How-to create a P.808 HIT

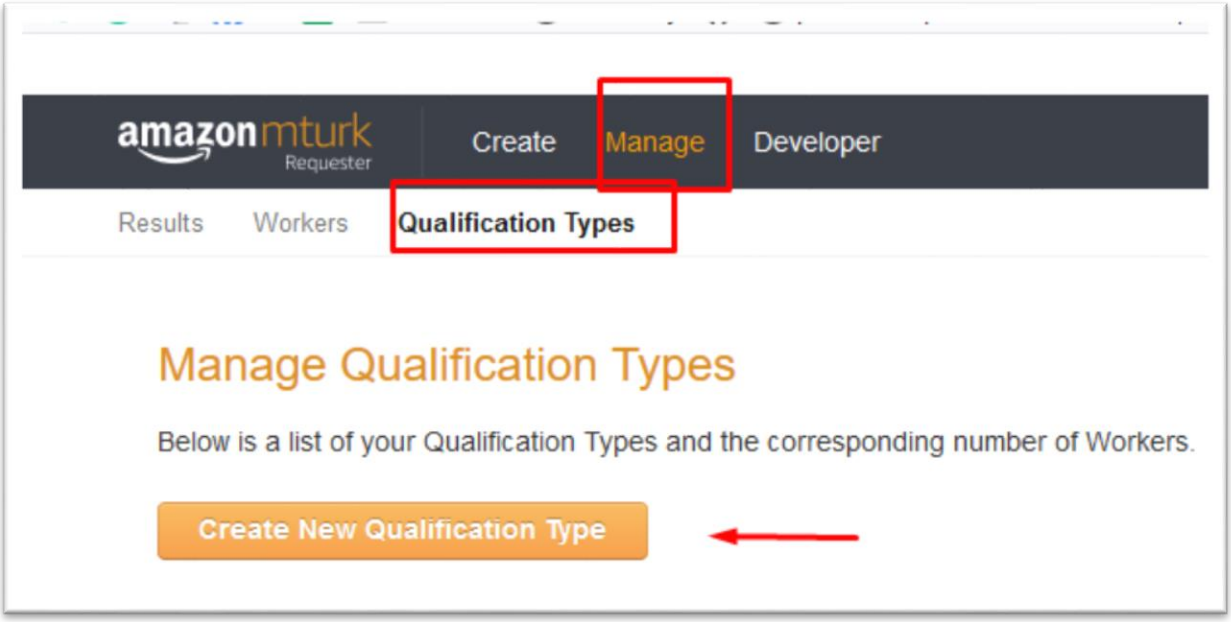
Document scope:
provide a brief set of instructions how to go from Batch_3824355_batch_results_output.csv to publishing the batch

Step 1: Assign custom qualification to the accepted workers

Step.1.1. Create custom qualification (Note: one-time step, if qualification exist, you do not need to do this step).

From requester account:

- Go to **"Manage" > "Qualification Types" > "Create New Qualification Type"**



Manage Qualification Types

Below is a list of your Qualification Types and the corresponding number of Workers.

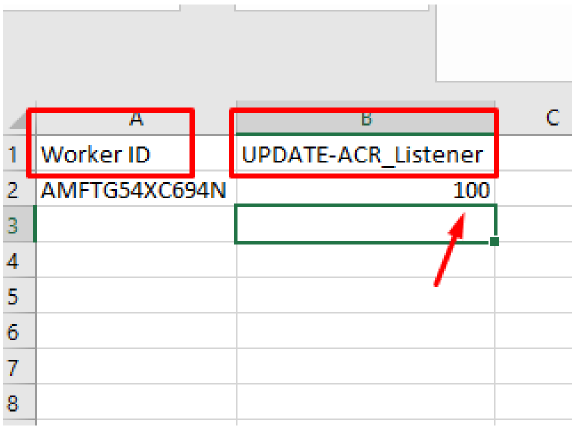
- Add a name: "ACR_Listener" and description: "XYZ" (Note: workers see descriptions, so write something like "You are qualified to perform Listening Only Test -ACR"). And click OK.
- After a while (30sec), it will show up in the list
- You have created the qualification type.

Step1.2. Assign the existing qualification type to accepted workers

From requester account:

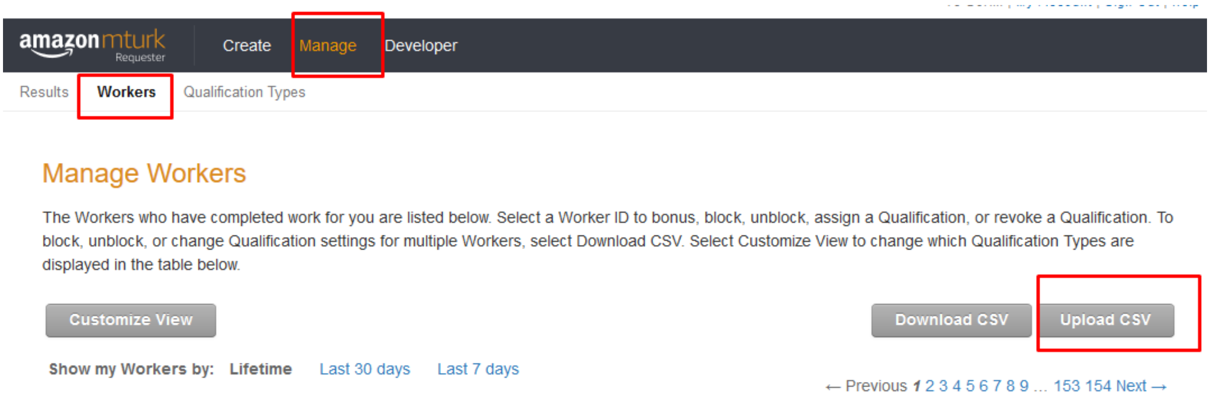
- Open the "Batch_3824355_batch_results_output.csv" file and edit it to have following fields: (change WorkerId → Worker ID, Add a column "UPDATE-ACR_Listener", and add value 100[1] for all workers in that column). Save the file.

[1] Any other number is also working, just use the same number in step3 - Worker requirements



	A	B	C
1	Worker ID	UPDATE-ACR_Listener	
2	AMFTG54XC694N	100	
3			
4			
5			
6			
7			
8			

- Go to **“Manage” > “Workers” > “Upload CSV”**

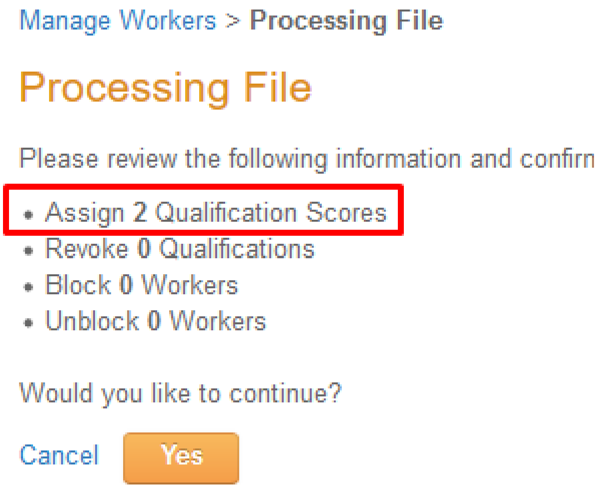


Manage Workers

The Workers who have completed work for you are listed below. Select a Worker ID to bonus, block, unblock, assign a Qualification, or revoke a Qualification. To block, unblock, or change Qualification settings for multiple Workers, select Download CSV. Select Customize View to change which Qualification Types are displayed in the table below.

Customize View
Download CSV
Upload CSV
Show my Workers by: Lifetime Last 30 days Last 7 days
← Previous 1 2 3 4 5 6 7 8 9 ... 153 154 Next →

- Use the “Browse” button and upload the new “Batch_3824355_batch_results_output.csv”
- Click on “Yes” to assign the qualification.



Manage Workers > **Processing File**

Processing File

Please review the following information and confirn

- Assign **2** Qualification Scores
- Revoke **0** Qualifications
- Block **0** Workers
- Unblock **0** Workers

Would you like to continue?

Step2: Prepare prerequisites

In this step you create and/or upload clips and other assets required for creating the HIT. All assets should be uploaded in a server and made publicly available. It is recommended to use any CDN service as well to make sure data are reachable for your participants.

Step 2.1. Create trapping stimuli

Use script “create_trapping_stimuli” from hitapp_p808/Scripts:

- **Edit “**hitapp_p808/Scripts/cfgs_and_inputs/trapping.cfg” if needed
    - **“input_directory= trapping”** it means all information will be find in a “trapping” directory. The “trapping” directory should contain three sub-directories: 1. messages, 2. source, 3. output. The directory path is relative to the script’s path. As a result, the script will look for following structure:

```
.
+-- create_trapping_stimuli.py
+-- trapping
|   +--messages
|      +--ACR_Bad_short.wav
|      +--ACR_Poor_short.wav
|      +--ACR_Fair_short.wav
|      +--ACR_Good_short.wav
|      +--ACR_Excellent_short.wav
|   +--source
|      +-- clip1.wav
|      +-- clip2.wav
|      +-- ….
|   +--output
```

- Add some clips from the dataset understudy in to ‘trapping/source’ folder. Make sure they include samples from every speaker, and different quality levels.
- Make ‘trapping/output’ directory empty: the trapping dataset will be created here
- **Run the script:**
    - **First check if requirements are installed:**
      ```
      pip install -r create_trapping_stimuli_requirements.txt
      ```
    - **Run the script**
      ```
      python create_trapping_stimuli.py --cfg
      cfgs_and_inputs/trapping.cfg
      ```

- Check the trapping/output directory: 5 clips per each source clip should be created. In addition, you can find list of clips and their correct answers in “output_report.csv”.
-

Step 2.2. Upload all resources

Create a csv file “row_input.csv” and fill it: use a template given in ‘hitapp_p808\P808Template\test input\row_input_template.csv’

- trapping dataset
    - upload the trapping dataset into your server.
    - Insert their URLs into the column “trapping_clips“ of “row_input.csv”
    - Insert their corresponding correct answer into column “trapping_ans” of “row_input.csv” (Note: you can find the correct answers in output_report.csv which was generated by “create_trapping_stimuli” script)

- Rating clips:
  - Upload your complete dataset into your server
  - Insert their URLs into column “rating_clips” of “row_input.csv”
- Gold standard clips:
  - Select and upload your gold-standard clips into your server
  - Insert their URLs into column “gold_clips” of “row_input.csv”
- Math questions:
  - Upload all clips in ‘\P808Template\assets\clips\math’ into your server
  - Insert their URLs into column “math” of “row_input.csv”
- Environment Test (setup section /pair comparison)
  - Upload all clips in ‘\P808Template\assets\clips\sample_jnd’ into your server
  - Insert URLs of files ‘50*.wav’ in ‘pair_a’ and ‘42*.wav’ into column “pair_b” of “row_input.csv”
  - Note: each row should match i.e. belonging to same speaker e.g. ‘42S_female1.wav’ and ‘50S_female1.wav’
- Training
  - Upload your training clips into your server
  - Insert their URLs into the ACR.html file (var config[‘trainingUrls’]).
  - In case you want to have a trapping question in the training:
    - Insert its URL also into the config[‘knownQuestionInTrainingUrl’]
    - Insert its correct answer into the config[‘knownQuestionInTrainingAns’]
- Other resources
  - Following resources should be also uploaded into a server and their URL should be changed in the ACR.html
  - Volume setting: “hitapp_p808\P808Template\assets\clips\signal_level.wav”
  - Image in Instruction: “hitapp_p808\P808Template\assets\img\process_2.png”
  - Image in setup: “hitapp_p808\P808Template\assets\img\ attention.pn”

Step 3: create input.csv

- **Use script “create_input_acr” from hitapp_p808.**

The script needs two input files:

- A configuration file. Example is given in “Scripts\cfgs_and_inputs\create_input.cfg”
- row_input.csv which was create in Step2.

The row_input file should contains following columns:

- 'rating_clips': urls of all clips which needs to be rated
- 'math': url of various math questions to proof usage of two-eared headphones.
- 'pair_a','pair_b': pairs will appear in the setup section, to check the environment of user.
- 'trapping_clips','trapping_ans': url to all trapping questions, and a number which shows the correct answer.
- 'gold_clips': (optional) list of gild clips

- **Run the script:**
  - **First check if requirements are installed:**

```
pip install -r create_input_acr_requirements.txt
```

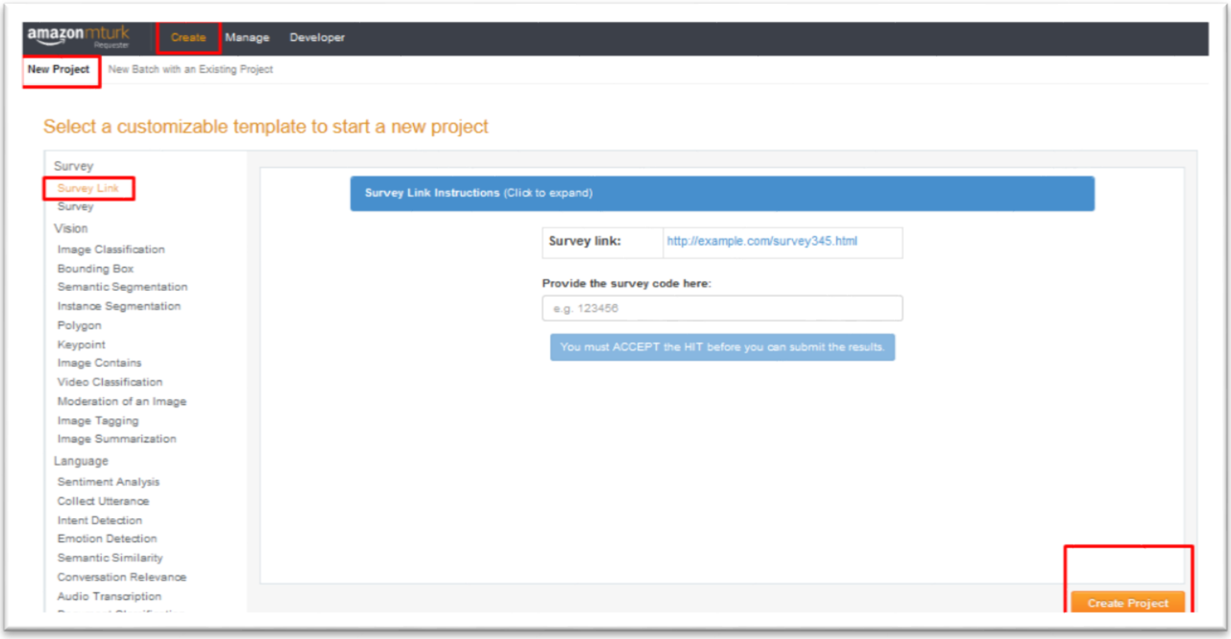
- **Run the script**

```
          python create_input_acr.py --cfg cfgs_and_inputs/create_input.cfg
--row_input cfgs_and_inputs/ row_input_librivox.csv
```

The script generates an input csv file: xxx_publish_batch.csv

Step4: Create the ACR Project

- **Go to “Create” > “New Project” > “Survey Link” > “Create project”**

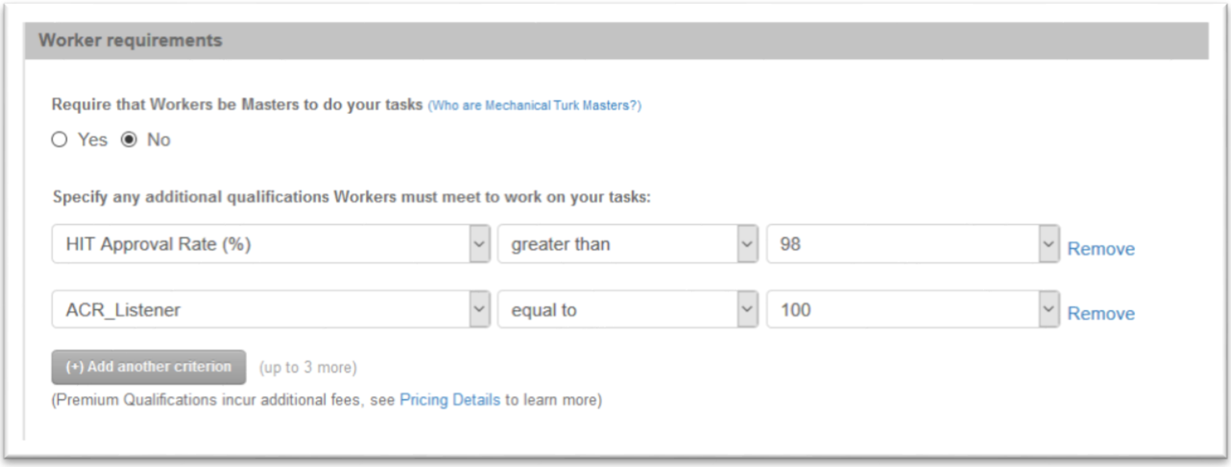


Select a customizable template to start a new project

Survey link: http://example.com/survey345.html

Provide the survey code here: e.g. 123456

- **Fill information in “1 – Enter Properties”,** important ones:
  - **“Setting up your survey”**
    - **Payment**
    - “Number of respondents”**: 9**
    - “Time allotted per Worker”**: 1 Hours**
  - **“Worker requirements”**
    - **Use following qualifications:**



Worker requirements

Require that Workers be Masters to do your tasks (Who are Mechanical Turk Masters?)

○ Yes ● No

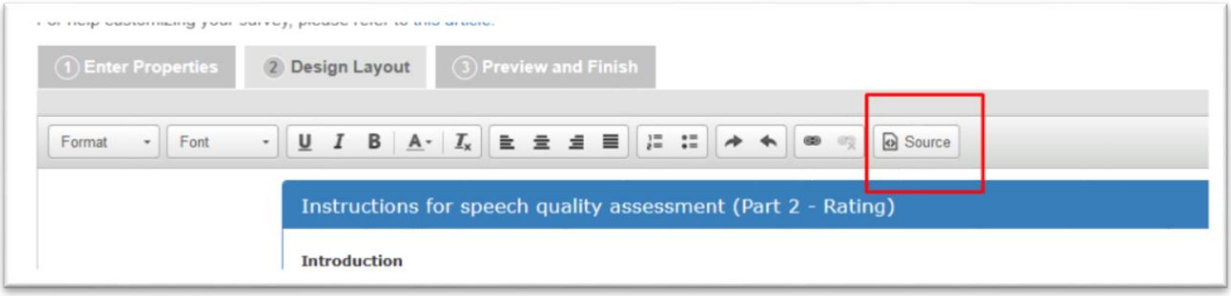
Specify any additional qualifications Workers must meet to work on your tasks:

HIT Approval Rate (%)	greater than	98	Remove
ACR_Listener	equal to	100	Remove

(up to 3 more)

(Premium Qualifications incur additional fees, see Pricing Details to learn more)

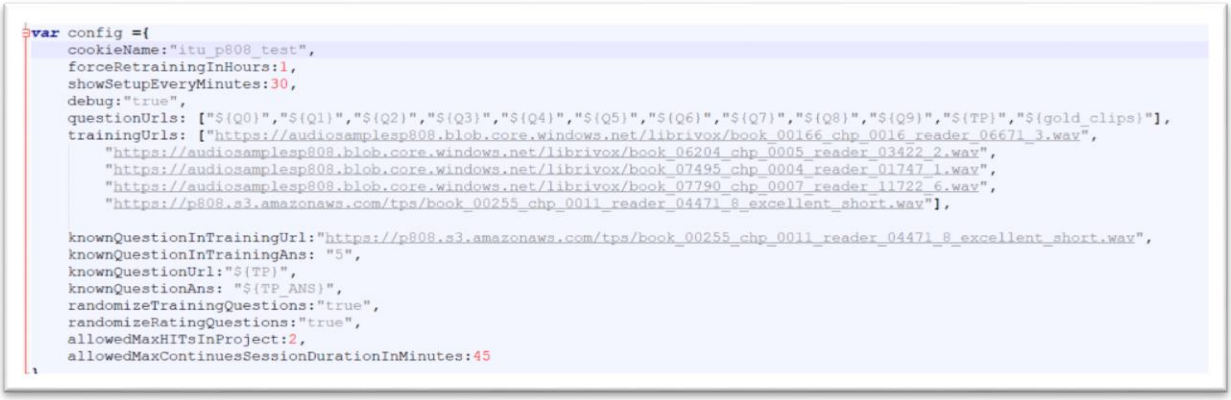
- Save and Go to “Design Layout”
- Click on “Source”



Instructions for speech quality assessment (Part 2 - Rating)

Introduction

- Copy-paste the ACR.html here
    - NOTE: you should consider to change some information in the “Introduction” section: like “**Payment”, or the image on top (up to 60x Ratings) …**
    - Note: you may consider to change the config object as well- variables should match with input.csv



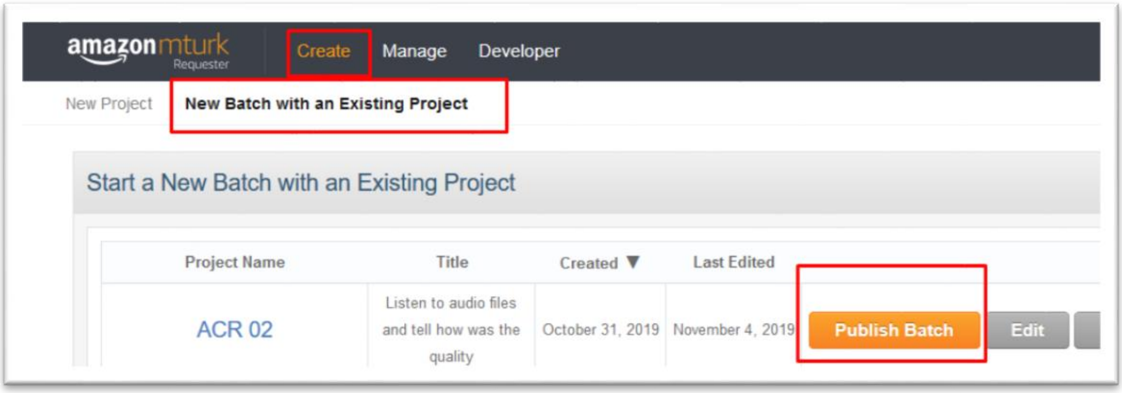
```
var config ={
    cookieName:"itu_p808_test",
    forceRetrainingInHours:1,
    showSetupEveryMinutes:30,
    debug:"true",
    questionUrls: ["${Q0}","${Q1}","${Q2}","${Q3}","${Q4}","${Q5}","${Q6}","${Q7}","${Q8}","${Q9}","${TP}","${gold_clips}"],
    trainingUrls: ["https://audiosamplesp808.blob.core.windows.net/librivox/book_00166_chp_0016_reader_06671_3.wav",
        "https://audiosamplesp808.blob.core.windows.net/librivox/book_06204_chp_0005_reader_03422_2.wav",
        "https://audiosamplesp808.blob.core.windows.net/librivox/book_07495_chp_0004_reader_01747_1.wav",
        "https://audiosamplesp808.blob.core.windows.net/librivox/book_07790_chp_0007_reader_11722_6.wav",
        "https://p808.s3.amazonaws.com/tps/book_00255_chp_0011_reader_04471_8_excellent_short.wav"],

    knownQuestionInTrainingUrl:"https://p808.s3.amazonaws.com/tps/book_00255_chp_0011_reader_04471_8_excellent_short.wav",
    knownQuestionInTrainingAns: "5",
    knownQuestionUrl:"${TP}",
    knownQuestionAns: "${TP_ANS}",
    randomizeTrainingQuestions:"true",
    randomizeRatingQuestions:"true",
    allowedMaxHITsInProject:2,
    allowedMaxContinuesSessionDurationInMinutes:45
}
```

- Click on “source”
- Click on “Save” and “Preview”
- Click on “Finish”

Step 5: Publish the batch

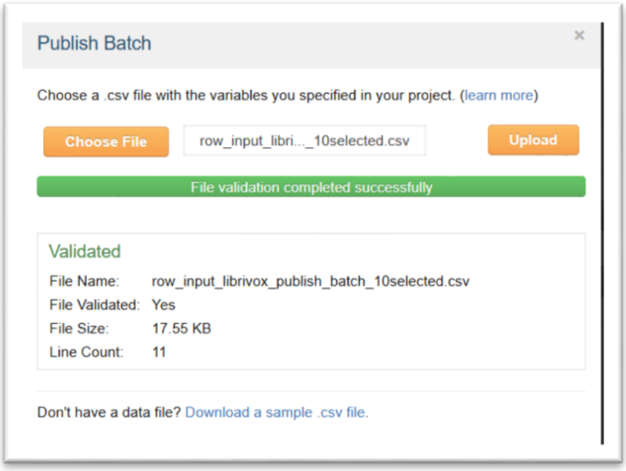
- **Go to “Create”> “New Batch with an Existing Project”**



Start a New Batch with an Existing Project

Project Name	Title	Created ▼	Last Edited
ACR 02	Listen to audio files and tell how was the quality	October 31, 2019	November 4, 2019

- Find your project, and click on **“Publish Batch”**
- Upload your input.csv file created in last section (xxx_publish_batch.csv)



Publish Batch

Choose a .csv file with the variables you specified in your project. (learn more)

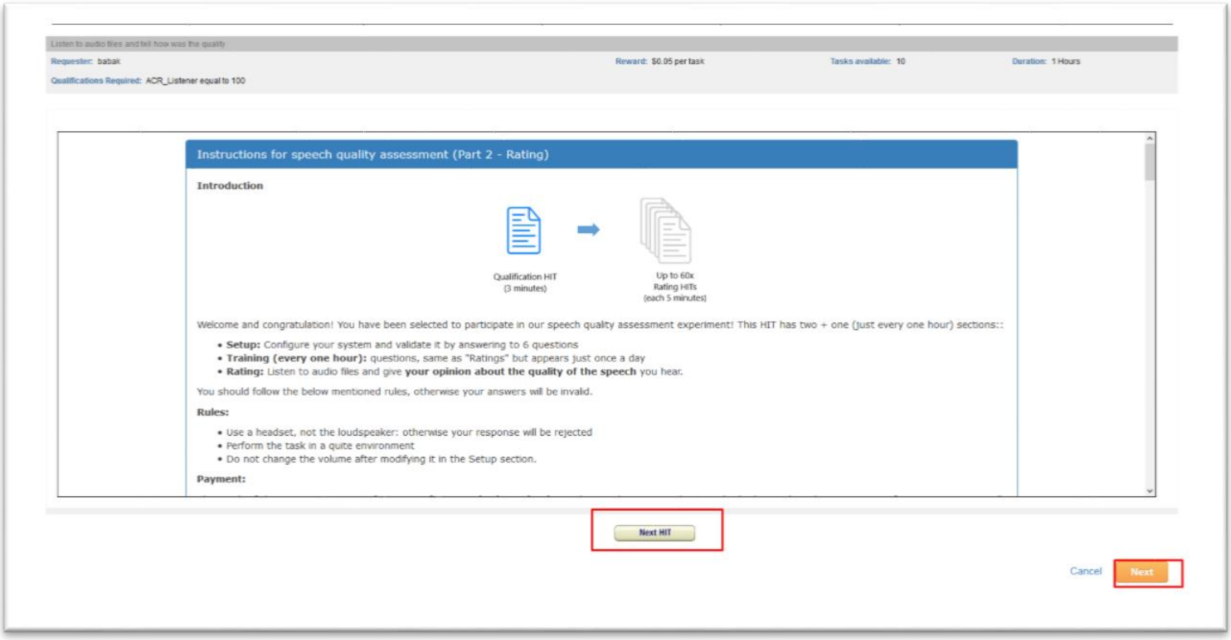
File validation completed successfully

Validated

File Name:	row_input_librivox_publish_batch_10selected.csv
File Validated:	Yes
File Size:	17.55 KB
Line Count:	11

Don't have a data file? Download a sample .csv file.

- If everything is green, go ahead with “upload” otherwise there is variable missing in the input.csv.
- Check the HITs, and if everything is ok click on Next



Listen to audio files and tell how was the quality

Requester: babak   Reward: $0.05 per task   Tasks available: 10   Duration: 1 Hours

Qualifications Required: ACR_Listener equal to 100

Instructions for speech quality assessment (Part 2 - Rating)

Introduction

Welcome and congratulation! You have been selected to participate in our speech quality assessment experiment! This HIT has two + one (just every one hour) sections::

- **Setup:** Configure your system and validate it by answering to 6 questions
- **Training (every one hour):** questions, same as "Ratings" but appears just once a day
- **Rating:** Listen to audio files and give **your opinion about the quality of the speech** you hear.

You should follow the below mentioned rules, otherwise your answers will be invalid.

Rules:

- Use a headset, not the loudspeaker: otherwise your response will be rejected
- Perform the task in a quite environment
- Do not change the volume after modifying it in the Setup section.

Payment:

- Check the calculation and “Publish”
- You may send emails to workers to inform them about availability of this project (see next page)

Sending Emails to Workers:

- Edit the configuration file 'Scripts\mturk.cfg'
  - Add 'aws_access_key_id', and 'aws_secret_access_key' of your requester account (if you do not have them, follow Step1 and Step2 here: https://requester.mturk.com/developer)
  - Make sure the 'endpoint_url' of production is active and the one for sandbox is commented
  - Add your "requester_client_id" (login as a worker to see it)
  - Edit the "subject", and "message" (note: the url in the message will refer to a page containing all HITs created by your requester_client account)
  - Add a comma separated list of "worker_ids"
- **Run the script:**
  - **First check if requirements are installed:**

```
pip install -r mturk_utilis_requirements.txt
```

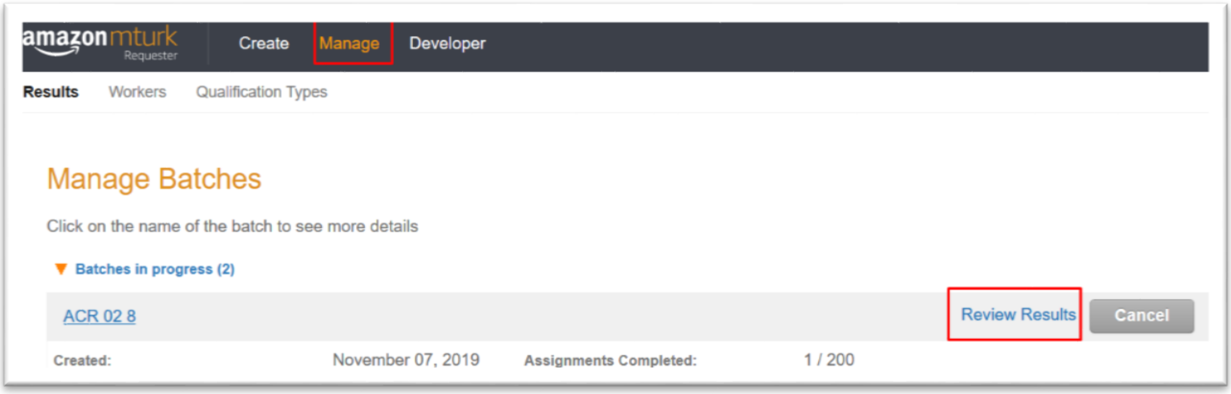
  - **Run the script**

```
python mturk_utils.py --cfg mturk.cfg send_emails
```

Evaluate the Results:

Use the `acr_result_parser` script to evaluate the results you get from ACR project.
Download the answers from MTurk (hereafter answer.csv):

- Go to "Manage" >> find the Batch from "Batches in progress" >> click on "Review Results"



Manage Batches

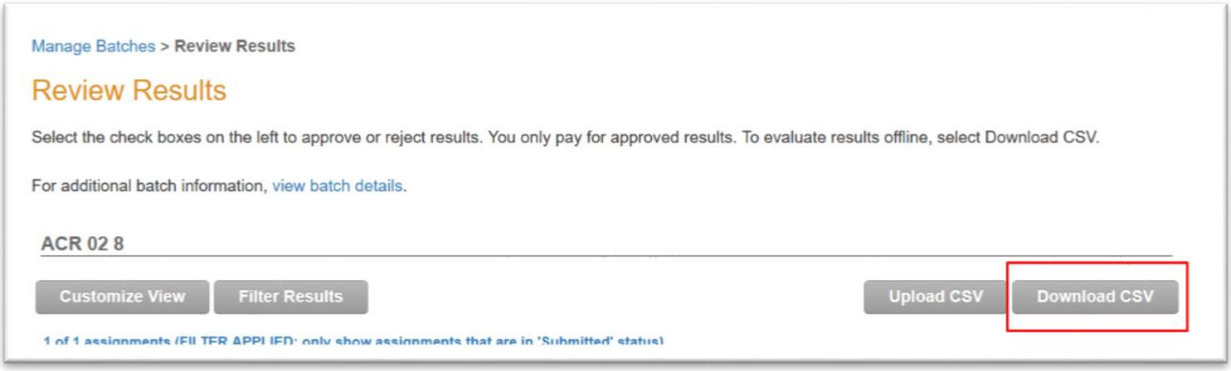
Click on the name of the batch to see more details

▼ Batches in progress (2)

ACR 02 8

Created: November 07, 2019 Assignments Completed: 1 / 200

- Click on "Download CSV".



Manage Batches > Review Results

Review Results

Select the check boxes on the left to approve or reject results. You only pay for approved results. To evaluate results offline, select Download CSV.

For additional batch information, view batch details.

ACR 02 8

1 of 1 assignments (FILTER APPLIED: only show assignments that are in 'Submitted' status)

- **Run the script:**
    - **First check if requirements are installed:**

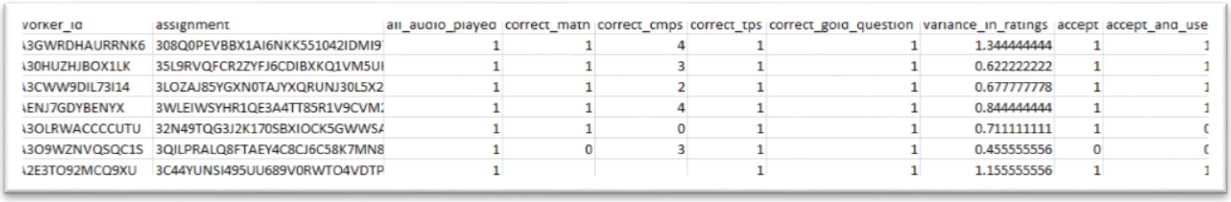
      ```
      pip install -r acr_result_parser_requirments.txt
      ```

    - **Run the script** (where the [ANSWER_CSV] is file you downloaded)

      ```
      python acr_result_parser.py --answers [ANSWERS_CSV]
      ```

    - NOTE: you can change the configuration which the script works based on including is working the correct answers, question_names, and acceptance criteria. Just edit the json object in the `acr_result_parser.py`

- The script will create following output files:
    - [ANSWER_CSV]_data_cleaning_report.csv: check the "accepted" column, 1 means accepted.



vorker_id	assignment	all_audio_played	correct_math	correct_cmps	correct_tps	correct_gold_question	variance_in_ratings	accept	accept_and_use
3GWRDHAURRNK6	308Q0PEVBBX1AI6NKK551042IDMI9	1	1	4	1	1	1.344444444	1	1
30HUZHJBOX1LK	35L9RVQFCR2ZYFJ6CDIBXKQ1VM5UI	1	1	3	1	1	0.622222222	1	1
3CWW9DIL73I14	3LOZAJ85YGXN0TAJYXQRUNJ30L5X2	1	1	2	1	1	0.677777778	1	1
ENJ7GDYBENYX	3WLEIWSYHR1QE3A4TT85R1V9CVM	1	1	4	1	1	0.844444444	1	1
3OLRWACCCCUTU	32N49TQG3J2K170SBXIOCK5GWWS	1	1	0	1	1	0.711111111	1	0
3O9WZNVQSQC1S	3QILPRALQ8FTAEY4C8CJ6C58K7MN8	1	0	3	1	1	0.455555556	0	0
2E3TO92MCQ9XU	3C44YUNSI495UU689V0RWTO4VDTP	1			1	1	1.155555556	1	1

- all_audio_played: 1 if all clips for questions are played until the end.
- correct_math: 1 if the math question answered correctly, otherwise 0 or empty (when the setup was not shown)
- correct_cmp: number of pair comparisons that are answered correctly (max:4, empty when the setup was not shown)
- correct_tps: 1 if the answer to the trapping question in ACR part is correct
- correct_gold_question: 1 if the answer to the gold_standard question is correct
- variance_in_rating: variance between ratings given in the session. 0 means the user always gave a same vote
- accept: 1 if the answer should be accepted
- accept_and_use: 1 if the session was enough reliable to be used in further analyzes.

    - [ANSWER_CSV]_bonus_report.csv: list of workers and amount of bonuses which should be paid (see Pay Bonus)

    - [ANSWER_CSV]_votes_per_clip.csv: one row per file, including votes, MOS, STD, and N of votes and 95%CI
    - [ANSWER_CSV]_rejection.csv: List of assignment which should be rejected.
    - [ANSWER_CSV]_accept.csv: List of assignment which should be approved.
    - [ANSWER_CSV]_accept_reject_gui.csv: List of assignments which should be accepted and rejected with proper message, formatted to be used with "upload csv" in the website.

Pay Bonus

Using the `mturk_utils` script and the outcome of `acr_result_parser`, you can assign bonuses to the workers.

- Edit the configuration file 'Scripts\mturk.cfg'
    - Add 'aws_access_key_id', and 'aws_secret_access_key' of your requester account (if you do not have them, follow Step1 and Step2 here: https://requester.mturk.com/developer)
    - Make sure the 'endpoint_url' of production is active and the one for sandbox is commented

- **Run the script:**
  - **First check if requirements are installed:**

    ```
    pip install -r mturk_utilis_requirements.txt
    ```

  - **Run the script**

    ```
    python mturk_utils.py --cfg [CONFIG_FILE] --send_bonus
    cfgs_and_inputs/bonus.csv
    ```

    - You can use the bonus report generated by the `acr_result_parser` script

Approve/Reject answer

Using the `mturk_utils` script and the outcome of `acr_result_parser`, you can accept/reject assignment.

- Edit the configuration file 'Scripts\mturk.cfg'
  - Add 'aws_access_key_id', and 'aws_secret_access_key' of your requester account (if you do not have them, follow Step1 and Step2 here: https://requester.mturk.com/developer)
  - Make sure the 'endpoint_url' of production is active and the one for sandbox is commented
- **Run the script:**
  - **First check if requirements are installed:**

    ```
    pip install -r mturk_utilis_requirements.txt
    ```

  - **Run the script**

    ```
    Approve: python mturk_utils.py --cfg [CONFIG_FILE] --approve
    cfgs_and_inputs/approve.csv
    ```

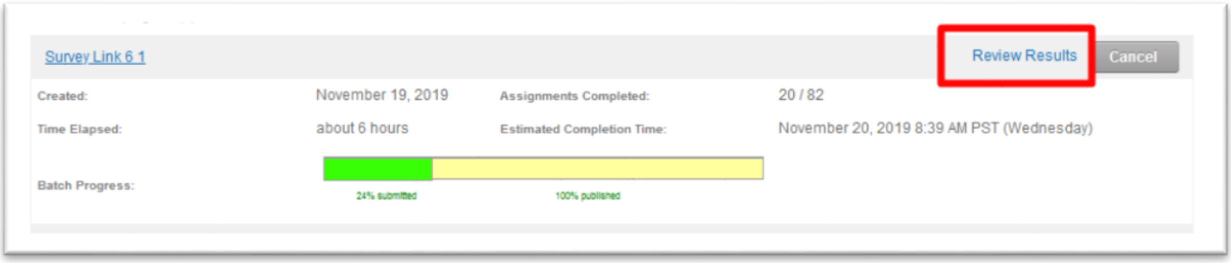
    →You can use the *_accept.csv file generated by the `acr_result_parser` script

    ```
    Reject: python mturk_utils.py --cfg [CONFIG_FILE] -- reject
    cfgs_and_inputs/reject.csv
    ```

    →You can use the *_rejection.csv file generated by the `acr_result_parser` script

Approve/Reject Answers using GUI

- Go to "Manage"> "Results"> from your Batch select "Review Results"

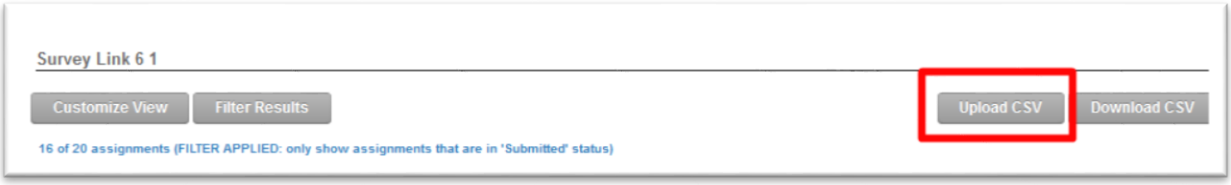


Created: November 19, 2019 Assignments Completed: 20 / 82

Time Elapsed: about 6 hours Estimated Completion Time: November 20, 2019 8:39 AM PST (Wednesday)

Batch Progress: 24% submitted 100% published

- Click on "Upload CSV"



Survey Link 6 1

16 of 20 assignments (FILTER APPLIED: only show assignments that are in 'Submitted' status)

- Use “Browse” button and upload “[ANSWER_CSV]_accept_reject_gui.csv” (outcome of `acr_result_parser)`