CrowdSeed: Query Processing on Microblogs

Zhou Zhao, Wilfred Ng and Zhijun Zhang
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Hong Kong, China
{zhaozhou,wilfred}@cse.ust.hk,zzhangac@stu.ust.hk

ABSTRACT

Databases often offer poor answers with respect to judgemental queries such as asking the best among the movies shown in recent months. Processing such queries requires human input for providing missing information in order to clarify uncertainty or inconsistency in queries. Nowadays, it is common to see people seeking answers on micro-blogs through asking or sharing questions with their friends. This can be easily done via smart phones, which diffuse a question to a large number of users through message propagation in microblogs. This trend is important and known as *CrowdSearch*. Due to conflicting attitudes among crowds, the majority vote is employed as a crowd-wisdom aggregation schema. In this demo, we show the problem of minimizing the monetary cost of a crowdsourced query, given the specified expected accuracy of the aggregated answer. We present CrowdSeed, a system that automatically integrates human input for processing queries imposed on microblogs. We demonstrate the effectiveness and efficiency of our system using real world data, as well as presenting interesting results from a game called "Who is in the CrowdSeed?".

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications

Keywords

Crowdsourcing, Uncertainty, Microblogs

1. INTRODUCTION

In recent years, Crowdsourcing databases [2, 6, 3, 7] have attracted substantial interest in the research community. Many fundamental infrastructures are proposed to support various kinds of query processing on the crowd. Amazingly, the wisdom of crowds has been proved to outperform computer programs at various kinds of tasks, especially for image tagging, natural language processing and so on. Crowdsourcing relies on human workers to complete, at least partially, the job of query processing.

However, humans are prone to error, which may provide extremely poor quality crowdsourcing results. To address the above problems, crowdsourcing applications often enroll a number of workers to process the replicated queries. If the collected results from workers are conflicting, the majority vote is adopted to determine which is correct. Currently, crowdsourcing applications duplicate the issued queries and publish them on designed platforms, such as Amazon M-Turk[1], CrowdFlower[2] and the like.

In this paper, we present a system *CrowdSeed* that enables crowdsourced queries to be processed on microblogs. Typically, we focus on addressing the following issues:

- **Diversity of Answers.** The replication of queries may not fully solve this problem. If the number of replicated queries are few, we may not have enough confidence to infer a reliable answer. However, if we duplicate too many queries, we may have to suffer high cost [4].
- **MinCost.** Some users may not be reluctant to process the query until they can receive some reward. So, we aim to minimize the monetary cost of processing the query on microblogs, given a specified accuracy threshold of a crowdsourced query.
- **Online Aggregation.** The latency of humans to complete a query is difficult to predict, which is a major drawback of crowdsourcing. To tackle this problem, we present an online fusion algorithm that progressively aggregates the answers.

Some papers have already addressed the problem of conflicting results [3, 7]. However, none of them resolves all the above issues satisfactorily. Furthermore, we also study the query diffusion on the microblogs.

The underlying idea of our approach is as follows. First, we compute the number of users whose aggregate answers can meet the specified accuracy threshold α, denoted by τ. Next, we try to find a *MinCost* set of users whose query diffusion can research τ replies, called C. Then, we "tweet" the query to the set C and give them the required reward. Finally, we let the users of C "retweet" the query on microblogs. After that, the *CrowdSeed* progressively aggregates the replies and provides the correct answer with confidence.

[1] `https://www.mturk.com/mturk/welcome`
[2] `http://crowdflower.com/`

EDBT/ICDT '13 March 18 - 22 2013, Genoa, Italy

This paper is organized as follows. Section 2 introduces the design of *CrowdSeed* and formulates the problem while Section 3 presents our algorithm. Section 4 then introduces the architecture of our system. Section 5 discusses the demonstration plan and we conclude the paper in Section 6.

2. CROWDSEED DESIGN

2.1 User Reward

The existing crowdsoucing platforms pay each user a fixed amount of money for completing a query. To encourage more users to join our system, *CrowdSeed* enables each user w_i to set a price for processing and "tweeting" the query. The users can be paid when they are selected in the seed C. The monetary cost of a crowdsourced query Q is given by

$$cost(Q) = \sum_{w_i \in C} m(w_i) \tag{1}$$

where $m(w_i)$ is the price set by user w_i.

2.2 Majority Voting Rule

If the replies from users are conflicting, we resolve it by adopting *majority voting rule*. The formula of majority voting is given by

$$f(V) = \begin{cases} 1 \text{ if } \sum_{w_i \in C} v(w_i) \geq \frac{n+1}{2} \\ 0 \text{ otherwise} \end{cases} \tag{2}$$

where the vote of each user $v(w_i)$ is binary (i.e. 0 or 1) and V represents the collected votes. We select an answer which is supported by more than half users as the correct one.

However, the output of *majority voting rule* may not be reliable if the number of voters is few. On the other hand, the cost may be huge if we enroll too many users. To tackle this problem, we propose a probabilistic model to estimate the accuracy of *majority voting rule*.

Suppose the accuracy of users that have processed the query are $\{a(w_1), \ldots, a(w_\tau)\}$, where $a(w_i)$ is the probability of w_i giving the correct answer. The output answer by *majority voting* is correct only when at least half of the users provide the correct answers.

Given a set of votes V, we can estimate the correctness probability of *majority voting rule*, denoted as $Pr(f(V))$. We denote that A represents the set of the users who give the correct answer. Suppose that we have τ votes from the users, then the correctness probability of *majority voting rule* is given by

$$\begin{aligned} Pr(f(V)) &= Pr(|A| \geq \frac{|\tau|+1}{2}) \\ &= \sum_{k=\frac{|\tau|+1}{2}}^{|\tau|} \sum_{A \in F_k} \prod_{w_i \in A} a(w_i) \prod_{w_j \notin A} (1 - a(w_j)) \end{aligned} \tag{3}$$

where F_k is a set of subsets of size k. For example, given that we have collected results from three users, then $F_2 = \{\{w_1, w_2\}, \{w_1, w_3\}, \{w_2, w_3\}\}$.

However, it may be difficult to compute the Equation 3 since it is hard to estimate the accuracy of each user in a microblog. To tackle this problem, we turn to utilize the professionalism of the microblog to compute the expectation of Equation 3. We view the professionalism of the microblog as the average accuracy of the users in it. Then, the expected accuracy of *majority voting rule* is given by

$$\begin{aligned} E[Pr(f(V))] &= E[\sum_{k=\frac{|\tau|+1}{2}}^{|\tau|} \sum_{A \in F_k} \prod_{w_i \in A} a(w_i) \prod_{w_j \notin A} (1 - a(w_j))] \\ &= \sum_{k=\frac{|\tau|+1}{2}}^{|\tau|} \binom{|\tau|}{k} \mu^k (1-\mu)^{n-k} \end{aligned} \tag{4}$$

where the details of derivation can be found in [4].

By using the Chernoff Bound, we have the lower bound of the expected correctness, given by

$$\sum_{k=\frac{|\tau|+1}{2}}^{|\tau|} \binom{|\tau|}{k} \mu^k (1-\mu)^{|\tau|-k} \geq 1 - e^{-2|\tau|(\mu - \frac{1}{2})^2}. \tag{5}$$

Given an expected correctness α, the minimum number of required workers is given by

$$|\tau| \geq \frac{-\ln(1-\alpha)}{2(\mu - \frac{1}{2})^2}. \tag{6}$$

We aim to select a set of users in the microblog such that: (1) we could have at least τ replies from the users. (2) The monetary cost of the seed C is minimized.

2.3 Query Diffusion

We study the crowdsourced query diffusion on microblogs based on *word of mouth* effect. We denote that the user w_j is willing to further processing the query diffused from the user w_i when w_i is able to influence w_j (i.e. $I(w_i, w_j)$). The probability of the *word of mouth* effect can be estimated by the similarity between two users [1]. The number common friends is often used to measure the similarity between two users and their co-influence. Then the probability of the *word of mouth* effect (i.e. $Pr(I(w_i, w_j))$) can be represented by *weighted Jaccard Distance* of the friends of two users, given by

$$Pr(I(w_i, w_j)) = \lambda \frac{|N(w_i) \bigcap N(w_j)|}{|N(w_i) \bigcup N(w_j)|} \tag{7}$$

where $N(w_i)$ is the set of user w_i's friends and λ is an coefficient depending on the specific microblogs. Equation 7 is very reasonable in the real life. If two users share a lot of friends, they may be close friends. Furthermore, it is easier to diffuse query from one to another.

Using the models proposed above, we define the problem of query processing on social networks below.

PROBLEM 1. *Given a graph of the microblog $G(V, E)$ and a crowdsourced query Q, we aim to find a set of users for query diffusion such that the total monetary cost is minimized and the expected accuracy of the majority vote is greater than an expected correctness threshold α.*

3. TECHNICAL DETAILS

In this section, we introduce our algorithms and discuss their underlying idea. It is challenging to compute the *Crowd-*

Seed for the complexity of this problem. We prove the problem of *CrowdSeed Selection* is NP-hard by reducing a known NP-hard *budgeted maximum coverage* problem to it. We also show the complexity of estimating the expected diffusion of selected seed is #P-hard by reducing a known #P-hard *s-t connectness in a directed graph* problem to it. For brevity, we omit the details of the proof. As a result, we devise a greedy algorithm to find the *CrowdSeed* in order to minimize the monetary cost. To compute the expected diffusion of the found *CrowdSeed* C, we propose a sampling algorithm for estimation.

The algorithm *CrowdSeed* greedily adds one user with the highest expected diffusion to the seed C until the query diffusion from these users reaches the accuracy threshold α. We denote the query diffusion of the seed as *Diff(C)* and the underlying idea of our algorithm as follows:

While the query diffusion of seed C is less than τ users.

1. Select a user w_j who can maximize the improvement of the *query diffusion to monetary cost* ratio between $C \bigcup w_j$ and C (i.e. $arg\max_{w_j}(\frac{\text{Diff}(C \bigcup w_j)}{cost(C \bigcup w_j)} - \frac{\text{Diff}(C)}{cost(C)})$.
2. Add a selected user w_j to C and return to Step 1.

Then we utilize a sampling algorithm to estimate *Diff(C)*. Each edge $e(w_i, w_j)$ is sampled based on the probability of query diffusion (i.e. $Pr(I(w_i, w_j))$). For each sample graph $G_k(V, E_k)$, we compute the number of connected nodes from C, denoted as $d_k(C)$. After we have obtained n samples, the diffusion of selected seed C is estimated by $\overline{\text{Diff(C)}} = \frac{\sum_{k=1}^{n} d_k(C)}{n}$. Using *Hoeffding's Inequality*, we have

$$Pr(|\overline{Diff(C)} - Diff(C)| \geq \epsilon) \leq 2\exp(-\frac{2\epsilon^2 k^2}{\sum_{i=1}^{k}(|V|-1)^2}) \leq \delta \quad (8)$$

and we can achieve (ϵ, δ) approximation of estimating the *Diff(C)* if the number of samples

$$k \geq \frac{(|V|-1)^2 \ln(\frac{2}{\delta})}{2\epsilon^2 |C|^2} \quad (9)$$

where $|C|$ is the size of the selected seed. The number of samples depends on the specified error and confidence threshold values ϵ and δ, respectively. For brevity, we omit the proof where the similar proof can be found in [5].

4. CROWDSOURCED QUERY PROCESSING

We now explain the crowdsourced query processing. The microblog users can ask for a monetary reward to install *CrowdSeed*. After that, our system asks for the right to publish the crowdsourced query in the message box of the user's homepage and set a monetary account for this user. If the user is selected in the crowdseed C, the system publishes the crowdsourced query on his page and gives the required reward to his account.

Currently, we build the user graph using the recent *DBLP data*[3]. The accuracy of the user is estimated by *PageRank Algorithm*. The underlying intuition is that the users may be more reliable, if they have many publications or their publications have very high citations. The coauthors are regarded as the neighbors and the diffusion probability is set based on the similarity of two users by Equation 7.

The main workflow of our system is as follows:

[3]http://www.informatik.uni-trier.de/~ley/db/

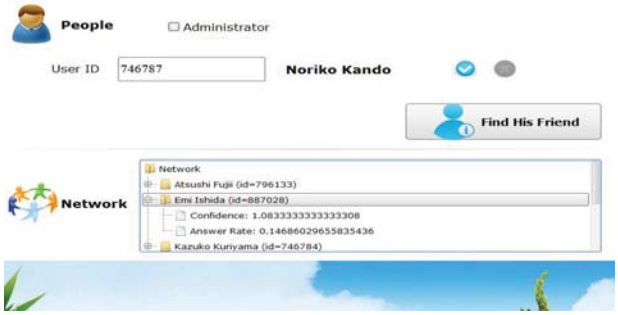

Figure 1: A Graph of Users

- First, the customer submits a query Q with the expected accuracy threshold α to our system.
- Next, the system computes seed C such that it minimizes the monetary cost.
- Then, the system publishes the query in the microblog users' message box as well as gives the required reward to their accounts.
- Finally, the system aggregates the replies and chooses the correct one.

The user graph is stored in the database of our system and only the administrator has the right to access it. We also store the estimated accuracy and query diffusion of each user. The administrator is able to search the user and his friends by clicking the "Find His Friends" button. Then the system returns the information of his friends in the text box. The interface for the administrator is given in Figure 1.

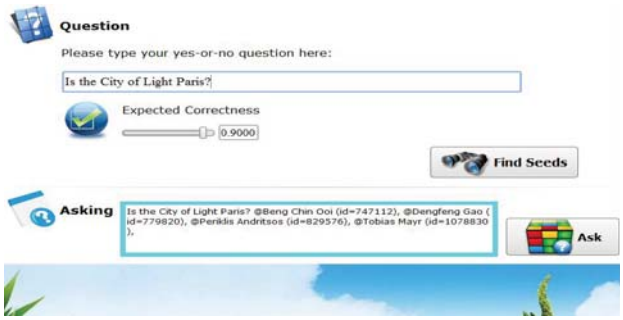

Figure 2: Game for Demo "Whom to Ask?"

Given the expected accuracy threshold α, the administrator is able to click the "Find Seeds" button to find the microblog users' ids for the seed C. Next, the system returns the selected users and lists them in the text box in the format of *@id*. Then, the administrator clicks the "Ask" button to issue the query to these users. In this example, we issue a crowdsourced query "Is the city of light Paris?" and set the expected accuracy to 0.9.

After issuing the query to seed C, the system simulates the query diffusion in *DBLP* network based on the *word of mouth* effect. Each user w_j has a probability of $Pr(I(w_i, w_j))$ to answer the query and diffuses the query to his friends.

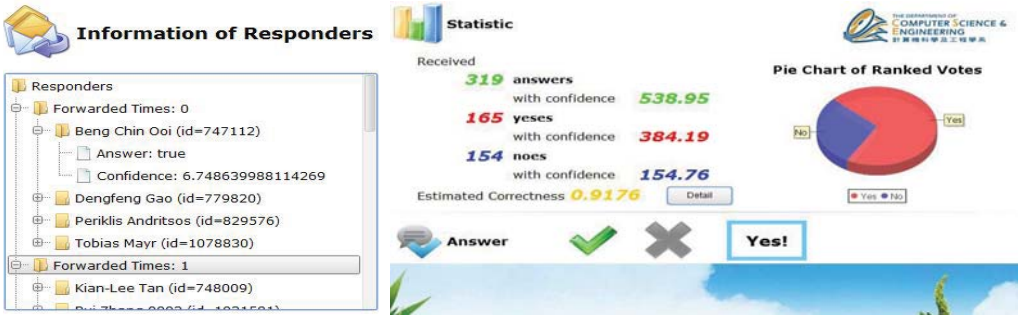

Figure 3: Tracked Responses && Answer Aggregation

The system keeps collecting the replied answers at all iterations. To avoid the latency of the users, our system progressively aggregates the user replies and chooses the correct one in the answer aggregation screen. Figure 3(a) illustrates the process of our system tracking and storing the answers at different timestamps.

We utilize the *majority voting rule* to aggregate the conflicting answers, whose expected accuracy has been proved to be greater than α. We also visualize the collected replies in a pie chart, where "true" responses are represented in red and "false" responses are blue. Figure 3(b) illustrates how our system aggregates the conflicted answers and choose the correct one.

5. DEMONSTRATION PLAN

In this demonstration, we plan to engage the attendees to participate in *CrowdSeed*. We start a game called "Who is in the *CrowdSeed*?" as shown in Figure 2.

Currently, we have used recent *DBLP data* to build a graph to model the relationships among the users. Each user is associated with two attributes: reputation and influence. The reputation(accuracy) of each user is estimated by *PageRank Algorithm*. The query diffusion between two users are estimated by the similarity of their friends (coauthors).

Our audience plays the game on the allocated laptops. In parallel, we explain how *CrowdSeed* works. Then we invite the audience to login into our system and set their expected reward (greater than zero) to activate their account. We request one of the attendees to pose a query Q and set the expected accuracy α. An example of posing a crowdsourced query is given in Figure 2. Then *CrowdSeed* finds the seed-s(i.e. users) from the participants whose accounts have been activated, to pose query Q to their home page.

The greedy *CrowdSeed* algorithm minimizes the total monetary cost of the query Q by enrolling the right users to the seed C. The users in the seed answer and diffuse the query at the first iteration. The underlying idea of enrolling the crowdseed users is that we want to select the people with high influence to answer and diffuse the query. In the meantime, we avoid selecting people of high monetary cost and low influence.

6. CONCLUSIONS

In this demo, we explore a new issue of crowdsourced query processing on mircoblogs. First, we explain the problem of *CrowdSeed* selection in microblogs. This problem is NP-hard and its computation is #P-hard. The *CrowdSeed* selection returns a set of users such that the monetary cost of a crowdsourced query is minimized and the expected accuracy is at least α. Then, we propose a greedy sampling algorithm which achieves the (ϵ, δ) approximation for the *CrowdSeed* Selection problem. Based on the algorithm, we devise a demonstration program on *DBLP* data and engage the attendees to participate our system. We devise an interactive game "Who is in the CrowdSeed" to demonstrate the effectiveness of our system.

ACKNOWLEDGEMENTS: We are grateful to the anonymous reviewers for their insightful comments on this paper. We thank the help from the UROP This work is partially supported by RGC GRF under grant number HKUST 617610 and 618509.

7. REFERENCES

[1] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri. Feedback effects between similarity and social influence in online communities. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 160–168. ACM, 2008.

[2] M. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. Crowddb: answering queries with crowdsourcing. In *SIGMOD Conference*, pages 61–72, 2011.

[3] S. Guo, A. Parameswaran, and H. Garcia-Molina. So who won?: dynamic max discovery with the crowd. In *Proceedings of the 2012 international conference on Management of Data*, pages 385–396. ACM, 2012.

[4] X. Liu, M. Lu, B. Ooi, Y. Shen, S. Wu, and M. Zhang. Cdas: a crowdsourcing data analytics system. *Proceedings of the VLDB Endowment*, 5(10):1040–1051, 2012.

[5] C. Long and R. Wong. Minimizing seed set for viral marketing. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 427–436. IEEE, 2011.

[6] A. Marcus, E. Wu, D. Karger, S. Madden, and R. Miller. Human-powered sorts and joins. *Proceedings of the VLDB Endowment*, 5(1):13–24, 2011.

[7] A. Parameswaran, H. Garcia-Molina, H. Park, N. Polyzotis, A. Ramesh, and J. Widom. Crowdscreen: Algorithms for filtering data with humans. In *Proceedings of the 2012 international conference on Management of Data*, pages 361–372. ACM, 2012.