Crowdsourcing Applications and Platforms: A Data Management Perspective

AnHai Doan¹,², Michael J. Franklin³, Donald Kossmann⁴, Tim Kraska³

¹University of Wisconsin, Madison
²Walmart Labs
³University of California, Berkeley
⁴ETH Zurich
Tutorial Context

• Other Crowdsourcing Tutorials:

• This tutorial:
  – Emphasis on hybrid computer/human systems for query processing.
  – Integration with traditional database models, algorithms and systems architectures.
Billions of Devices
Ubiquitous Connectivity
CROWDSOURCING EXAMPLES
Citizen Science

NASA “Clickworkers” circa 2000

Pixels indicate Clickworker’s identified craters
Citizen Journalism and Participatory Sensing
DB specific
SO WHAT IS IT?
One View of Crowdsourcing

The Way Industry Looks At It
Taxonomies

• Doan, Halevy, Ramakrishnan; (Crowdsourcing) CACM 4/11
  – nature of collaboration (implicit vs. explicit)
  – architecture (standalone vs. piggybacked)
  – must recruit users/workers? (yes or no)
  – What do users/workers do? 

• Bederson & Quinn; (Human Computation) CHI ’11
  – Motivation (Pay, Altruism, Enjoyment, Reputation)
  – Quality Control (mechanisms – see Tim’s section)
  – Aggregation (how are results combined?)
  – Human Skill (Visual recognition, language, ...)
  – ...
Participatory Culture - Explicit
Participatory Culture – Implicit

John Murrell: GM SV 9/17/09

...every time we use a Google app or service, we are working on behalf of the search sovereign, creating more content for it to index and monetize or teaching it something potentially useful about our desires, intentions and behavior.
# Types of Tasks

<table>
<thead>
<tr>
<th>Task Granularity</th>
<th>Examples</th>
</tr>
</thead>
</table>
| Complex Tasks      | • Build a website  
                     • Develop a software system  
                     • Overthrow a government? |
| Simple Projects    | • Design a logo and visual identity  
                     • Write a term paper |
| Macro Tasks        | • Write a restaurant review  
                     • Test a new website feature  
                     • Identify a galaxy |
| Micro Tasks        | • Label an image  
                     • Verify an address  
                     • Simple entity resolution |

MICRO-TASK MARKETPLACES
Amazon Mechanical Turk (AMT)
Microtasking – Virutalized Humans

• Current leader: Amazon Mechanical Turk
• Requestors place Human Intelligence Tasks (HITs)
  – Minimum price: $0.01
  – #of replicas (assignments), expiration, User Interface
  – API-based: “createHit()”, “getAssignments()”, “approveAssignments()”, “forceExpire()”
  – Requestors approve jobs and payment
• Workers (a.k.a. “tinkers”) choose jobs, do them, get paid
### All HITs

1-10 of 1157 Results

Sort by: HIT Creation Date (newest first) Show all details | Hide all details

<table>
<thead>
<tr>
<th>Requester</th>
<th>HIT Expiration Date</th>
<th>Reward</th>
<th>Time Allotted</th>
<th>Hits Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mr Doe</td>
<td>Aug 27, 2011 (1 day 5 hours)</td>
<td>$0.01</td>
<td>60 seconds</td>
<td>8424</td>
</tr>
<tr>
<td>Redwood Technologies</td>
<td>Aug 27, 2011 (11 hours 59 minutes)</td>
<td>$0.01</td>
<td>15 minutes</td>
<td>6</td>
</tr>
<tr>
<td>Redwood Technologies</td>
<td>Aug 27, 2011 (11 hours 58 minutes)</td>
<td>$0.03</td>
<td>15 minutes</td>
<td>8</td>
</tr>
<tr>
<td>Mr Doe</td>
<td>Aug 27, 2011 (1 day 5 hours)</td>
<td>$0.01</td>
<td>60 seconds</td>
<td>12390</td>
</tr>
</tbody>
</table>

**Validate Brand/Product Information from Product Picture**

**Copy Brand/Product Information from Product Picture**

---

274,745 Hits available now

Find HITs containing that pay at least $ 0.00
'Are these two pictures of the same person?'

Yes ☐ No ☐

Please ACCEPT the hit before submitting.
Amount per Assignment: $0.01
Amount to Approve Outstanding Assignments: $0.01
Pending Review: 1
Reviewed: 0
Remaining: 0
Total: 1

EXPIRATION DATE
Aug 27, 2011, 02:25 PM PDT

View HIT

Review Submitted Assignments (showing page 1 of 1)
Select Assignments to approve or reject then click “Submit.” When you approve an Assignment, the Worker is paid automatically. You will not be charged for Assignments you reject.

<table>
<thead>
<tr>
<th>Approve</th>
<th>Reject</th>
<th>Worker ID</th>
<th>Result</th>
<th>Less</th>
</tr>
</thead>
<tbody>
<tr>
<td>All: None</td>
<td>All: None</td>
<td>A5RPKDYH18EWQ</td>
<td>crowdEqual: no</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>jsessionid: 48f4e460c2ac00c3df977f1cdebe5b13d</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>crowdDbCallback:</td>
<td><a href="http://128.32.45.115:6082/crowdEqaul.do">http://128.32.45.115:6082/crowdEqaul.do</a></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>jsessionid=48f4e460c2ac00c3df977f1cdebe5b13d</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>wlm:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>id: 10202458</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Submission Date: Aug 28, 2011, 02:33 PM PDT</td>
<td></td>
</tr>
</tbody>
</table>

You've chosen to:

approve: none selected
reject: none selected

Submit
Microtask Aggregators

Enterprise Crowdsourcing Solutions

CrowdFlower’s technology engages a global workforce to solve your large-scale data problems.

Business Listing Verification
Correct inaccurate business listings.

Search Relevance
Assess the relevance of your search results.

Product Categorization
Categorize large data sets.

Content Generation
Get quality content in real time.

Custom Solutions
Tailored solutions to fit your needs.
Samasource.org
HYBRID HUMAN/COMPUTER COMPUTATION
Not Exactly Crowdsourcing, but...

Man-Computer Symbiosis

J. C. R. Licklider
IRE Transactions on Human Factors in Electronics, volume HFE-1, pages 4-11, March 1960

“The hope is that, in not too many years, human brains and computing machines will be coupled together very tightly, and that the resulting partnership will think as no human brain has ever thought and process data in a way not approached by the information-handling machines we know today.”
Thinking About Hybrid Systems
Example: Hybrid Image Search

Yan, Kumar, Ganesan, CrowdSearch: Exploiting Crowds for Accurate Real-time Image Search on Mobile Phones, Mobisys 2010.
Example: Hybrid Data Integration

<table>
<thead>
<tr>
<th>paper</th>
<th>conf</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data integration</td>
<td>VLDB-01</td>
</tr>
<tr>
<td>Data mining</td>
<td>SIGMOD-02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>title</th>
<th>author</th>
<th>email</th>
<th>venue</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLAP</td>
<td>Mike</td>
<td>mike@a</td>
<td>ICDE-02</td>
</tr>
<tr>
<td>Social media</td>
<td>Jane</td>
<td>jane@b</td>
<td>PODS-05</td>
</tr>
</tbody>
</table>

- **Generate plausible matches**
  - paper = title, paper = author, paper = email, paper = venue
  - conf = title, conf = author, conf = email, conf = venue

- **Ask users to verify**

Does attribute paper match attribute author?

<table>
<thead>
<tr>
<th>paper</th>
<th>conf</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data integration</td>
<td>VLDB-01</td>
</tr>
<tr>
<td>Data mining</td>
<td>SIGMOD-02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>title</th>
<th>author</th>
<th>email</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLAP</td>
<td>Mike</td>
<td>mike@a</td>
</tr>
<tr>
<td>Social media</td>
<td>Jane</td>
<td>jane@b</td>
</tr>
</tbody>
</table>

Yes  No  Not sure

McCann, Shen, Doan: Matching Schemas in Online Communities. ICDE, 2008
Example: Hybrid Query Processing

Use the crowd to answer DB-hard queries

Where to use the crowd:
• Find missing data
• Make subjective comparisons
• Recognize patterns

But not:
• Anything the computer already does well

The Rest of the Tutorial

• Sample “relational” data management applications
  – information extraction
  – schema matching
• Crowdsourcing DB platforms
  – data models, query languages
  – query processing, optimization
  – quality issues
• Sample “beyond-relational” data management applications
  – graph search / classification
  – mobile image search
  – social media analysis (WalmartLabs)
• Research challenges
Crowdsourced Data Management
Applications

● Relational
  – information extraction
  – schema matching
  – entity resolution
  – data spaces
  – building structured KBs
  – sorting
  – top-k
  – ...

● Beyond relational
  – graph search
  – classification
  – mobile image search
  – social media analysis
  – question answering
  – NLP
  – text summarization
  – sentiment analysis
  – semantic wikis
  – ...

(Apologies if we forget your favorite applications)
Key Issues

- The role of machine (i.e., algorithm) and humans
  - use only humans? both? who’s doing what?
- Recruiting models
  - pay? volunteer?
- User interfaces
- Quality control
- Optimization
Problem: Populate Infoboxes
Solution: IE Using Machine + Human

Hemingway was an American author ...

The American readers ...

Infobox

Born July 21, 1899
Nationality American

Train a “nationality” extractor

Apply extractor to new pages to extract nationalities.

Verify with crowd

Raybradbury is a well-known American author ...

⇒ Nationality = American

Christa Wolf is a German author ...

⇒ Nationality = German
Recruiting Model: Search Advertising

- **Interrupt user in the middle of a primary task**
  - searching for information on Ray Bradbury
- **Ask if user is willing to contribute**
- **Evaluate three UIs: pop-up, highlight, icon**
  - in terms of intrusiveness and willingness to contribute
Ray Bradbury
From Wikipedia, the free encyclopedia

Ray Douglas Bradbury (born August 22, 1920) is an American literary, fantasy, horror, science fiction, and mystery writer best known for *The Martian Chronicles*, a 1950 book which has been described both as a short story collection and a novel, and his 1953 dystopian novel *Fahrenheit 451*. He is widely considered to be one of the greatest and most popular American writers of speculative fiction during the twentieth century.

Contents [show]

Beginnings

Bradbury was born in Waukegan, Illinois, to a Swedish father and telephone lineman. His paternal grandfather and grandmother were from Sweden. His mother was the daughter of a Swedish immigrant from the tiny Swedish community of Waukegan. Bradbury was a reader and writer throughout his youth. He attended the local library called Waukegan Public Library and used it as a setting for much of his novel *Something Wicked This Way Comes*, and depicted Waukegan as "Green Town" in some of his other semi-autobiographical novels — *Dandelion Wine, Farewell Summer* — as well as in many of his short stories.[8]

He attributes his lifelong habit of writing every day to an incident in 1932 when a carnival entertainer, Mr. Electrico,[9] touched him with an electrified sword, made his hair stand on end, and shouted, "Live forever!"

The Bradbury family lived in Tucson, Arizona, in 1926-27 and 1932-33 as his father pursued employment, each time returning to Waukegan, and eventually settled in Los Angeles in 1934, when Ray was thirteen.

Bradbury graduated from the Los Angeles High School in 1936 but chose not to attend college. Instead, he sold newspapers at the corner of South Noron Avenue and Olympic Boulevard. He continued to educate himself at the local library, and having been influenced by science fiction heroes like Flash Gordon and Buck Rogers, he began to publish science fiction stories in *fanzines* in 1938. Ray was invited by Forrest J. Ackerman to attend the now legendary Clifton’s Cafeteria Science Fiction Club. Here Ray met the writers Robert A. Heinlein, Emil Petaja, Frederic Brown, Henry Kuttner, Leigh Brackett, and Jack Williamson.
Ray Bradbury

Ray Douglas Bradbury (born August 22, 1920) is an American literary, fantasy, horror, science fiction, and mystery writer best known for The Martian Chronicles, a 1950 book which has been described both as a short story collection and a novel, and his 1953 dystopian novel Fahrenheit 451. He is widely considered to be one of the greatest and most popular American writers of speculative fiction during the twentieth century.

Contents [show]

Beginnings

Bradbury was born in Waukegan, Illinois, to a Swedish immigrant mother and a father who was a power and telephone lineman. His paternal grandfather and great-grandfather were newspaper publishers.

Bradbury was a reader and writer throughout his youth, spending much time in the Carnegie Library in Waukegan. He used this library as a setting for much of his novel Something Wicked This Way Comes, and depicted Waukegan as "Green Town" in some of his other semi-autobiographical novels — Dandelion Wine, Farewell Summer — as well as in many of his short stories.

He attributes his lifelong habit of writing every day to an incident in 1932 when a carnival entertainer, Mr. Electrico, touched him with an electrified sword, made his hair stand on end, and shouted, "LIVE FOREVER!"

The Bradbury family lived in Tucson, Arizona, in 1928-27 and 1932-33 as his father pursued employment, each time returning to Waukegan, and eventually settling in Los Angeles in 1934, when Ray was thirteen.

Bradbury graduated from the Los Angeles High School in 1936 but chose not to attend college. Instead, he sold newspapers at the corner of South Naron Avenue and Olympic Boulevard. He continued to educate himself at the local library, and having been influenced by science fiction heroes like Flash Gordon and Buck Rogers, he began to publish science fiction stories in fanzines in 1938. Ray was invited by Forrest J Ackerman to attend the now legendary Clifton’s Cafeteria Science Fiction Club. Here Ray met the writers Robert A. Heinlein, Emil Petaja, Frederik Brown, Henry Kuttner, Leigh Brackett, and Jack Williamson.
Ray Bradbury
From Wikipedia, the free encyclopedia

Ray Douglas Bradbury (born August 22, 1920) is an American literary, fantasy, horror, science fiction, and mystery writer best known for The Martian Chronicles, a 1950 book which has been described both as a short story collection and a novel, and his 1953 dystopian novel Fahrenheit 451, considered to be one of the greatest and most popular American writers of the 20th century.

Contents [show]

Beginnings
Bradbury was born in Waukegan, Illinois, to a Swedish immigrant mother and a French and telephone lineman. His paternal grandfather and great-grandfather were newspaper publishers.

Bradbury was a reader and writer throughout his youth, spending much time in the Carnegie Library in Waukegan. He used this library as a setting for much of his novel Something Wicked This Way Comes, and depicted Waukegan as "Green Town" in some of his other semi-autobiographical novels — Dandelion Wine, Farewell Summer — as well as in many of his short stories.

He attributes his lifelong habit of writing every day to an incident in 1932 when a carnival entertainer, Mr. Electroy, touched him with an electrified sword, made his hair stand on end, and shouted, "Live forever!"

The Bradbury family lived in Tucson, Arizona, in 1926-27 and 1932-33 as his father sought employment, each time returning to Waukegan, and eventually settled in Los Angeles in 1934, when Ray was thirteen.

Bradbury graduated from the Los Angeles High School in 1936 but chose not to attend college. Instead, he sold newspapers at the corner of South Noron Avenue and Olympic Boulevard. He continued to educate himself at the local library, and having been influenced by science fiction heroes like Flash Gordon and Buck Rogers, he began to publish science fiction stories in fanzines in 1938. Ray was invited by Forrest J Ackerman to attend the now legendary Clifton's Cafeteria Science Fiction Club. Here Ray met the writers Robert A. Heinlein, Emil Batail, Fredric Brown, Henry Kuttner, Leigh Brackett, and Jack Williamson.
Ray Bradbury

From Wikipedia, the free encyclopedia

Ray Douglas Bradbury (born August 22, 1920) is an American literary, fantasy, horror, science fiction, and mystery writer best known for The Martian Chronicles, a 1950 book which has been described both as a short story collection and a novel, and his 1953 dystopian novel Fahrenheit 451. He is widely considered to be one of the greatest and most popular American writers of speculative fiction during the twentieth century.

Contents [show]

Beginnings

Bradbury was born in Waukegan, Illinois, to a Swedish immigrant mother and a father who was a power and telephone lineman. His paternal grandfather and great-grandfather were newspaper publishers.

Bradbury was a reader and writer throughout his youth, spending much time in the Carnegie Library in Waukegan. He used this library as a setting for much of his novel Something Wicked This Way Comes, and depicted Waukegan as "Green Town" in some of his other semi-autobiographical novels — Dandelion Wine, Farewell Summer — as well as in many of his short stories.

He attributes his lifelong habit of writing every day to an incident in 1932 when a carnival entertainer, Mr. Electrico, touched him with an electrified sword, made his hair stand on end, and shouted, "Live forever!"

The Bradbury family lived in Tucson, Arizona, in 1926–27 and 1932–33 as his father pursued employment, each time returning to Waukegan, and eventually settled in Los Angeles in 1934, when Ray was thirteen.

Bradbury graduated from the Los Angeles High School in 1936 but chose not to attend college. Instead, he sold newspapers at the corner of South Noron Avenue and Olympic Boulevard. He continued to educate himself at the local library, and having been influenced by science fiction heroes like Flash Gordon and Buck Rogers, he began to publish science fiction stories in fanzines in 1938. Ray was invited by Forrest J Ackerman to attend the now legendary Clifton's Cafeteria Science Fiction Club. Here Ray met the writers Robert A. Heinlein, Emil Petaja, Frederic Brown, Henry Kuttner, Leigh Brackett, and Jack Williamson.
Ray Bradbury

From Wikipedia, the free encyclopedia

Ray Douglas Bradbury (born August 22, 1920) is an American literary, fantasy, horror, science fiction, and mystery writer best known for The Martian Chronicles, a 1950 book which has been described both as a short story collection and a novel, and his 1953 dystopian novel Fahrenheit 451. He is widely considered to be one of the greatest and most popular American writers of speculative fiction during the twentieth century.

Contents [show]

Beginnings

Bradbury was born in Waukegan, Illinois, and attended the University of Iowa. He is a retired telephone lineman. He was a reader and writer throughout his life, and he used his library as a source for many of his stories. He was a reader and writer throughout his life, and he used his library as a source for many of his stories.

He attributes his lifelong habit of writing to the influence of Mr. Einstein, whom he met when Ray was thirty.

The Bradbury family lived in Tucson, Arizona, where their employment, each time returning to Illinois. Instead, he sold newspapers at the corner of South Norton Avenue and Olympic Boulevard. He continued to work at the local library, and having been influenced by science fiction heroes like Flash Gordon and Buck Rogers, he began to publish science fiction stories in fanzines in 1938. Ray was invited by Forrest J Ackerman to attend the now legendary Clifton's Cafeteria Science Fiction Club. Here, Ray met the writers Robert A. Heinlein, Emil Pataki, Frederic Brown, Henry Kuttner, Leigh Brackett, and Jack Williamson.
Ray Bradbury

From Wikipedia, the free encyclopedia

Ray Douglas Bradbury (born August 22, 1920) is an American literary, fantasy, horror, science fiction, and mystery writer best known for *The Martian Chronicles*, a 1950 book which has been described both as a short story collection and a novel, and his 1953 dystopian novel *Fahrenheit 451*. He is widely considered to be one of the greatest and most popular American writers of speculative fiction during the twentieth century.

### Beginnings

Bradbury was born in Waukegan, Illinois, to a Swedish immigrant mother and a father who was a power and telephone lineman. His paternal grandfather and great-grandfather were newspaper publishers.

Bradbury was a reader and writer throughout his youth, spending much time in the Carnegie Library in Waukegan. He used this library as a setting for much of his novel *Something Wicked This Way Comes*, and depicted Waukegan as "Green Town" in some of his other semi-autobiographical novels — *Dandelion Wine, Farewell Summer* — as well as in many of his short stories.

He attributes his lifelong habit of writing every day to an incident in 1932 when a carnival entertainer, Mr. Electrico, touched him with an electrified sword, made his hair stand on end, and shouted, "Live forever!"

The Bradbury family lived in Tucson, Arizona, in 1926–27 and 1932–33 as his father pursued employment, each time returning to Waukegan, and eventually settled in Los Angeles in 1934, when Ray was thirteen.

Bradbury graduated from the Los Angeles High School in 1938 but chose not to attend college. Instead, he sold newspapers at the corner of South Norton Avenue and Olympic Boulevard. He continued to educate himself at the local library, and having been influenced by science fiction heroes like Flash Gordon and Buck Rogers, he began to publish science fiction stories in *Famous* in 1939. Ray was invited by Forrest J.
Ray Bradbury

From Wikipedia, the free encyclopedia

Ray Douglas Bradbury (born August 22, 1920) is an American literary, fantasy, horror, science fiction, and mystery writer best known for The Martian Chronicles, a 1950 book which has been described both as a short story collection and a novel, and his 1953 dystopian novel Fahrenheit 451. He is widely considered to be one of the greatest and most popular American writers of speculative fiction during the twentieth century.

Contents [show]

Beginnings

Bradbury was born in Waukegan, Illinois, to a Swedish immigrant mother and a father who was a power and telephone lineman. His paternal grandfather and great-grandfather were newspaper publishers.

Bradbury was a reader as a young boy, often spending much time in the Carnegie Library in Waukegan. He had a great affection for the library, and he included it in his novel Something Wicked This Way Comes. He also played a minor role in some of his other semi-autobiographical stories. He attributes his lifelong love of libraries to his experiences there.[3]

He attributes his lifelong love of libraries to his experiences there. He attributes his lifelong love of libraries to his experiences there.

The Bradbury family lived in Tucson, Arizona, in 1926–27 and 1932–33 as his father pursued employment, each time returning to Waukegan, and eventually settled in Los Angeles in 1934, when Ray was thirteen.

Bradbury graduated from the Los Angeles High School in 1938 but chose not to attend college. Instead, he sold newspapers at the corner of South Norton Avenue and Olympic Boulevard. He continued to educate himself at the local library, and having been influenced by science fiction heroes like Flash Gordon and Buck Rogers, he began to publish science fiction stories in fanzines in 1938. Ray was invited by Forrest J. Ackerman to join him in writing regular columns for the science fiction magazine Amazing Stories, which ran for more than two decades.
Ray Bradbury

From Wikipedia, the free encyclopedia

Ray Douglas Bradbury (born August 22, 1920) is an American literary, fantasy, horror, science fiction, and mystery writer best known for The Martian Chronicles, a 1950 book which has been described both as a short story collection and a novel, and his 1953 dystopian novel Fahrenheit 451. He is widely considered to be one of the greatest and most popular American writers of speculative fiction during the twentieth century.

Beginnings

Bradbury was born in Waukegan, Illinois, to a Swedish immigrant mother and a father who was a power and telephone lineman. [1] His paternal grandfather and great-grandfather were newspaper publishers. [2]

Bradbury was a reader and writer throughout his youth, spending much time in the Carnegie Library in Waukegan. He used this library as a setting for much of his novel Something Wicked This Way Comes, and depicted Waukegan as "Green Town" in some of his other semi-autobiographical novels — Dandelion Wine, Farewell Summer — as well as in many of his short stories. [3]

He attributes his lifelong habit of writing every day to an incident in 1932 when as an entertainer, Mr. Electrico, touched him with an electrified sword, made his hair stand, and shouted, "Live forever!"

The Bradbury family lived in Tucson, Arizona, in 1926–27 and 1932–33 as his father pursued employment, each time returning to Waukegan, and eventually settled in Los Angeles in 1934, when Ray was thirteen.

Bradbury graduated from the Los Angeles High School in 1938 but chose not to attend college. Instead, he sold newspapers at the corner of South Norton Avenue and Olympic Boulevard. He continued to educate himself at the local library, and having been influenced by science fiction heroes like Flash Gordon and Buck Rogers, he began to publish science fiction stories in fanzines in 1938. Ray was invited by Forrest J.
Ray Bradbury

From Wikipedia, the free encyclopedia

Ray Douglas Bradbury (born August 22, 1920) is an American literary, fantasy, horror, science fiction, and mystery writer best known for The Martian Chronicles, a 1950 book which has been described both as a short story collection and a novel and his 1953 dystopian novel Fahrenheit 451 which are among the most popular American works of speculative fiction. We think the article includes Ray Bradbury's nationality.

Does the article say?

- American
  It seems likely the article says this in the sentence:
  "He is widely considered to be one of the greatest and most popular American writers of speculative fiction during the twentieth century."

- British
  It seems likely the article says this in the sentence:
  "A chance encounter in a Los Angeles bookstore with the British expatriate writer Christopher Isherwood gave Bradbury the opportunity to put The Martian Chronicles into the hands of a respected critic."

- None of the above

The Bradbury family lived in Waukegan, Illinois, in hardship during his youth. Returning to Waukegan, and eventually Clear Spring, in 1934, when Ray was thirteen.

Bradbury graduated from the Los Angeles High School in 1938 but chose not to attend college. Instead, he sold newspapers at the corner of South Norton Avenue and Olympic Boulevard. He continued to educate himself at the local library, and having been influenced by science fiction heroes like Flash Gordon and Buck Rogers, he began to publish science fiction stories in fanzines in 1938. He was invited by Forrest J. Ackerman...
Important to Get the Wording Right

Ray Douglas Bradbury (born August 22, 1920) is an American literary, fantasy, and mystery writer best known for *The Martian Chronicles*, a 1950 book which

“We think Ray Bradbury’s nationality is American. Is this correct?”

“We think the summary should say Ray Bradbury’s nationality is American. Is this what the article says?”

“If I knew what it really is, I would really need to look”

“Please check with the Britannica!”
Experimental Results

- Deployed with Yahoo / Google
  - 7 days
  - $1500
  - 2473 visitors
  - 224 answers

- Icon < Highlight < Popup in intrusiveness
- Icon < Highlight < Popup in willingness to contribute

- Out of 224 answers
  - 90% marked correct are indeed correct
  - 58% marked incorrect are indeed incorrect

- Training on noisy answers improves extraction accuracy
Schema Matching

Matching Schemas in Online Communities: A Web 2.0 Approach. R. McCann, W. Shen, A. Doan. ICDE-08
### Schema Matching

<table>
<thead>
<tr>
<th>paper</th>
<th>conf</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data integration</td>
<td>VLDB-01</td>
</tr>
<tr>
<td>Data mining</td>
<td>SIGMOD-02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>title</th>
<th>author</th>
<th>email</th>
<th>venue</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLAP</td>
<td>Mike</td>
<td>mike@a</td>
<td>ICDE-02</td>
</tr>
<tr>
<td>Social media</td>
<td>Jane</td>
<td>jane@b</td>
<td>PODS-05</td>
</tr>
</tbody>
</table>

- Generate plausible matches & ask users to verify
  - paper = title, paper = author, paper = email, paper = venue
  - conf = title, conf = author, conf = email, conf = venue

### Does attribute paper match attribute author?

<table>
<thead>
<tr>
<th>paper</th>
<th>conf</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data integration</td>
<td>VLDB-01</td>
</tr>
<tr>
<td>Data mining</td>
<td>SIGMOD-02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>title</th>
<th>author</th>
<th>email</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLAP</td>
<td>Mike</td>
<td>mike@a</td>
</tr>
<tr>
<td>Social media</td>
<td>Jane</td>
<td>jane@b</td>
</tr>
</tbody>
</table>
Recruiting Model

- Make users pay for service
Quality Control

- **Classify users into trusted/untrusted**
  - if (U has correctly answered X out of Y evaluation questions) AND (Y >= t₁) AND (X/Y >= t₂) → U is trusted

- **Monitor trusted answers to question Q. Stop when**
  - at least t₃ answers
  - gap between yes/no answers is at least t₄
  - if gap has not been reached by t₅ answers, stop with majority voting

- **Example**
  - minimum # of answers t₃ = 6, gap t₄ = 3, maximum t₅ = 9

<table>
<thead>
<tr>
<th>paper = author?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes, No, No, Yes, Yes, Yes, Yes</td>
</tr>
<tr>
<td>Yes, Yes, Yes, No, Yes, No, No, No, No</td>
</tr>
</tbody>
</table>

- **Must continuously monitor and re-evaluate trusted users**
Optimization

- **Exploit constraints**
  - paper = title
  - paper = author
  - paper = email
  - paper = venue
  - conf = title
  - conf = author
  - conf = email
  - conf = venue

- **Use algorithm to re-rank lists & remove certain matches**
  - paper = title, .8
  - paper = author, .6
  - paper = email, .3
  - conf = author, .7
  - conf = venue, .6
  - conf = email, .4

- **Round robin**
  
  Q₁
  Q₂
  Q₃
  Q₄
  Q₅
  Q₆
  
  If “human oracle” is correct with prob 0.95
  - prob of correctly answering Q₆ = 0.77
Crowd-sourcing Database Systems
Two kinds of systems

• **Specific applications that make use of crowds**
  – crowd-source a particular kind of information
  – e.g., a particular HCI experiment on AMT

• **Platforms that enable apps with crowds**
  – automate certain tasks that involve crowds
  – provide higher-level services than AMT et al.
  – we mostly interested in “database services”
Databases and Crowds

• How can crowds help databases?
  – fix broken data: entity resolution, inconsistencies, ...
  – add missing data
  – subjective comparisons

• How can databases help crowd apps?
  – lazy data acquisition: only get data that is necessary
  – game the workers market
  – semi-automatically create user interfaces
  – manage the data sourced from the crowd

• All systems discussed here cover BOTH sides
Example Systems

- CrowdDB (Berkeley, ETH)
- Qurk (MIT)
- Scoop (Stanford)
- Hlog (Wisconsin)
- Freebase (Google)
- FusionCOMP

- Apologies if we forgot your favorite system
  - Disclaimer: bias to CrowdCrowd meeting (6/2011)
Overview of Topics Covered

• Data Model and Query Language

• Query Processing Techniques

• Quality Control

• *(Again, apologies that we cannot cover all aspects.)*
Data Model: Some Big Questions

• Q1: Same model for electronic and crowd data?
  – Yes, in all systems we are aware of

• Q2: User view vs. system view on data?
  – Different in most systems (for various reasons)

• Q3: Open vs. closed world?
  How to bound human input required for a query?
  – closed world: database determines required input
  – open world 1: limit queries that you may ask
  – open world 2: crowd (+system) answer this question, too
Data Model & Language: Fine Print

• How to deal with uncertainty?
• Support for lineage?
• When is data crowd-sourced?
• How to specify how to get data?
• User view vs. system view on data?
• Support for programmatic updates?
Qurk (MIT)

- **Goal:** crowd-source comparisons, missing data
- **Basis:** SQL3 + UDF
  - UDF encapsulate crowd input
  - special template language for crowd UDFs
  - specify UI, quality control, ... (possibly opt. hints)
- **References**
  - [Marcus et al. CIDR 2011]
  - [Marcus et al. SIGMOD 2011]
  - [Marcus et al. CrowdCrowd 2011]
**Qurk Example** [Markus et al. CrowdCrowd 2011]

- **Task:** Find all women in a “people” database

- **Schema**
  ```sql
  CREATE TABLE people(
      name varchar(256),
      photo blob);
  ```

- **Query**
  ```sql
  SELECT name
  FROM people p
  WHERE isFemale(p);
  ```
Qurk Example [Markus et al. CrowdCrowd 2011]

- Task: Find all women in a “people” database
- Schema
  
  ```sql
  CREATE TABLE people(
    name  varchar(256),
    photo blob );
  ```
- Query

  ```sql
  SELECT name
  FROM  people p
  WHERE isFemale(p);
  ```

  TASK isFemale(tuple) TYPE: Filter
  Question: “is %s Female”,
  tuple[“photo”]
  YesText: “Yes”
  NoText: “No”
Qurk Example [Markus et al. CrowdCrowd 2011]

Is ____ Female?

men in a “people” database

\( \text{people}(\text{256}), \)

\text{TASK isFemale(tuple)} \text{ TYPE: Filter}

\text{Question: “is %s Female”}
\text{tuple[“photo”]}

\text{YesText: “Yes”}
\text{NoText: “No”}
The magic is in the Templates

• Templates generate UIs for different kinds of crowd-sourcing tasks
  – filters: Yes / No questions
  – joins: comparisons between two tuples (equality)
  – order by: comparisons between two tuples (gt?)
  – generative: crowd-source attribute values

• Templates also specify quality control; e.g.,
  COMBINER: MajorityVote
CrowdDB (Berkeley, ETH)

• Goal: crowd-source comparisons, missing data
  – (almost) the same as Qurk
  – difference: crowd-source new tuples

• Design: SQL with slight extensions to DML, QL
  – CROWD keyword to indicate crowd-sourced data
  – syntactic sugar for crowd-sourced comparisons

• References
  – [Franklin et al. SIGMOD 2011]
  – [Xin et al. CrowdCrowd 2011]
  – [Feng et al. VLDB 2011]
CrowdSQL

• CROWD columns
  – entities known, properties of entities may be unknown
  
  ```sql
  CREATE TABLE company (  
  name STRING PRIMARY KEY,  
  hq_address CROWD STRING);
  ```

• CROWD table
  – entities unknown, crowd-source new entities
  
  ```sql
  CREATE CROWD TABLE department (  
  university STRING,  
  department STRING,  
  phone_no STRING)  
  PRIMARY KEY(university, department) );
  ```
CrowdSQL Semantics

• `INSERT INTO company(name) VALUES("Apple");`
  – set `hq_address` to a special crowd-source unknown
  – (also possible to set `hq_address` as part of INSERT.

• `SELECT * FROM company;`
  – return all companies with `name` and `hq_address`
  – if `hq_address` is special unknown, then ask crowd
  – store crowd-source `hq_address` in database
    • i.e., update as a side-effect of a query!!!
Fill up the following information about:

COMPANY

Company Name

Headquarter Address

Submit
CrowdSQL Semantics (ctd.)

• Give me 10 universities that have a CS dept.
  SELECT university FROM department
  WHERE department = "Computer Science"
  LIMIT 10;

• This query is executed as follows...
  – evaluate all departments already recorded in the DB
    • possibly crowd-source the department, university attrs.
    • (actually, not needed in this case because part of the key)
  – if not sufficient, ask the crowd for additional depts.
  – store newly crowd-sourced depts. in the database
  – (Actually, many alternative ways to do this -> see later)
Deco (Stanford, Scoop Project)

• **Goal:** crowd-source missing data
  – focus on data model (comparisons are orthogonal)
  – generalizes + foundations for crowd-sourced models

• **Design:** relational data model
  – clean separation between user + system view
  – fetch rules: add crowd-source tuples & attributes
  – resolution rules: resolve conflicts between workers

• **References**
  – [Widom et al. CrowdCrowd 2011]
Deco: User vs. System View

• **Schema Designer**
  – defines conceptual, relational schema (as usual)
  – defines functional dependencies (~ as usual)
  – defines fetch and resolution rules (new)

• **User**
  – sees conceptual, relational schema and writes app

• **System**
  – defines system (internal) schema to store data
  – conceptual schema is a view on internal schema
Layers of Deco Data Model

designer

Conceptual Schema

Internal Schema

Database

user

system

crowd
Deco (simplified) Example [Widom 2011]

• Conceptual schema (user perspective)
  – Restaurant(name, [rating], [cuisine])

• Fetch / Resolution rules
  – name -> rating (resolution with Method “AVG”)
  – name -> cuisine (resolution with Method “DUPL-ELIM”)
  – Ø -> name
  – rating, cuisine are dependent attributes
  – name is anchor (often, but not necessarily key)

• Internal schema
  – one table for each anchor
  – one table for each fetch rule to resolve conflicts
Deco Example

Internal view

<table>
<thead>
<tr>
<th>name</th>
<th>rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anton’s</td>
<td>5</td>
</tr>
<tr>
<td>McDonald</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>name</th>
<th>cuisine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anton’s</td>
<td>French</td>
</tr>
<tr>
<td>Anton’s</td>
<td>Continental</td>
</tr>
<tr>
<td>McD</td>
<td>FastFood</td>
</tr>
</tbody>
</table>

User view

<table>
<thead>
<tr>
<th>name</th>
<th>rating</th>
<th>cuisine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anton’s</td>
<td>4</td>
<td>French</td>
</tr>
<tr>
<td>Anton’s</td>
<td>4</td>
<td>Continental</td>
</tr>
<tr>
<td>McDonald</td>
<td>3</td>
<td>FastFood</td>
</tr>
</tbody>
</table>
Hlog (Wisconsin)

- **Goal:** Specify workflows with human input
  - materialize human input for reuse
- **Design:** based on Datalog
  - datalog rules define final goal of workflow
  - datalog rules define steps to reach that goal
  - datalog rules define crowd-sourced input
- **References**
  - [Chen et al. VLDB 2007]
  - [Chai et al. SIGMOD 2009]
Hlog Example [Doan 2011]

- Extract PC chairs of conferences from Web page
  - crawl pages:
    - pages(page) :- dataSources(URL, page), crawl(URL, page)
  - extract conferences:
    - conferences(c, p) :- pages(p), extractConf(p, c)
  - extract people:
    - names(n, p) : pages(p), extractName(p, n)
  - extract roles (e.g., PC Chair):
    - role(n, r, p) :- names(n, p), extractRole(n, r, p)
  - putting it together:
    - service(n, r, c) :- conferences(c, p), role(n, r, p)

- (no crowd-sourcing yet...)
Hlog with Crowd-Sourcing

• Ask people to extract the roles from pages
  – $CSroles(n, r, p#no-edit)#spreadsheet-UI :- role(n, r, p)$
  – Page p is an input
  – User creates spreadsheet of $(n, r, p)$ tuples

• Other pragmas for other purposes
  – e.g., #form-UI, #wiki-UI
Other Hlog Features

• **Reuse of crowd-sourced input**
  – user defined tuples are maintained in database
  – datalog rules can be seen as (materialized) views
  – keep lineage in order to distinguish between crowd-sourced and computed input for views
  – (similar approach to CrowdDB and Deco)

• **When to crowd-source**
  – Hlog supports “drive-by crowd-sourcing”
    • get information from workers when workers are ready
    • not necessary when app is ready to consume information
  – (not considered in Qurk, CrowdDB, Deco)
Freebase / MetaWeb (Google)

• **Goal:** Describe the whole world
  – 22 million entities and their relationships
  – entity = person | place | thing
  – entity resolution with single WebID (~ barcode, URI)

• **Foundation:** Semantic Web
  – database is a graph
  – anybody can contribute data to that graph
    • new entities, resolve same entities, build schemas (ontologies)
  – data can be donated at any time (drive-by sourcing)
  – use MQL (~ SPARQL) to query the data

• **Reference**
  – [www.freebase.com](http://www.freebase.com)
Summary: Data Model & Languages

- **Deal with uncertainty?**
  - expose to end user (app developer) or not?
    - Deco: no
    - CrowdDB: no
    - Qurk: no
    - Hlog: yes
    - Freebase: yes / no
  - **who defines resolution?**
    - Deco: schema designer
    - CrowdDB: system (for now)
    - Qurk: UDF designer
    - Hlog: system
    - Freebase: system, crowd (curators)
Summary: Data Model & Languages

• What are the foundations?
  – Data Model and Query Language
    • Deco: SQL, relational model
    • CrowdDB: SQL, relational model
    • Qurk: SQL, relational model
    • Hlog: relational model, workflow (rules)
    • Freebase: “Semantic Web” (graph + SPARQL dialect)

• When is data crowd-sourced?
  • Deco: side-effect of processing queries
  • CrowdDB: side-effect of processing queries
  • Qurk: side-effect of processing queries / workflow
  • Hlog: scheduled vs. drive-by crowdsourcing (CQP)
  • Freebase: continuous independent of query processing
Summary: Data Model & Languages

• How to specify how to get data?
  • Deco: fetch rules, P methods by schema designer
  • CrowdDB: annotations by schema designer, semi-automatically generated templates
  • Qurk: TASK language for UDFs
  • Hlog: (Datalog-like) rules
  • Freebase: system (prefined APIs to contribute)

• Open vs. closed-world assumption?
  • Deco: open world by fetching “anchors”
  • CrowdDB: open world by CROWD tables
  • Qurk: open world by UDF
  • Hlog: open (like CrowdDB)
  • Freebase: open (like Semantic Web)
Summary: Data Model & Languages

• **User view vs. system view on data?**
  - Deco: conceptual vs. actual schema
  - CrowdDB: mostly no difference, cache for comparisons
  - Qurk: mostly no difference
  - Hlog: provenance for human transactions for allowing to crowd-source a view
  - Freebase: unknown (proprietary system)
Query Processing Overview

• **User Interface Generation**
  – usability issues (Qurk)
  – sourcing sourcing objects across tables (CrowdDB)

• **Optimizations**
  – what to optimize for
  – gaming the market

• **Crowd-sourced operators**
  – graph search (Scoop)
  – sorting
  – ...
UI Generation

• Almost support generation of UIs
  – Qurk: special templates for different tasks
  – CrowdDB: generate UIs from schema
  – Deco: generate UIs from fetch rules (?)
  – Hlog: pragmas in crowd-sourcing rules
  – Freebase: dedicated UIs for specific tasks

• Almost all systems are semi-automatic
  – allow developer to tweak pre-generated UI
CrowdDB Examples

Crowd Columns

Please fill out the missing company data!
Name: IBM
Headquarter address:
Submit

with foreign keys

Crowd Tables

Please fill out the missing professor data
Name: Carey
E-Mail
Department: CS
Submit

Please find a professor and fill in her data
Name: 
E-Mail
Department
Submit

with foreign keys

Please fill out the missing professor data
Name: Carey
E-Mail
Department: CS
Submit

with predicate push-down

Please fill out the missing department data
Name:
Phone:
Submit

denormalized
Some Best Practices for UIs

• Have output fields first, then input fields (CrowdDB)
  – do not mix input/output fields in a single form
  – output fields provide context for the expected input

• Have a “none of the above” button
  – otherwise, workers enter garbage when they are lost

• Worker fatigue for low selectivity predicates (Qurk)
  – system adds artificial “positives” periodically
  – worker does not get into the rhythm of “no” all the time
  – helps with quality control

• Play games: e.g., ESP
  – $ not always the best incentive

• Still a great deal of research to do!!!
Query Optimization

• What to optimize for?
  – cost ($)
  – response time
  – precision (certainty that answers are correct)

• Other “online” considerations
  – whole query result
  – some results with high confidence
  – all results (no confidence guarantees)

• Why is this difficult?
  – crowd gives little guarantees about time, precision
  – no good cost model exists for the crowd
Batching

• **Idea:** While you at it...
  – horizontal: ask for birth date and place at once
  – vertical 1: ask for birth date of several professors
  – vertical 2: three-way comparisons for sort

• **Gives rise to prefetching and new algorithms**
  – merge join >> quicksort?
  – source info even though not relevant for query

• *(Applicable to almost all systems)*
Batching

- Idea: While you at it...
  - horizontal: ask for birth
  - vertical 1: ask for birth
  - vertical 2: three-way cross drill
- Gives rise to prefetching
  - merge join >> quicksort
  - source info even though
- (Applicable to almost all)
Where to invest?

- Get the name of two German composers

<table>
<thead>
<tr>
<th>name</th>
<th>country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beethoven</td>
<td>NULL</td>
</tr>
<tr>
<td>Mozart</td>
<td>NULL</td>
</tr>
<tr>
<td>Gershwin</td>
<td>USA</td>
</tr>
</tbody>
</table>

- Two alternative strategies: which is better?
  - crowd-source the country of known composers
  - crowd-source new (German) composers
Gaming the Platform: Examples

- **Set price according to “competition”**
  - a penny more per task might not help
  - wait for the right moment
  - high rewards attract spammers

- **Understand how tasks are published & found**
  - AMT search ranks by #hits per group
  - that parameter can be controlled by system
% HITs Completed vs. Price

[Franklin et al, SIGMOD 2011]
%With $\geq 1$ Assignment Completed

[Franklin et al, SIGMOD 2011]
Response Time vs. Group Size

[Franklin et al, SIGMOD 2011]
HITs Completed per Group

[Franklin et al, SIGMOD 2011]
Managing Quality

Tim Kraska
Query Optimization

• What to optimize for?
  – cost ($)
  – response time
  – quality/precision (certainty that answers are correct)

• Other “online” considerations
  – whole query result
  – some results with high confidence
  – all results (no confidence guarantees)

• Why is this difficult?
  – crowd gives little guarantees about time, precision
  – no good cost model exists for the crowd
Query Optimization

- What to optimize for?
  - cost ($)
  - response time
  - quality/precision (certainty that answers are correct)
- Other “online” considerations
  - whole query result
  - some results with high confidence
  - all results (no confidence guarantees)
- Why is this difficult?
  - crowd gives little guarantees about time, precision
  - no good cost model exists for the crowd
How Can You Trust the Crowd?
How Can You Trust Your Employee(s)?

CAROL, YOU'RE THE WORST SECRETARY EVER. I HAVE TO LET YOU GO.

YOU CAN'T FIRE ME UNTIL YOU FILL OUT THE INVOLUNTARY TERMINATION FORM 904-B.

CAN YOU GET ME ONE OF THOSE?

YEAH. I'LL GET RIGHT ON THAT.
Outline

• General Techniques
  – Approval Rate / Demographic Restrictions
  – Qualification Test
  – Gold Sets/Honey Pots
  – Redundancy
  – Verification/Review
  – Justification/Automatic Verification

• Query Specific Techniques

• Worker Relationship Management
Outline

• General Techniques
  – Approval Rate / Demographic Restrictions
  – Qualification Test
  – Gold Sets/Honey Pots
  – Redundancy
  – Verification/Review
  – Justification/Automatic Verification

• Query Specific Techniques

• Worker Relationship Management
# Approval Rate & Demographic Restrictions

<table>
<thead>
<tr>
<th>Classify text about consumer electronics</th>
<th>View a HIT in this group</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requester:</strong> Buzz Evaluation</td>
<td><strong>HIT Expiration Date:</strong> Sep 7, 2011 (1 week 6 days)</td>
</tr>
<tr>
<td><strong>Reward:</strong> $0.02</td>
<td><strong>Time Allotted:</strong> 20 minutes</td>
</tr>
<tr>
<td><strong>HITs Available:</strong> 3966</td>
<td><strong>Description:</strong> Classify text for positive, negative, mixed or neutral tone</td>
</tr>
<tr>
<td><strong>Keywords:</strong> buzz, classify, coding, tag, sentiment, text, analysis, twitter, blog, social</td>
<td></td>
</tr>
</tbody>
</table>

**Qualifications Required:**
- HIT approval rate (%) is not less than 95
- Location is US

**Pros:**
- Easy to setup
- Transparent

**Cons:**
- Easy to defeat
- Causes a lot of trouble
Turker Affinity and Errors

[Franklin, Kossmann, Kraska, Ramesh, Xin: CrowdDB: Answering Queries with Crowdsourcing. SIGMOD, 2011]
Turker Affinity and Errors

[Franklin, Kossmann, Kraska, Ramesh, Xin: CrowdDB: Answering Queries with Crowdsourcing. SIGMOD, 2011]
Approval Rate

HIT Group » I recently did 299 HITs for this requester.... Of the 299 HITs I completed, 11 of them were rejected without any reason being given. Prior to this I only had 14 rejections, a .2% rejection rate. I currently have 8522 submitted HITs, with a 0.3% rejection rate after the rejections from this requester (25 total rejections). I have attempted to contact the requester and will update if I receive a response. Until then be very wary of doing any work for this requester, as it appears that they are rejecting about 1 in every 27 HITs being submitted. posted by ...

fair:2 / 5   fast:4 / 5   pay:2 / 5   comm:0 / 5

[Franklin, Kossmann, Kraska, Ramesh, Xin: CrowdDB: Answering Queries with Crowdsourcing. SIGMOD, 2011]
Outline

• General Techniques
  – Approval Rate / Demographic Restrictions
  – **Qualification Test**
  – Gold Sets/Honey Pots
  – Redundancy
  – Verification/Review
  – Justification/Automatic Verification

• Query Specific Techniques

• Worker Relationship Management
Qualification

What is a Qualification?
Some HITs are available only to Amazon Mechanical Turk users with certain Qualifications. Requesters can use Qualifications to make sure their HITs are completed by users that have demonstrated their ability to give high quality answers. You can obtain a Qualification by browsing or searching through the available Qualifications and requesting ones that appeal to you. Qualifications related to your performance on HITs are assigned automatically and cannot be requested. Some Qualifications may require you to complete a test before they are granted. Qualifications requiring you to complete a test must be completed within the specified time.

Qualifications
1-10 of 6479 Results
Sort by: Qualification Name (A-Z) +

<table>
<thead>
<tr>
<th>Qualification Name</th>
<th>Author</th>
<th>Qualified Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebReviews Qualification Master Test</td>
<td>Francis Dierick</td>
<td>0</td>
</tr>
<tr>
<td>&quot;Audio Transcript Verification II&quot;</td>
<td>Spencer Lord</td>
<td>2351</td>
</tr>
<tr>
<td>&quot;Audio Transcript Verification&quot;</td>
<td>Spencer Lord</td>
<td>1521</td>
</tr>
<tr>
<td>&quot;Find the Best Wikipedia Page for this Phrase&quot; Qualification</td>
<td>Crowd Task</td>
<td>11561</td>
</tr>
<tr>
<td>&quot;Headshot&quot; Image Qualifier</td>
<td>Fred Graver</td>
<td>15392</td>
</tr>
<tr>
<td>&quot;Headshot&quot; Qualifier</td>
<td>Fred Graver</td>
<td>4979</td>
</tr>
<tr>
<td>&quot;Powerset knowledge semantic resources qualification II&quot;</td>
<td>Microsoft</td>
<td>714</td>
</tr>
</tbody>
</table>

Show all details | Hide all details

1 2 3 4 5 > Next >> Last
Qualifications (ctd)

Test your knowledge of English and German. Choose the best translation! Be careful, you will get the qualification only if you do not make any mistakes.

Before going to the opening ceremony, Peter took a shower.

- Bevor er in zur Einweihungsfeier ging, nahm Peter eine Dusche.
- Bevor er zudem Eröffnungsfeier ginge, Peter nahm eine Dusche.
- Bevor er zur Eröffnungsfeier ging, nahm Peter eine Dusche.
- Vor der Einweihungsfeier nahm Peter eine Dusche.

No matter how much Julia wanted to control herself, she couldn’t help trembling of fear.

- Egal wie sehr sich Julia beherrschen wollte, sie konnte nicht anders, als vor Angst zu schreien.
- Egal wie sehr sich Julia beherrschen wollte, sie konnte nicht anders, als vor Angst zu zittern.
- Es war umsonst, wie Julia beherrschen wollte, sie konnte nicht anders, als vor Angst zu zittern.
- Egal wie sich Julia beherrschen wollte, sie konnte nicht umhin, vor Angst zu schreien.
Qualifications (ctd)

• Pros
  – Very powerful
  – Can be used for training (context, knowledge, ...)

• Cons
  – May turn off workers, hurt completion time
  – Not always applicable (e.g., opinions/judging)
  – Good qualification choice is hard
Outline

• General Techniques
  – Approval Rate / Demographic Restrictions
  – Qualification Test
  – **Gold Sets/Honey Pots**
  – Redundancy
  – Verification/Review
  – Justification/Automatic Verification

• Query Specific Techniques

• Worker Relationship Management
Gold sets / Honey Pots

- Gold derived from
  - Experts
  - Crowd using high quorum
- Interject trap questions
- Block users in trap and invalidate answers
- Pros
  - Often very effective
  - Cost efficient
- Cons
  - Not always applicable
  - Digging gold is hard
Defeating Honey Pots: reCAPTCHA
\[ \text{Accuracy}_{gold} = 1 - \frac{N_{\text{missed}}}{N_{\text{shown}}} \]

- Show Task (Gold or Normal)
- Provide Feedback for Gold
- Worker can complain about gold answer
  - Warn user if Accuracy < \( t_{\text{warn}} \)
  - Block user if Accuracy < \( t_{\text{reject}} \)

CrowdFlower: Generating Gold

1. Identify worker errors to manual audits
2. Define a set of data-mutation functions
3. Collect set of units with known correct answers
4. Use gold target distribution (Le et al. 2010) to create more gold

<table>
<thead>
<tr>
<th>4 iterations</th>
<th>Gold</th>
<th>Units</th>
<th>Accuracy</th>
<th>MAX work/worker</th>
<th>Gold Ratio</th>
<th>Time (hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 manual</td>
<td>213</td>
<td>95.0%</td>
<td>100</td>
<td>1:10</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>60 programmatic</td>
<td>3,004</td>
<td>89.0%</td>
<td>1,000</td>
<td>1:17</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>208 programmatic</td>
<td>3,103</td>
<td>94.0%</td>
<td>1,500</td>
<td>1:8</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>233 programmatic</td>
<td>15,509</td>
<td>92.5%</td>
<td>1,500</td>
<td>1:7</td>
<td>27</td>
<td></td>
</tr>
</tbody>
</table>

Outline

• General Techniques
  – Approval Rate / Demographic Restrictions
  – Qualification Test
  – Gold Sets/Honey Pots
  – **Redundancy**
  – Verification/Review
  – Justification/Automatic Verification

• Query Specific Techniques

• Worker Relationship Management
Redundancy: Quorum Votes

• Replicate the task (e.g., 3 times)
• Use majority to determine the right value
• Pros:
  – Easy to implement
  – Hard to defeat
• Cons:
  – Not always applicable (opinions, etc.)
  – Increased cost
  – masking cases of ambiguity or diversity, “tail” behaviors
  – Does not cover bias
CrowdDB – Simple Quorums

[ Carey, mjcarey@ics.uci.edu, Computer Science, (949) 824-7427 ]

MTJoin
(Dep)
p.dep = d.name

MTProbe
(Professor)
name=Carey

Please fill out the missing department data

<table>
<thead>
<tr>
<th>Department</th>
<th>Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>Phone</td>
<td>(949) 824-7427</td>
</tr>
</tbody>
</table>

Submit

[(949) 824-7427],
[949 - 82 47 42 7],
[949 - 12 34 56 7]

Please fill out the missing professor data

<table>
<thead>
<tr>
<th>Name</th>
<th>Carey</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-Mail</td>
<td><a href="mailto:mjcarey@ics.uci.edu">mjcarey@ics.uci.edu</a></td>
</tr>
<tr>
<td>Dep</td>
<td>Computer Science</td>
</tr>
</tbody>
</table>

Submit

[mjcarey@ics.uci.edu, Computer Science],
mjcarey@ics.uci.edu, Electrical Engineering],
carey@bea.com, Computer Science

[Franklin, Kossmann, Kraska, Ramesh, Xin: CrowdDB: Answering Queries with Crowdsourcing. SIGMOD, 2011]
Disagreement

• Examples:
  – All different answers
  – 3 out of 5 agree on answer A, the other 2 on answer B

• Counter-Technique:
  – Use experts to break ties
  – Dynamically collect more judgments
Improving Quorums – Labeling tasks

Inter-rater reliability Measures

• Consensus estimates
  – Purpose: Exact agreement
  – Statistics: Percent agreement, Cohen’s kappa, Scott's Pi, ...
Percent agreement

100 names, 2 raters:
Labels: “yes” -> is a company
   “no”-> is not a company

Percentage Agreement
Pr(a)=(40+30)/100 = 0.70
Cohen’s kappa

100 names, 2 raters:
Labels: “yes” -> is a company
“no”-> is not a company

Percentage Agreement Pr(a)
Pr(a)=(40+30)/100 = 0.70

Rater A says “Yes” 60% of the time
Rater B says “Yes” 50% of the time

Pr(e) = Probability of random agreement
 = (0.6*0.5) + (0.4*0.5) = 0.5

$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)}$$

$$K = (0.7 - 0.5) / (1 - 0.5) = 0.4$$
Improving Quorums – Labeling tasks

Inter-rater reliability Measures

• Consensus estimates
  – Purpose: Exact agreement
  – Statistics: Percent agreement, Cohen’s kappa, Scott's Pi, ...

• Consistency estimates
  – Consistently apply a scoring rubric
  – Statistics: Pearson’s r, Spearman’s rho, ....

• Measurement estimates
  – Goal is the preserve as much information as possible from each judge and to incorporate that information into the model
  – Statistics: Principal components analysis, Generalizability theory

Probalistic Graphical Model

- Presence/Absence of an object in the image
- Annotator noise
- Vector of (hidden) attributes
- Individual component weight
- Individual threshold
- Label for image I from annotator j
- A version of the signal corrupted by annotator and image specific noise

Probalistic Graphical Model

Taks: Select images containing at least one duck

Identified 3 groups of annotators
- Group 1: differentiates ducks from everything else
- Group 2: ducks and grebes from everything else
- Group 3: ducks, grebes, and geese from everything else

75.4% correct
Majority vote: 68.3% correct

Outline

• General Techniques
  – Approval Rate / Demographic Restrictions
  – Qualification Test
  – Gold Sets/Honey Pots
  – Redundancy
  – **Verification/Review**
    – Justification/Automatic Verification

• Query Specific Techniques
• Worker Relationship Management
Verification

- Pros
  - Very effective
  - Applicable for a variety of use cases
  - Hard to cheat
- Cost
  - Expensive (time + cost)
  - Bias
Soylent: Multi-Level Review

Automatic clustering generally helps separate different kinds of records that need to be edited differently, but it isn’t perfect. Sometimes it creates more clusters than needed, because the differences in structure aren’t important to the user’s particular editing task. For example, if the user only needs to edit near the end of each line, then differences at the start of the line are largely irrelevant, and it isn’t necessary to split based on those differences. Conversely, sometimes the clustering isn’t fine enough, leaving heterogeneous clusters that must be edited one line at a time. One solution to this problem would be to let the user rearrange the clustering manually, perhaps using drag-and-drop to merge and split clusters. Clustering and selection generalization would also be improved by recognizing common text structure like URLs, filenames, email addresses, dates, times, etc.

Soylent: Find-Fix-Verify

- Programming crowds today is haphazard, similar to UI technology before design patterns like Model-View-Controller
- Find-Fix-Verify is a design pattern for programming crowds to complete open-ended tasks

Soylent: Find-Fix-Verify

**Find**

"Identify at least one area that can be shortened without changing the meaning of the paragraph."

**Fix**

"Edit the highlighted section to shorten its length without changing the meaning of the paragraph."

**Verify**

"Choose at least one rewrite that has style errors, and at least one rewrite that changes the meaning of the sentence."

Outline

• General Techniques
  – Approval Rate / Demographic Restrictions
  – Qualification Test
  – Gold Sets/Honey Pots
  – Redundancy
  – Verification/Review
  – Justification/Automatic Verification

• Query Specific Techniques

• Worker Relationship Management
Other

• Justification/Feedback/Comments
  – Helps to detect ambiguities/unclarity
  – Can be used by Review tasks to determine if the worker really tried
  – NOTE: If you don’t put a comment box, they will write them somewhere!

• Automatic Checks
  – Applicable in cases where finding a solution is hard but verifying is easy
  – Example: Protein Folding (http://fold.it/)
Outline

• General Techniques
  – Approval Rate / Demographic Restrictions
  – Qualification Test
  – Gold Sets/Honey Pots
  – Redundancy
  – Verification/Review
  – Justification/Automatic Verification

• Query Specific Techniques

• Worker Relationship Management
Cost vs. Quality

[Image: Two graphs showing the relationship between the number of tasks completed and the frequency of inaccuracies, respectively, across different pay per task levels ($0.00$, $0.01$, $0.05$, $0.10$) for different image datasets (2, 3, 4 images).

Quality as an Optimization Goal

Comparison Sort

Rating Sort

User Interface vs. Quality

MTJoin (Professor)
\( p.name = "carey" \)

MTJoin (Dep)
\( p.dep = d.name \)

MTProbe (Dep)
Please fill out the missing professor data
Name: Carey
Department: CS
E-Mail: 

MTProbe (Professor)
Please fill out the missing department data
Name: Carey
Phone: 

MTProbe (Professor, Dep)
Please fill out the missing professor data
Name: Carey
E-Mail: 
Department: 
Phone: 

\( \sigma \) name="Carey"

\( \times \) p.dep=d.name

Professor
Department

≈10% Error-Rate
[Franklin, Kossmann, Kraska, Ramesh, Xin: CrowdDB: Answering Queries with Crowdsourcing. SIGMOD, 2011]
Outline

• General Techniques
  – Approval Rate / Demographic Restrictions
  – Qualification Test
  – Gold Sets/Honey Pots
  – Redundancy
  – Verification/Review
  – Justification/Automatic Verification

• Query Specific Techniques

• Worker Relationship Management
Paying BAD Turkers?

• Pay for bad work?
  – Pros:
    • Less hassle with turker management
    • better requester reputation
    • Often not sure, who is wrong (requester/turker)
  – Cons:
    • Reputation for paying everything
    • undermines reputation system

• Use bonus as incentive instead of rejections
• Spammers -> Block from future tasks
Worker Relationship Management
Worker Relationship Management (ctd)

Wait, what? 57 states? Did Texas re-divide into territories again or something?

----------

It's a reference to a gaffe Barack Obama made when he was a bit too tired from the campaign trail. It's funny how people keep referring to that instead of his more flagrant mistakes, like continuing the misadventures in Iraq and Afghanistan, throwing money at Wall Street instead of investing it in infrastructure, and treating Republicans with respect.

----------

Barry's teleprompter told him there were 57 states. Barry says whatever the teleprompter tells him to say because when Barry wanders off script problems soon follow. There are other funny gaffes out there. Youtube has a bundle of them.

----------

State of Confusion, State of Mind, State of DeNile....

----------

Uh-oh. Politics has enter the thread. Just watch the discussion disintegrate into name calling.

----------

Yeah, I guess he "misunderestimated" the difficulty of speaking in public, unlike that huge intellect who declared "Mission Accomplished" way back when...
Worker Relationship Management (ctd)

Check out Barry referring to a guy in the military as a "corpseman". Repeatedly. Way to go, big guy. You wonder if he picked up those vocabulary skills in Indonesia as a kid or if it just comes naturally to his massive "intellect". You know, that intellect that prevented him from writing anything as president of the law review at Havahd.

Clark: Nah, I'm not going to call anyone here names. I like everyone here and respect the help I've received from the posters. I'll toss a few lightening bolts, though.

-------------

Okay, back to the reason for these HITs - some of you gave some possible answers so I guess there could be a legitimate use for the data. Some of these HITs really make me wonder, though....

-------------

As amusing as it might be to some people that "corpse" and "corps" are homophones, he was referring to him as a corps man which is technically accurate. While it is awkward it isn't laugh out loud funny compared to the gems we heard from Bush and from some of the people campaigning for president right now.

-------------

Just fyi, core and corps are homophones. The "corps of cadets" is pronounced "core of cadets." The "press corps" is "press core". When used as a plural, you'd say the s on the end, but unlike
Other practical tips

• “Eat your own dog-food”
• Monitor discussion forums
  – Turker Nation: http://turkers.proboards.com
  – http://www.turkalert.com #
  – Turkopticon: report/avoid shady requestors
  – Amazon Forum for Mturk
• Address feedback
• Make the process as transparent as possible (payment, validation, intent, ...)

145
Crowdsourced Data Management Applications

● Relational
  – information extraction
  – schema matching
  – entity resolution
  – data spaces
  – building structured KBs
  – sorting
  – top-k
  – ...

● Beyond relational
  – graph search
  – classification
  – mobile image search
  – social media analysis
  – question answering
  – NLP
  – text summarization
  – sentiment analysis
  – semantic wikis
  – ...

146
Human-Assisted Graph Search: It’s Okay to Ask Questions, A. Parameswaran, A. Das Sarma, H. Garcia-Molina, N. Polyzotis, J. Widom. VLDB-11
Human Assisted Graph Search

- Given a DAG $G$
  - Containing unknown target nodes
  - Find target nodes by asking humans search questions at nodes in $G$
    - “Is there a target node reachable from the current node?”

- Applicable to many problems
  - classification, workflow debugging, filter synthesis, interactive search
Example: Classifying Images into a Taxonomy

To classify image of an accord car

**Diagram:**
- **Vehicle**
  - **Car**
    - **Nissan**
      - **Maxima**
      - **Sentra**
    - **Honda**
    - **Toyota**

**Questions:**
- Is the image a type of vehicle? **YES!**
- Is the image a type of Toyota? **NO!**
- Is the image a type of Honda? **YES!**

**Target Node:**

Is the image a type of X? = Is the target node reachable from X?
Classification Problem Setting

- Don’t want to ask all nodes or serially
  - too expensive or too slow
- Given a limit of k questions, find the best k nodes to ask in parallel
  - Assuming humans answer correctly, processing k answers yields a superset of the target nodes
  - we want to minimize the size of this superset
- Practical usage
  - to classify into a huge taxonomy, first ask k questions
  - is the superset small enough that some in-house humans can do a final processing?
  - If not, ask next k questions, and so on
Solution

- Given a limit of \( k \) questions, find the best \( k \) nodes to ask in parallel

- Reduces to a partition problem
  - given an undirected tree, find \( k \) edges such that their deletion minimizes the size of the largest connected component

- Runs in \( O(n \log n) \) time (\( n = \# \) of nodes)

- Paper considers many other practical settings
  - multiple target nodes, unlimited resources, etc.
Mobile Image Search

CrowdSearch: Exploiting Crowds for Accurate and Real-Time Image Search on Mobile Phones. T. Yan, V. Kumar, D. Ganesan. MobiSys-2010
Mobile Image Search

Automated image search engine
CrowdSearch: Use Crowdsourcing

Yes, Yes, No, No, No

No, Yes, Yes, Yes, Yes

Yes, No, Yes, Yes, No

Automated image search engine
Money vs. Delay Tradeoffs

- **To minimize money**
  - send tasks sequentially
  - increases delay

- **To minimize delay**
  - send tasks in parallel
  - increases money

- **Goal: return one validated image before deadline, while minimizing money**

Yes, Yes, No, No, No

No, Yes, Yes, Yes, Yes

Yes, No, Yes, Yes, No
Key Idea

- Send out the first task

- Suppose has received No, Yes, No
- Need to receive Yes, Yes in order to have a match
- Compute $P(\text{Yes, Yes} \mid \text{No, Yes, No})$
- Compute $P(\text{receive the above seq before deadline})$
- If their product $< \text{threshold}$ ➔
  first task likely to fail, send out the second task

- And so on ...
Probability Estimation

- $P(\text{Yes, Yes} \mid \text{No, Yes, No})$
- $P(\text{receive the above seq before deadline})$

- Model the crowd behavior
  - what is the probability that a user will answer Yes, Yes, given No, Yes, No?
  - given a task, how long would it take until a user starts answering?
  - given an answer on a task, how long until the next answer arrives?

- Use training data to estimate model parameters
- Use the model to estimate the above probabilities
Event Detection and Monitoring in Social Media

Work done and deployed at Kosmix, now @WalmartLabs
Generating tsunami that hit wide areas of the Pacific coast. The quakes reportedly caused many injuries and fatalities.

Moment of the Japan quake

Video captured the moment the 8.9-magnitude earthquake hit Japan.

Added Mar. 14, 2011

705 tweets in last 5 minutes
2,461,804 tweets from 1,327,142 people since 5 days ago viewed 63297 times

jendubh06 someone got mad at me cuz i donated $25 to the japan animal rescue fund for animals that were injured/homeless since the tsunami

jennifer lee: my name is jen and i'm an emt in orange county, ca. message me if you wanna k... (9 followers)
12 mins ago via web | 1 similar tweets

@dimse2u Me: do u know what happened in Japan? Torahim? Was that a tsunami? Is that a tsunami? He thought I was talking about his nanol

@jenniazure Does it make sense 2 U? (22 followers)
10 mins ago via Twitter for BlackBerry?? | 1 similar tweets

@lissabieber @PrayForJapan everyone... LOVE AND PEACE ARE SENT IN OUR PRAYERS &;)
Lissa Bieber: Justin Bieber is my inspiration and my idol. I am a Fieber who has been the... (94 followers)
10 mins ago via web | 1 similar tweets

@KushBST #PRAYFORJAPAN wtf get this back on lil worldwide we don't need to lose a country
Wagavamp is all about the in under 160 characters: S, well... god there is so much to... (419 followers)
10 mins ago via web | 1 similar tweets

Popular Headlines

Japan Earthquake: before and after
1 hr ago via aljazeera.net | 514746 similar tweets

Satellite Photos - Japan Before and After Tsunami - Interactive Feature - NYTimes.com
1 hr ago via nytimes.com | 165554 similar tweets

2011 Japanese Earthquake and Tsunami
23 hrs ago via google.com | 73562 similar tweets

Japan Earthquake & Tsunami: 7 Simple Ways to Help
1 day ago via mosholic.com | 39122 similar tweets

Popular Tweets
Twitter: Egypt in Real-Time

Mehrnush b/c 8 MILLION protesters nationwide are minority. RT @atlnav
Dr Kamel says the people in Tahrir Square do not represent the majority. #egypt

half min ago

WorldPeace2Day Prayers RT @Anony_Ops: Today a dark day 4journalism in #Egypt. Journalist Ahmed Mahmoud dead, AlJazeera's licence revoked, Cairo station burned

1 min ago

wadt OH and we say Happy Friday to all, Peace to all who yearn 4 peace, May Freedom and true Democracy

tweetbeat
Event Detection

**Sample heuristics**

- exploit keyword popularity & correlations
  - Egypt/revolt, Egypt/protest, Egypt/Tahrir
- exploit foursquare checkins
- exploit Twitter users who often tweet upcoming events
  - cnnbrk
Event Detection: Add Crowdsourcing

- For each event
  - find users who tweet about the event
  - if do not find enough users, drop event
- Can be viewed as “implicit crowdsourcing”
Event Dection: Add Crowdsourcing

- For each event, show
  - phrase pairs: Egypt/revolt, Egypt/protest, Arab/spring
  - sample tweets, most popular URLs
- Ask: is this an interesting event?
- Different populations have different quality
  - show “high-quality” events first on Tweetbeat
Event Monitoring: Current Solutions

**Manually write rules to match tweets to events**
- e.g., tweet contains certain keywords / userids → positive
- conceptually simple, relatively easy to implement
- often achieve high initial precision

**Limitations**
- expensive, don’t scale
- manually writing good rules can be hard
- rules often become invalid/inadequate over time
  - e.g., Baltimore shooting → John Hopkins shooting
Event Monitoring: Kosmix Solution

Event: Baltimore shooting

Twitter firehose

Initial profile: \{Baltimore, shoot\}

New profile: \{Baltimore, shoot, John Hopkins\}

Learning algorithm

Tweets:

- “Baltimore shooting on TV5!”
- ...

Event Monitoring: Kosmix Solution
Major Problem 1: Constructing Good Initial Profile

- **Event as output by event detector:**
  - phrase pairs: Egypt/revolt, Egypt/protest, Arab/spring
  - sample tweets, most popular URLs
- **Can automatically infer initial profile from these**
- **But accuracy is limited ⇒ difficult to learn from**
- **Solution: use crowdsourcing**
  - ask “is this an interesting event?”
  - if yes, ask “list people/organization/location involved”
  - use answers to craft initial profile
  - turned out the best profile should include people/org/loc
Second Major Problem: Preventing Overgeneralization

• Also known as snowballing
  – start with “Baltimore shooting”
  – soon learn profile subsuming all current events in Baltimore

• Well-known serious problem in machine learning

• Solution: use crowdsourcing
  – build multiple levels of generalization into learning algo
  – humans monitor tweet streams of an event
  – if detect snowballing, alert algorithm
  – algorithm automatically backtracks, and moves to a more conservative level of generalization
Summary: The Critical Roles of Crowdsourcing in Three Places
Research Challenges

Tim Kraska
Ground Breaking Changes

with different intents/motivation
Ground Breaking Changes (ctd)

Closed-World

Open-World
Ground Breaking Changes (ctd)
Ground Breaking Changes (ctd)

#include <iostream.h>

int main(){
    int num1;
    cout << "Welcome to Tom's Odd and Even Evaluator Game!!\n\n";
    cout << "Enter an integer\n\n";
    cin >> num1;
    if ( num1 % 2 == 0 )
        cout << "\nGuess what?\n\n" << num1
            << " is an even number!\n\n";
    if ( num1 % 2 != 0 )
        cout << "\nGuess what?\n\n" << num1
            << " is an odd number!\n\n";
    return 0;
}
Ground Breaking Changes (ctd)
Research Agenda

- Data Model/Query Languages
- Query Optimization
- Query Execution
- Quality Control
- Storage/Caching
- User interfaces
- New sort/join algorithms Algorithms

- Turk behavior
- Worker Relationship Management
- Interactivity
- Platforms
- Hybrid Machine Learning/
Human Algorithm
- Repeatability
- .....
Repeatable Experiments

Avg. #HITS

5 minute Interval

200 HITs

- $0.01
- $0.02
- $0.03
- $0.04
- $0.05
DB $\rightarrow$ Crowdsourcing++

Crowd-Hard Problems:

- **Programming Language: GUI**
  Question design, ambiguities, granularity, ...
- **Many, many knobs to turn**
  Price, replication factor, HIT group size, expiration time,....
- **Changing platform behavior**
  Increasing market size, new platform features,...
- **Jungle of different techniques**
  Rejection policy (Quorum-Vote, Test-Set,...), Quality control (Quorum, iterative models,...)
- **Learning effects / Community Management**
- ...

The DB-Approach

- **Data independence**
  If HW changes, app need not change
- **DBMS optimizes queries**
  - Decide what to crowdsource
  - Statistics about the market place, question ordering,...
Research Agenda

- Data Model/Query Languages
- Query Optimization
- Query Execution
- Quality Control
- Storage/Caching
- User interfaces
- New sort/join algorithms Algorithms

- Turker Behavior
- Worker Relationship Management
- Interactivity
- Platforms
- Hybrid Machine Learning/Human Algorithm
- Repeatability
- .....
References

Other Tutorials

Motivating Examples
- Nasa Clickworkers, http://clickworkers.arc.nasa.gov/top
- Foldit, http://fold.it/
- Galaxy Zoo, http://www.galaxyzoo.org/
- Stackoverflow, http://stackoverflow.com/

Taxonomy
  - www.crowdsourcingresults.com

Platforms
- Amazon Mechanical Turk, https://www.mturk.com/
- SamaSource, http://samasource.org/
- CrowdFlower, http://crowdflower.com/
References ctd.

Crowdsourcing DBMSs

• CrowdDB
  – Amber Feng, Michael Franklin, Donald Kossmann, Tim Kraska, Sam Madden, Sukriti Ramesh, Andrew Wang, and Reynold Xin: CrowdDB: Query Processing with the VLDB Crowd. VLDB 2011
  – Michael Franklin, Donald Kossmann, Tim Kraska, Sukriti Ramesh, and Reynold Xin: CrowdDB: Answering Queries with Crowdsourcing. SIGMOD 2011

• Deco

• Qurk:
  – Marcus, E. Wu, D. Karger, S. Madden, and R. C. Miller: Human-powered Sorts and Joins. PVLDB 2012
  – A. Marcus, E. Wu, D. Karger, S. Madden, and R. C. Miller: Crowdsourced Databases: Query Processing with People. CIDR 2011

• Hlog
  – Predicates.X. Chai, B. Vuong, A. Doan, J. F. Naughton: Efficiently incorporating user feedback into information extraction and integration programs. SIGMOD 2009

• FusionCOMP
References ctd.

Relational Applications

- Hybrid Data Integration:
- Information Extraction:
- D. Deutch, O. Greenshpan, B. Kostenko, T. Milo: Using Markov Chain Monte Carlo to play Trivia. ICDE 2011

Beyond Relational

- T. Yan, V. Kumar, D. Ganesan: CrowdSearch: Exploiting Crowds for Accurate and Real-Time Image Search on Mobile Phones. MobiSys-2010
- Hybrid Image Search:
References ctd.

Quality

- John Le, A. Edmonds, V. Hester1, and L. Biewald: Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution. CSE 2010
- S. Stemler: A comparison of consensus, consistency, and measurement approaches to estimating interrater reliability. Practical Assessment, Research & Evaluation 2004
- M. Bernstein, G. Little, R.C. Miller, B. Hartmann, M. Ackerman, D. R. Karger, D. Crowell, K. Panovich: Soylent: A Word Processor with a Crowd Inside. UIST 2010
- W. Mason, D. J. Watts: Financial Incentives and the 'Performance of Crowds’, HCOMP 2009