

# Crowdsourcing

**MICHAEL BERNSTEIN**

**SPRING 2013**

[cs376.stanford.edu](http://cs376.stanford.edu)



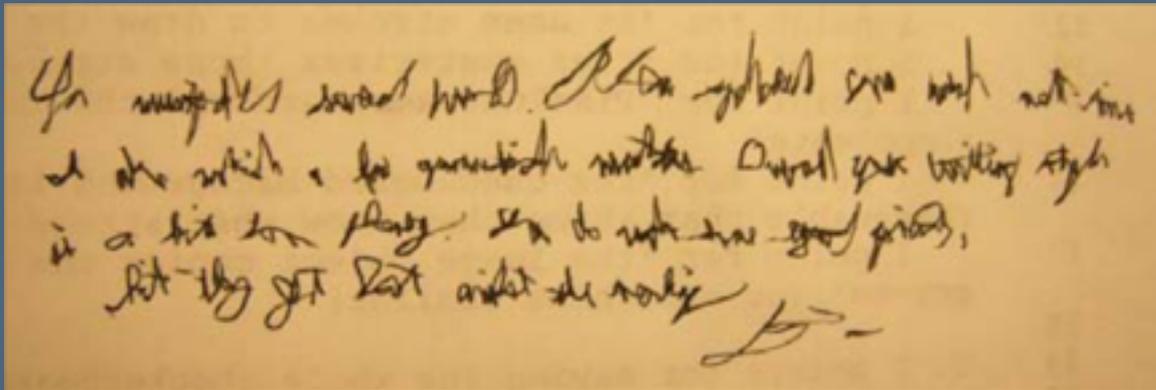
# Early crowdsourcing research

- Two distributed workers work independently, and a third verifier adjudicates their responses

When we think about early influential crowdsourcing research, we often think of things like Greg Little's work on iterative algorithms and transcription. Greg got a group of crowd members to collectively transcribe this text --- accurately!

# Early crowdsourcing research

- Two distributed workers work independently, and a third verifier adjudicates their responses

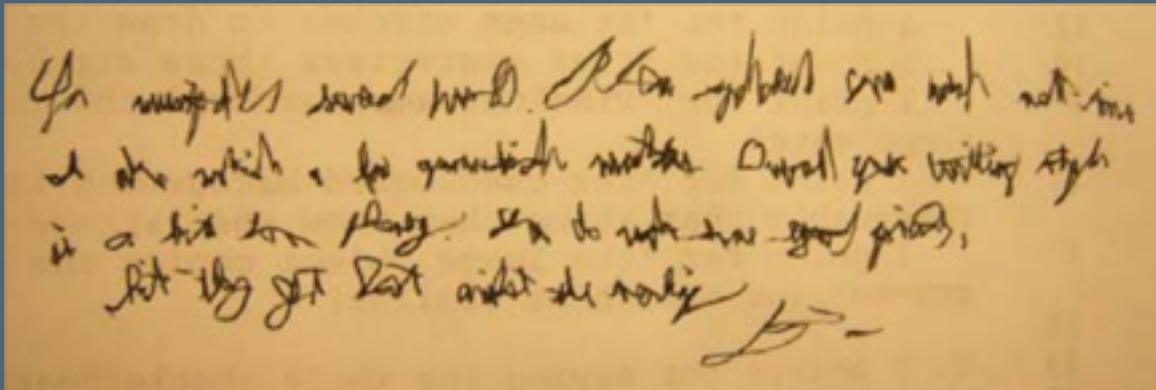


[Little et al.,  
HCOMP 2009]

When we think about early influential crowdsourcing research, we often think of things like Greg Little's work on iterative algorithms and transcription. Greg got a group of crowd members to collectively transcribe this text --- accurately!

# Early crowdsourcing research

- Two distributed workers work independently, and a third verifier adjudicates their responses



[Little et al.,  
HCOMP 2009]

You (misspelled) (several) (words). Please spellcheck your work next time. I also notice a few grammatical mistakes. Overall your writing style is a bit too phoney. You do make some good (points), but they got lost amidst the (writing). (signature)

When we think about early influential crowdsourcing research, we often think of things like Greg Little's work on iterative algorithms and transcription. Greg got a group of crowd members to collectively transcribe this text --- accurately!

# Early crowdsourcing research

[Grier 2007]

- Two distributed workers work independently, and a third verifier adjudicates their responses



## 1760

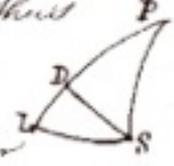
British Nautical Almanac  
Neil Maskelyne

But in reality, early crowdsourcing research actually dates back more to the 18th century, when the British Royal Astronomer Neil Maskelyne was tasked with creating nautical sea charts.

Greenwich Sept 30 1788

Sir,  
In answer to your letter of the 23<sup>d</sup> instant  
Compute D<sup>d</sup> dist. from a star by logarithms, thus

Let the pole of the ecliptic, *N* or *S*, *P*, be at  
the top of a circle, *AD* a part of the ecliptic  
at the *D*, the star, *P* a perpendicular  
great circle. Let fall from *P* to *AD*, perpendicular  
to it in *T*. There suppose *P* to be less than 90°  
By log.  $c, P+t, P^2=t, PD$  &  $PL-P^2=LD$   
 $c, LD=-c, PD+c, PS+c, LD$   
 $c, P+s, PS-s, PD+c, LD$

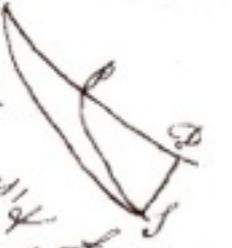


The latter formula must be used when *PD*  
is large, or near 90°; but may be used safely  
in all cases.

Example

$P 20.30$	$c$	$9.9715876$	
$P 25.30$	$t$	$13.2575588$	
$P 25.30$	$c$	$9.9557589$	$9.9557589$
$P 25.30$	$t$	$12.0977260$	$9.9999879$
$P 25.30$	$t$	$12.0534829$	$0.0000170$
$P 25.30$	$c$	$9.9986631$	$9.9986631$
$P 25.30$	$c$	$9.9544269$	

If *P* should be greater than 90°  
then  $PL+P^2=LD$  as in this case



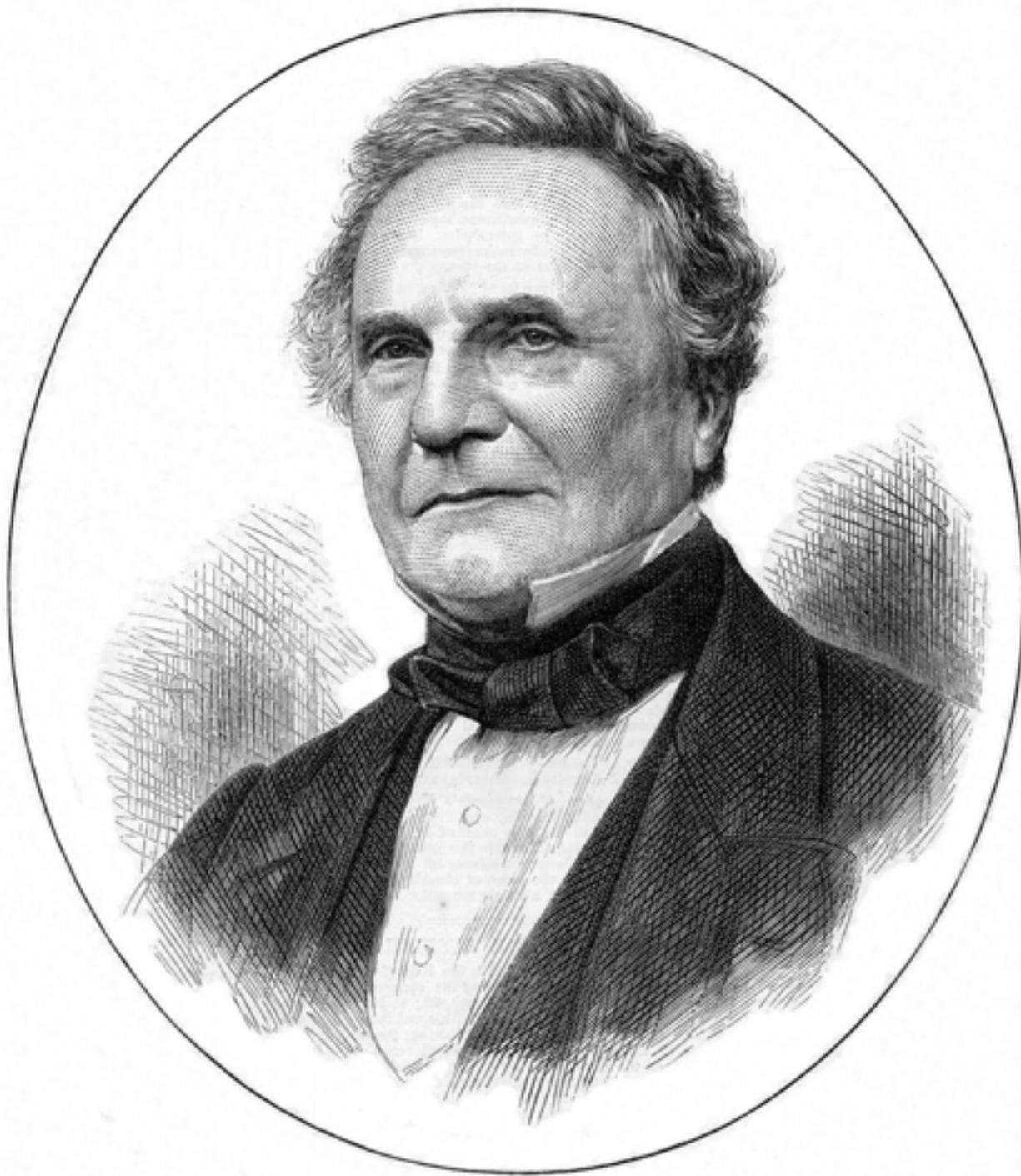
As the planets *placis*, excepting the  
*sun* and *moon* are only set down to minutes,  
there is no occasion to allow for nutation &  
aberration. — In computing the occultations of  
stars by the *D* you must calculate their longitudinal *dist*  
to *placis* at *D* & allow nutation & aberration *sample* to  
be of the *form* of the *form* of *sample* *sample*  
to the *33* of *sample*

JULY 1799

D <sup>d</sup>	Longit. at Noon.			Latit. at Noon.			Star of the D	Star of the D
	S.	D.	M. S.	D.	M.	S.		
1	2	20	0	51	2	58	29	1
2	3	2	45	8	3	48	52	2
3	3	15	46	23	4	28	29	3
4	3	29	3	54	4	54	28	4
5	4	12	35	47	5	4	34	5
6	4	26	19	33	4	57	26	6
7	5	10	12	31	4	32	51	7
8	5	24	12	11	3	51	51	8
9	6	8	16	36	2	56	35	9
10	6	22	24	16	1	50	19	10
11	7	6	34	4	0	37	68	11
12	7	20	44	43	0	38	25	12
13	8	4	54	27	1	51	21	13
14	8	19	0	45	2	57	1	14
15	9	3	0	17	3	51	19	15
16	9	16	49	18	4	31	12	16
17	10	0	24	8	4	54	54	17
18	10	13	41	50	5	1	59	18
19	10	26	40	41	4	53	8	19
20	11	9	20	28	4	29	54	20
21	11	21	42	30	3	54	21	21
22	0	3	49	35	3	8	48	22
23	0	15	45	40	2	15	32	23
24	0	27	35	32	1	16	47	24
25	1	9	24	26	0	14	46	25
26	1	21	17	52	0	48	14	26
27	2	3	21	8	1	49	51	27
28	2	15	32	59	2	47	31	28
29	2	27	15	49	3	48	23	29
30	3	9	24	31	4	47	34	30

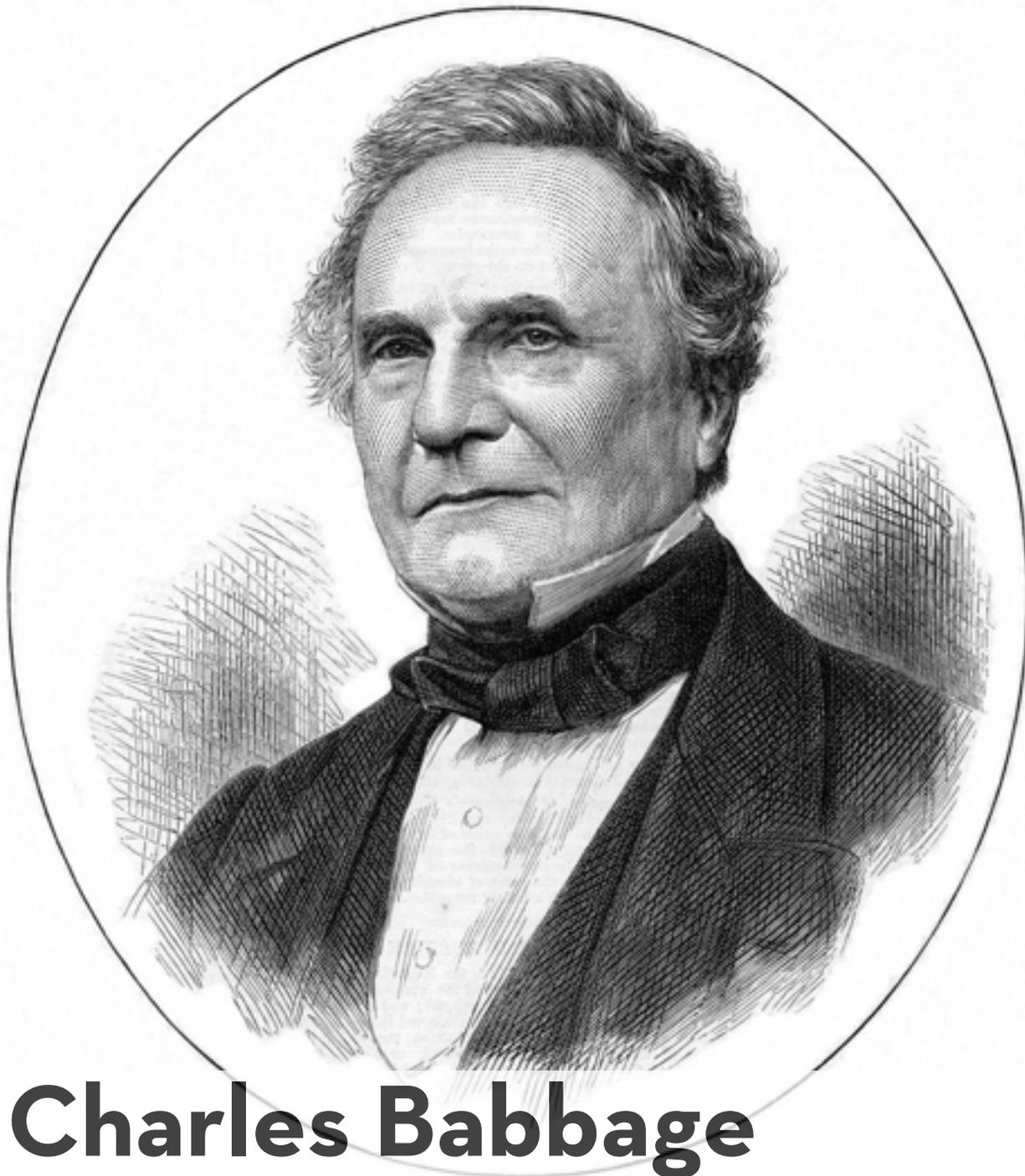
# Work distributed via mail

He did so by mailing out spreadsheets like this. People would fill in columns, then mail them to another person to cross-check multiple people who filled out the same data, and arbitrate any differences.



In fact, this early pioneer of computing was also a pioneer in crowdsourcing.

Charles Babbage was one of the early people who helped lead the British Royal Almanac process. He also developed several process improvements that we'd be smart to embed in crowdsourcing today. For example, his rule of errors: that two people who do the same task in the same way will make the same errors.

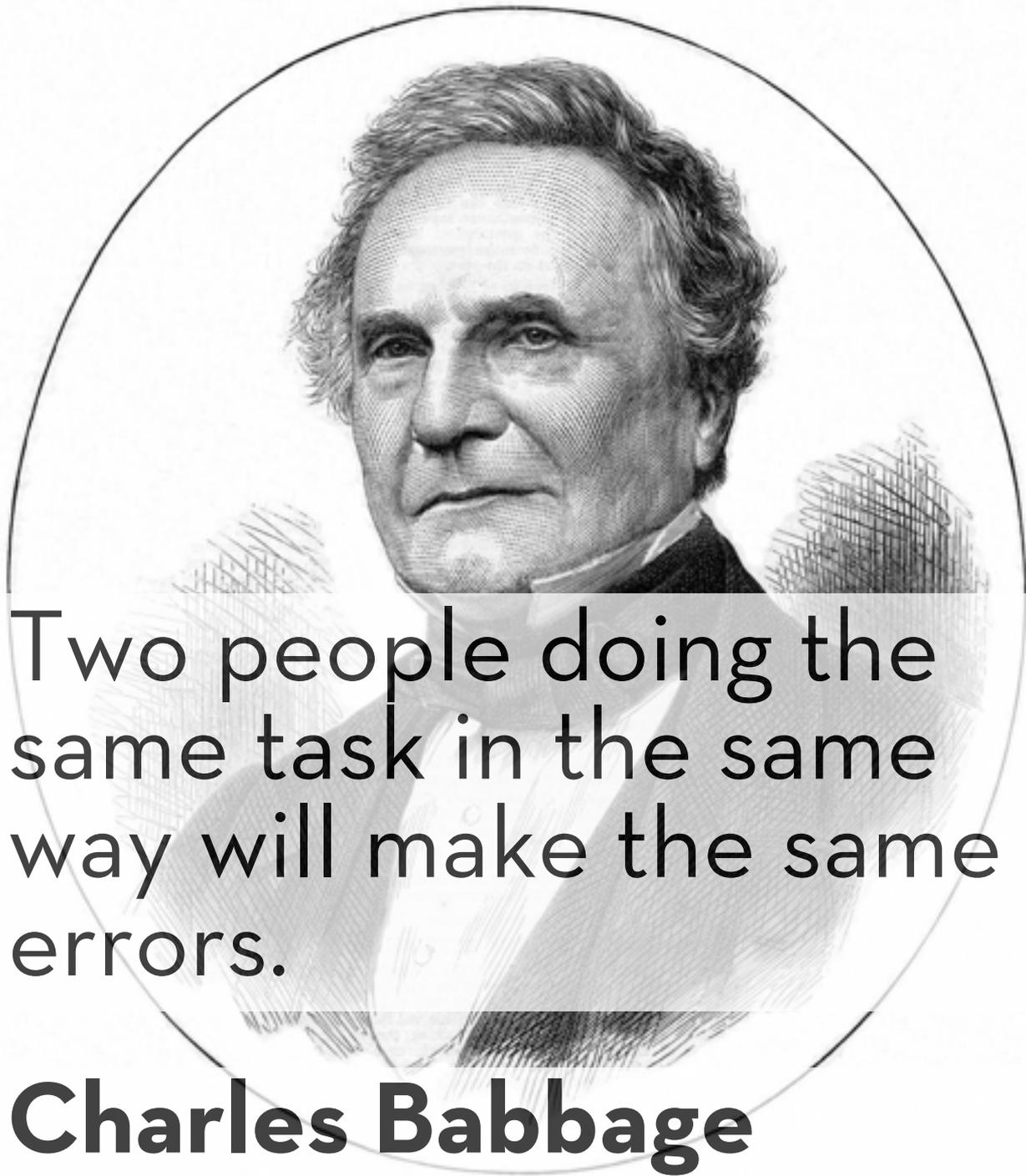


## Charles Babbage

5

In fact, this early pioneer of computing was also a pioneer in crowdsourcing.

Charles Babbage was one of the early people who helped lead the British Royal Almanac process. He also developed several process improvements that we'd be smart to embed in crowdsourcing today. For example, his rule of errors: that two people who do the same task in the same way will make the same errors.

A black and white engraving of Charles Babbage, an elderly man with wavy hair, wearing a dark coat and a white cravat. The portrait is enclosed in a thin circular border.

Two people doing the same task in the same way will make the same errors.

## **Charles Babbage**

5

In fact, this early pioneer of computing was also a pioneer in crowdsourcing.

Charles Babbage was one of the early people who helped lead the British Royal Almanac process. He also developed several process improvements that we'd be smart to embed in crowdsourcing today. For example, his rule of errors: that two people who do the same task in the same way will make the same errors.

# Mathematical Tables Project

- WPA project, begun 1938
- Calculated tables of mathematical functions
- Employed 450 human computers



# Mathematical Tables Project

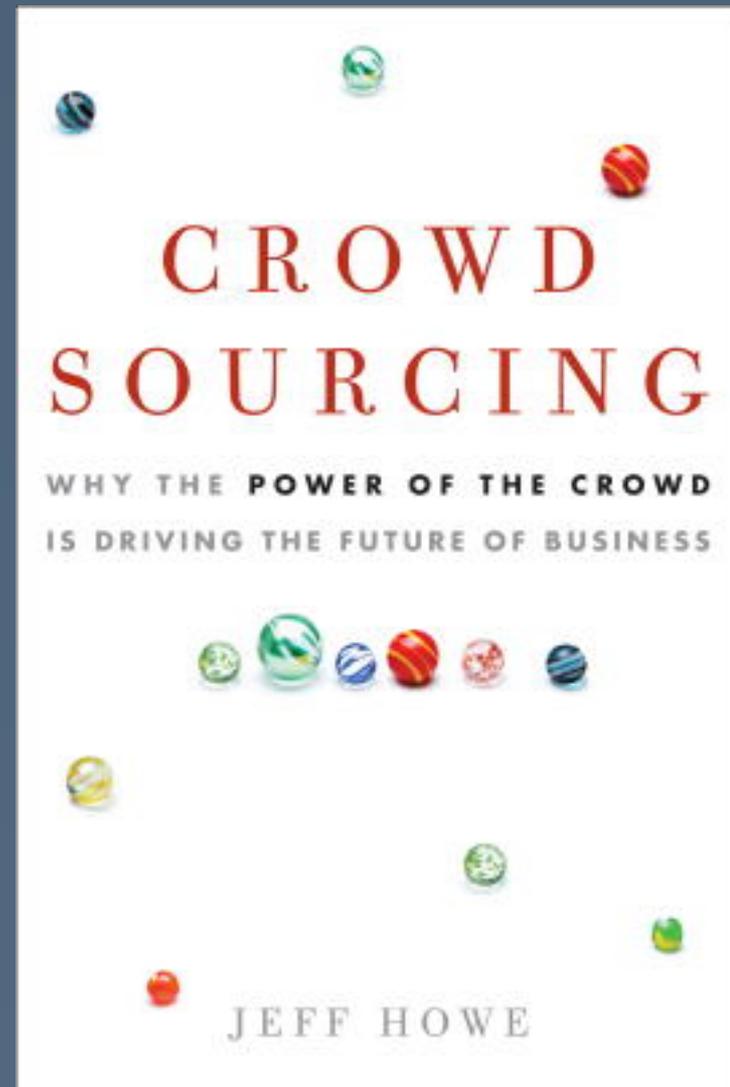
- WPA project, begun 1938
- Calculated tables of mathematical functions
- Employed 450 human computers



The origin of the  
term **computer**

# Origins of the term

- Jeff Howe, 2006 in Wired
- “Taking [...] a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call.”



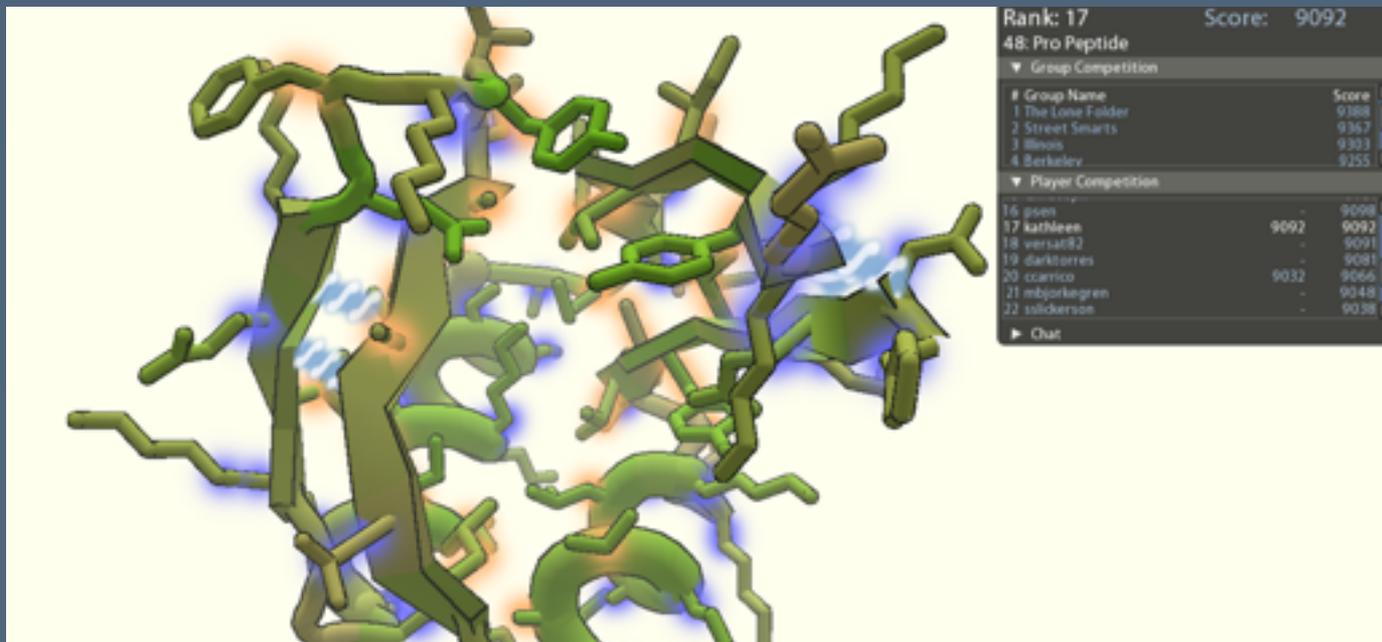
# Success: games with a purpose

Label every image on the internet using a game  
[von Ahn and Dabbish, CHI '06]



# Success: scientific collaboration

- FoldIt: protein-folding game
- Amateur scientists have found protein configurations that eluded scientists for years



# More successes



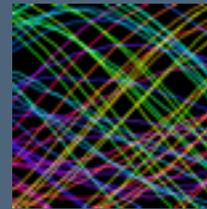
Largest  
encyclopedia  
in history



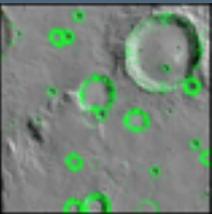
Disaster  
reporting



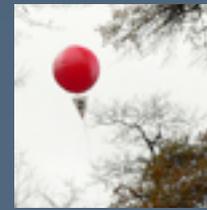
Kasparov vs.  
the world



Collaborative  
math proofs



NASA  
Clickworkers



DARPA Red  
Balloon Challenge

# Paid Crowdsourcing

- Pay small amounts of money for short tasks
- Amazon Mechanical Turk: Roughly five million tasks completed per year at 1-5¢ each  
[Ipeirotis 2010]

**Label an image**

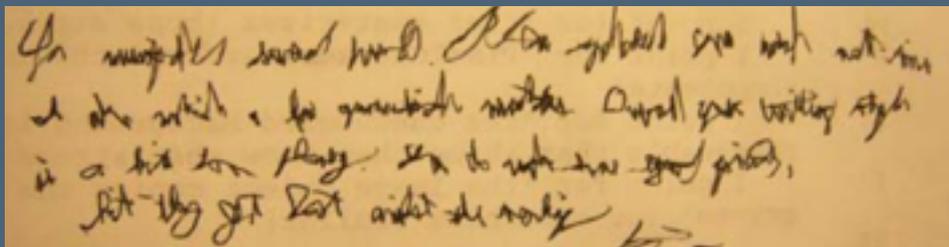
Reward: \$0.02

**Transcribe audio clip**

Reward: \$0.05

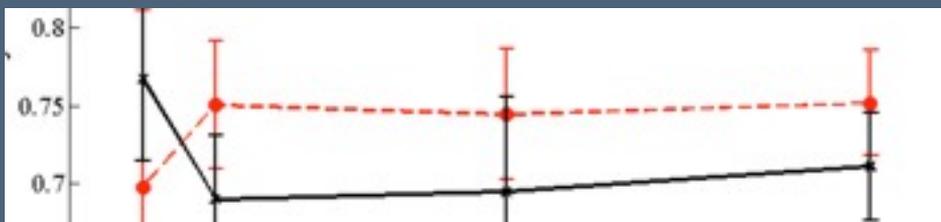
- Population: 40% U.S., 40% India, 20% elsewhere
- Gender, education and income are close mirrors of overall population distributions  
[Ross 2010]

# Major topics of research



## Crowd algorithms

[Little et al., HCOMP 2009]



## Incentives and Quality

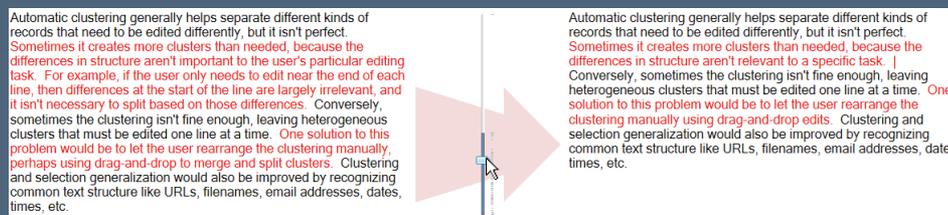
[Mason and Watts, HCOMP 2009]

[Dow et al., CSCW 2012]

## Crowd-powered systems

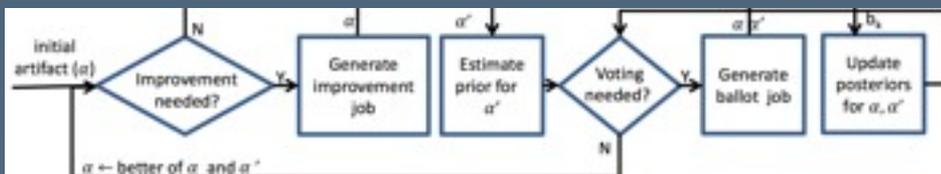
[Bernstein et al., UIST 2010]

[Bigham et al., UIST 2010]



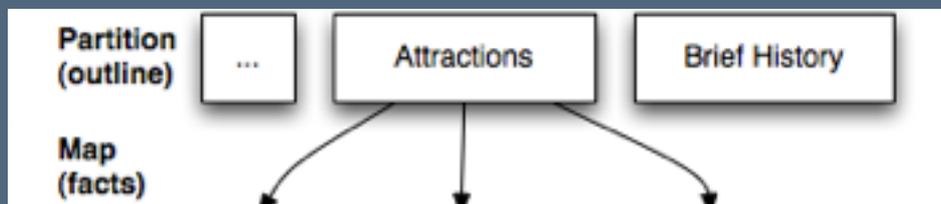
## AI for HCOMP

[Dai, Mausam & Weld, AAAI 2010]



## Complex Work

[Kittur et al., UIST 2011]



12

Not to mention work on data management systems like done at Stanford, at Berkeley and MIT.

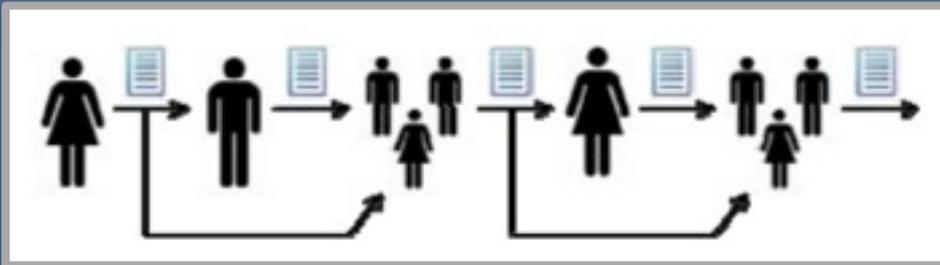
# Crowdsourcing algorithms

# Goal: guide crowds as they work

- Designing crowdsourcing algorithms is often like designing a user interface that will keep a user “in bounds” on your application
- Challenges
  - Taking unexpected action
  - Trying too hard
  - Trying not hard enough

# Crowdsourcing algorithm

- A generalized version of a workflow
- Iterative algorithms [Little et al. 2009]
  - Hand off from one worker to the next



- Most crowdsourcing processes are more parallel, but less interesting algorithmically

# Crowdsourcing algorithms

- Open-ended editing: Find-Fix-Verify [Bernstein et al., UIST '10]
- Graph search [Parameswaran et al., VLDB '11]
- Clustering [Chilton et al., CHI '13]
- and many more...
  
- When write an algorithm?  
*If you tried this in a straightforward way, would crowds fail? Why?*

# Incentives and quality

# Incentives

- Does paying more produce better work?

# Incentives

- Does paying more produce better work?
  - More work, but not higher-quality work  
[Mason and Watts, HCOMP '09]

# Incentives

- Does paying more produce better work?
  - More work, but not higher-quality work  
[Mason and Watts, HCOMP '09]
- Does feedback produce better work?

# Incentives

- Does paying more produce better work?
  - More work, but not higher-quality work  
[Mason and Watts, HCOMP '09]
- Does feedback produce better work?
  - Self-assessment and expert assessment both improve the quality of work  
[Dow, Kulkarni, Klemmer and Hartmann, CSCW '11]

# Incentives

[Shaw, Horton and Chen, CSCW '11]

- Which of these approaches improve quality?
  - Comparison to other workers
  - Normative claims: “it’s important that you try hard”
  - Solidarity: your team gets a bonus if you are right
  - Humanization: “thanks for working; I’m Aaron.”
  - Reward or punish accuracy with money
  - Reward or punish agreement with money
  - Bayesian truth serum: predict others’ responses
  - Bet payment on the accuracy of your responses

# Incentives

[Shaw, Horton and Chen, CSCW '11]

- Which of these approaches improve quality?
  - Comparison to other workers
  - Normative claims: “it’s important that you try hard”
  - Solidarity: your team gets a bonus if you are right
  - Humanization: “thanks for working; I’m Aaron.”
  - Reward or punish accuracy with money
  - Reward or **punish agreement with money**
  - **Bayesian truth serum: predict others’ responses**
  - Bet payment on the accuracy of your responses

# Motivations

[Antin and Shaw, CHI '12]

- Ask workers: “I am motivated to do HITs on Mechanical Turk...”
  - To kill time
  - To make extra money
  - For fun
  - Because it gives me a sense of purpose

# Motivations

[Antin and Shaw, CHI '12]

- Ask workers: “I am motivated to do HITs on Mechanical Turk...”
  - To kill time
  - To make extra money
  - For fun
  - Because it gives me a sense of purpose
- List experiment: vary which reasons appear in the list, and ask how many reasons the participant agrees with
  - This technique counters social desirability bias

# Motivations

[Antin and Shaw, CHI '12]

- US workers
  - 40% overreporting of money as a reason to work
- India-based workers
  - 142% underreporting of killing time and 60% underreporting fun as reasons
  - Money was not over- or under-reported

# Hacking motivation: communitysourcing [Heimerl et al., CHI '12]

## Communitysourcing

Engaging Local Crowds to Perform  
Expert Work Via Physical Kiosks

Kurtis Heimerl, Brian Gawalt, Kuang Chen  
Tapan Parikh, Björn Hartmann  
University of California, Berkeley

CHI 2012

# Judging quality explicitly

- Gold standard judgments [Le et al., SIGIR CSE '10]
  - Include questions with known answers
  - Performance on these “gold standard” questions is used to filter work

# Judging quality explicitly

- Gold standard judgments [Le et al., SIGIR CSE '10]
  - Include questions with known answers
  - Performance on these “gold standard” questions is used to filter work
- Get Another Label [Sheng, Provost, Ipeirotis, KDD '08]
  - Estimate the correct answer and worker quality jointly
  - Try it! <https://github.com/ipeirotis/Get-Another-Label>

# Judging quality implicitly

[Rzeszotarski and Kittur, UIST '12]

- Observe low-level behaviors
  - Clicks
  - Backspaces
  - Scrolling
  - Timing delays
- SVMs on these behaviors predict work quality
- Limitation: models must be built for each task

# Crowd-powered systems

# Main idea

- Embed crowd intelligence inside of user interfaces and applications we use today



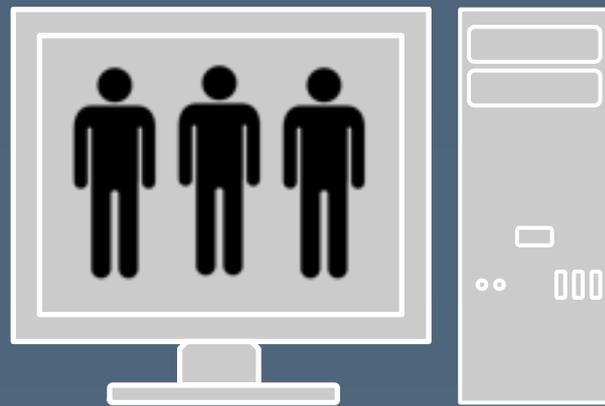
Interface



Wizard of Oz

# Main idea

- Embed crowd intelligence inside of user interfaces and applications we use today



Wizard of  
Turk

# Soylent

[Bernstein et al, UIST '10]

## Shortn

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.



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, as structure differences aren't important to the editing task. Conversely, if the 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 using drag-and-drop edits. Clustering and selection generalization would also be improved by recognizing common text structure like URLs, filenames, email addresses, dates, times, etc.

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 structural differences aren't important to the user's editing task. 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 using drag-and-drop edits. Clustering and selection generalization would also be improved by recognizing common text structure like URLs, filenames, email addresses, dates, times, etc.

# VizWiz

[Bigham et al., UIST '10]

- Visual question answering for the blind

What color is this pillow?	What denomination is this bill?	Do you see picnic tables across the parking lot?	What temperature is my oven set to?	Can you please tell me what this can is?	What kind of drink does this can hold?
					
(89s) I can't tell. (105s) multiple shades of soft green, blue and gold	(24s) 20 (29s) 20	(13s) no (46s) no	(69s) it looks like 425 degrees but the image is difficult to see. (84s) 400 (122s) 450	(183s) chickpeas. (514s) beans (552s) Goya Beans	(91s) Energy (99s) no can in the picture (247s) energy drink

- 1 to 2 minute responses by keeping workers on fake tasks until needed

# Crowd-powered databases

- Database with open-world assumptions:  
`SELECT * FROM ice_cream_flavors`
- Several university flavors
  - Berkeley: CrowdDB [Franklin et al., SIGMOD '11]
  - MIT: Qurk [Marcus et al., CIDR '11]
  - Stanford: Deco [Parameswaran et al. '11]
- Tackling many important optimization questions: e.g., joins, ranking, sorting

# Realtime crowdsourcing

[Bernstein et al., UIST '11]



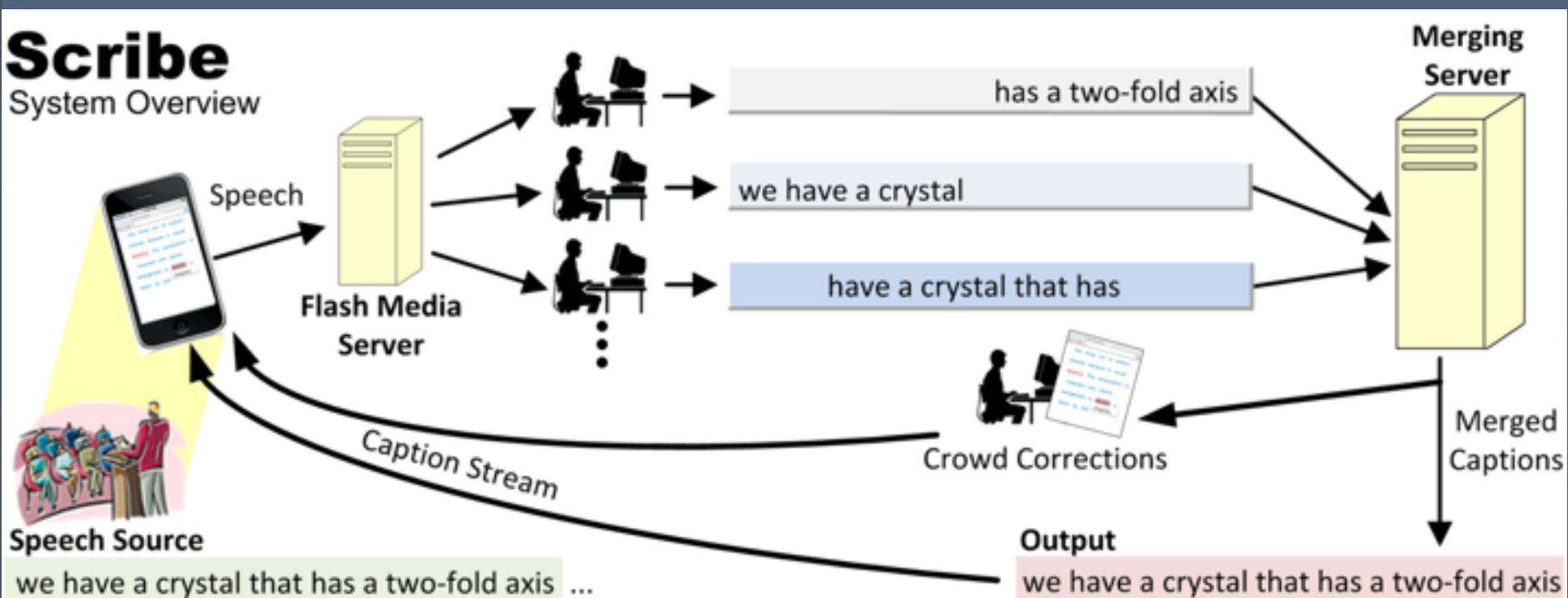
Find photo in this clip



# Realtime crowdsourcing

[Lasecki et al., UIST '11 and UIST '12]

- Realtime captioning using shotgun gene sequencing techniques



# Artificial intelligence for crowds

# TurKontrol: AIs guiding crowds

[Dai, Mausam and Weld, AAAI '10]

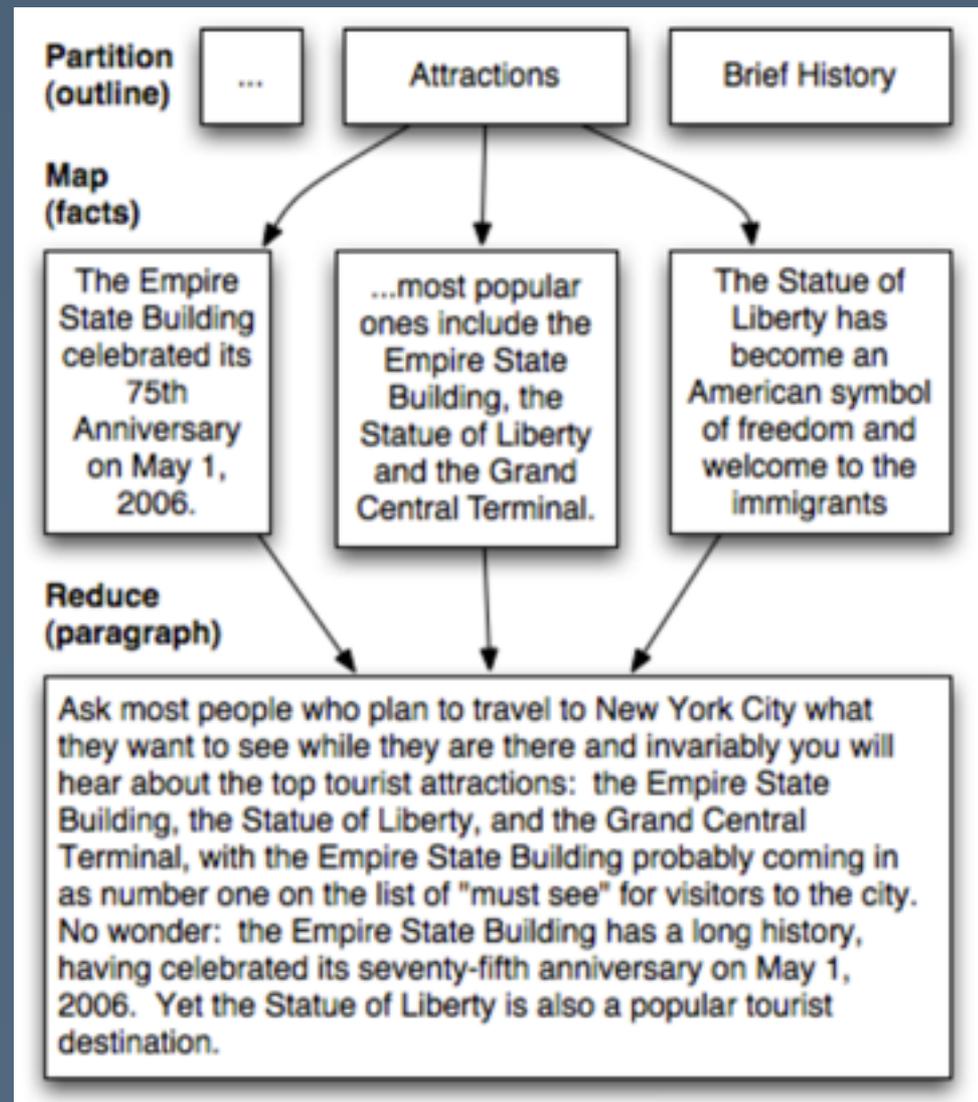
- Workflow planning as a decision-theoretic optimization problem
- Trade off quality vs. number of workers required
  - POMDP to decide: do we need a vote? do we need more voters? do we need more improvement?

# Complex work

# CrowdForge

[Kittur et al., UIST '11]

- Crowdsourcing as a map-reduce process
- To write a wikipedia page, partition on topics, map to find facts and then reduce into a paragraph



# Turkomatic

[Kulkarni, Can, and Hartmann, CSCW '12]

- Let the workers decide on task design
- Is a task too complicated for  $\$D$ ? If so, ask for sub-tasks and recurse. If not, do it yourself.
- Creating a blog with content:

