ACCELERATING DATA HARMONISATION USING MODERN DATA SCIENCE TOOLS

Authors: Peter Marsh[1], Lisa Van Aardenne[1], Pierre Kloppers[1], Chris Jack[1] , Nicholas Brink[2]

Background:

The HE2AT Center project is embarking on an ambitious endeavor to retrospectively harmonise a diverse collection of African cohort studies and trials. The ultimate goal is to compile a comprehensive longitudinal clinical database. So far, the project has meticulously identified over 200 eligible health studies and trials through a systematic mapping review. Presently, we have acquired and initiated the harmonisation process for 22 datasets.

Dealing with the sheer number and diversity of these studies has made the retrospective harmonisation process a monumental task. To tackle this challenge, we have developed a variety of tools and standard operating procedures. These tools serve to standardize data formats, extract metadata, and map variables of interest to a standardized ontology. Notably, the last step in this process, which we refer to as metadata harmonisation, has proven to be the most laborious and time-consuming.

We are convinced that the tools and methods we have devised for metadata harmonisation can be applied in various contexts. As a result, we have created and released an open-source desktop application that utilizes these tools. This application is intended to benefit the broader data harmonisation community. Figure 1. highlights where this tool fits into the broader HE2AT Center harmonisation process.

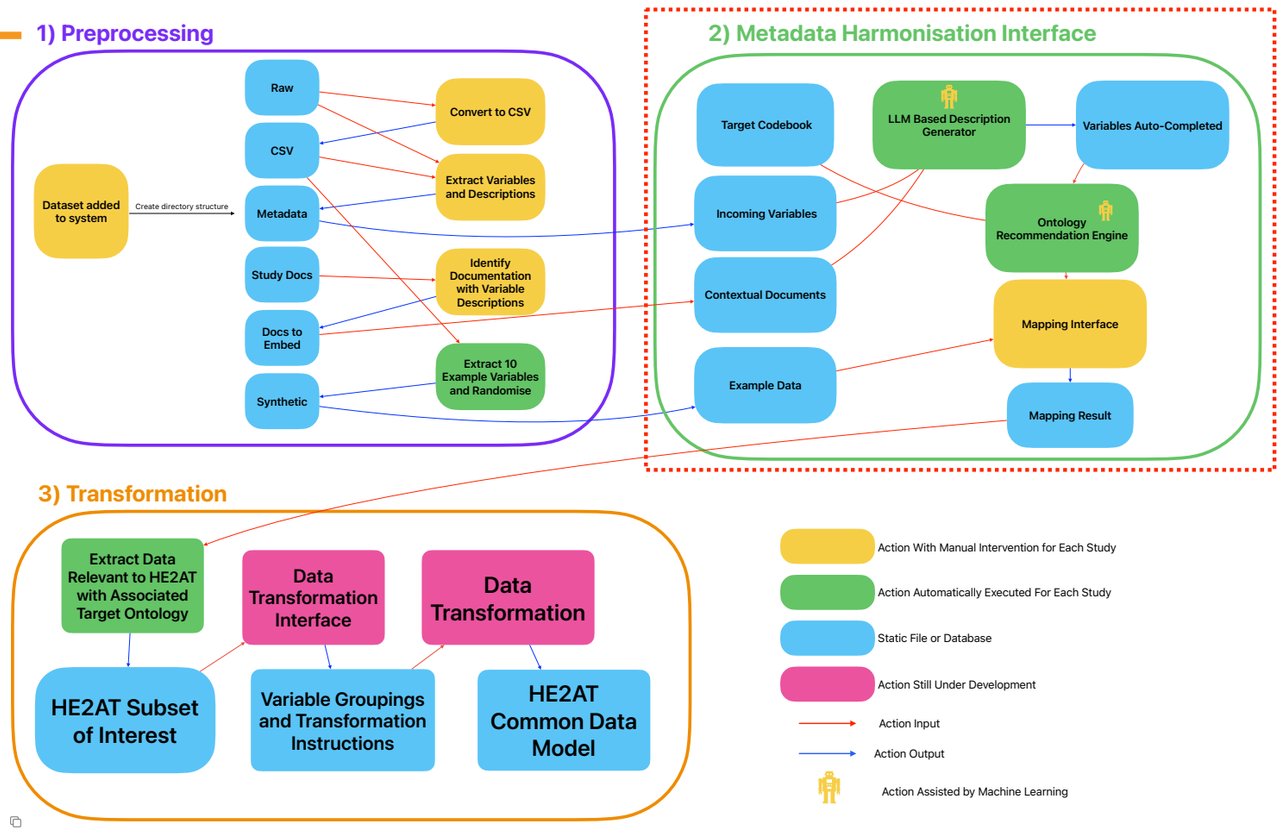

Figure 1: Schematic of the HE2AT Centre Data Harmonisation Workflow. The Metadata Harmonisation Tool is spun out of section 2 of the workflow.

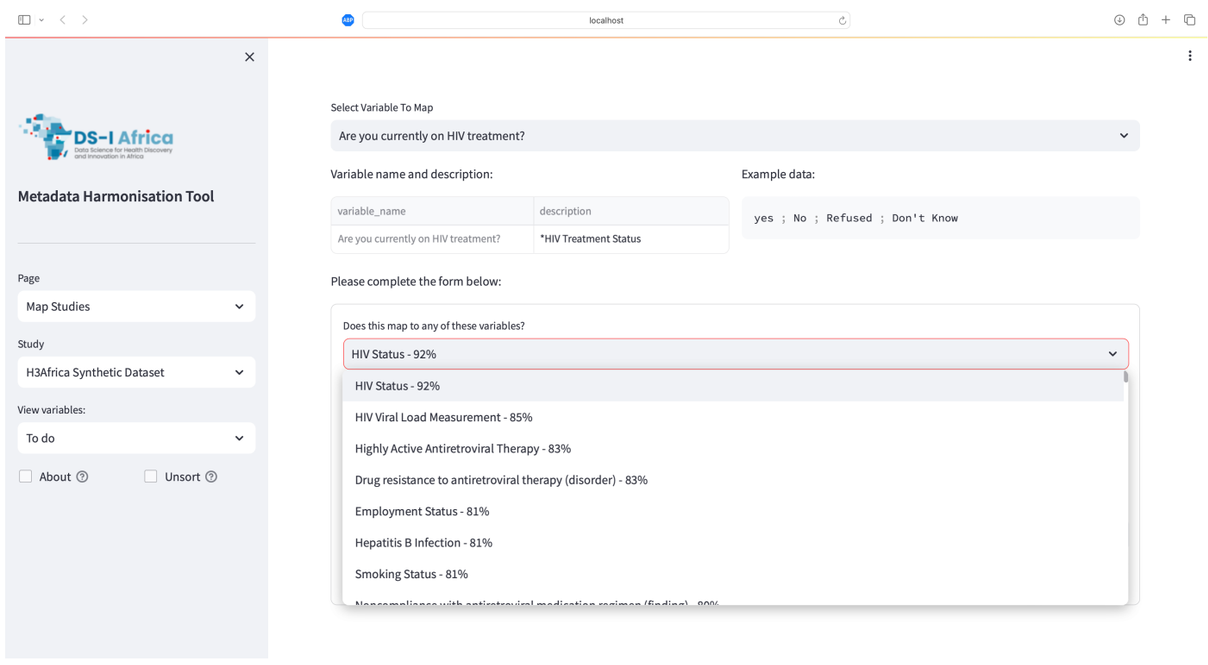

Figure 2: A Screenshot of the Metadata Harmonisation Tool being used to map the H3Africa Synthetic dataset. The * before the variable description indicates it has been automatically generated using a Large Language Model.

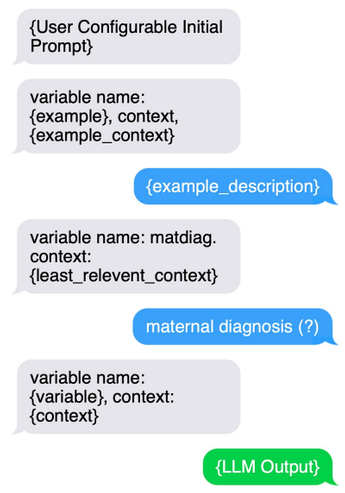

Figure 3: The prompt template used by the description generator. {} indicates a place holder for variable data.

What it does:

The Metadata Harmonisation Interface provides a convenient portal to match variables from an incoming dataset to a target set of ontologies. In this way the tool provides a similar role to that of the White Rabbit tool utilised by the OHDSI community (https://github.com/OHDSI/WhiteRabbit).

This tool differentiates itself by using Large Language Models to generate variable descriptions where none have been provided and by recommending the most likely target variable to map to. A confidence indication is provided alongside mapping recommendations. This dramatically speeds up the mapping process. Figure 2 shows the Metadata Harmonisation Interface in action.

How to use it:

The general workflow of using the Metadata Harmonisation Interface is as follows:

1. Follow the instructions on the GitHub page (https://github.com/csag-uct/Metadata-Harmonisation-Tool). To configure the python environment and run the Streamlit application on your local computer
2. Create and upload a target codebook. This is simply an excel or CSV file with two columns; 'var_name' and 'description'.
3. Upload the following information from datasets you would like to match to this codebook.
   - Study Name (required) [Text input]
   - Study Description (optional) [Text input]
   - Variables Table (required) [CSV table with *'var_name'* and *'description'* columns]
   - Example Data (optional) [CSV file with column headers corresponding to *'var_name'*s]
   - Contextual Documents (optional) [PDF file of codebook or study protocol]
4. Fine tune the variable completion prompt. The default is: "*As an AI, you're given the task of translating short variable names from a public health study into the most likely full variable name given some context. Only return the name no context. Use (?) to signify unhelpful context.*"
5. Confirm the correct data has been uploaded and API keys have been detected before running the description generation and recommendation engine.
6. Using the interface shown in Figure 2 map variables from incoming datasets to the target codebook.
7. Download the mapping results as a simple CSV file matching variables of interest to target variables. Alongside an autogenerated mapping confidence and any notes made during the mapping process.

SCAN ME FOR A VIDEO DEMONSTRATION

(or visit grco.de/beVQwG)

How it works:

The Metadata Harmonisation Interface compromises of two key parts:

First the LLM-based description generator provides a way to quickly and easily extract variable description information from complex free text documents such as study protocols or journal articles. While in an ideal world descriptions should come from a codebook and should match to standardised ontologies, in our experience this is often not the case. The description generator works by taking in a PDF document and converting it to plain text using the pdfminer python package. Next, we use a text-splitter from the Llangchain suite of python functions. This works by recursively splitting the text by the special characters: "\n\n", "\n", " " and "" until a text length of 1000 characters is reached. An overlap of 20 character between chunks is preserved to ensure no information is lost between chunks. A text embedding model is then used to get a vector representation of each chunk. This information is stored as a simple Numpy array. Next a prompt is constructed by taking an already completed variable and description pair and retrieving the most relevant context, calculated as the spatial distance between the chunk embeddings and the variable name embedding. A hard coded variable and description pair alongside the least relevant context is also included with a (?) appended to the description. This is an attempt to get the LLM to return some indication of whether the context has been useful. The prompt template is shown above in figure 3. This prompt style is known as few-shot prompting. If no context is provided by the user a similar prompt pattern is followed without providing the LLM with context.

The next step in the process is the ontology recommendation engine. This again uses text embeddings to retrieve vector representations of variable names and descriptions for both the target codebook and incoming datasets. Recommendations are then calculated using the spatial distance between vectors weighted 80/20 to descriptions. The interface utilises DuckDB to retrieve these recommendations from plain csv files.

Future Improvements:

- The tool is intended for use only with metadata therefore data privacy is not a limitation. For this reason, configuring and hosting this desktop application as a website is feasible. This will make the application more accessible to users not familiar with configuring a python environment.
- The tool is reliant on OpenAI for embeddings from the "text-embedding-ada-002" model and for variable autocompletions from the "gpt-3.5-turbo" large language model. This means that users of the app need to create an OpenAI account and will incur a nominal cost from using the app. Porting the app to open-source equivalents is a feasible step.

Author Affiliations
1. Climate System Analysis Group, the University of Cape Town, Cape Town, South Africa
2. Wits Reproductive Health and HIV Institute (Wits RHI), University of the Witwatersrand, Johannesburg,

Acknowledgment: The research was supported by the Fogarty International Center and National Institute of Environmental Health Sciences (NIEHS), and OD/Office of Strategic Coordination (OSC) of the National Institutes of Health under Award Number U54TW012083. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

Corresponding Author: Peter Marsh
Email: peter.marsh@.uct.ac.za