Amazon Mechanical Turk

Requester Best Practices Guide

Amazon Mechanical Turk:

Requester Best Practices Guide

Contents

Planning, Designing and Publishing Your Project

Amazon Mechanical Turk is a marketplace for work where businesses (aka Requesters) publish tasks (aka HITS), and human providers (aka Workers) complete them. Amazon Mechanical Turk gives businesses immediate access to a diverse, global, on-demand, scalable workforce and gives Workers a selection of thousands of tasks to complete whenever and wherever it's convenient.

There are many ways to structure your work in Mechanical Turk. This guide helps you optimize your approach to using Mechanical Turk to get the most accurate results at the best price with the turnaround time your business needs. Use this guide as you plan, design, and test your Amazon Mechanical Turk HITs.

Topics

What is an Amazon Mechanical Turk HIT?

An Amazon Mechanical Turk HIT or Human Intelligence Task is the task you ask a Worker to complete. It may be a task that is inherently difficult for a computer to do.

Divide Your Project Into Steps

In the planning stage, define the goal of your project and break it down into specific steps.

Suppose your website provides a directory of businesses for consumers to search. You want to make sure that each business in your directory is properly categorized (restaurant, dry cleaner or grocery store) and that each address and phone number is correct. Your project is to clean your directory database. But you shouldn't ask a Worker to "clean my database".

Instead you want to structure the project so that many Workers can work in parallel to get your project done faster. You can do this by having one Worker categorize and verify the address and phone number for one business.

Keep HITs Simple

In our example above asking each Worker to perform the categorization as well as the verification of address and phone number is not efficient for the Worker. They have to "switch gears" from categorization to address and phone number verification (which may require a website search or a phone call). It's better to have a Worker perform one type of task in one HIT. So rather than ask for both categorization and verification in one HIT, have a HIT that just asks for categorization and a separate HIT that asks for verification. This keeps the Worker focused on one activity for each group of HITs. It also allows you to give the categorizations HITs to Workers who are good at categorization and the verification HITs to Workers who are good at that.

Tip

> Keep in mind that a HIT should target a particular Worker capability or skill. Try not to create HITs that require the Worker to have several different capabilities.

Make Instructions Clear and Concise

The answers you receive from Workers are only as good as the instructions you provide. To give accurate answers, Workers must understand the questions, and must have clear direction on what is or is not acceptable. Here are some suggestions for how to write good instructions:

- **Be as *specific* as possible in your instructions**

  Asking a Worker `'Is a Mazda Miata a sports car?'` is not the same as asking `'Can a Mazda Miata accelerate from 0 to 60 mph in 5 seconds or less?'` The second question provides a common definition of a "sports car" so Workers have a common frame of

reference when answering your question. Whenever possible provide Workers with specific criteria in your instructions.   Likewise asking a Worker `Is this photo offensive?` is very different than asking if it contains nudity. Even 'nudity' can have different meanings (frontal, total). The more specific you are in your instructions the more accurate and consistent your results will be.

The instruction `Provide tags for each image`, is not specific and is open to Worker interpretation. However, `Provide 3 one-word tags for each given image` sets a very specific objective for the Worker and allows the Requester to assess accuracy easily.

- **Make Instructions easy to read**

  Use bulleted lists and short, clear sentences.

  The following is an example of long, complicated, and poorly conceived instructions:

  ```
  Your job is to go to any resource you like and find restaurant
  reviewers in the specified area, include their names, addresses, and
  any other contact information you can find, and then specify whether
  they work for a particular publisher and, if so, for how long.
  ```

  The following instructions are better:

  ```
  1. Go to the link provided to view an article in the Food and Dining
     section of the NY Times.

  2. Copy the name of the reviewer from the by-line and paste it into the
     following box.

  3. Click on the name of the reviewer to see a list of his or her
     reviews.

  4. Copy the URL of the list and paste the URL into the following box.
  ```

- **Include examples**

  Examples help make your expectations clear. If you are asking Workers to `Tag each photo with the name of the city, state and country where the photo was taken` For example: `Seattle, Washington, USA.`

- **Specify formatting requirements**

  If you want the answer in a specific format, use UI elements that force that formatting. For instance In previous example, if you want to see the state abbreviation ("WA") instead of Washington provide the state field as a drop-down list.

- **Don't ask for 'All' or 'Every'**

  Don't ask for 'all' or 'every' as the Workers will only do work that they believe they can successfully complete. If you ask for `every tag appropriate for this image`, Workers may avoid the assignments as too difficult or too easily rejected.

- **Explain what will NOT be accepted**

  If you ask for 3-5 tags for an image, but 3 are required indicate assignments with less than 3 will

be rejected. If certain words (such as colors) should not be used as tags, indicate this as well.

Use negative examples sparingly and make them very clear. Using one negative example to prevent a common mistake is helpful. Using lots of negative examples can confuse the Workers.

- **Be open about the approval process**

If you are using an automated review process (for instance asking 3 Workers to do the same HIT and then comparing results to determine which to approve or reject) inform Workers of this.

Also if granting bonuses indicate how you decide who will be paid a bonus.

- **Be specific about the tools or methods you want Workers to use**

If you want Workers to use a specific method, tell them. Do you want them to call to verify an address instead of using the internet?

Do you want them to use a search engine other than Google? You can make Workers more efficient by including the link to the search engine with the search preloaded.

Will you be using a specific website to verify the content they provide is not plagiarized? If so tell them what tool and what percent of their content needs to be unique so they can test it for themselves before submitting it.

💡**Tip**

Before publishing your HITs, give your instructions, without explanation, to a colleague and have them do some of your HITs "on paper" according to these instructions. See if they get confused or are uncertain how to perform your HITs.

Set your Price

How you pay Workers will determine who will work for you and how many HITs they will do.

- **Pay Fairly**

 Look at HITs similar to yours to see what the "going rate" is for HITs on Mechanical Turk, afterall it is a marketplace and Workers will be looking at how other Requesters are paying when they decide whether to do your HITs. Don't expect a Worker to complete a 1 hour video transcription for $0.07 when there is a 5 minute audio transcription available at $0.05.

When you test your HITs, time how long it takes you to complete a HIT. How many could you do in an hourly? Use this to determine how to price an individual HIT.

- **Pay the same amount of money for the same amount of work**

All HITs within a group of HITs (aka a batch or HIT Type) that pay the same should require approximately the same amount of work.

 If some HITs in a batch pay $0.01 for one question, but others pay $0.01 for 4 questions (as in the example below), Workers may 'cherry-pick' and only do the HITs that require one answer since they pay better.

```
Does this picture include a car?
  ○ YES   ○ NO

If yes, please enter the make, model and year of the car below.
Make  [                    ]
Model [                    ]
Year  [                    ]
```

To fix this you could create two separate HITs: one that asks if there is a car in the picture and a second HIT for pictures that include cars in order to collect the additional information. Or you could give a bonus to Workers when they correctly answer the additional 3 questions.

Test Your HITs

Testing your HITs helps you discover technical errors and confusing instructions. Before publishing 10,000 new HITs, consider publishing 10-100 and ask Workers for feedback on them. This gives you an opportunity to revise your HIT based on feedback from Workers.   Often, Workers have great insight into ways to improve your HITs. Providing a comment box like the following on each HIT lets Workers provide thoughts on each task they complete.

Test your HITs in all major browsers (Internet Explorer and Firefox) to make sure your HIT has the same functionality. If you have links to pictures or videos in your HITs, make sure the links work. The best way to check the links is to actually complete some of your HITs using the Amazon Mechanical Turk sandbox. Part of the test should include verifying the format of the answers is acceptable.

Tip

If you would like to publish HITs but not have Workers work on them, use the sandbox (https://requestersandbox.mturk.com). You'll need to create a Requester account on this sandbox website. To view your HITs on the sandbox site as a Worker, create a Worker account on https://workersandbox.mturk.com and search for your HITs. There is no charge for using the Mechanical Turk sandbox.

Designate an Owner

Many successful Requesters designate a team member to be an Amazon Mechanical Turk administrator. The administrator interacts with the Worker community, receives feedback about the HITs, responds to questions, verifies accuracy, manages Worker qualifications, designs new HITs, and organizes the results.

Reach out to the Worker Community

Many Workers communicate with each other through online message boards and forums, for example, www.turkernation.com. Consider introducing yourself to Workers on these forums.

Make Workers More Efficient

A well designed HIT that is efficient for a Worker to do can pay the same reward but yield a higher effective hourly rate than a poorly designed HIT. Here are some Best Practices for designing for Worker Efficiency:

- **Minimize Keystrokes and Scrolling**

    Scrolling reduces Worker efficiency and as a result reduces their effective hourly pay rate. Find a way to make your HIT fit on one page. Make long instructions collapsible and keep them collapsed by default.

- **Open web pages in a new window**

    If a Worker must open a new window to complete your HIT, open it in a new browser window so that the Worker can continue working on the HIT in the current window while referring to the data in the new window.

    💡**Tip**

    To make a hyperlink open in a new window, include the target attribute in the HTML of your HIT. For example: `<a href="http://www.mturk.com"` `"target=_blank">Amazon Mechanical Turk</a>`

- **Don't ask redundant questions**

    The following questions are redundant.

    To remove the redundancy, eliminate the first set of radio buttons and include a "No car in the picture" option.

- **Position answer fields strategically**

    Place the answer field close to the content so that the Worker doesn't have to scroll between the content and the answer fields. If the Worker has a number of fields to enter, for instance when transcribing an address from an image, you might put the image at the top and bottom of the HIT so it is viewable regardless of which field a Worker is working on.

Managing Worker Accuracy

As a Requester using Amazon Mechanical Turk, you only pay when satisfied with the work results. You have the ability to review work prior to approving or rejecting work submissions. There are also a number of other tools that allow help you manage work accuracy. This section discusses how to optimize your use of these tools to get the most accurate results.

Rejecting Work

Workers know that approval rates are used by many Requesters to determine which Workers can do their work. The best Workers are very cognizant of their approval rate and want to preserve it.

If a Worker submits a wrong or unacceptable answer (for instance an answer that does not follow your instructions) you should reject the work. If you do not reject work, you should expect that Workers will continue to make the same mistakes. It is better to reject the work and provide feedback to the Worker on what they are doing wrong. This will train a Worker who is making a specific mistake.

When work is rejected, the Worker isn't paid so Workers have an incentive to work on HITs that will be accepted. Not all Workers will be good at all HITs. Rejecting work gives Workers the message that maybe they should focus their energy on HITs that they can complete successfully.

Adjudicating fairly and quickly will build your Requester reputation.

Blocking Workers

If a Worker continually submits unacceptable answers you should block the Worker. Blocking a Worker will prevent that Worker from completing any more HITs for you. This can be important when you find a particularly inaccurate Worker, or a Worker who is not responding to feedback, training, or other communication.

💡 Tip

> When you block a Worker who is a poor performer, you are helping all Requesters by providing valuable feedback to Mechanical Turk about the Worker. A Worker who receives multiple blocks from different Requesters will be suspended from working on Mechanical Turk.

Qualifications

When you create a HIT in Mechanical Turk, you can require Workers meet certain criteria or have certain Qualifications in order to work on your HITs. Mechanical Turk provides some basic Qualifications for instance a Worker's Approval Rate across all work on the system. Mechanical Turk also allows you to create your own Qualifications. You can create a Qualification via the Command Line Tools or the API. Qualifications are an important tool in that they allow you to direct your work to a specific group of Workers.

Tip

> Keep in mind that requiring Workers have numerous Qualifications to complete your HITs will limit the number of Workers able to do your HITs.

Plurality

When you create HITs in the system, you can indicate how many Workers you want to complete each HIT. This concept is called plurality. For HITs with a limited set of possible answers (for instance Yes/No), plurality can be a power tool to assess accuracy. If multiple Workers provide the same answer, you can be more confident in the answer than if you'd only asked one Worker.

Gold Standards

Gold Standard Data is used in many industries to assess the validity of test results. Creating HITs that you already know the answers (aka Gold Standard HITs) is any easy way to assess the accuracy of a Worker.

You could create HITs that you know the answers to in Mechanical Turk and compare the results to your known answers to identify Workers that accurately complete your HITs. You could then assign your Qualification to these Workers and route future work to them.

You could also mix Gold Standard HITs into all of your HITs on an on-going basis. You could use this to measure Worker accuracy over time and determine if a Worker's accuracy is consistent or if the Worker's Qualification should be revoked.

A Strategy for Accuracy

The tools we've presented are just the building blocks for accuracy. Combining these tools together will create a system that can deliver the accuracy you need.

As an example, if your HIT has a limited set of possible answers, you might use this approach:

1. Create Gold Standards HITs to identify accurate Workers
2. Create a Qualification Type and assign the accurate Workers from the Qualification created in Step 1
3. Create HITs in the system
   a. Require Workers to have your Qualification to complete the HITs
   b. Ask multiple Workers to complete each HIT
   c. Include a sampling of Gold Standard HITs
4. Adjudicate Results
   a. If all Workers agree, approve the Assignments
   b. If Workers disagree:
      i. Approve based on majority/consensus, or
      ii. Extend the HIT to ask another Worker
5. Use Gold Standards to adjust Qualification score for a Worker

If your HIT is more 'free form', such as `Write an abstract of this article` or `Create tags for this image`, you might combine the accuracy building blocks something like this:

1. Create HITs and manually review to identify accurate Workers
2. Create a Qualification Type and assign the Qualification to the accurate Workers identified in step 1
3. Create HITs in the system that require Workers to have your Qualification
4. Manually audit the HITs or create another Qualification Type for Workers you feel can act as 'Reviewers'
    a. Ask Reviewer Workers to rate the answer provided in the HITs from Step 3
    b. Based on the Reviewer's decision, approve or reject the HITs