# Run Akshay’s Python Script

Get link to [script](https://github.com/rhubal/ArticleSearch)

Verify what file types are accepted in this script, tweak as needed. I believe only .csv or .xlsx

# For any un-identified fields, continue with Liezl’s script to use the OpenAI API code (via Assistant feature)

Set up [GH repo](https://github.com/lmaefos/CDE_detective/tree/master)

Test functionality/accuracy of [script](https://github.com/lmaefos/CDE_detective/blob/master/CDE_ID.py)

* Ongoing development. See GH for updates
* Tracking progress in [Jira](https://renci.atlassian.net/jira/software/projects/DUG/boards/2/backlog?selectedIssue=DUG-470&text=470)

# Parse unidentified field(s) that need further evaluation

Create script that pulls the row(s) (or entries if JSON) that Akshay’s code could not identify

# Transform un-identified field(s) to JSON

After parsing the unidentified field(s), incorporate [this script](https://github.com/lmaefos/CDE_detective/blob/master/JSONconverter_forsubmittedDD_v2024_07-01.py) to transform into JSON format (if needed)

*Note: see* ***JSONconverter\_forHEALCDEDD\_v2024\_06-25.py***

# Load JSON data dictionary

Incorporated

## OpenAI Call to CDE ID Detective Part 1

Incorporated

*Note: You cannot have file retrieval enabled AND JSON formatting response enabled at the same time, it was recommended to have 2 different assistants performing these tasks independently (1. Retrieving from the CDE knowledge base and creating a response and 2. Take the response from part 1 and standardize the output in desired JSON format)*

## OpenAI Call to CDE ID Detective Part 2

Script intended to utilize JSON response setting for parsing relevant information

## OpenAI Call to Fine-Tuning CDE ID Model

Create training data (in progress)

*Note: see* ***trainingdata\_v2024-07-24\_module1.jsonl*** *visit* [*OpenAI guide*](https://platform.openai.com/docs/guides/fine-tuning/common-use-cases) *for formatting.*

Load training data

*Note: Tried to load sample training data on 2024-07-25 and it failed ☹*

Test training data for accuracy

Create python script

# OpenAI Output

Ensure consistency in output responses

# Pull in Gaurav’s code block to get the CDE link

Get link to [GH repo](https://github.com/heal-data-stewards/heal-cdes)

# Complete Output

Create example of desired complete output

Identify which desired output works best for the code.

*Note: Dug team prefers JSON file, but Excel can work. See 2 versions of desired outputs (JSON file and Akshay’s output file)*