User Profile based system to optimize Crowdsourcing Operations

L. D. C. Wijetunga

2020

User Profile based system to optimize Crowdsourcing Operations

A dissertation submitted for the Degree of Master of Computer Science

L. D. C. Wijetunga
University of Colombo School of Computing
2020

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.
To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: L. D. Chintaka Wijetunga
Registration Number: 2017MCS095
Index Number: 17440951

LDWijetunga

Signature: Date: 14/11/2020

This is to certify that this thesis is based on the work of
Mr. L. D. Chintaka Wijetunga
under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:
Supervisor Name: Dr. (Mrs). M.D.J.S.Goonetillake

J. Goonetillake

Signature: Date: 17/11/2020

ABSTRACT

Crowdsourcing is a leading concept in problem solving. It allows thousands of individuals to gather online to solve machine-hard problems and achieve solutions. Quality, Cost and Latency are considered as control factors of crowdsourcing, that is, the performance of crowdsourcing depend on these factors. Traditionally, optimization refers to reduction in time, that is latency of getting a job done.

The expected optimization of crowdsourcing is based on three control factors, quality, cost and latency. So, it is very important that these three factors are optimized properly to achieve an overall optimization in the platform. The problem is, it is not easy to achieve the full optimization in terms of quality, cost and latency. Therefore, it is necessary in terms of crowdsourcing for a solution that would enhance all the three control factors.

To overcome this problem, a software solution was developed to cater these requirements. The software would optimize the process of crowdsourcing from entering a task to the point of viewing solutions for the task. In between, it will make sure sub processes work in a manner that enables a smooth flow of the system. The process would target the workers who work on the task based on the task domain, process the task to find pre crowdsourced solutions, a notification system to interact the workers quickly and a method to handle recurring type of tasks.

The features of the solution are dependent on the information provided by the users during the registration process to the crowdsourcing platform. This can be defined as a limitation of this solution.

Finally, this process can be inherited by any crowdsourcing platform to enhance its crowdsourcing operations.

ACKNOWLEDGMENT

As with most MSc journeys, mine had its fair share of hurdles, obstacles, and confusion. I'm both relieved and humbled to reach the end of this long journey, filled with nervous excitement on what the future might bring. This journey would not have been possible if not for the sacrifices, support, and guidance of many people who actively contributed to the successful completion of my project. I owe my heartfelt gratitude to those who were by my side, patiently and resiliently, throughout this journey. Let me attempt to acknowledge their efforts.

I would like to express my deep gratitude to my project supervisor, Dr. (Mrs.) M. D. J. S. Goonetilake for her patience, guidance, enthusiastic encouragement, and useful critiques of this research work.

My thanks to the MCS Project Coordinator, Dr. Randil Pushpananda and the support staff for their coordination and timely information sharing throughout the project.

My thanks are also extended to my manager of my workplace, Mrs. Indika Gunadasa for encouraging me to complete my master's degree program successfully, and for granting me leaves when I required, and all colleagues who supported me to achieve this. And, my special thanks to everyone who assisted me who participated in my interviews and contributing their ideas for requirements gathering, testing, and critical evaluation.

I would also like to thank my father and mother, for their advice and assistance in keeping my progress schedule, and for reviewing my thesis and supportive documents.

Finally, I wish to express my very profound gratitude towards my beloved wife, Disni for her co-operation and encouragement throughout the program, and my parents for their support and continuous guidance. This accomplishment would not have been possible without them.

Thank you!

CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE

INTRODUCTION

Despite crowdsourcing provides an operational way to process machine-hard tasks by utilizing hundreds of thousands of ordinary workers (i.e., the crowd), it's effectiveness in terms of cost, quality and latency is still problematic in both industry and academia. In the context of crowdsourcing, quality refers here to the quality of the solutions given for a task, cost refers to the cost incurred to get a valid and acceptable solution for a task and the latency refers to the time taken to get a valid and acceptable solution for a task, when a task is crowdsourced. Traditional data management systems usually focus on optimizing one optimization goal - reducing the latency. However, due to the different features of workers (not free, error prone and diverse, dynamic) the solutions given by them have variations. To reduce these effects from workers, it is important to optimize the crowdsourcing process by considering the trade-off between the three optimization goals in crowdsourcing as shown in Fig. 1.1.

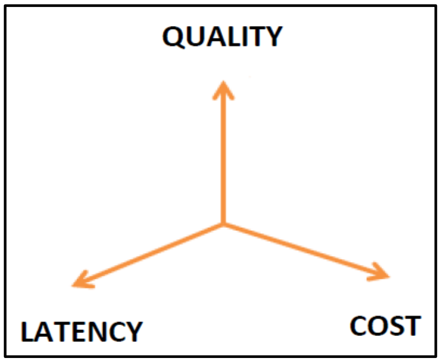


Figure 1.1: Three Dimensions of Trade-off in Crowdsourcing System

Accordingly, if system designers attempt to improve the quality of human submissions, they may have to compensate workers more generously or wait for a longer time. On the other hand, if designers want to shorten the process latency, reducing the time span between the task release and completion, they would either increase incentives to motivate crowd workers or lower their confidence on the accuracy of the submissions they collected. Similarly, if the objective is to reduce the cost, then the quality and latency of the submissions they collect may have to be compromised. Most existing algorithms (procedures) emphasize balancing quality, cost, and latency, and this study examines existing techniques to address these challenges.

1.1 Research Gap and Contribution

At present, many crowdsourcing platforms facilitate users to interact with each other by allowing users (requesters) to submit their tasks to be solved and other users (workers) to accept those tasks and give solutions. The current design and the available research indicate that different solutions such as truth inference, explicit task assignment, task pruning and answer deduction for the optimization of crowdsourcing operations with respect to Quality, Cost, and Latency[22].

The problem is, solutions given facilitate the optimization of one or two control factors which are mentioned previously by compromising the other. That is, if the quality is increased and the cost is decreased, the latency factor would increase. There are vague solutions provided by scholars to enhance all the three factors but according to the literature such solutions are not adequate. The gap in research identified here is the optimization of all three factors, Quality, Cost, and Latency of crowdsourcing operations without compromising each other.

1.2 Background of the Study

As mentioned in the previous selection, importance it to optimization of Quality, Cost and Latency of Crowdsourcing operations without or minimum negative effect between each other. This study will focus on this factor and try to come up with a solution to tackle this requirement.

For this study, major focus will be on Data crowdsourcing and its optimization factors. One of the major crowdsourcing platforms, AmazonTurk, behaviour will be analysed during this study, and its practical application and improvement that can be made will be addressed during the study.

And, as the finally a comprehensive solution for the optimization techniques will be derived and will be implemented.

1.3 Research Problem

As mentioned in the above section, the three key factors which affect the overall crowdsourcing operation are Quality, Cost, and Latency. Traditionally, the systems are designed in a manner to estimate the computation cost of each process and choose the one with the minimum estimated cost. However, this process turns to be quite challenging in a crowdsourcing environment because;

- there are three optimization objectives (result quality, monetary cost, and latency) that need to be considered together and
- humans are much more unpredictable than machines.

This study would discuss a solution that would enhance all three factors in a way to build a strong platform to cater to requesters and workers’ demands.

1.3 Research Questions

Based on the above discussion the following three questions raised in the study.

1. Does maintaining a worker rating system and using rating figures to prioritize task solutions would make high-quality answers to be accessed quickly?
2. Does targeting workers in a Crowdsourcing platform by their interests, expertise, and geographical location, and send similar tasks to them would result in a small solution pool, thus the cost in getting a solution reduced?
3. Do tasks that require solutions on a daily/weekly basis, could be set into a schedule and automatically published in the platform at the correct time to speed up the task resolution process?

1.4 Contribution

1.4.1 Goal of the project

The goal of the project is to propose and develop a new framework for a crowdsourcing platform to enhance its efficiency with respect to quality, latency and cost.

1.4.2 Objectives of the project

To achieve the above-mentioned goal, some objectives must be covered. These objectives can be further listed as below.

- Figure out the most important areas in crowdsourcing where the optimization should be carried upon in terms of quality, cost and latency.
- Develop an efficient crowdsourcing platform with user profile-based approach.
- Incorporate an effective notification deliver system to optimize the operations within the platform.
- Develop the capability of handling similar crowdsourcing tasks.

1.5 Methodology

This project has spread over different domains in the computer science field such as data gathering (as the form of tasks and solutions), analysis and storing, natural language processing, and mainly crowdsourcing. To achieve the said aim and objectives, following methods were taken.

The analysis on the research gap was done, in focus of quality, cost and latency in focus of developing a new solution.

The outcome of the analysis was used to propose and implement a new solution to achieve the aim and objectives of this project.

Once the solution was finalized the evaluation plan was formulated using a mixed method by combining both questionnaires which will gather feedback from a group of individuals regarding the proposed solution and experimental based feedback where a group of individuals are given a hands-on experience on the proposed solution.

As the final step, the evaluation was done in accordance to the evaluation plan to verify the acceptance of the proposed solution.

1.8 Summary of Chapters

This chapter focussed a brief introduction to the project, discussed the problem, the research gap, the research question, aim and objects, a brief methodology, the proposed solution and the scope and limitations of the project.

In the next chapter, a comprehensive literature review on the crowdsourcing domain in terms of the control facts, quality cost and latency will be discussed and presented. This section will discuss how past studies and current applications use crowdsourcing optimization and a comparison between these.

Afterwards, a detailed methodology chapter will discuss the process that followed to solve the issue. Further in this section, all the relevant areas will are affected along with a detailed description on the solution will be addressed and presented.

The evaluation chapter will discuss how this proposed and implemented system will be evaluated considering real world scenarios while the same section will discuss the outcome of the evaluation.

Finally, the conclusion section will discuss the overall achievements of the study, limitations and some of the expected future works.

CHAPTER TWO

LITERATURE REVIEW

2.1 Chapter Overview

This chapter will explain how the researchers have involved in their research and the technical area associated with the crowdsourcing domain. It will focus on the key terms, operations, techniques, and technologies used in the project. At the same time, the literature associated with this project's solution will be discussed widely. As mentioned in the previous chapter, the focus of this project is on the control factors of crowdsourcing. So, throughout this chapter, an in-depth literature review is discussed on the quality, cost and latency.

2.2 Crowdsourcing domain: Quality, Cost and Latency

Computers and the internet have changed the way of working in the word by storing and processing a huge amount of information. Still, many machine-hard problems and tasks cannot be completely solved by automated processes, and which require a human touch. For example, analysis of a sentence or a picture and identify the sentiment behind it or image recognition.

Although powerful machine learning algorithms are available for these tasks, human interaction would give better performance and quality. For instance, a human can identify the sentiment of a message, recognize handwriting, and even recognize an image, yet being much more accurate than a machine algorithm.

Crowdsourcing provides an effective way to process such machine-hard tasks by utilizing thousands of ordinary workers, which is the crowd [2].

The optimization of crowdsourcing depends on three control factors; quality, latency and cost.

2.2.1 Terms used in the project

2.2.1.1 Crowdsourcing

Crowdsourcing is a data sourcing model in which individuals or organizations obtain goods and services, including ideas, voting, micro-tasks and finances, from a large, relatively open and often rapidly-evolving group of participants [1].

There are different branches of crowdsourcing. It consists of wikies, open source software development. micro-tasking, idea generation, journalism etc. Though the application of each branch is different, the same concept of crowdsourcing is used in all the domain.

That is, if an individual or an organization require the help of the crowd to execute an operation, the crowd will be notified, and they can work on the operation to find solutions. In a simpler manner, crowdsourcing is simply an operation in which a worker shares his or her expertise to solve an issue. The figure 2.1 illustrates the branches [3].

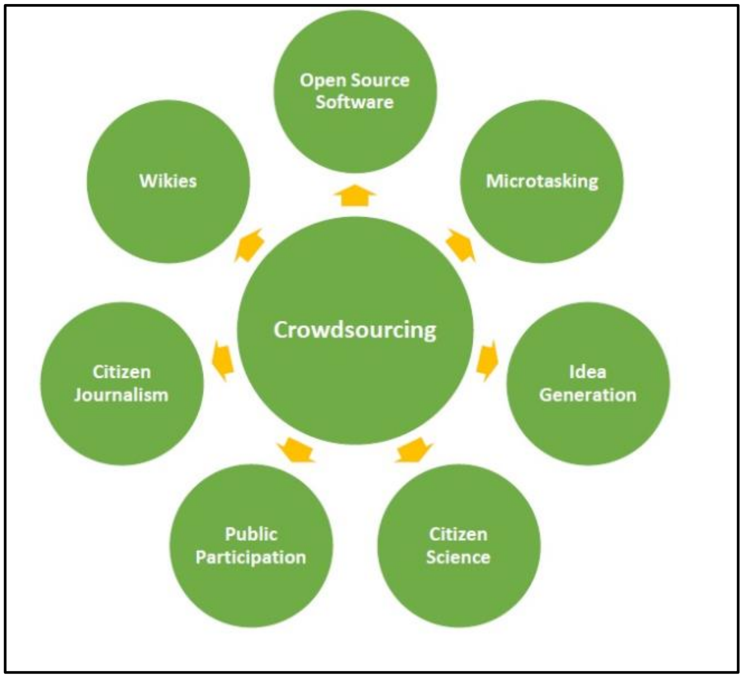


Figure 2.1: Branches of crowdsourcing.

Another branch is crowdfunding, where an individual can ask out help from the crowd to help with their financial state, which not will be discussed in this project [4].

2.2.1.2 Requester

A requester is a person or an organization who publishes tasks in a crowdsourcing platform, which is one of the key parties associated in the crowdsourcing process. A requester must design the tasks, set up the tasks, publish the tasks, and keep on monitoring the tasks until a valid solution/s are available [2].

Further, a requester must reward the workers who complete the task. It is up to the requester to reward workers with bonuses for high-quality work or even pay less for low quality work. Also, a requester can block a worker from attempting any future tasks published by them [5].

2.2.1.3 Worker

The worker needs to browse through a set of tasks in the crowdsourcing platform, assign tasks and submit answers for those, which is the most important party of the crowdsourcing process. The ultimate goal of workers is to earn money from solving tasks [6].

Compared to computers, workers have different characteristics. They are not free. Workers need to be paid for answering a task. They are error-prone and diverse. Workers' solutions are not always correct. It may contain erroneous or noisy solutions. As workers are from different backgrounds, expertise, geographically, etc, their answering style and quality differ. Finally, they are dynamic. That is, workers are not always online to answer the tasks. This will affect the latency of task solutions [2].

2.2.1.4 Crowdsourcing Platform

A marketplace or a platform that allows the requesters to deploy tasks and workers to answer them. There is a payment required for this service, and requesters need to pay the platform for using its service [2].

Businesses use these platforms differently. For complex projects that require workers with special specialties, multiple platforms will be used. For example, if the task is software development related, it may use GitHub, while if the tasks are business or market related it may use consumer-facing platforms [7].

Important features of these kinds of platforms are, they are scalable, fills knowledge gaps, accelerates processes, and even reduce unnecessary costs [7].

2.2.1.5 Microsoft Azure Database

MS Azure is a cloud computing service created by Microsoft. This can be used to build, test, deploy, and manage applications and services through Microsoft Managed data centres [8].

It is ideal for of project which are web based since the database should be centralized and should be accessed by different clients at the same time.

2.2.1.6 Natural Language Processing

Natural Language Processing, NLP, mainly focuses on the interaction between computers and human natural languages. NLP has connections towards linguistic, computer science, information engineering, and artificial intelligence.

In simpler terms, NLP gives the ability to program computers to process and analyse a large amount of natural language data [9].

2.2.2 Similar researches done in the area

Data management is a key aspect of crowdsourcing. Traditional data management concepts focus on the optimization of one optimization goal, which is latency. But, the behaviour of the workers (not free, error-prone, and diverse, dynamic in the crowdsourcing platforms, there are three main optimization goals in crowdsourcing. They are, improving the quality, reducing the cost, and controlling the latency [10]. So, the base of optimization in crowdsourcing depend on these three control factors, quality, cost and latency.

Figure 2.2 illustrate how these optimization goals are connected with all the components of a crowdsourcing operation [2].

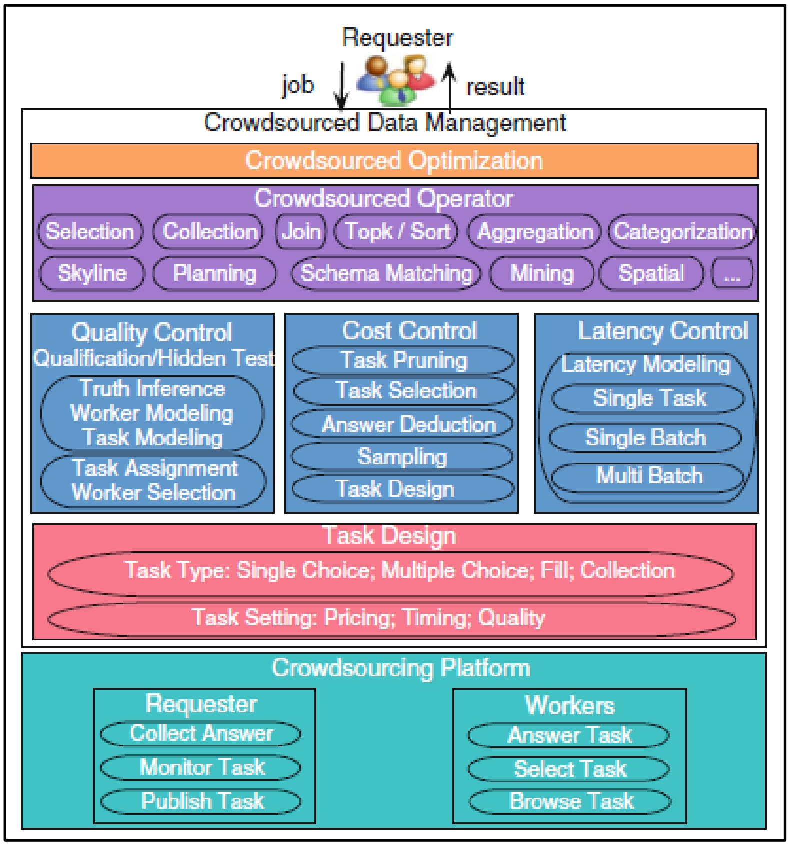


Figure 2.2: Overview of crowdsourced data management

According to Figure 2.2, quality, cost and latency control which are the three control mechanisms are crucial in crowdsourcing. Therefore, these three control mechanisms, Quality, Cost, and Latency play a wide role in crowdsourcing operations and its optimization.

2.2.2.1 Quality Control

Quality is an important factor in any operation. In crowdsourcing, the results collected from the worker community during the crowdsourcing operations are not always reliable. Reliability has a direct link to the workers. The solutions are high in quality, the requester is satisfied, and the worker will get an incentive. But a solution with low quality will reduces the requester's chance to get a proper solution for their task, thus the workers do not get an incentive.

Some malicious workers would spoil the process by submitting poor quality or unrelated answers. Further, some of the tasks posted in the crowdsourcing platforms are complex, and some of the workers are not capable of giving high-quality answers, but still submit answers, but with poor quality. As a result, the solution pool of a particular task becomes polluted with incorrect and low-quality solutions [2].

Following figure 2.3, show the overview of how the quality factor is depicted in a simple crowdsourcing operation.

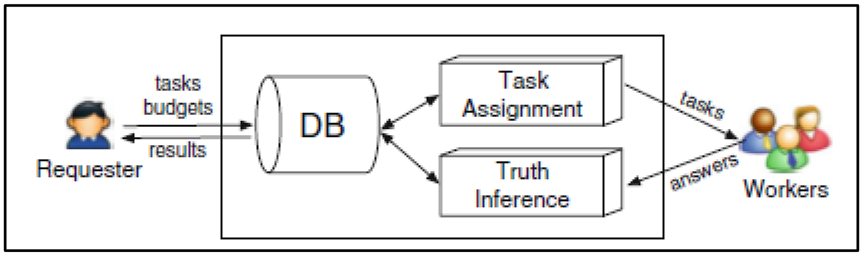


Figure 2.3: Crowdsourcing quality-control framework [2]

1. *Task Assignment:* This simply refers to the assignment of tasks for different workers. Since the workers have different backgrounds, diverse qualities, and different domains of expertise, an intelligent way of assigning tasks is vital to select appropriate workers [2].
2. *Truth Inference:* Here, when multiple workers provide answers to a particular task, these answers will be processed and does a balancing with other answers. Since some of the workers will provide low-quality answers, a mechanism to reduce crowd errors and infer high-quality results is very important [2].

The two operations mentioned above, *Task Assignment* and *Truth Inference* are the two areas where the quality factor can be optimized. According to the current work done by different scholars, several algorithms and methods have been implemented or suggested [2].

Guoliang Li in 2018, the author of Crowdsourced Data Management - Hybrid Machine-Human Computing, identifies further two subsections for the Task Assignment process.

1. *Task Assignment Setting* also known as *online task assignment problem*, refers to the context when a worker is available, which subset of tasks should be assigned for him. In *Amazon Mechanical Turk* which is a well-known crowdsourcing platform, the requester can design the task in a manner by pre-assigning a set of workers to it. This allows only a subset of tasks in the platform to be worked by a worker [2].

2. *Worker Selection Setting* also known as *jury selection problem*, refers to the scenarios when there are tasks, which set of workers should be given the chance to work on those. For example, a wide range of workers attempt to solve tasks. Workers have different qualities and budget requirements. Here, a choice is made considering the workers and decide which subset of workers are given the chance to work on tasks [2].

Most crowdsourcing platforms use Human Intelligence Tasks (HIT) when performing their crowdsourcing operations. HIT is a collection of multiple tasks where the same task can be allocated into multiple HITs,s which are set by a requester. Task Assignment focuses on the way the workers are mapped with the HITs [11]. Further going into the issue, the base decision is based on the Worker Quality.

The figure 2.4, gives an example of this process.

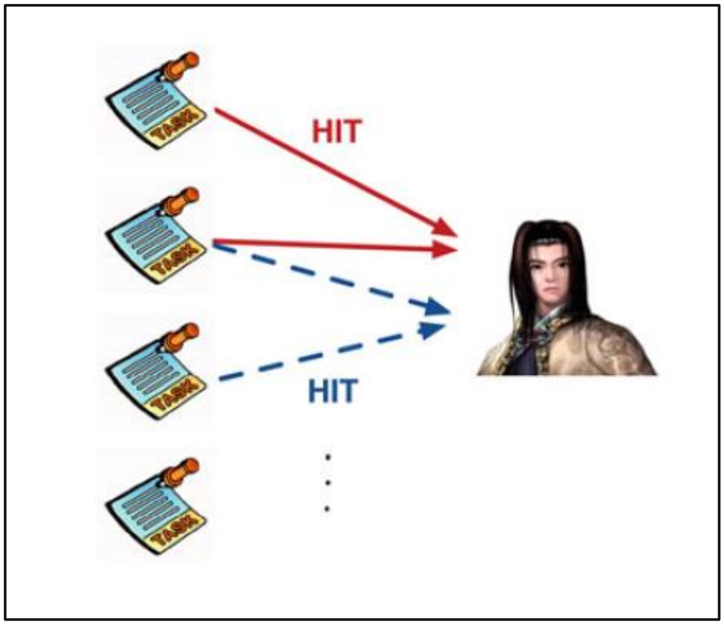


Figure 2.4: Online task assignment problem

Guoliang Li suggests that the task depending on the current worker’s quality, the task/HIT should be assigned only if the worker’s expertise is in line with the task’s domains [10]. The problem behind this process is that the same set of workers will be taking on tasks in each domain because there is no new room for workers who are new to the domain with less expertise. There is no grooming of the worker population. Because of that new knowledge sharing will be limited to a certain extend.

To maintain and observe the worker’s quality factor, Kazai has suggested that different worker profile attributes and personality traits need to be studied [12]. According to the study performed by Kazai, Age correlates with the output accuracy, while the gender does not seem to have any quality impact. But the issue with that statement is that the gender factor may affect in some cases. That is, the domain knowledge may be gender-specific, so the gender will play a role in quality factor.

The location is specific to a worker. But according to Eickhoff, the location does not affect the quality of a task solution [13]. But there is an obvious problem with that statement. If the task contains a requirement that specifically requires information on a certain geographical area, then the worker’s knowledge of that area would be vital for a high-quality solution. For example, the task is about market growth in Hambantota, Sri Lanka during the years 2011-2020. A person who is in the Hambantota area probably knows much more related development information rather than a worker who is in Sydney, Australia. So, when worker profiles are used to choose workers to do certain tasks, it is important to consider worker location as well, if the task contains any location/geographical information.

Wei Niu from the University of Michigan has suggested a Location Sensitive User Profiling which uses a dataset where it contains information mapping between different domains and locations [14]. Figure 2.5 shows such an example.

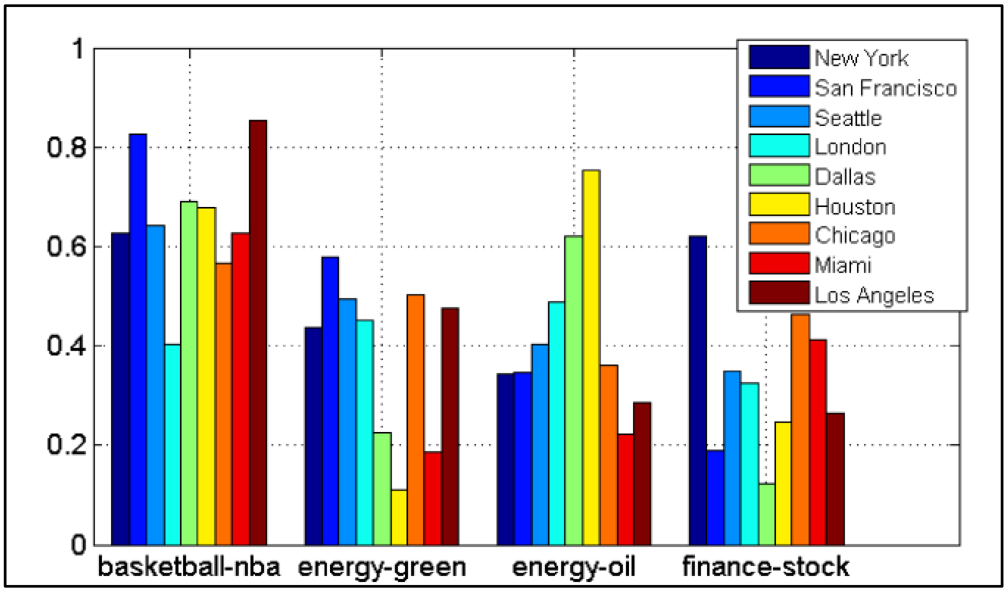


Figure 2.5: Example Tag Pairs Similarity.

In figure 2.5, the relationships of a group of tag pairs at multiple locations using cosine similarity. The x-axis shows each tag pair and each colour bar represents the similarity at a location. According to the listed domains (basketball-nba, energy-green…), some are having a location impact, such as in energy-oil, but some do not have a location impact, such as basketball-nba. The outcome is a set of workers from specific locations who can solve different tasks in different domains. So, the task assignment will be based accordingly. The downside of this method is, though it will target based on their location, it will ignore the expertise areas. For example, there could be a worker who is in Houston who is an expert in energy-green. But according to the above graph, that worker are omitted from the task assignment process [14].

Truth Inference comes once the solutions are available for a given task. This is the process in which a solution is compared with other solutions for the same task and conclude about the quality of the task. Based on the existing work, Task Modelling, Worker Modelling, and Applied Techniques are three factors that affect the truth inference [2].

Task Modelling is based on the task difficulty and the latent topic of the task. Both task difficulty and latent topics are models generated based on facts considering the worker's quality [2].

Worker Modelling is on worker's truth probability and diverse skills. Now, this is very important because the truth probability and diverse skills are two crucial factors that would have a direct impact on the solution quality [2].

Applied Techniques is simply how the above two factors, task modelling, and worker modelling would work when those two are put into together [2].

In summary, the above discussion indicates the importance of quality control in crowdsourcing. Further, the above content illustrates how past studies have done studies around quality and the outcome. One noticeable outcome is, there is a trade-off among the control types when a quality control technique is applied. That is, either the cost or latency must be sacrificed in order to achieve a good quality.

2.2.2.2 Cost Control

Crowdsourcing platforms provide a very cheap way to ask humans to get some work done. But still it is quite expensive because of the amount of work and the type of the workers. Therefore, the biggest challenge is to control the cost when performing crowdsourcing. Controlling the cost directly mean the reduction of the cost, but it is important to keep the good result quality at the same time [2].

Cost of completing a crowdsourcing job can be defined as follows.

*Cost = (Number of tasks in the HIT * Cost Per Task) * No. of workers with correct solution*

There are three parameters here, Number of Tasks in the HIT, Cost Per Task and No. of workers with correct solution.

In order to reduce the total cost, above three parameters could be reduced. But reducing the cost per task is risky. Reducing the cost may result in low quality task solutions. Scholars such as Wei Niu, Eickhoff, Guoliang Li have studied other two factors to a certain extend to come up to solutions to reduce them.

When reducing the Number of Tasks in the HIT; *task pruning, answer deduction, task selection* and *sampling* can be used [6]. But, the way of reducing the number of workers working with task is not discussed much in the literature.

Task pruning is a pre-processing technique which would use computer algorithms to analyse the tasks and decide whether each task need a human interaction to be solved. The underlying logic is, many tasks could be solved by computers, thus no human interaction is needed [2]. Apart from task pruning, a concept called worker pruning is discussed by Peng Cheng [15]. The concept talks about eliminating the workers considering the worker’s wages. The issue here is, workers who would crowdsource for less wage may return low quality results.

Threshold Selection is another concept which is discussed in literature [16]. It is a kind of a parameter which allows a trade-off between cost and quality. That is based on the budget the requester has, the number of tasks will be decided. For example, assume the requester has a budget of $50, and is willing to give $0.1 per correct solution per task. Therefore, 500 tasks could be crowdsourced in a controlled scenario. Threshold selection is this case is 500, and the most important 500 tasks will be chosen for crowdsourcing.

Answer deduction could be stated as a more logical way to prune tasks. Also, it can be used as a direct way to reduce the cost of crowdsourcing unnecessary tasks. Once solutions were submitted for certain tasks, those solutions can be used to deduce solutions for some other tasks. This would save the cost that require to crowdsource the new tasks. In answer deduction technique, there are two major operations; Iterative Workflow and Presentation Order [10].

The Iterative Workflow, the algorithm is designed in a way that iteration of tasks will be sent out to the worker community to be solved. At the end of each iteration, using the solutions for the tasks, the unsolved and non-crowdsourced tasks will be deduced. Figure 2.6 shows such iteration behaviour.

Step 1: Pick up some tasks with no solutions from a task pool.

Step 2: Collect their answers from the crowd.

Step 3: Deduce the answers of some other tasks in the pool.

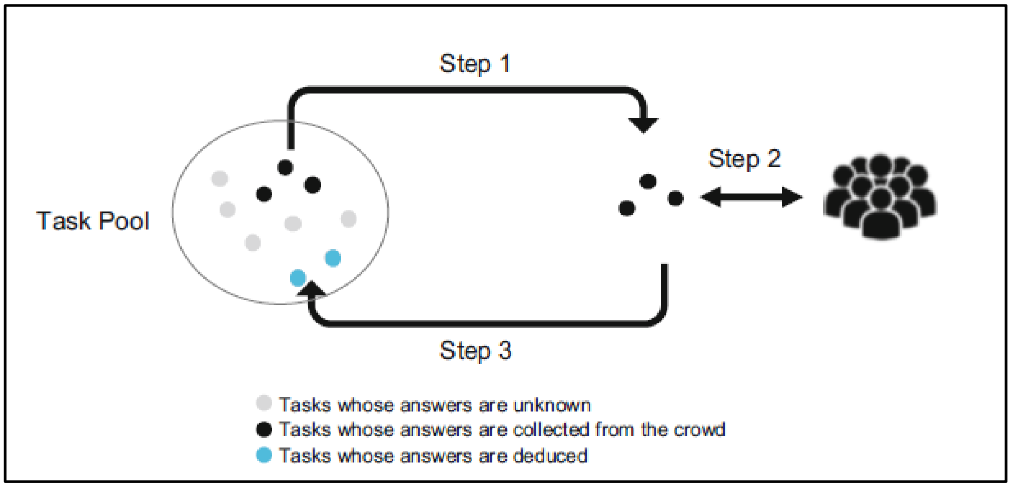

Step 1
Step 2
Task Pool
Step 3
● Tasks whose answers are unknown
● Tasks whose answers are collected from the crowd
● Tasks whose answers are deduced

Figure 2.6: Answer-deduction workflow

The challenge is the process of choosing which tasks to be crowdsourced. The problem in this method is, if the correct set of tasks is not chosen in the step 1, a complete crowdsourcing will have to be done, and the iterative method will not be a viable solution.

The Presentation Order is important in the Answer Deduction and as well as in Iterative Workflow. Here, the order the tasks need to be chosen to be crowdsourced is discussed. That is, if chosen a wrong order, it will be a complex operation. Though the presentation order has a major impact on cost reduction, incorrect practice would result in negative results.

The downside of task pruning is, it may neglect important tasks which should involve humans to be solved. As a result of this, the quality of the overall process is reduced [2].

The Answer Deduction technique avoids unnecessary crowdsourcing operations, thereby reduce the overall cost. But, at the same time, this technique has negative impact on quality and latency. Quality will be impacted negatively if the crowd answers a task incorrectly and if that solution is used for the deduction of another task. In terms of latency, since the complete set of tasks are not sent to the crowd at the same time and since it uses an iterative approach, it will take much more time since it may not fully utilize the workers who are available [10].

Task Selection is another process mentioned in literature which can be used for cost reduction as well as for quality improvement [2]. The idea behind this is, once the most important tasks are identified to be crowdsourced, those tasks only can be sent to the worker population and the rest of the tasks could be solved by other means. Due to its effectiveness in cost optimization, task selection strategy is used and widely studied in a large variety of crowdsourcing operations [6], [10], [16].

According to Guoliang Li, Task Selection strategies can be categorized as Model Driven and Problem Driven [2].

The model driven task selection strategy is also known as query strategy in some literature. Query Strategy is an algorithm that decides which of the data points need to be labelled. Query strategies are complex and there is a direct connection to machine learning. One of the widely use query strategy is uncertainty sampling. This process would select unlabelled data point whose label has the highest uncertainty in the current model [17].

The problem driven approaches are much simpler compared to model driven approaches. In problem driven approach, solving a task takes a hybrid path. That is, first the task will be solved to a certain level by the computers/machines and then will be sent to the worker community to improve the solution as much as possible [2].

Similar to Answer Deduction technique, the task reduction technique will also affect the latency factor negatively. The issue here is, the use of iterative approach to decide which tasks to be crowdsourced next. Each iterative process which use a limited or small number of tasks to be crowdsourced which would ignore the fact the availability of a large pool of crowd workers [10].

Some individuals chose to be workers to do jobs not only for money; but for fun and to learn something new. A Non-monetary Incentives could motivate such workers to take on tasks and work on them. Designing the task in a form of a game would be interesting for some users [14].

The main disadvantage in this technique is that, same design concepts, or game designs to make the worker community interested would be rather difficult to keep up. That is, with time may will lose interest on the game. In the other hand, if a new interface is brought up, the workers will have to invest more time to study the interface before attempting the task. Since time is wasted, the workers would expect more reward as well. Thus, this technique has a latency issues and may affect the cost as well [2].

In summary, the above literature states the importance of cost controlling and probable cost controlling techniques currently used. But, according to the literature, when optimizing the cost factor, there is always a trade-off between quality and latency. That is, either quality or latency or both may have to be compromised to reduce the cost of crowdsourcing [2].

2.2.2.3 Latency Control

So far, the contribution in terms of quality and cost was discussed. Methods such as task assignment and truth inference optimize the quality while methods such as task pruning, answer deduction, task selection and sampling reduce the cost factor.

Latency refers to the total time taken to complete a specific task. Since humans are slower than machines, even to complete a simple task, for example, labelling a set of unknown images, will take hours or even days. Thus, another big challenge in crowdsourcing is to minimize the latency factor. That is, the job completion time should be minimize while still keeping a good quality and low cost [2].

Guoliang Li discuss about who has done vast research on crowdsource data management optimization discuss on three categorize of Latency Control. *1. Single Task Latency Control. 2. Single Batch Latency Control. 3. Multiple Batch Latency Control* [16]. Figure 2.7 illustrates their differences.

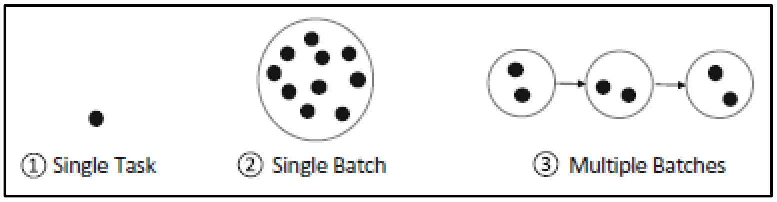


Figure 2.7: Three classes of existing latency-control techniques

The Single Task Latency Control focus on reducing the latency of processing one task. For example, labelling each individual image of a set of images.

The Recruitment Time is a factor which affects the latency of a single task. It refers to the time taken from the point which a task is posted in a crowdsourcing platform until it is accepted by a worker. The goal is to reduce the recruitment time. For this, Bernstein et al has suggested the retainer model. The retainer model would notify all the workers available in the platform that a new task has being posted, and a small amount of money would be awarded for just opening that task. The advantage of this method is, workers would quickly start on crowdsourcing the task. But it will create low quality task solutions since random workers would start working on the tasks. Further, the cost factor will be affected negatively because the model suggest to pay for the workers just to open the task [18].

Some crowdsourcing platforms have implemented a concept called *Qualification Test Time* to overcome the quality issues which affected by the retainer model. The idea behind his is, before attempting the actual task, the worker has to solve a number of tasks in which the actual solution is known. Once done, the worker will be permitted to crowdsource the actual task only if the trail tasks are a success. Though this will address the quality factor, it will have a severe negative impact on the latency [2].

Another factor which affect the latency is the Work Time, that is the time taken by a worker to complete a task. One already being discussed method would be designing user friendly interfaces for the workers to react easily [16] .

The Single Batch Latency Control focus on reducing the latency of processing a batch of tasks. The actual goal is to reduce the latency of the last set of tasks in a batch. In this method, there are another two sections coming into play; Statistical Model and Straggler Mitigation.

In *the statistical model*, the tasks latency is modelled using statistics and use other strategies such as good pricing to reduce the latency. For the model, two delays are addresses. 1. The delay which occurs from the time of posting the task to the point the first solution is available. 2. The inter-arrival time of solutions. The model suggests different strategies to overcome these tow delays. Issue here is the strategies used for the optimization. For example, using a higher price per task would have negative impact on the cost factor [19]. Further, Guoliang suggests using time limits to attempt a task. That is, depending on the difficulty of the tasks, the platform or the requester could state a time limit for the task to be active. Once that limit is exceeded, the worker won't be allowed to work on the task. The problem is, this limit would make the solutions less in quality [2].

The Multiple Batch Latency Control focus on reducing the latency of multiple batches of tasks. This method inherits its method from the iterative process. Unlike in iterative process, Multiple Batch Latency Control method will iteratively be sent multiple tasks as a batch to the worker community.

The figure 2.8 illustrates such mechanism.

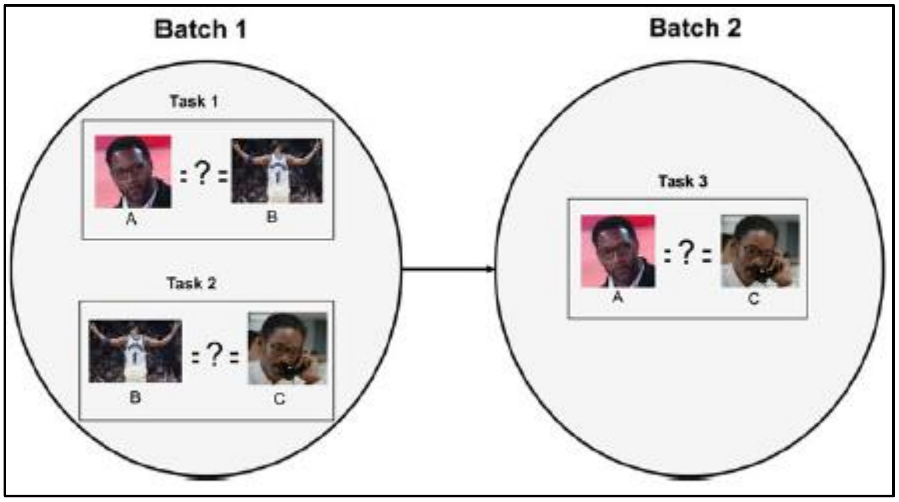


Figure 2.8: An example of multi-batch latency control.

The advantage in this case is, once the Batch 1 is done, the crowdsourcing for batch 2 is not necessary since solution for the Task 3 can be deduced using Task 1 and Task 2 solutions. Thus, latency is improved.

The fundamental problem here is the balancing of the trade-off between cost and latency. That is when using the Multiple Batch Latency Control mechanisms there is no way to optimize both cost and latency at the same time. For example, the best way to improve latency is to design one batch with all of the tasks, but it will have a negative impact in cost since the iterative process cannot be applied. On the other hand, publishing the batches with single task improve the cost but it will omit the possibility of crowdsourcing in parallel, thus the latency is affected [20].

Verroios et al. has mentioned that, latency can be optimized by batching multiple tasks in one round. That is, the same task could be repeated in multiple batched. Accordingly, the latency could be optimized by increasing parallelism at the cost of redundancy [21].

In most of the literature, the latency optimization is always having an effect on the cost and the quality. These investigate different methods to balance the trade-off between the latency and cost. Similarly, the latency-quality trade-off is also being discussed. This means that, to reduce the latency of crowdsourcing, either the cost will be high, or the quality will be poor. So, the existing studies focus on different problem settings to overcome this issue. 1. Optimizing the quality, given a fixed cost. 2. Minimize the cost by sacrificing the quality a bit. 3. Minimizing the cost with latency and quality constraints [2].

In summary, the latency control mechanism suggested in the literature will optimize the latency factor to an extent, but the current studies have failed to come up with a good solution to optimize the latency without a considerable impact on cost and quality.

2.2.2.4 Trade-off between control factors

According the literature, when optimizing each of the control factors; quality, cost and latency, there is always a trade-off against other factors. That is, when the quality optimizing techniques are followed, it would affect the cost and latency. Similarly, when the cost is optimized, the quality and latency will be affected. Similar effect is visible with latency optimization.

This study will try to address this problem, that is optimizing each of the all the three control factors, quality, cost and latency, by minimizing the negative effect on each of the control type.

2.3 Chapter Summary

This chapter discussed how the literature and current setup in crowdsourcing platforms has identified the optimization issues in crowdsourcing. In all of the literature, the three control factors; Quality, Cost and Latency is discussed, and said to be the main point of optimization.’

Further, different techniques have being identified by different scholars to achieve crowdsourcing optimizations. These techniques have both positive and negative impact on the above mentioned three control factors.

The ultimate goal should be optimization of crowdsourcing operations without a negative impact on the control factors; Quality, Cost and Latency. This will be achieved by a set of objectives. 1. Worker rating processing, 2. Worker alert system, and 3. Batch scheduling recurring tasks.

Next chapter explains the steps followed in identifying and applying best methods to address this study.

CHAPTER THREE

METHODOLOGY

3.1 Chapter Overview

This chapter discuss how the project quest has been handled and addressed with the knowledge gained by the literature review. As mentioned in the Chapter 1, this project has spread over different domains in the computer science field. Different strategies have considered to meet the goals and objectives of the project. As mentioned in the previous chapters, the goal of this project is to optimize the crowdsourcing operations. To achieve this goal in terms of quality, cost and latency, objectives such as *1. Develop an efficient crowdsourcing platform with user profile based approach. 2. Incorporate an effective notification deliver system within the platform. 3. Develop the capability of handling similar crowdsourcing tasks* will be achieved.

This chapter will also provide an in-depth analysis on the research problem, the proposed solution's design, and the implementation steps which have been carried out to make the project goals and objectives a success. Further, the evaluation plan to verify project outcome will be discussed in detail.

In summary, this chapter would give a clear and comprehensive overview of the steps taken to implement the work related to the project in order to make it a success.

3.1.1 Representation of the Problem

The aim of this study is to find an optimization solution for crowdsourcing operations. The goal is to optimize the crowdsourcing operations in terms of quality, cost and latency without having any compromise between each other.

As mentioned in the previous chapters the optimization of crowdsourcing platforms should be based on quality, cost and latency. This section will identify some of the key problems the crowdsourcing platforms face which are not fully addressed by previous scholars.

Crowdsourcing platforms have been active for many years, and millions of crowdsourcing tasks have been addressed and solved using the platforms. Daily, thousands of new tasks will be processed using these platforms. Because of that, there is a high probability that similar, sometimes the same tasks are posted in the crowdsourcing platforms. Since someone previously has already done the crowdsourcing, it is not ideal to do crowdsourcing again in terms of efficiency. An effective method is required to search and map already crowdsourced tasks with the new ones and provide the requesters the ability to get solutions without even doing crowdsourcing for their tasks. This will directly ensure that time and cost is optimized since there is no crowdsourcing process involved. Further, since the previous solution is accepted by a previous requester, it can be said that the solution is good in quality.

Workers are not always online. Therefore, there is a high probability that the latest tasks will be missed by the workers who are cable of giving quality answers. So, it is important to notify the worker's when a new task is posted in the platform. But workers would not want to be notified always for all the tasks since they may not have any interest nor expertise to do every task. So, notifications should be made based on certain defined attributes. Maintaining a proper notification system would make sure that proper workers would attend to the tasks promptly and provide with quality solutions. Further, if the notification system omitted providing notifications to all the workers, and target a proper worker pool, spamming of solutions can be reduced thus, cost will be reduced.

There are instances where the same task is required to be crowdsourced repeatedly daily or weekly. For example, currency exchange rates. So, if a requester required such crowdsourcing, daily the requester must post tasks in the platform. It will be a troublesome as well as a time consuming that. Requirement is there to create and posts task automatically for recurring tasks. If such automatic process is available, time factor could be improved greatly.

When viewing solutions for the tasks, there are features to group and order the tasks based on the date of submission of the solution and the worker's identity. But such grouping would not ensure that quality results are appeared first. When a requester goes through the task solutions, it is important that proper and valid solutions are viewed quickly. So, it is important to order and group the solutions based on a quality index. Since the solution is a product of the workers, maintaining a quality index with respect to a worker is important. Such index could be utilized to group the solutions. Using such mechanisms make sure that the requesters come across with proper solutions at the early stage of answer evaluation.

So, it is clear that maintaining a worker's profiles based on their interests, expertise, performance and past crowdsourcing information is important and can be used further to optimize the crowdsourcing processes.

3.2 Analysis and Design

3.2.1 Proposed Model and Design

In the proposed model, the goal is to create a framework/solution which can be recognized as an add-on that can be integrated with any crowdsourcing platform. As mentioned, it has two phases, that is, the optimization of the requesters' side and the optimization of the workers' side.

The proposed solution will be done in two steps. The requester side optimization and the worker side optimization.

The requester's side optimization would focus on one main aspect, that is, when a new task is entered the platform would suggest previously crowdsourced tasks and solutions which matches the pattern and form of the newly entered task. This is not properly addressed in current systems and it fails to identify previously crowdsourced tasks based on the key words and entities of the task. So, the proposed solution is capable of analysing the new task using NLP, extracting entities and different keywords from the tasks and map it with other available tasks and solutions. In addition, repetitive tasks will be handled automatically. During the initial submission of the task, the requester must specify the repetition frequency so that system has the relevant information such as frequency of the repetitive task. This information will be used by the proposed system to automatically post the tasks to the platform. Such method of repetitive task handling is not discussed properly in current systems.

The worker side optimization is modelled considering more parameters. Main feature that is considered here is using worker profile information to optimize the processes. Current crowdsourcing platforms utilize the worker's country to filter out tasks and even confirmation of the account is done based on the country of the user. Further, the workers could subscribe for certain type of tasks. The problem of subscription is that workers could subscribe for any type of tasks, even the ones which they do not have any experience with. The proposed solution will utilize some of the feature and enhance the existing method by allowing the users to define their expertise and interested areas at the user registration and use a notification system to notify when such task is available. Though a similar behaviour can be expected by the worker population, with the proposed solution, the platform will maintain a worker rating system based on worker's previous solutions. So, the given solutions will be prioritized based on the rating. This will make sure that spamming of solutions is not happening. The quality and latency factors will also be affected by the number of tasks that a worker is assigned to. A threshold value to limit the number of assigned tasks for the worker will be maintained considering worker's ability to handle multiple tasks at once.

3.2.1.1 Use Case for the Requester

This section discusses how the requester get involved with the proposed system. Mainly, the requester *Add New Tasks, View Solutions and Receive Notification* when a solution is ready.

The crowdsourcing operations are started by the requester. The figure 3.1 show requester's interaction with the proposed system.

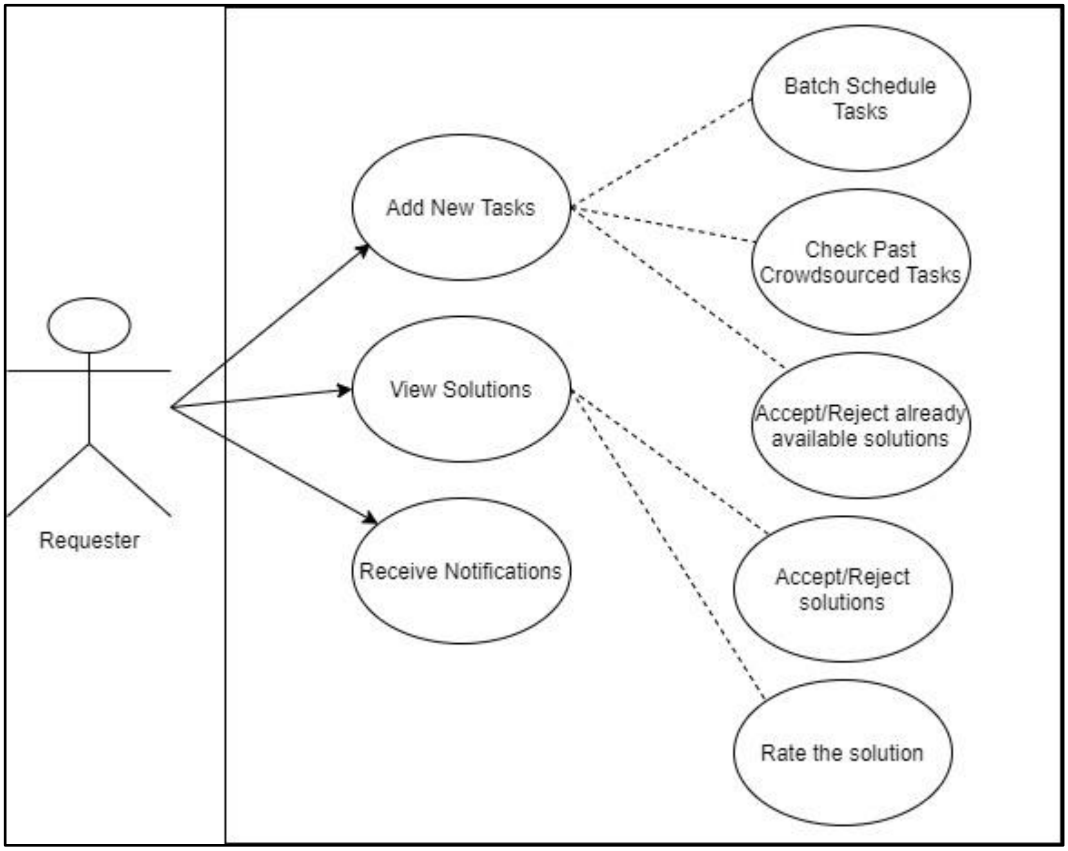


Figure 3.1: Use Case for the interaction of the requester with the system

As illustrated in the above use case diagram, the requester mainly interacts with the system to *Add New Tasks, View Solutions* and *Receive Notification when a solution is ready*. The use case shows that, there are sub tasks associated for Add New Tasks and View Solutions operation.

Depending on the situation, the requester would interact with those sub tasks as well. For example, the requester can batch schedule the tasks, check past crowdsourced solutions for the task, etc.

3.2.1.2 User Case for the Worker

This section discusses how the worker get involved with the proposed system.

The worker is another key role in the crowdsourcing operation. Once a requester posts a task in the crowdsourcing platform, it is the worker who provide solutions. Figure 3.2 below shows the worker's interaction with the system.

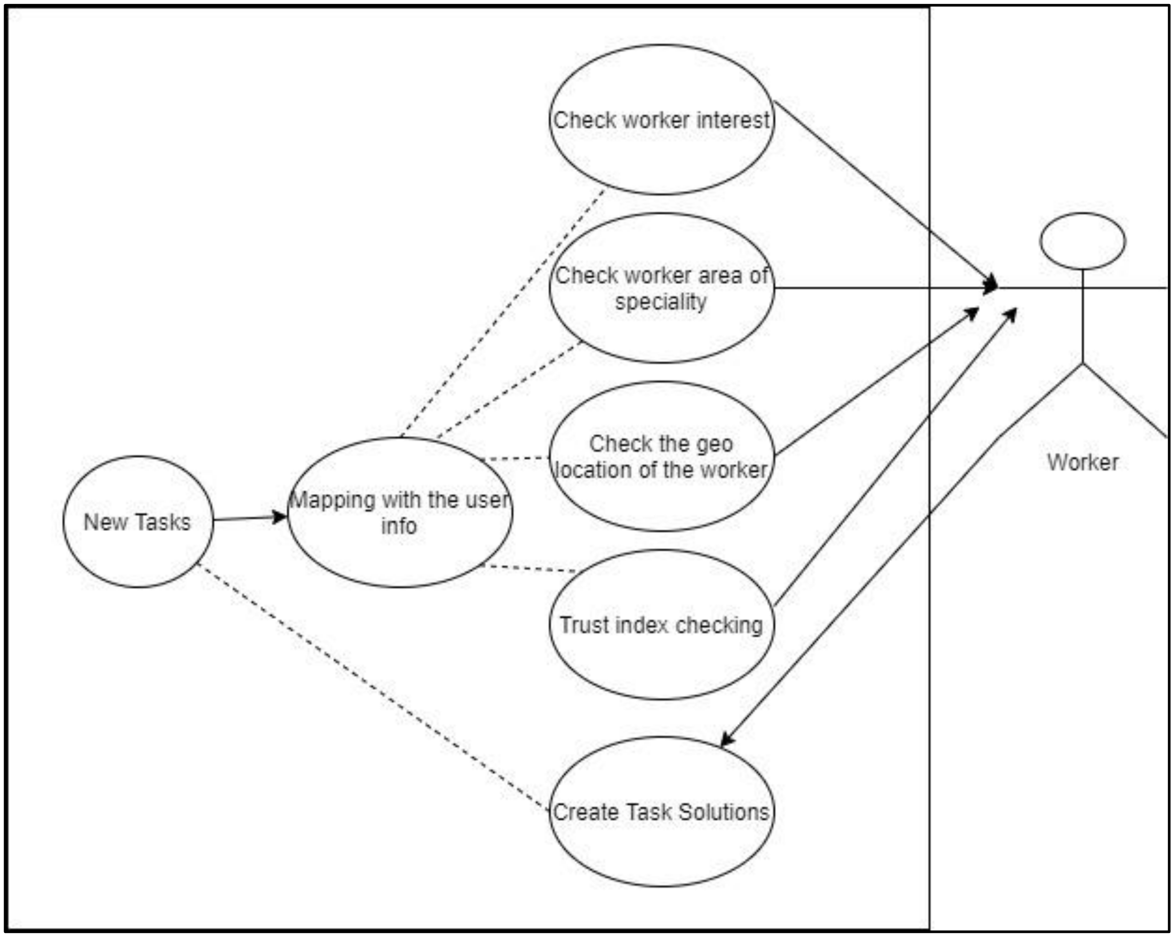

Check worker interest
Check worker area of speciality
Check the geo location of the worker
New Tasks
Mapping with the user info
Trust index checking
Create Task Solutions
Worker

Figure 3.2: Use Case for the interaction of the worker with the system

As illustrated in the above use case diagram, the main interaction of the worker with the system is by providing solutions for the tasks enters by the requester, that is *Create Task Solutions*. For the system to work smoothly and to ensure the proposed solution to work according, addition subtasks are associated with the user, for example, *Trust Index checking* and *Check worker interest areas*. These sub tasks are not mandatory, but it will directly affect the performance of the system.

3.2.1.3 Overview of the Proposed System

As mentioned in the above two sections, the requesters and workers interact with the system in different ways. The figure 3.3 shows an overview of how different components are connect with the proposed model.

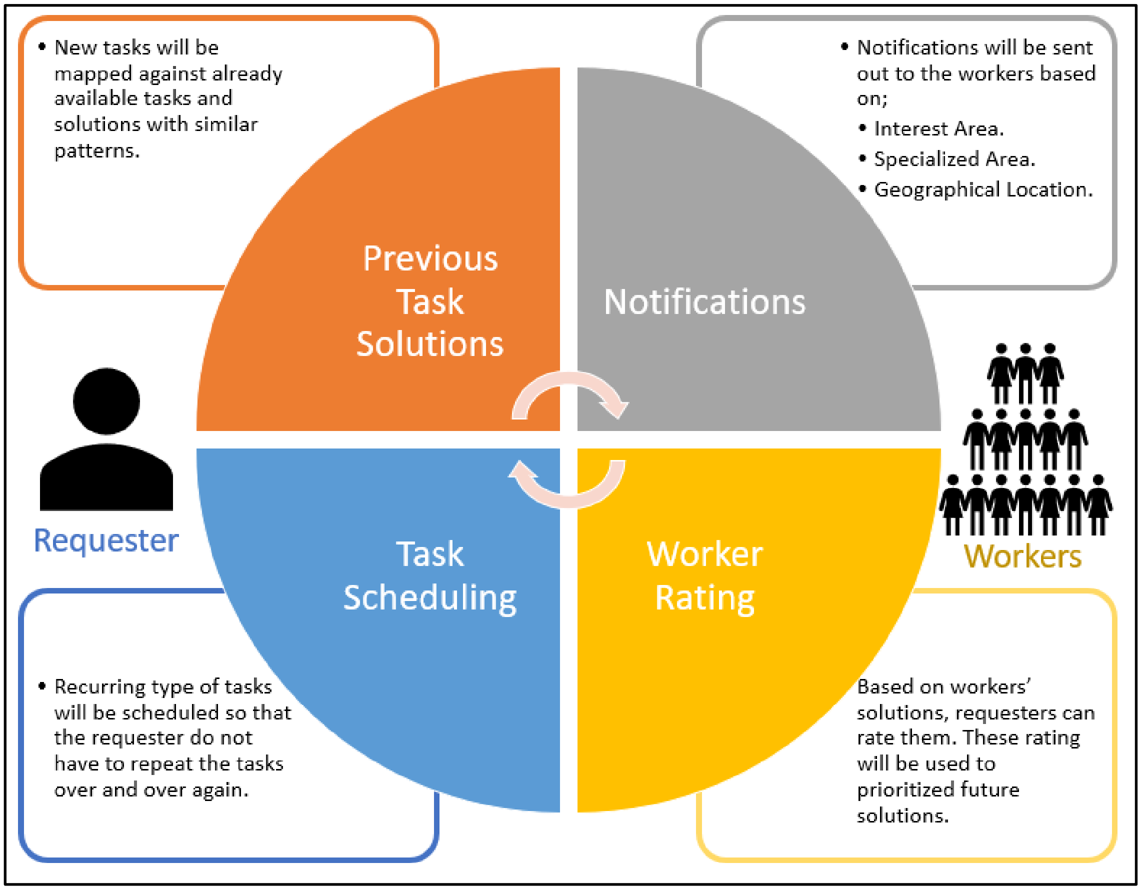

• New tasks will be mapped against already available tasks and solutions with similar patterns.
• Notifications will be sent out to the workers based on;
• Interest Area.
• Specialized Area.
• Geographical Location.
Previous Task Solutions
Notifications
Requester
Workers
Task Scheduling
Worker Rating
• Recurring type of tasks will be scheduled so that the requester do not have to repeat the tasks over and over again.
Based on workers' solutions, requesters can rate them. These rating will be used to prioritized future solutions.

Figure 3.3: Overview Diagram for the Proposed Model

As shown in the diagram, there are two main parties in a crowdsourcing operation; *Requesters* and *Workers*. Considering the facts mentioned in the 3.1.1 section, in the proposed design to optimize the performance, few key aspects have been focussed upon.

1. *Previous Tasks Solutions*

   Once a requester enters a new task, using the keywords and entities it contains (automatically generated by the system or can be manually entered by the user), the system would query past tasks and their solution which maps with the key words.

2. *Tasks Scheduling*

If the same task is needed to be recursively posted in the crowdsourcing platform in a given frequency of time, the requester could create a batch schedule. The system would use that information to automatically to create new tasks and post in the platform.

3. *Notification Mechanism*
    a. *Process according to the Interested Areas*

The worker's profile contains his/her interested areas or domains. When a requester enters a new task, using the keywords and entities it contains (automatically generated by the system or can be manually entered by the user), the system would send alerts to such users whose interest areas are mapped with the keywords and entities extracted from the task.

b. *Process according to the Specialized Areas*

Similar to above scenario, based on worker's specialized area (using the information stored in worker profile), alerts will be sent to the worker's whose specialized areas are mapped with the keywords and entities extracted from the task.

c. *Process according to the Geographical Location*

Some tasks are location specific. For example, a task would be, population variation in Kadawatha area in the past 10 years. These kinds of examples could be solved comparatively easily by a person who lives in Kadawatha area. So, in such scenarios, based on worker's location, alerts will be sent to them.

*** The processing of notification in terms of interested areas, specialized areas and geographical location, it is important to have certain balance between them. The requester will be given the option whether to utilize one or more notification functionalities or not to utilize any of them during the task submission. So, unless the requester opted for such behaviour, no notification will be sent for the workers.

4. *Trust Index/Rating per user*

For each solution provided by the worker, the requester can rate the solution from 1 to 5. This rating is done per domain. When there are multiple solutions per tasks, each solution will be sorted according to the respective worker's rating. That is, solution with highest worker rating will be shown first.

3.2.1.4 Architecture of the Proposed System

The architecture of the design can be illustrated as follows.

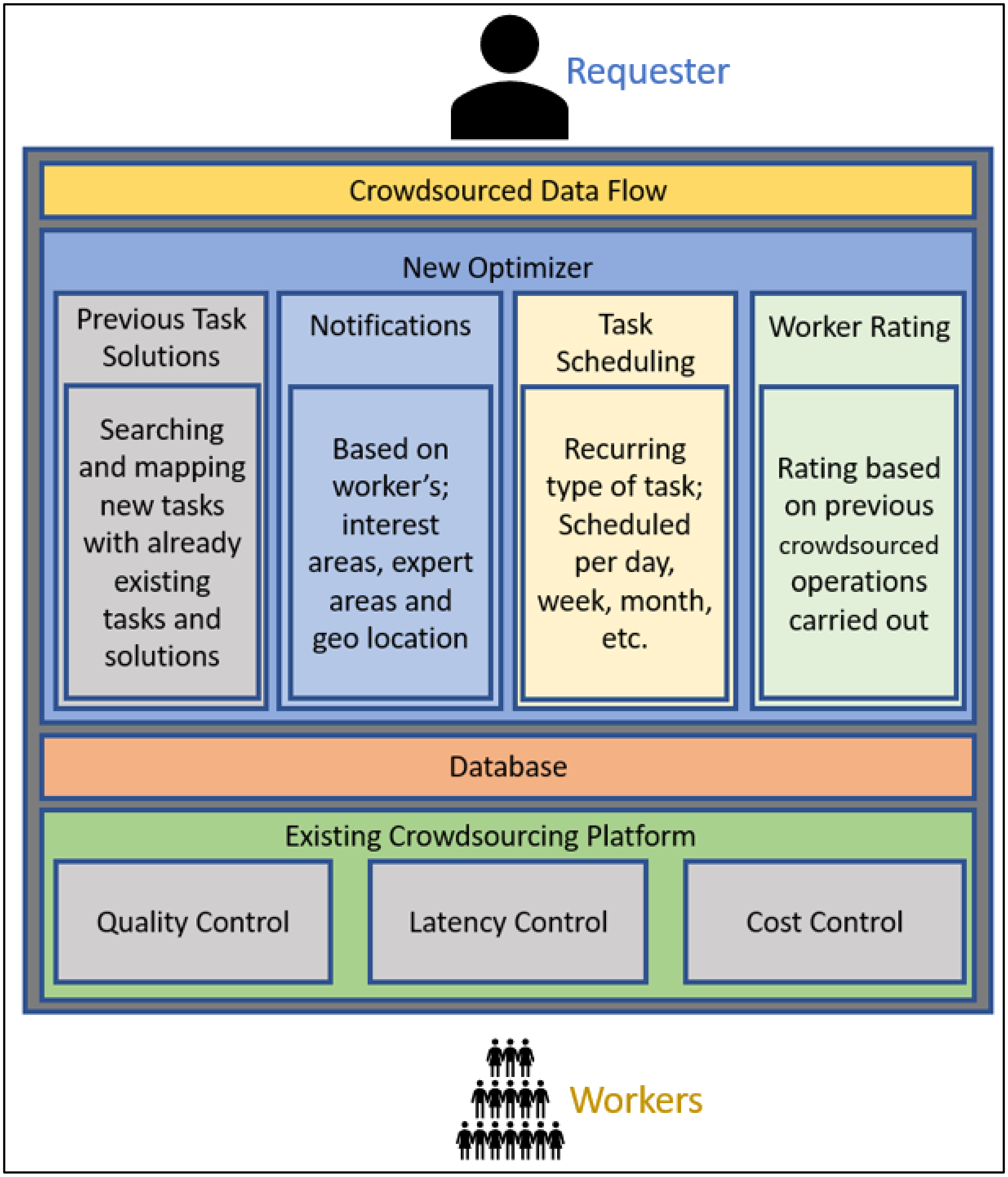

Requester
Crowdsourced Data Flow
New Optimizer
Previous Task Solutions
Searching and mapping new tasks with already existing tasks and solutions
Notifications
Based on worker's; interest areas, expert areas and geo location
Task Scheduling
Recurring type of task; Scheduled per day, week, month, etc.
Worker Rating
Rating based on previous crowdsourced operations carried out
Database
Existing Crowdsourcing Platform
Quality Control
Latency Control
Cost Control
Workers

Figure 3.4.1: High-level architecture of the proposed solution.

The proposed system will behave as an extended system to the existing crowdsourcing platforms. That is, the deployment of it will be on top of the existing platforms as an additional layer. As mentioned in the figure 3.4.1, the new optimizer contains 4 sections; *1. Previous Task Solutions, 2. Notifications, 3. Task Scheduling and, 4. Worker Rating*. These 4 can be considered as the 4 pillars of optimization in the proposed solution.

3.2.1.5 The complete workflow of the proposed system

The proposed system utilizes the current behaviour of crowdsourcing, that is, the requester enters a task and the worker provide a solution. The proposed system will optimize this process by including sub processes to the flow. These sub-processes are mention in the above section (3.2.1.3).

The complete flow of the proposed system is as follows.

Prerequisites: The Requester and the worker should create user profiles. Information related to worker's location, interested domain areas, expert domain areas and contact information are mandatory aspects in the worker profile. Without this information, the proposed system would not work as intended

The following flow chart, figure 3.4.2 illustrate the complete flow of the proposed model. Here, all the components and operations proposed by the new model starting from new task entering to answer evaluation process is illustrated.

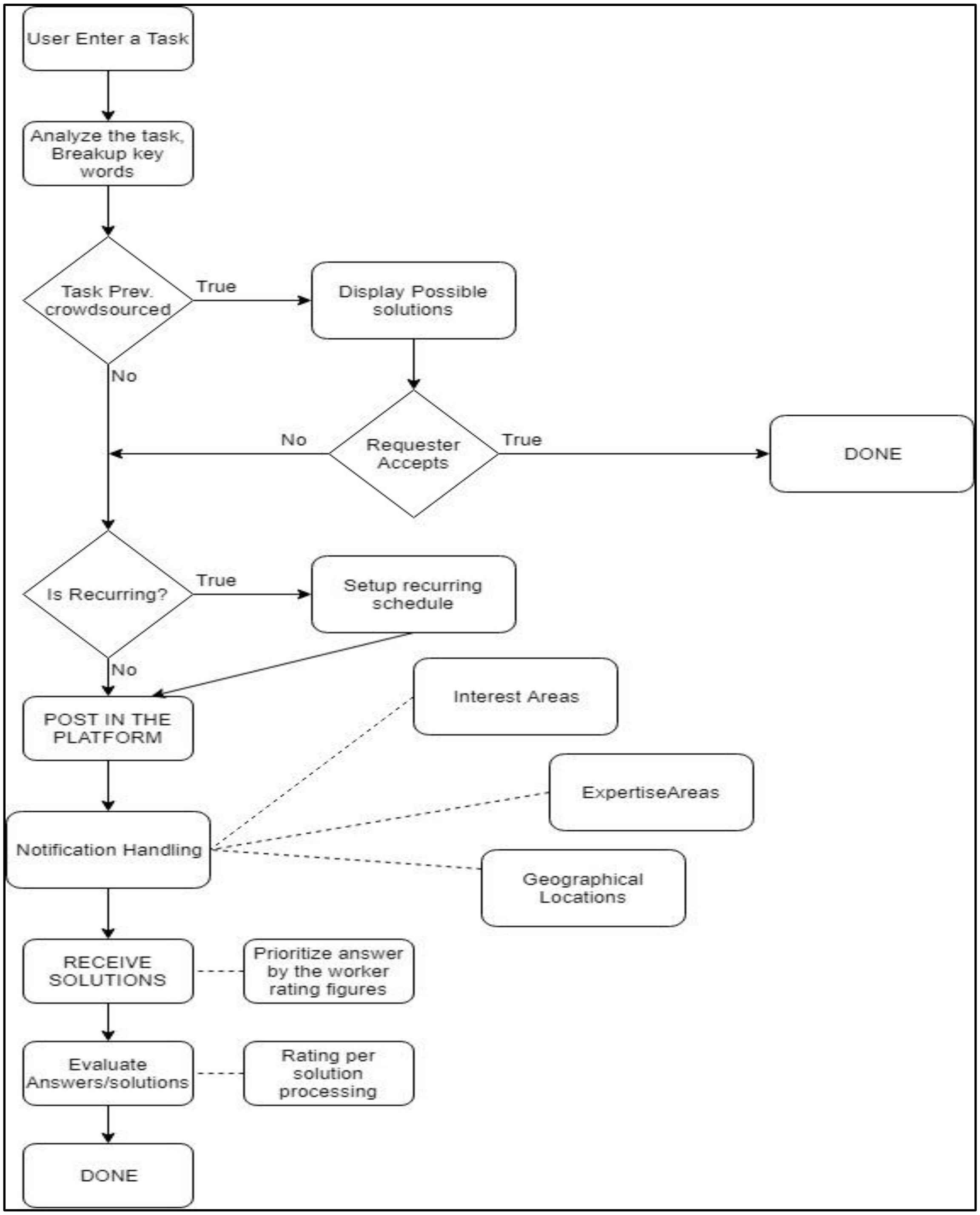


Figure 3.4.2: Proposed Model – Flow Chart

- <u>User enters a Task.</u>

  The process starts with the requester enters a new task. Once the task is entered, the system will extract keywords and entities from the task title and task description. As soon as the user submit the task to the system, the keywords and entities thus extracted will be used to create a connection with already processed and solved tasks. If such solutions exist, the system would show them to the requester. The requester has the option to either stick with one pre-crowdsourced solutions or to perform a new crowdsourcing task. If the requester chooses an already crowdsourced task's solution as the task solution, there is no need of further processing, and the crowdsourcing operation will end there.

The objective addressed here is, by performing pre crowdsourced tasks mapping, the time taken for a whole new crowdsourcing operation will be eliminated, thus the time factor is addressed. Further, the cost associated will be saved since there won't be a need to compensate the workers.

- <u>Recurring Tasks.</u>

  If the requester is not satisfied with the solutions, he/she can continue the crowdsourcing process and submit the task to the platform. If the task is a recurring task, that is if the requester wants to post the same tasks repetitively in a frequency of time (daily, weekly, monthly, etc.), he/she can setup those setting before the task submission. This is a sub process that will be handled automatically as a background process without any user intervention. If a task is set as a scheduled batch task, the system will handle posting the task according to the schedule created. That is, if the schedule states that the task should be posted weekly, a new task will be created weekly and get submitted to the platform automatically.

  The objective addressed mainly here is the time factor. Since there won't be a necessity to enter the same task repeatedly, the time will be saved.

- <u>Notification Mechanism</u>

  When the submission takes places, the system will do the following. A list of workers who are most suitable to answer the new task will be created considering the following. The number of the worker's who gets the notification could be specified by the requester. If specified, the topmost qualified worker's based on the following points will receive the notification, else all the worker's satisfying the following conditions will receive the notifications.

  1. If the task contains any geographical information (city, country, etc.), that information will be checked against worker's location information.
  2. Worker's specialized areas, expert areas and interested areas will be considered and connect them with the keywords and entities extract with from the task title and description.
  3. Worker's trust index and success answer rate (a numerical figure of 1 to 5) which will be depicted by the worker domain rating will be considered. That is, worker with high rating will be focussed.

> Once the list of candidate workers is created, notifications will be sent out to the worker's in the list stating that a task which they would be interested in is posted in the platform. To send out the notifications, the contact details provided by the worker's will be used. That is, the email, mobile number or if they are online at that moment, a simple toast notification could be utilized.

At this moment, workers can assign tasks, work on them and submit solutions. Multiple worker can assign a single task, so ideally there could be multiple solutions and probably multiple correct/acceptable solutions. So, when the solutions are viewed by the requester, these solutions will be prioritized according to the rating per domain area, for example the rating for User A in computer Science domain. That is, solutions provided by workers whose rating in the task's domain is high, will be given priority and shown on top.

At this point the process is done and the requester can view the tasks, mark acceptable solutions and rate the solutions as well. Rating the solution is a key feature operation in the proposed system. This information provided by the requester will be used in future crowdsourcing operations.

So far, we have discussed the nature of the proposed model and design including all the characteristics and entities related to it.

3.3 Dataset Management

The dataset requirement for the proposed model is discussed in this section.

Though the proposed model's concept is valid for any language, tools provided for certain operations does support only English. So, the tasks and the solution must be in English language for the proposed solution to work correctly.

Here, the different types of data required for the model will be discussed with examples.

3.3.1 Choosing a valid dataset

A valid and realistic set of data is required for the proposed model to be run against. Mainly, the following data which are associated with crowdsourcing is required.

1. User profile information.
    a. Worker information
        i. Name of the worker
        ii. Contact and address information with valid city and country

        iii. Interested domain areas (for example, Computer Science, Music, Biology)
    b. Requester information
        i. Name of the worker
        ii. Contact information
2. Tasks to be crowdsourced.
    a. Task title
    b. Task description
    c. Keywords related to the task title (for example, Database, Marketing, Currency)
    d. Solutions per crowdsourced tasks.
3. Solution Description
    a. Worker associated with the task solution
    b. Rating of the task based on the worker

3.3.1.1 User Profile Information

The user profiles are of two types; *Workers* and *Requesters*. Since these two parties interact with the system in different ways, it is important to maintain a dataset unique to each party.

Information related to the Requester does not contain critical information. Basic personnel information such as name and contact information will be enough since there is no operational specific behaviour is dependent on the information provided.

Information related to the worker is an extended version of requester information. Apart from the basic information, some mandatory information should be addressed such as the contact information. Additional information such as worker's interest and expert areas is also required. This additional information will be used in the proposed system to perform the intended functionalities.

3.3.1.2 Tasks to be crowdsourced

The tasks play a major role in the system. As mentioned in previous sections, tasks will be analysed upon entering, and keywords and entities will be extracted. So, the standard of the tasks matter. The standard refers to the grammar, spelling and the structure of the task title. These should be correctly specified for the system to identify the task.

The importance of the standard of the task is, the task title will be used to extract main entities and keyword. These entities and keywords will be used by the system to build a mapping with already crowdsourced tasks. So, unless the task title is not properly set, entity extraction would not proper work, thus it will reduce the efficiency of the system.

3.3.1.3 Solutions for Crowdsourcing Tasks

There could be multiple solutions per task which are submitted by different workers. These solutions will be shown to the respective requester once the solution is submitted. Along with the solution, the related worker’s rating per the domain area of the task will be recorded. This rating information will be used to group and order the solutions when the requester tries to view them. That is, task solutions with higher rating value will be given priorities and will be shown on the top solution list.

3.3.2 Summary

In summary, the dataset associated with the proposed solution required to be in English in order for the intended functionalities to work correctly. Further, the validity and the standard of the tasks is valuable.

3.4 Implementation

This section will discuss the implementation of the proposed model using a commonly known programming language. When it comes to a crowdsourcing platform, it is a web-based application which utilizes web features. That is, the front end of the application should be web-based so that thousands of people could easily accessed. But the backend logic can be written in any language since there is no language specific coding related to the proposed solution.

3.4.1 Development Technologies and Tools Used

When implementing the proposed model, any recognized programming tool could be used. For the ease of development, Java was used as the programming language for the implementation The solution follows a framework style, that is, it should be easily mapped with properly design web-based frontend.

One reason for using Java is its ability to easily map with another application. In this case, since the front end must be a web-based, it would be quite easy to map a java framework/application with a we-based application with a set of small adjustments. As the IDE, Eclipse IDE was used.

To extract keywords and entities from the tasks, a 3rd party NLP tool known as TextRazor was used. This is a powerful web-based tool, which can be used to extract keywords and entity information from a phrase.

As the database, a MySQL Server Database is used which is hosted in Microsoft Azure. As one of the leading Cloud service providers, using Microsoft Azure can be quite easily used for most databases and is a powerful tool specially for web-based applications.

3.4.1.1 Pricing and Licencing of the technologies and tools

This section will present the pricing and licencing information for different technologies and tools that are used during the implementation process.

***Java*:** As the development is align with Oracle Java Free Licensing agreement, there is no payment required for this.

***Eclipse IDE*:** Eclipse has a free version that could be downloaded. So, there is no additional payment required.

***TextRazor Tool*:** TextRazor provides 500 free phrases processing per day. In the development stage, it is unlikely to exceed this limit. So, no fee is required. But in the long run, there could be many requests to use the tool. If so, package starting from $200 per month is available for purchase.

***MS Azure – MySQL Server*:** Microsoft provide a limited free version per year. Most of the services will be provided for a fair user policy. But there is a limit in transactions/services usage per month. If that amount is exceeded, a payment is required for the additional services utilized. Roughly $20 will be required monthly.

Twilio APIs: A online tool which provides users to send SMS, MMS, WhatsApp and Viber messages by utilizing their APIs.

3.4.2 Feature Development

This section will discuss the process of the development of different operations in the proposed model. A collection of different features is included in the proposed model. The development process if these features will be discussed further in separate sub sections.

For the operations executed from the client (a Java-based client), Stored Procedures and Functions (MySQL Server) are created in the server side. These will be called upon when related actions are executed by the user.

The main features of the proposed system are as follows:

1. Create a mapping between new tasks and previously crowdsourced task solutions.
2. Notifications based on worker's profile.
3. Task Scheduling for repetitive tasks.
4. Worker Rating based on the solutions which are provided.

3.4.2.1 Previously crowdsourced task mapping

When the user enters the task details, the system will extract keyword and entity information from the task. This is a client process that uses the java package TextRazor. The TextRazor package is capable of extracting entities and keywords in a sentence.

Once extracted, the system will try to create a mapping between the new task's entities and keywords and already available tasks and solutions. This process will query and process thousands of available tasks and solutions from the database and run a mapping based on the extracted entity and keywords information. The end result would be a set of solutions which has a positive mapping against the new task.

The results will be shown to the requester, and the requester can either accept one or more available solution or opted to go for a crowdsourcing operation.

3.4.2.2 Notifications based on worker's profile

For the notification handling, the entities and keywords extracted using the TextRazor package will be utilized along with additional information such as the task category and labels the requester provide during the task creation.

At the registration process of the workers, information such as worker's expertise and interest areas were recorded. These are set of predefined domains such as Computer Science, Information Systems, Information Technology, Mathematics, Algorithms, Music, Food, Market Analysis, etc. So, the entities or keywords that were extracted using the task information have a matching result with the worker's interested or expertise areas, notifications will be created and sent out to the users.

Notifications can be the form of email, text messages or WhatsApp messages. Creation of emails can be done by utilizing Javax packages related to emails. Online APIs such as Twilio enables to send SMS and WhatsApp messages by utilizing their APIs.

These technologies have been used in order to achieve the notification handling.

3.4.2.3 Task Scheduling

Requesters can specify whether a task need to be creating a scheduled recurring job or not. For example, the requester can specify that a particular task needs to be repetitively posted in the platform every 2 days.

This information will be utilized by the system to create a scheduled job, so that the same task is automatically posted in the platform according to the date specified.

MySQL provides EVENTS operations to do these kinds of operations such as scheduling and repetitive scheduling. During the implementation, MySQL EVENTS was used to develop the scheduled task posting operations.

3.4.2.4 Worker Rating Mechanism

Rating figures are calculated based on the feedback given by the requesters for the solutions they receive by the workers. As mentioned previously, these rating figures will be utilized to group and order the task solutions to give the requesters a clear and efficient view of the set of the solutions.

The rating is dependent on the worker and the domain area. As mentioned in the previous sections, domain areas can be Computer Science, Algorithms, Medicine, etc. So, rating figure will be maintained and continuously calculated in terms of the worker and the domain area.

For example, the worker John will have a rating 3.5 for Music and 4. For Algorithms. For the calculation mechanism, PLSQL functions were created in the sever level.

3.5 Evaluation Plan

This section discusses the evaluation plan for the proposed model. Section will be subdivided as research questions, evaluation approach and the evaluation of the results.

3.5.1 Research Questions

There are three research questions addressed during when proposing the new model.

1. *Does maintaining a worker rating system and using rating figures to prioritized task solutions would make sure high-quality answers accessed quickly?*

   The idea behind this is, the system will maintain records related to worker's solution ratings. That is, when a worker submits a task solution, the requester can rate the solution from 1 to 5.

That rating information will be stored in the database in the form of 'rating per worker per domain'. Usage of this operation is, when the workers submit answers, the system will use the rating figure to prioritized/order the way solutions are shown. For example, solutions given by a worker with rating 4.5 will be given priority over a solution given by a worker with rating 3.5. this process makes sure high quality solutions are shown first for the requester.

2. *Does targeting workers in a Crowdsourcing platform by their interests, expertise, and geographical location, and send similar tasks to them would result in a small solution pool, thus the cost in getting a solution reduced?*

   The main factor behind the question is, when a worker does a task that is more comfortable for him/her, there is a greater possibility that the results will be correct. For example, if Worker Alex's domain knowledge is in computer networking, there is a good probability that a task in the computer networking domain could be answered by Alex in a more efficient, correct and speedy manner. Thus, once a new task is posted on the platform, the tasks could be evaluated and send a notification for the users based on their interests, expertise and geographical location would make the crowdsourcing process much efficient.

3. *Do tasks that require solutions on a daily/weekly basis, could be set into a schedule and automatically published in the platform at the correct time speed up the task resolution process?*

   Certain tasks require continuous solutions. Because of this, requesters would require entering the same task daily on the platform. For example, what is the predicted weather in the Galle area today? Though the weather report is available, it is much reliable for a person who lives in that area, Galle, answers it with his/her experience. Suppose this requester is a tourist guide who does tours in the Galle area, and he needs to know the weather every day. But the problem, in this case, is, once he posts this task on the platform, he won't get solutions within seconds, he needs to wait till a worker does it. So, if he can schedule the task to be posted on the platform daily at a given time, he could get the results once he logs into the system after the scheduled time, thus saving his time of reposting the same tasks repeatedly.

3.5.2 Evaluation Approach

This project proposes a new method to process and evaluate tasks in a Crowdsourcing platform. The new concept will address a common problem in Crowdsourcing platforms. So, the given solution must evaluate with the respective audience. The most effective method of evaluation is performing a live experiment with a set of individuals and exposing them to the system to perform a crowdsourcing operation. But practical limitations, such as time and resources restrict a full experimental evaluation.

As a result, a mixed evaluation approach of both experimental based and Opinion Based evaluation will be considered.

3.5.2.1 Experimental Based Evaluation

All three main questions could be, for a certain extend be evaluated using an experimental based method with help from several individuals. The crowdsourcing domain is not easy to be evaluated since it requires a huge number of participants. But the most effective way to evaluate is to use the experimental methodology.

The experiment should be executed in two systems. One system is with the solution proposed by this project applied, and the other system is without the solution. A group of 10-15 individuals will have to take part in this experiment. Since the population for the experiment is low, a given individual will have to play both the worker role and the requester role.

i. Dataset

The dataset, an example set of tasks used for experimentation is a live dataset that is used in an actual crowdsourcing platform. For this purpose, live tasks posted in Amazon Mechanical Turk (mTurk - a marketplace that makes it easier for individuals and businesses to outsource their processes and jobs to a distributed workforce who can perform these tasks virtually) are used.

More than 100 actual tasks were extracted,, and a local database was built. As these tasks are actual tasks that are used in crowdsourcing platforms, the desired goal can be verified using this, which is the performance gain in crowdsourcing platforms.

This data is open for any individual and can be freely obtained. Each task is added by requester which contains the following.

a. Requester Name (mandatory information)
b. Task Title (mandatory information)
c. Task Description – A overview of the task. (mandatory information)

d. Task Body/Information – (mandatory information) All the information related to the task. What and how to proceed with the task with every clue available. This is the most important part of the task since it contains every bit of information related to the task.
e. This may be a dataset, paragraph, link to a webpage, etc.
f. Time Allocation – The time which the task required. (non-mandatory information)
g. Expiration – When the task will be expired. No solutions will be accepted after that date. (non-mandatory information)
h. Reward – The incentive paid for the accepted answers. (non-mandatory information)

As mentioned previously, the above dataset will be manually imported to the local database thus creating a fully-grown local database with actual tasks. The other important set of data is the solutions for the tasks. Unfortunately, the mTurk does not allow users to investigate the solutions given for a certain task. As a result, no actual set of solutions could be extracted. But, a dummy set of solutions could be manually imported to the database. One evaluation point of the system is to determine whether the requester is satisfied with a previous solution given for a previous task.

Apart from the tasks, user information should be stored in the database. For the requesters, simple personal information would be enough. But for the workers, a detailed set of information is required. Apart from the worker's basic personnel information, the region, the city he lives in, contact information such as the mobile number and email, expertise or interested domain areas, for example, IT, Music, etc. should be stored in the database. This information will be vital when processing tasks.

ii. Question Evaluation

In this section, it will be discussed how each Question would be tested during the experiment. The participants should have a general idea on how crowdsourcing work to attempt the

Question 01 – *Does maintaining a worker rating system and using rating figures to prioritized task solutions would make sure high-quality answers accessed quickly?*

In Question 01, both the worker’s and requester’s activities are evaluated. Mainly the requester’s interference with the system will be focussed.

Prerequisite: Rating information for multiple workers should exists in relevant task domains.

In the existing platforms, task solutions will be ordered or prioritized by the time of submission. Also, it can be manually ordered or prioritized by the worker’s name, time took by the worker to solve, etc.

The proposed solution will use the ratings per worker per domain to order and prioritize the task solutions.

Here, time taken by the requester to find the best solution for the task will be recorded for both the existing and the proposed system. If the time taken using the proposed system is less than that of the system without the proposed solution, it can be concluded that the prioritizing the task solution based on the rating does have a positive influence with respect to the time factor.

Further, as another outcome would be a decrease in cost. That is, the requester do not need to spend unnecessary cost on workers with low quality solutions because the requester is able to find a quality solution in a less time.

Question 02 – *Does targeting workers in a Crowdsourcing platform by their interests, expertise and geographical location, and send similar tasks to them would result in getting a solution for the tasks in a much more efficient manner?*

In Question 02, time refers to the time taken since the requester submits a task until he/she get a valid solution. Here, mainly the worker is focused. In general, how quickly a valid solution is accepted is considered.

For this Question to be evaluated, input from the evaluator/requester is required. Upon that, the system would evaluate the task given.

The task entered by the requester will be evaluated, taking in to account different keywords. This process to extract any countries, regions or cities if they are included in the task. Further, this process would extract the domain of the task if available. For example, music, information technology, market survey, algorithms are some of the domain areas.

Depending on the extracted information, task’s domain area, geographical information, alerts will be sent out for the workers who have similar interests, expertise or who love in the same geographical location as in the tasks.

The thinking of the process is that, if a worker comes across a task that is comfortable for him/her, the probability that the worker successfully completes the task is prominent. Another side is since the alerts are sent out workers come to know that a task is posted in the platform which he would interest in. So, the worker would most probably visit the platform at that point to answer the task without waiting. Thus, resulting in quick flow of solutions for the tasks.

The average time taken for similar tasks to be processed in both the systems with and without the proposed solution implemented will be considered. If the time taken by the system with the proposed solution is implemented is lesser than that of the other system, we can decide that the proposed solution is a success.

Question 03 – *Does tasks that require solutions on a daily/weekly basis, could be set into a schedule and automatically published in the platform at the correct time speed up the task resolution process?*

This Question is quite difficult to be tested experimentally. It is because, the same task should be solved by the workers in each time interval. Participants will not be enthusiastic on doing such process continuously.

In Question 03, time refers to the time taken since the requester submits a task until he/she get a valid solution.

For this Question to be evaluated, input from the evaluator/requester is required. Upon that, the system would evaluate the task given.

There is a separate setting for the requester to fulfil. During the task insertion process, he/she needs to set up the batch schedule mechanism. Here, the requester will have input information such as the frequency of the tasks which have to be posted as a batch, for example, daily, weekly, monthly or a customized manner.

During the evaluation, since it is time-bound, a custom time interval will be introduced to schedule the task in minutes. So, each minute, the same task will be posted repeatedly.

The evaluation criteria are the average time taken by the system to get a solution for the requester each time a task is posted.

All the above three questions testing will take place with all the solutions are working together. That is, when Question 3 is being tested, it will be affected by the solutions given for questions 1 and 2.

3.5.2.2 Evaluation based on Questionnaire

This section will discuss how the questionnaire based evaluation approach was designed to gather information about the proposed model.

To get an in-depth evaluation based on the proposed model, set of questions are designed to get feedback from group of individuals who have a knowledge on crowdsourcing. The target population was the individuals in the areas of software development, academic, marketing and various business.

The questions are defined to be in a specific format, that is the one who takes the survey can provide their answers or thought based n the rate of the agreement. The questionnaire will contain five possible answers:

a. Strongly Disagree
b. Disagree
c. Neutral
d. Agree
e. Strongly Agree

Question Evaluation

In this section, it will be discussed how each Question would be evaluated during questionnaires using sub questions. The questionnaires will address the proposed solution functionalities that was proposed to solve each of the questions. Some of the key questionnaires will be given as examples in this section.

Question 01 – *Does maintaining a worker rating system and using rating figures to prioritized task solutions would make sure high-quality answers accessed quickly?*

Following are the questionnaires based on the Question.

1. Maintaining a worker rating system is a good approach. (workers will be rated from 1 to 5. For example: Worker ‘Lal’ has a 4.5 rating for the domain Computer Science).
2. Using the rating mechanism to prioritize task solutions will enhance the probability of getting a high-quality task solution quickly.
3. Grouping/ordering task solutions using the workers rating per the tasks domain will increase the quality of the final approved task solution.
4. Grouping/ordering task solutions using the workers rating per the tasks domain will reduce the time taken to achieve a final approved task solution.
5. Grouping/ordering task solutions using the workers rating per the tasks domain will reduce the cost taken to achieve a final approved task solution.

Question 02 – *Does targeting workers in a Crowdsourcing platform by their interests, expertise and geographical location, and send similar tasks to them would result in getting a solution for the tasks in a much more efficient manner?*

Following are the questionnaires based on the Question.

1. Storing a worker profile with information related to workers is important because they can be used to categorize the workers.
2. Sending notification to the workers based on the task's domain information will enhance the quality of the solution.
3. Sending notification to the workers based on the task's domain reduce the time taken to receive a solution.
4. Sending notification to the workers based on the task's domain information reduce the cost of receiving a solution.

Question 03 – *Does tasks that require solutions on a daily/weekly basis, could be set into a schedule and automatically published in the platform at the correct time speed up the task resolution process?*

Similar to the questionnaire for Question 1 and 2, Question 3 will also have a questionnaire which corresponds to the Question evaluation.

1. Scheduling recurring tasks enhance the quality.
2. Scheduling recurring tasks reduce the cost of task processing.
3. Scheduling recurring tasks reduce the latency of task processing.

3.5.3 Evaluation of results

Results will be presented and evaluated in detail in the next chapter.

3.6 Chapter Summary

This chapter discussed the general project management methodologies followed to solve the task. Analysis and design of a solution, dataset management, implementation methods and the evaluation plan were discussed in detail.

The chapter indicates how the design and development methodologies worked together to achieve the said goals and objectives by enhancing the crowdsourcing operations in terms of quality, cost and latency without having any major trade-off in between. In the next chapter, the results of the evaluation will be studied to see how fit the proposed solution is.

CHAPTER FOUR

EVALUATION

This project proposes and develops a new framework for a crowdsourcing platform to enhance its efficiency with respect to quality, latency, and cost. Implementation and evaluation are a very important subset of a project. Chapter 3 subsection 3.4 discussed the implantation of the proposed solution and subsection 3.5 discussed the evaluation plan on how the proposed system will be evaluated.

This chapter will analyse the evaluation results. Here, the results gathered through both Experimental and Questionnaire based evaluation will be analysed in depth to find out the key strengths and the areas for improvement of the proposed model. As a key base point, the results will be analysed in terms of the three control factors; Quality, Cost and Latency.

4.1 Analysis: Experimental Based Evaluation Results

This section discusses the experimental evaluation process results on the three main research questions. Mainly, the individuals were asked the execute a crowdsourcing flow in the proposed system and compared it to the ordinary system without the proposed model implemented as mentioned in the evaluation plan section of the Methodology. Their responses were recorded and mapped against each question and the control factors.

4.1.1 Question One

Does maintaining a worker rating system and using rating figures to prioritized task solutions would make high-quality answers accessed quickly?

The general idea of the implementation related for the research question was to order and categorize the solutions given by workers using the worker rating factor for the task domain. Visible outcome was, the solutions given by workers with high rating factor will be given priority and shown on top of the solution set.

In terms of quality, the overall feedback was, there was a good chance that good results were on the top of the solution section. According to the feedback, it was mentioned that it was easy for the requesters to identify good solutions on top of the section and if they are satisfied, they can omit going through other solutions unnecessarily.

In terms of latency, it was clearly visible from the feedback that the latency for the crowdsourcing was clearly improved. As mentioned above, quality answers were identified on top of the solution list. Therefore, according to some feedback the requesters do not need to go through all the solutions to get a good solution. Thus, the time taken to identify a acceptable solution is much less. As a results, the latency factor is improved.

In terms of cost, there was no direct feedback for this question. But it can be deduced that the cost for crowdsourcing is reduced because of prioritizing the solution list, requesters were able to find good results without going through all the solutions, thus a payment was not required for all the solutions.

When analysing all the feedback related to the three control types, quality, cost and latency, it is clearly seen that the quality and latency are having a direct improvement with the implementation of a worker rating system while the cost has an indirect improvement.

4.1.2 Question Two

Does targeting workers in a Crowdsourcing platform by their interests, expertise, and geographical location, and send similar tasks to them would result in getting a solution for the tasks in a more efficient manner?

The general idea of the implementation related for the research question was to target a subset of workers in the worker pool of the crowdsourcing platform using different aspects of the workers such as their interests, expertise and geographical location. Once the subset of workers is identified, a notification will be sent to them stating that a task that they might be interested is now available.

In terms of quality, there was many feedbacks stating that the quality was optimized using this method. The generalized comment of the feedback was, since most of the solutions available are solved by individuals with a probable expertise in the task domain, the solutions were good in quality. Also, there was feedback stating that, since the number of solutions were fairly low because of choosing a subset of workers, it was easy for the requesters to identify the best results. But there were certain feedback stating that, because of the subset of workers, some results from other reliable workers will be missed.

In terms of latency, the feedback was very much towards positive improving the time factor. All the feedback states that the targeting a given subset of workers by sending notifications to them definitely improved the latency. Users who acted a worker stated that, it was easy for them to start working on new tasks if they get a prior notification. Also, feedback was available, it was easy and time was saved because the tasks were related to their domain

In terms of cost, feedback from the requesters was positive. Most of the feedback states that since unnecessary solutions were not available, the requesters were not forced to pay any unnecessary fee for the workers, thus the cost was decreased.

When analysing all the feedback related to the three control types, quality, cost and latency, it is very clearly seen that the implementation as solution for the research question has the direct positive impact on quality, latency and cost.

4.1.3 Question Three

Do tasks that require solutions on a daily/weekly basis, could be set into a schedule and automatically published in the platform at the correct time speed up the task resolution process?

The general idea of the implementation related to the research question was to identify tasks which the requester wants to post in the crowdsourcing platform on a daily/weekly basis and post them automatically without the intervention of the requester.

In terms of quality, there no much feedback from the users who used the solution. But there were certain user feedback stating that, there was a possibility that the same worker would assign the task and provide solution, thus the quality of the solution will remain same without much variation.

In terms of latency, there were many feedbacks stating an improvement of the whole crowdsourcing process. The general feedback was, the time spent by the requester was not taken place, the overall time as reduced. Further since the tasks were posted consistently in a given time, the workers who have an idea of the recurring nature of the task could accept it easily every time. Thus, the time taken for the task to be received by a worker is reduced.

In terms of cost, there was no such feedback form the users. But using the feedback in terms of quality and latency, a deduction ca be made that the cost is also improved because the unnecessary involvement of the requester with the platform can be omitted.

4.2 Analysis: Questionnaire Based Evaluation

As mentioned in the evaluation plan section, to get an in-depth evaluation based on the proposed model, set of questions are designed to get feedback from group of individuals who have a knowledge on crowdsourcing. Mainly, the participants of the survey were asked their thoughts of the proposed models using a set of questions based on the three main research questions, in terms of Quality, Latency and Cost. There responses were recorded and mapped with each control factors.

According to the survey results, a detailed summary of the most critical questions is as below.

4.2.1 Question One

Maintaining a worker rating system and using rating figures to prioritized task solutions would make high-quality answers accessed quickly.

Sub Question 1: Maintaining a worker rating system is a good approach. (workers will be rated from 1 to 5. For example, Worker 'Lal' has a 4.5 rating for the domain Computer Science)

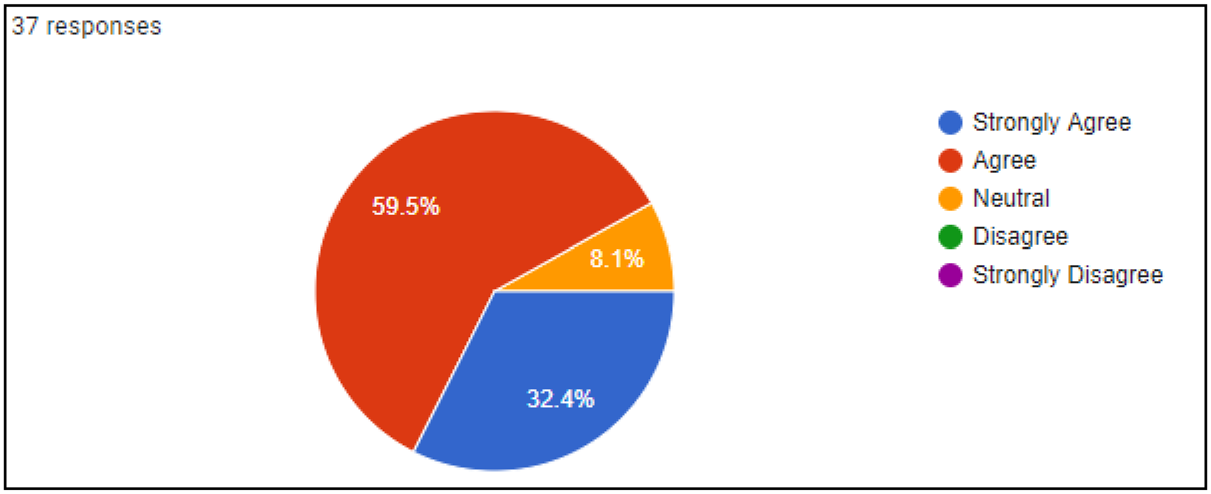


Figure 4.1: Results of Sub Question 1 of evaluating Research Question 1

Majority feedback, 91.9% (Strongly Agree and Agree), from the survey indicates that many of the individuals who participated in the survey agreed on the fact that maintaining a worker rating system is a good approach.

Sub Question 2: Using the rating mechanism to prioritize task solutions will enhance the probability of getting a **high-quality task solution** quickly.

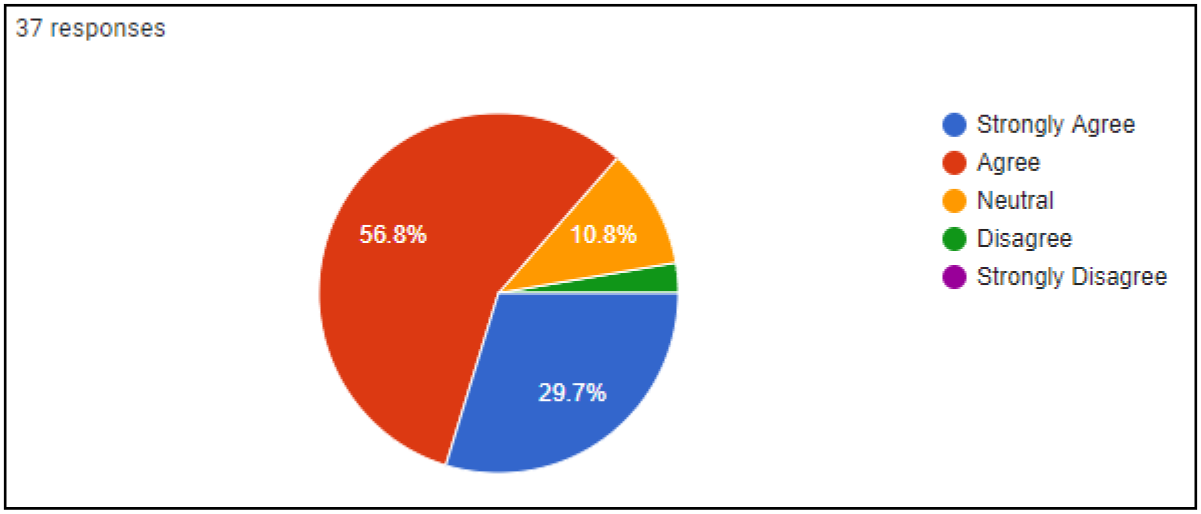


Figure 4.2: Results of Sub Question 2 of evaluating Research Question 1

A majority portion of 86.5% (Strongly Agree and Agree), agreed on the fact that high-quality solutions could have arrived if prioritizing task solutions is based on worker rating. But a very small percentage (2.7%) disagree with that.

Sub Question 3: Grouping/ordering task solutions using the workers' rating per the tasks domain will **reduce the time taken** to achieve a final approved task solution.

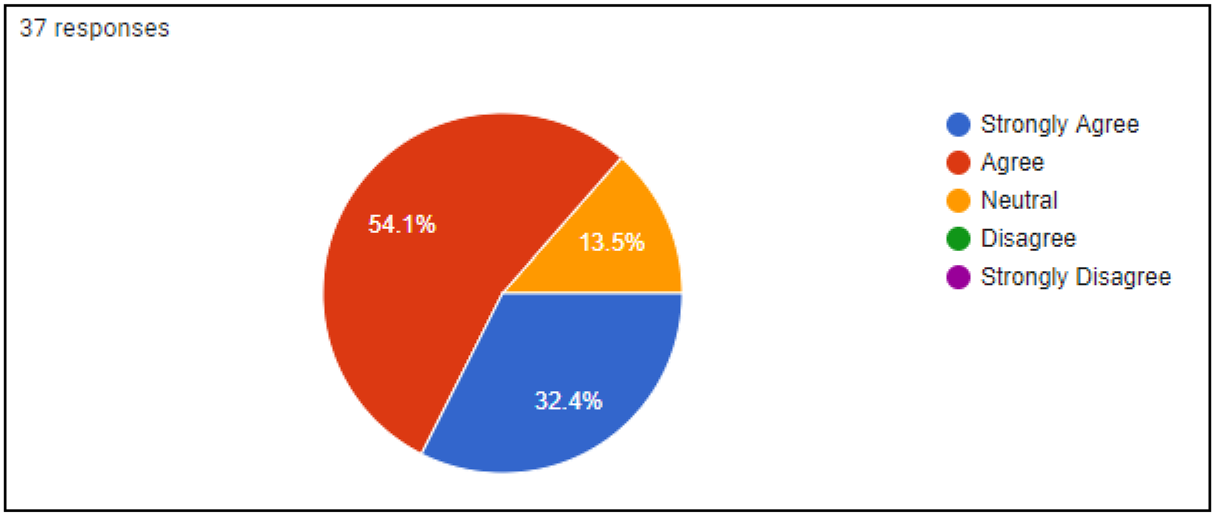


Figure 4.3: Results of Sub Question 3 of evaluating Research Question 1

A majority of the population, 86.5% (Strongly Agree and Agree), agreed that grouping/ordering tasks solutions using a worker rating system will reduce the time taken for a final approach task solution.

Sub Question 4: Grouping/ordering task solutions using the workers' rating per the task's domain will reduce **the cost taken** to achieve a final approved task solution.

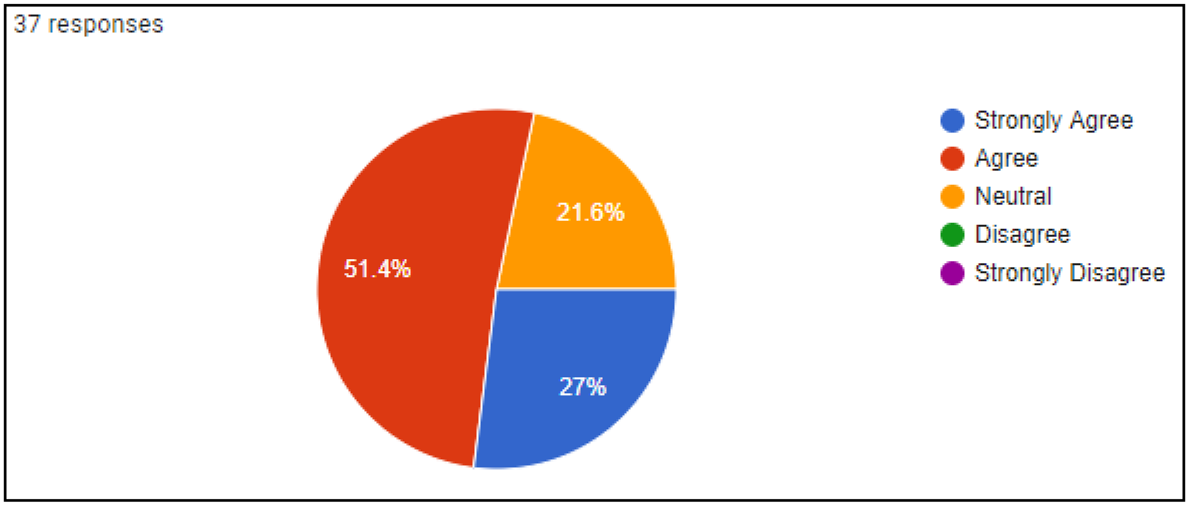


Figure 4.4: Results of Sub Question 4 of evaluating Research Question 1

A percentage of 78.4% (Strongly Agree and Agree), agrees that cost can be reduced when grouping/ordering task solutions based on worker rating.

4.2.2 Question Two

Targeting workers in a Crowdsourcing platform by their interests, expertise, and geographical location, and send similar tasks to them would result in getting a solution for the tasks in a much more efficient manner.

Sub Question 1: Storing a worker profile with information related to workers is important (for example, location, contact information, interested areas, expert areas)

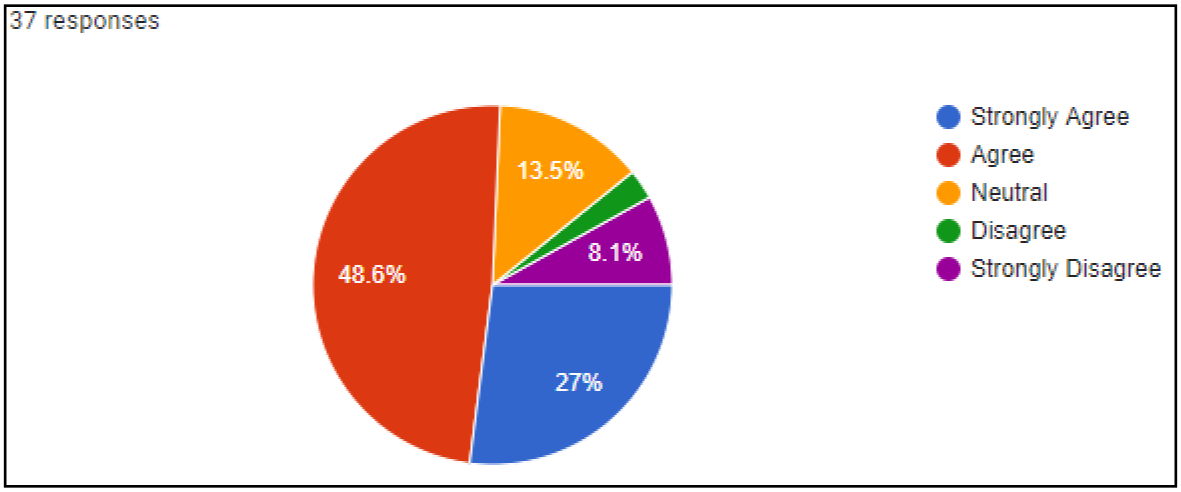


Figure 4.5: Results of Sub Question 1 of evaluating Research Question 2

The majority of the survey participants believe that worker profile information maintenance is important in crowdsourcing. But, 8.1% of strongly disagree on the fact.

Sub Question 2: Sending notification to the workers based on the task's domain information (Computer Science, Music, Medicine, etc.) will **enhance the quality** of the solution. (for example, notifications will be sent to the workers who are interested or have an expertise in that area).

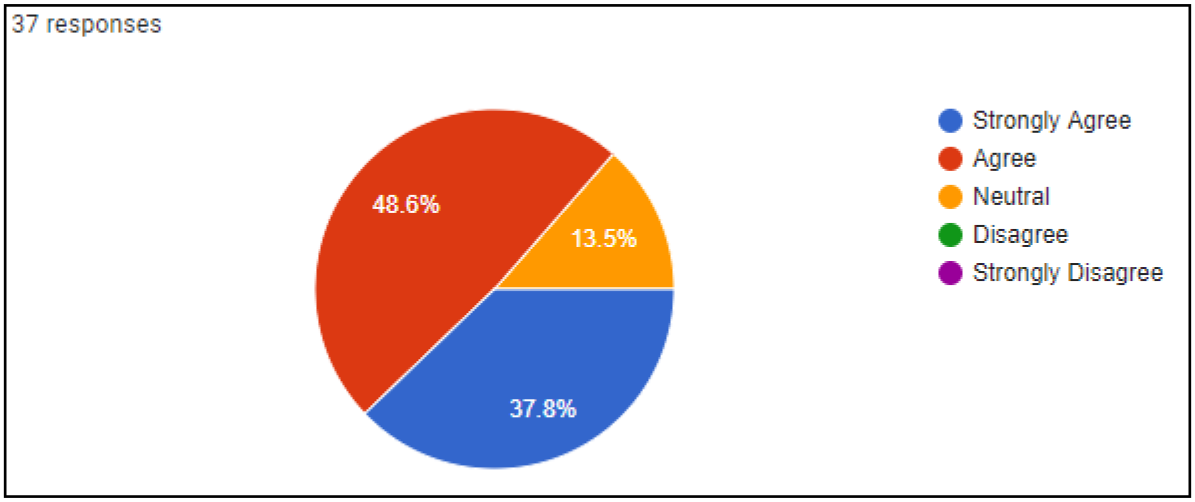


Figure 4.6: Results of Sub Question 2 of evaluating Research Question 2

A majority of the feedback, 86.4% (Strongly Agree and Agree), agreed the quality solutions could have arrived if notifications are sent to the workers based on tasks' domain information.

Sub Question 3: Sending notification to the workers based on the **task's location information** will **reduce the time** taken to receive a solution. (for example, if the task contains a city/country information, notifications to the workers who are connected to those cities will be sent).

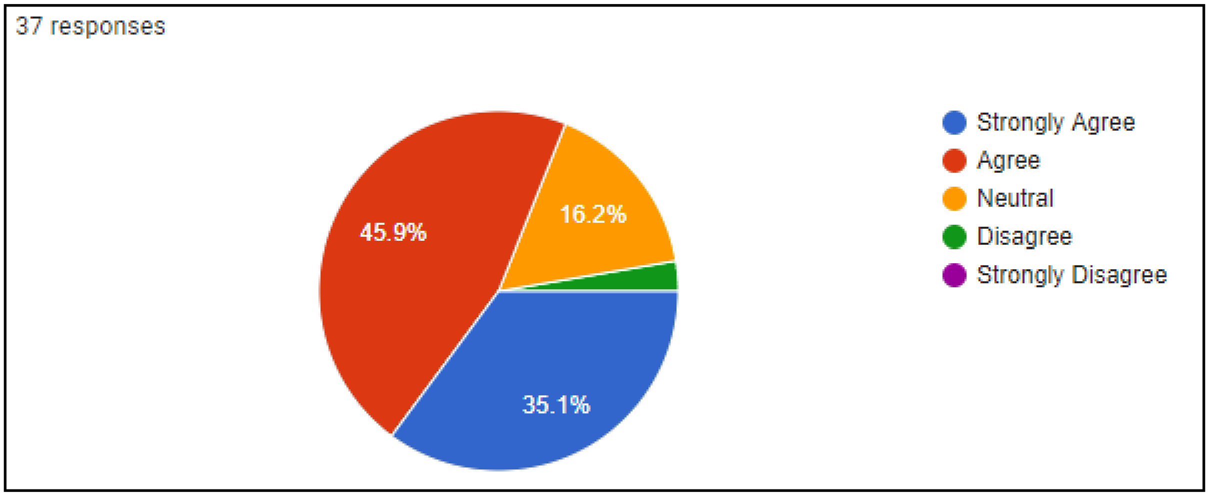


Figure 4.7: Results of Sub Question 3 of evaluating Research Question 2

A percentage of 81% (Strongly Agree and Agree), agree that time taken to arrive a final solution could be reduced by sending notifications to the workers based on the task's location information.

Sub Question 4: Sending notification to the workers based on the **task's domain information** (Computer Science, Music, Medicine, etc.) **reduces the time** taken to receive a solution. (for example, notifications will be sent to the workers who are interested or have an expertise in that area).

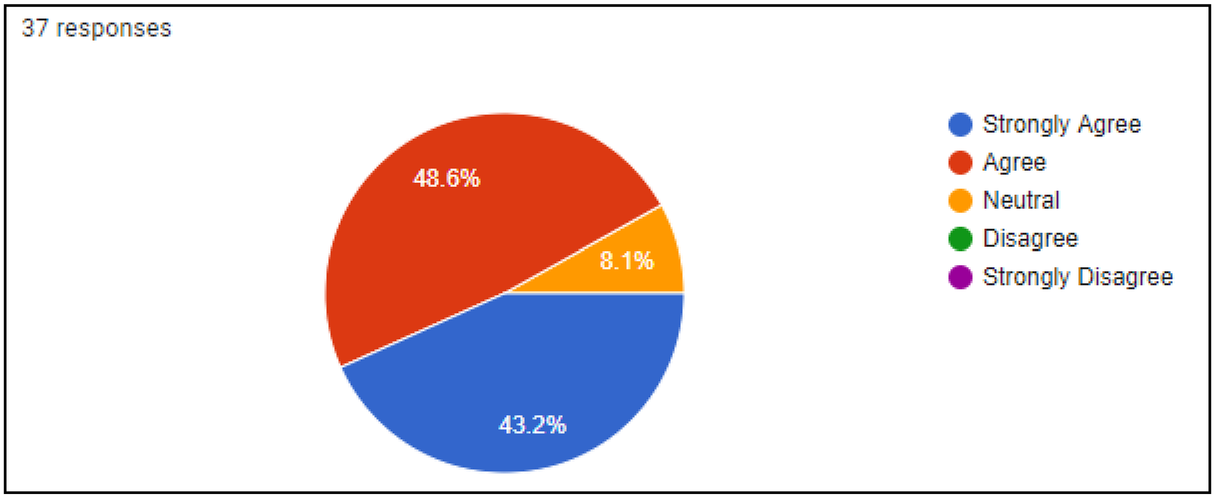


Figure 4.8: Results of Sub Question 4 of evaluating Research Question 2

A percentage of 91.8% (Strongly Agree and Agree) of the survey participants agree that the time taken to come up with a final solution could be reduced if workers are notified based on task domain information.

Sub Question 5: Sending notification to the workers based on the **task's location information will reduce the cost** of receiving a solution. (for example, if the task contains a city/country information, notifications to the workers who are connected to those cities will be sent).

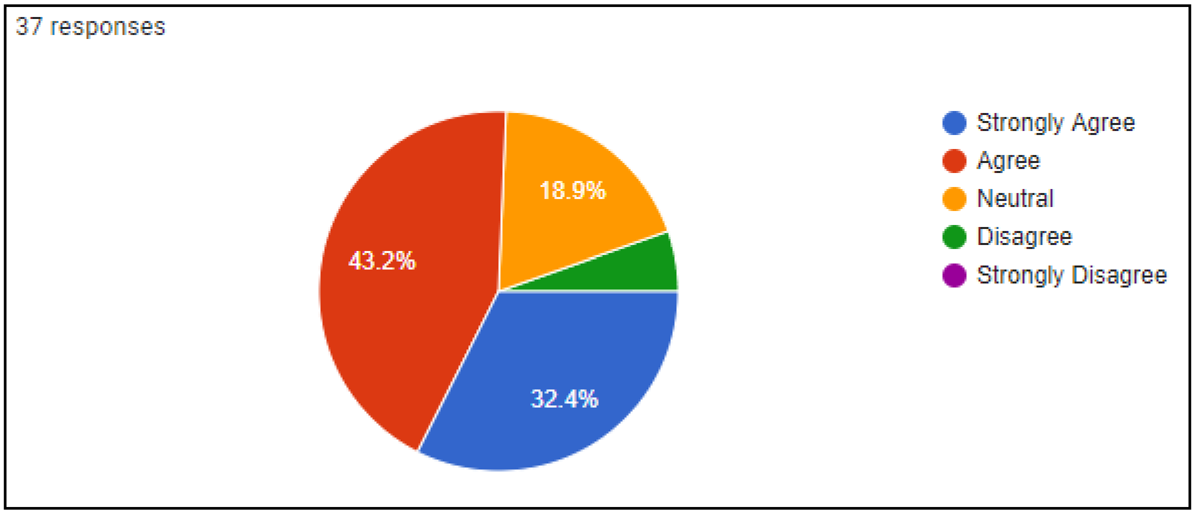


Figure 4.9: Results of Sub Question 5 of evaluating Research Question 2

Most of the feedback, 75.6% (Strongly Agree and Agree), indicate that using tasks' location-related information to send notifications to the workers will reduce the cost of receiving a task solution.

Sub Question 6: Sending notification to the workers based on the **task's domain information** (Computer Science, Music, Medicine, etc.) **reduce the cost** of receiving a solution. (for example, notifications will be sent to the workers who are interested or have an expertise in that area).

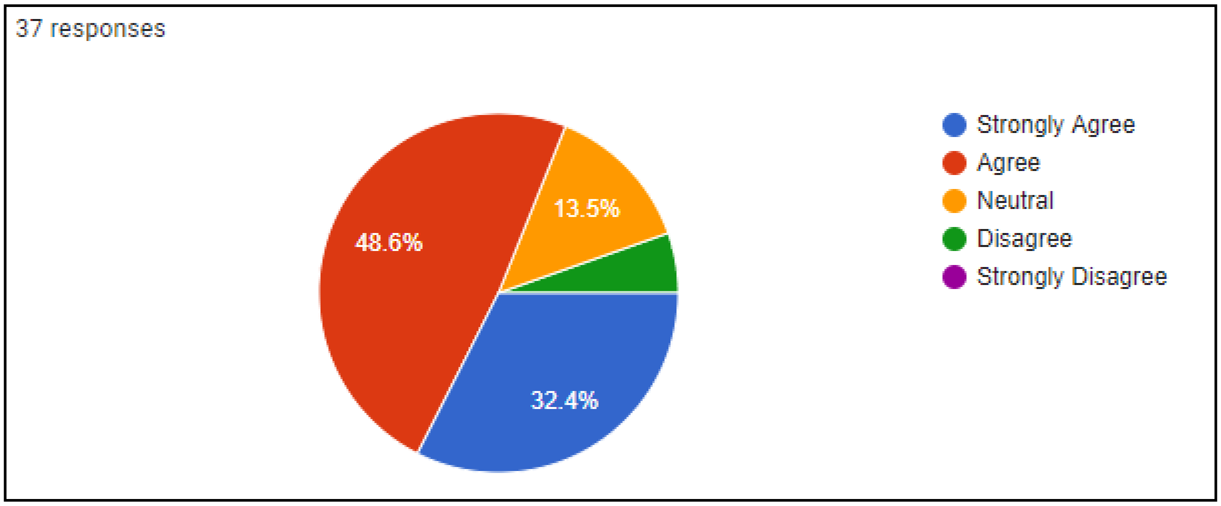


Figure 4.10: Results of Sub Question 6 of evaluating Research Question 2

A majority of 82% (Strongly Agree and Agree), agreed that the cost can be reduced if notifications are sent to the workers based on the task's domain information.

4.2.3 Question Three

Do tasks that require solutions on a daily/weekly basis, could be set into a schedule and automatically published in the platform at the correct time speed up the task resolution process?

Sub Question 1: Scheduling recurring tasks could **enhance quality**. (for example, if the same task must be posted in the platform weekly, the system will automatically handle it)

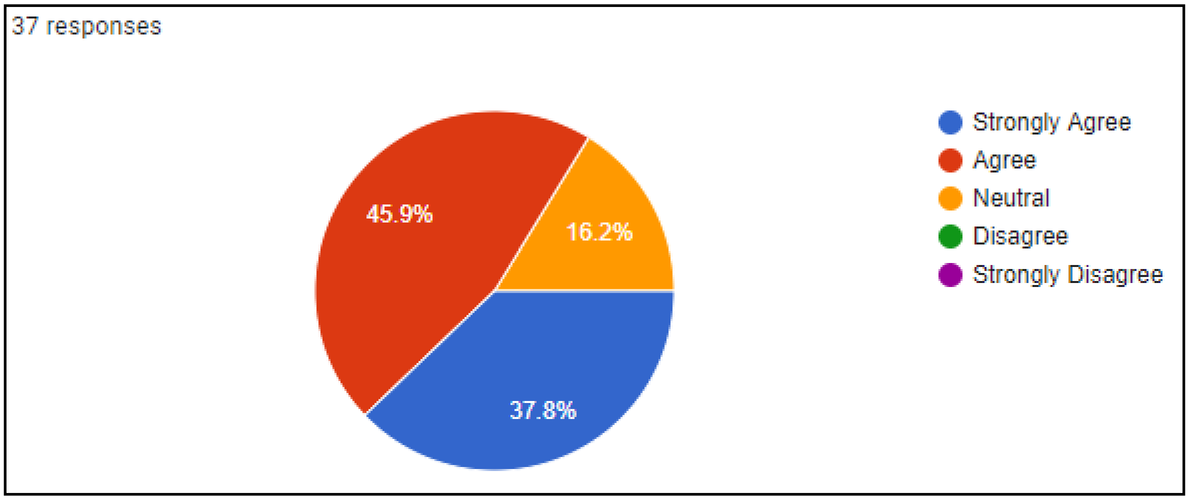


Figure 4.11: Results of Sub Question 1 of evaluating Research Question 3

Most of the feedback, 83.7% (Strongly Agree and Agree), agrees that the quality could be enhanced if recurring tasks are scheduled.

Sub Question 2: Scheduling recurring tasks **reduce the time** taken to receive a solution.

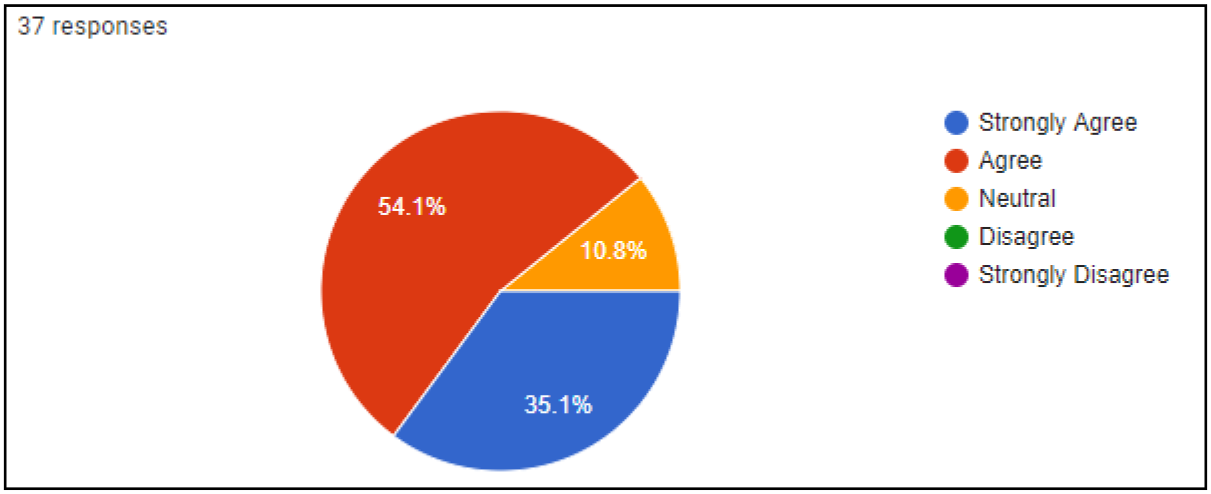


Figure 4.12: Results of Sub Question 2 of evaluating Research Question 3

89.2% of the participants agree that the rime could be reduced to receive a solution if a scheduling mechanism is available for recurring tasks.

Sub Question 3: Scheduling recurring tasks will **reduce the cost** of receiving a solution.

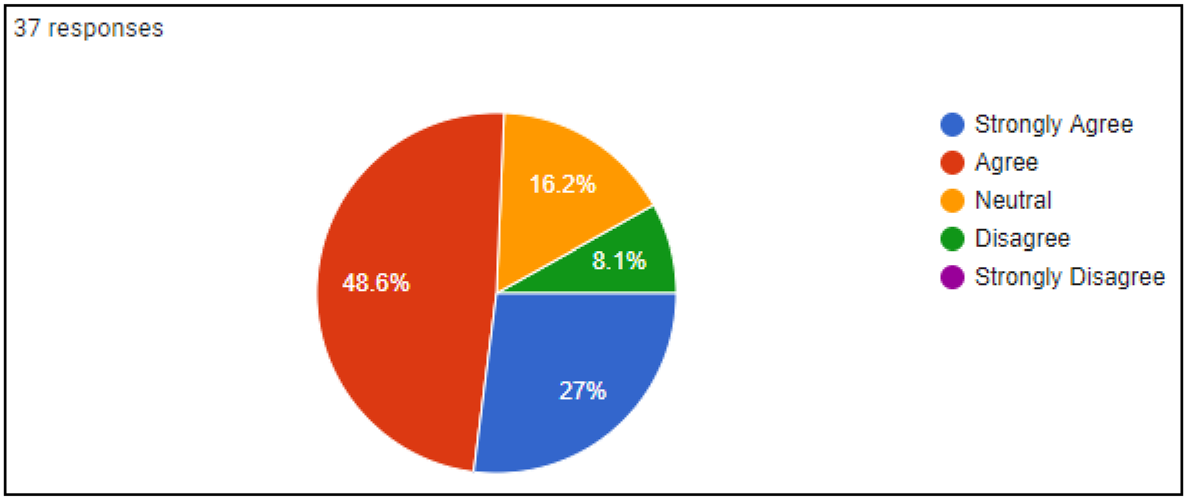


Figure 4.13: Results of Sub Question 3 of evaluating Research Question 3

Another majority of 75.6% (Strongly Agree and Agree), agree that a cost reduction can be achieved when a scheduling mechanism is used for recurring tasks.

4.3 Limitations of the Evaluations

When performing the evaluation, there were certain limitations which affected the results obtained. This section will discuss these limitations briefly.

4.3.1 Limitations in Experimental Based Evaluation

To test and try-out the proposed system, access was given to a set of individuals. Most of them were either Software Engineers or in the field of marketing. During the experimental evaluation, some limitation was visible. They may have an impact on the final results of the feedback but the limitations do not change the general feedback given by all the users.

1. Issues with the type of the worker:
   The variety of the workers were limited, and the knowledge expertise was limited. Thus, most of the tasks were in the same domain. Few of the domains were, Information System, Computer Science, Algorithm, Market Fluctuation, Currency Variations. Since the limitation of workers in numbers and in knowledge, of the problems which may arise was not visible.
2. Sample solutions were given:
   Some of the workers were giving sample solutions using different webpages. Because of that, the credibility of the solutions was problematic.
3. Workers were always online:
   Since the workers were always online during the evaluation, all the tasks were visible by them even though a notification came or not.

4.3.2 Limitations in Questionnaire Based Evaluation

As a common problem for questionnaires, noisy and contradictory answers were available. So, during the analysis such answers were omitted. Compared to the experimental based evaluation, the questionnaire based evaluation was carried out much more efficiently because the participants were much more relaxed to share the feedback.

4.4 Summary of the results

The summary of the results can be divided into two sections, summary for the results of experimental based and questionnaire based.

Experimental Based Results: The general feedback from the users of the proposed system was that the proposed system is cable of delivering a good solution to the optimization of the crowdsourcing processes in terms of quality, latency and cost. When analysed further, it is identified that the the trade-off between the control types were minimized.

Questionnaire Based Results: If each question was aggregated, more than 80% of the survey participants agreed on the fact that the crowdsourcing operations could be optimized in terms of quality, latency and cost.

However, some of these results are subjected to some limitations a mentioned in section 4.3.2.

Accordingly, as a combined result, the proposed system's evaluation states that the crowdsourcing operations would be enhanced in terms of the control factors, quality, latency and cost.

CHAPTER FIVE

RESULTS

5.1 Chapter Overview

This chapter will discuss the results of the evaluation and its outcome. Results overview was presented in the previous chapter by selecting few of the main questions raised during the questionnaire process, and this chapter will further analyse all the questionnaires in connection to each optimization component. Analysis is based on mainly on questionnaire evaluations.

5.2 Questionnaire Based Results Analysis

5.2.1 Quality Factor

Different questions related to the implementation associated with quality enhancements were asked by the participants, and their honest responses were recorded. All the questions are based on how extend they Agree on the enhancements.

Summarized results are as follows.

Category	Responses	Percentage of Responses
Strongly Agree	49	33.1%
Agree	77	52.02%
Neural	21	14.18%
Disagree	1	0.7%
Strongly Disagree	0	0%

Table 5.1: Reponses related to Quality.

Based on the results, the majority agrees on the fact that the quality will be enhanced with the proposed and implemented techniques. The nature of the questions raised, as mentioned in the previous section, clearly highlight the importance of the proposed and implemented solution to enhance the Quality of crowdsourcing.

According to the results obtained, 126 responses that is 85.12% agree on the fact that the proposed and implemented solution will enhance the quality and will be an important improvement for the crowdsourcing operations.

5.2.2 Cost Factor

Different questions related to the implementation associated with cost related enhancements were asked by the participants, and their honest responses were recorded. All the questions are based on how extend they Agree on the enhancements.

Summarized results are as follows.

Category	Responses	Percentage of Responses
Strongly Agree	50	27.02%
Agree	86	46.48%
Neural	42	22.7%
Disagree	7	3.78%
Strongly Disagree	0	0

Table 5.2: Reponses related to Cost.

Based on the results, the majority agrees on the fact that the cost will be enhanced with the proposed and implemented techniques. The nature of the questions raised, as mentioned in the previous section, clearly highlight the importance of the proposed and implemented solution to reduce the Cost of crowdsourcing.

According to the results, a higher number of responses indicates that that they agree on the fact that the proposed and implemented solution would reduce the cost of crowdsourcing.

5.2.3 Latency Factor

Different questions related to the implementation associated with Latency related enhancements were asked by the participants, and their honest responses were recorded. All the questions are based on how extend they Agree on the enhancements.

Summarized results are as follows.

Category	Responses	Percentage of Responses
Strongly Agree	94	28.2%
Agree	183	54.9%
Neural	48	14.4%
Disagree	8	2.4%
Strongly Disagree	0	0%

Table 5.3: Reponses related to Latency.

Based on the results, the majority agrees on the fact that the Latency will be enhanced with the proposed and implemented techniques. The nature of the questions raised, as mentioned in the previous section, clearly highlight the importance of the proposed and implemented solution to reduce the Latency of crowdsourcing.

It can be deduced that the proposed system is a workable solution since the responses from the survey indicates that a higher number of, that is 277 agree on the fact that the solution would result in reducing the latency factor.

CHAPTER SIX

CONCLUSION

6.1 Chapter Overview

This chapter summarizes this project by providing the final comments, thoughts, and future works of this study. This study was based on crowdsourcing. It focused on optimizing crowdsourcing techniques to improve the quality, latency, and cost of crowdsourcing processes. As proposed solutions, worker rating information maintenance, notifications to workers based on different criteria, and scheduling recurring tasks were suggested.

6.2 Problem Addressed and Solutions Provided

As described in the above chapters, the optimization of crowdsourcing operations is very important, because it is important for the users who use crowdsourcing platform to achieve good productivity.

The main control mechanisms of the optimization are based on quality, cost, and latency improvements. Most of the studies based on crowdsourcing focus on optimization one or two of the control mechanisms. That is, an optimization technique which optimize all the control factors without any compromise in-between is not discussed.

The proposed solution is to the optimization of all the three control mechanisms without deprioritizing any control mechanisms.

As a solution for the above problem, the software framework to optimize crowdsourcing operations that has been developed as a result of this study is capable of providing enough functionalities to the users of the crowdsourcing platforms to add more productivity to their work by optimizing the operations based on quality, latency, and cost.

The software framework provided the users with three redefined functionalities to achieve crowdsourcing optimization.

1. *To query similar type of task solutions when a new task is entered*. Once a new task is submitted, previously crowdsourced task solutions will be checked to identify any connection to the new task.

2. *Sending notifications to workers based on the tasks' domain, keywords, and location-based information*. As a result, the workers will get the chance to identify new tasks that are mapped with their area of expertise and interesting areas. As a result, high-quality solutions will be available in a very short time, which would result in a reduction of cost as well.
3. *Automatic task posting using task scheduling for recurring tasks*. Here, tasks will be automatically posted without the intervention of the requester if the task is set as a scheduled task. The main outcome is the optimization of the latency factor. That is, the requester's time will be saved, and the results will be available in a quick time.
4. *A crowdsourcing worker rating mechanism*. The rating figures will be used to prioritize the workers' task solutions. That is, the solutions provided by workers with high ratings will be given priority. Thus, time taken by the requesters to find high-quality solutions using a given set of solutions will be reduced. Indirectly the cost will be reduced as well.

These solutions provided were proven logical and would contribute to the optimization of the crowdsourcing operations.

Based on the evaluation performed with a group of individuals both experimentally and survey-based, the feedback support that the solutions provided will have a positive impact on crowdsourcing.

During the experimental evaluation, feedback related to the proposed system's capabilities suggested that the aim of this project was covered by its objectives, that is providing a solution for crowdsourcing to optimize its operations in terms of quality, cost and latency.

Also, during the evaluation using questionnaires, it gave a further indication that the proposed solution will cater the optimization requirement required by the users who use crowdsourcing platforms.

This study fills the gap in previous solutions provided by past studies. In terms of quality, cost and latency, past studies have given different solutions to optimize each factor. But, going forward, this study proposed the solution to optimize the crowdsourcing process in terms of the control factors without any trade-off between quality, cost and latency.

Based on all the above facts, it can be concluded that the outcome of this study would give an immense contribution to the crowdsourcing operations.

6.3 Future Work

Studies related to crowdsourcing faces many challenges. One challenge is, crowdsourcing is about information gathering using humans. Because of the unpredictability of the humans it is difficult to propose solutions in the crowdsourcing domain.

During this study, the limitation is the identification of valid users. This can be an open study for the future, on how to verify user profiles based on the information thy provide. As the future work of crowdsourcing optimization, the usage of Artificial Intelligence and machine learning could be combined with the current logic to enhance the operations. That is, the biases of some of the results in the systems could be omitted with AI and ML. For example, rating per user per domain may restrict the user to be recognized as a good contributor in a new domain. So, performing the AI and ML techniques on the user could identify the user as a credible person and make the system more reliable.

REFERENCES

[1] "Digital-2020-Social-media-use-spans-almost-half-global-population." [Online]. Available: https://www.globenewswire.com/news-release/2020/01/30/1977203/0/en/Digital-2020-Social-media-use-spans-almost-half-global-population.html.

[2] G. Li, *Crowdsourced data management: hybrid machine-human computing*. New York, NY: Springer Berlin Heidelberg, 2018.

[3] "how-to-get-suggest-great-ideas-online-crowdsourcing-and-its-main-examples." [Online]. Available: brandba.se/blog/2016/4/25/how-to-get-suggest-great-ideas-online-crowdsourcing-and-its-main-examples.

[4] "Crowdfunding-Wiki." [Online]. Available: https://en.wikipedia.org/wiki/Crowdfunding.

[5] G. Paolacci, "Running experiments on Amazon Mechanical Turk," *Judgm. Decis. Mak.*, vol. 5, no. 5, p. 9, 2010.

[6] H. Garcia-Molina, M. Joglekar, A. Marcus, A. Parameswaran, and V. Verroios, "Challenges in Data Crowdsourcing," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 4, pp. 901–911, Apr. 2016, doi: 10.1109/TKDE.2016.2518669.

[7] "what-is-crowdsourcing-15026002." [Online]. Available: https://www.thestreet.com/personal-finance/education/what-is-crowdsourcing-15026002.

[8] "Microsoft_Azure." [Online]. Available: https://en.wikipedia.org/wiki/Microsoft_Azure.

[9] "Natural_language_processing." [Online]. Available: https://en.wikipedia.org/wiki/Natural_language_processing.

[10] G. Li, J. Wang, Y. Zheng, and M. J. Franklin, "Crowdsourced Data Management: A Survey," p. 23.

[11] M. Hossain, "Crowdsourcing in business and management disciplines: an integrative literature review," *J. Glob. Entrep. Res.*, vol. 5, no. 1, p. 21, Dec. 2015, doi: 10.1186/s40497-015-0039-2.

[12] K. Natasa, Gabriella, and Milic-Frayling, "Worker types and personality traits in crowdsourcing relevance labels." 2011.

[13] Carsten Eickhoff, Christopher G. Harris, Arjen P. de Vries, and Padmini Srinivasan, "Quality Through Flow and Immersion: Gamifying Crowdsourced Relevance Assessments." 2012.

[14] W. Niu, J. Caverlee, and H. Lu, "Location-Sensitive User Profiling Using Crowdsourced Labels," p. 8.

[15] P. Cheng, X. Lian, L. Chen, J. Han, and J. Zhao, "Task Assignment on Multi-Skill Oriented Spatial Crowdsourcing," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 2201–2215, Aug. 2016, doi: 10.1109/TKDE.2016.2550041.

[16] G. Li, Y. Zheng, J. Fan, J. Wang, and R. Cheng, "Crowdsourced Data Management: Overview and Challenges," in *Proceedings of the 2017 ACM International Conference on Management of Data - SIGMOD '17*, Chicago, Illinois, USA, 2017, pp. 1711–1716, doi: 10.1145/3035918.3054776.

[17] Y. Yan, R. Rosales, G. Fung, and J. G. Dy, "Active Learning from Crowds," p. 8.

[18] M. S. Bernstein, J. Brandt, R. C. Miller, and D. R. Karger, "Crowds in two seconds: enabling realtime crowd-powered interfaces," in *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*, Santa Barbara, California, USA, 2011, p. 33, doi: 10.1145/2047196.2047201.

[19] Yan, T., Kumar, and Ganesan, "Crowdsearch: exploiting crowds for accurate real-time image search on mobile phones." 2010.

[20] S. Eram and J. S. Patil, "Query Optimization to Minimize Cost and Latency for Crowdsourcing Systems," vol. 5, no. 11, p. 7.

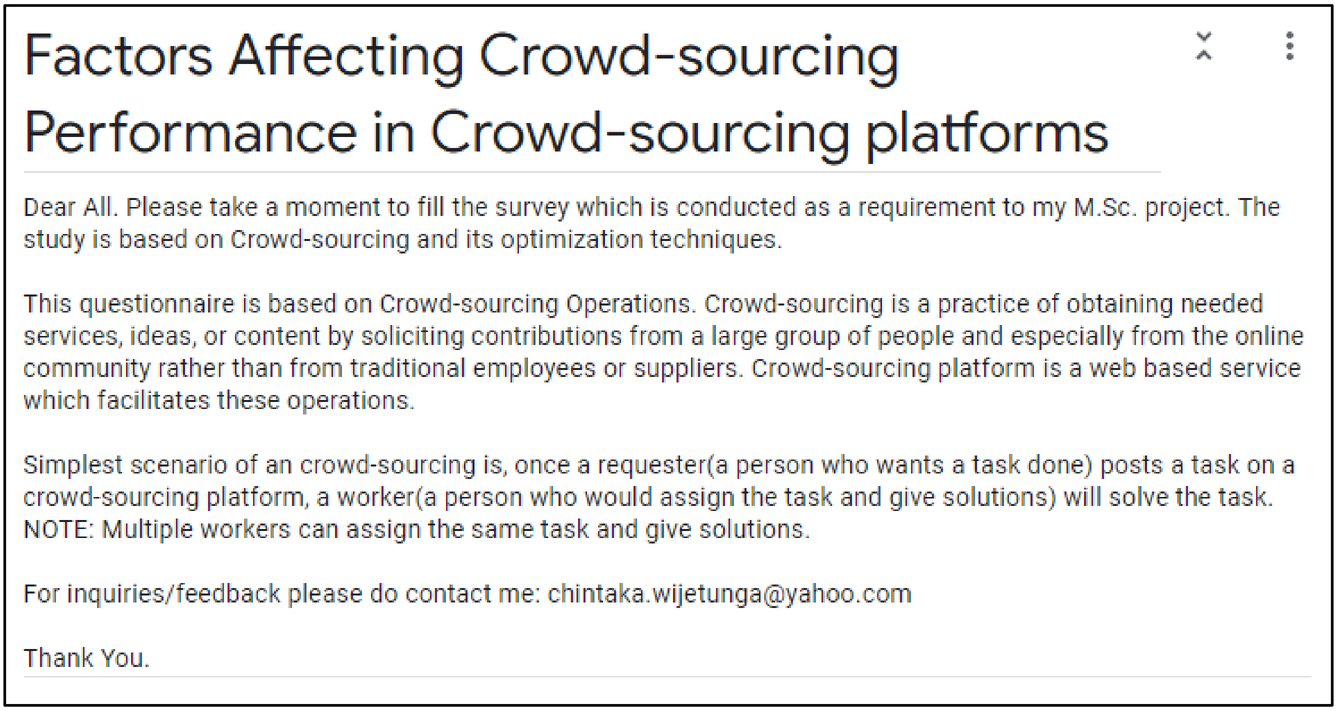
[21] V. Verroios, P. Lofgren, and H. Garcia-Molina, "tDP: An Optimal-Latency Budget Allocation Strategy for Crowdsourced MAXIMUM Operations," in *Proceedings of the 2015 ACM SIGMOD*

International Conference on Management of Data - SIGMOD ’15, Melbourne, Victoria, Australia, 2015, pp. 1047–1062, doi: 10.1145/2723372.2749440.

[22] D. Evans, P. Gruba, and J. Zobel, *How to Write a Better Thesis*. Cham: Springer International Publishing, 2014.

APPENDIX

Questionnaire



Factors Affecting Crowd-sourcing Performance in Crowd-sourcing platforms

Dear All. Please take a moment to fill the survey which is conducted as a requirement to my M.Sc. project. The study is based on Crowd-sourcing and its optimization techniques.

This questionnaire is based on Crowd-sourcing Operations. Crowd-sourcing is a practice of obtaining needed services, ideas, or content by soliciting contributions from a large group of people and especially from the online community rather than from traditional employees or suppliers. Crowd-sourcing platform is a web based service which facilitates these operations.

Simplest scenario of an crowd-sourcing is, once a requester(a person who wants a task done) posts a task on a crowd-sourcing platform, a worker(a person who would assign the task and give solutions) will solve the task. NOTE: Multiple workers can assign the same task and give solutions.

For inquiries/feedback please do contact me: chintaka.wijetunga@yahoo.com

Thank You.

All the questionnaire will contain 5 possible answers;

a. Strongly Agree
b. Agree
c. Neutral
d. Disagree
e. Strongly Disagree

Following are the questionnaire:

1. Crowdsourcing is an effective way to gather information from a set of remote workers through internet.
2. A central platform is required for a smooth flow of operations.
3. Some problems exist which can be easily solved by humans rather than machines (eg: identify images, translating sentences)
4. Quality is one of the important factors in crowdsourcing.
5. Cost reduction is one of the important factors in crowdsourcing.
6. Reducing the Latency is one of the important factors in crowdsourcing.

7. Solutions given by all the workers are high in quality all the time
8. Some workers will provide low quality solutions deliberately.
9. Workers personal background, qualities and different domain of expertise affect the quality of the solution.
10. Tasks should always be assigned to workers with an experience in the task domain area.
11. Age of the worker affect the quality of the solution.
12. Gender of the worker affect the quality of the solution.
13. Location of the worker affect the quality of the solution.
14. Quality improvement techniques may affect an increase in cost.
15. Quality improvement techniques may affect an increase in Latency.
16. A payment/incentive should be given to the workers who solve the tasks.
17. Varying the payment/incentive depending on the difficulty level of the task is a good approach.
18. Solutions for new tasks could be deduced using the solutions provided for the previously crowdsourced tasks.
19. Sending multiple tasks to the crowd rather than one task at a time will reduce the overall latency.
20. Grouping/ordering the tasks according to the difficulty level of the tasks will reduce the latency.
21. Reducing the cost will affect the latency of the solution.
22. Reducing the cost will affect the quality of the solution.
23. Latency can be improved by bundling similar type of tasks together and send to the workers.
24. Overall latency is affected by the time taken by the workers to assign a task.
25. The presentation of the task will affect the latency
26. Grouping/ordering the tasks according to the difficulty level of the tasks will reduce the cost.
27. Reducing the latency will affect the Cost.
28. Reducing the latency will affect the Quality.
29. Storing a worker profile with information related to workers is important(eg: location, contact information, interested areas, expert areas).
30. Maintaining a worker rating system is a good approach. (workers will be rated from 1 to 5. Eg: Worker ‘Lal’ has a 4.5 rating for the domain Computer Science).
31. Sending notification to the workers based on the task’s location information will enhance the quality of the solution. (eg: if the task contains a city/country information, notifications to the workers who are connected to those cities will be sent).

32. Sending notification to the workers based on the task’s domain information (Computer Science, Music, Medicine, etc.) will enhance the quality of the solution. (eg: notifications will be sent to the workers who are interested or having an expertise in that area).
33. Scheduling recurring tasks enhance the quality. (eg: if the same task must be posted in the platform weekly, the system will automatically handle it)
34. Grouping/ordering task solutions using the workers rating per the tasks domain will increase the quality of the final approved task solution.
35. Sending notification to the workers based on the task’s location information will reduce the time taken to receive a solution. (eg: if the task contains a city/country information, notifications to the workers who are connected to those cities will be sent).
36. Sending notification to the workers based on the task’s domain information (Computer Science, Music, Medicine, etc.) reduce the time taken to receive a solution. (eg: notifications will be sent to the workers who are interested or having an expertise in that area).
37. Scheduling recurring tasks reduce the time taken to receive a solution. (eg: if the same task must be posted in the platform weekly, the system will automatically handle it)
38. Grouping/ordering task solutions using the workers rating per the tasks domain will reduce the time taken to achieve a final approved task solution.
39. Sending notification to the workers based on the task’s location information will reduce the cost of receiving a solution. (eg: if the task contains a city/country information, notifications to the workers who are connected to those cities will be sent).
40. Sending notification to the workers based on the task’s domain information (Computer Science, Music, Medicine, etc.) reduce the cost of receiving a solution. (eg: notifications will be sent to the workers who are interested or having an expertise in that area).
41. Scheduling recurring tasks will reduce cost of receiving a solution. (eg: if the same task must be posted in the platform weekly, the system will automatically handle it)
42. Grouping/ordering task solutions using the workers rating per the tasks domain will reduce the cost taken to achieve a final approved task solution.
43. Improving the Quality, Cost and Latency factors will enhance the crowdsourcing operations