How to Setup a Crowdsourcing Task

Lecture 3
Gianluca Demartini
University of Sheffield
Outline

• Micro-task Crowdsourcing Challenges
  – Design the User Interfaces
  – Define the right Incentives
  – Task Patterns
  – Quality
  – Scalability
Design of a Task on MTurk
A Task on MTurk

Choose the best category for this image

View Instructions↓

Select the room location in home for this picture. Seating areas outside are outside not living. Offices or dens are living not bedrooms. Bedrooms should contain a bed in the picture.
High-level Issues in Crowdsourcing

• Process
  – Experimental design, annotation guidelines, iteration

• Choose crowdsourcing platform (or roll your own!)

• Human factors
  – Payment / incentives, interface and interaction design, communication, reputation, recruitment, retention

• Quality Control / Data Quality
  – Trust, reliability, spam detection, consensus labeling
Typical Workflow

- Define and design what to test
- Sample data
- Design the experiment
- Run experiment
- Collect data and analyze results
- Quality control
Task Design

- One of the most important parts
- Part art, part science
- Instructions are key
- Prepare to iterate
Task Design

• Ask the right questions
• Workers may not be experts so don’t assume the same understanding in terms of terminology
• Instructions matter!
• Show examples
• Hire a technical writer
  – Engineer writes the specification
  – Writer communicates
Task Design - UI

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

• Presentation
• Document design
• Highlight important concepts
• Colors and fonts
• Need to grab attention
• Localization
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)
• Cognitive load: mental rigor needed to perform task
Bad Example

- Asking too much, task not clear, “do NOT/reject”
- Worker has to do a lot of stuff

Help us describe How-To Videos! Earn $2.50 bonus for every 25 videos entered!

Watch a how-to video, and write a keyword-friendly synopsis describing the video.

1. Click on the link to watch the Film & Theater how-to video – 332492 Get a 35mm film look with a depth of field adapter
2. Write a description of the video linked in 4 or more sentences.
3. Be detailed in your description. Describe how the procedure is done.
4. Description should be at least 100 words.
5. Description should be fewer than 2000 characters.
6. Use the character and word counters below to help you stay within the limits.
7. You must complete 25 video descriptions in order to earn the $2.50 bonus. Bonuses are distributed after HITs have been completed. The more HITs completed and approved, the more you will earn.
8. It is not necessary to repeat the headline in your entry. It will NOT count toward your word count.
9. Do NOT describe the following: the format, where the video comes from, or how long the video is. This information is IRRELEVANT.
10. Do NOT describe the video in the following manner: “She turns around to face the camera. Then she faces left.” Follow the examples below.

Current Word Count: 0  Current Character Count: 0 / 2000

Criteria for REJECTION:

1. Entries with obvious and multiple spelling or grammatical errors will be rejected.
2. Entries with fewer than 100 words will be automatically rejected.
3. Text copied from the web or other places will be rejected. Multiple plagiarized answers will lead to being BLOCKED. You may use a quotation, but the majority of your content must be ORIGINAL.
4. Incomplete and blank answers will be rejected. Multiple blank answers will result in being blocked.
5. Tasks submitted without descriptions will be rejected.
6. Tasks submitted with inaccurate descriptions will be rejected as well.
7. Do NOT add any personal opinions. Entries with personal opinions or reviews will be automatically REJECTED.
8. If you notify us that a link is broken, we appreciate it but will not be able to accept the submission. The notification will result in rejection.
9. Entries that transcribe the video will be REJECTED.
Good Example

• All information is available
  – What to do
  – Search result
  – Question to answer
Form and Metadata

- Form with a close question (binary relevance) and open-ended question (user feedback)
- Clear title, useful keywords
- Workers need to find your task

<table>
<thead>
<tr>
<th>Describe your HIT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Title</strong></td>
</tr>
<tr>
<td>Describe the task to workers. Be as specific as possible, e.g. &quot;answer a survey about movies&quot;, instead of &quot;short survey&quot;, so workers know what to expect.</td>
</tr>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Give more detail about this task. This gives workers a bit more information before they decide to view your HIT.</td>
</tr>
<tr>
<td><strong>Keywords</strong></td>
</tr>
<tr>
<td>Provide keywords that will help workers search for your HITs.</td>
</tr>
</tbody>
</table>
How Much to Pay?

• Price commensurate with task effort
  – Ex: $0.02 for yes/no answer + $0.02 bonus for optional feedback

• Ethics & market-factors
  – e.g. non-profit SamaSource contracts workers refugee camps

• Uptake & time-to-completion vs. Cost & Quality
  – Too little $$, no interest or slow
  – too much $$, attract spammers

• Accuracy & quantity
  – More pay = more work, not better (W. Mason and D. Watts, 2009)
Development Framework

• Similar to a UX
• Build a mock up and test it with your team
  – Yes, you need to do some tasks
• Incorporate feedback and run a test on MTurk with a very small data set
  – Time the experiment
  – Do people understand the task?
• Analyze results
  – Look for spammers
  – Check completion times
• Iterate and modify accordingly
Development Framework

• Introduce quality control
  – Qualification test
  – Gold answers (honey pots)
• Adjust passing grade and worker approval rate
• Run experiment with new settings & same data
• Scale on data
• Scale on workers
Quality Control

• Extremely important part of the experiment
• Approach 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.
Quality Control

• Approval rate: easy to use, & just as easily defeated
• Mechanical Turk Masters
  – Recent addition, only for specific tasks
• Qualification test
  – Pre-screen workers’ ability to do the task (accurately)
• Assess worker quality as you go
  – Trap questions with known answers ("honey pots")
  – Measure inner-annotator agreement between workers
Qualification tests: pros and cons

• Advantages
  – Great tool for controlling quality
  – Adjust passing grade

• Disadvantages
  – Extra cost to design and implement the test
  – May turn off workers, hurt completion time
  – Refresh the test on a regular basis
  – Hard to verify subjective tasks like judging relevance

• Try creating task-related questions to get worker familiar with task before starting task in earnest
Methods for measuring agreement

• What to look for
  – Agreement, reliability, validity
• Inter-agreement level
  – Agreement between judges
  – Agreement between judges and the gold set
• Some statistics
  – Percentage agreement
  – Cohen’s kappa (2 raters)
  – Fleiss’ kappa (any number of raters)
• With majority vote, what if 2 say relevant, 3 say not?
  – Use expert to break ties
  – Collect more judgments as needed to reduce uncertainty
Quality Control & Assurance

• Filtering
  – Approval rate (built-in but defeatable)
  – Geographic restrictions (e.g. US only, built-in)
  – Worker blocking
  – Qualification test
    • Con: slows down experiment, difficult to “test” relevance
    • Solution: create questions to let user get familiar before the assessment
  – Does not guarantee success

• Identify workers that always disagree with the majority
• Ask workers to rate the difficulty of a task
Other quality heuristics

• Justification/feedback as quasi-captcha
  – Should be optional
  – Automatically verifying feedback was written by a person may be difficult (classic spam detection task)

• Broken URL/incorrect object
  – Leave an outlier in the data set
  – Workers will tell you
  – If somebody answers “excellent” for a broken URL => probably spammer
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

• If spammer “caught”, block from future tasks
  – May be easier to always pay, then block as needed
Build Your Reputation as a Requestor

• Word of mouth effect
  – Workers trust the requester (pay on time, clear explanation if there is a rejection)
  – Experiments tend to go faster
  – Announce forthcoming tasks (e.g. tweet)
Crowd Worker Communities

Turkopticon.com
Mturkforum.com
Turkernation.com
Summary

• Things that work
  – Qualification tests
  – Honey-pots
  – Good content and good presentation
  – Economy of attention

• Things to improve
  – Manage workers in different levels of expertise including spammers and potential cases.
  – Mix different pools of workers based on different profile and expertise levels.
What can go wrong?

• Low-quality results can be due to:
  – Bad instructions
  – Pay not high enough or too high
  – Not enough assignments: ask multiple times

• Answer aggregation
  – Majority vote
  – Weighted average of answers
  – ZenCrowd (learn weights for workers)
  – Aggregate based on worker similarity
Crowdsourcing Patterns
Microtask vs Macrotask

Microtask vs Macrotask

• Longer to perform a task using microtasks than macro-tasks.

• Micro-task: higher quality work, easier to complete, robust to interruption

• Task decomposition may be difficult
Crowdsourcing Patterns

• Majority Vote Aggregation
  – Select the answer among a set of candidates
  – Pick the most popular answer

• Find-Fix-Verify
  – Creative process
  – Three-steps iterative crowdsourcing

• Interaction Protocol (for hybrid human-machine systems)
  – Upfront
  – Iterative
Interaction Protocol

How often can we refer to the crowd?

1. **Upfront**: Ask all the B queries at once

2. **Iterative**: Ask K queries to the crowd and use them to improve the system. Repeat this B/K times
Measures Used for Selection

- **Uncertainty**: Asking hardest (most ambiguous) questions
- **Explorer**: Ask questions with potential to have largest impact on the system
Soylent: Find-Fix-Verify

Find

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

Independent agreement to identify patches

Fix

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

Randomize order of suggestions

Verify

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

Find-Fix-Verify

• Machine Translation example

• Find
  – Show automatically translated text
  – Ask if they are grammatically correct

• Fix
  – Ask to translate those which contain errors (multiple times)

• Verify
  – Select the best translation among the available ones
References

• “Crowdsourcing for Information Retrieval: Principles, Methods, and Applications” SIGIR 2011 Tutorial.


• “When to Ask a Noisy Crowd: Active Learning Meets Crowd” Barzan Mozafari et al.