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

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

Example

Entity Resolution Application

<table>
<thead>
<tr>
<th>ID</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>O₁</td>
<td>iPhone 2nd Gen</td>
</tr>
<tr>
<td>O₂</td>
<td>iPhone Two</td>
</tr>
<tr>
<td>O₃</td>
<td>iPhone 2</td>
</tr>
<tr>
<td>O₄</td>
<td>iPad Two</td>
</tr>
<tr>
<td>O₅</td>
<td>iPad 2</td>
</tr>
<tr>
<td>O₆</td>
<td>iPad 3rd Gen</td>
</tr>
</tbody>
</table>

Questions

- iPhone 2nd Gen = iPhone Two? (equal, non-equal)
- iPad 2 = iPad 3rd Gen? (equal, non-equal)

Crowd workers
Amazon Mechanical Turk [1]

- **Three Roles**
  - **Requesters**
  - **HIT** (k questions)
  - **Workers**

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

---


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.

  ![Example](example.png)

  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)

![Diagram](image)

- **Focus on a specific label (“equal”)**
F-score : harmonic mean of Precision and Recall
(a metric that measures the quality of a specific label)

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

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

Precision

Recall

accurateness

coverage

returned results that are target label

[ widely used in information retrieval applications ]
Different applications use different evaluation metrics

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

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

Target: Requester-specified Evaluation Metric -> Assignment

---


When a worker ( ) comes,
for each set of k questions, we will estimate the improvement of quality if the k questions are answered by worker,
and we will select the best set of k questions that maximize the improvement to the coming worker.

①

②

improvement:

- : 9%
- : 6%
QASCA System Architecture

Crowdsourcing Applications by Requesters

Deploy Application

App Manager

QASCA

Task Assignment

Web Server

Database

Get Results

Crowdsourcing Platforms (e.g., AMT)

http://i.cs.hku.hk/~ydzeng2/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,

   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.

   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)

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

Distribution matrix

question 1

question 2

L1 (equal)  L2 (non-equal)

0.8  0.2

0.4  0.6
Solution to the 1st challenge (Cont’d…)

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

  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

  \[
  50\% \times 0.32 + 100\% \times 0.48 + 0\% \times 0.08 + 50\% \times 0.12 = 70\%
  \]

  I want to select out the **optimal result of each question!!!**
Addressing 2 problems (1st challenge)

- **Accuracy**
  1. Expectation:
     \[
     \text{Accuracy}(T, R) = \frac{\sum_{i=1}^{n} 1_{\{t_i = r_i\}}}{n}
     \]
     \[
     \mathbb{E}[\text{Accuracy}(T, R)] = \frac{\sum_{i=1}^{n} Q_{i,r_i}}{n}
     \]
  2. Optimal result:
     Selecting the label which corresponds the highest probability

- **F-score**
  1. Expectation:
     \[
     \mathbb{E}[\text{F-score}(T, R, \alpha)] \approx \frac{\sum_{i=1}^{n} Q_{i,1} \cdot 1_{\{r_i = 1\}}}{\sum_{i=1}^{n} [\alpha \cdot 1_{\{r_i = 1\}} + (1 - \alpha) \cdot Q_{i,1}]} \]
  2. Optimal result:
     Compare the probability of the target label with some threshold

⭐ Solving the two problems in $O(n)$. 
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.

\[
\begin{bmatrix}
0.35 & 0.65 \\
0.55 & 0.45
\end{bmatrix}
\]

48.58%  \quad \begin{bmatrix}
0.35 & 0.65 \\
0.55 & 0.45
\end{bmatrix}

53.58%

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

\[
\begin{bmatrix}
0.35 & 0.65 \\
0.55 & 0.45
\end{bmatrix}
\]

0.31

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

0.55 > 0.31
Solution to the 2\textsuperscript{nd} 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 $O\left(\binom{n}{k} \cdot n\right)$ to $O(n)$.
Experiments- Real Datasets (Setup-datasets)

- Five Datasets (known ground truth for evaluation)
  - Films Poster (FS)
    - compare the publishing year
  - Sentiment Analysis (SA)
    - choose the sentiment of tweet
  - Entity Resolution (ER)
    - finding the same entities
  - Positive Sentiment Analysis (PSA)
    - positive with high confidence
  - Negative Sentiment Analysis (NSA)
    - negative as many as positive
## Five Systems (End-to-End Comparison)

<table>
<thead>
<tr>
<th>System</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Baseline</strong></td>
<td>randomly select k questions to assign</td>
</tr>
<tr>
<td>CDAS [2]</td>
<td>quality-sensitive answering model</td>
</tr>
<tr>
<td></td>
<td>randomly assign k non-terminated questions</td>
</tr>
<tr>
<td></td>
<td>assign the k most uncertain questions</td>
</tr>
<tr>
<td>MaxMargin</td>
<td>iteratively select next question with the highest expected marginal improvement</td>
</tr>
<tr>
<td>ExpLoss</td>
<td>iteratively select the next question by considering the expected loss</td>
</tr>
</tbody>
</table>

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)**

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

  **Entity Resolution (ER)**

  - 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 \(\rightarrow\) Accuracy*

- **Original Definition of** \(F()\): evaluation metric

\[ F(T, R) : \text{evaluate the quality of returned results } R \text{ based on the known ground truth } T \]

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

\[ T : \text{unknown} \quad \times \quad \Rightarrow \quad \text{distribution matrix } Q \quad \checkmark \]

\[ F(T, R) \]

\[ \text{Accuracy}(T, R) = \frac{\sum_{i=1}^{n} 1\{t_i=r_i\}}{n} \]

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

\[
\begin{bmatrix}
0.8 & 0.2 \\
0.6 & 0.4 \\
0.25 & 0.75 \\
0.5 & 0.5 \\
0.9 & 0.1 \\
0.3 & 0.7 \\
\end{bmatrix} = \frac{0.8 + 0.6 + 0.25 + 0.5 + 0.9 + 0.3}{6} = 55.83\%
\]
Given $Q$, what results $R$ should be returned?

We want to choose the optimal $R^*$ such that

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

To quantify the quality of $Q$, we use the best quality that $Q$ can reach to evaluate the quality of $Q$.

$$F(Q) = \max_R F^*(Q, R) = F^*(Q, R^*)$$

*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.,

$$r_i^* = \arg \max_j Q_{i,j}$$
* 1\textsuperscript{st} challenge: Definition of F-score -> F-score*

- **F-score**: harmonic mean of Precision and Recall

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

**Controlling parameters**: \( \alpha \in [0, 1] \)

**Focus on a target label**

**Expectation**: hard to compute \( \times \)

**Approximation**

\[
\mathbb{E}\left[ \frac{A}{B} \right] \approx \frac{\mathbb{E}[A]}{\mathbb{E}[B]}
\]

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

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

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

**Example**

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

\( \alpha = 0.5 \)

\[
\frac{0.8 + 0.6 + 0.25 + 0.5 + 0.9 + 0.3}{5 + 0.5 (0.8 + 0.6 + 0.25 + 0.5 + 0.9 + 0.3)} = 71.66\%
\]
1st challenge: Maximize F-score*

(Accuracy) treat each question independently

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

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

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

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

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

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

$\begin{bmatrix} 0.35 & 0.65 \\ 0.9 & 0.1 \end{bmatrix}$ $\lambda^* \cdot \alpha = 0.4$
**1st challenge: Maximize F( )- F-score (Algorithm)**

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

```
Algorithm 1: Measure the Quality of Q for F-score

Input: Q, α
Output: λ

1. \( \lambda = 0 \); // initialized as 0 (\( \lambda_{init} = 0 \))
2. \( R' = [ ] \);
3. while True do
4.   \( \lambda_{pre} = \lambda \); // record \( \lambda \) for this iteration
5.   // construct new \( R' = [r'_1, r'_2 \ldots r'_n] \)
6.   for i = 1 to n do
7.     if \( Q_{i,1} \geq \lambda \cdot α \) then \( r'_i = 1 \) else \( r'_i = 2 \)
8.     \( \lambda = \frac{\sum_{i=1}^{n} Q_{i,1} \cdot 1_{\{r'_i=1\}}}{\sum_{i=1}^{n} (\alpha \cdot 1_{\{r'_i=1\}} + (1-\alpha) \cdot Q_{i,1})} \); // F-score*(Q, R', α)
9.   if \( \lambda_{pre} == \lambda \) then
10.      break
11.    else
12.      \( \lambda_{pre} = \lambda \)
13. return \( \lambda \)
```
Define the Benefic of assigning each question

\[ \text{Benefit}(q_i) = Q^w_{i,r_i^w} - Q^c_{i,r_i^c} \]

Selecting \( k \) questions with largest benefits

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

- F-score Online Assignment Algorithm

```plaintext
Algorithm 2 F-score Online Assignment

Input: \( Q^c, Q^w, \alpha, k, S^w \)
Output: HIT

1: \( \delta = 0 \); // initialized as 0 (\( \delta_{init} = 0 \))
2: while True do
3: \( \delta_{pre} = \delta \)
4: // get the updated \( \delta_{t+1} \) and its corresponding \( X \)
5: \( X, \delta = Update(Q^c, Q^w, \alpha, k, S^w, \delta) \)
6: if \( \delta_{pre} == \delta \) then
7: break
8: else
9: \( \delta_{pre} = \delta \)
10: // construct HIT based on the returned \( X \)
11: for \( i = 1 \) to \( n \) do
12: if \( x_i == 1 \) then
13: \( \text{HIT} = \text{HIT} \cup \{q_i\} \)
14: return \( \text{HIT} \)
```

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

- **local Update**

Algorithm 3 Update

<table>
<thead>
<tr>
<th>Input:</th>
<th>$Q^c$, $Q^w$, $\alpha$, $k$, $S^w$, $\delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output:</td>
<td>$X$, $\lambda$</td>
</tr>
</tbody>
</table>

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

\[
\begin{align*}
    b_i &= Q_{i,1}^w \cdot 1_{\{\tilde{r}_i^w = 1\}} - Q_{i,1}^c \cdot 1_{\{\tilde{r}_i^c = 1\}} \\
    d_i &= \alpha \cdot (1_{\{\tilde{r}_i^w = 1\}} - 1_{\{\tilde{r}_i^c = 1\}}) + (1 - \alpha) \cdot (Q_{i,1}^w - Q_{i,1}^c) \\
    \beta &= \sum_{i=1}^n Q_{i,1}^c \cdot 1_{\{\tilde{r}_i^c = 1\}} \\
    \gamma &= \sum_{i=1}^n [\alpha \cdot 1_{\{\tilde{r}_i^c = 1\}} + (1 - \alpha) \cdot Q_{i,1}^c],
\end{align*}
\]
Computing of Distribution Matrices

- **Current Distribution Matrix**

$$Q_{i,j}^C = P(t_i = j | D_i) = \frac{P(D_i | t_i = j) \cdot P(t_i = j)}{P(D_i)}.$$ 

- **Estimated Distribution Matrix**

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

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

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

$$Q_{i,j}^w \propto Q_{i,j}^C \cdot P(a_i^w = l_i^w | t_i = j).$$
Experiments- Simulated Dataset (F-score)

- **Generation of Datasets**

\[
E \left[ \frac{A}{B} \right] \approx \frac{E[A]}{E[B]} \quad \iff \quad E \left[ \frac{A}{B} \right] = \frac{E[A]}{E[B]} + O(n^{-1})
\]

**Approximation Error**

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

- Varying \( \alpha \)
  - \( E[ \text{Precision}(T, R) ] = F\text{-score}^*(Q, R, 1) \)
  - \( E[ \text{Recall}(T, R) ] \approx F\text{-score}^*(Q, R, 0) \)

- Varying \( n \)
  - \( O(n^{-1}) \)

\( Q_{i,1} \in [0, 1] \quad Q_{i,2} = 1 - Q_{i,1} \)
Experiments - Simulated Dataset (F-score)

- Improvement of the Optimal vs Maximal Results

Optimal Results \( R^* = \arg\max_R \ F\text{-score}^*(Q, R, \alpha) \)

Maximal Results \( \tilde{R} \)

\[
\begin{align*}
\tilde{r}_i &= 1 \text{ if } Q_{i,1} \geq Q_{i,2} \\
\tilde{r}_i &= 2 \text{ if otherwise}
\end{align*}
\]

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

Varying \( \alpha \) results in >10% improvement
Explanation of a graph

- Why asymmetric?

For some unknown $\alpha'$, if $\tilde{R}$ is equal to $R^*$ (or $\tilde{R} = R^*$), (1) as $\tilde{R}$ 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 $R^* = \tilde{R}$, we have

$$\lambda^* = \frac{\sum_{i=1}^{n} 1\{Q_{i,1} \geq 0.5\} \cdot Q_{i,1}}{\alpha' \cdot \sum_{i=1}^{n} 1\{Q_{i,1} \geq 0.5\} + (1-\alpha') \cdot \sum_{i=1}^{n} Q_{i,1}}.$$ 

Taking $\lambda^* \cdot \alpha' = 0.5$ inside, we can obtain $\sum_{i=1}^{n} Q_{i,1} \cdot 1\{Q_{i,1} \geq 0.5\} = 0.5 \cdot \left[ \sum_{i=1}^{n} 1\{Q_{i,1} \geq 0.5\} + \left( \frac{1}{\alpha'} - 1 \right) \cdot \sum_{i=1}^{n} Q_{i,1} \right]$. 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 $0.75 \cdot \frac{n}{2} = 0.5 \cdot \left[ \frac{n}{2} + \left( \frac{1}{\alpha'} - 1 \right) \cdot 0.5 \cdot n \right]$, 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

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

Simulated Datasets

Real Datasets: average quality improvement of each system by applying our optimal \( R^* \)

\[
\hat{\Delta} = \text{F-score}(T, R^*, \alpha) - \text{F-score}(T, \tilde{R}, \alpha).
\]
Experiments - Real Datasets (More Comparison)*

- **Efficiency Comparison**
  - 
  - Estimated & Real Worker Quality
  - 

- **Efficiency**
  - Worst case assignment time
  - All can finish within 0.06s
  - Fairly efficiency in real situations

- **Mean Estimation Deviation**
  - Better leverage estimated worker quality to judge how the worker answer might affect the quality metric if questions are assigned
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.
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.
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.
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.
**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 $X^* = \arg\max_X F(Q^X)$. 

\[
Q^c = \begin{bmatrix}
0.8 & 0.2 \\
0.6 & 0.4 \\
0.25 & 0.75 \\
0.5 & 0.5 \\
0.9 & 0.1 \\
0.3 & 0.7
\end{bmatrix}, \\
Q^w = \begin{bmatrix}
0.923 & 0.077 \\
0.818 & 0.182 \\
0.75 & 0.25 \\
0.125 & 0.875
\end{bmatrix}
\]

\[
S^w = \{q_1, q_2, q_4, q_6\}
\]

\[
k = 2
\]

\[
D = \{D_1, D_2, D_3, D_4, D_5, D_6\}
\]

\[
X_1 = [1, 1, 0, 0, 0, 0]
\]

\[
X_6 = [0, 0, 0, 1, 0, 1]
\]

\[
\arg\max_{X} F(Q^X)
\]
To be specific, question model

**Current Distribution Matrix**

<table>
<thead>
<tr>
<th></th>
<th>Watch Two = iPad2?</th>
<th>iPad Two = Mac2?</th>
<th>iPhone 4s = Air three?</th>
<th>iPhone 4 = iPhone four?</th>
<th>iPhone 3 = iPhone?</th>
<th>iPad 2 = iPad 2nd?</th>
</tr>
</thead>
<tbody>
<tr>
<td>quality:</td>
<td>0.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>probability of each label to be the ground truth of the corresponding question</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>0.8</th>
<th>0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>0.7</td>
<td></td>
</tr>
</tbody>
</table>

**Estimated Distribution Matrix**

<table>
<thead>
<tr>
<th></th>
<th>0.923</th>
<th>0.077</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.818</td>
<td>0.182</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>0.7</td>
<td></td>
</tr>
</tbody>
</table>

**Derived Matrix**

If we choose question 1 & 2 to assign
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.

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