Lecture 13
Complex Tasks: Workflow Design

Chien-Ju (CJ) Ho

Logistics: Assignments

- Assignment 3: Due yesterday
  - Assume you have enough late days, tomorrow is the final due

- Assignment 4: TBD

Nov 22	Buffer	Assignment 4 (Expected Due: Nov 29)
Nov 24	**Thanksgiving (No Class)**	

Logistics: Project Milestone 2

- Milestone 2: Due Nov 4
  - Summarize your progress
    - Should make enough progress to know **whether the project is feasible**
  - Last chance to convert a research project to an extensive literature survey

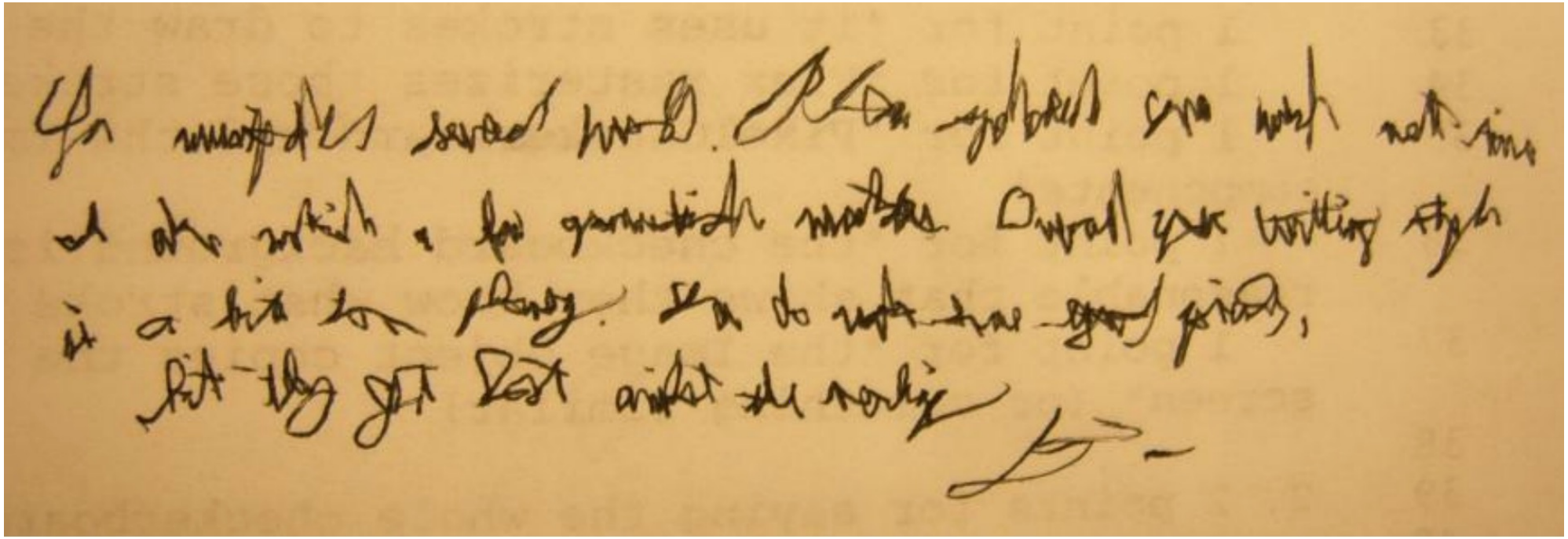
- Midterm Pitch: Nov 1 (Tue), in lecture
  - Opportunity to engage other classmates
  - Every group will get some time from other groups
  - If you want to engage the entire class (for data collection, etc)
    - Let me know at least one week in advance
    - There might be time limits (e.g., 2~3 minutes)
  - Plan your progress to utilize this discussion lecture

Complex Tasks: Workflow Design

Crowdsourcing

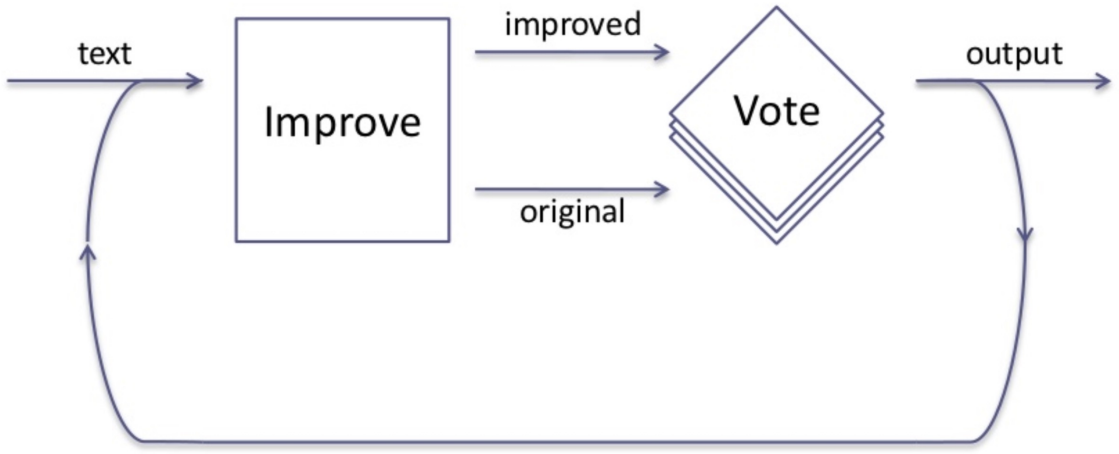
- A lot of focuses are on ***microtasks***
  - Labeling images
  - Transcribing a one-minute audio clips
  - Finding the address of a business
  - …
- A common quality control mechanism
  - Duplicate tasks and perform aggregation
  - Design incentives

What about Complex Tasks?



Designing Workflows

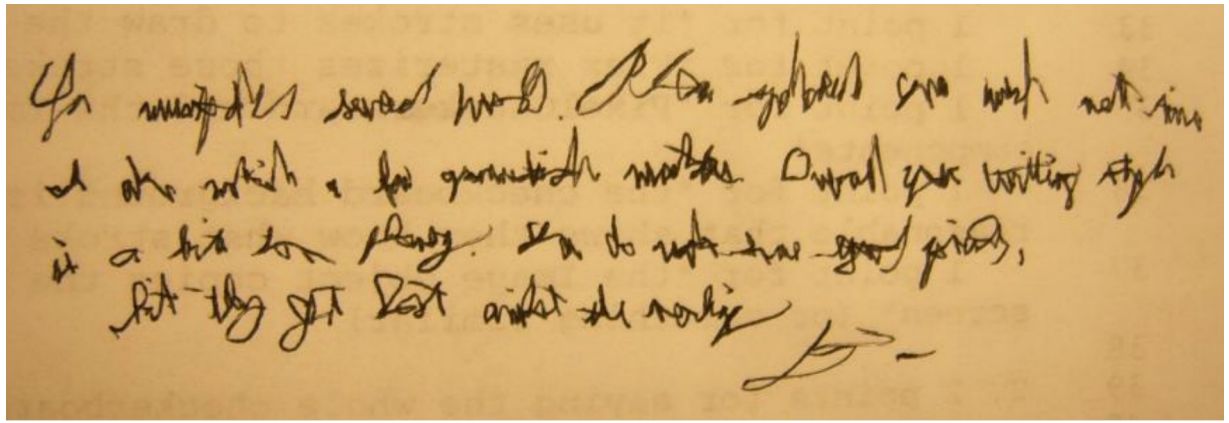
- Dictionary definition of workflow
  - "the sequence of steps involved in moving from the beginning to the end of a working process"
- Decompose a complex task into a set of microtasks



- Some workers are asked to perform **improvement** tasks
- Some workers are asked to **vote** on whether the improvement is good

Little et al. "TurKit: Tools for Iterative Tasks on Mechanical Turk". HCOMP 2009

Designing Workflows



version 1:

You (?) (?) (?) (work). (?) (?) (?) work (not) (time). I (?) (?) a few grammatical mistakes. Overall your writing style is a bit too (phoney). You do (?) have good (points), but they got lost amidst the (writing). (signature)

version 4:

You (misspelled) (several) (words). (?) (?) (?) work next (time). I also notice a few grammatical mistakes. …

version 5:

You (misspelled) (several) (words). (Plan?) (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)

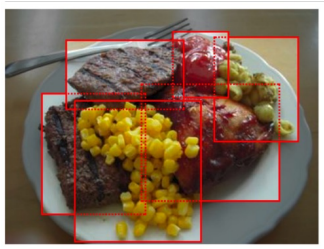
version 6:

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)

Little et al. “TurKit: Tools for Iterative Tasks on Mechanical Turk”. HCOMP 2009

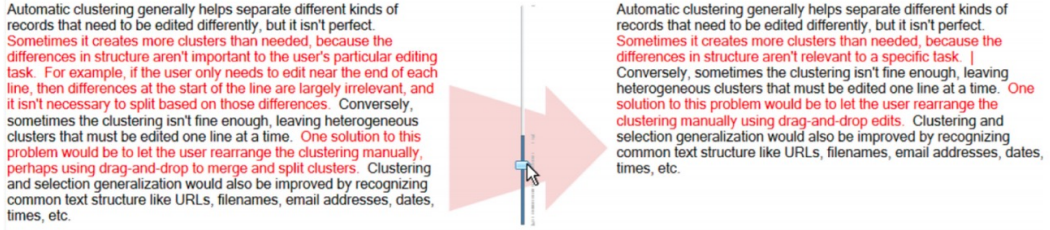
Other Workflows

- Calculate nutrition values of a food photo [Noronha et al. UIST 2011]



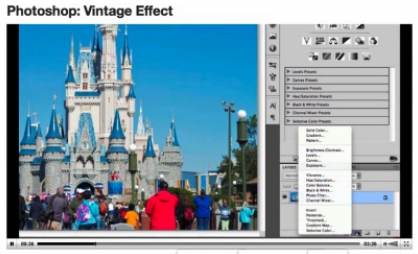
Tag – Identify - Measure

- Proofreading and document editing [Bernstein et al. UIST 2010]



Find – Fix - Verify

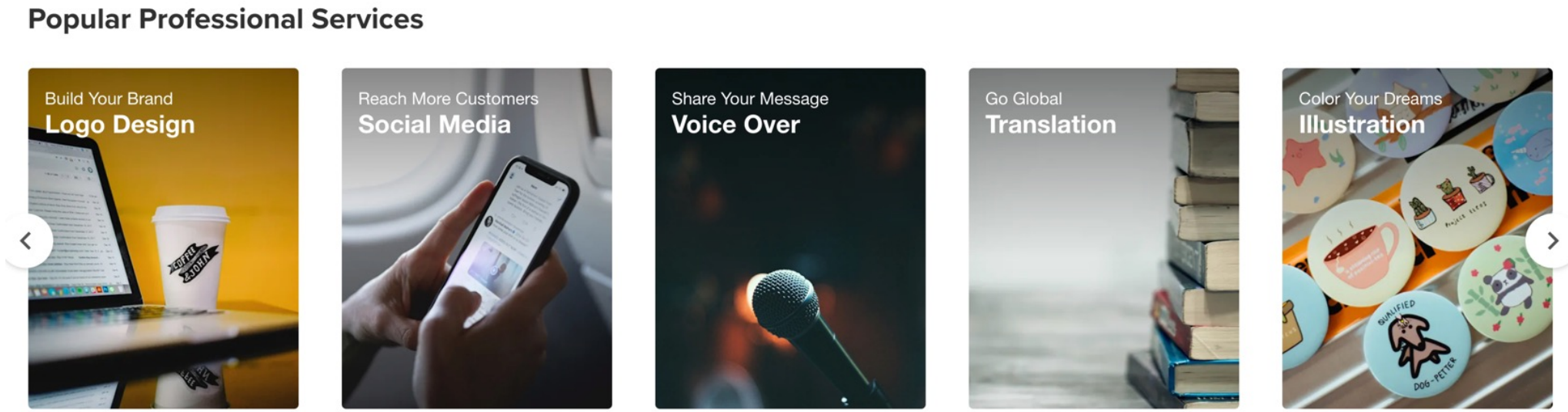
- Extracting steps from how-to videos [Kim et al. CHI 2014]



Find – Verify – Expand

Another Approach [Focus of Next Lecture]

- We can recruit *experts* in doing the work
  - Traditional workplace approach
  - There exist such crowdsourcing marketplaces as well (e.g., Upwork or Fiverr)



Designing Workflows

- Treating humans as computation units

- Question:
  - Can we incorporate humans in the computational framework?
  - Can we implement human computation as function calls?

```
ideas = []
for (var i = 0; i < 5; i++) {
    idea = mturk.prompt(
        "What's fun to see in New York City?
         Ideas so far: " + ideas.join(", "))
    ideas.push(idea)
}

ideas.sort(function (a, b) {
    v = mturk.vote("Which is better?", [a, b])
    return v == a ? -1 : 1
})
```

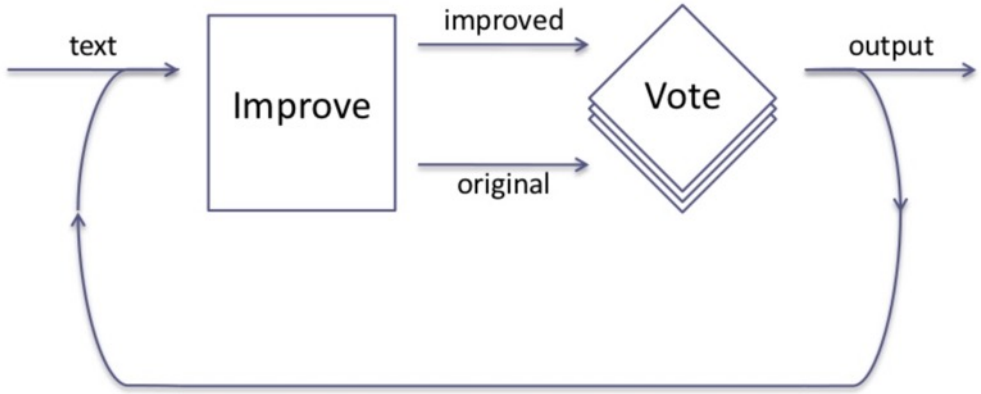
- We are simultaneously collecting data from humans to train AI/ML.
- May gradually replace human computation function calls with automatic process when AI/ML catches up.

TurKit: Human Computation Algorithms

Little et al. UIST 2010

Example: Iterative Writing

- Task: Have workers iteratively refine the description for the image.



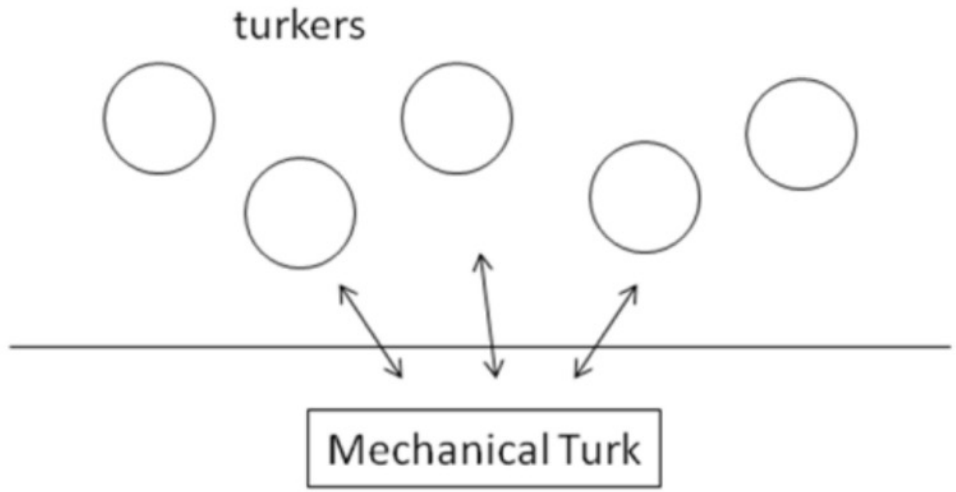

```
// generate a description of X
// and iterate it N times
var text = ""
for (var i = 0; i < N; i++) {
    // generate new text
    var newText = mturk.prompt(
        "Please write/improve this paragraph
         describing " + X + ": " + text)

    // decide whether to keep it
    if (vote("Which describes " + X + " better?",
        [text, newText]) == newText) {
        text = newText
    }
}
```

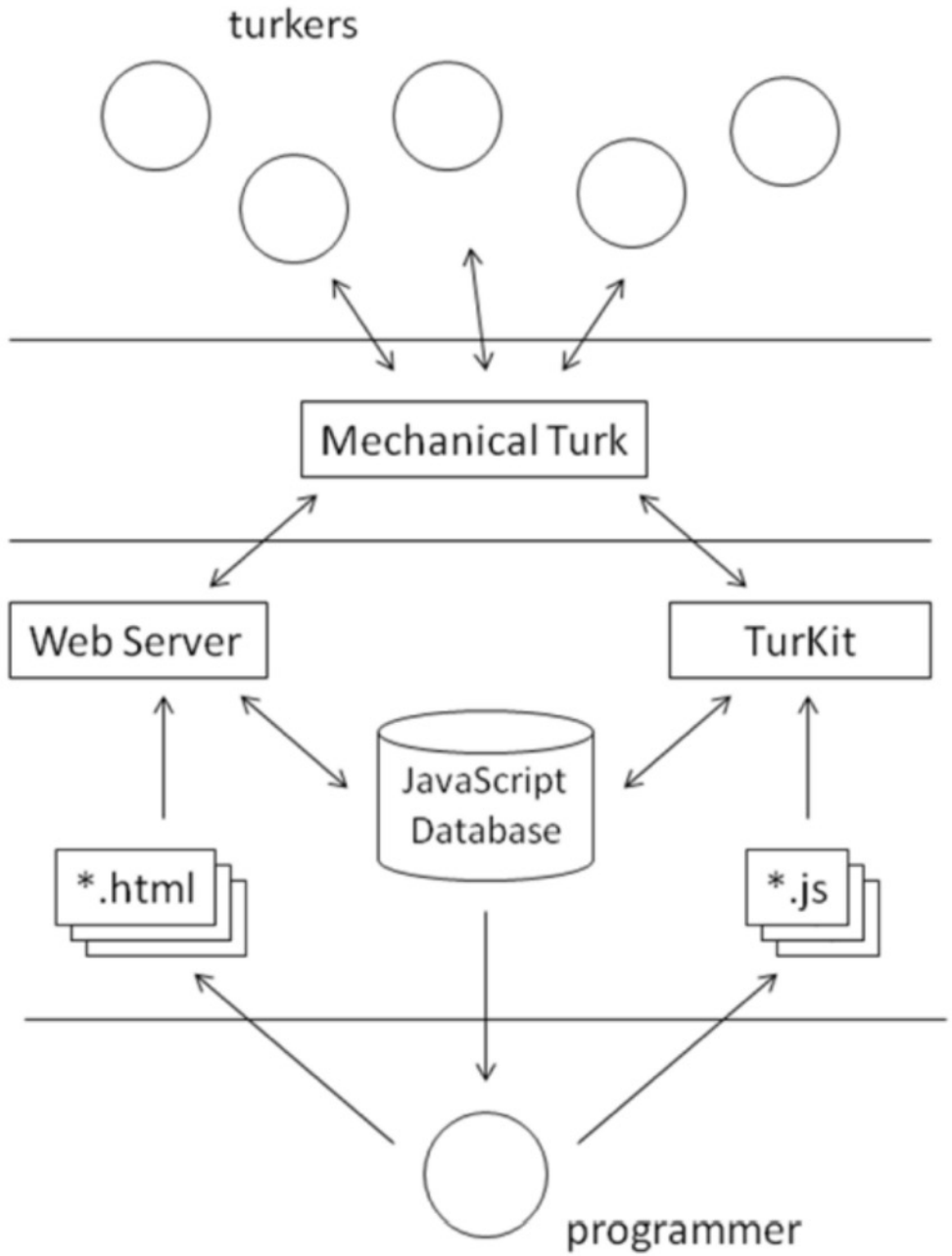
Example: Iterative Writing

- Task: Have workers iteratively refine the description for the image.

Iteration 1: Lightening strike in a blue sky near a tree and a building.
Iteration 2: The image depicts a strike of fork lightening, striking a blue sky over a silhoutted building and trees. (4/5 votes)
Iteration 3: The image depicts a strike of fork lightning, against a blue sky with a few white clouds over a silhouetted building and trees. (5/5 votes)
Iteration 4: ~~The image depicts a strike of fork lightning, against a blue sky- wonderful capture of the nature.~~ (1/5 votes)
Iteration 5: This image shows a large white strike of lightning coming down from a blue sky with the tops of the trees and rooftop peaking from the bottom. (3/5 votes)
Iteration 6: This image shows a large white strike of lightning coming down from a blue sky with the silhouettes of tops of the trees and rooftop peeking from the bottom. The sky is a dark blue and the lightening is a contrasting bright white. The lightening has many arms of electricity coming off of it. (4/5 votes)


turkers
Mechanical Turk


programmer


turkers
Mechanical Turk
Web Server
TurKit
JavaScript
Database
*.html
*.js
programmer

Human Computation Algorithms

- Key Properties:
  - Human computation **costs more** than machine computation
  - Web applications are generally prone to errors

- Goal: Design a robust algorithmic framework for human computation algorithms

Crash-and-Rerun Programming

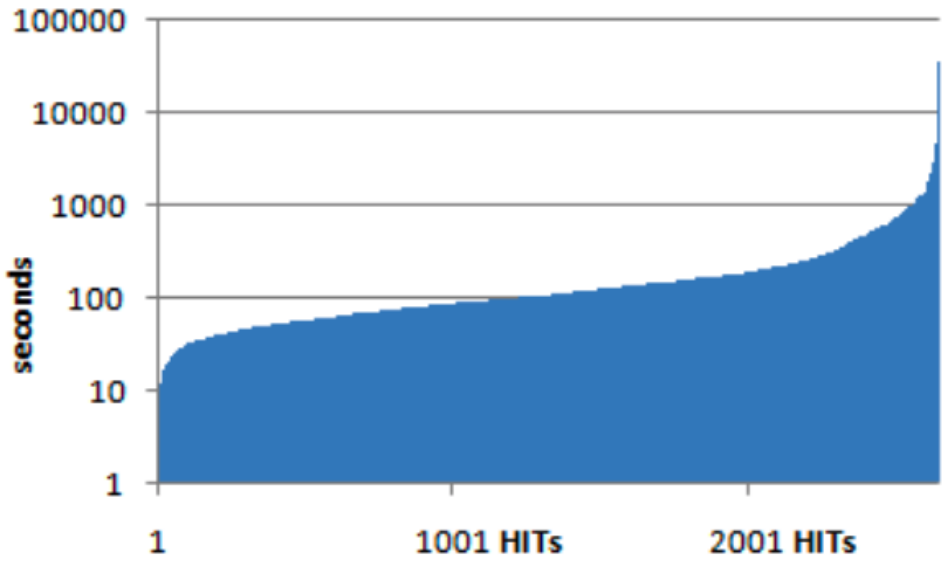
- Intuitions:
  - Store the results of human computation function calls to database (DB).
  - If error happens, **crash** the program and **rerun** from the start
    - For costly operations, obtain results from the DB instead of running it

- Implementation: introduce keyword *once*
  - Remove non-determinism
  - Store costly operations

- Provide functionalities
  - prompt, vote, sort
  - fork, join
  - work with MTurk APIs

Performance Evaluation

- The paper demonstrates the results for a couple of applications.
- They also measure the time for completion for 1-cent HITs.



Can we reduce the latency?

Real-time crowdsourcing

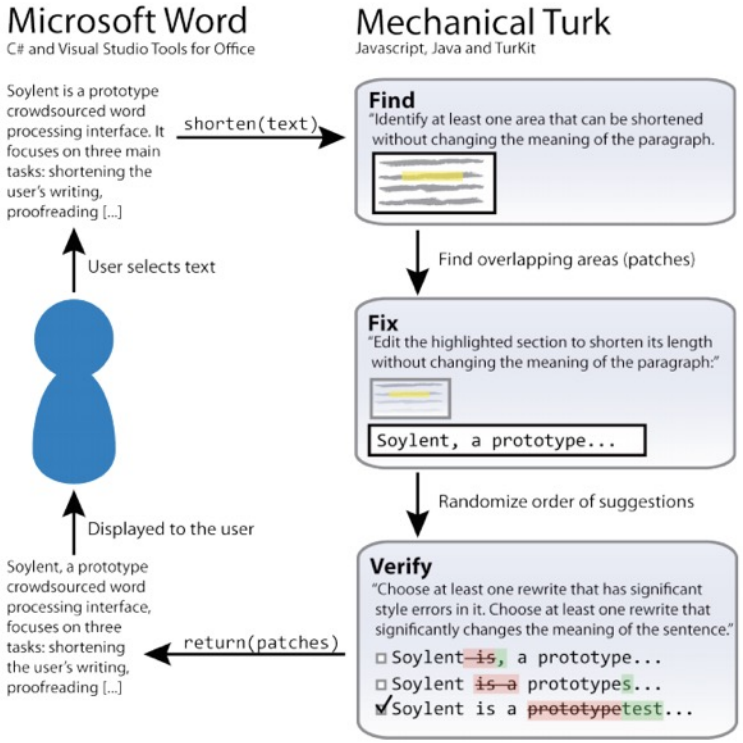
Other Frameworks

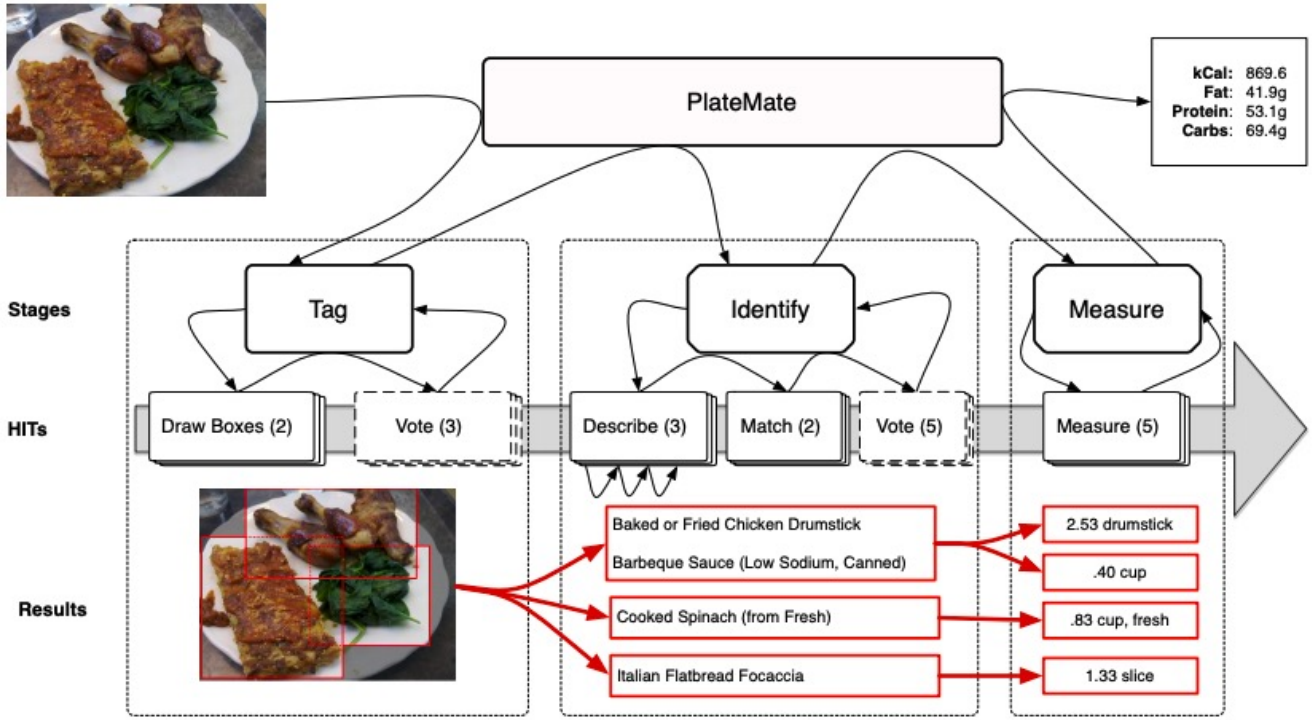
- AutoMan
  - https://emeryberger.com/research/automan/
  - Developed by the team at UMass Amherst
  - Provide support for quality and budget control

- TurkServer
  - https://github.com/TurkServer
  - Support multi-worker collaborations

We can write workflows with TurKit, however…

- Workflow design is domain specific and requires a great amount of effort





The Ultimate Goal

How many calories does the food in the photo contains?

CROWDSOURCE IT!

Can we utilize the crowd to design workflows…

Can we utilize AI to help design workflows…

Discussion

- What's your general thoughts on TurKit? What additional functionalities do you hope they also provide?

- We usually use time and space complexity to characterize how good an algorithm. Imagine you are now trying to develop a similar theory for human computation algorithms, what do you think are the key factors that should be considered?

- How can we utilize the crowd to design workflows?
  - Assume we require each worker can only do microtasks worth less than 20 cents

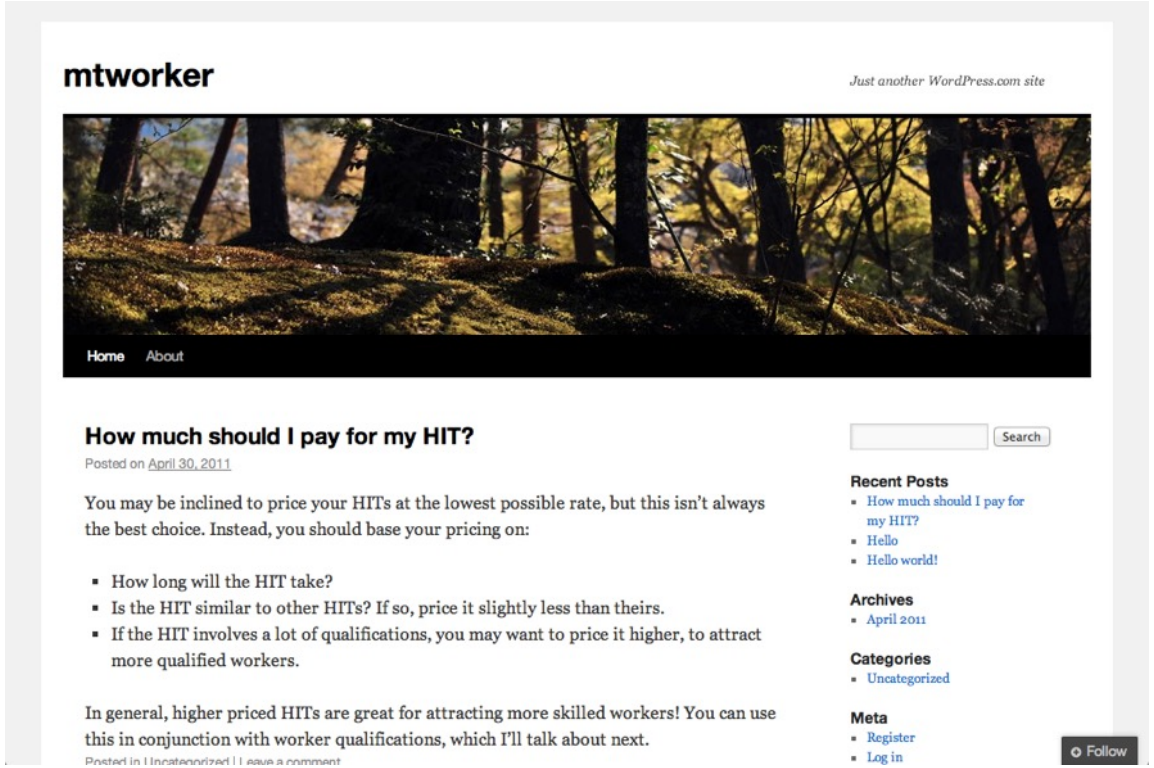
Turkomatic: A workflow for designing workflows

Kulkarni, Can, and Hartmann. CSCW 2012.

Example Task

Create a new blog about Mechanical Turk with two posts.

CROWDSOURCE IT!



Price-Divide-Solve

- A divide-and-conquer workflow

- Price: Can this task be solved for 20 cents

- If yes: Solve the task and return the answer

- If no: Divide the task into multiple steps
  For each step, start from the price step
  Merge steps into solutions

Price-Divide-Solve

- A divide-and-conquer workflow

- **Price**: Can this task be solved for 20 cents

- If yes: **Solve** the task and return the answer

- If no: **Divide** the task into multiple steps

  For each step, start from the price step

  Merge steps into solutions

Create a new blog about Mechanical Turk with two posts.

CROWDSOURCE IT!

Price

Can we solve it for 20 cents?

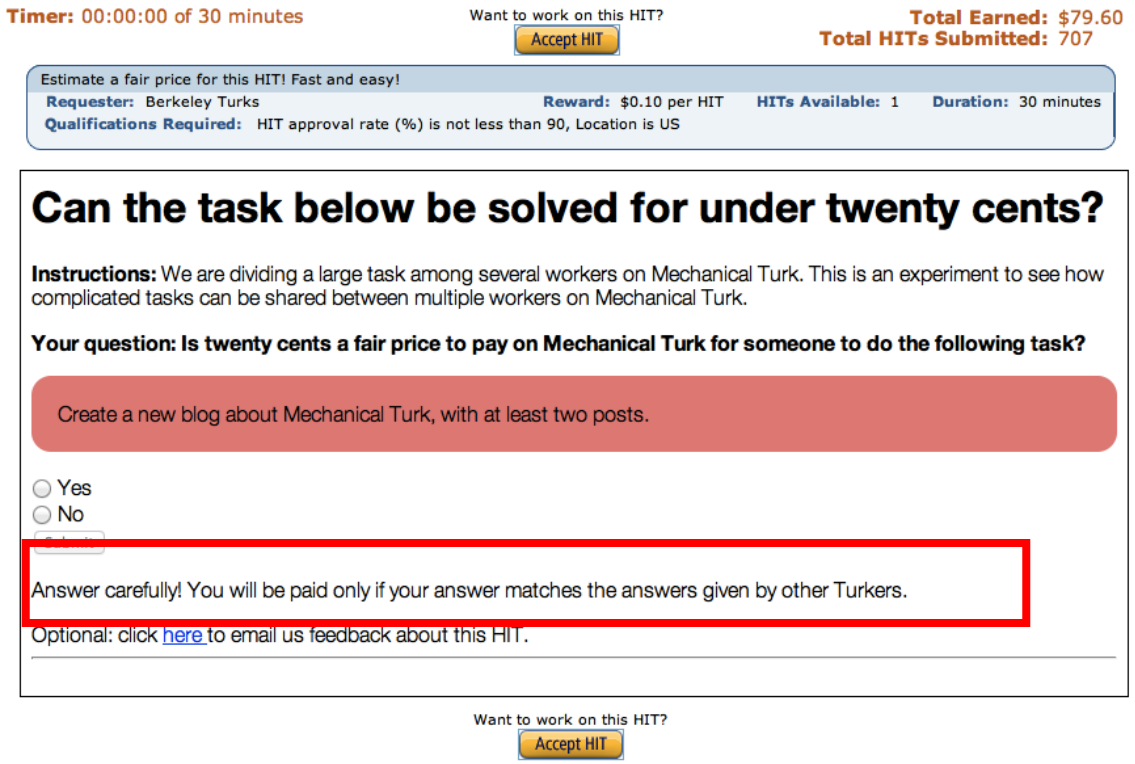
No.

Can this task be solved for 20 cents?

Write a blog about Mechanical Turk

- ○ Yes
- ○ No

Submit



Create a new blog about Mechanical Turk with two posts.

CROWDSOURCE IT!

Divide

Divide it into two or more steps.

Break down the following task.

Write a blog about Mechanical Turk

Step 1:

Step 2:

Add Step | Submit

Create a new blog about Mechanical Turk with two posts.

CROWDSOURCE IT!

Divide

Divide it into two or more steps.

Create a new blog on Wordpress.com.

Write one entry for a blog.

Write a second entry for a blog.

Create a new blog about Mechanical Turk with two posts.

CROWDSOURCE IT!

Price

Create a new blog on Wordpress.com.

Can we solve it for 20 cents?

Yes.

Write one entry for a blog.

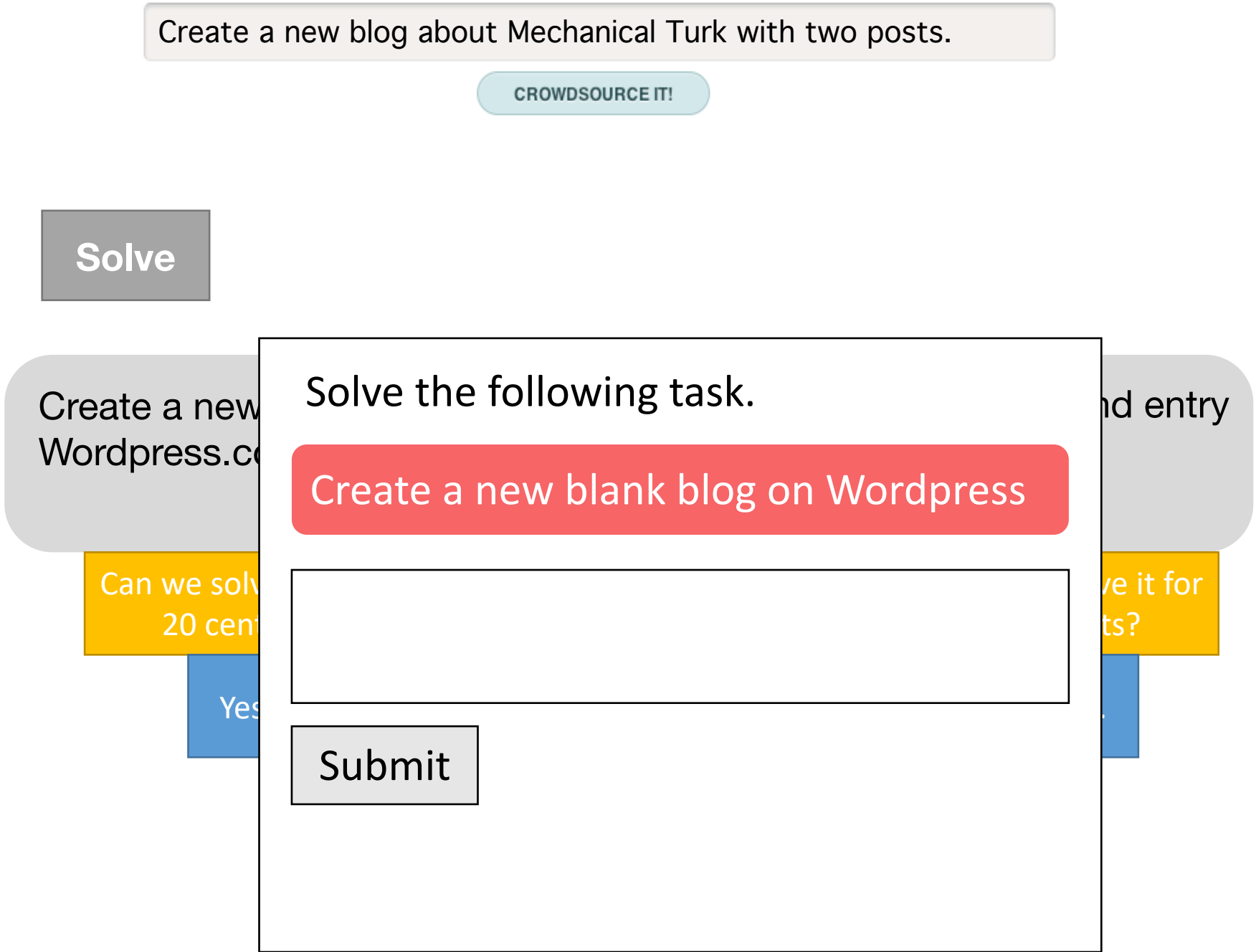
Can we solve it for 20 cents?

Yes.

Write a second entry for a blog.

Can we solve it for 20 cents?

Yes.


Create a new blog about Mechanical Turk with two posts.
CROWDSOURCE IT!
Solve
Create a new
Wordpress.c
nd entry
Can we solv
20 cen
ve it for
ts?
Yes
Solve the following task.
Create a new blank blog on Wordpress
Submit

Create a new blog about Mechanical Turk with two posts.

CROWDSOURCE IT!

Solve

Create a new blog on Wordpress.com.

Write one entry for a blog.

"Welcome to my blog about Mechanical Turk! Here, I'll be posting some of my favorite recipes for Mechanical Turk. You'll be able to follow along at home and create delicious HITs. From the comfort of your own home! Stay tuned and i'll show you some of the best strategies for keeping your Turk workers engaged."

Write a second entry for a blog.

"You may be inclined to price your HITs at the lowest possible rate, but this isn't always the best choice. Instead, you should base your pricing on:

-How long will the HIT take?
-Is the HIT similar to other HITs? If so, price it slightly less than theirs.
-If the HIT involves a lot of qualifications, you may want to price it higher, to attract more qualified workers."

Create a new blog about Mechanical Turk with two posts.

CROWDSOURCE IT!

Merge

Combine the results of solved steps.

Create a new blog on

Write one entry for a blog.

Write a second entry for a blog.

Merge the following subtasks.

Write a blog about Mechanical Turk

Workers previously divided this task into simpler steps and solved each step. Combine their work into a complete solution.

Step 1: Create a blank blog about Mechanical Turk [answer: www...]

Step 2: Write a blog post about Mechanical Turk. [answer: This post is...]

Submit

mtworker.wordpress.com

mtworker

Just another WordPress.com site

Home About

How much should I pay for my HIT?

Posted on April 30, 2011

You may be inclined to price your HITs at the lowest possible rate, but this isn't always the best choice. Instead, you should base your pricing on:

- How long will the HIT take?
- Is the HIT similar to other HITs? If so, price it slightly less than theirs.
- If the HIT involves a lot of qualifications, you may want to price it higher, to attract more qualified workers.

In general, higher priced HITs are great for attracting more skilled workers! You can use this in conjunction with worker qualifications, which I'll talk about next.

Posted in Uncategorized | Leave a comment

Search

Recent Posts

- How much should I pay for my HIT?
- Hello
- Hello world!

Archives

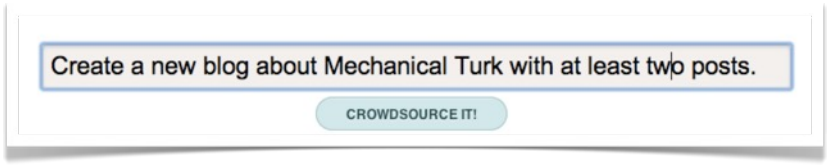
- April 2011

Categories

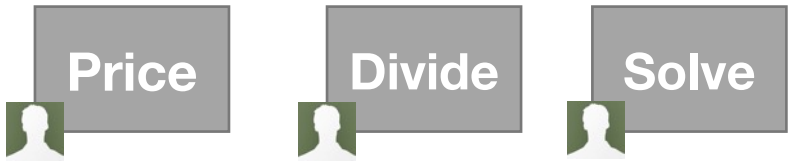
- Uncategorized

Meta

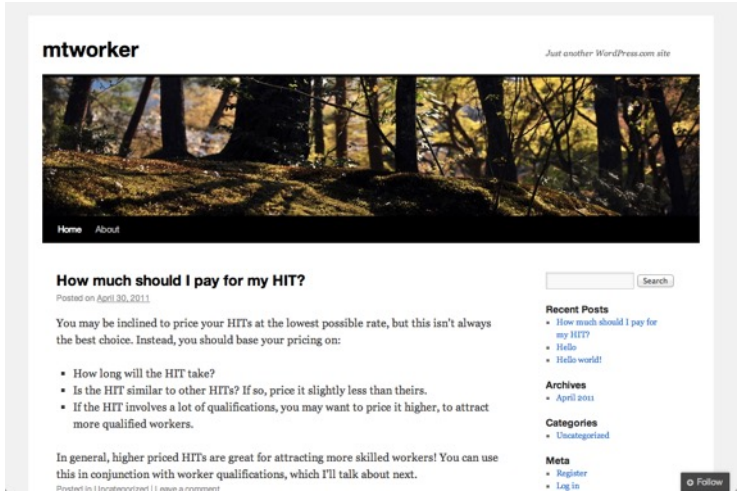
- Register
- Log in



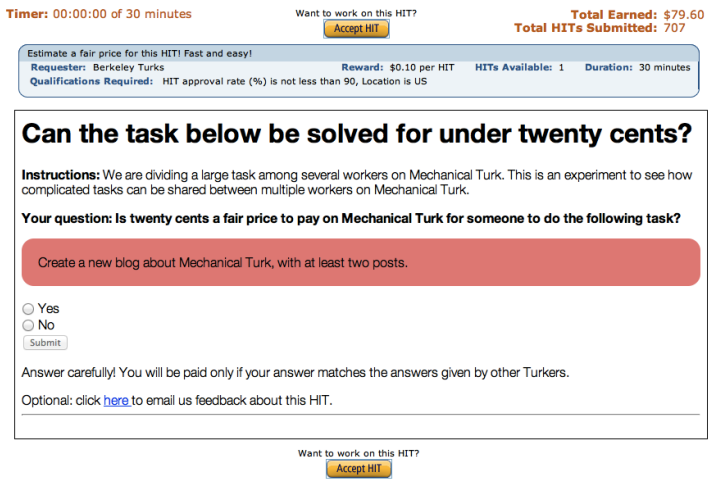
Requester Interface



Algorithm

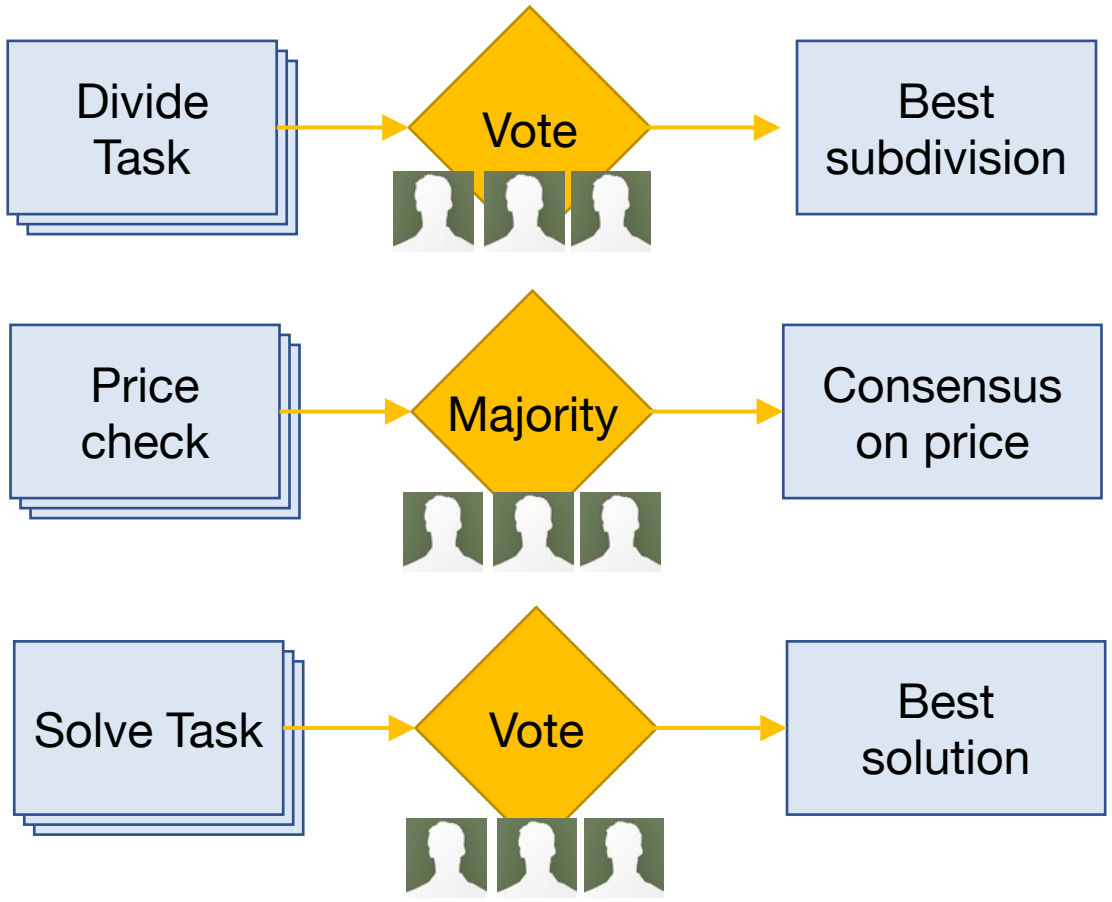


System Output



Worker Interface

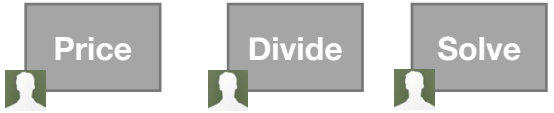
Using Majority Vote to Ensure Quality



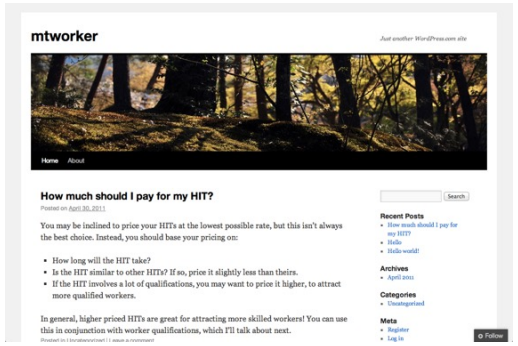
Short Discussion

- Can you come up with scenarios/situations that this approach might fail? What are the potential fixes?

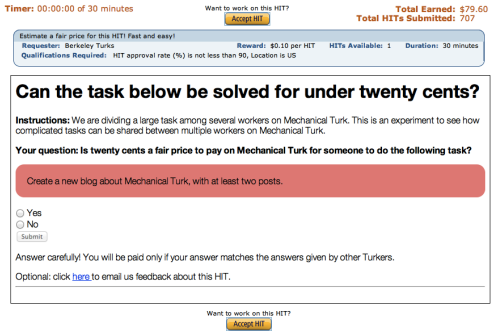
Requester Interface



Algorithm

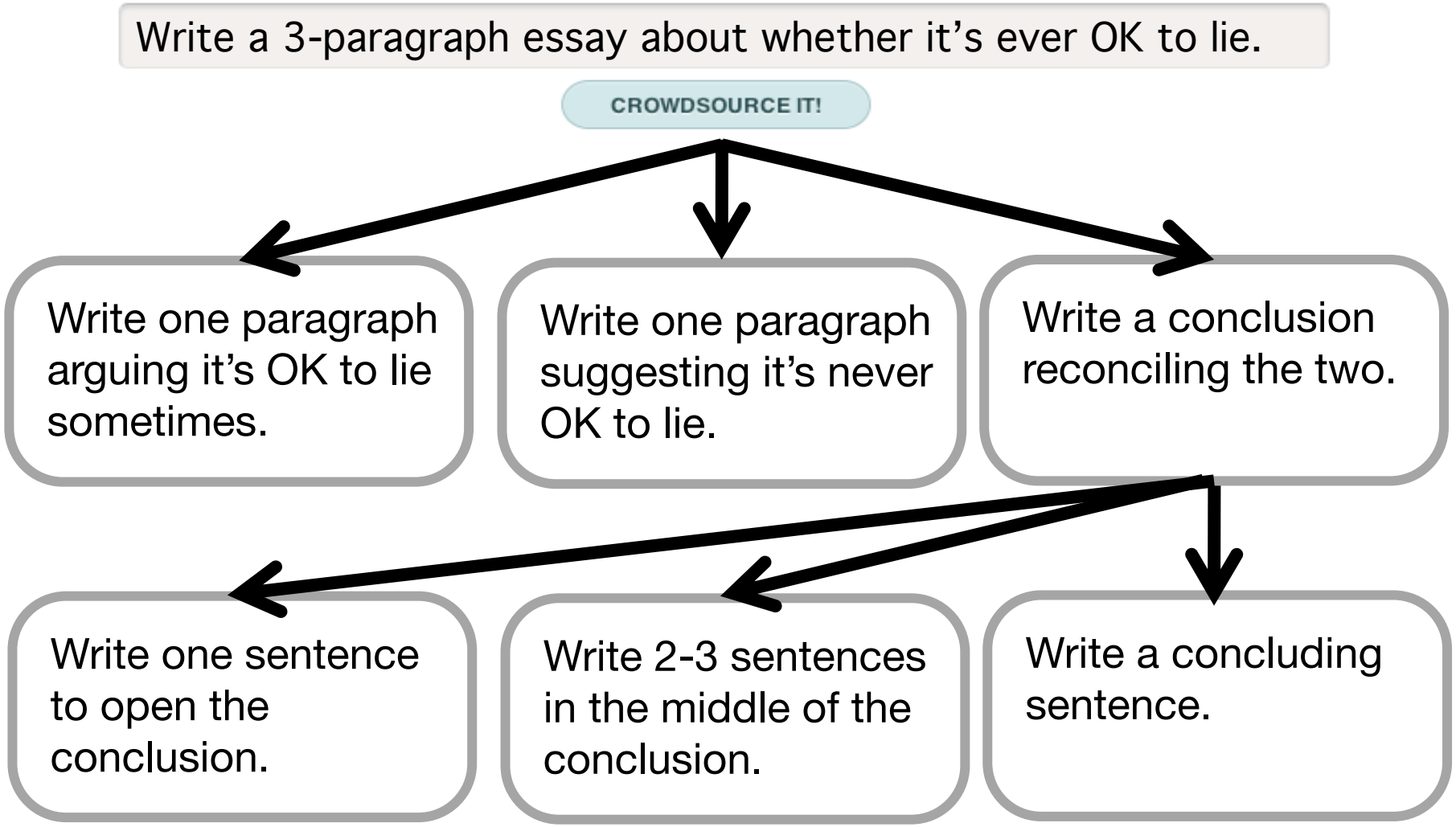


System Output



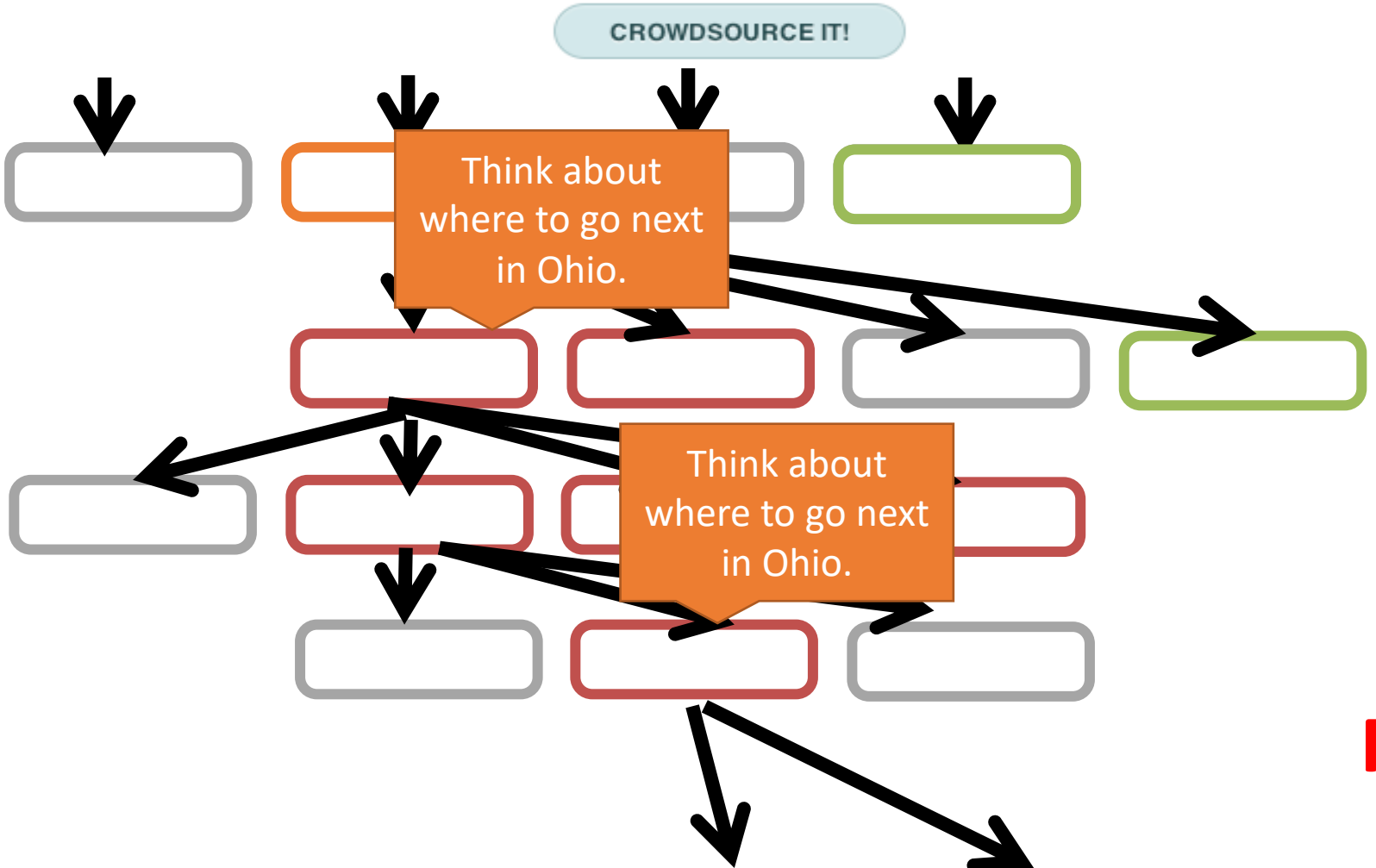
Worker Interface

Success Experiment



Failed Experiment 1

Plan a trip from New York to S.F. that visits 5 interesting places.



Failing to Terminate

Failed Experiment 2

List the department chairs of the top 20 US programs in CS.

CROWDSOURCE IT!

aalto armchair
poang lounge chair
adirondack chair
aeron chair
balans chair
ball chair
....

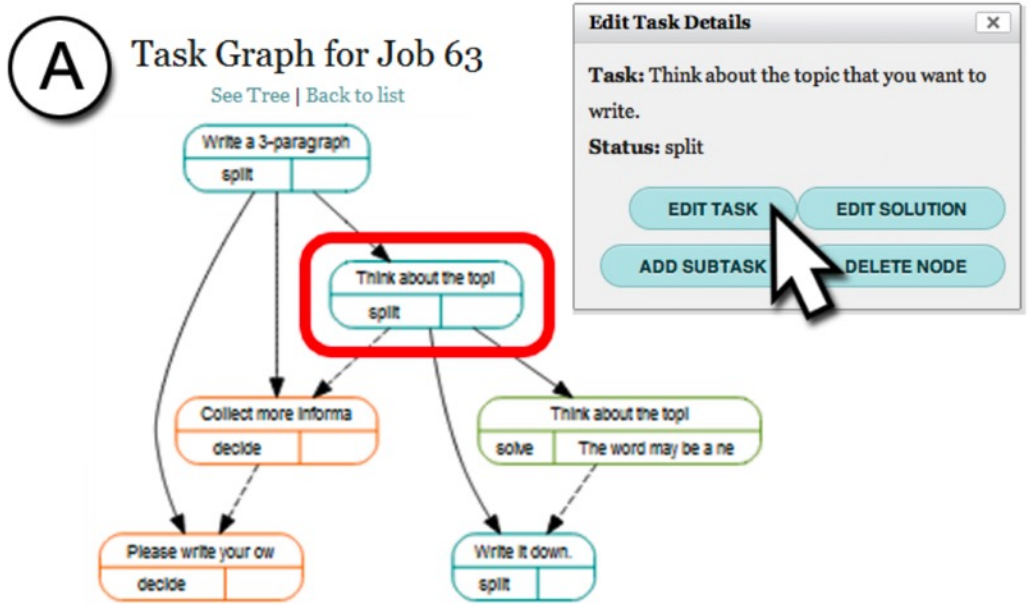
Turkomatic worker:

"…I've taken a look at your instructions, and I understand them perfectly. However, this task seems to have been inadvertently sabotaged by other turkers who do not understand what you are asking them to do…"

Early mistakes propagate

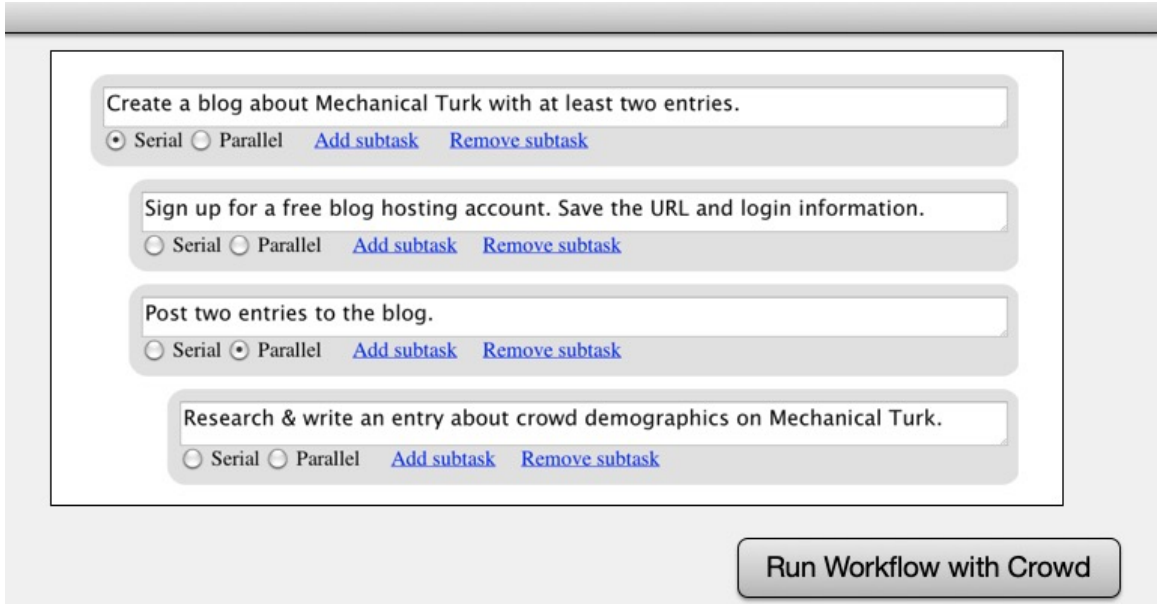
Proposed Fixes

- Utilizing more skilled workers
  - The experiments using graduate students as workers are all successful
- Including requester interventions
  - Designing visualizations, and enable requesters to modify the workflow on the fly



Proposed Fixes

- Utilizing more skilled workers
  - The experiments using graduate students as workers are all successful
- Including requester interventions
  - Designing visualizations, and enable requesters to modify the workflow on the fly
  - Allow requesters to provide seed workflows



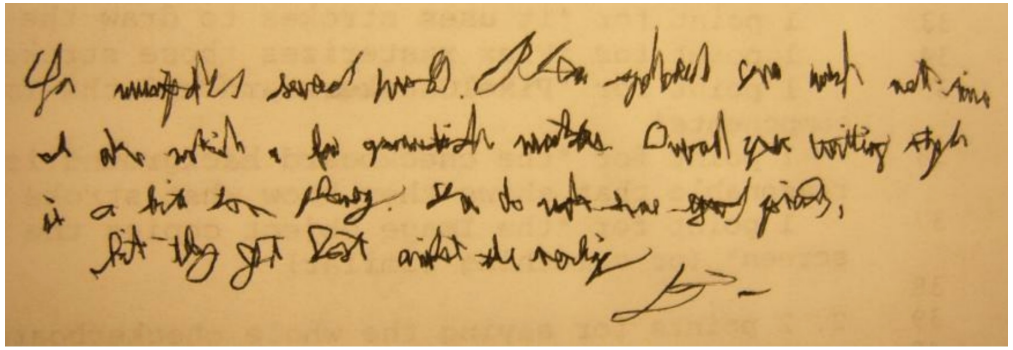
AI-Assisted Workflow Design

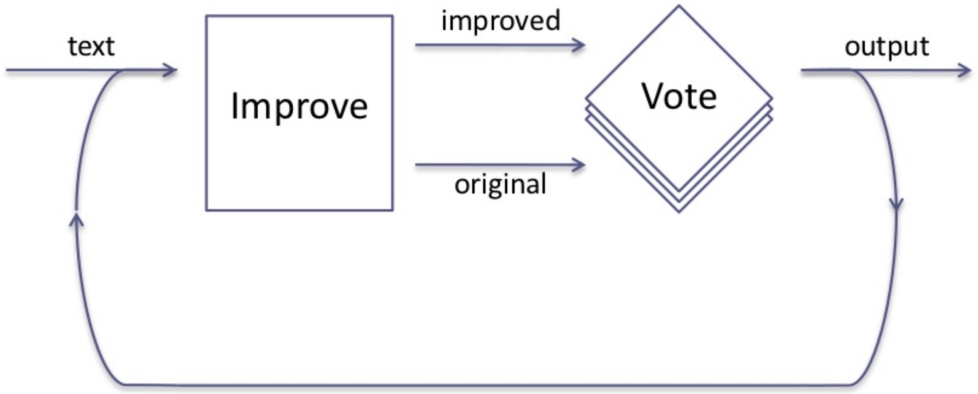
(utilizing AI to help tune the parameters of workflow)

Decision-theoretic Control of Crowd-sourced Workflows.

Peng Dai, Christopher H. Lin, Mausam, Daniel S. Weld. AI Journal 2013.

Example



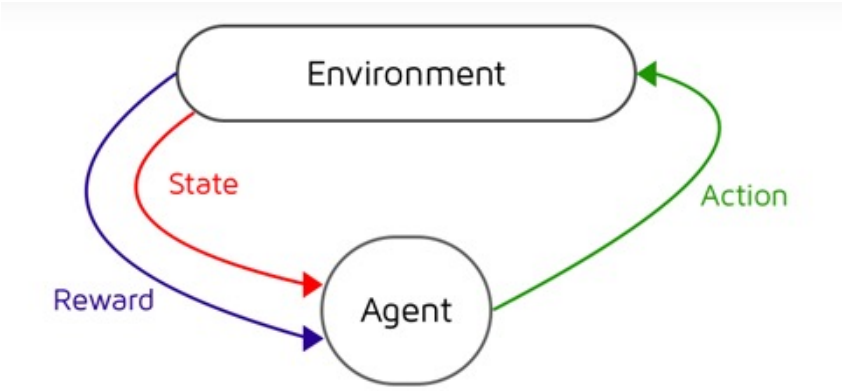


- Decisions to make at each step
  - Should we keep voting?
  - Should we create an improvement task?
  - Should we terminate the process and output the results?

This is a sequential decision-making problem under uncertainty.

Markov Decision Process

- Markov decision process (MDP)

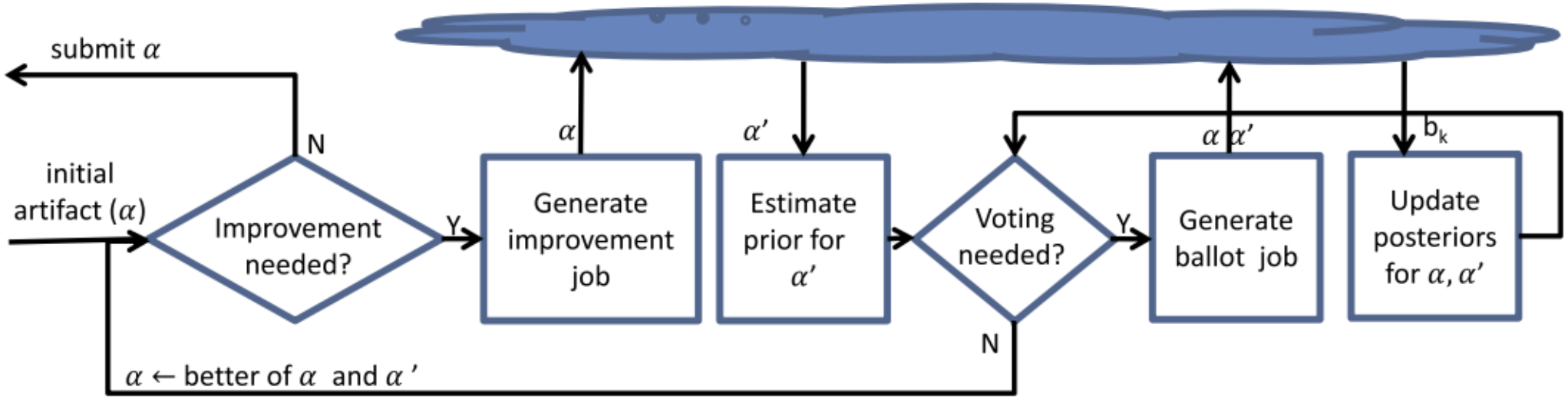


- Main components
  - States (of the environment)
  - Action
  - State transitions
  - Rewards

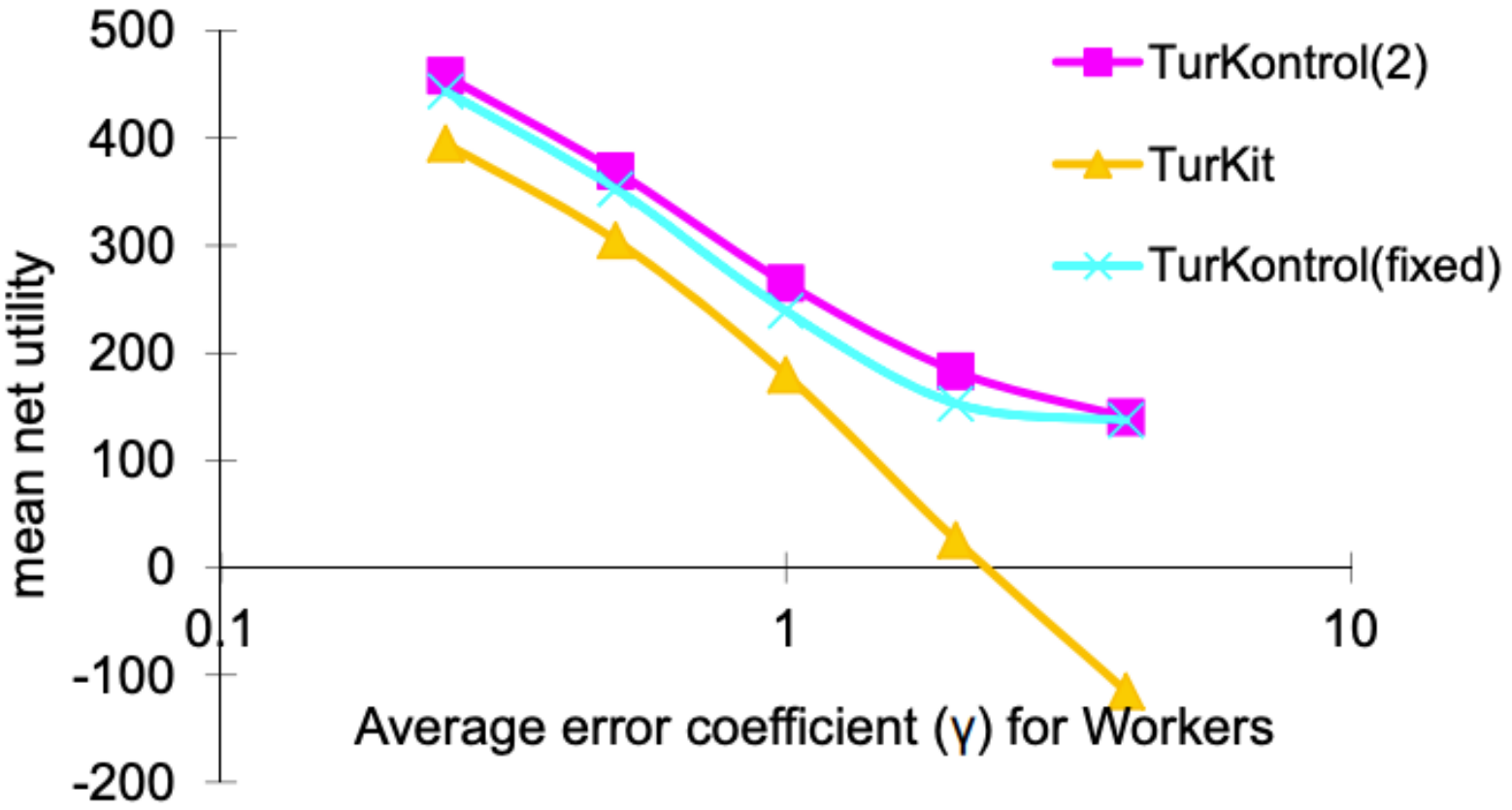
- Partial observable Markov decision process (POMDP)
  - State is not directly observable,
  - We can observe some output/signals depending on the state
    - (think about problem 2 of assignment 3)
  - Require additional reasoning

TurKontrol

- POMDP formulation
  - Actions = {vote, improve, terminate}
  - State = {(q, q')} (q: the original quality of the article, q' the improved one)
  - State transition, observation, rewards are defined w.r.t. to the inference



Improvements over Fixed Parameters



Discussion

- General thoughts on the two approaches
  - Utilizing the crowd to design workflows
  - Utilizing AI to design workflows

- These approaches are obviously not perfect yet. What are the other aspects of workflow design we can utilize AI/crowd to help? How?

Can we use the idea of workflows in our daily tasks?

Selfsourcing Personal Tasks

Teevan, Liebling, and Lasecki. CHI WiP 2014.

How long can you continuously focus on work?

before being interrupted by

- checking emails
- checking Facebook/Instagram/Twitter
- …

Microproductivity: Designing Workflows for Yourself

Task Structure – Break tasks into microtasks
- What we know: Examples of many complex tasks can be broken down
- Still needed: What tasks can be broken down? How to capture context?

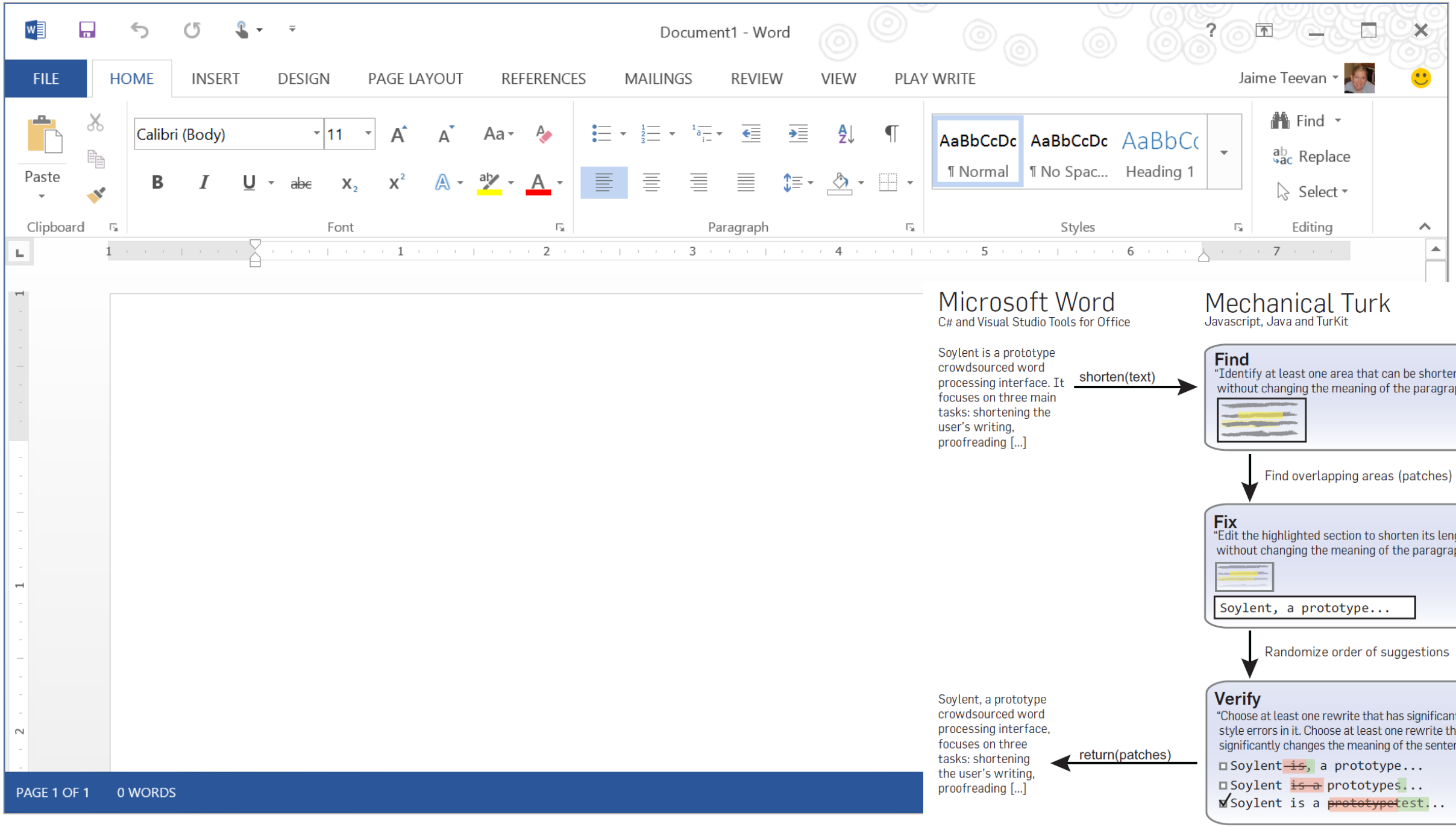
Task Completion – Make it easy to do microtasks
- What we know: Microtasking is easier, especially when mobile
- Still needed: When to break tasks down? How to prioritize microtasks?

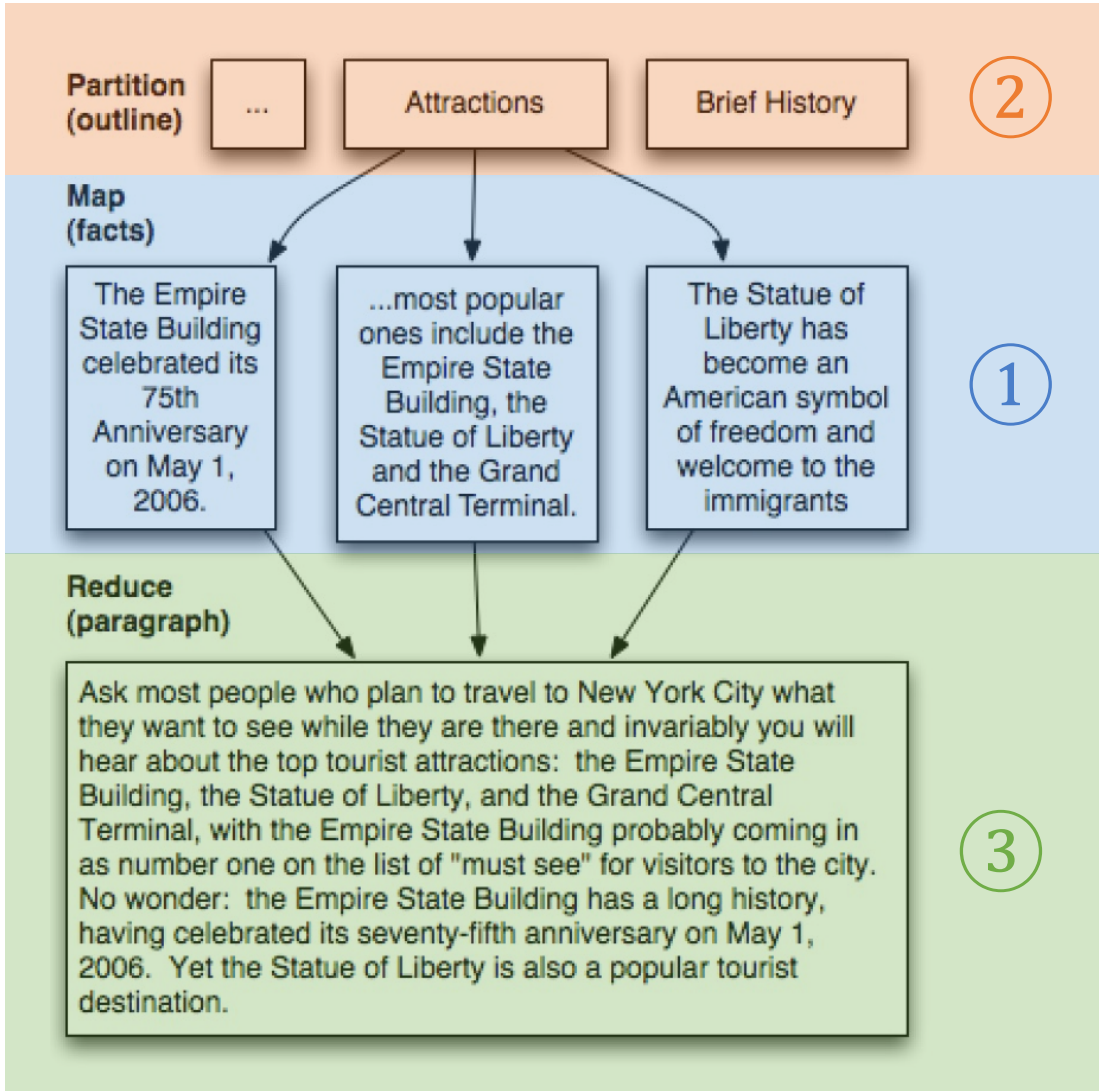
Task Sharing – Get microtasks to the right person
- What we know: Reduce collaboration overhead with colleagues, crowd
- Still needed: Who is the best person to do a microtask?

Task Automation – Learn microtasks via hybrid intelligence
- What we know: Possible to incorporate automation into workflows
- Still needed: When can AI systems benefit from human input?

Example: Article Writing

Slides from Jaime Teevan


Document1 - Word
FILE
HOME
INSERT
DESIGN
PAGE LAYOUT
REFERENCES
MAILINGS
REVIEW
VIEW
PLAY WRITE
Jaime Teevan
Paste
Calibri (Body)
11
Clipboard
Font
Paragraph
Styles
Editing
Find
Replace
Select
AaBbCcDc ¶ Normal
AaBbCcDc ¶ No Spac...
AaBbC Heading 1
PAGE 1 OF 1
0 WORDS
Microsoft Word
C# and Visual Studio Tools for Office
Soylent is a prototype crowdsourced word processing interface. It focuses on three main tasks: shortening the user's writing, proofreading [...]
shorten(text)
Mechanical Turk
Javascript, Java and TurKit
Find
"Identify at least one area that can be shortened without changing the meaning of the paragraph."
Find overlapping areas (patches)
Fix
"Edit the highlighted section to shorten its length without changing the meaning of the paragraph."
Soylent, a prototype...
Randomize order of suggestions
Verify
"Choose at least one rewrite that has significant style errors in it. Choose at least one rewrite that significantly changes the meaning of the sentence."
Soylent is, a prototype...
Soylent is a prototypes...
Soylent is a prototypetest...
return(patches)
Soylent, a prototype crowdsourced word processing interface, focuses on three tasks: shortening the user's writing, proofreading [...]



1. Collect content
2. Organize content
3. Turn content into writing

① Collect Content

The MicroWriter breaks writing into microtasks

Microtasks can be shared with collaborators

Microtasks can be done while mobile

Collaborative writing typically requires coordination

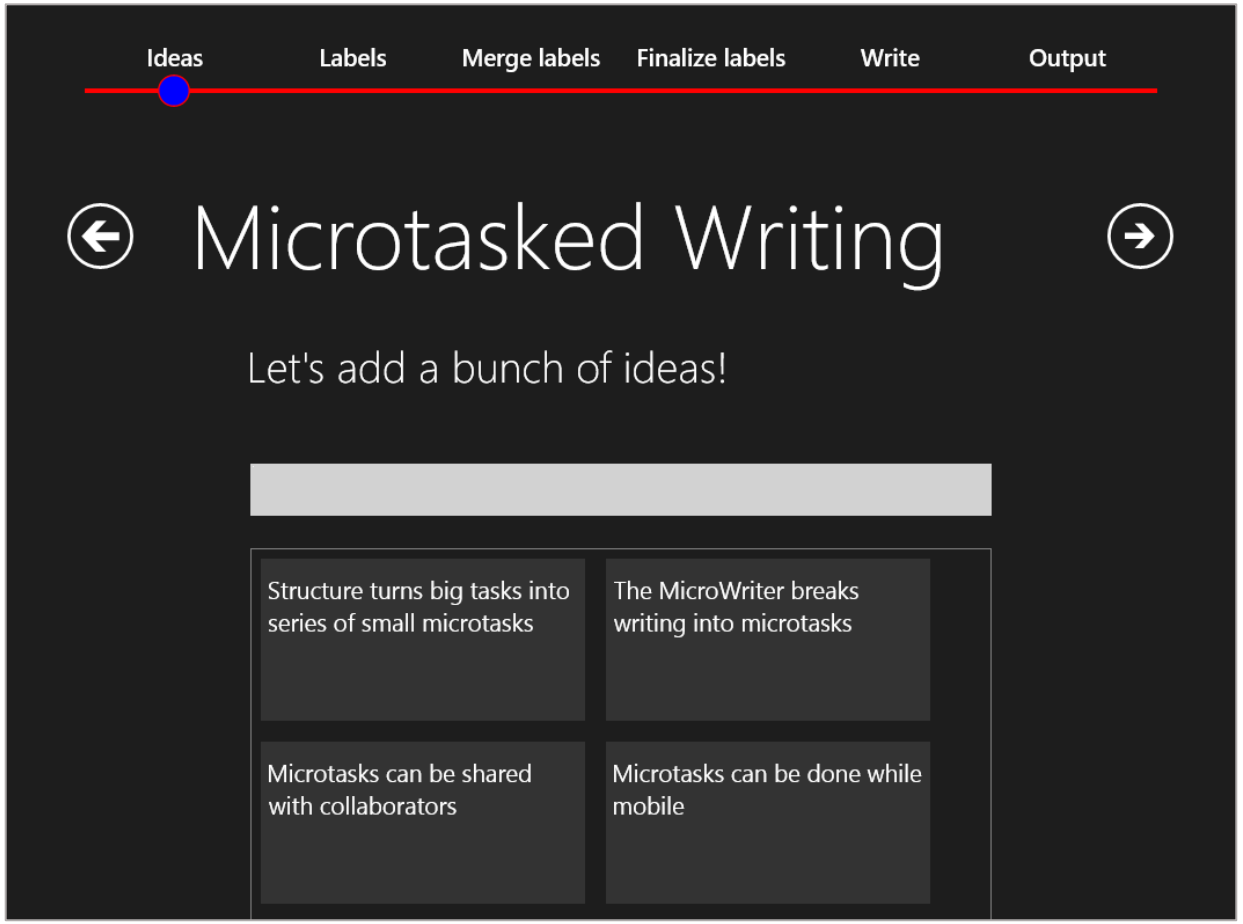
Collaborators can be known or crowd workers

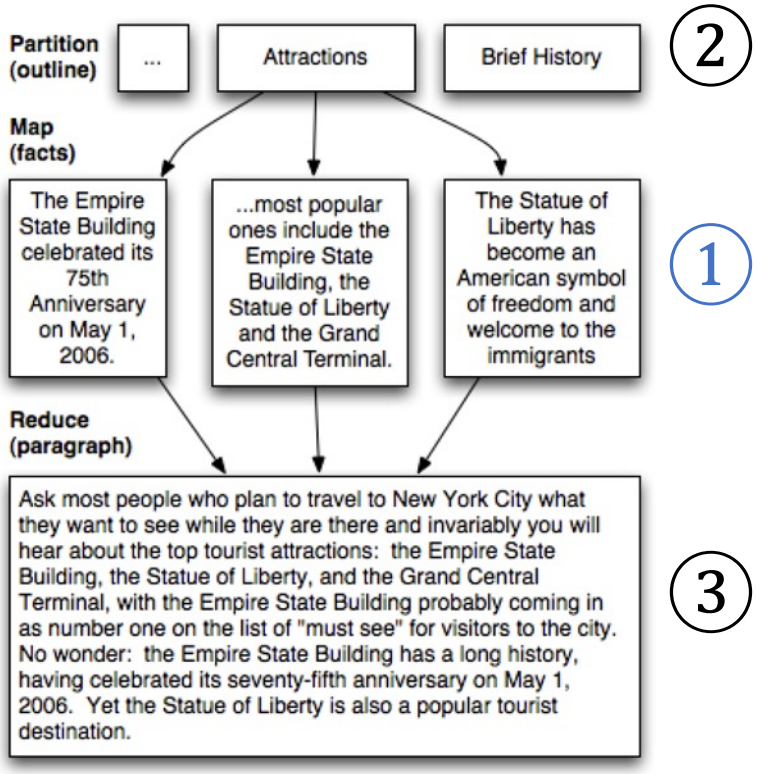
People have spare time when mobile

Structure turns big tasks into series of small microtasks

Microtasks make it easy to get started

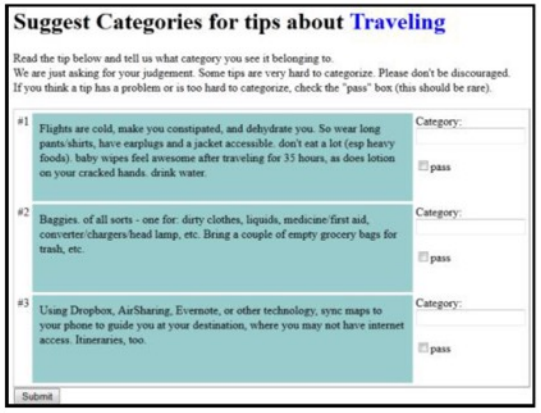
① Collect Content

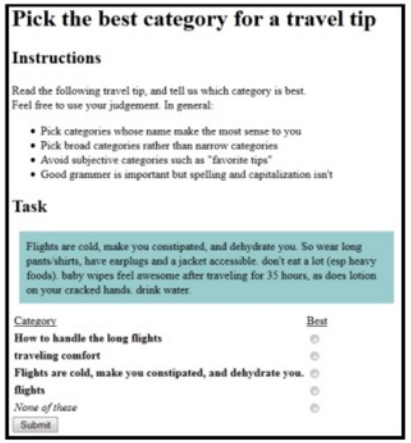




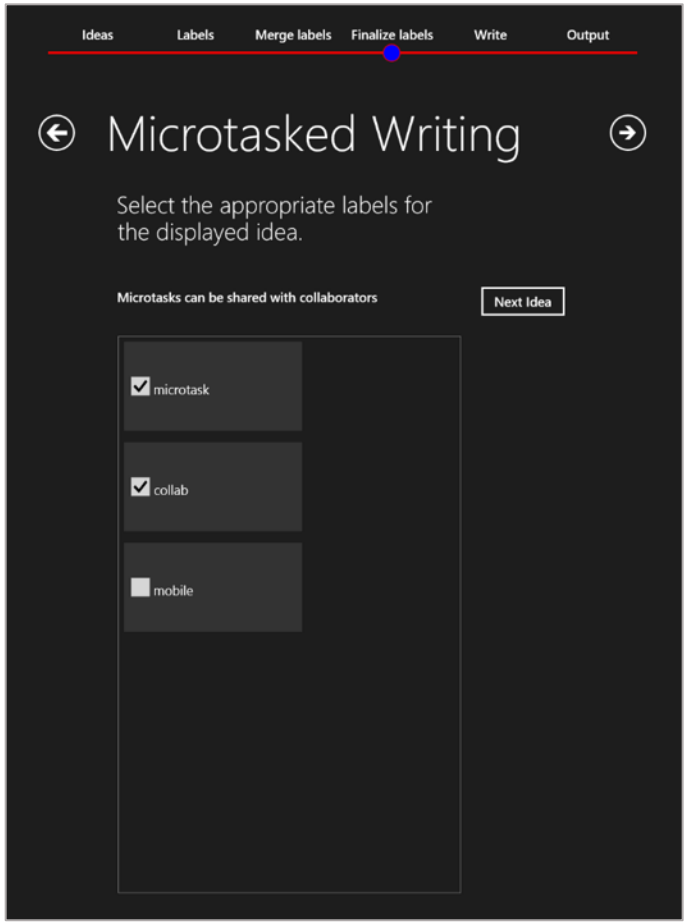
1. Collect content
2. Organize content
3. Turn content into writing

1

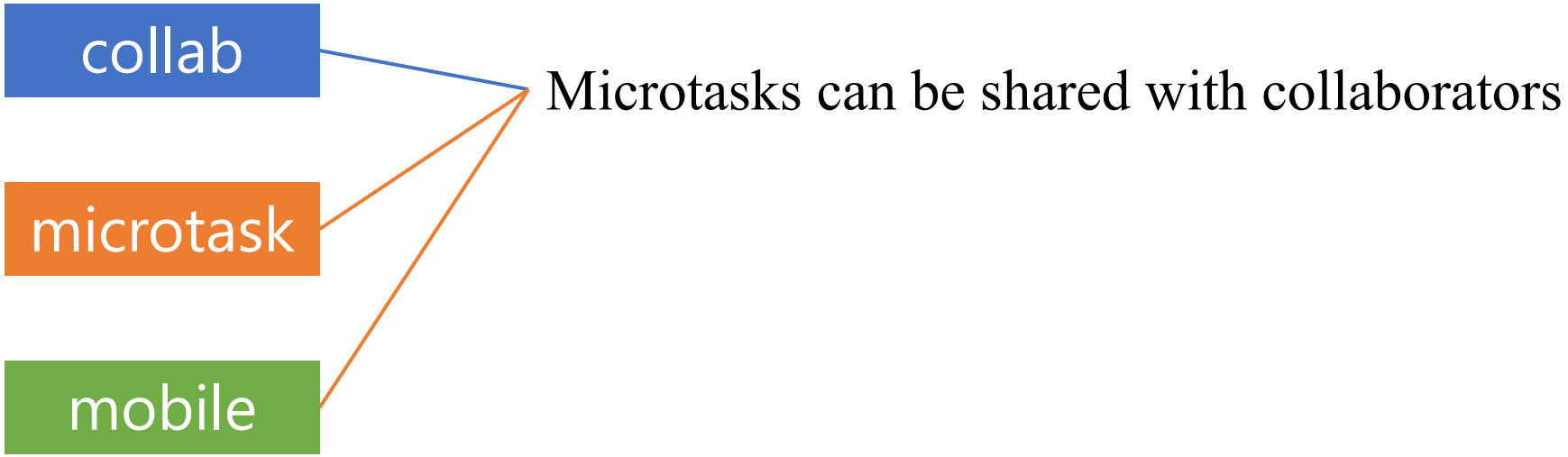




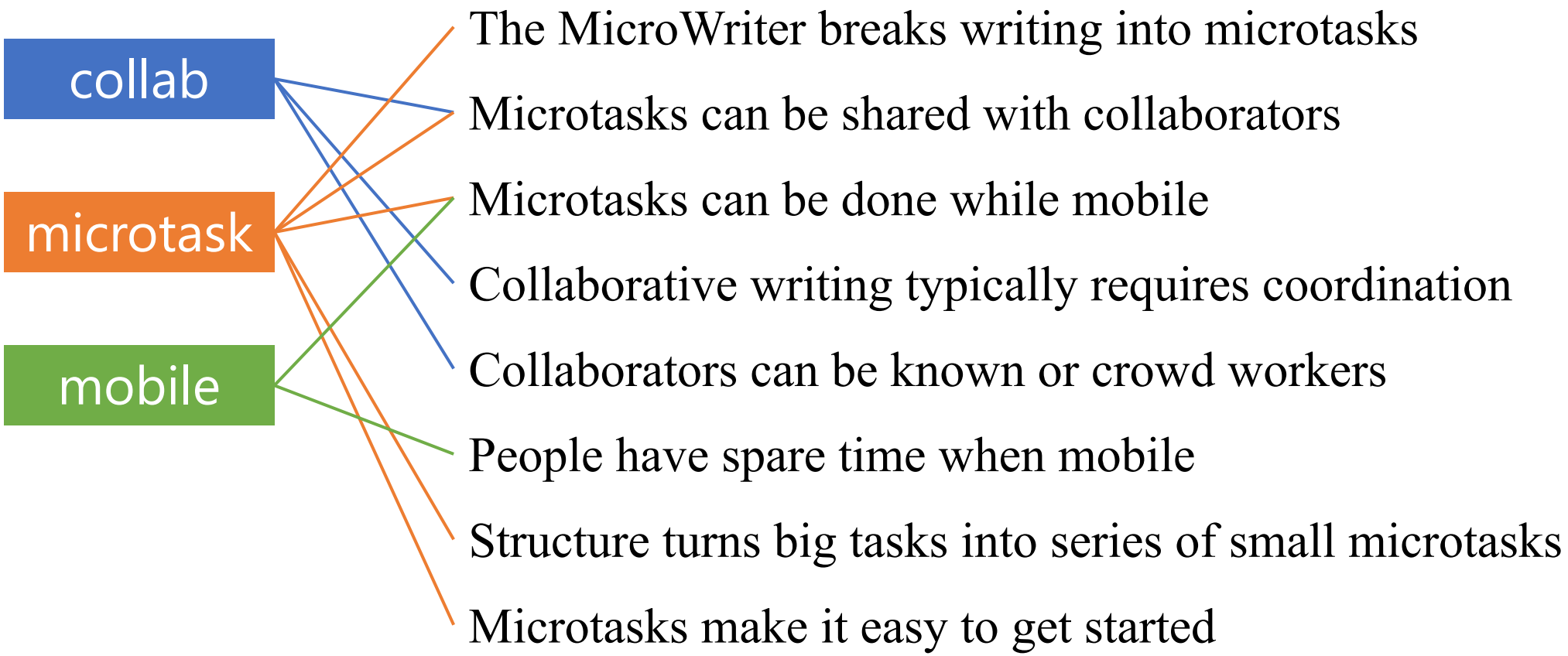
② Organize Content

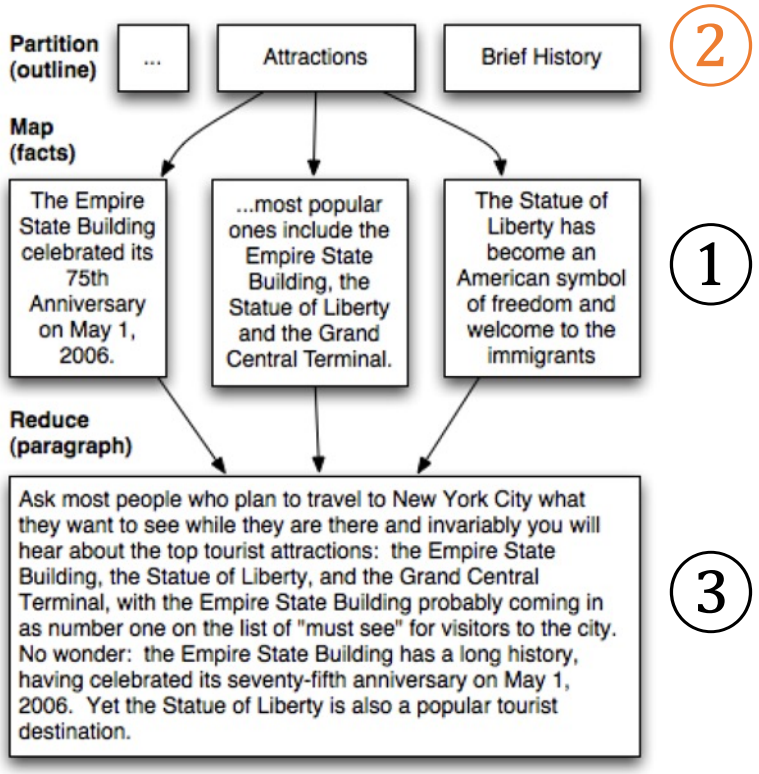


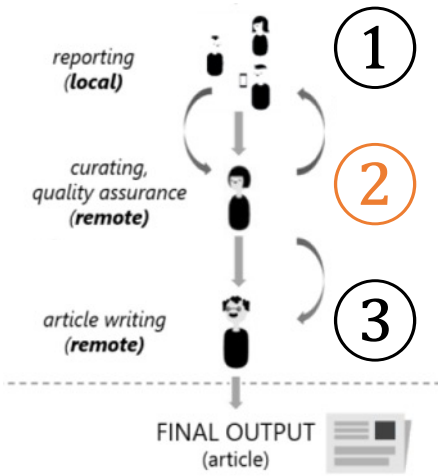
② Organize Content



② Organize Content





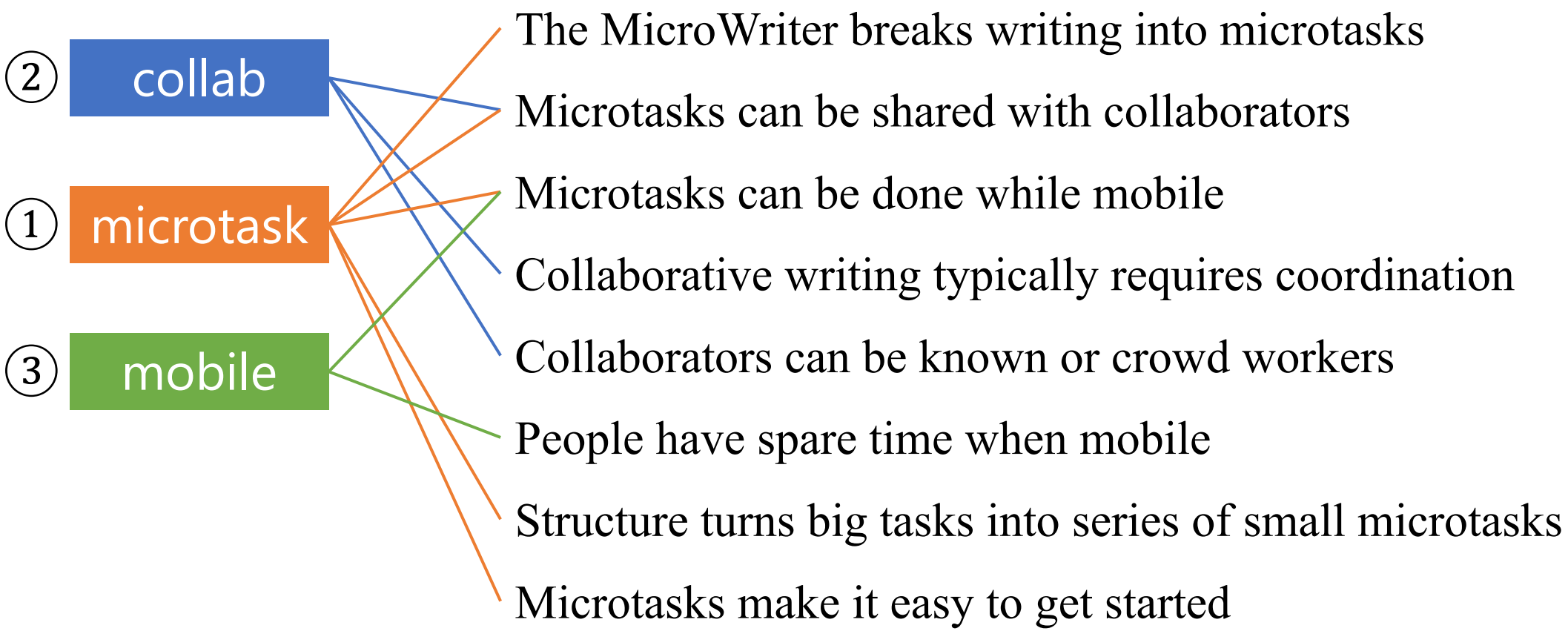


1. Collect content
2. Organize content
3. Turn content into writing

1

2

③ Turn Content into Writing



③ Turn Content into Writing

microtask

- The MicroWriter breaks writing into microtasks
- Structure turns big tasks into series of small microtasks
- Microtasks make it easy to get started

collab

- Microtasks can be shared with collaborators
- Collaborative writing typically requires coordination
- Collaborators can be known or crowd workers

mobile

- Microtasks can be done while mobile
- People have spare time when mobile

③ Turn Content into Writing

③ Turn Content into Writing

collab

- Microtasks can be shared with collaborators
- Collaborative writing typically requires coordination
- Collaborators can be known or crowd workers

Collaborative writing typically requires coordination. However, microtasks are easy to share with collaborators without the need for coordination. The collaborators can be known colleagues, or paid crowd workers.

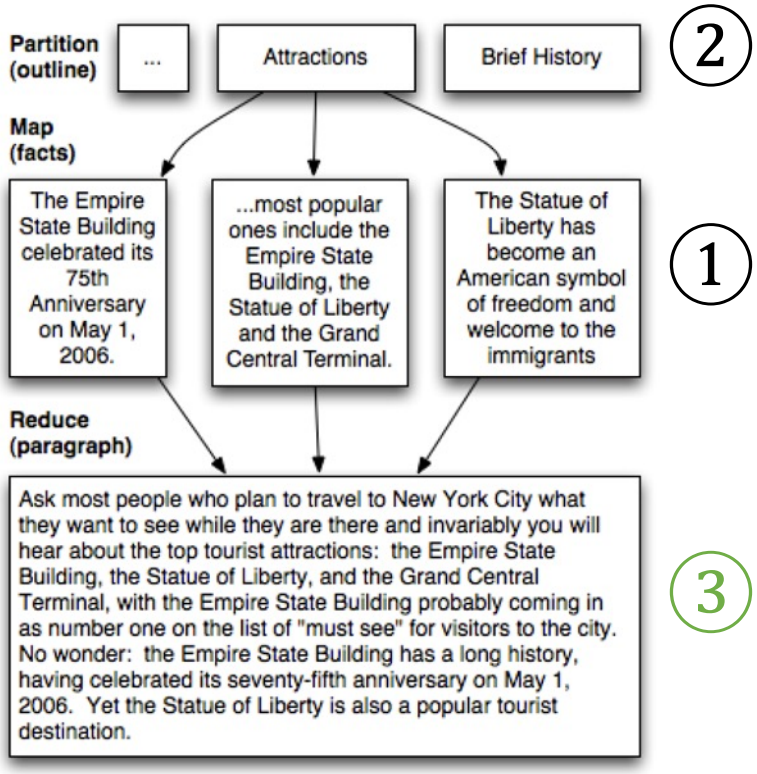
③ Turn Content into Writing

- Complete output:

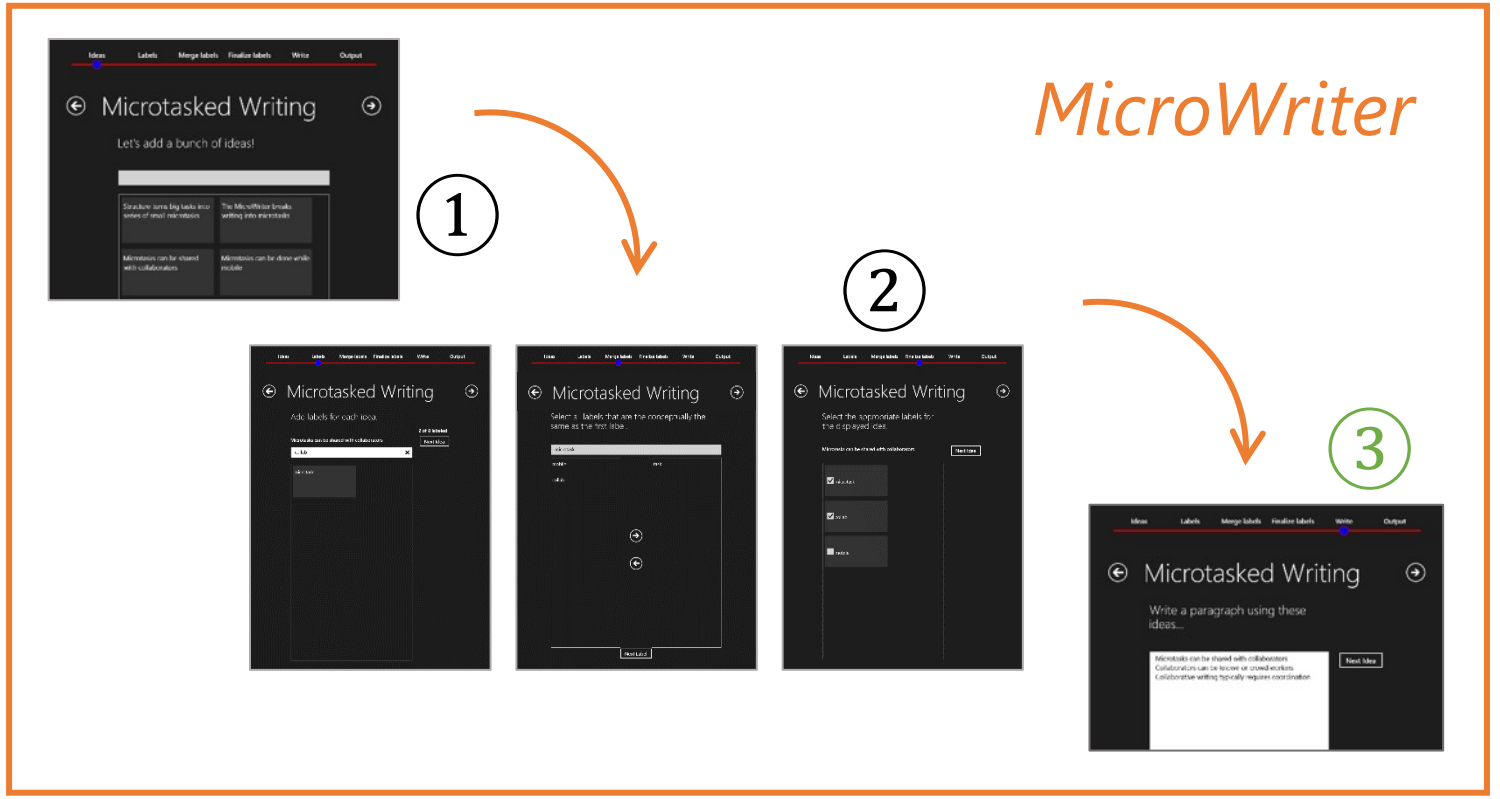
Structure makes it possible to turn big tasks into a series of smaller microtasks. For example, the MicroWriter breaks writing into microtasks. These microtasks make the larger task easier to start.

Collaborative writing typically requires coordination. However, microtasks are easy to share with collaborators without the need for coordination. The collaborators can be known colleagues, or paid crowd workers.

People have spare time when mobile, and these micromoments are ideal for doing microtasks.



1. Collect content
2. Organize content
3. Turn content into writing



Discussion

- What tasks in your life do you think that might benefit through adopting microproductivity?

- How should you decompose the task? Can you share some of the microtasks? Can you automate some of the microtasks?