Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/092L17

A Quality Enhancement of Crowdsourcing based on Quality Evaluation and User-Level Task Assignment Framework

Sooyoung Lee, Kangsoo Jung and Seog Park[*]

Dept. of Computer Science and Engineering, Sogang University, Seoul, Korea

Received: 16 May 2014, Revised: 17 Jul. 2014, Accepted: 18 Jul. 2014
Published online: 1 Apr. 2015

Abstract: Crowdsourcing has recently been used in various applications, and the possibility of its utilization and importance is expected to increase continuously in the future. However, crowdsourcing cannot ensure the precision of the results because it is performed by unspecified individuals who cannot guarantee the quality of results. In particular, a more sophisticated task that has complex problems is harder to get an accurate result. In this paper, we propose a novel framework to improve the quality of work in a crowdsourcing environment. We analyze the characteristics of workers and allocate the appropriate task to workers to improve the quality of results. We propose a cumulative voting for correct assessment instead of the majority representation method which is more commonly used. Our experiments show that proposed framework improves quality of results by effective work allocation and quality analysis.

Keywords: Crowdsourcing, Task Distribution, Quality Analysis

1 Introduction

Crowdsourcing is concept of getting services, ideas, or content by contributions from a crowd of people, usually using online community, rather than traditional employees or suppliers. This practice has attracted significant interest recently because it is expected to solve various real-world problems that cannot be handled properly by traditional computing methods. The concept of crowdsourcing was first introduced by Howe [1], and [2] defined crowdsourcing as a online, distributed problem-solving and production model. Crowdsourcing has been applied to handle real-world problems, such as reCAPTCHA[3], Duolingo[4], and Amazon Mechanical Turk[5]. These frameworks provide platforms to trade crowdsourcing task via web.

However, the current focus on the crowdsourcing task is too simplistic and does not consider the capability of the public sufficiently. Existing crowdsourcing tasks assume that a set of simple jobs will be handled, even though the public can resolve complex and difficult work, such as article working[7], translation[8][9], and planning[10]]. Wikipedia[11]is a great example that shows public capability. Therefore, it is important that assign appropriate tasks to each individuals who can resolve the problem properly according to the characteristics of the task for crowdsourcing. Moreover, to increase the quality of the results obtained through crowdsourcing, an accurate evaluation of the results of each task is important.

We suggest a framework that can improve the quality of results in an environment to solve problems by crowdsourcing. Proposed framework consists of task management, worker management, task distribution, and quality analysis. The remainder of this paper is organized as follows: Section 2 reviews the related literature in crowdsourcing. Section 3 explains the proposed architecture and algorithms to assign and evaluate the task. Section 4 describes our use of simulations to validate the results. Finally, the last section summarizes the contributions of this paper.

2 Related Work

Most existing works on crowdsourcing typically focus on the extension of traditional techniques into crowdsourcing

[*] Corresponding author e-mail: spark@sogang.ac.kr

and its optimization. CrowdDB[12], Qurk[13] and Deco[14] are techniques that use crowdsourcing to extend existing database systems and can use SQL-like query in a crowdsourcing platform. Studies on query utilization for crowdsourcing have been researched, including the optimizations of approaches such as SELECT, JOIN, SORT, MAX, GROUP BY, and Entity Resolution. Others works focused on query result size prediction under an open world assumption[15], selecting the best workers from the set of specified workers and budget[16], and adopting workers in advance and grasping the work quickly to reduce searching space[17][18].

Reliability management is important for crowdsourcing. [19] suggests a method for managing the stepwise reputation of workers. Moreover, studies on the control of spammers, who solve a task randomly, and streakers, who resolve most of the work alone, were introduced. [20] suggests a task-worker matrix that recommends work based on the history of past tasks for efficient allocation of work.

Numerous works address query optimization for cost reduction. However, these works rely on simple worker selection, which does not consider the characteristics of each worker. And by using a majority representation system as a quality evaluation, there are some waste of quality and low accuracy of evaluation.

As it can be seen from Wikipedia, crowdsourcing can solve more complex problem although existing works just focus on simple task using crowdsourcing. We introduce a novel framework for high quality crowdsourcing to solve more sophisticated problem using efficient work distribution and evaluation system.

3 SYSTEM ARCHITECTURE

We provide a novel framework that consists of task and worker management, task distribution, and quality analysis, as illustrated in Figure 1. The task and worker management component analyzes and manages properties of tasks and registered workers. Then the task distribution component utilizes this information to assign the appropriate tasks to workers. Finally, the quality evaluation component evaluates the results of crowdsourcing and elects the best qualified result to be returned to the service requester.

3.1 Task management

A total task set T=$\{t_1,...,t_m\}$ should be considered, and the size of task is $|T|$=m. Task Level TL_j refers to task difficulty, which is determined by analyzing the crowdsourcing task characteristics. For example, crowdsourcing translation can use FleschKincaid Grade Level (FKGL)[21] to determine the task difficulty. Each task information such as predetermined difficulty, actual

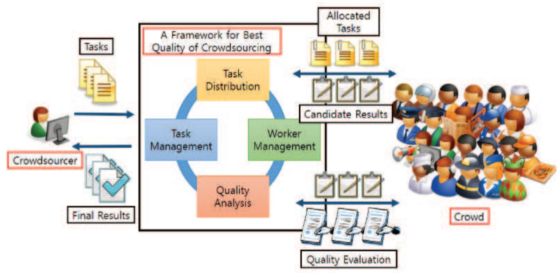


Fig. 1: A framework for high quality of crowdsourcing

evaluation of worker, duration of labor, and analysis of the result can be collected to calculate for determining task level of future works. This analysis is used to assign task to appropriate workers.

3.2 Work management

Let worker set W=$\{w_1,...,w_n\}$ and the total number of the workers be $|W|$=n. Further, expected arriving time of each workers is $Time_i=\{time_{i1},...,time_{il}\}$, and the time duration is d_i that is needed to resolve the task of each worker w_i. The worker management component also deals with this information and analyzes the capability of workers and evaluates each workers Skill Level SL_i. $W_c ur$ denotes the set of workers that are currently connected to the system. Let system variable DL be the waiting time of tasks that the system can wait for additional workers. Additional workers who are expected to arrive sooner or later are denoted by W_{in} as presented below.

$W_{in} = \{w_i$ |for all i and l, $time_{il} + d_i \leq time_{now}$ + DL $\}$

As a result, the total set of available workers is W_{avail}= $W_{cur} \bigcup W_{in}$ at $time_{now}$. Additionally, we assume that the number of tasks is greater than the number of workers. It is reasonable assumption because in real world, there are tons of problems that wait for worker who can solve it, and the number of workers is much less than the number of problems.

3.3 Task distribution

Assigning the appropriate tasks to workers significantly affects the quality of the task in a crowdsourcing environment to solve complex problems. For example, we assume that we have a task t_j with TL_j=10 and workers w_a, w_b with SL_a=10 and SL_b=5. The task should be assigned to w_a than to w_b. We assume that the proposed task distribution method try to minimize the difference between the level of skill of workers and the difficulty of the tasks because the number of workers is less than the number of tasks. Specifically, the problem of finding the best work placement is as follows:

```
Algorithm 1. The best task distribution as user and task level

Procedure BestTaskDistribution
  input: n', m,     // size of W and T
         SL, TL     // Skill and Task level
  output: f[]       // best distribution function
begin
  W[n'] = {SL1, ..., SLn'};  // SLa ≤ SLb if a ≤ b
  T[m] = {TL1, ..., TLm};    // TLa ≤ TLb if a ≤ b
  d[n'][m] = {∞};            // d(i, j)
  Tr[n'][m] = {(0, 0)};      // trace of d(i, j)
  /* calculate d[][] by dynamic programming */
  for i = 1 to n' do
    for j = 1 to m do
      if i ≤ j then
        /* calculate d(i, j) and save the trace */
        preMin = min_{i≤k≤j-1}(d[i-1][k] + |SLi-TLj|);
        d[i][j] = min(preMin, d[i][j-1]);
        /* preMinK is index of preMin in T */
        Tr[i][j] = preMin ≤ d[i][j-1] ? (i-1, preMinK) : (i, j-1);
      end if
    end for
  end for
  /* find best f */
  f[n'] = (n', minm) where d[n'][minm] is the min of all d[n'][];
  (x, y) = Tr[n'][minm]; i = n' - 1;
  while i > 0 do
    f[i] = (x, y); (x, y) = Tr[x][y];
    if x == i - 1 then i = i - 1;
  end while
  return f[];      // return best distribution
end of Procedure BestTaskDistribution
```

Fig. 2: An algorithm for finding the best task distribution by dynamic programming

Problem 1. The Best Task Distribution Given two sets T and W_{avail}, find the function f that has the minimum $\sum_{w_i \in W_{avail}} |SL_i - TL_{f(i)}|$ in all possible one-to-one functions $f : W_{avail} \to$ T.

Brute-force search is the simplest method to solve problem 1, but we have ${}_mP_{n'}$ ways, and the time complexity is $O(m!/(m-n')!)$ in the worst case. To reduce time complexity, we use dynamic programming and greedy algorithm. The greedy method matches each element of W_{avail} to each element of T based on the minimum difference between skill level and task difficulty. Although this method does not guarantee the optimal solution, it has only $O(mn')$ time complexity. Meanwhile, by using dynamic programming, we can find optimal solution. In Figure 2, the dynamic programming algorithm for solving Problem 1 is described. First, the algorithm sorts the elements of T and W_{avail} based on SL and TL. Then the minimum summation of difference between task and skill level d(i,j) is as follows:

d(i,j)=min$\{min_{1\leq k\leq j-1}\{$d(i-1,k)+ $|SL_i - TL_j|\}$, d(i,j-1)$\}$, $i \leq j$

This algorithm has $O(m^2$n')time complexity, such that we can solve the problem 1 in polynomial time.

3.4 Quality analysis

When the tasks are completed, the result is reported to the framework. In the case of a simple task, an existing work assignment guarantees sufficient quality. But for the difficult and complex tasks, the best performance result is hard to select that which one is the correct answer among several receiving candidates.

Assume that we have two candidates, and we should select the better one. For simple task example, ”What is the biggest number between 2 and 3?” This case is trivial because 3 is easily determined to be bigger than 2. Quality analysis can be quickly and easily implemented through a plurality voting system, which is every worker has one ballot.

However, in some cases, selecting the better option is difficult despite high worker accuracy because the answer of task results cannot be determined simply. For example, Which translation is better than the other or ”Which place should we go to between the East Gate and the Namsan Tower if someone visit Seoul for the first time?” Answers to these questions are subjective, which makes it difficult to determine a correct answer. In this case, unanimity is not guaranteed despite 100% worker accuracy because workers vote by their subjective opinion.

As a result, query result evaluation in crowdsourcing is a difficult problem. We define this difficulty in quality evaluation in crowdsourcing as follows:

Definition 1. Hardness of Quality Evaluation When we evaluate the result of crowdsourcing, we cannot guarantee unanimity every time, even if the accuracy of all workers is 100%. We define this issue as hardness of quality evaluation.

Definition 2. Relative Quality When all of the workers' accuracy is 100% for the hardness of quality evaluation, we define the proportion of vote of each candidate as candidates relative quality.

Definition 3. Correct answer in hard evaluation Correct answer in hard evaluation is determined to the candidate answer that has highest relative quality

In hardness of quality evaluation, the majority representation systems have increased error rate because each worker casts one vote. For example, we assume that two candidates, A and B, which have a relative quality $q_A = 0.7$ and $q_B = 0.3$,

respectively. Also we assume the average accuracy of each worker is 90%. When five workers are evaluate the two candidates based on the majority representation voting system, the probability of selecting A is 0.7×0.9+0.3×0.1=0.66, whereas that for B is 0.34. Then the probability of selecting A as the higher quality, i.e. the correct answer for this evaluation, $(0.66)^3$+$(0.66)^3(0.34)_3C_1$+$(0.66)^3(0.34)^2{}_4C_2$=0.78. Also the required average ballots for this evaluation, i.e. the cost, is
$3(0.66)^3$+$4(0.66)^3(0.34)_3C_1$+$5(0.66)^3(0.34)^2{}_4C_2$+
$3(0.34)^3$+$4(0.34)^3(0.66)_3C_1$+$5(0.34)^3(0.66)^2{}_4C_2$=3.98

If the relative qualities of two candidates are 1.0, 0.0, the average correct answer rate is 0.99, and the average cost is 3.31. The majority representation voting system decrease answer rate and increased average cost when it apply to hard evaluation problem.

In our proposed framework, we apply a cumulative voting system for the hard evaluation. In hard evaluation, there is no candidate which has absolutely outstanding quality. The cumulative voting system, that each worker has more than one ballots, can close to the relative quality more quickly and accurately than majority representation voting system. In the proposed framework, we apply the cumulative voting systems as follows: Each worker grades each candidate from 0 to 100 points in increments of 10 points and the sum of all points is 100. Then the candidate answer with the highest point is to be the correct answer.

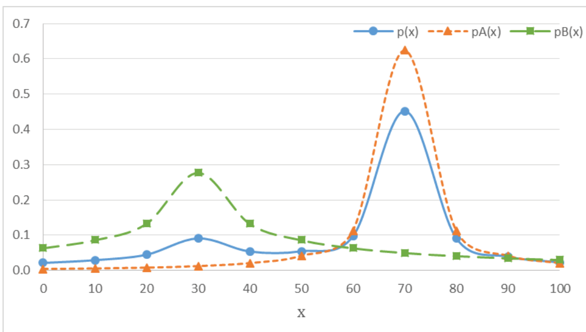


Fig. 3: An example of cumulative voting modeling

We use Zipf-like distribution [22] and gravity model for estimation modeling to predict the evaluation under the cumulative voting system. For example, as the previous example, we assume the worker who prefer A gives 70 point to A and 30 points to B by following the relative qualities. We define $p_A(x)$ as the probability that a worker who prefer A gives x points to A. And $p_B(x)$ is also a probability that a worker who prefer B gives x points to B. We set up two Zipf-like probability density functions as shown in Figure 3 and 4.

The skewness of each probability density function, namely, $_A$, $_B$, can be calculated using the gravity model. We assume that, in the previous example, Q_A=q_A100=70, Q_B=q_B100=30, and $r = |Q_A - Q_B|$. To define the skewness, we use Equation (1).

$$F = G\frac{Q_A \times Q_B}{r^2} \tag{1}$$

where G is a constant to define the skewness. Then we calculate $_A$ and $_B$ using F as below

$$F = Q_A \times \frac{1}{\theta_A} = Q_B \times \frac{1}{\theta_B} \tag{2}$$

That is, θ_A=$r^2/(Q_B \times G)$, θ_B=$r^2/(Q_A \times G)$. The proposed model defines the skewness of two distributions using G. So, if we want to estimate workers average accuracy, we can determine Gusing average accuracy. Let the workers average accuracy is as the probability that every worker casts over 50 points to their preferred candidate based on relative quality. α is described by Equation (3).

$q_A = 0.7$, $q_B = 0.3$, $\alpha = 0.9$, $G = 21.566777$, $\theta_A = 2.4729$, $\theta_B = 1.0598$
$p_K(x) = (1/Rank_K(x))^{\theta_K} / \sum_{for\ all\ x}(1/Rank_K(x))^{\theta_K}$, $p(x) = p_A(x) \times q_A + p_B(x) \times q_B$

x	$Rank_A(x)$	$Rank_B(x)$	$(1/Rank_A(x))^{\theta_A}$	$(1/Rank_B(x))^{\theta_B}$	$p_A(x)$	$p_B(x)$	$p(x)$
0	8	4	0.00584	0.23010	0.00365	0.06369	0.02166
10	7	3	0.00813	0.31213	0.00508	0.08640	0.02947
20	6	2	0.01190	0.47969	0.00743	0.13278	0.04503
30	5	1	0.01868	1.00000	0.01166	0.27680	0.09120
40	4	2	0.03244	0.47969	0.02025	0.13278	0.05401
50	3	3	0.06609	0.31213	0.04126	0.08640	0.05480
60	2	4	0.18012	0.23010	0.11245	0.06369	0.09782
70	1	5	1.00000	0.18164	0.62427	0.05028	0.45207
80	2	6	0.18012	0.14972	0.11245	0.04144	0.09115
90	3	7	0.06609	0.12716	0.04126	0.03520	0.03944
100	4	8	0.03244	0.11038	0.02025	0.03055	0.02334

Fig. 4: The probabilities of cumulative voting modeling example

$$\begin{aligned}\alpha &= \sum_{x=0,10,\ldots,50} (P_A(100-x)q_A + p_B(x)q_B) \\ &= \sum_{x=0,10,\ldots,50} \Big(\frac{(1/Rank_A(100-x))^{\theta_A}}{\sum_{for all k}(1/Rank_A(100-k))^{\theta_A}} q_A \\ &\quad + \frac{(1/Rank_B(x))^{\theta_B}}{\sum_{for all k}(1/Rank_B(k))^{\theta_B}} q_B\Big)\end{aligned} \tag{3}$$

As a result, we can determine G by finding the solution of the equation g(G)=0.

$$\begin{aligned}g(G) = \sum_{x=0,10,\ldots,50} \Big(&\frac{(1/Rank_A(100-x))^{\theta_A}}{\sum_{for all k}(1/Rank_A(100-k))^{\theta_A}} q_A \\ &+ \frac{(1/Rank_B(x))^{\theta_B}}{\sum_{for all k}(1/Rank_B(k))^{\theta_B}} q_B\Big) - \alpha\end{aligned} \tag{4}$$

g(G) is a monotone decreasing function where G is a positive real number and it has only one solution for g(G)=0. Therefore we can use the NewtonRapshon method to solve the equation. In the previous example, we apply q_A=0.7, q_B=0.3, and α =0.9 to this equation, then we obtain G $\approx$ 21.566777. Figure 3 and 4 presents the whole probability distribution of this model as p(x)=p_A(x)$\times$ q_A+p_B(x)$\times$ q_B.

4 EXPERIMENTAL RESULTS

We perform experiments to validate proposed frameworks accuracy and efficiency by simulation and implementation.

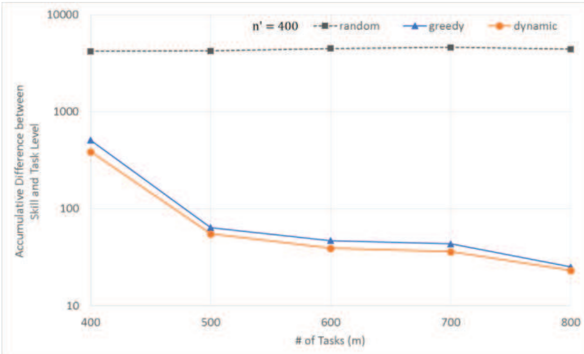


Fig. 5: The best distribution experiments result

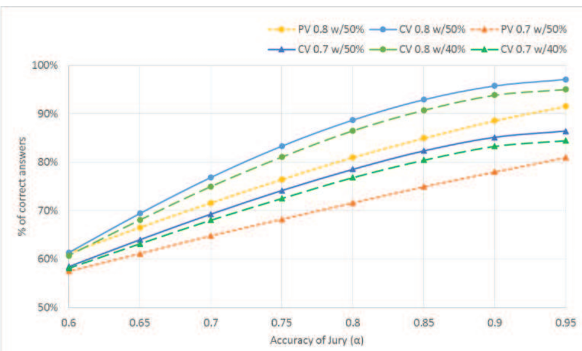


Fig. 6: The accuracy of jury vs. % of correct answers.

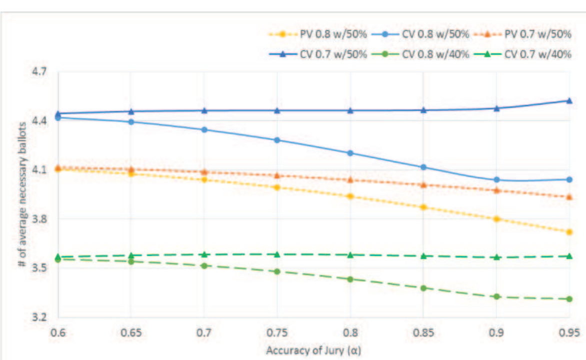


Fig. 7: The accuracy of jury vs. # of average necessary ballots.

We carry out two experimental evaluations by simulation: (1) the best distribution experiments result, (2) comparing between Plurality voting (PV) vs. Cumulative voting (CV) and implement web application crowdsourcing for translating using proposed framework to validate accuracy and efficiency.

4.1 Best distribution

We begin with experiments on normal distribution data sets with values from 20 to 80 and deviation of 10. The results of best distribution experiments are shown in Figure 5. These result shows that using dynamic programming results makes a significant reduction rather than random distribution. When the number of workers(n') is fixed and the number of tasks(m) various, the greedy and dynamic method can achieve best distribution(or almost that). But using the random method cannot get the proper distribution at any n' and m. In most cases, we can use the dynamic method for find best distribution. But when the size of problems is too large to using dynamic method, we can use the greedy method for the close approximation of the best distribution.

4.2 Plurality voting (PV) vs. Cumulative voting (CV)

The result of the comparison between PV and CV is depicted in Figure 6, 7, and 8. In the Figure 6 and 7, PV and CV are compared by five evaluators with precision on the hard evaluation problem, where q_A=0.7,0.8. In addition, for the needed shares to conclude the evaluation, experiments are conducted by setting 40% (200 points) and 50% (three votes, 250 points).

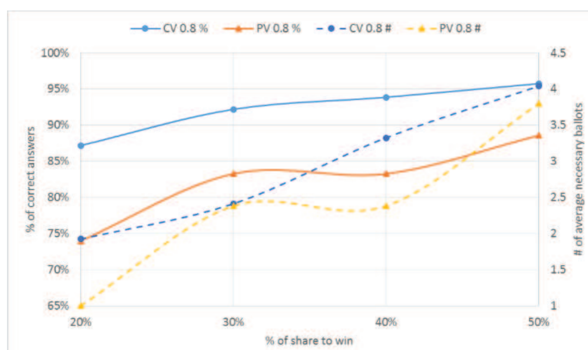


Fig. 8: The % of shares to win vs. % of correct answers(solid line) and # of average necessary ballots(dotted line)

Figure 6 shows that the precision of CV with 50% needed share(CV w/50%) is more accurate than that of PV with 50% share(PV w/50%). It shows that CV is more suitable for the hard evaluation problems than PV. However, in Figure 6, the number of average necessary ballots of CV w/50% is greater than that of PV w/50%. In other words, CV w/50% need more cost for more accurate decision. However, in Figure 6, although we reduce the needed share for the CV from 50% to 40%, the rate of correct answer of CV is not decrease as so much, and it is higher than PV as ever. Also, in Figure 7, the number of average necessary ballots of CV w/40% is lower than that of the CV w/50% and even taht of the PV. So, we can reduce the cost by reduce the needed share when we use the CV. The proposed framework can determines the CV model according to the average accuracy of evaluators and the relative qualities of candidates. Then the framework can predict the expected rate of correct answers and the necessary ballots(costs). Adopting this prediction can control the rate of correct answers and the average number of necessary ballots through the management of needed share for determination the evaluation result. Figure 8 shows the result of a comparison between the average necessary number of ballots (dotted line) and the ratio of correct answers(solid line) of CV and PV in accordance with the needed share for determination of evaluation (α=0.9, q_A=0.8). CV has a higher ratio of correctness than PV at any needed share, but the necessary cost of CV is also higher than that of PV as we seen before. Nevertheless, the decrement of correctness ratio of CV is tolerable, we can reduce the needed share rate and cost to focus on the target correctness ratio. For example, where the needed share is 50%, the correctness ratio of PV is 88.6%, and the average number of necessary ballots is 3.8. Meanwhile, in CV, where the needed share is 20%, the correctness ratio is 87.2%, and the average number of necessary ballots is 1.9. In conclusion, cumulative voting system maintains the correctness ratio by modeling the votes and can obtains a higher correctness ratio with relatively less or small additional cost.

4.3 Implementation of propsoed framework

We implement crowdsourcing service using proposed framework to confirm accuracy and efficiency of proposed framework. The translation service have been targeted for experiment. We have selected 60 sentence of CNN News to translate and the number of worker is 16. Our estimation measure is BLEU point[23] that is used to estimate quality of machine translation. At first, we measure BLEU point of workers translation ability using sample sentence that have different level of difficulty. As a result, the best point of BLEU is 22.08 and the worst point is 8.11, and average point is 12.66. Difficulty of sentence is also measured using FKGL[21]. By this measurement, the most difficult sentences FKGL point is 18.0 and the least difficult sentences point is 3.1, and average FKGL point is 11.42. And then, we distribute sentences to worker by sentence difficulty and workers level of translation using proposed framework. Experimental result is shown in Figure 9.

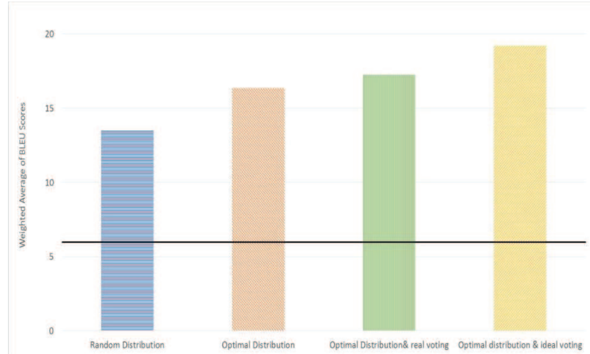


Fig. 9: Evaluation of crowdsourcing quality using proposed framework in real world implementation

We compare google translation result to crowdsourcing translation. Google translation results BLEU point is 5.98. This result is 64% less than other translation technique using crowdsourcing. It means that crowdsourcing can obtain better quality of translation result than machine translation. We also compare proposed task assignment technique to other crowdsourcing technique. We carry out 4 type of method to comparison: random distribution, optimal distribution, optimal and real world voting, and optimal distribution and ideal voting. Optimal distribution is a technique that we proposed using dynamic algorithm to assign appropriate task to worker, and voting means that evaluate translation results vote using cumulative voting. Ideal voting is calculate based on assumption that every evaluator vote 100% correct, and real voting is the results that evaluator vote in real implemented crowdsourcing framework. We confirm that proposed assignment technique get a higher BLEU point than random distribution. That is, optimal distribution succeed to get better translation results by appropriate distribution. We also compare ideal voting and real voting using optimal distribution. Real voting using optimal distribution real voting cannot get the same translation quality than ideal voting, but it can get better result than optimal distribution without voting that evaluate query results.

5 CONCLUSION

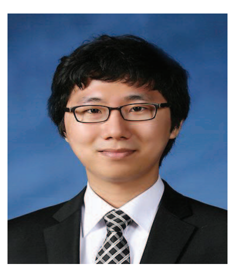
In this paper, we presented a novel framework to improve the quality of result in a crowdsourcing environment. This framework using the best task distribution and cumulative voting system for the hard evaluation. As presented in the experimental results, we show that we can get the best task distribution by dynamic programming and apply cumulative voting system for higher answer rate and lower cost quality evaluation by proposed modeling method. For the future work, we plan to compare various criteria to apply to real-world task distribution. Also, more accurately estimating the relative quality and average correctness of evaluators, which are parameters of the model, is needed.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2013R1A1A2013172)

References

[1] J. Howe, "The Rise of Crowdsourcing.", Wired magazine, **14**, 1-4, 2006.
[2] D. C. Brabham, "Crowdsourcing as a Model for Problem Solving an Introduction and Cases.", Convergence: the International Journal of Research into New Media Technologies, **14**, 75-90, 2008.
[3] L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, "recaptcha: Human-Based Character Recognition via Web Security Measures.", Science, **321**, 1465-1468, 2008.
[4] http://www.duolingo.com/
[5] http://www.mturk.com/
[6] H. Zhang, E. Horvitz, R. C. Miller, and D. C. Parkes, "Crowdsourcing General Computation.", In Proc. of the 2011 ACM SIGCHI International Conference on Human Factors in Computing Systems, 2011.
[7] M. S. Bernstein, G. Little, R. C. Miller, B. Hartmann, M. S. Ackerman, D. R. Karger, D. Crowell, and K. Panovich, "Soylent: A Word Processor with A Crowd Inside.", In Proc. of the 23nd Annual ACM Symposium on User Interface Software and Technology, pp. 313-322, 2010.
[8] V. Ambati, S. Vogel, and J. Carbonell, "Collaborative Workflow for Crowdsourcing Translation.", In Proc. of the ACM 2012 Conference on Computer Supported Cooperative Work, pp. 1191-1194, 2012.
[9] O. Zaidan, and C. Callison-Burch, "Crowdsourcing Translation: Professional Quality from Non-Professionals.", In Proc. of the 49th Annual Meeting of the Association for Computational Linguistics, pp. 1220-1229, 2011.
[10] H. Kaplan, I. Lotosh, T. Milo, and S. Novgorodov, "Answering Planning Queries with the Crowd.", In Proc. of the Very Large DataBases Endowment, **6**, 697-708, 2013.
[11] http://www.wikipedia.org/
[12] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. "CrowdDB: Answering Queries with Crowdsourcing.", In Proc. of the 2011 ACM SIGMOD International Conference on Management of Data, pp. 61-72, 2011.
[13] A. Marcus, E. Wu, D. R. Karger, S. R. Madden, and R. C. Miller, "Crowdsourced Databases: Query Processing with People.", In Proc. of 5th Biennial Conference on Innovative Data Systems Research, pp. 211-214, 2011.
[14] H. Park, H. Garcia-Molina, R. Pang, N. Polyzotis, A. Parameswaran, and J. Widom, "Deco: A System for Declarative Crowdsourcing.", In Proc. of the Very Large DataBases Endowment, **5**, 1990-1993, 2012.
[15] M. J. Franklin, B. Trushkowsky, P. Sarkar, and T. Kraska, "Crowdsourced Enumeration Queries.", In Proc. of the 2013 IEEE International Conference on Data Engineering, pp. 673-684, 2013.
[16] C. C. Cao, J. She, Y. Tong, and L. Chen, "Whom to Ask?: Jury Selection for Decision Making Tasks on Micro-Blog Services.", In Proc. of the Very Large DataBases Endowment, **5**, 1495-1506, 2012.
[17] J. P. Bigham, C. Jayant, H. Ji, G. Little, A. Miller, R. C. Miller, R. Miller, A. Tatarowicz, B. White, S. White, and T. Yeh, "VizWiz: Nearly Real-Time Answers to Visual Questions.", In Proc. of the 23nd Annual ACM Symposium on User Interface Software and Technology, pp. 333-342, 2010.
[18] M. S. Bernstein, J. Brandt, R. C. Miller, and D. R. Karger, "Crowds in Two Seconds: Enabling Realtime Crowd-Powered Interfaces.", In Proc. of the 24th Annual ACM Symposium on User Interface Software and Technology, pp. 33-42, 2011.
[19] M. Allahbakhsh, A. Ignjatovic, B. Benatallah, S. Beheshti, E. Bertino, and N. Foo, "Reputation Management in Crowdsourcing Systems.", In Proc. of the 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing, pp. 664-671, 2012.
[20] M. C. Yuen, I. King, and K. Leung, "Task Recommendation in Crowdsourcing Systems." In Proc. of the 1st International Workshop on Crowdsourcing and Data Mining, pp. 22-26, 2012.
[21] J. P. Kincaid, R. P. Fishburne Jr, R. L. Rogers, and B. S. Chissom, "Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel", In Naval Technical Training Command Research Branch Report, no. 8-75, 1975.
[22] G. K. Zipf, "Human Behavior and the Principles of Least Effort", Addison-Wesley, 1949.
[23] K. Papineni, S. Roukos, T. Ward and W. J. Zhu, "BLEU: a method for automatic evaluation of machine translation.", In Proc. of the 40th annual meeting on association for computational linguistics, pp. 311-318, 2002.
[24] http://translate.google.com



Sooyoung Lee received the M.S degree in Computer Engineering from Sogang University in 2014. His research interests are in the areas of crowdsorucing, privacy, and database security.

Kangsoo Jung received the M.S degree in Computer Engineering from Sogang University in 2009. His research interests are in the areas of data privacy, access control.

Seog Park is a professor of Computer Science and Engineering at Sogang University. He received the B.S. degree in Computer Science from Seoul National University in 1978, the M.S. and the Ph.D. degrees in Computer Science from Korea Advanced Institute of Science and Technology (KAIST) in 1980 and 1983, respectively. Since 1983, he has been working in the Department of Computer Science and Engineering, Sogang University. His major research areas are database security, real-time systems,data warehouse, digital library, multimedia database systems, role-based access control and web database. He is a member of the IEEE Computer Society, ACM and the Korean Institute of Information Scientists and Engineers (KIISE). Also, he has been a member of Database Systems for Advanced Application (DASFAA) Steering Committee since 1999.