IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 2, 2021, accepted November 8, 2021, date of publication November 12, 2021,
date of current version November 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3127994

Aggregating Reliable Submissions in Crowdsourcing Systems

AYSWARYA R. KURUP[1], G. P. SAJEEV[1,2], (Senior Member, IEEE), AND J. SWAMINATHAN[1]
[1]Department of Computer Science and Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Amritapuri Campus, Kollam 690525, India
[2]Department of Electronics and Communication Engineering, Government Engineering College Kozhikode, Kozhikode 673005, India

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

ABSTRACT Crowdsourcing is a cost-effective method that gathers crowd wisdom to solve machine-hard problems. In crowdsourcing systems, requesters post tasks for obtaining reliable solutions. Nevertheless, since workers have various expertise and knowledge background, they probably deliver low-quality and ambiguous submissions. A task aggregation scheme is generally employed in crowdsourcing systems, to deal with this problem. Existing methods mainly focus on structured submissions and also do not consider the cost incurred for completing a task. We exploit features of submissions to improve the task aggregation for proposing a method which is applicable to both structured and unstructured tasks. Moreover, existing probabilistic methods for answer aggregation are sensitive to sparsity. Our approach uses a generative probabilistic model that incorporates similarity in answers along with worker and task features. Thereafter, we present a method for minimizing the cost of tasks, that eventually leverages the quality of answers. We conduct experiments on empirical data that demonstrates the effectiveness of our method compared to state-of-the-art approaches.

INDEX TERMS Answer aggregation, expertness estimation, probabilistic model, quality control, cost minimization.

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

The associate editor coordinating the review of this manuscript and approving it for publication was Giuseppe Desolda.

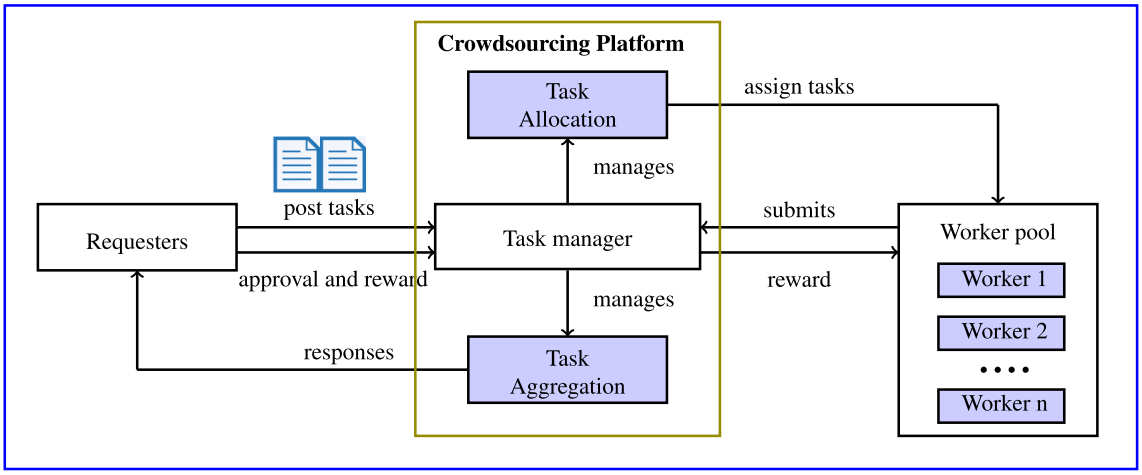


FIGURE 1. Workflow in Crowdsourcing: Illustrates the interaction among requesters and workers, administered by crowdsourcing platform, to accomplish a task.

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 focus on inferring the most appropriate and reliable answers based on worker features such as ability, expertness, 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. II discusses the previous works on task aggregation. Section. III 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 effectual quality control mechanisms. They have developed models that differ from the common platforms 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 mechanism for identifying acceptable submissions. They noticed the lack of proper studies on quality control methods that combine the requester's objectives, characteristics of tasks, and feedback. Gao *et al.* [11] observed that the current aggregation methods take structured data and do not consider unstructured tasks. Methods to extract and aggregate unstructured data are yet to be studied. Hung *et al.* [13] has compared various aggregation approaches and observed that EM-based approaches give the highest accuracy even if the worker group contains spammers.

As yet, several works on task aggregation mechanisms have been proposed that infer the true answers. In most of the works, the correct answers are estimated using one of the following approaches: i) based on the quality of workers, infer the correct answers or ii) based on the correct answer, estimate the quality of the workers. Most of the inference algorithms use the parameters of workers and tasks for estimating the results. Certain unsupervised methods consider the information on submissions and do not contemplate any prior information.

The most classical method MV [38], [42] is considered to be popular and effective. It assigns equal weights to all participating workers in a task. It assumes that all the workers are equally good. Hence, when the number of spammers increases, MV tends to give incorrect answers. Another classical aggregation model is proposed by Dawid *et al.* [5] to evaluate the credibility of diagnosis data from multiple doctors and is optimized by the EM algorithm. Thenceforth, this approach is applied in various classification problems in which the training data is created by using low-cost noisy workers. Moreover, the benefits of such data are studied in the context of supervised learning for classification problems. However, these are methods are more sensitive to data sparsity, and no investigation is conducted that proves the performance.

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 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.

Lyu *et al.* [29] infers the true answers using sophisticated probabilistic methods that use submission and worker features. However, they do not incorporate the worker reliability which is significant in computing the correct answers. However, it overcomes the demerits of using a two-stage review process. Moayedikia *et al.* [30] estimate the expertness of workers using an unsupervised 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 based model that make use of task features for predicting answers is detailed 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 address 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) [33] 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.

TABLE 1. Notations.

Notation	Definition
I	Worker set
J	Task set
K	Submission set
S	All set of correct submissions
e_i	Expertness
R_{ij}	Reliability
Te	Task-easiness
T_c	Task category
a_i	Worker-ability
t_i	Trust-factor of worker i
C	Cost or reward

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.

1) 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 λ_{in} is the n^{th} feature of the worker i and $\Delta \in \{0, 1\}$ be the

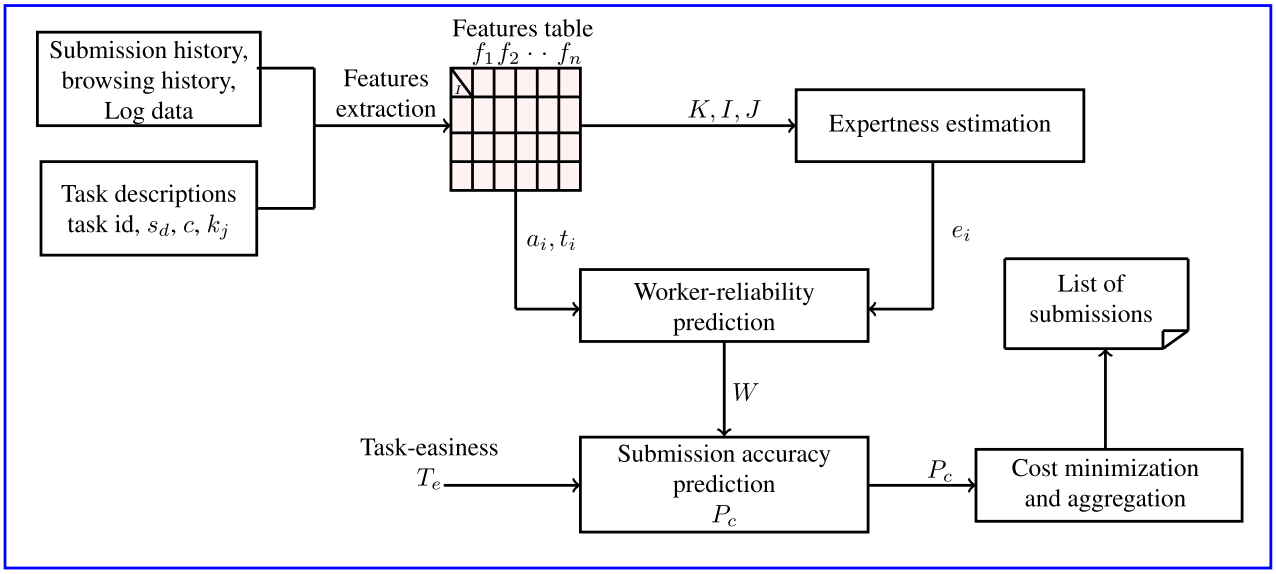


FIGURE 2. The proposed task aggregation method receives history, task descriptions, and similarity in answers as input to predict worker-reliability and submission accuracy.

answer feature matrix in which Δ_{jm} denotes m^{th} feature of the submission J. We use these features during the parameter estimation as auxiliary information.

B. 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*, *expertness*, *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 *expertness* 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 *expertness* 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* $Te_j \in [0, 1]$ which is a measure of the proportion of workers who have enough skills for correctly solving J. *Te* 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 *expertness* using the similarity of answers. Then *worker-ability*, *trust-factor*, *expertness*, 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 *expertness* and ii) *task-easiness* degree. Thus the probability of a submission k_{ij} is given by

$$P_c[k_{ij} = s_{ij} \mid A, t, e, Te] = \begin{cases} WTe, & \text{if } k_{ij} = s_{ij} \\ \dfrac{1 - WTe}{T_c - 1}, & \text{otherwise} \end{cases} \quad (1)$$

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 Te are independent. Hence, the probability of the answer is correct is given by WTe. For the rest of the answers those are incorrect, we choose them with equal probability to occur. Apparently, $\{1 - W\ Te\}$ is the probability that the answer being incorrect.

C. 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 *expertness* e_i. Hence, R is represented as a combination of these parameters as

$$R_{ij} = m_1 A_{ij} + m_2(ct_i) + m_3 e_i \quad (2)$$

Here we consider the features are equally likely, hence $m_1 = m_2 = m_3 = 1$. Therefore,

$$R_{ij} = (A_{ij} + ct_i) + e_i \quad (3)$$

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 expertness 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

$$W = P(W = 1) = \frac{1}{1 + exp(-R_{ij})} \tag{4}$$

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

$$\begin{aligned} P(K \mid \theta, S) &= P(K \mid A, t, Te, S)\, P(S \mid \pi) \\ &= \Pi_{j\in J}\pi_{s_j}\ \Pi_{k_{ij}\in K} P(k_{ij} \mid a_i, t_i, Te_j, s_j) \end{aligned} \tag{5}$$

Fig. 3 depicts the graphical representation of the prediction method. π, 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 π.

D. ESTIMATING THE WORKERS' EXPERTNESS

The computation of the expertness parameter is challenging since we need to handle unstructured tasks also. Recall, existing methods deals 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], [41]. In this, a set of entities are extracted from each answer $k_x \in K$ 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.

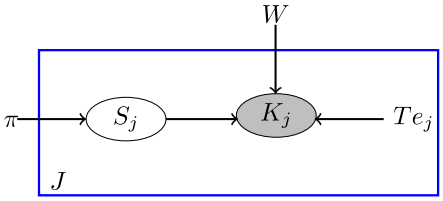


FIGURE 3. Plate notation of the prediction method that represents parameters, hidden variables, and observed variables.

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 [36]. 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], [47], 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* [26] 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

$$sim(|k_x, k_y|) = \frac{\Sigma_i t_1^i * t_2^i}{|t_1| * |t2|} \tag{6}$$

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

$$e_i = \frac{\Sigma_{k_x,k_y\in K, k_x\neq k_y} sim(|k_x, k_y|)}{K - 1} \tag{7}$$

Comparing with all possible answers, the expertness of a worker is more if the answer is supported by other similar answers. This kind of expertness 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 *expertness* 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 [35] and feature-based matrix factorization are used to exploit auxiliary information.

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

$$\hat{\theta} = \underset{S,\theta}{\operatorname{argmax}} \; log\, P(K, S, W \mid \theta) \tag{8}$$

We estimate the parameters for reliability and answer correctness prediction. Our goal is to compute the correct submissions s_{ij} as well as θ 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 θ is updated in the maximization step. Here $\theta^{(0)} = A^{(0)}, t^{(0)}, e^{(0)}, Te^{(0)}$ denotes the initialized parameters. $\theta^{(i)} = A^{(i)}, t^{(i)}, e^{(i)}, Te^{(i)}$ 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 θ from the maximization step. θ is a conditional independent with s_{ij}. The current estimate of the parameters are $\theta^{(i)}$, for the $i+1$ iteration of E-step is given by

$$\begin{aligned} Q(\theta, \theta^{(i)}) &= E[logP(K|W, Te, S)] \\ &= E[log\Pi_j(p(s_j)p(K_{*j}|s_j, W, Te, S))] \\ &= \Sigma_j\Sigma_k p(k_{ij})\; log\; p(k_{ij} = s_{ij}) \\ &\quad + \Sigma_{ij}\; \Sigma_k p(k_{ij}) log\; p(k_{ij}|s_j = k_j, W, Te, S) \end{aligned} \tag{9}$$

M-step: The objective of maximization 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 θ that maximizes $Q(\theta, \theta^{(i)})$ in the $i + 1^{th}$ iteration as

$$\theta^{(i+1)} = \underset{\theta}{\operatorname{argmax}}\; Q(\theta, \theta^{(i)}) \tag{10}$$

where we infer the parameters $\theta = [A, t, Te, S]$ such that $\Sigma_K \theta_{j,k} = 1$ and $\Sigma_{sj}\pi_{i,k} = 1$ so that (5) is maximized. Finally, we use $P_c(k_{ij})$ to estimate the correct answers $\{s_j\}$, such that $s_j = \underset{\theta}{\operatorname{argmax}}\; P_c(k_{ij})$.

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

$$\left\| Q(\theta^{i+1}, \theta^{(i)}) - Q(\theta^{i}, \theta^{(i)}) \right\| \leq tr \tag{11}$$

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 λ and Δ 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 *Ii*, set of tasks *Jj*, set of answers K_{ij} and the estimated outputs are *task-easiness Te*, *reliability* $R\{r_i\}$ of workers i, priors of $\pi\{\pi_S\}$, and posterior probabilities 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 (θ, π, A, t, Te) is performed in the next lines 3 to 5. The similarity of answers is calculated. The expertness 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.

A. 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], [28]. Workers receive incentives for their work, but usually, the requesters have limited budgets. Besides, structured tasks such as data annotation may require multiple 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

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**
  $\theta, P_{j,k} \leftarrow$ estimate using EM from K and S
  Compute the expertness e_i as
  $e_i = sim(|k_x, k_y|) = \frac{\Sigma_i t_1^i * t_2^i}{|t_1| * |t2|}$ ▷ t_1^i and t_2^i 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; ▷ Probability of giving correct answers
  Update K as $K \cup$ 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 Te_j, R_i, e_i 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 Te, $W \leftarrow 1 \leq j \leq n$, $1 \leq i \leq C_0/3$ using EM method;
Set initial threshold Tr^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 Te;
  Collect answers K;
  Estimate Te, W using TAM;
  $C = C - \Sigma_{i \in 1,\ldots |J_{avl}|} s_i^r$;
  Compute Tr^r;
  for *each* $j = 1, 2, .., n$ **do**
    if $w^r \geq Tr^r$ **then**
      $J_{avl} = J/j$;
    end
  end
end

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 $W < Tr$, 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. 4 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. III. In the initial round $r = 0$, starts with the EM phase, i workers and an initial cost or incentive is allocated. The parameters Te and W are estimated using the EM approach and the corresponding submissions K are collected from the workers. The initial threshold Tr is also computed. In the next round, allocate workers for each task based on the estimated Te. The procedure for minimizing the cost of tasks is explained in Algorithm 2.

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.

The proposed task aggregation method is abbreviated as TAM.

A. DATASET

We select three real dataset collected from Figure Eight [1]. 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 statistics of the dataset is given in Table. 2. Among the responses, only 2.67% (data annotation) to 3.96% (sentiment analysis) responses are selected for reward, which suggests the need for automating the answer inference process.

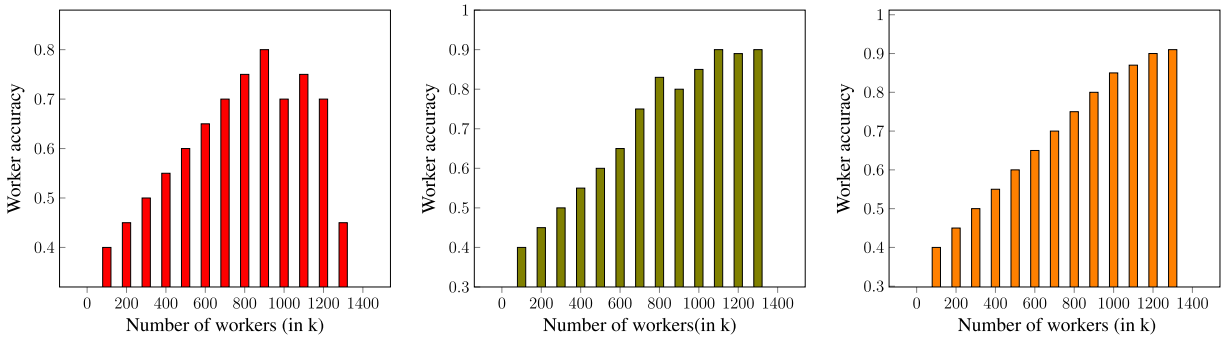
B. PERFORMANCE COMPARISON

We compare the proposed task aggregation method with the following baseline methods and state-of-the-art approaches.

- Majority voting (MV): The most conventional method that selects the submission which has the highest votes.
- Logistic regression(LoR): This method takes the feature vector $X_{i,j}$ as input and output the feedback $c_{i,j}$. For a task with multiple answers, it predicts the feedback score of each answer and ranks the answers in descending quality [13].

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



(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 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.

- DSM: A truth inference method based on EM estimation that considers only the worker features [5], [12].
- BGM: An active learning approach that use Bayesian graphical model to explore worker correlation [45].
- WSM: A probability based approach for general crowdsourcing tasks [29].

These are the most popular methods used in mainstream studies. Among the compared methods MV and DSM use only the submissions as input. On the other hand, LoR, BGM and WSM utilize the object features as well.

C. EVALUATION METRICS

We evaluate the effectiveness and importance of task aggregation using the following metrics. *Mean Average Precision* (mAP) is used as a primary evaluation metric. It is considered as an appropriate criterion for ranking problems [32]. The average precision (AP) is the mean of precision obtained after each relevant document is retrieved. The mean average precision for a submission is the mean of all these AP scores for each topic in the submission, which quantifies how good our method is for retrieving the query. [46]. Another metric we use is *accuracy* which measures the accuracy in aggregation of responses. It is measured as the ratio of N_c and M, where N_c indicates the number of correct responses and M is the total number of responses.

D. METHODOLOGY

The dataset are sorted according to the arrival time of tasks. 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_d, s_l, n_t, and R_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.

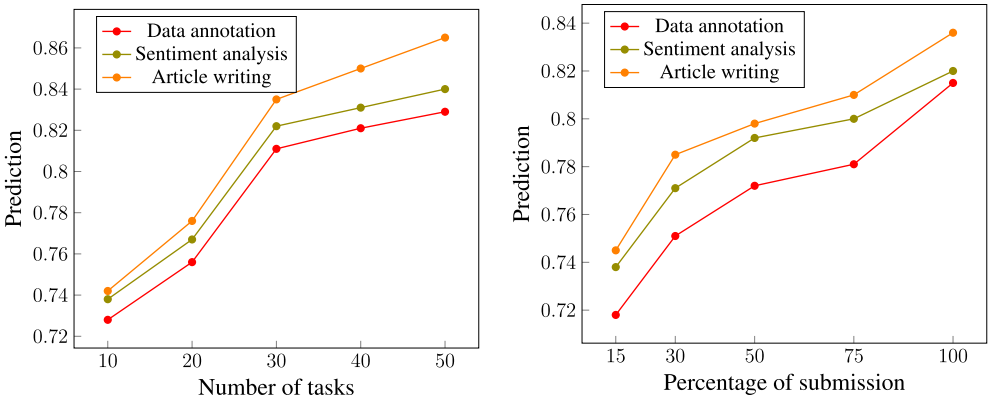
Fig. 4 illustrates the worker accuracy distribution. The dataset is sorted in descending order concerning the mean accuracy of the workers in each type of task. The performance of all the methods is evaluated. We observe that accuracy in data annotation is lower than that of other tasks.

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



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

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

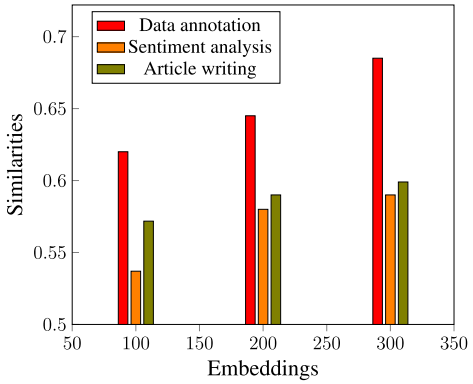


FIGURE 6. Average cosine similarities on various dataset.

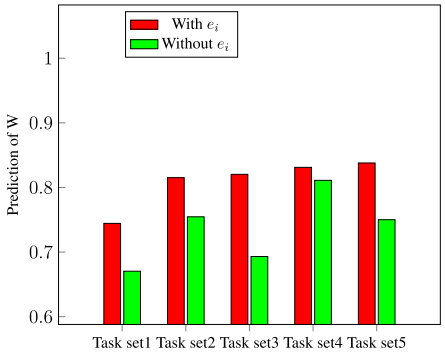


FIGURE 7. Effect of expertness estimation.

corpus and *Word2vec* [2] 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.

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. 4 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 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

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

TABLE 4. Comparison of various methods.

Methods	mAP	Accuracy
MV	0.6604	0.7932
LoR	0.6928	0.8681
TAM	**0.7502**	**0.9132**
DSM	0.7102	0.8205
BGM	0.6352	0.8415
WSM	0.7118	0.9083

TABLE 5. mAP results of feature set on sentiment analysis.

Features	mAP	Rank
SF + WF	0.5280	1
SF	0.4891	2
WF	0.3930	3

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.

TABLE 6. Performance comparison of various methods for sentiment analysis task.

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

TABLE 8. Performance comparison of various methods for article writing.

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

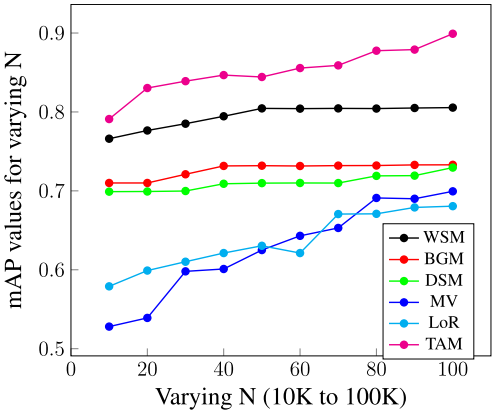
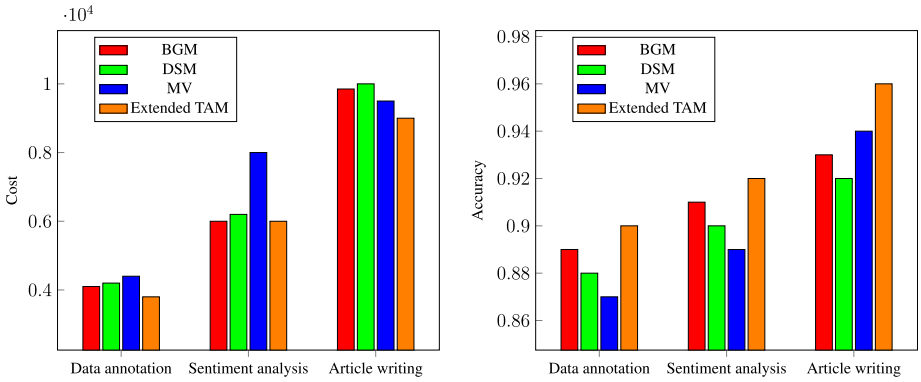


FIGURE 8. mAP values of the aggregation methods for different N values.

3) ARTICLE WRITING

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



(a) Results in terms of cost for various dataset **(b)** Results in terms of accuracy for various dataset

FIGURE 9. Evaluation of cost minimization in terms of accuracy and cost.

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, 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.

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 [7]. 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 Te are estimated in each round r. Then, based on the values of W and Te, C is allocated. The results are given in Fig. 9. 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 all 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.

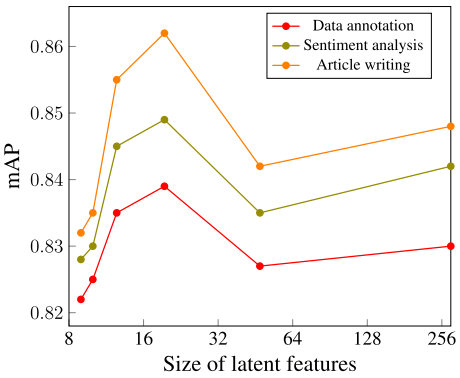


FIGURE 10. Performance of TAM by varying number of latent features.

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 approaches since its complexity is linear to the number of submissions. However, it has the worst accuracy 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

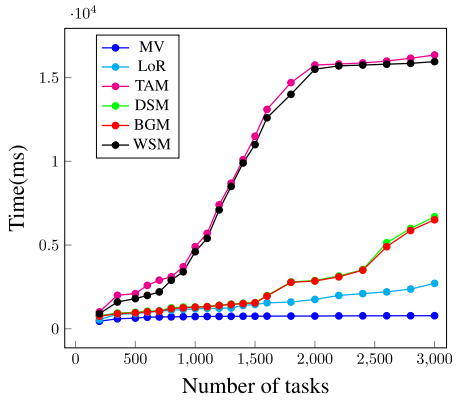


FIGURE 11. Performance comparison of various methods against execution time.

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 increases mAP values also increase, indicating the merit of our approach.

We observe that the methods that uses both the worker and task specific features improves the aggregation process more competently, especially when the submissions are in unstructured form. Specifically, the parameters *worker-reliability*, *expertness*, *trust-factor*, and *task-easiness* are 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*. Accessed: Feb. 5, 2021. [Online]. Available: https://www.figure-eight.com/
[2] V. K. Ayyadevara, "Word2vec," in *Pro Mach. Learn. Algorithms*. Berkeley, CA, USA: Apress, 2018, pp. 167–178.
[3] Y. Baba and H. Kashima, "Statistical quality estimation for general crowdsourcing tasks," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2013, pp. 554–562.
[4] F. Daniel, P. Kucherbaev, C. Cappiello, B. Benatallah, and M. Allahbakhsh, "Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions," *ACM Comput. Surv.*, vol. 51, no. 1, pp. 1–40, Apr. 2018.
[5] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the EM algorithm," *J. Roy. Statist. Soc. C Appl. Statist.*, vol. 28, no. 1, pp. 20–28, 1979.
[6] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc. B, Methodol.*, vol. 39, no. 1, pp. 1–22, 1977.
[7] D. E. Difallah, G. Demartini, and P. Cudré-Mauroux, "Scheduling human intelligence tasks in multi-tenant crowd-powered systems," in *Proc. 25th Int. Conf. World Wide Web*, Apr. 2016, pp. 855–865.
[8] J. Fan, G. Li, B. C. Ooi, K.-L. Tan, and J. Feng, "ICrowd: An adaptive crowdsourcing framework," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, May 2015, pp. 1015–1030.
[9] R. Feldman, *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge, U.K.: Cambridge Univ. Press, 2007.
[10] J. Feller, P. Finnegan, J. Hayes, and P. O'Reilly, "'Orchestrating' sustainable crowdsourcing: A characterisation of solver brokerages," *J. Strategic Inf. Syst.*, vol. 21, no. 3, pp. 216–232, Sep. 2012.
[11] J. Gao, Q. Li, B. Zhao, W. Fan, and J. Han, "Truth discovery and crowdsourcing aggregation: A unified perspective," *Proc. VLDB Endowment*, vol. 8, no. 12, pp. 2048–2049, 2015.
[12] A. Ghezzi, D. Gabelloni, A. Martini, and A. Natalicchio, "Crowdsourcing: A review and suggestions for future research," *Int. J. Manage. Rev.*, vol. 20, no. 2, pp. 1468–2370, Apr. 2017.
[13] N. Q. V. Hung, N. T. Tam, L. N. Tran, and K. Aberer, "An evaluation of aggregation techniques in crowdsourcing," in *Proc. Int. Conf. Web Inf. Syst. Eng.* Berlin, Germany: Springer, 2013, pp. 1–15.
[14] J. Jiang, B. An, Y. Jiang, D. Lin, Z. Bu, J. Cao, and Z. Hao, "Understanding crowdsourcing systems from a multiagent perspective and approach," *ACM Trans. Auto. Adapt. Syst.*, vol. 13, no. 2, pp. 1–32, Nov. 2018.
[15] L. Jiang, H. Zhang, F. Tao, and C. Li, "Learning from crowds with multiple noisy label distribution propagation," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, May 31, 2021, doi: 10.1109/TNNLS.2021.3082496.
[16] A. R. Kurup. (2020). *Task-Aggregation*. [Online]. Available: https://github.com/ayswaryarkurup/task-aggregation.git
[17] A. R. Kurup and G. P. Sajeev, "Aggregating unstructured submissions for reliable answers in crowdsourcing systems," in *Proc. 9th Int. Symp. Embedded Comput. Syst. Design (ISED)*, Dec. 2019, pp. 1–7.
[18] A. R. Kurup and G. P. Sajeev, "Analysing the characteristics of crowdsourcing platforms for improving throughput," *Int. J. Web Eng. Technol.*, vol. 14, no. 3, pp. 255–279, 2019.
[19] A. R. Kurup and G. P. Sajeev, "A task recommendation scheme for crowdsourcing based on expertise estimation," *Electron. Commerce Res. Appl.*, vol. 41, May 2020, Art. no. 100946.
[20] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 2177–2185.

[21] G. Li, J. Wang, Y. Zheng, and M. J. Franklin, "Crowdsourced data management: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 9, pp. 2296–2319, Sep. 2016.
[22] H. Li, B. Zhao, and A. Fuxman, "The wisdom of minority: Discovering and targeting the right group of workers for crowdsourcing," in *Proc. 23rd Int. Conf. World Wide Web*, 2014, pp. 165–176.
[23] J. Li, Y. Baba, and H. Kashima, "Incorporating worker similarity for label aggregation in crowdsourcing," in *Proc. Int. Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, 2018, pp. 596–606.
[24] S.-Y. Li, S.-J. Huang, and S. Chen, "Crowdsourcing aggregation with deep Bayesian learning," *Sci. China Inf. Sci.*, vol. 64, no. 3, pp. 1–11, Mar. 2021.
[25] Y. Li, C. Liu, N. Du, W. Fan, Q. Li, J. Gao, C. Zhang, and H. Wu, "Extracting medical knowledge from crowdsourced question answering website," *IEEE Trans. Big Data*, vol. 6, no. 2, pp. 309–321, Jun. 2020.
[26] Y.-S. Lin, J.-Y. Jiang, and S.-J. Lee, "A similarity measure for text classification and clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 7, pp. 1575–1590, Jul. 2014.
[27] J. Liu, F. Tang, L. Chen, and Y. Zhu, "Exploiting predicted answer in label aggregation to make better use of the crowd wisdom," *Inf. Sci.*, vol. 574, pp. 66–83, Oct. 2021.
[28] Q. Liu, J. Peng, and A. T. Ihler, "Variational inference for crowdsourcing," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 692–700.
[29] S. Lyu, W. Ouyang, H. Shen, and X. Cheng, "Finding high-quality unstructured submissions in general crowdsourcing tasks," in *Proc. China Conf. Inf. Retr.* Cham, Switzerland: Springer, 2018, pp. 198–210.
[30] A. Moayedikia, W. Yeoh, K. Ong, and Y. Boo, "Improving accuracy and lowering cost in crowdsourcing through an unsupervised expertise estimation approach," *Decis. Support Syst.*, vol. 122, pp. 1–10, Jan. 2019.
[31] A. Moayedikia, W. Yeoh, K.-L. Ong, and Y.-L. Boo, "Framework and literature analysis for crowdsourcing's answer aggregation," *J. Comput. Inf. Syst.*, vol. 60, no. 1, pp. 49–60, Jan. 2017.
[32] A. Moffat and J. Zobel, "Rank-biased precision for measurement of retrieval effectiveness," *ACM Trans. Inf. Syst.*, vol. 27, no. 1, pp. 1–27, Dec. 2008.
[33] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 47–60, Nov. 1996.
[34] R. T. Nakatsu, E. B. Grossman, and C. L. Iacovou, "A taxonomy of crowdsourcing based on task complexity," *J. Inf. Sci.*, vol. 40, no. 6, pp. 823–834, Dec. 2014.
[35] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[36] F. Pech, A. Martinez, H. Estrada, and Y. Hernandez, "Semantic annotation of unstructured documents using concepts similarity," *Sci. Program.*, vol. 2017, Dec. 2017, Art. no. 7831897.
[37] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds," *J. Mach. Learn. Res.*, vol. 11, pp. 1297–1322, Apr. 2010.
[38] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng, "Cheap and fast–but is it good? Evaluating non-expert annotations for natural language tasks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2008, pp. 254–263.
[39] T. Sunahase, Y. Baba, and H. Kashima, "Pairwise hits: Quality estimation from pairwise comparisons in creator-evaluator crowdsourcing process," in *Proc. 21st AAAI Conf. Artif. Intell.*, 2017, pp. 977–983.
[40] S. Suran, V. Pattanaik, and D. Draheim, "Frameworks for collective intelligence: A systematic literature review," *ACM Comput. Surv.*, vol. 53, no. 1, pp. 1–36, May 2020.
[41] X. Tang, L. Chen, J. Cui, and B. Wei, "Knowledge representation learning with entity descriptions, hierarchical types, and textual relations," *Inf. Process. Manage.*, vol. 56, no. 3, pp. 809–822, May 2019.
[42] F. Tao, L. Jiang, and C. Li, "Label similarity-based weighted soft majority voting and pairing for crowdsourcing," *Knowl. Inf. Syst.*, vol. 62, no. 7, pp. 2521–2538, Jul. 2020.
[43] C. Tauchmann, J. Daxenberger, and M. Mieskes, "The influence of input data complexity on crowdsourcing quality," in *Proc. 25th Int. Conf. Intell. User Interface Companion*, Mar. 2020, pp. 71–72.
[44] M. Venanzi, J. Guiver, G. Kazai, P. Kohli, and M. Shokouhi, "Community-based Bayesian aggregation models for crowdsourcing," in *Proc. 23rd Int. Conf. World Wide Web*, 2014, pp. 155–164.
[45] M. Venanzi, J. Guiver, P. Kohli, and N. R. Jennings, "Time-sensitive Bayesian information aggregation for crowdsourcing systems," *J. Artif. Intell. Res.*, vol. 56, pp. 517–545, Jul. 2016.
[46] E. M. Voorhees, "Variations in relevance judgments and the measurement of retrieval effectiveness," *Inf. Process. Manage.*, vol. 36, no. 5, pp. 697–716, Sep. 2000.
[47] B. Wang, A. Wang, F. Chen, Y. Wang, and C.-C.-J. Kuo, "Evaluating word embedding models: Methods and experimental results," *APSIPA Trans. Signal Inf. Process.*, vol. 8, pp. 1–14, Dec. 2019.
[48] J. Zhang, V. S. Sheng, and J. Wu, "Crowdsourced label aggregation using bilayer collaborative clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 10, pp. 3172–3185, Oct. 2019.
[49] J. Zhang, X. Wu, and V. S. Sheng, "Learning from crowdsourced labeled data: A survey," *Artif. Intell. Rev.*, vol. 46, no. 4, pp. 543–576, Dec. 2016.
[50] Y. Zhao and Q. Zhu, "Evaluation on crowdsourcing research: Current status and future direction," *Inf. Syst. Frontiers*, vol. 16, no. 3, pp. 417–434, 2014.
[51] L. Zheng and L. Chen, "DLTA: A framework for dynamic crowdsourcing classification tasks," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 5, pp. 867–879, May 2019.

AYSWARYA R. KURUP received the bachelor's degree in information technology from Amrita University, India, in 2010, and the master's degree in computer science and engineering from Mahatma Gandhi University, India, in 2014. She is currently a Researcher with the Department of Computer Science and Engineering, Amrita University. Her research interests include crowdsourcing, information retrieval, and recommender systems.

G. P. SAJEEV (Senior Member, IEEE) received the Ph.D. degree from the National Institute of Technology Calicut, India. He is currently serving as an Associate Professor at the Department of Computer Science and Engineering, School of Engineering, Amritapuri Campus. He has published many research papers in international journals and conferences. His research interests include open systems, complex networks, web science, network graphs, and crowdsourcing. He is a reviewer of *International Journal of Computers and Electrical Engineering* (Elsevier) and IEEE Access, and a TPC member of IEEE organized conferences.

J. SWAMINATHAN is currently an Associate Professor at the Department of Computer Science and Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Amritapuri Campus, Kollam, Kerala. He has over 22 years of experience in industry, research, and academia. He is also passionate about teaching. He leads Code@Amrita Club. He has published in several reputed international journals and conferences. His research interests include program analytics, visualization, and verification.