Documentation Deployment Masters Thesis Graf

https://datatalkturk.ifi.uzh.ch/list_folder_structure.php

This is a PHP script that uses the `listFolderFiles` function to recursively scan the current directory and display all files and subfolders of the project in an ordered list format on an HTML page.

Pipelines

All_chat_messages_into_database_pipeline.php

URL: https://datatalkturk.ifi.uzh.ch/pipelines/all_chat_messages_into_database_pipeline.php

This is a PHP script that imports data from a JSON file called "merged_results.json" located in a folder called "result_files_from_jupyter". The script opens the file, reads the contents, and decodes the JSON data. Then, it iterates through the data, extracting specific information from each record, and inserts it into a database table called "chat_messages" using the PDO (PHP Data Objects) library. If there is an error executing the insert statement, the script will print an error message and exit. If no import is set to 1 in the beginning, there will be no import and the script exits.

hypotheses_into_database_pipeline.php

URL: https://datatalkturk.ifi.uzh.ch/pipelines/hypotheses_into_database_pipeline.php

This script is similar to the previous one, but it is designed to extract and store hypothesis statements from the JSON data. The script first defines an array called "hypothesis_question_ids" that maps experiment conditions to a list of question IDs for which the script should extract hypotheses. Then, it opens the "merged_results.json" file and reads the contents, similar to before.

Then, it iterates through the data, and for each record, it checks the "json_answers" field and compares the question ID of each answer with the question IDs in the "hypothesis_question_ids" array. If a match is found, the script extracts the hypotheses from the answer and stores each hypothesis in the "hypotheses" table in the database using the PDO library.

Additionally, it does some additional filtering and printing of the data if the conversational style is "viz" and the length of the hypothesis is less than 20 characters.

quality_into_jupyter_pipeline.php

URL: https://datatalkturk.ifi.uzh.ch/pipelines/quality_into_jupyter_pipeline.php

This script exports data from a database table called "hypotheses" and "hypotheses_quality_responses" and creates a CSV file. It uses the PHP Data Objects (PDO) library to execute a SQL query that selects all rows from the "hypotheses" and "hypotheses_quality_responses" table where the "hypotheses.id" is equal to "hypotheses_quality_responses.hypothesis_id" and the "hypotheses_quality_responses.created_on" is before '2022-11-30 20:00:00' and orders them by dataset, conversational_style, information_elements and hypothesis_id.

The script then opens a file 'output.csv' and writes the results of the query to that file. The script uses the PHP function "fputcsv" to write each row of the query result to the file as a new line. The script also sets a flag "EVEN_RATING_ACROSS_CONDITIONS" to false and if it is set to true it will only export 48 rows per conditions.

The script also sends the appropriate headers to the browser so that the generated CSV file is downloaded as an attachment rather than being displayed as plain text. By that, the url can be directly inserted into the jupyter notebook.

time_into_jupyter.php

URL: https://datatalkturk.ifi.uzh.ch/pipelines/time_into_jupyter.php

This script exports data from a database table called "chat_messages" and creates a CSV file. It uses the PHP Data Objects (PDO) library to execute a SQL query that selects all rows from the "chat_messages" table and orders them by worker_id and timestampInMillis.

The script then opens a file 'output.csv' and writes the results of the query to that file. The script uses the PHP function "fputcsv" to write each row of the query result to the file as a new line.

It also calculates the time between last message and current message for each worker and check if the message is a hypothesis based on the message_nr and the condition.

The script also sends the appropriate headers to the browser so that the generated CSV file is downloaded as an attachment rather than being displayed as plain text. By that, the url can be directly inserted into the jupyter notebook.

tlx_into_jupyter_pipeline.php

URL: https://datatalkturk.ifi.uzh.ch/pipelines/tlx_into_jupyter_pipeline.php

This script exports the last six messages of each worker's conversation to a CSV file. It queries the `chat_messages` table in the database and retrieves the worker's id, the assignment id, the message number, the message text, the dataset, the conversational style, and the information elements. It then calculates the message number by subtracting the message number from the maximum message number for that worker. The script then exports the data to a CSV file and sends the appropriate headers to the browser so that the generated CSV file is downloaded as an attachment rather than being displayed as plain text. By that, the url can be directly inserted into the jupyter notebook.