

# Crowd Sourcing Training Data For Machine Learning

with Amazon Mechanical Turk

A dark blue diagonal gradient bar that starts from the bottom left corner and extends towards the top right corner, covering the lower half of the slide.

# Prerequisites

- Python 2.7 or 3.5
- Amazon AWS account
- MTurk Requester Account
- AWS Command Line Tool
- IAM User Profile with
  - AmazonMechanicalTurkCrowdFullAccess
  - AmazonMechanicalTurkFullAccess

# Natural Language Processing APIs Available

- sentiment-analysis
- emotion-detection
- named-entity-recognition
- coreference-resolution
- key-phrase-extraction
- semantic-similarity
- collect-utterance-for-intent
- intent-detection
- text-categorization

# Computer Vision APIs Available

Bounding Box Instructions (Click to collapse)

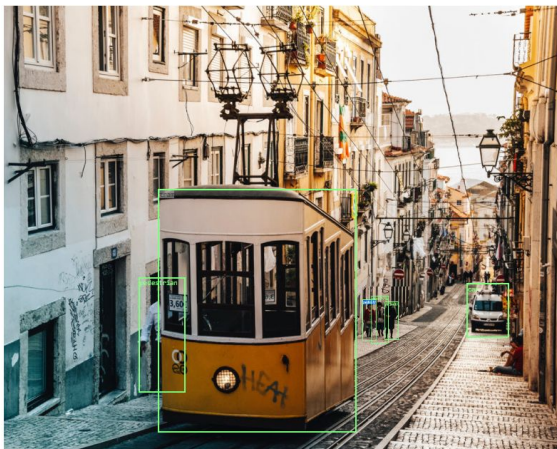


- Draw a rectangle using your mouse over each instance of an object, then choose the label.
- Make sure the box does not cut into the object, leave a 2 - 3 pixel margin
- When objects are overlapping, draw a box around each object, include all known parts of the object in the box, even if they are covered by another object
- Avoid shadows, they're not considered as a part of the object
- If there are no car, pedestrian, tram, select the "None" option

Draw boxes around the following objects: car, pedestrian, tram

If you can't find the specified objects, select this option:

None



Reset This is preview

- bounding-box
- image-contains
- image-categorization
- image-similarity

# Install Python client and Boto 3

```
pip install --upgrade mturk-crowd-beta-client  
--ignore-installed six
```

# Build Python code

```
from mturk_crowd_beta_client import MTurkCrowdClient
from boto3.session import Session
session = Session(profile_name='mturk-crowd-caller')
crowd_client = MTurkCrowdClient(session)

api_to_use = 'sentiment-analysis'
task_name = 'pycascades2018'
text = {'text': 'Off to a fantastic start at @pycascades.'}
put_result = crowd_client.put_task(api_to_use, task_name, text)
get_result = crowd_client.get_task(api_to_use, task_name)
```

# Jupyter Notebook – github/aws-samples/mturk-jupyter

## sentiment-analysis

```
Input: { "text": "Everything is wonderful!" }
```

```
Result: {'sentiment': 'positive'}
```

Max length of the input text is 400 characters. Sentiment is one of positive, negative, neutral or cannot determine.

When you create a Task using the sentiment-analysis API, you're automatically creating a Human Intelligence Task (HIT) on worker.mturk.com. Here's an example of a sentiment analysis HIT.

**Sentiment Analysis Instructions** (Click to collapse)

Basic [Details & Examples](#)

Please select the sentiment of the text between positive, negative, and neutral. Only consider the attitude of the author, not your feelings about the contents of the text. If you cannot make a determination, whether because the language is wrong, the text is gibberish, or some other reason, please select "Cannot Determine".

I read many reviews regarding the brown ones and their lack of quality, however, I have them in black and grey and LOVE them. Well, I should have heeded the advice given on the brown. I got them for Christmas 2016, so maybe 9 weeks ago and they are falling apart. I think I've worn them a total of 10 times.

**Sentiment expressed by the content:**

Positive
Neutral
Negative
Cannot Determine

For more information, refer to the [sentiment-analysis API documentation](#).

# Jupyter Notebook – github/aws-samples/mturk-jupyter

## Create a Task

```
In [ ]: #set the function_name to the name of the API
function_name = 'sentiment-analysis'
```

```
In [ ]: # automatically generate a random task ID
task_name = 'my-test-task-' + uuid.uuid4().hex
print(task_name)
```

```
In [ ]: # define the text that you want analyzed, up to 400 characters
text = 'The trip by @VP Pence was long planned. He is receiving great praise for leaving game after the pl
ayers showed such disrespect for country!'
```

```
In [ ]: # create a single task with the input you specified above
put_result = crowd_client.put_task(function_name,
                                   task_name,
                                   {'text': text})
print('PUT response: {}'.format(
    {'status_code': put_result.status_code, 'task': put_result.json()}))
```



# Worker Preview

## Sentiment Analysis Instructions (Click to collapse)

Basic Details & Examples

Please select the sentiment of the text between positive, negative, and neutral. Only consider the attitude of the author, not your feelings about the contents of the text. If you cannot make a determination, whether because the language is wrong, the text is gibberish, or some other reason, please select "Cannot Determine".

Everything is wonderful.

Sentiment expressed by the content:

Positive
Neutral
Negative
Cannot Determine

You must ACCEPT the HIT before you can submit the results.

*Note for Workers: If you contact the Requester with a question about this HIT, please include the text to be analyzed in your message*

# Jupyter Notebook – [github/aws-samples/mturk-jupyter](https://github.com/aws-samples/mturk-jupyter)

## Get the result

Wait a few minutes before calling `get_task` to give Workers a chance to submit answers.

```
In [ ]: get_result = crowd_client.get_task(function_name, task_name)

print('GET response: {}'.format(
    {'status_code': get_result.status_code, 'task': get_result.json()}))
```

# Sample Response

Example response for a successful Task:

```
{  
  "taskName": "my-task-name",  
  "input": { "text": "Python is wonderful!" },  
  "problemDetails": null,  
  "state": "completed",  
  "result": { "sentiment": "positive" }  
}
```