QASCA:
A Quality-Aware Task Assignment System for Crowdsourcing Applications

Yudian Zheng*, Jiannan Wang$, Guoliang Li#, Reynold Cheng*, Jianhua Feng#

#Tsinghua University, *University of Hong Kong, $UC Berkeley

Crowdsourcing

- Crowdsourcing

Coordinate a crowd to answer questions that solve computer-hard applications.

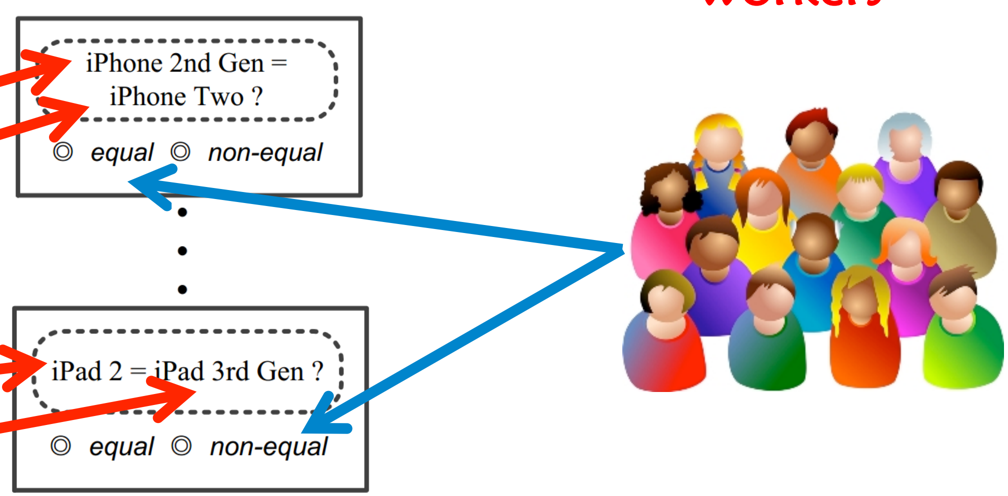
- Example

Entity Resolution Application

questions

crowd workers

ID	Object
o_1	iPhone 2nd Gen
o_2	iPhone Two
o_3	iPhone 2
o_4	iPad Two
o_5	iPad 2
o_6	iPad 3rd Gen

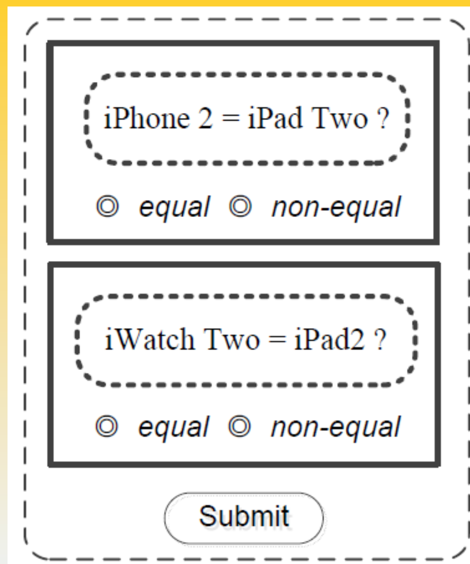


Amazon Mechanical Turk [1]

- Three Roles

- Requesters

- HIT (k questions)



- Workers

[1] https://www.mturk.com/mturk/welcome

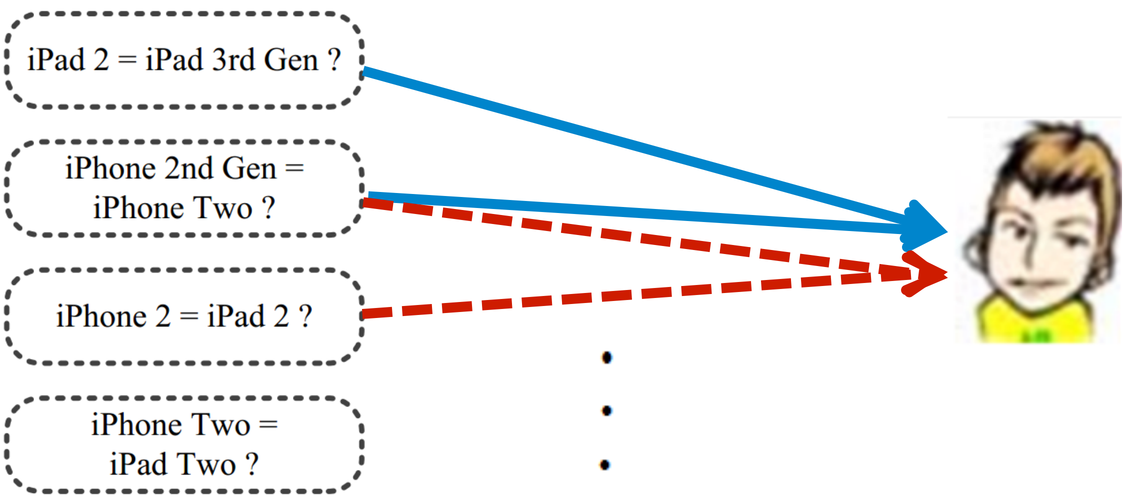
Task Assignment Problem

- Given n questions specified by a requester, when a worker comes, which k questions should be batched in a HIT and assigned to the coming worker ?

Example:

There are n=4 questions in total

A HIT contains k=2 questions.



Existing works

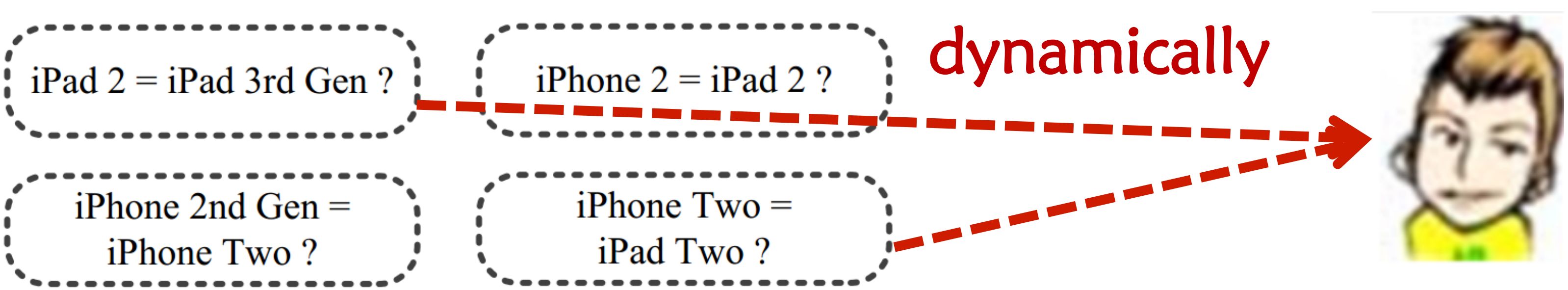
- Measure the Uncertainty of Each Question

CDAS [2] : quality-sensitive answering model

randomly assign k non-terminated questions

Askit! [3] : entropy-like method

assign the k most uncertain questions



[2] X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang. Cdas: A crowdsourcing data analytics system.PVLDB, 5(10):1040–1051, 2012.
[3] R. Boim, O. Greenshpan, T. Milo, S. Novgorodov, N. Polyzotis, and W. C. Tan. Asking the right questions in crowd data sourcing. InICDE, 2012.

Limitations of Existing works

- Miss an important factor:

How is the quality defined by an application ?

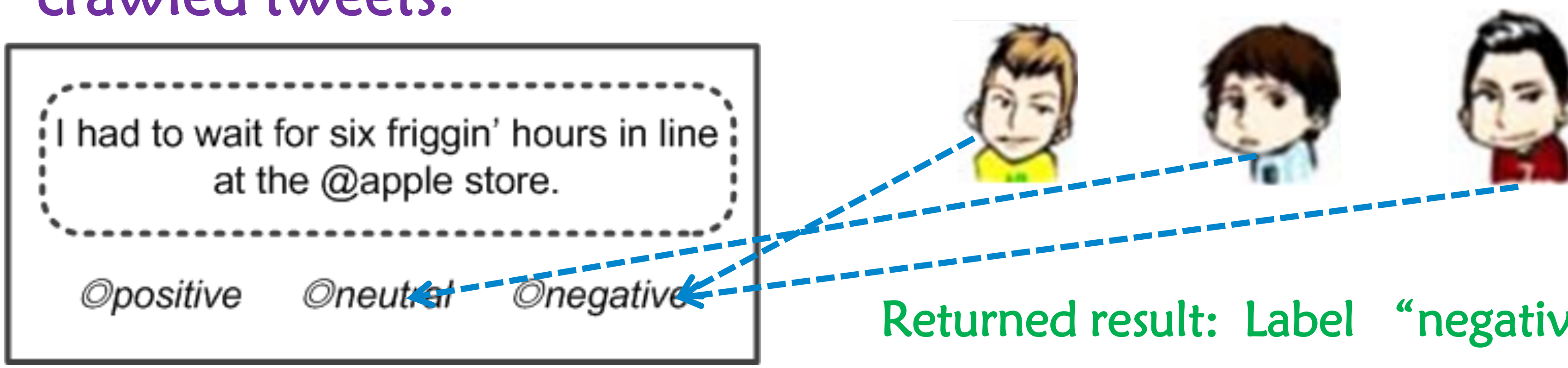
- "Evaluation Metric"

(e.g., Accuracy, F-score)

Defined by the requester

Sentiment Analysis Application

- Target: Find the sentiment (positive, neutral or negative) of crawled tweets.



Returned result: Label "negative"

- Accuracy : fraction of returned results that are correct

  [widely used in classification problems]

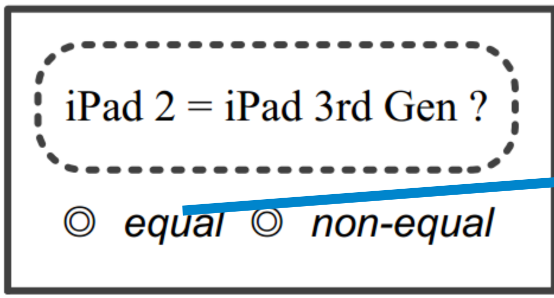
Example:

Suppose We have 100 questions, and there are 80 questions whose labels are correctly returned.

Accuracy: 80/100= 80%.

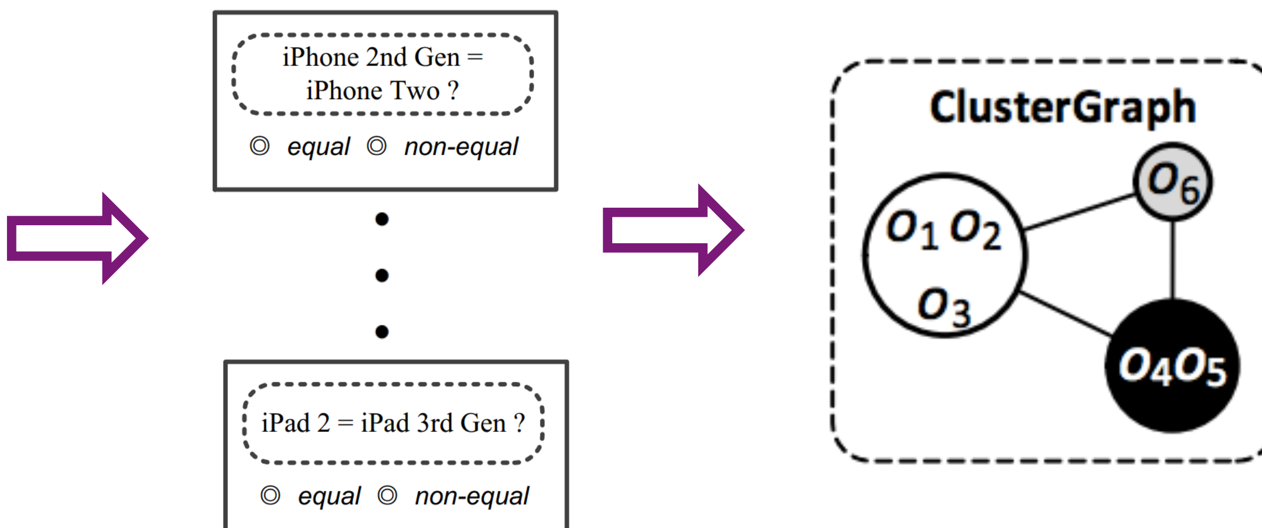
Entity Resolution Application

- **Target:** Find pairs of objects that are "equal" (referring to the same real world entity)



Focus on a specific label ("equal")

ID	Object
o_1	iPhone 2nd Gen
o_2	iPhone Two
o_3	iPhone 2
o_4	iPad Two
o_5	iPad 2
o_6	iPad 3rd Gen



Entity Resolution Application (Cont ’ d…)

- F-score : harmonic mean of Precision and Recall

(a metric that measures the quality of a specific label)

$$\text{F-score} = \frac{1}{\alpha \cdot \frac{1}{\text{Precision}} + (1-\alpha) \cdot \frac{1}{\text{Recall}}}$$

target label

controlling parameter $\alpha \in [0,1]$: trade-off Precision and Recall

Precision $\xrightarrow{\text{accurateness}}$ returned results that are target label

Recall $\xrightarrow{\text{coverage}}$ returned results that are target label

[widely used in information retrieval applications]

Target: Application's Evaluation Metric -> Assignment

- Different applications use different evaluation metrics

I want to select out "equal" pairs of objects in my generation questions !!!

- Existing works (CDAS[2], AskIt![3] etc.) do not consider the requester-specified evaluation metric in the assignment

★ Target: Requester-specified Evaluation Metric -> Assignment

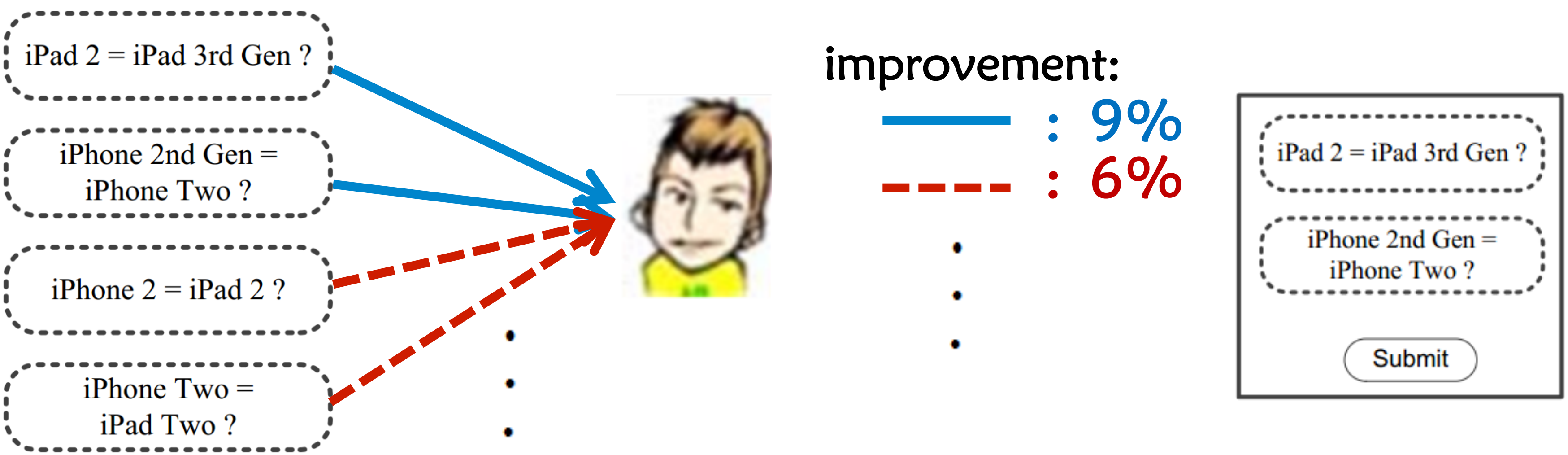
[2] X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang. Cdas: A crowdsourcing data analytics system.PVLDB, 5(10):1040–1051, 2012.
[3] R. Boim, O. Greenshpan, T. Milo, S. Novgorodov, N. Polyzotis, and W. C. Tan. Asking the right questions in crowd data sourcing. InICDE, 2012.

Solution Framework

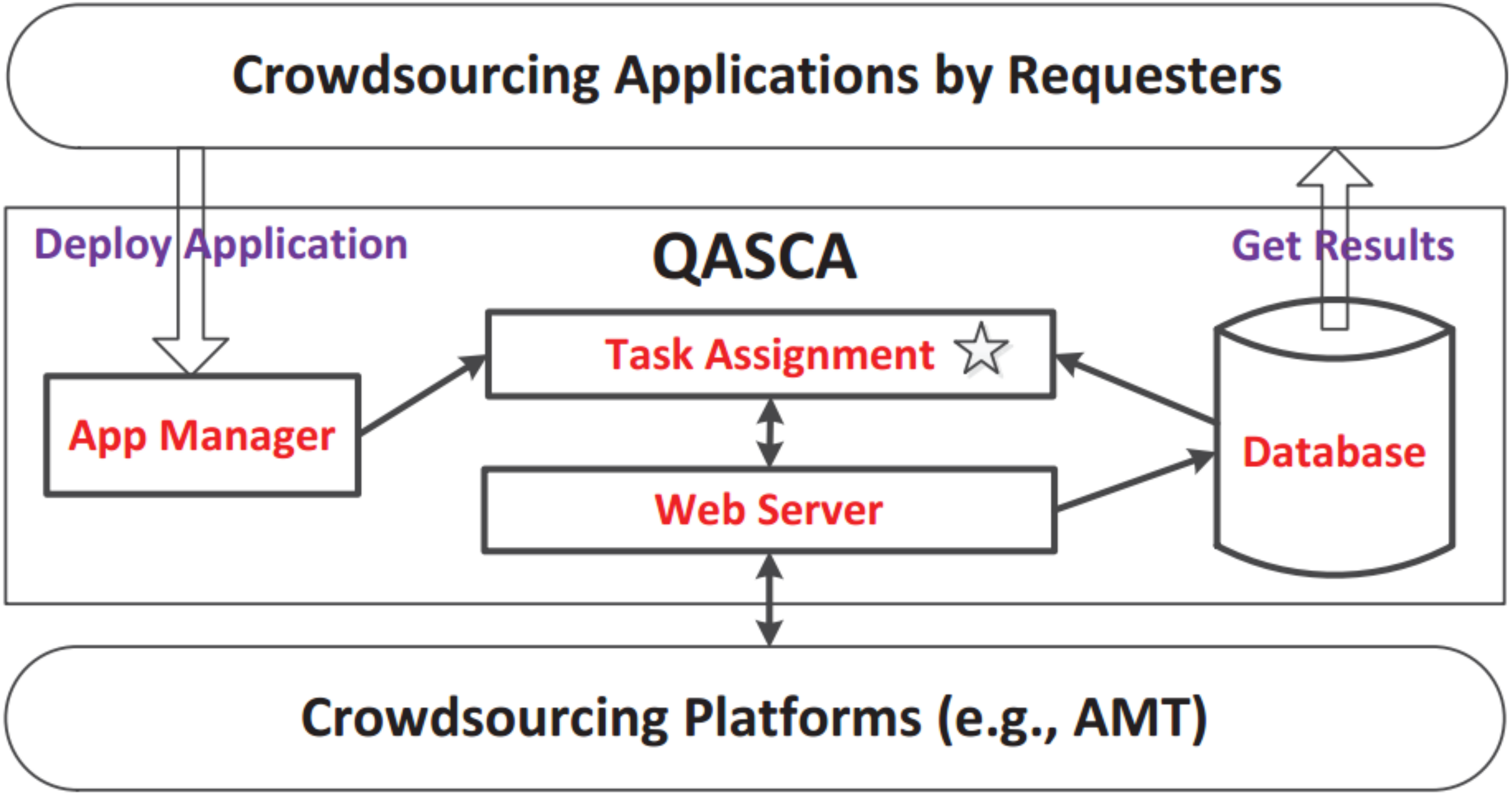
When a worker () comes,

① for each set of k questions, we will estimate the improvement of quality if the k questions are answered by worker,

② and we will select the best set of k questions that maximize the improvement to the coming worker.



QASCA System Architecture



http://i.cs.hku.hk/~ydzheng2/QASCA/

Two key challenges

① for each set of k questions, we will estimate the improvement of quality if the k questions are answered by worker,

ground truth unknown

Evaluation Metric is defined to measure the quality of returned results based on the ground truth

HOW TO ESTIMATE THE QUALITY OF RETURNED RESULTS WITH UNKNOWN GROUND TRUTH ?

② and we will select the best set of k questions that maximize the improvement to the coming worker.

expensive enumeration

The space of enumerating all assignments is exponential $\binom{n}{k}$

HOW TO EFFICIENTLY COMPUTE THE OPTIMAL ASSIGNMENT IN ALL K-QUESTION COMBINATIONS ?

Solution to the 1st challenge (Unknown Ground Truth)

iPad Two = iPad2 ?

◎ *equal* ◎ *non-equal*

ground truth is "equal" or "non-equal" (unknown)

quality: 0.8

answer with L1: equal

quality: 0.6

answer with L2: non-equal

$$[0.8 \quad 0.2]$$

The probability that the first label ("equal") to be the ground truth is 80% .

iPad Two = iPad2 ?

◎ *equal* ◎ *non-equal*

iPad 2 = iPad 3rd Gen ?

◎ *equal* ◎ *non-equal*

$$\begin{array}{cc} & \text{L1 (equal)} \quad \text{L2 (non-equal)} \\ \begin{array}{c} \text{question 1} \\ \text{question 2} \end{array} & \begin{bmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{bmatrix} \end{array}$$

Distribution matrix

Solution to the 1st challenge (Cont ' d…)

- How to evaluate the quality of results with the assistance of distribution matrix ?

$$\begin{bmatrix} \underline{0.8} & 0.2 \\ 0.4 & \underline{0.6} \end{bmatrix}$$

Suppose our returned results are (L1,L2)

ground truth: (L1,L1) Accuracy: 50% probability: 0.8 * 0.4 = 0.32

ground truth: (L1,L2) Accuracy: 100% probability: 0.8 * 0.6 = 0.48

ground truth: (L2,L1) Accuracy: 0% probability: 0.2 * 0.4 = 0.08

ground truth: (L2,L2) Accuracy: 50% probability: 0.2 * 0.6 = 0.12

$$\boxed{50\% * 0.32 + 100\% * 0.48 + 0\% * 0.08 + 50\% * 0.12 = 70\%}$$

Addressing 2 problems (1st challenge)

- Accuracy

1.Expectation:

$$\boxed{\text{Accuracy}(T,R)} = \frac{\sum_{i=1}^{n} \mathbb{1}_{\{t_i=r_i\}}}{n}. \Rightarrow \boxed{\mathbb{E}[\ \text{Accuracy}(T,R)\ ]} = \frac{\sum_{i=1}^{n} Q_{i,r_i}}{n}.$$

2.Optimal result:

Selecting the label which corresponds the highest probability

- F-score

1.Expectation:

$$\boxed{\mathbb{E}[\ \text{F-score}(T,R,\alpha)\ ]} \approx \frac{\sum_{i=1}^{n} Q_{i,1} \cdot \mathbb{1}_{\{r_i=1\}}}{\sum_{i=1}^{n} [\,\alpha \cdot \mathbb{1}_{\{r_i=1\}} + (1-\alpha) \cdot Q_{i,1}\,]}$$

2.Optimal result:

Compare the probability of the target label with some threshold

★ Solving the two problems in $O(n)$.

Cont ' d... (an interesting observation)

- For F-score, returning the label with the highest probability in each question may not be optimal

Example: Suppose the target label is the first label

$$\begin{bmatrix} 0.35 & \underline{0.65} \\ \underline{0.55} & 0.45 \end{bmatrix} \quad 48.58\% \qquad \begin{bmatrix} \underline{0.35} & 0.65 \\ \underline{0.55} & 0.45 \end{bmatrix} \quad 53.58\%$$

Solution: compare the probability of the target label with some threshold (>: target label; <=: the other label)

$$\begin{bmatrix} 0.35 & 0.65 \\ 0.55 & 0.45 \end{bmatrix} \quad 0.31 \quad \begin{matrix} 0.35 > 0.31 \\ 0.55 > 0.31 \end{matrix} \quad \begin{bmatrix} \underline{0.35} & 0.65 \\ \underline{0.55} & 0.45 \end{bmatrix}$$

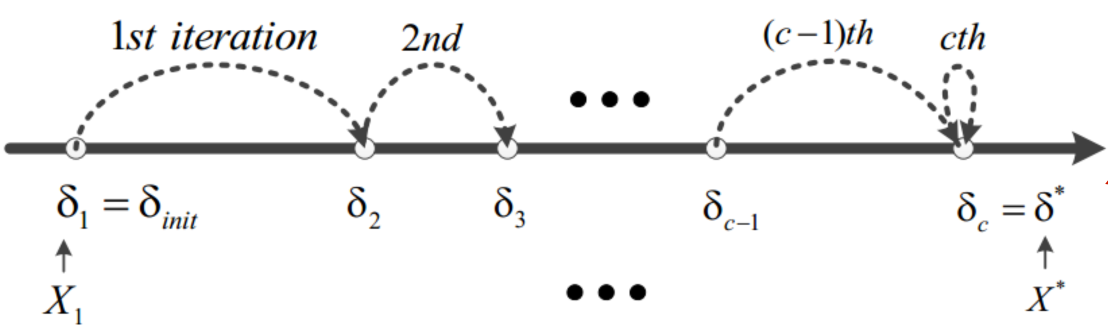
Solution to the 2nd Challenge (Optimal Assignment)

- Accuracy - TOP-K Benefit Algorithm

  Define the benefit of assigning each question

- F-score - Iterative Approach

  Local Update Algorithm



The assignment iteratively becomes better and better until convergence (optimal)

★ Reduce the complexity from $O(\binom{n}{k} \cdot n)$ to $O(n)$.

Experiments- Real Datasets (Setup-datasets)

- Five Datasets (known ground truth for evaluation)

Films Poster (FS)

- compare the publishing year

vs

Sentiment Analysis (SA)

- choose the sentiment of tweet

I had to wait for six friggin' hours in line at the @apple store.

◎positive ◎neutral ◎negative

Accuracy

Entity Resolution (ER)

- finding the same entities

iWatch Two = iPad2 ?

◎ equal ◎ non-equal

$\alpha = 0.5$

Positive Sentiment Analysis (PSA)

- positive with high confidence

Having major battery drain issue since updating iPhone 4 to iOS 5. Anyone else?

◎ positive ◎ non-positive

F-score

$\alpha = 0.75$

Negative Sentiment Analysis (NSA)

- negative as many as positive

Siri is down.

◎ negative ◎ non-negative

$\alpha = 0.25$

Experiments- Real Datasets (Setup-systems)

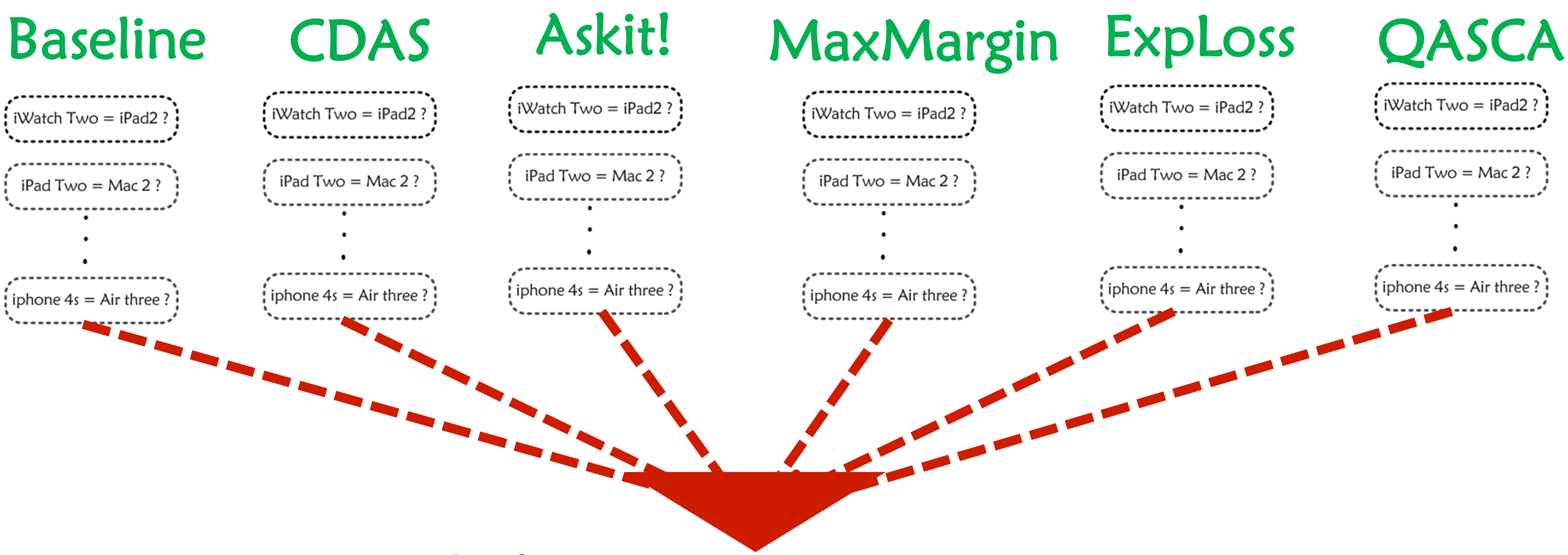
- Five Systems (End-to-End Comparison)

Baseline	randomly select k questions to assign
CDAS [2]	quality-sensitive answering model randomly assign k non-terminated questions
Askit! [3]	entropy-like method assign the k most uncertain questions
MaxMargin	iteratively select next question with the highest expected marginal improvement
ExpLoss	iteratively select the next question by considering the expected loss

[2] X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang. Cdas: A crowdsourcing data analytics system.PVLDB, 5(10):1040–1051, 2012.
[3] R. Boim, O. Greenshpan, T. Milo, S. Novgorodov, N. Polyzotis, and W. C. Tan. Asking the right questions in crowd data sourcing. InICDE, 2012.

Experiments- Real Datasets (settings)

- Parallel comparison



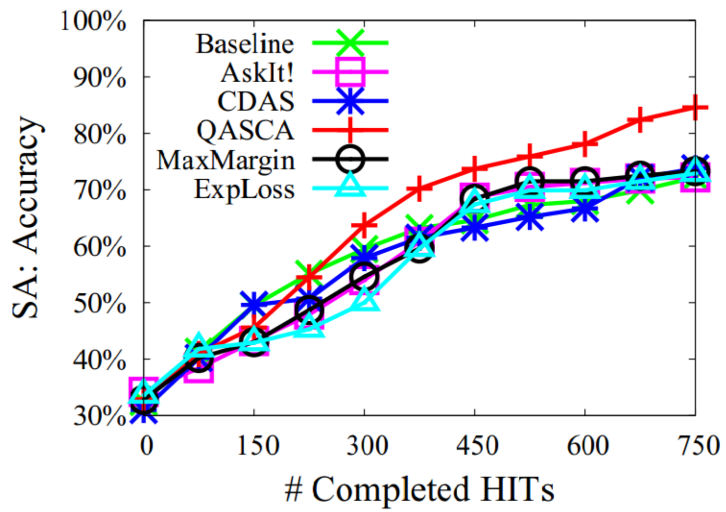
Each system assigns 4 questions

4X6=24 questions are batched in random order in a HIT

Experiments- Real Datasets (Comparison)

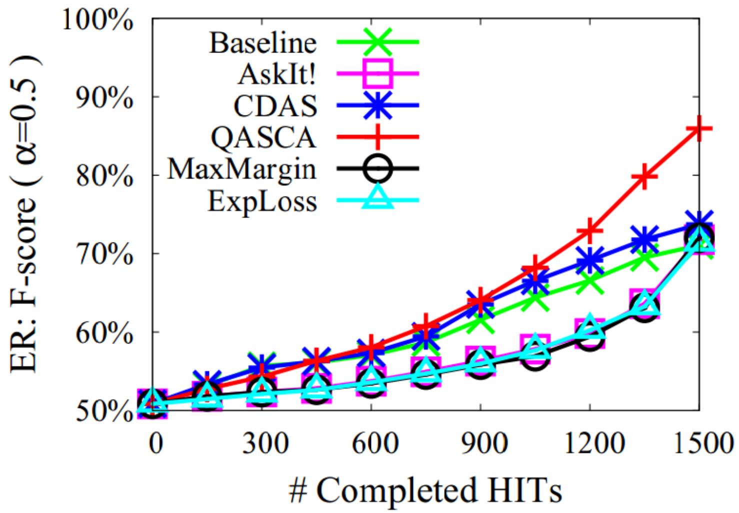
- End-to-End System Comparisons

Sentiment Analysis (SA)



SA: Accuracy

Entity Resolution (ER)



ER: F-score

QASCA outperforms other systems >8% improvement in quality when all HITs are completed

Conclusions

- Online Task Assignment Framework by considering the application-driven evaluation metrics
- Unknown Ground Truth (Distribution Matrix)

  1. Estimate the quality of returned results

  2. Optimal result of each question

- Expensive Enumeration of all assignments

  Two linear algorithms that can compute optimal assignments

- Experiments on AMT to validate our algorithms

Future Works

- Extend to more quality metrics (question-based, cluster-based etc.)

- Extend to questions of different types (heterogeneous questions)

- Consider the dependency between questions (dependency: work-flow, relations: transitive etc.)

Thank you !
Any Questions ?

Contact Info:

Yudian Zheng
ydzheng2 AT cs.hku.hk
Computer Science
The University of Hong Kong

Supplementary Slides

* 1st challenge: Definition of Accuracy -> Accuracy*

- Original Definition of F( ) : evaluation metric

F(T,R): evaluate the quality of returned results R based on the known ground truth T

For example, Accuracy: the results correctly answered 8 out of 10 questions, then 8/10=80%

T : unknown ⇒ distribution matrix Q

F(T,R) ⇒ F*(Q,R) = $\mathbb{E}$[F(T,R)]

$$\boxed{\text{Accuracy}(T,R)} = \frac{\sum_{i=1}^{n} \mathbb{1}_{\{t_i = r_i\}}}{n}.$$

⇓

$$\boxed{\text{Accuracy}^*(Q,R) = \mathbb{E}[\,\text{Accuracy}(T,R)\,]} = \frac{\sum_{i=1}^{n} Q_{i,r_i}}{n}.$$

$$\begin{bmatrix} \underline{0.8} & 0.2 \\ \underline{0.6} & 0.4 \\ \underline{0.25} & 0.75 \\ \underline{0.5} & 0.5 \\ \underline{0.9} & 0.1 \\ \underline{0.3} & 0.7 \end{bmatrix} \quad \frac{.8+.6+.25+.5+.9+.3}{6} = 55.83\%$$

* 1^{st} challenge: Maximize Accuracy*

- Given Q, what results R should be returned ?

  We want to choose the optimal R* such that

$$\boxed{R^*} = \arg\max_R \ F^*(Q, R)$$

To quantify the quality of Q,

we use the best quality that Q can reach to evaluate the quality of Q.

$$\boxed{F(Q)} = \max_R \ F^*(Q, R) = \boxed{F^*(Q, R^*)}.$$

THEOREM 1. *For Accuracy*, the optimal result r_i^* $(1 \leq i \leq n)$ of a question q_i is the label with the highest probability, i.e.,* $r_i^* = \arg\max_j \ Q_{i,j}$.

$$\begin{bmatrix} \underline{0.8} & 0.2 \\ \underline{0.6} & 0.4 \\ 0.25 & \underline{0.75} \\ \underline{0.5} & 0.5 \\ \underline{0.9} & 0.1 \\ 0.3 & \underline{0.7} \end{bmatrix}$$

optimal results

* 1st challenge: Definition of F-score -> F-score*

- F-score : harmonic mean of Precision and Recall

$$\text{F-score} = \frac{1}{\alpha \cdot \frac{1}{\text{Precision}} + (1-\alpha) \cdot \frac{1}{\text{Recall}}}$$

controlling parameters: $\alpha \in [0,1]$

focus on a target label

Expectation: hard to compute

$$\mathbb{E}[\,\text{F-score}(T,R,\alpha)\,] = \boxed{\sum_{T' \in \mathcal{T}}} \text{F-score}(T',R,\alpha) \cdot \prod_{i=1}^{n} Q_{i,t'_i}.$$

Approximation $\mathbb{E}\left[\frac{A}{B}\right] \approx \frac{\mathbb{E}[A]}{\mathbb{E}[B]} \Rightarrow \mathbb{E}\left[\frac{A}{B}\right] = \frac{\mathbb{E}[A]}{\mathbb{E}[B]} + \mathcal{O}(n^{-1})$

$$\boxed{\text{F-score}(T,R,\alpha)} = \frac{\sum_{i=1}^{n} \mathbb{1}_{\{t_i=1\}} \cdot \mathbb{1}_{\{r_i=1\}}}{\sum_{i=1}^{n}[\,\alpha \cdot \mathbb{1}_{\{r_i=1\}} + (1-\alpha) \cdot \mathbb{1}_{\{t_i=1\}}]}$$

$$\begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \\ 0.25 & 0.75 \\ 0.5 & 0.5 \\ 0.9 & 0.1 \\ 0.3 & 0.7 \end{bmatrix}$$

$\alpha = 0.5$

$$\frac{.8+.6+.25+.5+.9+.3}{.5*6+.5*(.8+.6+.25+.5+.9+.3)} = 71.66\%$$

$$\boxed{\text{F-score}^*(Q,R,\alpha) = \frac{\mathbb{E}[\,\sum_{i=1}^{n} \mathbb{1}_{\{t_i=1\}} \cdot \mathbb{1}_{\{r_i=1\}}\,]}{\mathbb{E}[\,\sum_{i=1}^{n}[\,\alpha \cdot \mathbb{1}_{\{r_i=1\}} + (1-\alpha) \cdot \mathbb{1}_{\{t_i=1\}}]\,]}} = \frac{\sum_{i=1}^{n} Q_{i,1} \cdot \mathbb{1}_{\{r_i=1\}}}{\sum_{i=1}^{n}[\,\alpha \cdot \mathbb{1}_{\{r_i=1\}} + (1-\alpha) \cdot Q_{i,1}\,]}.$$

* 1st challenge: Maximize F-score*

- (Accuracy) treat each question independently $\begin{bmatrix} 0.35 & \underline{0.65} \\ \underline{0.55} & 0.45 \end{bmatrix}$ 48.58%

✗ for F-score (even if $\mathbb{E}[\text{ F-score}(T, R, \alpha)\ ]$)

$\wedge$

Observation 1: Returning the label with the highest probability in each question may not be optimal (even for $\alpha = 0.5$);

$\begin{bmatrix} \underline{0.35} & 0.65 \\ \underline{0.55} & 0.45 \end{bmatrix}$ 53.58%

Observation 2: Deriving the optimal result of a question q_i does not only depend on the question's distribution (or Q_i) itself.

$\begin{bmatrix} 0.35 & \underline{0.65} \\ \underline{0.9} & 0.1 \end{bmatrix}$

THEOREM 2. *Given Q and α, for F-score*, the optimal result r_i^* $(1 \le i \le n)$ of a question q_i can be derived by comparing $Q_{i,1}$ with the threshold $\boxed{\theta = \lambda^* \cdot \alpha,}$ i.e., $r_i^* = 1$ if $Q_{i,1} \ge \theta$ and $r_i^* = 2$ if $Q_{i,1} < \theta$.*

$$\boxed{\lambda^* = \max_R \ \text{F-score}^*(Q, R, \alpha)}$$

global

0-1 FP

⇩

Dinkelbach

$\begin{bmatrix} \underline{0.35} & 0.65 \\ \underline{0.55} & 0.45 \end{bmatrix}$ $\lambda^* \cdot \alpha = 0.31$ $\begin{bmatrix} 0.35 & \underline{0.65} \\ \underline{0.9} & 0.1 \end{bmatrix}$ $\lambda^* \cdot \alpha = 0.4$

*1st challenge: Maximize F( )- F-score (Algorithm)

- Measure the Quality of Q for F-score ⇨ O(c * n) time

Algorithm 1 *Measure the Quality of Q for F-score*

Input: Q, α
Output: λ

$\lambda = 0$; // initialized as 0 ($\lambda_{init} = 0$)
$R' = [\,]$;
while True **do**
  $\lambda_{pre} = \lambda$; // record λ for this iteration
  // construct new $R' = [r'_1, r'_2 \ldots r'_n]$
  for $i = 1$ to n **do**
    if $Q_{i,1} \geq \lambda \cdot \alpha$ **then** $r'_i = 1$ **else** $r'_i = 2$
  $\lambda = \frac{\sum_{i=1}^{n} Q_{i,1} \cdot \mathbb{1}_{\{r'_i = 1\}}}{\sum_{i=1}^{n} [\, \alpha \cdot \mathbb{1}_{\{r'_i = 1\}} + (1-\alpha) \cdot Q_{i,1} \,]}$; // F-score* (Q, R', α)
  if $\lambda_{pre} == \lambda$ **then**
    break
  else
    $\lambda_{pre} = \lambda$
return λ

Dinkelbach Framework

*2nd Challenge: Optimal Assignments (Accuracy)

- Define the Benefic of assigning each question

$$Benefit(q_i) = Q^w_{i,r_i^w} - Q^c_{i,r_i^c}$$

Selecting k questions with largest benefits

EXAMPLE 4. *Consider Q^c and Q^w in Figure 2. We can obtain $R^c = [1,1,2,1,1,2]$ (or $[1,1,2,2,1,2]$) and $R^w = [1,1,0,1,0,2]$.*[4] *For each $q_i \in S^w$, we compute its benefit as follows: $Benefit(q_1) = Q^w_{1,r_1^w} - Q^c_{1,r_1^c} = 0.123$, $Benefit(q_2) = 0.212$, $Benefit(q_4) = 0.25$ and $Benefit(q_6) = 0.175$. So q_2 and q_4 which have the highest benefits will be assigned to worker w.*

Current Distribution Matrix

$$Q^c = \begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \\ 0.25 & 0.75 \\ 0.5 & 0.5 \\ 0.9 & 0.1 \\ 0.3 & 0.7 \end{bmatrix}$$

Estimated Distribution Matrix

$$Q^w = \begin{bmatrix} 0.923 & 0.077 \\ 0.818 & 0.182 \\ & \\ 0.75 & 0.25 \\ & \\ 0.125 & 0.875 \end{bmatrix}$$

*2nd Challenge: Optimal Assignments (F-score [1])

- F-score Online Assignment Algorithm

Algorithm 2 *F-score Online Assignment*

Input: Q^c, Q^w, α, k, S^w
Output: HIT

$\delta = 0$; // initialized as 0 ($\delta_{init} = 0$)
while True **do**
    $\delta_{pre} = \delta$
    // get the updated δ_{t+1} and its corresponding X
    $X, \delta = Update(Q^c, Q^w, \alpha, k, S^w, \delta)$ local Update
    if $\delta_{pre} == \delta$ **then**
        break
    else
        $\delta_{pre} = \delta$
// construct HIT based on the returned X
for $i = 1$ to n **do**
    if $x_i == 1$ **then**
        HIT = HIT $\cup \{q_i\}$
return HIT

*2nd Challenge: Optimal Assignments (F-score [2])

- local Update

Algorithm 3 *Update*

Input: $Q^c,\ Q^w,\ \alpha,\ k,\ S^w,\ \delta$
Output: X, λ

1: $\lambda = 0$; // initialized as 0 ($\lambda_{init} = 0$)
2: $X = [\,]$;
3: $\widehat{R}^c = [\,]$; $\widehat{R}^w = [\,]$;
4: $b = d = [0, 0, \ldots, 0]$; $\beta = 0$; $\gamma = 0$;
5: // construct $\widehat{R}^c$ ($\widehat{R}^w$) by comparing Q^c (Q^w) with $\delta \cdot \alpha$; (lines 6-9)
6: **for** $i = 1$ to n **do**
7:     **if** $Q^c_{i,1} \geq \delta \cdot \alpha$ **then** $\widehat{r}^c_i = 1$ **else** $\widehat{r}^c_i = 2$
8: **for** $q_i \in S^w$ **do**
9:     **if** $Q^w_{i,1} \geq \delta \cdot \alpha$ **then** $\widehat{r}^w_i = 1$ **else** $\widehat{r}^w_i = 2$
10: Compute b_i, d_i $(1 \leq i \leq n)$ and β, γ following the proof in Theorem 4;
11: // Update λ from λ_{init} until convergence; (line 12-21)
12: **while** True **do**
13:     $\lambda_{pre} = \lambda$
14:     compute TOP, a set which contains k questions in S^w that correspond to the highest value of $b_i - \lambda \cdot d_i$;
15:     **for** $i = 1$ to n **do**
16:         **if** $q_i \in TOP$ **then** $x_i = 1$ **else** $x_i = 0$
17:     $\lambda = \frac{\sum_{i=1}^{n}(\ x_i \cdot b_i\ ) + \beta}{\sum_{i=1}^{n}(\ x_i \cdot d_i\ ) + \gamma}$;
18:     **if** $\lambda_{pre} == \lambda$ **then**
19:         **break**
20:     **else**
21:         $\lambda_{pre} = \lambda$
22: **return** X, λ

$$
\begin{cases}
b_i = Q^w_{i,1} \cdot \mathbb{1}_{\{\widehat{r}^w_i = 1\}} - Q^c_{i,1} \cdot \mathbb{1}_{\{\widehat{r}^c_i = 1\}} \\
d_i = \alpha \cdot (\mathbb{1}_{\{\widehat{r}^w_i = 1\}} - \mathbb{1}_{\{\widehat{r}^c_i = 1\}}) + (1 - \alpha) \cdot (Q^w_{i,1} - Q^c_{i,1}) \\
\beta = \sum_{i=1}^{n} Q^c_{i,1} \cdot \mathbb{1}_{\{\widehat{r}^c_i = 1\}} \\
\gamma = \sum_{i=1}^{n} [\ \alpha \cdot \mathbb{1}_{\{\widehat{r}^c_i = 1\}} + (1 - \alpha) \cdot Q^c_{i,1}\ ],
\end{cases}
$$

Computing of Distribution Matrices

- Current Distribution Matrix

$$Q^c_{i,j} = P(t_i = j \mid D_i) = \frac{P(D_i \mid t_i = j) \cdot P(t_i = j)}{P(D_i)}.$$

quality: 0.8

answer with label 1

quality: 0.6

answer with label 2

$$Q_i^c = [0.8, 0.2]$$

- Estimated Distribution Matrix

① estimate the probability distribution that the coming worker will answer for each question

$$P(a_i^w = j' \mid D_i) = \sum_{j=1}^{\ell} P(a_i^w = j' \mid t_i = j, D_i) \cdot P(t_i = j \mid D_i).$$

quality: 0.6

$$[.8*.6+.2*.4, .8*.4+.2*.8] = [0.56, \underline{0.44}]$$

② integrate the computed distribution in computing estimated distribution matrix by weighted random sampling

$$Q^w_{i,j} \propto Q^c_{i,j} \cdot P(a_i^w = l_i^w \mid t_i = j)$$

$$(.8*.4):(.2*.6) = [.727, .273]$$

Experiments- Simulated Dataset (F-score)

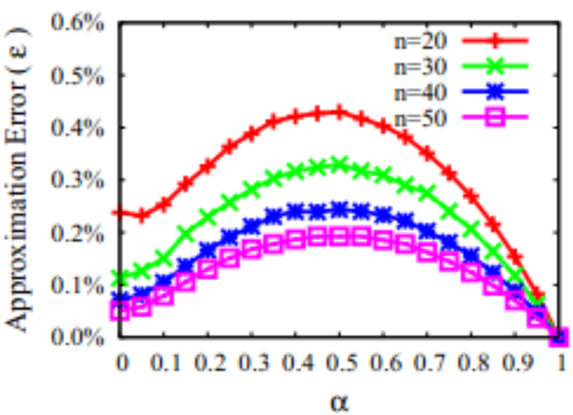
- Generation of Datasets

$$Q_{i,1} \in [0,1] \quad Q_{i,2} = 1 - Q_{i,1}$$

$$\mathbb{E}\left[\frac{A}{B}\right] \approx \frac{\mathbb{E}[A]}{\mathbb{E}[B]} \Rightarrow \mathbb{E}\left[\frac{A}{B}\right] = \frac{\mathbb{E}[A]}{\mathbb{E}[B]} + \mathcal{O}(n^{-1})$$

Approximation Error

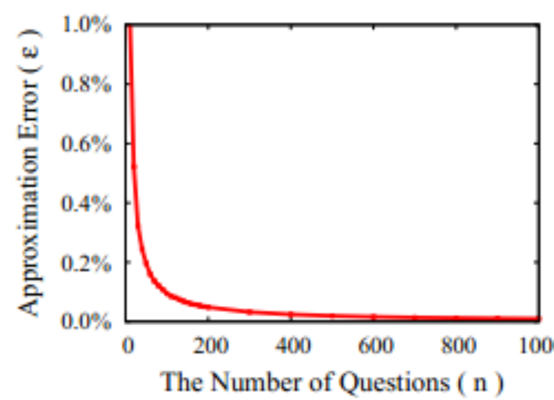
$$\epsilon = |\ \text{F-score}^*(Q,R,\alpha) - \mathbb{E}[\ \text{F-score}(T,R,\alpha)\ ]\ |$$



Varying α

$$\mathbb{E}[\ \text{Precision}(T,R)\ ] = \text{F-score}^*(Q,R,1)$$

$$\mathbb{E}[\ \text{Recall}(T,R)\ ] \approx \text{F-score}^*(Q,R,0)$$



$\mathcal{O}(n^{-1})$

Varying n

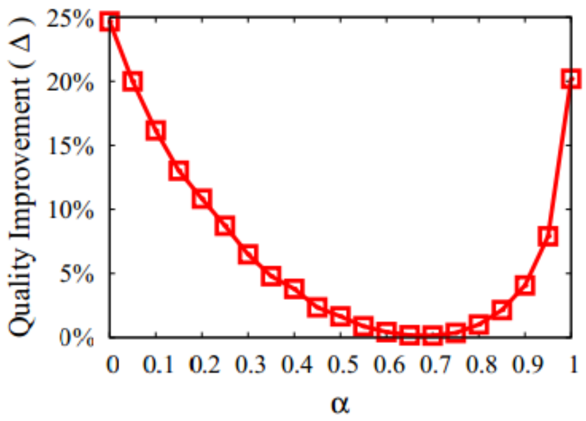
Experiments- Simulated Dataset (F-score)

- Improvement of the Optimal vs Maximal Results

Optimal Results $\boxed{R^*} = \text{argmax}_R \text{ F-score}^*(Q, R, \alpha)$

Maximal Results $\boxed{\widetilde{R}}$ $\begin{cases} \tilde{r}_i = 1 \text{ if } Q_{i,1} \geq Q_{i,2} \\ \tilde{r}_i = 2 \text{ if otherwise} \end{cases}$

$$\boxed{\Delta} = \mathbb{E}[\text{ F-score}(T, R^*, \alpha)] - \mathbb{E}[\text{ F-score}(T, \widetilde{R}, \alpha)]$$

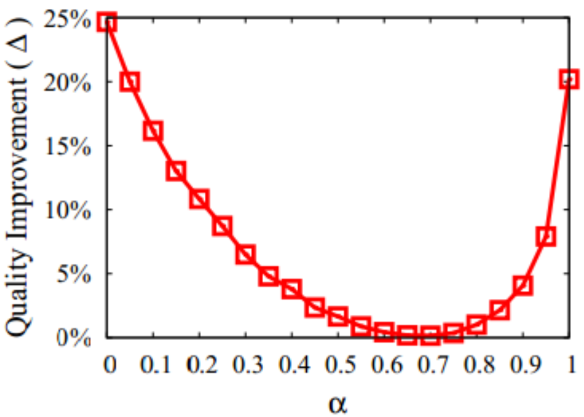


Varying α

25% α results in >10% improvement

*Explanation of a graph

- Why asymmetric ?



Δ is zero when α is around 0.65 ?

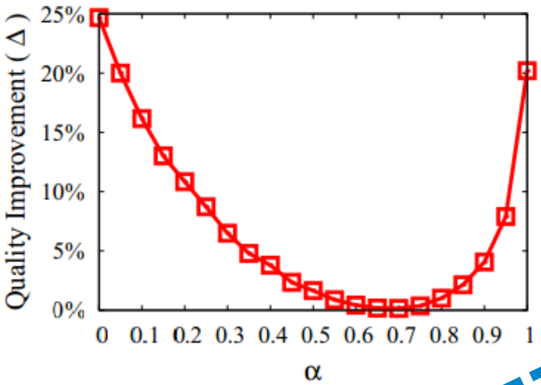
For some unknown α', if $\widetilde{R}$ is equal to R^* (or $\widetilde{R} = R^*$), (1) as $\widetilde{R}$ is constructed by comparing with the threshold 0.5, thus from Theorem 2 we know the threshold $\theta = \lambda^* \cdot \alpha' = 0.5$ and (2) as $\lambda^* = \text{F-score}^*(Q, R^*, \alpha')$, and $R^* = \widetilde{R}$, we have $\lambda^* = \frac{\sum_{i=1}^{n} \mathbb{1}_{\{Q_{i,1} \geq 0.5\}} \cdot Q_{i,1}}{\alpha' \cdot \sum_{i=1}^{n} \mathbb{1}_{\{Q_{i,1} \geq 0.5\}} + (1-\alpha') \cdot \sum_{i=1}^{n} Q_{i,1}}$. Taking $\lambda^* \cdot \alpha' = 0.5$ inside, we can obtain $\sum_{i=1}^{n} Q_{i,1} \cdot \mathbb{1}_{\{Q_{i,1} \geq 0.5\}} = 0.5 \cdot \left[\sum_{i=1}^{n} \mathbb{1}_{\{Q_{i,1} \geq 0.5\}} + (\frac{1}{\alpha'} - 1) \cdot \sum_{i=1}^{n} Q_{i,1}\right]$. Note that as we randomly generate $Q_{i,1}$ $(1 \leq i \leq n)$ for all questions, it makes $Q_{i,1}$ $(1 \leq i \leq n)$ uniformly distributed in $[0, 1]$. Thus if we take the expectation on both sides of the obtained formula, and apply the properties of uniform distribution, we can derive $0.75 \cdot \frac{n}{2} = 0.5 \cdot \left[\, \frac{n}{2} + (\frac{1}{\alpha'} - 1) \cdot 0.5 \cdot n \,\right]$, and then get $\alpha' = 0.667$, which verifies our observation (around 0.65).

Experiments- Real Datasets (F-score)*

- F-score improvements for other systems:

Other systems can all benefit from using optimal results

$$\boxed{\Delta} = \mathbb{E}[\,\text{F-score}(T, R^*, \alpha)\,] - \mathbb{E}[\,\text{F-score}(T, \widetilde{R}, \alpha)\,]$$



Simulated Datasets

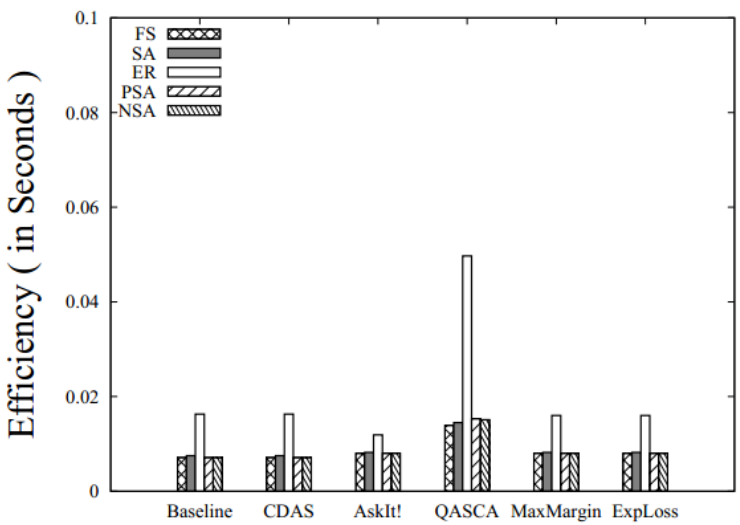
	Baseline	CDAS	AskIt!	MaxMargin	ExpLoss
ER ($\alpha = 0.5$)	2.59%	2.69%	4.56%	5.49%	4.32%
PSA ($\alpha = 0.75$)	4.14%	2.96%	1.26%	2.08%	1.66%
NSA ($\alpha = 0.25$)	14.12%	10.45%	12.44%	14.26%	9.98%

Real Datasets: average quality improvement of each system by applying our optimal R*

$$\boxed{\widehat{\Delta}} = \text{F-score}(T, R^*, \alpha) - \text{F-score}(T, \widetilde{R}, \alpha).$$

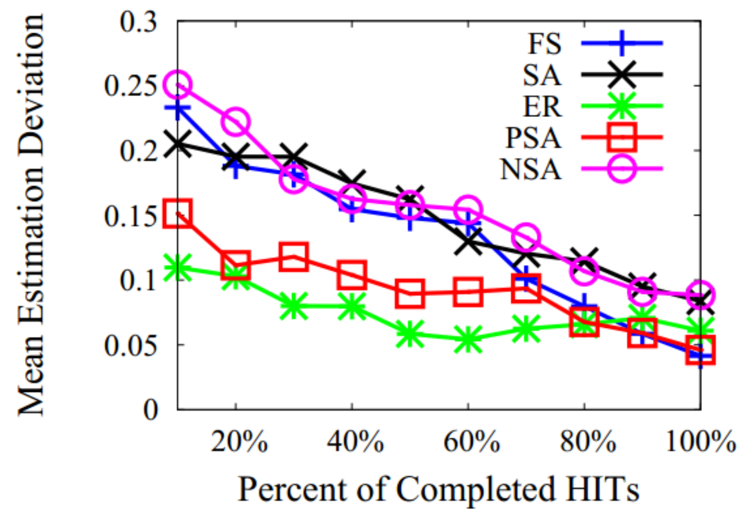
Experiments- Real Datasets (More Comparison)*

- Efficiency Comparison



(a) Efficiency

- Estimated & Real Worker Quality

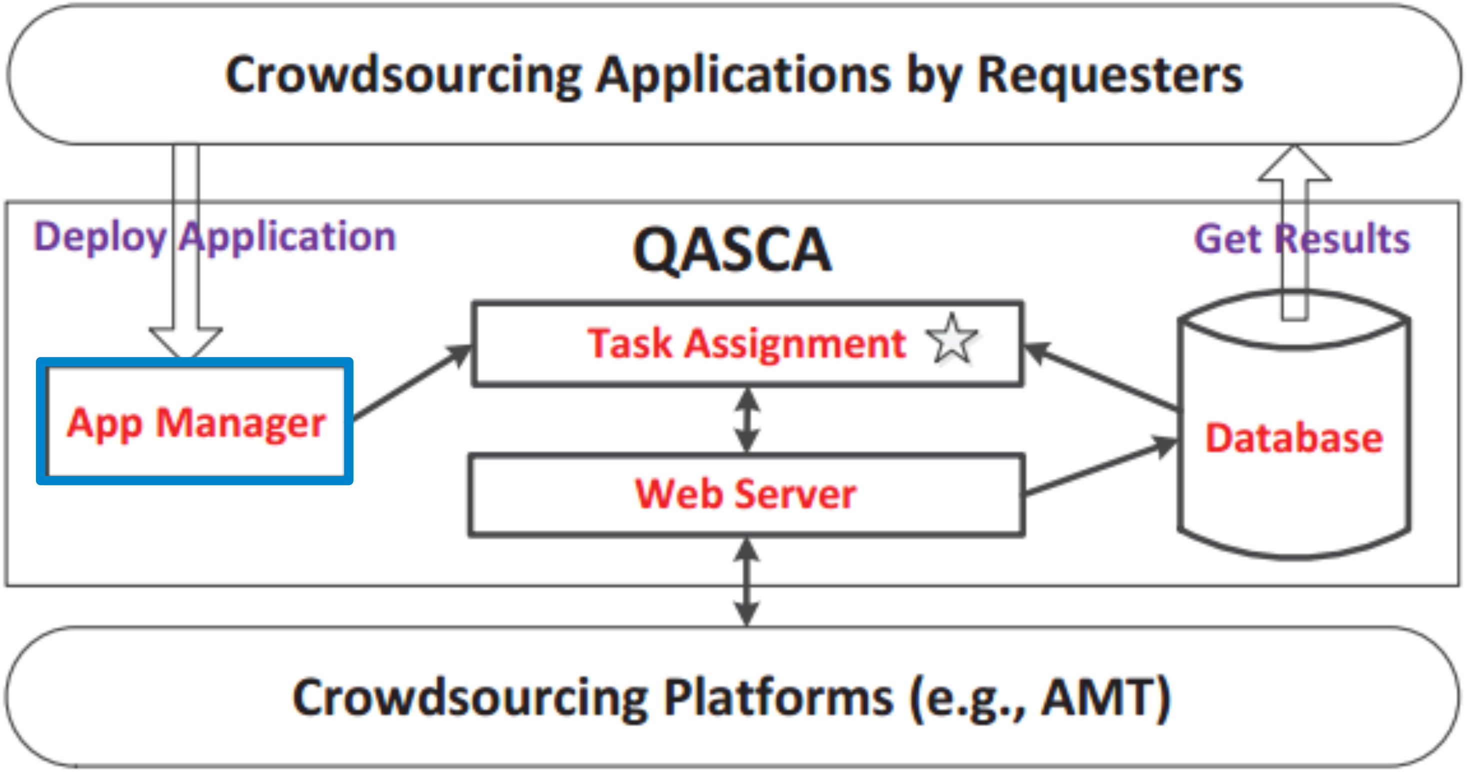


(b) Mean Estimation Deviation

worst case assignment time
All can finish within 0.06s
fairly efficiency in real situations

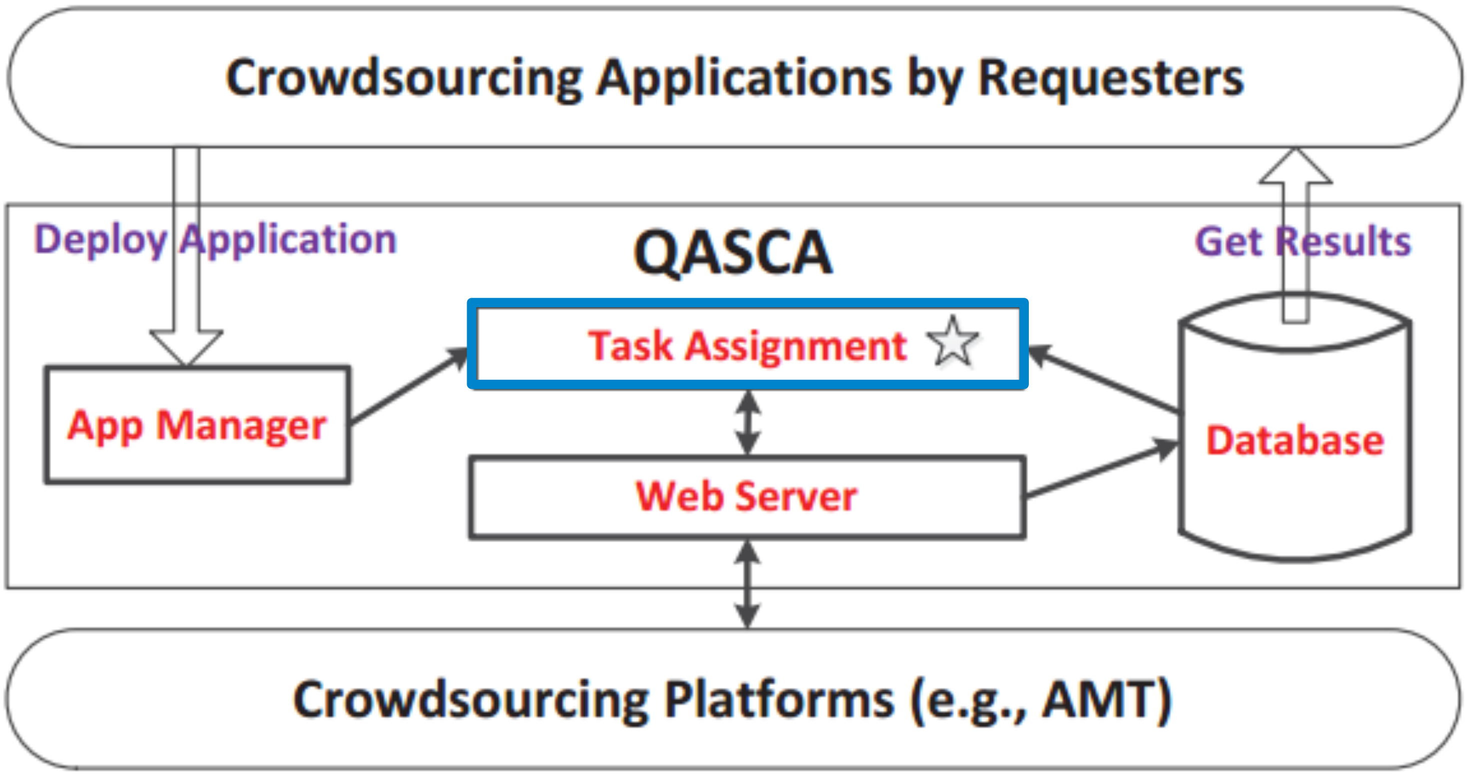
better leverage estimated worker quality to judge how the worker answer might affect the quality metric if questions are assigned

*QASCA System Architecture (1)



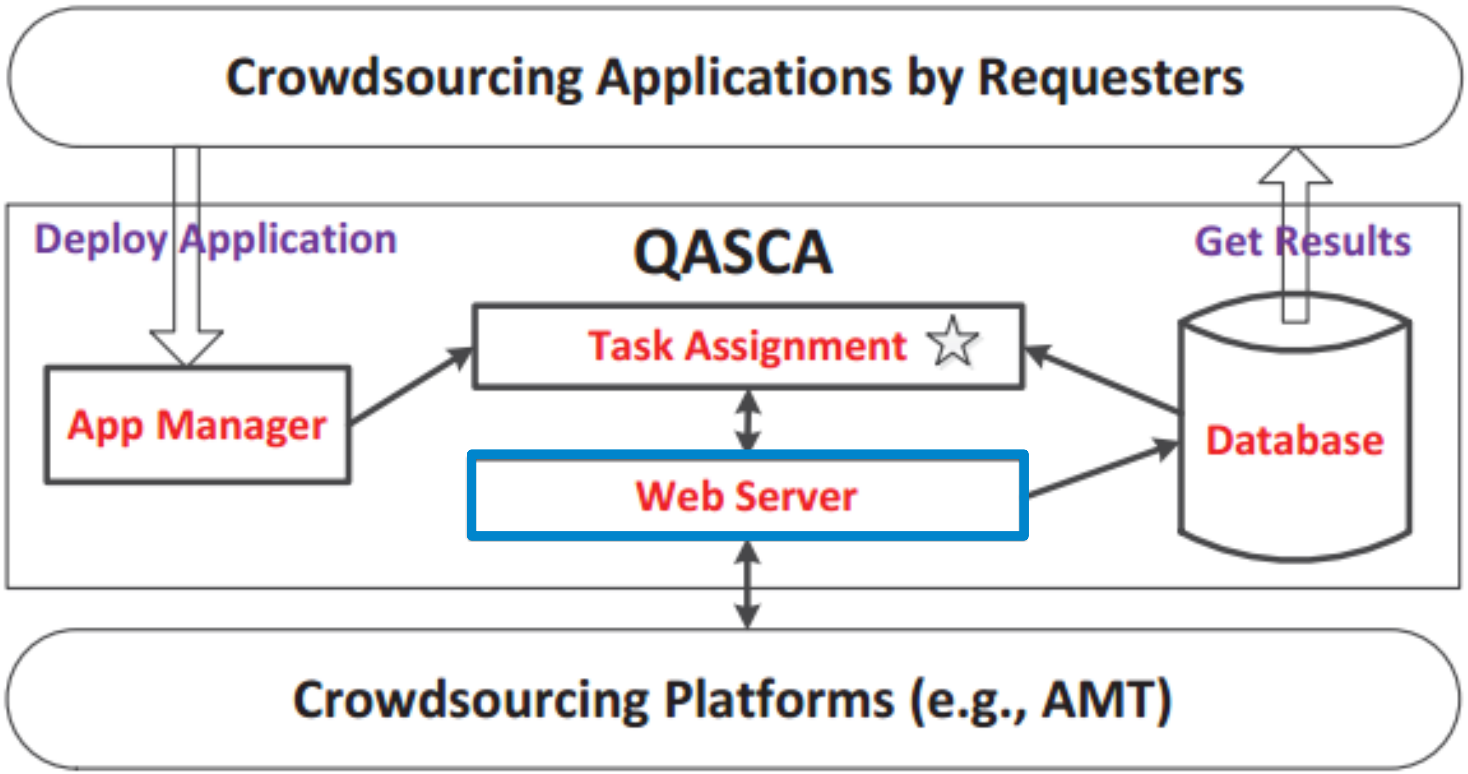
To deploy an application, the requester should set parameters in the **App Manager**. It stores the questions and other information (for example, budget, evaluation metric) required by the online assignment strategies.

*QASCA System Architecture (2)



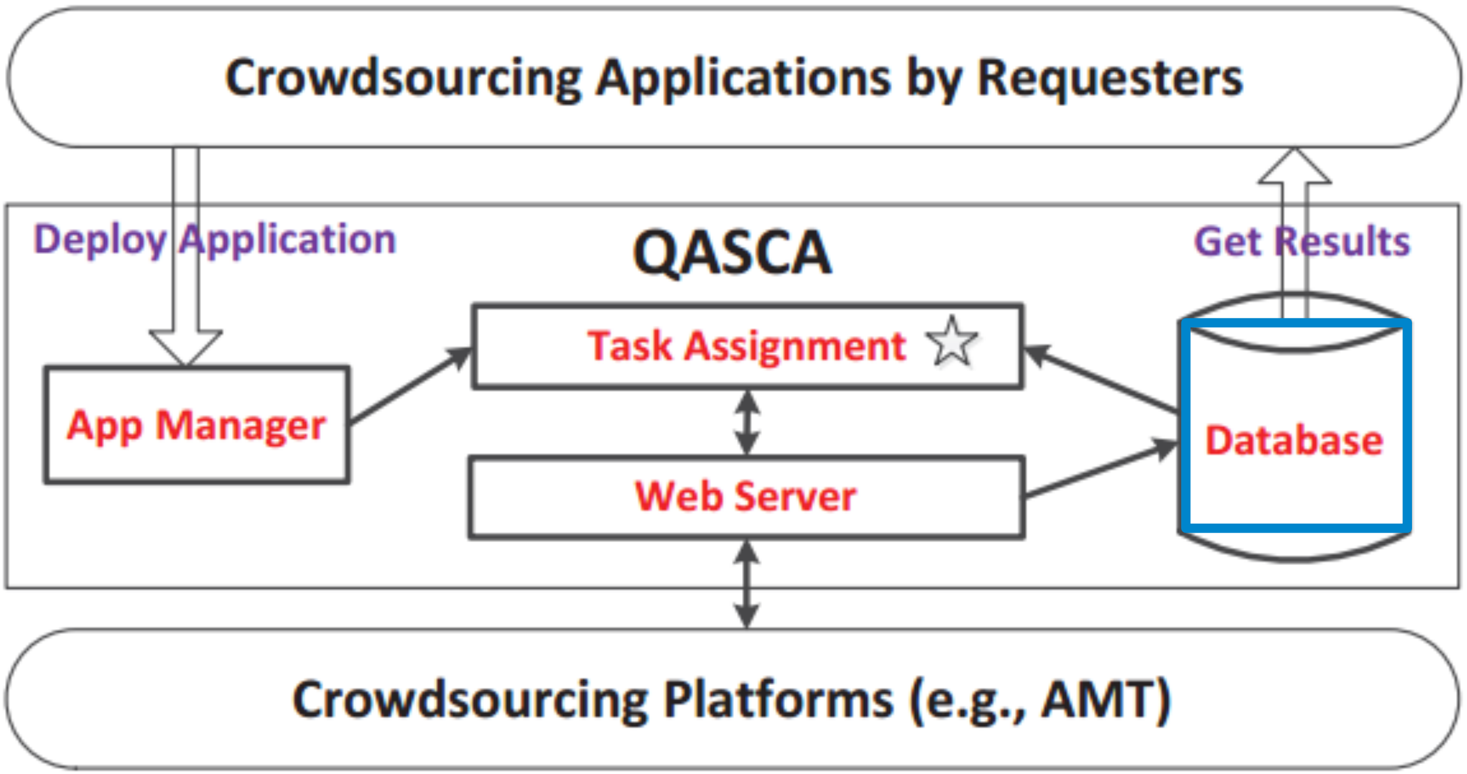
The Task Assignment runs the online assignment strategies and decides the best k questions w.r.t. the determined evaluation metric, and batch them in the HIT to assign to the coming worker.

*QASCA System Architecture (3)



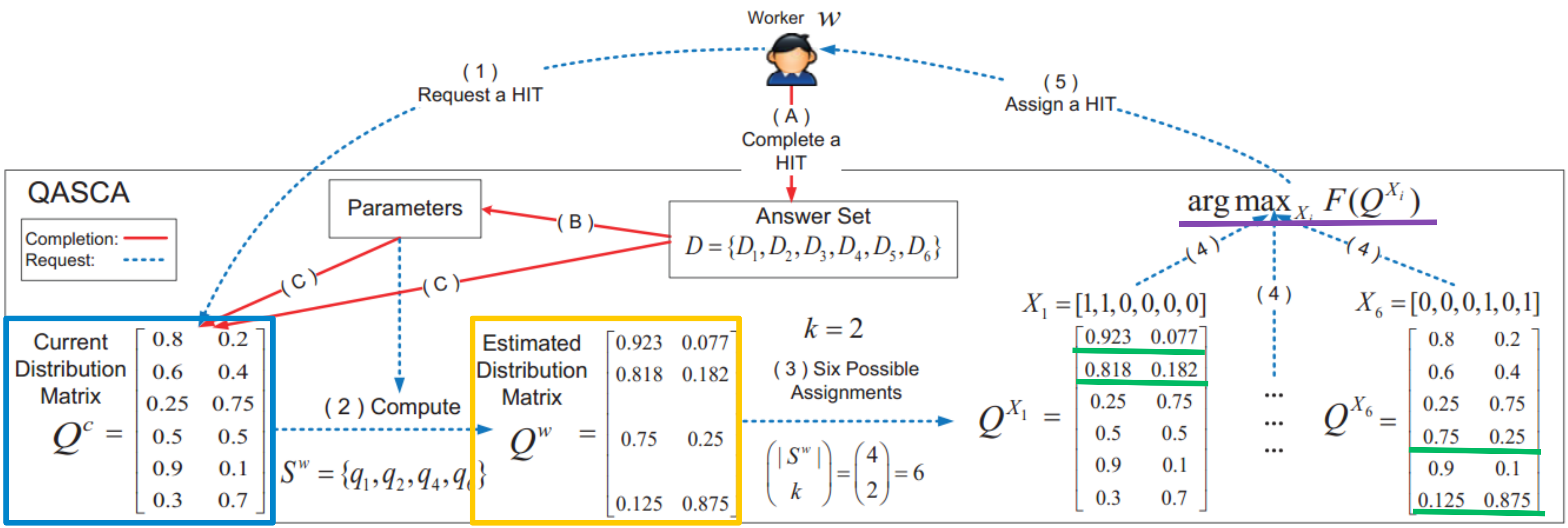
The Web Server accepts requests and give feedbacks to the workers. In HIT completion: it records the worker ID and her answers. In HIT request, it sends the HIT returned by the Task Assignment component and send it to the coming worker.

*QASCA System Architecture (4)



The **Database** stores parameters such as the workers' and questions' information. After an application has been fully accomplished, then it sends the results to the requesters.

QASCA Workflow & Problem Definition



- **Problem Definition**

DEFINITION 1. *When a worker w requests a HIT, given the current distribution matrix (Q^c), the estimated distribution matrix for the worker w (Q^w), and the function $F(\cdot)$, the problem of task assignment for the worker w is to find the optimal feasible assignment vector X^* such that $\boxed{X^* = argmax_X \ F(Q^X)}$.*

To be specific, question model

iWatch Two = iPad2 ?

iPad Two = Mac 2 ?

iphone 4s = Air three ?

iPhone 4 = iphone four ?

iPhone 3 = iphone ?

ipad 2 = ipad 2nd ?

quality: 0.8

Current Distribution Matrix

$$\begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \\ 0.25 & 0.75 \\ 0.5 & 0.5 \\ 0.9 & 0.1 \\ 0.3 & 0.7 \end{bmatrix}$$

the probability of each label to be the ground truth of the corresponding question

Estimated Distribution Matrix

$$\begin{bmatrix} 0.923 & 0.077 \\ 0.818 & 0.182 \\ & \\ 0.75 & 0.25 \\ & \\ 0.125 & 0.875 \end{bmatrix}$$

the estimated probability of each label to be the ground truth if the coming worker answers it

$$\begin{bmatrix} \boxed{0.923 \quad 0.077} \\ \boxed{0.818 \quad 0.182} \\ 0.25 \quad 0.75 \\ 0.5 \quad 0.5 \\ 0.9 \quad 0.1 \\ 0.3 \quad 0.7 \end{bmatrix}$$

Derived Matrix If we choose question 1 & 2 to assign

Target: Evaluation Metric-> assignment

I want to select out "equal" pairs of objects !!! (F-score for "equal" label)

- Consider the request-specified evaluation metric in the assignment process, that is,

When a worker () comes, we dynamically choose the best set of k questions batched in a HIT and assign it to the coming worker, by considering

(1) the coming worker 's quality,

(2) all questions ' answering information, and

(3) the specified evaluation metric