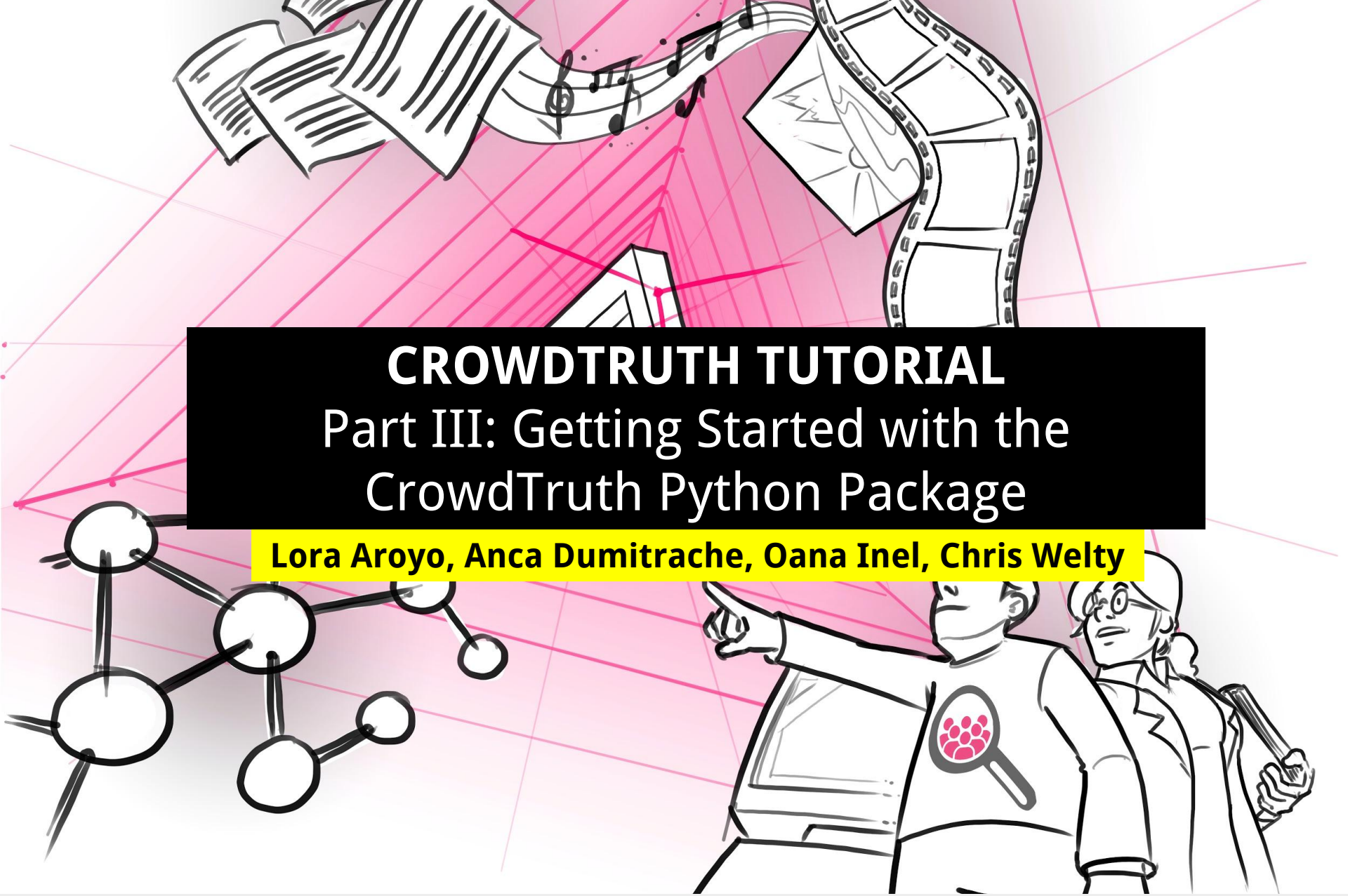

CROWDTRUTH TUTORIAL

Part III: Getting Started with the CrowdTruth Python Package

Lora Aroyo, Anca Dumitrache, Oana Inel, Chris Welty

Resources

GitHub repo: https://github.com/CrowdTruth/CrowdTruth-core

Step-by-step getting started guide: https://git.io/fAAn7

Exercises Session 3: https://git.io/fAAc0

If you have working **Python** & **Jupyter Notebooks** installations:

`pip install crowdtruth` installs Python package

```
git clone git@github.com:CrowdTruth/CrowdTruth-core.git
cd CrowdTruth-core/
python setup.py install
```

installs from source

If you don't have Python & Jupyter Notebooks, but you do have a Google account, use **Google Colab**:

- save the Colab notebooks to your personal Google Drive
- execute the notebooks straight from Drive

Step 1: Define the pre-processing configuration

```
import crowdtruth
from crowdtruth.configuration import DefaultConfig

class TestConfig(DefaultConfig):
  ...
```

The pre-processing configuration defines how to interpret the raw crowdsourcing input into an **annotation vector** that can be processed by the CrowdTruth metrics.

The configuration is defined as a **class**, inheriting from the Default configuration.

Configuration Class Attributes

```
import crowdtruth
from crowdtruth.configuration import DefaultConfig

class TestConfig(DefaultConfig):
  inputColumns = [...]  #list of input columns from the .csv file with the input data
  outputColumns = [...] # list of output columns with the answers from the workers

  open_ended_task = ... # boolean var, whether or not task is open-ended
  annotation_vector = [...] # list of possible annotations for closed tasks

  csv_file_separator = "," # column file separator
  annotation_separator = "," # output column annotation separator
  none_token = "NONE" # name of annotation vector component for no answer picked
  remove_empty_rows = True # whether to remove empty judgments from the data

  # method where any additional processing of the raw judgments is done
  def processJudgments(self, judgments):
    ...
    return judgments
```

Amazon Mechanical Turk and **Figure Eight** (former **Crowdflower**) output files are automatically processed by the package.

Custom file types can be processed using the `customPlatformColumns` attribute:

```
class TestConfig(DefaultConfig):
  customPlatformColumns = [
   # column names in the custom .csv file,
   # must be declared in this specific order
    "{$judgment_id}", # judgment ID column name
    "{$unit_id}", # unit ID column name
    "{$worker_id}", # worker ID column name
    "{$started_time}", # judgment start time
    "{$submitted_time}" # judgment submit time
  ]
  ...
```

Step 2: Pre-process the raw data from the crowd to get the annotation vectors

```
data, config = crowdtruth.load(

  # pick 1 out of file, directory, data_frame
  file = "...",  # path to crowd file
  directory = "...", # path to folder with crowd files,
                     # all with same configuration
  data_frame = ..., # pandas data frame with crowd data

  config = TestConfig())
```

Annotation vectors are now in `data["judgments"]["output.X"]`, where `X` is the output column defined in the configuration.

Step 3: Calculate the CrowdTruth metrics

```
results = crowdtruth.run(data, config)
```

The CrowdTruth quality metrics are stored in:

- `results["units"]` - input unit quality metrics
  - `results["units"]["uqs"]` : unit quality score
  - `results["units"]["unit_annotation_score"]` : ratio of workers that picked the annotation vs. all workers, weighted by worker quality
- `results["workers"]` - worker quality metrics
  - `results["workers"]["wwa"]` : worker-worker agreement
  - `results["workers"]["wsa"]` : worker-unit agreement
  - `results["workers"]["wqa"]` : worker quality score
- `results["annotations"]` - annotation quality metrics
  - `results["annotations"]["aqs"]` : annotatio

Resources

GitHub repo: https://github.com/CrowdTruth/CrowdTruth-core

Step-by-step getting started guide: https://git.io/fAAn7

Exercises Session 3: https://git.io/fAAc0

Exercises Session 3: https://git.io/fAAc0

1. Install CrowdTruth package & learn how to run ✓

2. Explore notebooks implementing CrowdTruth metrics for different tasks

- for local Jupyter server, download tutorial folder https://git.io/fAAC3
- for Google Colab, follow links in https://git.io/fAAc0
  ➡ *save .csv input as Google Sheets file + use Python 2 environment*
- if you don't want to code, analyze input & results .csv files *(links in notebooks)*

3. Compare an **open-ended** vs. **closed** config for the same task

4. Observe the effect of **dimensionality reduction** techniques over the quality metrics

5. Implement the annotation vector from Session 2