Proceedings of the 3rd International Conference on Internet Multimedia Computing and Service*, pp. 34–38, Chengdu, China, August 2011.

[36] Crowdflower, 2020, http://www.crowdflower.com.

[37] S. Wang, X. Xiao, and C. Lee, "Crowd-based deduplication: an adaptive approach," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pp. 1263–1277, Melbourne, Australia, June 2015.

[38] Y. Zheng, J. Wang, G. Li, R. Cheng, and J. Feng, "QASCA: a quality-aware task assignment system for crowdsourcing applications," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pp. 1031–1046, Melbourne, Australia, June 2015.

[39] Y. Tong, C. C. Cao, C. J. Zhang, Y. Li, and L. Chen, "CrowdCleaner: data cleaning for multi-version data on the web via crowdsourcing," in *Proceedings of the 2014 IEEE 30th International Conference on Data Engineering*, pp. 1182–1185, Chicago, IL, USA, April 2014.

[40] H. Su, K. Zheng, J. Huang et al., "A crowd-based route recommendation system," in *Proceedings of the 2014 IEEE 30th International Conference on Data Engineering*, pp. 1144–1155, Chicago, IL, USA, May 2014.

[41] Y. Zheng, G. Li, and R. Cheng, "DOCS: domain-aware crowdsourcing system," *Proceedings of the Vldb Endowment*, vol. 10, no. 4, pp. 361–372, 2016.

[42] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng, "Truth inference in crowdsourcing: is the problem solved?" *Proceedings of the Vldb Endowment*, vol. 10, no. 5, pp. 541–552, 2017.

[43] J. Wu, S. Zhao, V. S. Sheng et al., "Weak-labeled active learning with conditional label dependence for multilabel image classification," *IEEE Transactions on Multimedia*, vol. 19, no. 6, pp. 1156–1169, 2017.

[44] B. Nicholson, J. Zhang, V. S. Sheng et al., "Label noise correction methods," in *Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–9, IEEE, Paris, France, October 2015.

[45] J. Zhang, V. S. Sheng, Q. Li, J. Wu, and X. Wu, "Consensus algorithms for biased labeling in crowdsourcing," *Information Sciences*, vol. 382-383, pp. 254–273, 2017.