Labelling Training Samples Using Crowdsourcing Annotation for Recommendation

Qingren Wang, Min Zhang, Tao Tao, and Victor S. Sheng

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

Copyright © 2020 Qingren Wang 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.

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 noiselabels 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

Recommendation 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 is called Labelling Human Intelligence Tasks (HITs) [12]. Since the structure of a single HIT supported by MTurk is essentially inherited from a single HIT supported by MTurk, the ones published by us are called Labelling Human Intelligence Tasks (L-HITs) [12]. 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.

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.

(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.
THE DOCUMENT

Title: Fusion of qualitative bond graph and genetic algorithms: A fault diagnosis application

In this paper, the problem of fault diagnosis via integration of genetic algorithms (GA’s) and qualitative bond graphs (QBGs) is addressed. We suggest that GA’s can be used to search for possible fault components among a system of qualitative equations. The QBG is adopted as the modeling scheme to generate a set of qualitative equations. The qualitative bond graph provides a unified approach for modeling engineering systems, in particular, mechatronic systems. In order to demonstrate the performance of the proposed algorithm, we have tested the proposed algorithm on an in-house designed and built floating disc experimental setup. Results from fault diagnosis in the floating disc system are presented and discussed. Additional measurements will be required to localize the fault when more than one fault candidate is inferred. Fault diagnosis is activated by a fault detection mechanism when a discrepancy between measured abnormal behavior and predicted system behavior is observed. The fault detection mechanism is not presented here.

Please select proper keyphrase(s):

1. fault
2. fault diagnosis
3. qualitative bond graph
4. qualitative graph
5. bond
6. ga
7. qualitative equation
8. float disc
9. fault present
10. fault diagnosis system
11. qualit sys
12. diagnosi
13. fault diagnosi
14. ga'qualit
15. float disc

Please provide additional proper keyphrase(s) from high to low according to their importance if it is necessary.

Submit

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 $k_{\text{inde}}$ occurs, the bigger positive probability the topic is delivered by $k$. The attribute meaningfulness is defined as follows:

$$P_{\text{pos}} = \begin{cases} \frac{\#\text{NI}/\#\text{TN}}{1}, & \text{if } \#\text{NI} < \#\text{TN} \\ 0, & \text{if } \#\text{NI} = \#\text{TN} \\ \frac{\#\text{NI}}{0}, & \text{if } \#\text{NI} = 0. \end{cases} \quad (1)$$

where $P_{\text{neg}} = 0$ 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 $k_{\text{sub}}$ occurs, the more unsteady the topic is delivered by $k$. The attribute uncertainty is defined as follows:

$$P_{\text{sub}} = \begin{cases} \frac{\#\text{NK}_{\text{S}}/\#\text{TN}}{1}, & \text{if } \#\text{NK}_{\text{S}} < \#\text{TN} \\ 0, & \text{if } \#\text{NK}_{\text{S}} = 0. \end{cases} \quad (2)$$

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:

$$P_{\text{neg}} = 1 - P_{\text{pos}} - P_{\text{sub}}. \quad (3)$$

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).

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

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

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:

$$G_{ij} = \frac{\sum_{j=1}^{m} Q_{ij}^E \times (RS_{ij})^2 \times H(k_{ij})}{m}, \quad (6)$$

$$G_{ij} = \frac{\sum_{j=1}^{m} Q_{ij}^E \times (RS_{ij})^2 \times H(k_{ij})}{m},$$

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, $Q_{ij}^E$ denotes the quality of a worker who provides the $j^{th}$ keyphrase list in the algorithm HILED, $Q_{ij}^E$ 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:

\[
P = \frac{\#\text{correct}}{\#\text{labelled}}
\]

\[
R = \frac{\#\text{correct}}{\#\text{expert}}
\]

\[
F_1 = 2 \times P \times \frac{R}{(P + R)}
\]

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, HIaled, and HILI are applied to infer a truth keyphrase list from these responses. The inferred results of IMLK, IMLK-I, IMLK-ED, HIaled, 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, HIaled, 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, HIaled, 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, HIaled, 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 HIaled 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 HIaled 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, HIaled always performs better than HILI. Moreover, we notice that with the increment of \#WorkerNum, the performance of IMLK, IMLK-I, IMLK-ED, HIaled, and HILI has a rising trend. Therefore, we can conclude that (1) both HIaled and HILI perform better than IMLK, IMLK-I, and IMLK-ED; (2) HIaled 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, HIaled, and HILI in terms of \( P, R, \) and \( F_1 \) scores are shown in Figure 3.

From Figure 3, we can see that IMLK, HIaled, 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
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 2: Comparisons among KeyRank, IMLK, IMKL-I, IMLK-ED, HILED, and HILI in all groups. (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...
semantics 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 quality of labelled training samples. 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 20190401020006), 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


