Aggregating Reliable Submissions in Crowdsourcing Systems

AYSWARYA R KURUP\textsuperscript{1}, G P SAJEEV\textsuperscript{1, 2} (Senior Member, IEEE), SWAMINATHAN J\textsuperscript{3}

\textsuperscript{1}Department of Computer Science and Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Amritapuri, India.
\textsuperscript{2}Dept of Electronics & Communication, Govt Engineering College Kozhikode, India, (e-mail:gpsajeev@gmail.com).
\textsuperscript{3}Department of Computer Science and Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Amritapuri, India, (e-mail:swaminathanj@am.amrita.edu).

Corresponding author: Ayswarya R Kurup (e-mail: ayswaryarkurup16@gmail.com).

I. INTRODUCTION
Crowdsourcing combines human intelligence and technology to solve problems challenging for automated processes. It is defined as "the process of outsourcing a piece of work to an undefined group of people called crowd via on-line platforms". Recently, it has achieved popularity as an effective tool for solving problems in a fast and cost-effective way. Few examples of crowdsourced tasks include sentiment analysis, data classification, article writing, and content generation. Amazon Mechanical Turk (AMT), Figure Eight and Innocentive are well-known examples of crowdsourcing systems for handling the applications mentioned above [12, 14].

Crowdsourcing systems have mainly three stakeholders, namely requesters, workers, and service platform [18]. In a crowdsourcing platform, the requesters post tasks, the platform allocates tasks to workers, upon completion of tasks, the workers submit solutions back to the platform. The submissions are verified by the requesters and approved for payment to selected submissions [19]. Fig. 1 illustrates the crowdsourcing workflow. When the platform engages different workers for same task, it aggregates submissions before delivering them to the requesters. Thus, task aggregation has a significant role in improving the quality of submissions and maintaining the stability of the platform [4].

The main challenge involved in task aggregation is to deal with the workers having varying levels of skills, expertise, and motivational factors. The imbalance in workers’ ability, expertise, and task complexity influences the answer reliability and results in biased answers. The studies [43, 49] investigate the influence of submission and worker features in inferring correct answers. The quality of answers strongly depends on the characteristics of both workers and tasks. However, according to recent studies, most of the workers only participate in a small fraction of tasks, and the collected submissions are sparse [4, 21]. Therefore, the task aggregation mechanisms that make use of the submission features should consider this as well.

In general, tasks are classified into structured and unstructured tasks [34]. A task is classified as structured when there is a well-defined form for answers. Examples are label classification and sentiment analysis. For unstructured tasks, a well-defined solution does not exist. Also, to accomplish such tasks, workers should possess some creative and exceptional skills. Examples include article writing, software code
development, and transcription services. Existing methods for task aggregation based on expertise information are intended for structured tasks and use the features such as worker’s behavior, task difficulty, and feedback. Hence they are not suitable for unstructured tasks. In general, unstructured tasks are more diverse and do not have gold-standard data. Hence, it is essential to utilize both workers and answer specific features for aggregating tasks meant for unstructured submissions [17].

Owing to the difficulty and complexity in answer aggregation, existing methods use part of the information from workers and tasks with assumptions for inferring the answers. Several system-oriented approaches for structured tasks have been proposed for achieving high accuracy in task aggregation, such as [11, 13, 31, 48]. They account for features such as worker-reliability, community, task difficulty, quantitative and classification claims. The studies on the stability of the crowdsourcing platforms indicate the importance of incorporating features such as worker-ability and trust [10]. Furthermore, the methods that work for structured tasks cannot be applied to unstructured tasks due to their diversity. The prior works overcome this issue by reviewing the answers with another set of expert workers. However, this additional crowdsourcing review increases the monetary cost and latency. Therefore, we propose an answer aggregation method that is compatible with both structured and unstructured submissions. In addition to this, the satisfaction of the requesters also depends on the cost of the task. They prefer to have good quality as well as a minimum cost.

The main objectives of our proposed work are i) inferring high-quality answers for general crowdsourcing tasks, ii) to overcome the issue of answer sparsity for improving the accuracy by incorporating similarity in answers, and iii) minimizing the cost. Specifically, we use an answer similarity-based method for aggregating the relevant answers. Hence, this work focuses on inferring the most appropriate and reliable answers based on worker features such as ability, expertise, and trust, along with the task-easiness degree. This is helpful in addressing the worker’s inconstancy while retrieving high-quality answers. Moreover, using answer similarity for expertness estimation is beneficial for aggregating unstructured tasks as well as the new workers to get the answers selected. The requester’s feedback on the past submission history is also included to estimate the reliable answers. The proposed method aggregates answers to maximize accuracy and quality. While exactly optimizing to meet the objective is difficult, we use an iterative probabilistic method followed by parameter estimation for maximizing it.

In an earlier work [17], we have presented a task aggregation method that uses an iterative probabilistic approach based on the reliability and requesters’ feedback. It yields better performance regarding the accuracy and Mean Average Precision. Nevertheless, we have not considered an expertness estimation method that works for general crowdsourcing tasks. Also, we have not addressed the problem of answer sparsity. We observe that incorporating more worker and task features improves performance. Besides, it does not consider minimizing the cost of tasks.

This work makes the following contributions. We improve the existing answer aggregation approaches by predicting the correctness of answers. The proposed method aggregates general crowdsourcing tasks including structured and unstructured submissions. It uses the similarity of submissions, worker’s trust, and expertness. The submission similarity alleviates the sparsity in answers and improves answer selection chances of new workers. Furthermore, a method is proposed to minimize the cost of tasks. It enhances the quality of aggregated answers as well. We compare the proposed method with several task aggregation approaches. The experimental results confirm its effectiveness.

To support our answer aggregation method, we propose a solution for the truth inference on the crowdsourced answers. For this, we use the probability distribution to characterize the workers. It helps in predicting the chances of a new worker answering the task. The rest of the paper is organized as
follows. Section [I] discusses the previous works on task aggregation. Section [II] describes the proposed method, and the parameter estimation is explained in Section [IV]. Section [V] deals with the experimentation study. We conclude the paper in Section [VI] with directions for future research.

II. RELATED WORK

One of the main challenges faced by crowdsourcing platforms is its quality control. Aggregation methods are necessary since most of the crowdsourcing platforms allow to assign the same tasks to multiple workers. Certain crowdsourcing platforms and frameworks such as [8 15 24 40 51] are developed to perform effective quality control mechanisms. They have developed models that differ from the common platforms and frameworks such as AMT, Topcoder, and Figure Eight, and have adopted optimization strategies for attaining high accuracy. However, these models are designed for specific types of tasks or workers, which is not otherwise possible in common platforms. Unlike these works, our focus is on approaches that can be applied to common crowdsourcing platforms and general tasks that improve aggregation accuracy.

Review articles on aggregation techniques such as [11 13 50] emphasize various issues that need to be addressed. Y. Zhao et al. [50] suggested the need for an evaluation worker group contains spammers. They have developed models that differ from the common aggregation methods that infer the true answers. In most of the works, classical aggregation model is proposed by Dawid et al. [3] has proposed a two-stage workflow for unstructured tasks in which additional crowdsourced review is performed. Another work by the same authors [39] use a pairwise comparison along with the two-stage evaluation procedure for aggregation. These approaches requisite the crowd to do extra workloads. Even though they are useful methods, it incurs an additional cost and latency in task execution.

Raykar et al. [37] has proposed a probabilistic approach for supervised learning in the absence of ground truth. They use an iterative approach for estimating the ground truth and refined the truth estimation process based on the performance in each iteration. Certain methods consider features of workers and tasks as the parameters for aggregating answers. These methods do not consider the cost of the tasks and proper budget allocation which is essential for a good aggregation mechanism. Besides, these methods are not good enough for general unstructured tasks.

Existing aggregation methods for unstructured tasks do not use auxiliary information. Such methods use responses for tasks given by workers. Baba et al. [3] has proposed a two-stage approach, for finding the quality of tasks. However, it uses only the worker specific parameters and does not consider the task features. Venanzi et al. [44 45] use Bayesian probabilistic model that measures the worker accuracy and the correct answer.

Some other approaches use the auxiliary information such as worker features and task features [22]. A worker similarity-based probabilistic model is used by Li et al. [23] which classifies the workers as experts and nonexperts and selects tasks of expert workers. The disadvantage of such a method is that the submissions of new workers get less priority even if they are of high quality. They do not consider the variation in worker abilities and strengthen only the expert workers, thus cause a cold-start problem. A neural network model that uses unstructured data is described in [27]. Zheng et al. [51] deploy a probabilistic method. Nevertheless, we diverge from the prior works by devising an answer aggregation model for generic crowdsourcing that includes structured and unstructured submissions. We use the similarity among the answers for expertise estimation that improves accuracy as well.

From the literature study, we observe that prior works on general aggregation mechanisms use simple tasks with a small set of workers and submissions to evaluate worker performance based on an exact match or gold standard. Moreover, the probability based approaches are not addressing the data sparsity in submissions. Also, they do not address the cold start problem. We notice that the new workers have a significant role in providing high quality submissions. Also,
the aggregation of unstructured tasks involves very large submission space, and workers are less likely to give identical submissions for the same task. For example, there are multiple ways to write an article on the same topic or to translate a sentence. Hence it is difficult to find an exact match and may have many acceptable submissions. Therefore, aggregating unstructured tasks is still an open problem that needs to be addressed.

In this paper, we propose a task aggregation method that addresses the problems mentioned above. We use methods for selecting the best available submissions for each task and for improving the scalability. In particular, a probabilistic approach followed by maximum likelihood estimation is devised to aggregate the general crowdsourcing tasks. The proposed method addresses the data sparsity problem using auxiliary information. The worker-ability, trust and expertise information along with the requester feedback are used for estimating high-quality answers. For enhancing the quality and for addressing the cold-start problem, similarity information of the submissions is utilized.

III. THE PROPOSED TASK AGGREGATION METHOD

An answer aggregation method intends to infer the correct answers for a set of tasks from the workers submissions. The proposed method essentially figure out the correct submissions using the submission features along with task and worker features. This helps in aggregating unstructured submissions. In order to reduce the data sparsity we use the submission features. We use a probability-based method that utilizes the parameters such as expertness, worker-ability, trust-factor, and task-easiness. The parameter expertness measures the similarity in submissions.

The proposed method uses Expectation-Maximization (EM) to estimate the parameters and hidden variables that provide the maximum likelihood. Even though, there are prior works that use the EM approach, we propose a new probability-based approach for the following reasons. We model both worker and task features such as expertness, ability, trust, and task-easiness. The auxiliary information from the submissions are used in EM to alleviate data sparsity. We deploy a worker model that uses the answer similarity for expertness estimation. Further, it helps the new workers to improve their success rate as well. This eases the prediction on behaviors of new and forthcoming workers in real crowdsourcing platforms. Fig. 2 illustrates the proposed aggregation method.

A. THE PROBLEM

Consider a crowdsourcing system with set of workers \( I \), set of tasks \( J \), set of submissions \( K = \{ k_{ij} \} \), where \( k_{ij} \) is the submission of worker \( i \) for the task \( j \) for which the correct answer is to be predicted and set of correct submissions \( S = \{ s_{ij} \} \) for each \( j \in J \). From \( K \) we identify the participating workers \( I_i \) who provide answers for the tasks. There are \( T_c \) categories of tasks that include both structured and unstructured. We assume that each task belongs to exactly one of them.

Given a set of submissions \( K \) produced by workers \( I \) for a set of tasks \( J \), we aim to infer the set of correct submissions \( s_{ij} \), for each task \( j \), such that the quality is maximized at a minimum cost. Each response \( k_{ij} \) is featured by the worker \( i \), worker reputation, task submission time, task type (e.g., article writing, sentiment analysis), task description, task-easiness, and the requesters’ feedback on previous submissions. Table. 1 tabulates the notations used.

The goal of our work is to resolve the disagreement among the submissions of workers. We examine the aggregated answers for its validity, quality, and accuracy. For this purpose, we incorporate the workers’ ability, trust, expertise estimation, and task-easiness for inferring the correct answers. In the first step, features required for the task aggregation are extracted from the available information. Then, these features are used for estimating the quality of submission using a prediction method.

Task and worker features

We consider two sets of features that are relevant for our method that characterize the quality of submissions. Firstly, the worker features which include the number of participated tasks, number of selected tasks, ratio of selected tasks, reputation and total reward earned. These features are used for predicting the worker-ability parameter. The task features include length of task description, submission delay, task submission version number, and reward. The above features contribute for estimating the parameter task-easiness.

In addition, features pertaining to worker-requester interactions are also relevant in our method. There are two significant interaction features, worker participation in tasks and answer selection by requester. Among this, the most important feature that correlates to quality is answer selection. Hence we use the requester feedback to get these features.

Let \( \lambda \in \{0, 1\} \) represent the worker feature matrix in which
the $\lambda_{in}$ is the $n^{th}$ feature of the worker $i$ and $\Delta \in \{0, 1\}$ be the answer feature matrix in which $\Delta_{jm}$ denotes $m^{th}$ feature of the submission $J$. We use these features during the parameter estimation as auxiliary information.

**THE PREDICTION METHOD**

The aggregation method for inferring correct answers works as follows. The probability of a worker $i$ giving accurate submissions relies upon worker-ability, expertise, trust, and task-easiness. The workers’ ability represents the potential of a worker to give reliable answers. It is a measure of reliability based on history. A logistic function is used to measure the probability of giving correct answers. Note that expertise is computed using similarity in answers. Intuitively, a worker is able to give the correct answer or win a reward only if he is an expert and reliable. The worker-ability is estimated using the worker’s past activities and submission history only, while the expertise represents the similarity in submissions. In addition to this, it is influenced by task-easiness. Each task $j \in J$ is associated with task-easiness $T_e, j \in [0, 1]$ which is a measure of the proportion of workers who have enough skills for correctly solving $J$. $T_e$ measures the toughness of a task.

Using the prediction score of the above parameters we infer the probability of correct answer $P_c$. Note that when $P_c = 1$, $k_{ij}$ and $s_{ij}$ will have equal values. Let $A_{ij}$ represent the ability of a worker and $A_{ij} \in [0, 1]$ which is the probability that a worker has the potential to give correct answers. $t$ represents the trust-factor of worker $i$ where $t_i$ is the probability that the worker is not a spammer. Note that, spammers give random answers with less quality.

At first, we estimate worker expertise using the similarity of answers. Then worker-ability, trust-factor, expertise, and task-easiness are used for predicting the correctness of answers $P_c$. Hence $P_c$ depends on the joint probability of i) worker-ability, trust-factor, and expertise and ii) task-easiness degree. Thus the probability of a submission $k_{ij}$ is given by

$$P_c[k_{ij} = s_{ij} | A_i, t_i, e_i, T_e] = \begin{cases} W T_e, & \text{if } k_{ij} = s_{ij} \\ 1 - W T_e \\ T_e - 1, & \text{otherwise} \end{cases}$$

where $W$ is the worker specific parameter which indicate the probability that a worker is reliable and expert. We duly note that the occurrence of the events $W$ and $T_e$ are independent. Hence, the probability of the answer is correct is given by $W T_e$. For the rest of the answers those are incorrect, we choose them with equal probability to occur. Apparently, $\{1 - W T_e\}$ is the probability that the answer being incorrect.

**PREDICTING WORKER-RELIABILITY**

In this section, we compute the worker specific parameter $W$. At first, we estimate the worker reliability parameter $R$. The reliability of a worker depends on his ability to give reliable answers as well as the correctness of the current submission. We compute reliability from the workers’ past submission history and similarity between the current submissions. Another parameter that affects the worker reliability is the trust-factor, which is computed from the requesters’ feedback. Thus, $R$ represents the probability of a worker giving reliable answers, which is influenced by the worker features such as worker-ability $A_{ij}$, trust-factor $t$, and expertise $e_i$. Hence, $R$ is represented as a combination of these parameters as

$$R_{ij} = m_1 A_{ij} + m_2 (t_i) + m_3 e_i$$

---

**FIGURE 2:** The proposed task aggregation method receives history, task descriptions, and similarity in answers as input to predict worker-reliability and submission accuracy.
Here we consider the features are equally likely, hence \( m_1 = m_2 = m_3 = 1 \). Therefore,

\[
R_{ij} = (A_{ij} + ct_i) + e_i
\]  \hspace{1cm} (3)

The trust factor is added to ensure the quality of submissions based on the feedback of requesters. It is based on the assumption that if a worker provides trustworthy information frequently, he is highly reliable. \( c \) is the weight assigned to the trust-factor based on the correctly given submissions. Intuitively, \( R \) indicates the probability that worker \( i \) is reliable and has enough expertise to give the correct answer. \( A_{ij} \) indicate \( i \) is able to perform a task which is estimated in the EM-step. \( e_i \) is the probability that \( i \) is an expert. \( t \) strengthens the worker-reliability by incorporating the trust-factor. A value ranges between \([0,1]\) is assigned for \( c \) based on the quality of past submissions (accepted or not). This improves the acceptance of answers. \( e_i \) is an independent feature hence multiplying with the other parameters.

The prediction score of a workers’ reliability is given by \( R_{ij} \) which is influenced by the workers’ potential to give the correct answer. Hence the probability that the worker is reliable and expert is computed as a logistic function

\[
W = P(W = 1) = \frac{1}{1 + \exp(-R_{ij})}
\]  \hspace{1cm} (4)

We use the priors of \( S \) as \( \pi(s) \) to represent the prior probability of an answer belongs to the list of correct submissions \( \{s_{ij}\} \), and \( \sum_{s_{ij} \in S} \pi(s) = 1 \). The maximum likelihood of observing \( S \) and \( K \) is given by

\[
P(K \mid \theta, S) = P(K \mid A, t, Te, S) P(S \mid \pi)
= \prod_{j \in J} \prod_{\pi_s} \prod_{k_{ij} \in K} P(k_{ij} \mid a_i, t, Te, s_j)
\]  \hspace{1cm} (5)

Fig. 3 depicts the graphical representation of the prediction method. \( \pi, W, \) and \( Te_j \) are the parameters of the proposed method. \( S_j \) is the hidden variable and \( K_j \) is the observed variable. That is the observations of the submissions \( K_j \) for a task \( j \) depends on the parameters task-easiness \( Te_j \), hidden variables \( S_j \), probability of a worker \( i \) is reliable and expert \( W_i \), and the priors \( \pi \).

**ESTIMATING THE WORKERS’ EXPERTNESS**

The computation of the expertness parameter is challenging since we need to handle unstructured tasks also. Recall, existing methods deal with structured tasks only. Here, we have to derive better representations for the unstructured tasks that contain unstructured data. Hence, we convert this unstructured data to structured data using an entity-based representation [9][11]. In this, a set of entities are extracted from each answer \( k_{ij} \in K \) that represent the original answer in which an entity is a word relevant for that particular task or answer. We compare words with the entities in the ontology for entity extraction. In case that a word from an answer is available in the dictionary, then that word is added into the entity set for this answer. Using the entity-based representation of data, the unstructured data is converted to structured representation. Therefore, data that have similar meanings are mapped into similar or even the same representations. For example, “Crowdsourcing is the process of outsourcing task” is converted to an entity set \(<\) crowdsourcing, process, outsourcing \(>\), so these are three concepts. Therefore, entity-based representation helps in grouping the answers with similar meanings.

The submissions for an unstructured task may contain multiple correct answers. These answers may draw some correlations among them [25]. This violates the very assumption of single truth value in task aggregation methods for structured tasks [35]. To address this problem, we calculate the similarity score that is the similarity between answers provided by the workers. We consider the answers are correct when the similarity score is greater than a threshold value. The threshold value is defined as the average similarity score which decide whether a submission is correct or not. If the computed similarity score is higher than that we consider the answer is correct and corresponding worker is assigned a higher expertness value based on the similarity score. At first, we represent the answer entities using the word embedding technique [20][27], in which each word (or document) represents a real-valued word vector. Hence, the vector representation of words is procured by training on a large corpus without any syntax analysis or labeling. The word embedding methods automatically learn similar real-valued vectors for similar words. Thus, it is easy to compute the similarity of the answer entities as a real value. Then the similarity between answers is used for computing the expertness parameter. We use the Cosine similarity [20] for this computation. Cosine similarity measures the angle between two vectors and returns a real value between 0 and 1. Then the similarity between two answers \( k_x \) and \( k_y \) is given by

\[
sim(|k_x, k_y|) = \frac{\sum_{i} t_1^i * t_2^i}{|t_1| * |t_2|}
\]  \hspace{1cm} (6)

where \( t_1 \) and \( t_2 \) are the vectors representing the topic associations of answers \( k_x \) and \( k_y \), \( t_1^{(i)} \) and \( t_2^{(i)} \) represent the number of terms in \( k_x \) and \( k_y \) respectively, which are associated with the topic \( i \). Here we choose cosine similarity since it is independent on document length.

A value “1” for \( sim(|k_x, k_y|) \) indicates that the answers are identical and a value “0” indicates that the documents are totally distinct. The values between 0 and 1 represent the
degree of similarity between the answers. We compare for all possible answers of a task and $e_i$ is computed as

$$e_i = \frac{\sum_{k_x, k_y \in K, k_x \neq k_y} \text{sim}(k_x, k_y)}{K - 1}$$

(7)

Comparing with all possible answers, the expertise of a worker is more if the answer is supported by other similar answers. This kind of expertise estimation helps in two ways. Firstly, it improves the prediction of worker-reliability since the similarity of correct submissions is higher. Secondly, if new workers submit correct answers, then the similarity among them will be high. Subsequently, this will improve their expertise parameter that is helpful in later predictions.

Indeed, expert levels will be different from worker to worker. Intuitively, a professional worker has a high $e_i$ value, and an inexperienced or a new worker bears a low $e_i$ value. Also, since the workers may attempt only a limited number of tasks, sparsity will increase when the dataset is bigger. Hence, we use the cosine similarity since it has low complexity on sparse vectors and works on null dimensions.

IV. PARAMETER ESTIMATION FOR PREDICTION

In our work, we aggregate answers by maximizing the chances that the answer is correct. Hence, we use the maximum likelihood approach for estimating the parameters. We use the reliability influencing parameters and prior information. To tackle the data sparsity, the feature vectors of submissions, workers, and reliability information are used as auxiliary information. The matrix transfer learning [5] and feature-based matrix factorization are used to exploit auxiliary information.

$\theta$ is the set of parameters we have to estimate. Here $\theta$ is $\{A, t, T, e, S\}$ which are the parameters to be observed. We aim to estimate the most probable values for the parameter $\theta$ and the hidden variables $\{s_{ij}\}$, by using the observed variables $k_{ij} \in K$. More specifically, $\theta$ and $S$ are estimated that give the maximum likelihood. Therefore, we have to find

$$\hat{\theta} = \arg\max_{S,\theta} \log P(K, S, W | \theta)$$

(8)

We estimate the parameters for reliability and answer correctness prediction. Our goal is to compute the correct submissions $s_{ij}$ as well as $\theta$ which are unobserved parameters. The EM algorithm is used for inferring the unobserved parameters. As stated in [6], EM is an effective iterative process for estimating the maximum likelihood when there are missing values in the dataset. In this method, the missing data are $s_{ij}$ similar to [51]. During the iterations, we estimate the $s_{ij}$ in the E-step, and $\theta$ is updated in the maximization step. Here $\theta^{(0)} = A^{(0)}, t^{(0)}, e^{(0)}, T e^{(0)}$ denotes the initialized parameters. $\theta^{(i)} = A^{(i)}, t^{(i)}, e^{(i)}, T e^{(i)}$ indicates the parameters in the $i^{th}$ iteration. The EM procedure of our method is as follows:

E-step: The posterior probabilities $s_{ij}$ is computed by observing the submissions $K$ and $\theta$ from the maximization step. $\theta$ is a conditional independent with $s_{ij}$. The current estimate of the parameters are $\hat{\theta}^{(i)}$, for the $i+1$ iteration of E-step is given by

$$Q(\theta, \hat{\theta}^{(i)}) = E[\log P(K|W, Te, S)] = E[\log P(k_{ij}|p(k_{ij}|W, Te, S))] = \sum_{k_i} \sum_{k_j} p(k_{ij}) \log p(k_{ij}|s_{ij})$$

(9)

M-step: The objective of maximizing step is for estimating the log-likelihood of all submissions $K$ and the correct answers $S$. Hence, it is computed using the posterior probability of $P_c(k_{ij} = s_{ij}|A, t, e, Te)$ in (3) which is computed on the previous E-step. Therefore, M-step finds out $\theta$ that maximizes $Q(\theta, \hat{\theta}^{(i)})$ in the $i + 1^{th}$ iteration as

$$\theta^{(i+1)} = \arg\max_{\theta} Q(\theta, \hat{\theta}^{(i)})$$

(10)

where we infer the parameters $\theta = \{A, t, Te, S\}$ such that $\sum_{k_i} \theta_{j,k} = 1$ and $\sum_{s_j} \pi_i,k = 1$ so that (5) is maximized. Finally, we use $P_c(k_{ij})$ to estimate the correct answers $\{s_{ij}\}$, such that $s_{ij} = \arg\max P_c(k_{ij})$.

Convergence: The log-likelihood of $Q$ in Eqn. 9 is evaluated and the E-step and M-step are continued until the convergence as

$$\left\| Q(\theta^{i+1}, \hat{\theta}^{(i)}) - Q(\theta^{i}, \hat{\theta}^{(i)}) \right\| \leq t_r$$

(11)

$tr$ is the threshold and in this method, we set $tr = 0.001$.

We expect that the correct answer prediction for a submission from a worker is related to the reliability, so to compensate the sparsity in answer set we use $W$. The coordinate system transfer (CST) method for matrix factorization is used. It leverages information of auxiliary matrices to improve the prediction performance. We apply the matrix factorization on $\lambda$ and $\Delta$ this information is used to transfer the knowledge from $W$ for the prediction.

To sum up, the input to the EM inference phase are set of workers $I_i$, set of tasks $J_j$, set of answers $K_{ij}$ and the estimated outputs are task-easiness $Te$, reliability $R$ which provide the probability of $P_{ij}$. The proposed method is shown in Algorithm 1 which generates the set of inferred answers. Firstly, the task $j$ is given to the crowd and the answers $K$ are collected. Then the parameter estimation of $(\theta, \pi, A, t, Te)$ is performed in the next lines 3 to 5. The similarity of answers is calculated. The expertise is estimated using Eqn. 4 and the reliability is computed using Eqn. 5. The current estimation of inferred answers is delivered to the user in the next step. After the crowd complete the task, the collected new submissions are merged with $K$. The task $S$ with the highest $P_c$ is inferred for the task $j$.

MINIMIZING THE COST OF TASKS

One of the main attraction of crowdsourcing is its cost effectiveness. So minimizing the cost of tasks has a great significance [18][25]. Workers receive incentives for their work, but usually, the requesters have limited budgets. Besides, structured tasks such as data annotation may require multiple
Algorithm 1 Algorithm for Task aggregation method.

input : Set of tasks \( J \), workers \( I \), submissions \( K \), Category \( C \)

output : List of inferred answers \( S = \{ s_i \} \)

Collect answers \( K \) from the crowd;

for each \( k_{ij} \in K \) do

\( \theta, P_{j,k} \leftarrow \text{estimate using EM from } K \text{ and } S \)

Compute the expertness \( e_i \) as

\[ e_i = \text{sim}(k_x, k_y) = \frac{\sum_{t1} t1^*_1 * t2^*_1}{|t1| * |t2|} \]

\( t1^*_1 \) and \( t2^*_1 \) are the number of terms in answers \( k_x \) and \( k_y \)

Estimate the reliability \( R_{ij} \) using

\[ R_{ij} = (A_{ij} + ct_i) * e_i \]

Compute the probability that the worker is reliable \( W \) as

\[ W = P(W = 1) = \frac{1}{1 + \exp(-R_{ij})} \]

Output the current estimation \( P_c; \gg \text{Probability of giving correct answers} \)

Update \( K \) as \( K \cup \text{new submissions} \);
end

return \( s_i \leftarrow \arg \max_{c,s} P_c \) for all \( j \).

Algorithm 2 Algorithm for minimizing the cost of tasks.

input : Set of tasks \( J \), workers \( I \), submissions \( K \), Cost \( C_0 \)

output : Cost allocation \( C, T_r \)

Initialization : Set \( T_{e_j}, R, e \) to a random value in \([0, 1] \)

\( J_{avl} = J, r = 0, C = C - (C_0/3), i = (C_0/3)/n \)

Initial round \( r=0; \)

Allocate \( i \) workers to \( j \) in \( J_{avl} \);

Collect answers \( K \);
Estimate \( T e, W \leftarrow 1 \leq j \leq n, 1 \leq i \leq C_0/3 \) using EM method;
Set initial threshold \( T r^r \);

while \( (C > 0) \&\& (J_{avl} \neq \emptyset) \) do

\( r = r + 1, l = |J_{avl}|; \)

Allocate \( i_1^r, \ldots, i_l^r \) workers to tasks \( j_1, \ldots, j_l \) based on \( T e \);

Collect answers \( K \);

Estimate \( T e, W \) using TAM;

\[ C = C - \sum_{i \in 1, \ldots, l} |J_{avl}| s_i^r \]

Compute \( T r^r \);

for each \( j = 1, 2, \ldots, n \) do

if \( w^r \geq T r^r \) then

\( J_{avl} = J/j; \)
end
end

V. VALIDATION

In this section, we validate the effectiveness of our proposed task aggregation method through experiments. The experiments serve two purposes. First, we evaluate the proposed task aggregation method by comparing it with state-of-the-art approaches. Then we test the accuracy of cost-minimization algorithm. Experiments are carried out using empirical dataset and a framework developed in Python \[16\]. We discuss the results in Section V-E.

A. DATASET

We select three real dataset collected from Figure Eight \[11\]. The tasks which are completed (at least one answer is selected for reward) are collected and classified as following. All the dataset are considered as a set of \( (I, J, K_{i,j}, S_{i,j}) \) which indicates worker \( w_i \) provides submission \( k_{i,j} \) on task \( t_j \).

1) Data annotation: It includes worker’s annotations for a set of items such as images, audio, or text.

2) Sentiment analysis: Workers have to determine the sentiment behind a given text or image.

3) Article writing: Workers are instructed to write an article as per the given specifications.

The dataset mentioned above belong to three different categories and represent various general crowdsourcing tasks including both structured and unstructured tasks. A brief

\[ \text{Algorithm 1} \]

input : Set of tasks \( J \), workers \( I \), submissions \( K \), Category \( C \)

output : List of inferred answers \( S = \{ s_i \} \)

Collect answers \( K \) from the crowd;

for each \( k_{ij} \in K \) do

\( \theta, P_{j,k} \leftarrow \text{estimate using EM from } K \text{ and } S \)

Compute the expertness \( e_i \) as

\[ e_i = \text{sim}(k_x, k_y) = \frac{\sum_{t1} t1^*_1 * t2^*_1}{|t1| * |t2|} \gg t1^*_1 \text{ and } t2^*_1 \text{ are the number of terms in answers } k_x \text{ and } k_y \]

Estimate the reliability \( R_{ij} \) using

\[ R_{ij} = (A_{ij} + ct_i) * e_i \]

Compute the probability that the worker is reliable \( W \) as

\[ W = P(W = 1) = \frac{1}{1 + \exp(-R_{ij})} \]

Output the current estimation \( P_c; \gg \text{Probability of giving correct answers} \)

Update \( K \) as \( K \cup \text{new submissions} \); end

return \( s_i \leftarrow \arg \max_{c,s} P_c \) for all \( j \).

answers, while unstructured tasks like article writing may require only a few answers. Therefore, it poses a challenge to spend the budget efficiently. However, the task reward is distributed statically with a fixed budget \( C_0 \). That is reward is allocated to a fixed number of workers, without accounting the difficulty-level of tasks. Besides, paying equally to all workers is not fair since unstructured tasks demand more knowledge and skills. In the case of structured tasks this helps in identifying expert workers and filtering out the spammers. Hence we extend the task aggregation method by incorporating a cost model. The principle of our method is as follows: i) the cost \( C \) or the reward is zero if the worker’s reliability parameter \( W < T r \), since it indicates the worker’s overall expertness and confidence measure is low and ii) the reward is directly proportional to \( W \). The value of \( W \) is computed for each worker, who participated in the task, using Eqn. \[3\] and accordingly the remuneration is allocated.

Our method works as follows: In the first round subset of available tasks \( J_{avl} \subseteq J \) is evaluated. The answers are aggregated for these tasks using the TAM method described in section. \[11\] In the initial round \( r = 0 \), starts with the EM phase, \( i \) workers and an initial cost or incentive is allocated. The parameters \( T e \) and \( W \) are estimated using the EM approach and the corresponding submissions \( K \) are collected from the workers. The initial threshold \( T r \) is also computed. In the next round, allocate workers for each task based on the estimated \( T e \). The procedure for minimizing the cost of tasks is explained in Algorithm \[2\].
We randomly pick 80% of tasks as training set and 20% tasks for testing. The experiments are repeated for 20 runs and the average performances are noted in terms of accuracy and mAP. Each task category contains an average of 31.1, 45.0, and 51.3 responses for sentiment analysis, data annotation, and article writing, respectively. The features such as $s_{i,j}$, $s_{i}$, $n_{i}$, and $R_{i}t$ are transformed to log scale to overcome the problem of large feature span. Then they are normalized to the range of $[0, 1]$ using the min-max normalization.

Fig. 4 illustrates the worker accuracy distribution. The dataset is sorted in descending order concerning the mean accuracy of the workers in each type of task. The performance of all the methods is evaluated. We observe that accuracy in data annotation is lower than that of other tasks.

1) Predicting the reliability and expertness of a worker
To evaluate the performance of our aggregation method in predicting correct answers, we calculate the accuracy in terms of $W$ (the probability that the worker is reliable and expert). The number of tasks is varied from 10 to 50 and submissions are considered for 5 discrete levels between 15% and 90%. Fig. 5 shows the predictions for 5 runs. This result indicates that our approach infers $W$ correctly since our method gets the leverage from features of workers and submission similarity. As can be seen, the accuracy increases with the number of tasks and percentage of submissions, which indicate that features of workers such as worker-ability, trust-factor, and expertness are beneficial for finding reliable workers.

2) Impact of the expertness computation
We have used a similarity computation to quantify the similarity among answers so that it is useful for prediction of $W$. We experimentally demonstrate the importance of expertness parameter in correlating similar answers. In order to learn the vector representations of entities, we have used a large corpus and Word2vec package is used for training the vector representations of words. We set the dimension of the vectors as 100, and the minimum occurrence count as 5.

Fig. 6 shows the results of expertness estimation. The most prominent feature we observed is the increase in similarity as the number of embeddings, that is the number of word vectors increase. The figure shows that it is stable, and performance increases as the size increases. Besides, all values are greater than 0.5. It is significant in the case of unstructured tasks where it includes a larger size of embeddings. The values obtained for data annotation tasks are higher than that of other tasks. Relatively low values obtained for the article writing dataset are because of the null pairwise values generated. When their dot product is null, it affects the similarity value. The values favor the use of an expertness estimation approach that captures the similarities between the answers.

Fig. 7 illustrates the effect of expertness estimation more clearly. We compute the reliability without expertness parameter $e_{i}$ and with $e_{i}$ for a set of tasks for five runs. In the first case, where $e_{i}$ is not considered, they have very different values for $W$. However, $W$ with $e_{i}$ corrects this and gives steady results. In our experiments we observe that setting threshold as 0.8 gives good accuracy. Moreover, incorporating $e_{i}$ significantly improves the prediction of $W$ by successfully modeling the correlation among answers.

E. RESULTS
In this section, we discuss the results of the experiments on various dataset. Initially, we evaluate various features for understanding their impact on the quality of submissions. The performance of submission features (SFs), worker features (WFs) and a combination of both, are evaluated using mAP.
TABLE 2: Details of dataset

| Data set          | Requesters | Workers | Tasks | Answers | Selected answers |
|-------------------|------------|---------|-------|---------|------------------|
| Data annotation   | 10,895     | 15,129  | 13,079| 652,643 | 9,205            |
| Sentiment analysis| 45,378     | 13,198  | 56,805| 9,81318 | 52,612           |
| Article writing   | 3,390      | 8,910   | 3,136 | 109,783 | 4,189            |

FIGURE 4: Distribution of worker accuracy on three different dataset. They are sorted according to the mean worker accuracy. Data annotation tasks have relatively low accuracy compared to other tasks.

(a) Data annotation (Mean worker accuracy = 0.695)
(b) Sentiment analysis (Mean worker accuracy = 0.839)
(c) Article writing (Mean worker accuracy = 0.865)

FIGURE 5: Predicting a worker is reliable and expert (W) in terms of number of tasks and percentage of submissions.

(a) Number of tasks versus prediction of W
(b) Percentage of submission versus prediction

1) Data annotation

Table 3 shows the mAP results of various features on data annotation set. We observe that the relative performance of different feature sets varies. In particular, the performance is better when SFs are used for experiments. This could be due to the fact that SFs capture the dynamics and diversity of the responses. Subsequently it is noticed that a combination of feature sets improve the performance even better. Therefore, we use the SF+WF feature set for further experiments. Table 3 shows the results of various methods in terms of mAP values and accuracy. We observe that methods such as TAM, DSM, and WSM those use object features achieve better performance than that of the models like MV, LoR and BGM that use only submission features. As object features bring additional

TABLE 3: mAP results of feature set on data annotation task

| Features  | mAP   | Rank |
|-----------|-------|------|
| SF + WF   | 0.8104| 1    |
| SF        | 0.7431| 2    |
| WF        | 0.4206| 3    |
information, this seems to be obvious to model these features for inferring correct answers. Furthermore, WSM and TAM have comparable performance than other methods. This shows that generative models are feasible for utilizing object features and submission information. TAM has a lower performance compared to WSM because it uses similarity information for expertness estimation. Hence, it has limitations on this kind of dataset in which similarity estimation has less to perform. Whereas, WSM uses a confusion matrix to represent each worker. However, the use of confusion matrix incurs more memory space. When the mean accuracy of workers is less and data quality is low, TAM tries to infer correct answers from low-ability workers and hence cause a slight variation in accuracy. Besides, TAM leads to convenience in modeling of workers characteristics and prediction of new workers success and reliability and hence the small difference in the performance is negligible.

For data annotation tasks, workers require only common linguistic knowledge that most of them are assumed to have [22]. The accuracy is higher compared to other kind of tasks, since the number of experts and participation of workers are relatively more. The results show that TAM effectively use worker-specific features that are significant to the crowdsourcing system. The results substantiate the relevance of our method.

2) Sentiment analysis
Table. 5 shows the mAP results on sentiment analysis. As stated in section. V-E1 the SF+WF features give better performance, hence we use the combination of features. Table. 6 shows the results obtained using various methods. Like data annotation tasks, sentiment analysis also has more experts, and hence TAM gives better results than all other approaches. Though DSM and WSM also give good results, TAM is superior. This shows that an aggregation method based on the reliability and expertness can give good performance on structured submissions.

3) Article writing
In Table. 7, we show the mAP results of various features evaluated on the article writing dataset. It is evident that

| Methods | mAP | Accuracy |
|---------|-----|----------|
| MV      | 0.7104 | 0.7962 |
| LoR     | 0.7612 | 0.7570 |
| TAM     | 0.8303 | 0.8513 |
| DSM     | 0.8014 | 0.8182 |
| BGM     | 0.7903 | 0.7973 |
| WSM     | 0.8101 | 0.8197 |

TABLE 7: mAP results of feature set on article writing

| Features | mAP | Rank |
|----------|-----|------|
| SF + WF  | 0.6021 | 1    |
| SF       | 0.5604 | 2    |
| WF       | 0.4153 | 3    |
We observed that the average performance of our method is this shows that in the case of unstructured submissions it shows comparable performance in data annotation tasks. The performance of various methods for cost and accuracy are given in Figures 9a and 9b, respectively. It is observed that our method performs better for both metrics. For data annotation and sentiment analysis, MV gives the worst performance in terms of cost incurred, whereas DSM and BGM show similar performance. This is because MV is a static approach which allocates the budget to the workers at the initial stage of crowdsourcing process itself.

While considering article writing, the cost incurred for DSM and BGM algorithms is more or less equal. The number of workers and tasks is less compared to other datasets. The methods MV, DSM, and BGM, have shown a similar performance. Further, Extended TAM outperforms in terms of accuracy for all the dataset by incurring a smaller cost. Therefore, we could reduce the cost of tasks by incorporating worker reliability and difficulty level to our method.

2) Model parameters
We also investigate the effect of model parameters on task aggregation. For this purpose, we have conducted experiments by varying number of latent features from 8 to 256 and recorded the corresponding mAP values. It is observed that mAP values are almost stable in the range of 0.822 to 0.839 (ref. Fig. 10). Note that the variation in mAP values is as less as 2% compared to a significant change occurred in the number of parameters.

3) Comparison of running time
Fig. 11 depicts the running time for the task aggregation process for all methods. The X-axis shows the number of tasks, and the Y-axis is the time (ms) taken for aggregation. MV has the least running time among all the compared aggregation methods. TAM and WSM have the highest running time (both are intended for general tasks). TAM takes more time since it includes the computation of $e_i$, but as the number of tasks increasing the difference in the running time is less. However, the accuracy is more for our algorithm.

4) Scalability
For examining the scalability of the task aggregation method, we use varying number of tasks represented as $N$ which takes values from the range 10K to 100K. Fig. 8 shows the mAP values of the methods. It is observed that the different values of $N$ does not affect the results of TAM and gives stable results. This suggests that TAM is scalable. Moreover, as $N$

| Methods | mAP  | Accuracy |
|---------|------|----------|
| MV      | 0.6310 | 0.731 |
| LoR     | 0.6706 | 0.812 |
| TAM     | 0.8402 | 0.9172 |
| DSM     | 0.7315 | 0.8545 |
| BGM     | 0.7338 | 0.7989 |
| WSM     | 0.8076 | 0.8987 |

F. DISCUSSION
We observed that the average performance of our method is higher. Most importantly, TAM achieves better performance than other methods in logo design and article writing while it shows comparable performance in data annotation tasks. In comparison with the state-of-the-art methods, our method has improved a 4.0% in precision and a 2.0% in accuracy on data annotation and by 2.02% precision and 7.91% accuracy on sentiment analysis and by 4.5% precision and 2.97% accuracy on article writing. This significant improvement clearly demonstrates that our approach is good in aggregating both structured and unstructured tasks. Also, it combines the features of both workers and submissions in a better way, to make the method robust.

FIGURE 8: mAP values of the aggregation methods for different N values

combining the worker and submission features is helpful for unstructured tasks as well. Table 8 shows the results of comparison with other baselines. We observe that for unstructured tasks, TAM achieves better performance. Even though, the number of experts and the number of tasks are less in article writing tasks, TAM yields better accuracy. This shows that in the case of unstructured submissions methods that model both the worker specific and task specific parameters perform better. Also our method, TAM works is superior for unstructured submissions.

TABLE 8: Performance comparison of various methods for article writing

1) Minimizing the cost
We compare our cost minimizing method (Extended TAM) with MV, DSM, and BGM. In case of MV, DSM, and BGM, we could use a round-robin allocation strategy for allocating $C$. However, in Extended TAM, we need to consider the variables such as initial cost $C_0$, number of rounds $r$, and threshold $T_r$. Initially, $C_0$ is uniformly allocated and $W$ and $T_e$ are estimated in each round $r$. Then, based on the values of $W$ and $T_e$, $C$ is allocated. The results are given in Fig. 9. When the number of experts and the number of tasks is less compared to other datasets.

While considering article writing, the cost incurred for DSM and BGM algorithms is more or less equal. The number of workers and tasks is less compared to other datasets. The methods MV, DSM, and BGM, have shown a similar performance. Further, Extended TAM outperforms in terms of accuracy for all the dataset by incurring a smaller cost. Therefore, we could reduce the cost of tasks by incorporating worker reliability and difficulty level to our method.

FIGURE 11: Running time for the task aggregation process for all methods.
more useful for improving the inference process. We observe that for structured tasks, expertness estimation based on the submission similarity has less of a role. However, in other two dataset, it has high accuracy which shows the effectiveness of the proposed method in answer aggregation and truth inference. These observations demonstrate that the proposed task aggregation method achieve high accuracy overall and is competent for both structured and unstructured submissions.

VI. CONCLUSION

Task aggregation has a strong impact on crowdsourcing. The problem of aggregating structured and unstructured submissions in crowdsourcing systems has been studied. We have proposed an aggregation method that estimates the quality of submissions using the similarity of submissions, workers’ reliability, expertness, and difficulty level of tasks. Our solution approach is comprised of two phases. Initially, the probability that a worker is reliable is estimated using the parameters such as worker-ability, trust-factor, and expertness. Based on the worker reliability and difficulty level of tasks, the quality of submissions is estimated using a probability model. The EM approach is used for estimating the parameters for improving the quality of the results. Secondly, a method for minimizing the cost of the task is presented. The proposed task aggregation method is compared with various state-of-the-art techniques. The results have demonstrated the effectiveness of the approach for estimating the quality of submissions. The results also confirmed the necessity of worker features and submission features to infer reliable answers. Notably, the similarity in submissions is useful for enhancing the quality of unstructured submissions and restraining the cold-start problem.

The primary focus of our research is for improving the unstructured submissions. In that, we have succeeded to some extent in achieving better results. However, our method has certain limitations. For instance, the threshold for cost minimization algorithm could be settled in an adaptive way. Also, more insight is required for estimating the expertness value using various similarity approaches. In future research,
generalizing our approach for other types of crowdsourcing tasks would be an interesting problem to investigate.

REFERENCES

[1] Figure eight. https://www.figure-eight.com/, Last accessed: 2021-02-05.
[2] V Kishore Ayyadevara. Word2vec. In Pro Machine Learning Algorithms, pages 167–178. Springer, 2018.
[3] Yukino Baba and Hisashi Kashima. Statistical quality estimation for general crowdsourcing tasks. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 554–562. ACM, 2013.
[4] Florian Daniel, Pavel Kucherbaev, Cinzia Cappiello, Boualem Benatallah, and Mohammad Allahbakhsh. Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. ACM Computing Surveys (CSUR), 51(1):1–40, 2018.
[5] AP Dawid. Maximum likelihood estimation of observer error-rates using the em algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1):20–28, 1979.
[6] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society: Series B (Methodological), 39(1):1–22, 1977.
[7] Djellal Eddine Difallah, Gianluca Demartini, and Philippe Codre-Mauroux. Scheduling human intelligence tasks in multi-tenant crowd-powered systems. In Proceedings of the 25th international conference on World Wide Web, pages 855–865, 2016.
[8] Ju Fan, Guoliang Li, Beng Chin Ooi, Kian-lee Tan, and Jianhua Feng. icrowd: An adaptive crowdsourcing framework. In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, pages 1015–1030, 2015.
[9] Ronen Feldman, James Sanger, et al. The text mining handbook: advanced approaches in analyzing unstructured data. Cambridge university press, 2007.
[10] Joseph Feller, Patrick Finnegan, Jeremy Hayes, and Philip O’Reilly. Orchestrating sustainable crowdsourcing: A characterisation of solver brokerages. The Journal of Strategic Information Systems, 21(3):216–232, 2012.
[11] Jing Gao, Qi Li, Bo Zhao, Wei Fan, and Jiawei Han. Truth discovery and crowdsourcing aggregation: A unified perspective. Proceedings of the VLDB Endowment, 8(12):2048–2049, 2015.
[12] Antonio Ghezzi, Donata Gabelloni, Antonella Martini, and Angelo Natalicchio. Crowdsourcing: a review and suggestions for future research. International Journal of Management Reviews, (2017), 1468-2370, 2017.
[13] Nguyen Quoc Viet Hung, Nguyen thanh Tam, Lam Ngoc Tran, and Karl Aberer. An evaluation of aggregation techniques in crowdsourcing. In International Conference on Web Information Systems Engineering, pages 1–15. Springer, 2013.
[14] Jiuchuan Jiang, Bo An, Yichuan Jiang, Donghui Lin, Zhan Bu, Jie Cao, and Zhifeng Hao. Understanding crowdsourcing systems from a multiagent perspective and approach. ACM Transactions on Autonomous and Adaptive Systems (TAAS), 13(2):8, 2018.
[15] Liangxiao Jiang, Hao Zhang, Fangna Tao, and Chaoqun Li. Learning from crowds with multiple noisy label distribution propagation. IEEE Transactions on Neural Networks and Learning Systems, pages 1–11, 2021.
[16] Ayswarya R Kurup. task-aggregation. https://github.com/ayswaryarkurup/task-aggregation.git, 2020.
[17] Ayswarya R Kurup and G P Sajeev. Aggregating unstructured submissions for reliable answers in crowdsourcing systems. In 2019 9th International Symposium on Embedded Computing and System Design (ISED), pages 1–7, 2019.
[18] Ayswarya R Kurup and G P Sajeev. Analysing the characteristics of crowdsourcing platforms for improving throughput. International Journal of Web Engineering and Technology, 14(3):255–279, 2019.
[19] Ayswarya R. Kurup and G.P. Sajeev. A task recommendation scheme for crowdsourcing based on expertise estimation. Electronic Commerce Research and Applications, 41:100946, 2020.
[20] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. Advances in neural information processing systems, 27:2177–2185, 2014.
[21] Guoliang Li, Jiannan Wang, Yudian Zheng, and Michael J Franklin. Crowdsourced data management: A survey. IEEE Transactions on Knowledge and Data Engineering, 28(9):2296–2319, 2016.
[22] Hongwei Li, Bo Zhao, and Ariel Fuxman. The wisdom of minority: Discovering and targeting the right group of workers for crowdsourcing. In Proceedings of the 23rd international conference on World wide web, pages 165–176. ACM, 2014.
[23] Jiyi Li, Yukino Baba, and Hisashi Kashima. Incorporating worker similarity for label aggregation in crowdsourcing. In International Conference on Artificial Neural Networks, pages 596–606. Springer, 2018.
[24] Shao-Yuan Li, Sheng-Jun Huang, and Songcan Chen. Crowdsourcing aggregation with deep bayesian learning. Science China-Information Sciences, 64(3), 2021.
[25] Yuliang Li, Chaochun Liu, Nan Du, Wei Fan, Qi Li, Jing Gao, Chenwei Zhang, and Hao Wu. Extracting medical knowledge from crowdsourced question answering website. IEEE Transactions on Big Data, 6(2):309–321, 2020.
[26] Yung-Shen Lin, Jung-Yi Jiang, and Shie-Jue Lee. A similarity measure for text classification and clustering. IEEE transactions on knowledge and data engineering, 26(7):1575–1590, 2013.
[27] Jiacheng Liu, Feilong Tang, Long Chen, and Yannin Zhu. Exploiting predicted answer in label aggregation to make better use of the crowd wisdom. Information Sciences, 574:66–83, 2021.
[28] Qiang Liu, Jian Peng, and Alexander T Ihler. Variational inference for crowdsourcing. In Advances in neural information processing systems, pages 692–700, 2012.

[29] Shanshan Lyu, Wentao Ouyang, Huawei Shen, and Xueqi Cheng. Finding high-quality unstructured submissions in general crowdsourcing tasks. In China Conference on Information Retrieval, pages 198–210. Springer, 2018.

[30] A Moayedikia, W Yeoh, K Ong, and Y Boo. Improving accuracy and lowering cost in crowdsourcing through an unsupervised expertise estimation approach. Decision Support Systems, 122:1–10, 2019.

[31] Alireza Moayedikia, William Yeoh, Kok-Leong Ong, and Yee-Ling Boo. Framework and literature analysis for crowdsourcing’s answer aggregation. Journal of Computer Information Systems, pages 1–12, 2017.

[32] Alistair Moffat and Justin Zobel. Rank-biased precision for measurement of retrieval effectiveness. ACM Transactions on Information Systems (TOIS), 27(1):1–27, 2008.

[33] Todd K Moon. The expectation-maximization algorithm. IEEE Signal processing magazine, 13(6):47–60, 1996.

[34] Robbie T Nakatsu, Elissa B Grossman, and Charalampos L Iacovou. A taxonomy of crowdsourcing based on task complexity. Journal of Information Science, 40(6):823–834, 2014.

[35] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10):1345–1359, 2010.

[36] Fernando Pech, Alicia Martinez, Hugo Estrada, and Yasmin Hernandez. Semantic annotation of unstructured documents using concepts similarity. Scientific Programming, 2017, 2017.

[37] Vikas C Raykar, Shengpeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. Journal of Machine Learning Research, 11(Apr):1297–1322, 2010.

[38] Rion Snow, Brendan O’Connor, Daniel Jurafsky, and Andrew Y Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In Proceedings of the conference on empirical methods in natural language processing, pages 254–263. Association for Computational Linguistics, 2008.

[39] Takeru Sunahase, Yukino Baba, and Hisashi Kashima. Pairwise hits: quality estimation from pairwise comparisons in creator-evaluator crowdsourcing process. In Thirty-First AAAI Conference on Artificial Intelligence, pages 977–983, 2017.

[40] Shweta Suran, Vishwajeet Pattanaik, and Dirk Draheim. Frameworks for collective intelligence: A systematic literature review. ACM Computing Surveys (CSUR), 53(1):1–36, 2020.

[41] Xing Tang, Ling Chen, Jun Cui, and Baogang Wei. Knowledge representation learning with entity descriptions, hierarchical types, and textual relations. Information Processing & Management, 56(3):809–822, 2019.

[42] Fangna Tao, Liangxiao Jiang, and Chaoqun Li. Label similarity-based weighted soft majority voting and pairing for crowdsourcing. Knowledge & Information Systems, 62(7), 2020.

[43] Christopher Tauchmann, Johannes Daxenberger, and Margot Mieskes. The influence of input data complexity on crowdsourcing quality. In Proceedings of the 25th International Conference on Intelligent User Interfaces Companion, pages 71–72, 2020.

[44] Matteo Venanzi, John Guiver, Gabriella Kazai, Pushmeet Kohli, and Milad Shokouhi. Community-based bayesian aggregation models for crowdsourcing. In Proceedings of the 23rd international conference on World wide web, pages 155–164. ACM, 2014.

[45] Matteo Venanzi, John Guiver, Pushmeet Kohli, and Nicholas R Jennings. Time-sensitive bayesian information aggregation for crowdsourcing systems. Journal of Artificial Intelligence Research, 56:517–545, 2016.

[46] Ellen M Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. Information processing & management, 36(5):697–716, 2000.

[47] Bin Wang, Angela Wang, Fenxiang Chen, Yuncheng Wang, and C-C Jay Kuo. Evaluating word embedding models: methods and experimental results. APSIPA transactions on signal and information processing, 8, 2019.

[48] Jing Zhang, Victor S Sheng, and Jian Wu. Crowdsourced label aggregation using bilayer collaborative clustering. IEEE transactions on neural networks and learning systems, 30(10):3172–3185, 2019.

[49] Jing Zhang, Xindong Wu, and Victor S Sheng. Learning from crowdsourced labeled data: a survey. Artificial Intelligence Review, 46(4):543–576, 2016.

[50] Yuxiang Zhao and Qinghua Zhu. Evaluation on crowdsourcing research: Current status and future direction. Information Systems Frontiers, 16(3):417–434, 2014.

[51] Libin Zheng and Lei Chen. Dlta: A framework for dynamic crowdsourcing classification tasks. IEEE Transactions on Knowledge and Data Engineering, 31(5):867–879, May 2019.

---

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/