Crowdsourcing: Challenges and Opportunities

Guoliang Li
Tsinghua University
Tutorial Objectives

- What is crowdsourcing?
- How and when to use crowdsourcing?
- How to do experiments for crowdsourcing?
- What are research challenges of crowdsourcing?
Tutorial Outline

- Introduction
- Applications
- Platforms
- Challenges
- Opportunities
What is Crowdsourcing?

- **Outsourcing** – 外包
  - A known agent (an employee)

- **Crowdsourcing** – 众包
  - An undefined, generally large group of people via a group call
  - The application of open source principles to fields outside of software

$ / \text{Customer}$

Large Businesses

# of Customers

Crowdsourcing @ HotDB2012 (4)
Crowdsourcing Definition

- Coordinating a **crowd** (a large group of people on the web) to do **micro-work** (small contributions) that solves **problems** (that software or one user can’t do)
- A collection of mechanisms and associated methodologies for scaling and directing crowd activities to achieve goals

Evolving & broadly defined
Example - Captcha

- Captcha: 200M every day
- ReCaptcha: 750M to date

Machine Translation

Problem:
- Manual evaluation on translation quality is slow and expensive

Crowdsourcing:
- Low cost of non-experts, $0.10 to translate a sentence
- High agreement/equivalent quality between non-experts and experts
- Complex tasks like human-mediated translation edit rate

- B. Bederson et al. Translation by Interactive Collaboration between Monolingual Users, GI 2010
Tingxin Yan, Vikas Kumar, Deepak Ganesan: CrowdSearch: exploiting crowds for accurate real-time image search on mobile phones. MobiSys 2010:77-90
How similar is the artistic style in the paintings above?

- Very similar
- Similar
- Somewhat dissimilar
- Very dissimilar

Human and Machine Detection of Stylistic Similarity in Art.
Adriana Kovashka and Matthew Lease. CrowdConf 2010
Audio Example

- Mobile service that aids blind users with “visual questions” in near-realtime
- An iPhone application

Jeffrey P. Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C. Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samuel White, Tom Yeh.

*VizWiz: nearly real-time answers to visual questions.* UIST 2010 (Best Paper Award)

http://www.cs.rochester.edu/u/jbigham/vizwiz/video/
Use crowd to answer DB queries
- Where to use crowd?
- How to use crowd?
- How to support SQL?
- How to devise a system?
- Quality?

Michael J. Franklin, Donald Kossmann, Tim Kraska, Sukriti Ramesh, Reynold Xin: CrowdDB: answering queries with crowdsourcing. SIGMOD 2011:61-72
Crowdsourcing Overview

- **Requester**
  - People submit tasks

- **Platforms**
  - Task management

- **Worker**
  - People work on tasks

---

Submit tasks → Collect answers

Publish tasks

Find interesting tasks → Return answers
Crowdsourcing vs Human Computation

- **Human Computation**
  - Design a solution using both automated computers and human computers
  - Maybe a closed set of workers
  - Crowdsourcing greatly **facilitates** human computation

- **Social computing**
  - Social behavior

- **Collective intelligence**
  - May not human

*Alexander J. Quinn, Benjamin B. Bederson: Human computation: a survey and taxonomy of a growing field. CHI 2011:1403-1412*
A Growing Field - Tutorials

- WWW 2011: Managing Crowdsourced Human Computation
- VLDB 2011: Crowdsourcing Applications and Platforms
- SIGIR 2011: Crowdsourcing for Information Retrieval: Principles, Methods and Applications
- AAAI 2011: Human Computation: Core Research Questions and State of the Art
- WSDM 2011: Crowdsourcing 101: Putting the WSDM of Crowds to Work for You
- CVPR 2010: Mechanical Turk for Computer Vision
- ECIR 2010: Crowdsourcing for Relevance Evaluation
- HCIC 2011: Quality Crowdsourcing for Human Computer Interaction Research
- CrowdConf 2011: Crowdsourcing for Fun and Profit
A Growing Field - Workshops

- KDD 2010: 2nd Human Computation Workshop - HCOMP 2010
- AAAI 2011: 3rd Human Computation Workshop - HCOMP 2011
- SIGIR 2010: Crowdsourcing for Search Evaluation
- SIGIR 2011: Crowdsourcing for Information Retrieval
- CVPR 2010: Advancing Computer Vision with Humans in the Loop
- NAACL 2010: Creating Speech and Language Data with Amazon's Mechanical Turk
- NIPS 2010: Computational Social Science and Wisdom of the Crowds
- Ubicomp 2010: Workshop on Ubiquitous Crowdsourcing
- WSDM 2010: Crowdsourcing for Search and Data Mining Workshop
- ICWE 2010: Enterprise Crowdsourcing Workshop
- CHI 2011: Workshop on Crowdsourcing and Human Computation
- AMTA 2010: Collaborative Translation Technology, Crowdsourcing and the Translator
- EC 2011: Workshop on Social Computing and User Generated Content

Crowdsourcing @ HotDB2012
Many Related Areas

- Human–Computer Interaction (HCI)
- Artificial Intelligence (AI)
- Machine Learning (ML)
- Information Retrieval (IR)
- Crowd Management (DB)
- Social Science
- Theory
- Statistics
Are there any platforms?
highly-available, cheap, programmable, a prototyping platform for crowd computing
## Tasks

- **What are tasks?**

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

- Cheap, Easy, and Fast
- Ready to use infrastructures
  - Payments, workforce, interface widgets
- Allow early, iterative, frequent trials
  - Test new ideas
- Many successful examples
  - Image search
  - reCaptcha
Human Intelligence Tasks (HITs)

- Human Intelligence Tasks – micro tasks

- Requesters create (HITs)
  - web services API/Dashboard
  - assess results, pay per HIT satisfactorily completed.

- Workers (sometimes called “Turkers”)
  - log in, choose HITs, perform them.

- Currently >200,000 workers from 100 countries
- Millions of HITs completed
The Requester

- Sign up with your Amazon account
- Amazon payments
- Purchase prepaid HITs
- There is no minimum or up-front fee
- AMT collects a 10% commission
- The minimum commission charge is $0.005 per HIT
- Approve/reject answers
Dashboard

- Three tabs
  - Design
  - Publish
  - Manage

- Design tab
  - HIT Template

- Publish tab
  - Make work available

- Manage tab
  - Monitor progress

---

### Edit HIT Template

Specify the properties that are common for all of the HITs created using this template.

#### Enter Properties

<table>
<thead>
<tr>
<th>Template Name:</th>
<th>TREC_binary_e4</th>
<th>This name is not displayed to workers.</th>
</tr>
</thead>
</table>

#### Describe your HIT

- **Title**: Relevance evaluation for news articles
- **Description**: Please help us evaluate relevance for the following document.
- **Keywords**: relevance, news articles, search, TREC, airport security, steel production, C

#### Working on your HIT

- **Time allotted per assignment**: 1 Hour
- **HIT expires in**: 13 Days

#### Worker must meet the following criteria to work on these HITs:

- **HIT approval rate (%)**: > 98%

#### Paying Workers

- **Reward per assignment**: $0.04
- **Number of assignments per HIT**: 5
- **Results are automatically approved in**: 5 Days

---

Crowdsourcing @ HotDB2012 (26)
Amazon Web Services API

- Rich set of services
- Command line tools
- More flexibility than dashboard

- **CreateHIT** (Requirements, Pay rate, Description) – returns HIT Id and HIT Type Id
- **SubmitAssignment** (AssignmentId) – notifies Amazon that this assignment has been completed
- **ApproveAssignment** (AssignmentID) – Requester accepts assignment, money is transferred, also RejectAssignment
- **GrantBonus** (WorkerID, Amount, Message) – Give the worker the specified bonus and sends message, should have a failsafe
- **NotifyWorkers** (list of WorkerIds, Message) – e-mails message to the workers.
Dashboard vs API

- Dashboard
  - Easy to prototype
  - Setup and launch an experiment in a few minutes

- API
  - Ability to integrate AMT as part of a system
  - Ideal if you want to run experiments regularly
  - Schedule tasks
The Worker

- Sign up with your Amazon account

Tabs

- Account: work approved/rejected
- HIT: browse and search for work
- Qualifications: browse and search for qualifications
Why Do Work on MTurk?

- Money ($$$)
- Fun (or avoid boredom)
- Socialize
- Earn acclaim/prestige
- Altruism
- Learn something new (e.g. English)
- Unintended by-product (e.g. re-Captcha)
- Create self-serving resource (e.g. Wikipedia)
- Multiple incentives are typically at work in parallel
Who Are My Workers?

- 2008-2009 studies found less global and diverse than previously thought
  - 47% US, 34% India, 19% others
  - Female
  - Educated
  - Bored
  - Money is secondary
Survey on Workers

- “Mturk money is always necessary to make ends meet.”
  - 5% U.S. 13% India
- “Mturk money is irrelevant.”
  - 12% U.S. 10% India
- “Mturk is a fruitful way to spend free time and get some cash.”
  - 69% U.S. 59% India

![Histogram of Reported Age of Turkers](image1)
![Histogram of Max Income Range](image2)
Mturk

- Advantages
  - More participants
  - More diverse participants
  - High speed
  - Low cost
  - Speed of experimentation
  - Diversity

- Disadvantages/Limitations
  - Lower quality feedback
  - Less interaction
  - Greater need for quality control
  - Less focused user groups
  - No control of users’ environment
  - Not designed for user studies
  - Spam - Uncertainty about user demographics, expertise
  - Lots of problems and missing features

Crowdsourcing != MTurk
Pay-based Marketplaces / Vendors

- Amazon Mechanical Turk (since 2005, www.mturk.com)
- Crowdflower (since 2007, www.crowdflower.com)
- CloudCrowd (www.cloudcrowd.com/)
- DoMyStuff (www.domystuff.com/)
- Livework (https://www.livework.com/)
- Clickworker (www.clickworker.com/)
- SmartSheet (www.smartsheet.com/crowdsourcing)
- uTest (www.utest.com/)
- Elance (www.elance.com/)
- oDesk (www.odesk.com/)
- vWorker (www.vworker.com/)
Microtask Aggregators

CrowdFlower

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.

Crowdsourcing @ HotDB2012 (35)
How and when to use crowdsourcing?
How and when to use crowdsourcing?
When to use crowdsourcing

- Computers cannot do
- A single person cannot do
- The work can be split into small tasks
How to Do Experiments in Crowdsourcing

- Experimental Design
- Choose crowdsourcing platform
- Decompose your tasks into micro tasks
- Publish your tasks and wait for answers
- Aggregate workers’ answers
Tutorial Outline

- Introduction
- Applications
- Platforms
- Challenges
- Opportunities
Research Challenges in Crowdsourcing

- Task Management
  - Task assignment, payment, discover
- Human–Computer Interaction
  - Payment / incentives, interface and interaction design, communication, reputation, recruitment, retention
- Quality Control / Data Quality
  - Trust, reliability, spam detection, consensus labeling
- Human-Processing Unit (HPU) and CPU
  - How to combine
- Scalability
  - Large scale data
Task Management

- How to decompose a complex task?
- How to find tasks for workers?
- How much is the payment of a HIT?
Supporting Complex Tasks

- Mturk works only for small tasks
- How to support complex tasks?
  - Task decomposition – large tasks are divided into small problems
  - Job distributed among multiple workers
  - Collect all answers and combine them
  - Verifying performance of heterogeneous CPUs and HPUs
CrowdForge - MapReduce framework for crowds

- Identify sights worth checking out (one tip per worker)
  - Vote and rank
- Brief tips for each monument (one tip per worker)
  - Vote and rank
- Aggregate tips in meaningful summary
  - Iterate to improve...

- Aniket Kittur, Boris Smus, Robert Kraut: CrowdForge: crowdsourcing complex work. CHI Extended Abstracts 2011:1801-1806
How to Discover Tasks?

- Task discovery is very important.
- Heavy tailed distribution of completion times.
Task Assignment

- **Push Methods**: system ➔ workers
  - The system takes complete control over who is assigned which task.
  - Worker expertise recording for task assignment (employer/task finds worker)

- **Pull Methods**: workers ➔ system
  - The system merely sets up the environment to allow workers to assign themselves (or each other) tasks.
  - Task organization for task discovery (worker finds employer/task)
Task Recommendation

- **Content-Based recommendation**
  - find similarities between worker profile and task characteristics.

- **Collaborative Filtering**
  - make use of preference information about tasks (e.g., ratings) to infer similarities between workers.

- **Hybrid**
  - a mix of content-based and collaborative filtering methods.
Task Payments

- How much is a HIT?
- Delicate balance
  - Too little, no interest
  - Too much, attract spammers
  - Paying a lot is a counter - incentive
  - Money does not improve quality but (generally) increase participation
- Payment based on user effort- Bonus
- Example: $0.04 (2 cents to answer a yes/no question, 2 cents if you provide feedback that is not mandatory)

3 main goals for a task to be done:

- Minimize Cost (cheap)
- Minimize Completion Time (fast)
- Maximize Quality (good)

Many optimization problems to tradeoff the three goals.
Human-Assisted Graph Search/Classification

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

Applications
- Classification, workflow debugging, interactive search

CrowdGraphSearch - Application

- Classify an image

![Diagram showing classification process]

- target node = intended category
  - Is the image a type of X? = Is the target node reachable from X?
CrowdGraphSearch - Optimization

- Do not ask all nodes or serially
  - Too expensive or too slow
- Given a limit of k questions, find the best nodes to ask in parallel
  - Minimize the size
  - To classify into huge taxonomy, first ask k question
  - Is the superset small enough?
CrowdSearch: Using crowd to improve image search

- How to ensure a high enough accuracy, say over 95%?

Tingxin Yan, Vikas Kumar, Deepak Ganesan: CrowdSearch: exploiting crowds for accurate real-time image search on mobile phones. MobiSys 2010: 77-90
CrowdSearch - Optimization

- To minimize money
  - Increase delay

- To minimize delay
  - Increase money

- Goal: return one validate image before deadline, while minimizing the money

Yes, Yes, No, No, No

No, Yes, Yes, Yes, Yes

Yes, No, Yes, Yes, No

Crowdsourcing @ HotDB2012 (54)
CrowdSearch - Optimization

![Diagram with arrows and probabilities]

- Start
- Current time (t)
- Send the next candidate image to validation

Probabilities:
- P(D1) = 0.9
- P(A1) = 0.3
- P(D2) = 0.6
- P(A2) = 0.9

P(success) = 0.66

P_{th} = 0.8

Crowdsourcing @ HotDB2012 (56)
Quality - Example

- Get people to look at sites and classify them as:
  - G (general audience)
  - PG (parental guidance)
  - R (restricted)
  - X (porn)
Quality Control

- Quality of workers’ answers is extremely important part of the experiment
- Approach it as “overall” quality – not just for workers
- Bi-directional channel
  - You may think the worker is doing a bad job.
  - The same worker may think you are a lousy requester.
Methods for Measuring Agreement

- What to look for
  - Agreement, reliability, validity

- Beforehand
  - Qualification test
  - Screening, selection, recruiting, training

- During
  - Accesses labels as workers produce them
  - Reward, penalize, weight

- After
  - Accuracy metrics
  - Filter, weight
Qualification Tests: Pros and Cons

- Qualification test
  - create questions on topics so user gets familiar before starting assessments

- Advantages
  - Great tool for controlling quality
  - Adjust passing grade

- Disadvantages
  - Hard to verify subjective tasks like judging relevance
  - Slows down the experiment, difficult to “test” relevance
  - Extra cost to design and implement the test
  - Try creating task-related questions to get worker familiar with task before starting task in earnest

- No guarantees - Still not a guarantee of good outcome
Quality Control

- **Majority vote**
  - 2 bad, 3 good $\rightarrow$ good
  - 3 bad, 2 good $\rightarrow$ bad

- **Weighted majority vote**
  - Identify workers that always disagree with the majority
  - Lower down the weight of such workers
Dealing with Bad Workers

- Pay for “bad” work instead of rejecting it?
  - Pro: preserve reputation, admit if poor design at fault
  - Con: promote fraud, undermine approval rating system
- Use bonus as incentive
  - Pay the minimum $0.01 and $0.01 for bonus
  - Better than rejecting a $0.02 task
- Worker blocking - spammer “caught”, block from future tasks
  - May be easier to always pay, then block as needed
Emails after **Rejection**

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
Emails after Rejection

- WORKER: this is not fair, you made me work for 10 cents and i lost my 30 minutes of time, power, and lot more and gave me 2 rejections at least you may keep it pending. please show some respect to turkers.

- WORKER: I understood the problems. At that time my kid was crying and i went to look after. that's why i responded like that. i was very much worried about a hit being rejected. The real fact is that i haven't seen that instructions of 5 web page and started doing as i do the dolores labs hit, then someone called me and i went to attend that call. sorry for that and thanks for your kind concern.
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
CrowdFlower: Programmatic Gold

\[ \text{Accuracy}_{\text{gold}} = 1 - \frac{N_{\text{missed}}}{N_{\text{shown}}} \]

- 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 < \( t_{\text{reject}} \)
Quality Control

- As a worker
  - I hate when instructions are not clear
  - I’m not a spammer – I just don’t get what you want
  - A good pay is ideal but not the only condition for engagement

- As a requester
  - A task that would produce the right results and is appealing to workers
  - I want your honest answer for the task
  - I want qualified workers and I want the system to do some of that for me

- Managing crowds and tasks is a daily activity and more difficult than managing computers
Getting input from users is important in HCI

- surveys
- rapid prototyping
- usability tests
- cognitive walkthroughs
- performance measures
- quantitative ratings
Practitioners know that HITs in 3rd page and after, are not picked by workers.

Many such HITs are left to expire after months, never completed.

Badly designed task discovery interface hurts every participant in the market!

Better modeling as a queuing system may demonstrate other such improvements.
Survey design

Questionnaire design
  - Right questions
  - Examples

UI design

Implementation

Generic tips
  - Experiment should be self-contained.
  - Keep it short and simple. Brief and concise.
  - Be very clear with the relevance task.
  - Engage with the worker. Avoid boring stuff.
  - Always ask for feedback (open-ended question) in an input box.
Other Design Principles

- Text alignment
- Legibility
- Reading level: complexity of words and sentences
- Attractiveness (worker’s attention & enjoyment)
- Multi-cultural / multi-lingual
- Who is the audience (e.g. target worker community)
  - Special needs communities (e.g. simple color blindness)
System Design

Algorithms

Watson/IBM

Google search

ORACLE

Hadoop

Machines

People

amplab

UC BERKELEY

yelp

amazon mechanical turk

Artificial Intelligence

MATLAB SIMULINK
DB & Crowd

- How can crowd help databases?
  - Fix broken data
  - Add missing data
  - Subjective comparison

- How can DB help crowd apps?
  - Lazy data acquisition
  - Game the workers market
  - Semi-automatically create user interfaces
  - Manage the data sourced from the crowd
CrowdDB

- Use crowd to answer DB queries
  - Find missing data
  - Make subjective comparison
- Recognize patterns
- Main operations
  - Join
  - Sort

CrowdSQL

![Diagram of CrowdDB system with components such as Parser, Optimizer, Executor, and HIT Manager.]
Research Challenges

- **Data Model**
  - Uncertainty
  - User view vs System view
  - How to get data?

- **Query processing**
  - User-defined functions (UDF)
  - CrowdSQL
Example DB Systems

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

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

<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>

Crowdsourcing @ HotDB2012
Tutorial Outline

- Introduction
- Applications
- Platforms
- Challenges
- Opportunities
Opportunities

- Problems with the current platform
  - Very rudimentary
  - No tools for data analysis
  - No integration with databases
  - Very limited search and browse features

- Opportunities
  - What is the database model for crowdsourcing?
  - MapReduce with crowdsourcing
  - Can you integrate human-computation into a language?
  - Task management
Opportunities

- Quality control
  - Human factors vs. outcomes
  - Pricing tasks
  - Predicting worker quality from observable properties (e.g. task completion time)
  - HIT / Requestor ranking or recommendation
  - Expert search: who are the right workers given task nature and constraints
- Privacy
  - Workers and requesters
  - Tasks
Opportunities

- Crowdsourcing is **cheap but not free**, and cannot scale to web without help
  - How to scale out?
  - Improve the accuracy and efficiency of human computation algorithms.
  - Indexing & Pruning techniques

- Dealing with uncertainty
  - Temporal and labeling uncertainty
  - Learning algorithms
  - Search evaluation

- Combining CPU + HPU
  - MapReduce with human computation?
  - Integration points with enterprise systems
Conclusion

- Crowdsourcing for relevance evaluation works
- Fast turnaround, easy to experiment, cheap
- Still have to design the experiments carefully!

Quality
- Worker quality
- User feedback extremely useful
- Usability considerations

Platform
- MTurk is a popular platform and others are emerging
- Lots of opportunities to improve current platforms
- Scale out
References - Tutorials

- Omar Alonso, Matthew Lease: Crowdsourcing for information retrieval: principles, methods, and applications. SIGIR 2011:1299-1300
- Omar Alonso, Matthew Lease: Crowdsourcing 101: putting the WSDM of crowds to work for you. WSDM 2011:1-2
- Omar Alonso: Crowdsourcing for Information Retrieval Experimentation and Evaluation. CLEF 2011:2

- This tutorial (for research purpose only) used some slides in the above tutorials.
- Thank the authors.
References - Survey Papers

- Alexander J. Quinn, Benjamin B. Bederson: Human computation: a survey and taxonomy of a growing field. CHI 2011:1403-1412
References - DB

- Salil S. Kanhere: Participatory Sensing: Crowdsourcing Data from Mobile Smartphones in Urban Spaces. Mobile Data Management 2011:3-6
- Michael J. Franklin, Donald Kossmann, Tim Kraska, Sukriti Ramesh, Reynold Xin: CrowdDB: answering queries with crowdsourcing. SIGMOD 2011:61-72
References - HCI

- Ido Guy, Adam Perer, Tal Daniel, Ohad Greenshpan, Itai Turbahn: Guess who?: enriching the social graph through a crowdsourcing game. CHI 2011:1373-1382
- Aniket Kittur, Boris Smus, Robert Kraut: CrowdForge: crowdsourcing complex work. CHI Extended Abstracts 2011:1801-1806
- Jeffrey Heer, Michael Bostock: Crowdsourcing graphical perception: using mechanical turk to assess visualization design. CHI 2010:203-212
References - NLP

- Omar Zaidan, Chris Callison-Burch: Crowdsourcing Translation: Professional Quality from Non-Professionals. ACL 2011:1220-1229
- Keith Vertanen, Per Ola Kristensson: The Imagination of Crowds: Conversational AAC Language Modeling using Crowdsourcing and Large Data Sources. EMNLP 2011:700-711
References - AI

- Yan Yan, Rómer Rosales, Glenn Fung, Jennifer G. Dy: Active Learning from Crowds. ICML 2011:1161-1168
- Peng Dai, Mausam, Daniel S. Weld: Decision-Theoretic Control of Crowd-Sourced Workflows. AAAI 2010
- Yen-ling Kuo, Jane Yung-jen Hsu: Resource-Bounded Crowd-Sourcing of Commonsense Knowledge. IJCAI 2011:2470-2475
- Edith Law, Haoqi Zhang: Towards Large-Scale Collaborative Planning: Answering High-Level Search Queries Using Human Computation. AAAI 2011
References - IR

- Roi Blanco, Harry Halpin, Daniel M. Herzig, Peter Mika, Jeffrey Pound, Henry S. Thompson, Duc Thanh Tran: Repeatable and reliable search system evaluation using crowdsourcing. SIGIR 2011:923-932
- Krishna Yeswanth Kamath, James Caverlee: Transient crowd discovery on the real-time social web. WSDM 2011:585-594
References - Theory

- Shuchi Chawla, Jason D. Hartline, Balasubramanian Sivan: Optimal crowdsourcing contests. SODA 2012:856-868
Thank You!

Q & A?