CrowdDB

Answering Queries with Crowdsourcing

Limitations of traditional RDBMS

- Missing data
- Fuzzy Comparisons

SELECT url FROM university WHERE name = "S.F.U";

Close world assumption

Completeness

- **Missing** a record for S.F.U

Correctness

- **Incorrect** record : S.F.W.

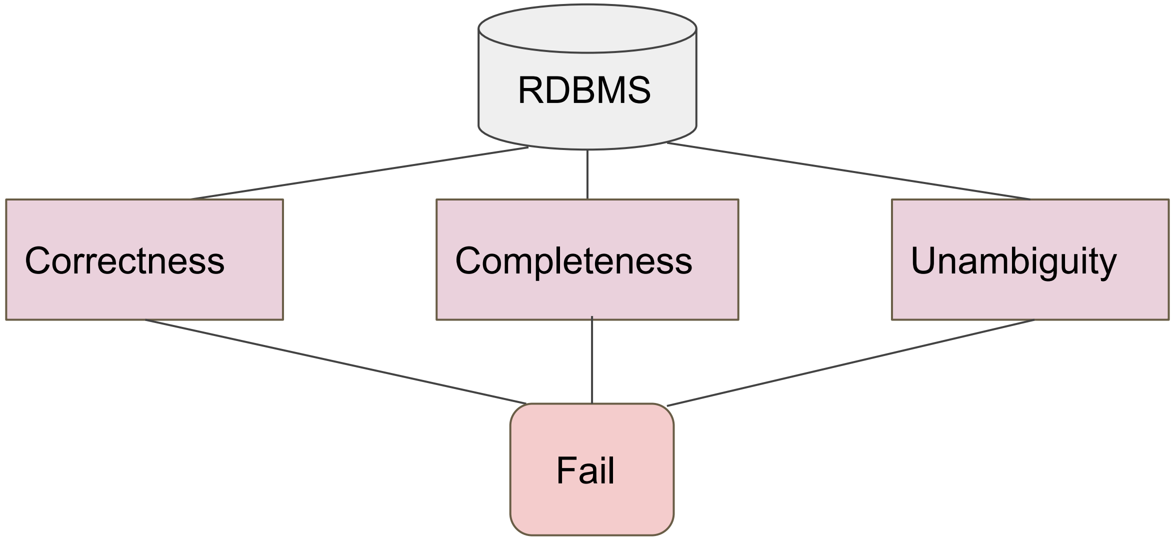
Unambiguity

- **Entity resolution** : Multiple ways to refer to the same real-world entity

  Simon Fraser University
  Simon Fraser U

Relational database systems

Problem: Closed world assumption



SELECT image FROM picture WHERE topic=”SFU Campus” ORDER BY relevance LIMIT 1 ;

Fuzzy Comparisons

- Obtain and store relevance of pictures beforehand

Limitations of traditional RDBMS

- Finding new data
- Comparing data

Can be easily answered by people but not RDBMS

. Is it possible to leverage human resources to extend the capabilities of relational databases

CrowdDB

- Add crowd functionality into a DBMS
- Extend traditional query engine
  - New operators
  - Generating and submitting work to microtask crowdsourcing platform
- Use two human capabilities
  - Finding new data
    - Open world
    - Search engines, reference sources
  - Comparing data
    - Image concept
    - Entity resolution

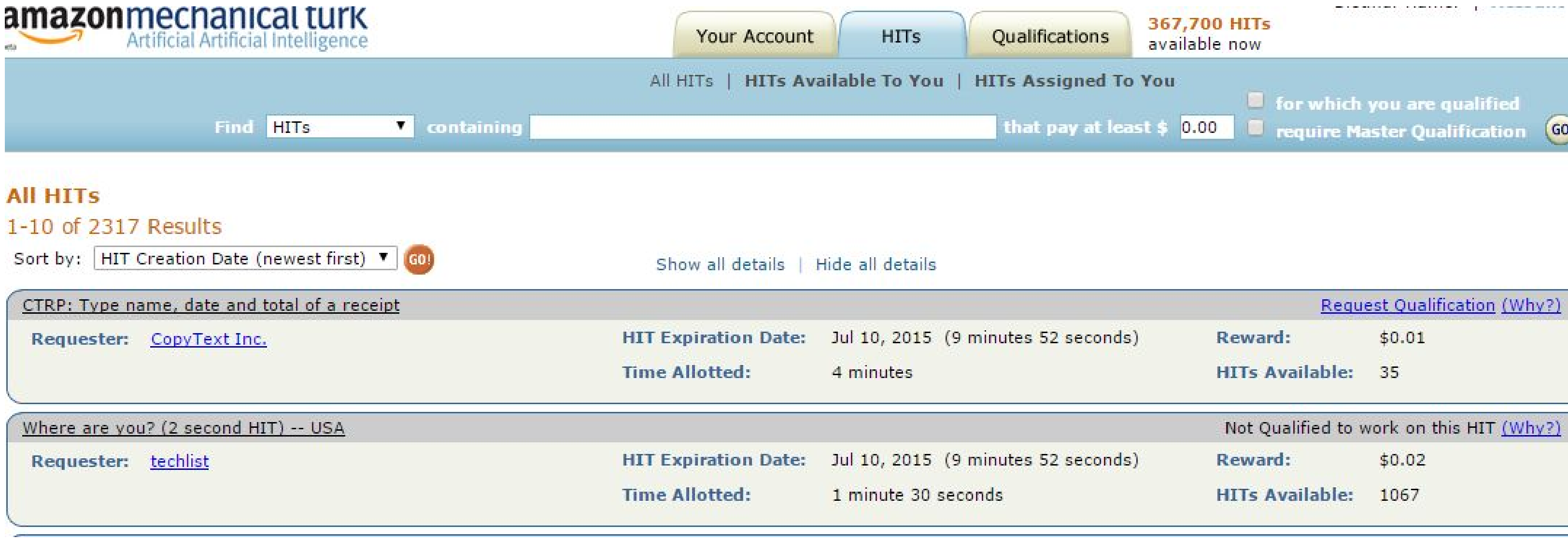
Contributions

- Simple SQL schema and query extensions that enables crowdsourced data processing
- Present new crowdsourced query
- Generate methods to generate effective UIs automatically
- Present experimental results to show CrowdDB is able to answer queries

Amazon Mechanical Turk

A marketplace on which **requesters** offer task and **workers** accept and work on tasks
supports micro-tasks

- Easy
- Fast


amazonmechanical turk
Artificial Artificial Intelligence
Your Account
HITs
Qualifications
All HITs | HITs Available To You | HITs Assigned To You
Find HITs containing that pay at least $ 0.00
for which you are qualified
require Master Qualification
Show all details | Hide all details
GO!

367,700 HITs available now

All HITs
1-10 of 2317 Results
Sort by: HIT Creation Date (newest first)

CTRP: Type name, date and total of a receipt — Request Qualification (Why?)

Requester:	CopyText Inc.	HIT Expiration Date:	Jul 10, 2015 (9 minutes 52 seconds)	Reward:	$0.01
		Time Allotted:	4 minutes	HITs Available:	35

Where are you? (2 second HIT) -- USA — Not Qualified to work on this HIT (Why?)

Requester:	techlist	HIT Expiration Date:	Jul 10, 2015 (9 minutes 52 seconds)	Reward:	$0.02
		Time Allotted:	1 minute 30 seconds	HITs Available:	1067

Mechanical Turk Basics

- Human Intelligent Task (HIT)
  - Smallest entity of works acceptable by a worker
  - Contains one or more jobs
- Assignment
  - Replication of each HIT
  - Used for quality assurance
- HIT Group
  - Contains similar HITs
  - Used by workers to choose a HIT to work

Mechanical Turk APIs

- *createHIT(title, description, question, keywords, reward, duration, maxAssignments, lifetime) → HitID*
- *getAssignmentsForHIT(HitID) → list(asnId, workerId, answer)*
- *approveAssignment(asnID)/rejectAssignment(asnID)*
- *forceExpireHIT(HitID)*

CrowdDB Architecture

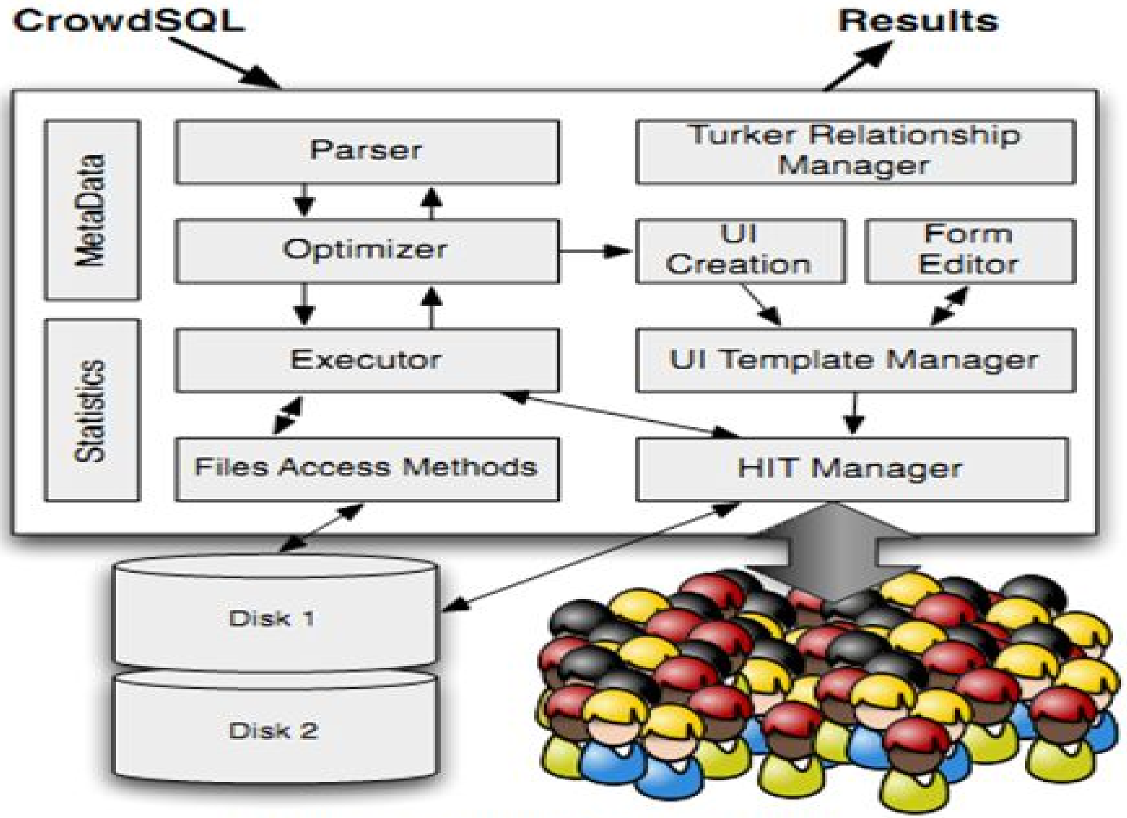


Figure 1: CrowdDB Architecture

New Component

- Turker relationship manager
  - Builds worker-requester relationship
  - Facilitates requester’s duties as accept/reject or grant bonuses
- User interface manager
  - Use annotated tables to automatically generate UI
  - Reduce errors by simple tasks like dropdown lists
  - Create specific forms by programmers if needed
- HIT manager
  - Manage interactions between crowdDB and crowdsourcing platform(AMT)
  - Interacts with storage engines to obtain values and store results

CrowdSQL

A minimal extension to SQL that handles

1. Incomplete data
2. Subjective comparisons

CrowdSQL - Incomplete Data

- SQL DDL Extensions
    - Specific attributes of tuples

Example (Crowdsourced column)

```
CREATE TABLE department (
  university STRING,
  name STRING,
  url CROWD STRING,
  Phone STRING,
  PRIMARY KEY (university,name)    );
```

- Entire tuples

Example (Crowdsourced table)

```
CREATE CROWD TABLE professor (
  name STRING PRIMARY KEY,
  email STRING UNIQUE,
  university STRING,
  department STRING,
  FOREIGN KEY (university,department)
  REF department (university,name)    );
```

CrowdSQL - Incomplete Data

- SQL DML Semantics
  - Introduce a new value to each SQL type : CNULL
  - A CNULL value should be crowdsources when it is first used
  - The default value of any CROWD column
  - Side effect of INSERT

INSERT INTO department (university, name) VALUES (“SFU”, “CS”) ;

The Value for phone : NULL   (Can be set by UPDATE)
The Value for url   : CNULL   (Can be set by UPDATE or Crowdsourced as side-effect of queries)

For a Crowd table all columns values will be set to CNULL except the the KEY

CrowdSQL - Incomplete Data

- Query semantics
  - Crowdsourcing is a part of a query processing

SELECT url FROM department WHERE name = “CS” ;

crowdsource url for existing CS department

*SELECT * FROM professor WHERE email LIKE “%sfu%” AND department = “CS” ;*

crowdsource for existing CNULL professor columns
crowdsource for possible new professors

CrowdSQL - Subjective Comparisons

Introduce two new build in functions :

- CROWDEQUAL (*lvalue, rvalue*)

  SELECT profile FROM department WHERE name ~= “CS” ;

    - Computer Science
    - Computing Science
    - CS

CrowdSQL - Subjective Comparisons

Introduce two new build in functions :

- CROWDORDER

```
CREATE TABLE picture (
   p IMAGE,
   subject STRING         ) ;


SELECT p FROM picture WHERE subject = " SFU CAMPUS" ORDER BY
.           CROWDORDER( p, "Which picture visualizes %subject better") ;
```

CrowdDB in practice

- Budget
  - Response time (crowd tables) and Cost (crowd columns and tables) can be unbounded
  - Use a LIMIT clause to limit the result of a query and so constrain the cost and response time of the query
- Lineage
  - Application programmer may wish to delete all results came from a spammer in the past
  - determine whether the data is outdated
-

User Interface

Two step process:

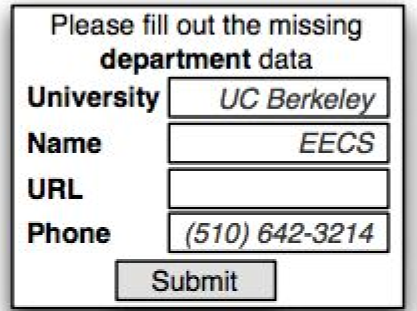
1. Compile time

   CrowdDB creates HTML templates to crowd-source missing information from all CROWD tables and all CROWD columns. Also JS is generated in addition to HTML to do type checking.

2. Runtime

   These templates are instantiated at runtime by filling known field values from a tuple into the HTML form.

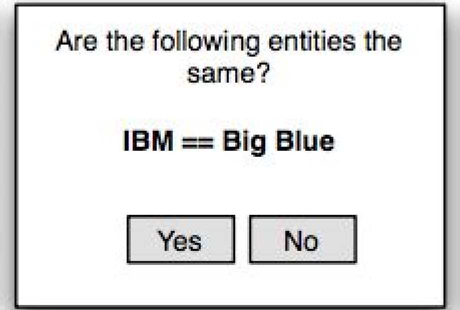
UIs for crowd tasks



Please fill out the missing **department** data

University	*UC Berkeley*
Name	*EECS*
URL	
Phone	*(510) 642-3214*

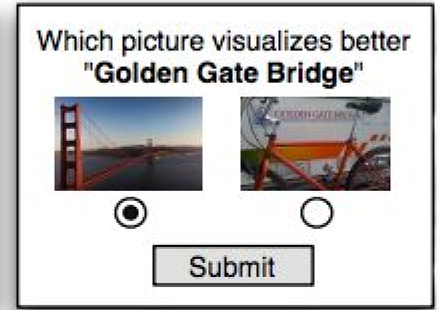
(a) Crowd Column & Crowd Tables w/o Foreign Keys



Are the following entities the same?

IBM == Big Blue

(b) CROWDEQUAL



Which picture visualizes better **"Golden Gate Bridge"**

[✓ ●] ○

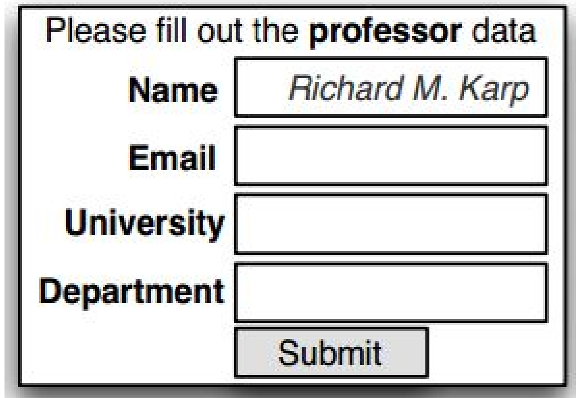
(c) CROWDORDER

Batching and **Prefetching** can also be applied

Multi-relation interface

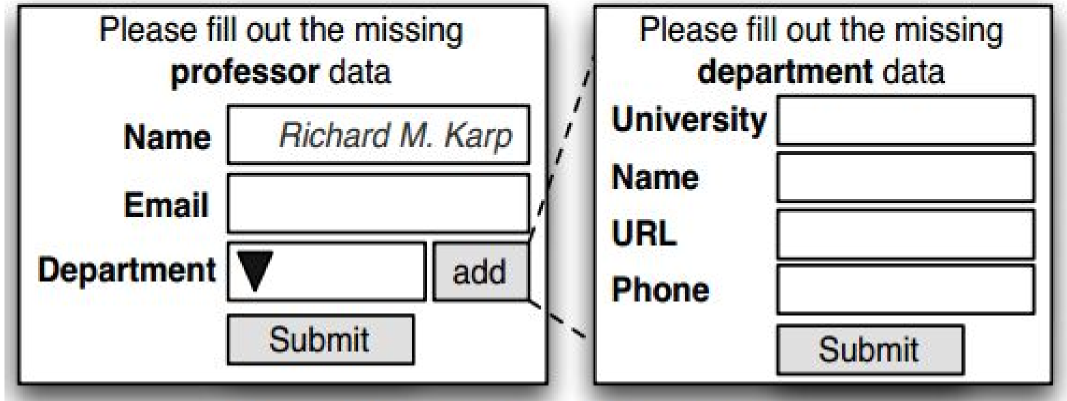
Normalised Interface : The worker inputs the value of foreign key but no other attributes of referenced tuple

Denormalised Interface : There is a select box and an add button which allows the worker to input a new department


Please fill out the **professor** data

Name: *Richard M. Karp*

(d) Foreign Key(normalized)


Please fill out the missing **professor** data

Name: *Richard M. Karp*

Please fill out the missing **department** data

(e) Foreign Key (denormalized)

Overview of Crowd operator

- Initially use a UI template and the standard HIT parameters
- At runtime, they add more tuples to the template
- Depending on the Crowd operator, crowdsource missing values of a tuple or crowdsource new tuples.
- Consume tuples from crowd and do quality control which is done by majority vote

Experimental Results

- Results from running over 25,000 HITs on AMT
- Varying parameters like *price*, *jobs per HIT*.
- The *response time* and *quality* of the answers provided by workers is measured
- Used a Micro Benchmark

Micro Benchmark

```
CREATE TABLE businesses (
name VARCHAR PRIMARY KEY,
phone_number CROWD VARCHAR(32),
address CROWD VARCHAR(256)
);
```

```
SELECT phone_number, address FROM businesses;
```

- Micro Benchmark of 3607 businesses in 40 US cities
- All experiments are repeated 4 times and report average values
- Groups of 100 HITs containing 5 assignments
- Default reward for each job was 1 cent
- 1 job per HIT

Response time vs HIT group size

As the HIT group size increases, the time to get first x responses decreases.

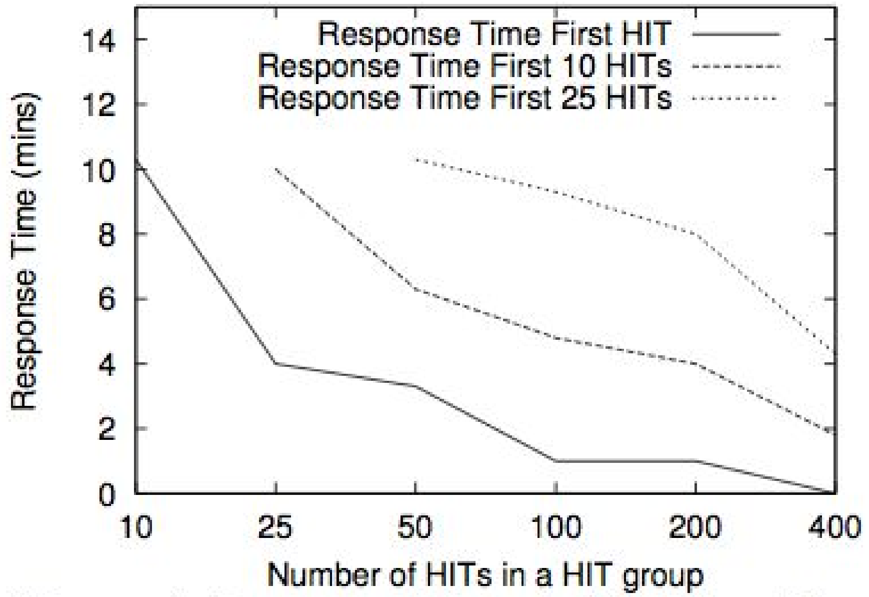


Figure 4: Response Time (min): Vary Hit Group *(1 Asgn/HIT, 1 cent Reward)*

%compeletion vs HIT group size

The percentage of HIT's completed in the 30 minutes increases and then decreases

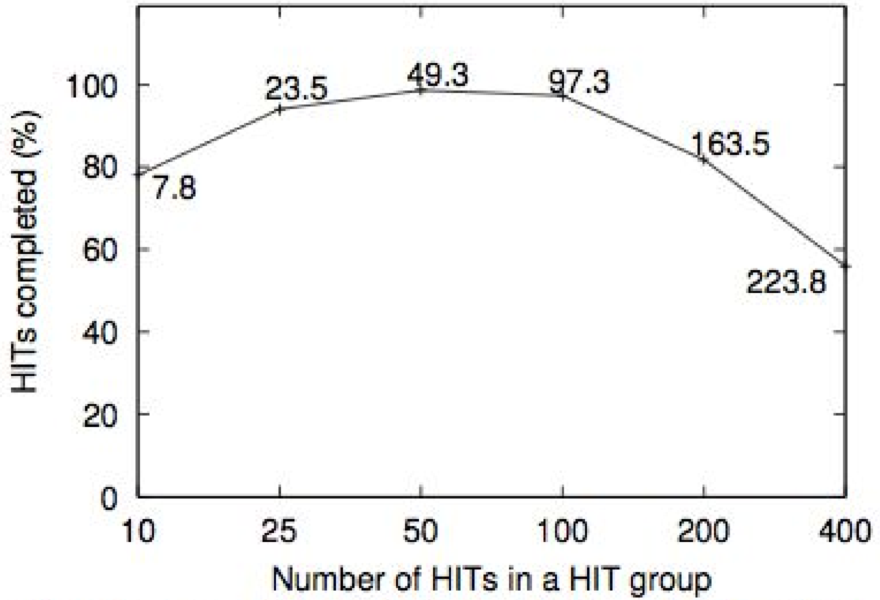


Figure 5: Completion (%): Vary Hit Group *(1 Asgn/HIT, 1 cent Reward)*

%completion vs reward

Paying of more than 2 cents barely makes any difference

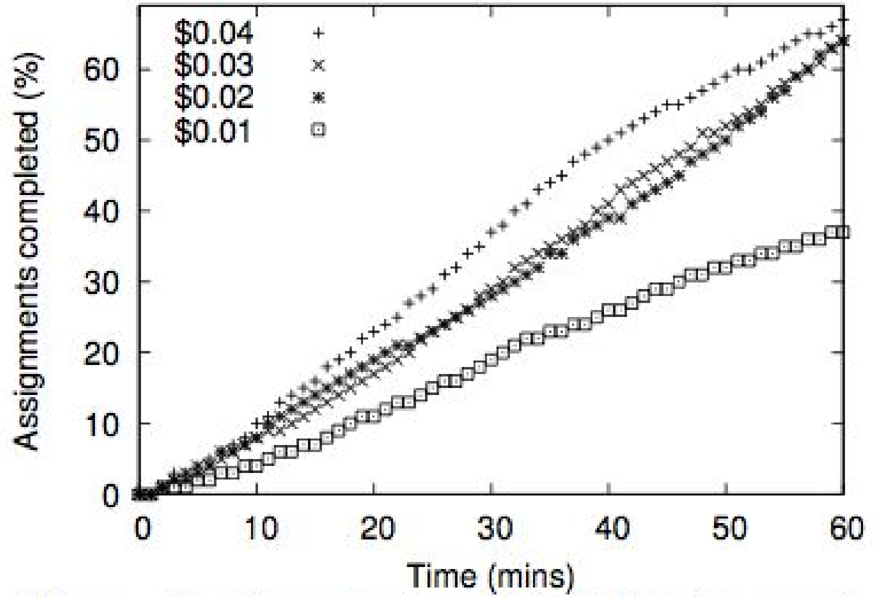


Figure 6: Completion (%): Vary Reward *(100 HITs/Group, 5 Asgn/HIT)*

Worker Affinity and Quality

750 workers

Expected the workers doing more hits to have lesser error but this behaviour not seen in experiments.

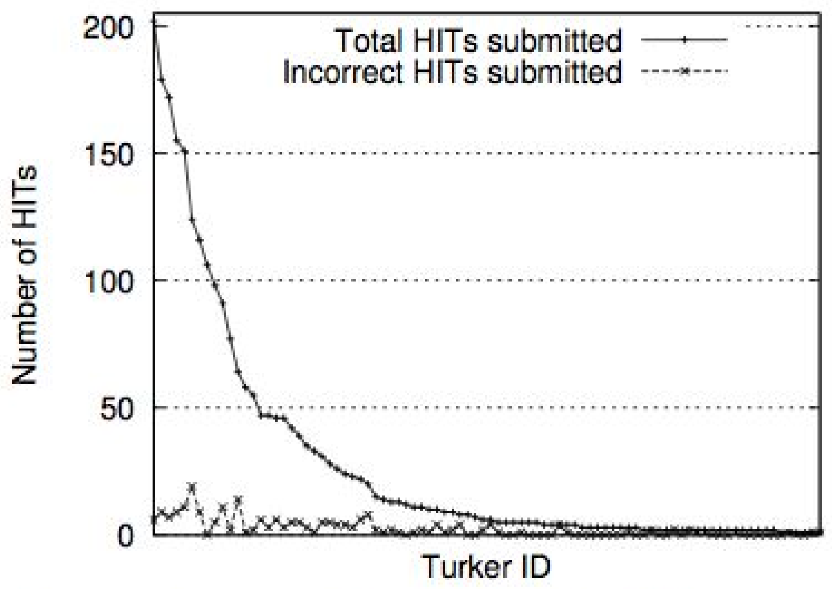
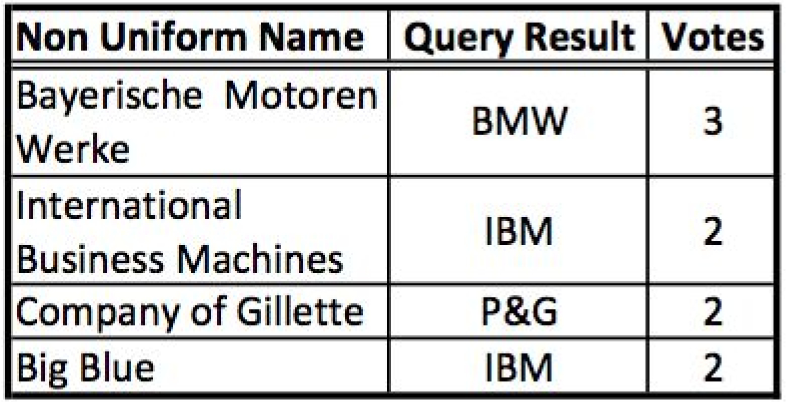


Figure 8: HITs/Quality by Worker *(Any HITs/Group, 5 Asgn/HIT, Any Reward)*

Complex queries: entity resolution

- Used a company schema with only name and headquarter address
- Populated with 100 companies
- Used four different samples of the query below
- Compare 10 company names
- 3 assignments per HIT

```
SELECT name FROM company WHERE
name~="[a non-uniform name of the company]"
```

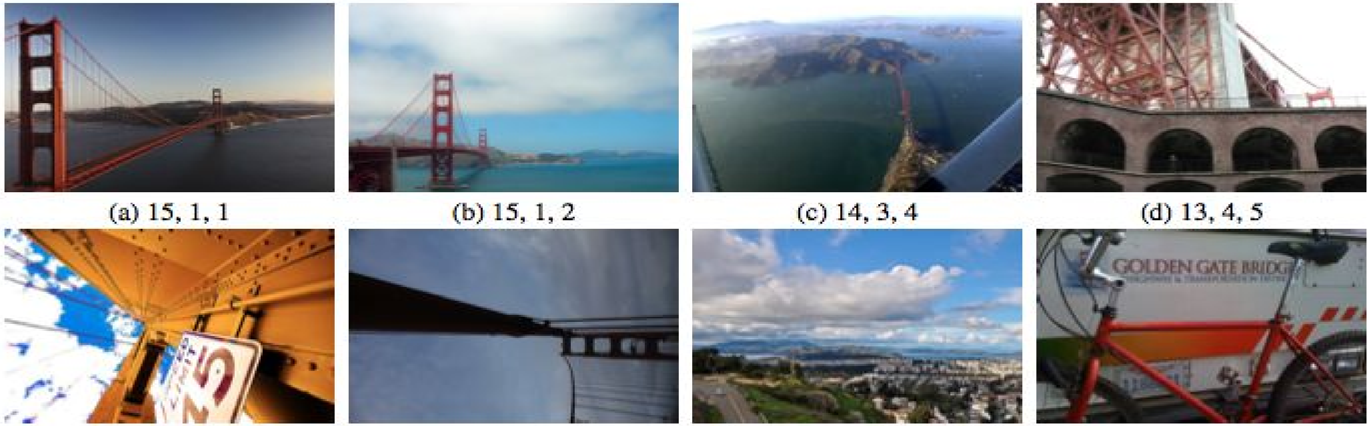


Non Uniform Name	Query Result	Votes
Bayerische Motoren Werke	BMW	3
International Business Machines	IBM	2
Company of Gillette	P&G	2
Big Blue	IBM	2

Figure 9: Entity Resolution on Company Names

Complex queries: Ordering pictures

- Used 8 pictures of 30 subjects
- Compare 4 pair of pictures in a HIT, used 210 HITs with 3 assignments
- Took 68 minutes to complete the experiments
- (# of workers vote, majority vote ranking, expert ranking)



(a) 15, 1, 1 (b) 15, 1, 2 (c) 14, 3, 4 (d) 13, 4, 5

(e) 10, 5, 6 (f) 9, 6, 3 (g) 4, 7, 7 (h) 4, 7, 8

Figure 10: Pictures of the Golden Gate Bridge [1] ordered by workers. The tuples in the sub-captions is in the following format: {the number of votes by the workers for this picture, rank of the picture ordered by the workers (based on votes), rank of the picture ordered by experts}.

Thank You!