Instruction for EyetrackerAnalysis Repository

This document is used for explaining the use the code stored in https://github.com/CPP-HAPII/EyeTrackerAnalysis that was used to analyze the eye-gaze data produced by Gazepoint Eye Tracker. Python3 and its requirement must be used and downloaded to use this program.

The main structure of the repository is as follow

```
EyetrackerAnalysis
|
|--- core (Dir)
|--- main.py
|--- README.md
|--- other_files (Dir)
|        |--- database (Dir)
|        |       |---users.sql
|        | …
|
|--- requirements.txt
|--- user_study_data
        |
        |--- Combined (Dir)
        |--- Combined_Median_Split (Dir)
        |--- All-Relevances.csv
        |--- Combined_duration_click.csv
        |--- combined_relevances.csv
        |--- results.csv
        |--- User13
        |       |
        |       |--- initial_file.csv
        |       |--- initial_fixation.csv
        |       |--- task_file.csv
        |
        |--- User14
        |       |
        |       |--- …
        |--- …
        |--- User48
        |       |
        |       |--- …
        |--- users_median_split.csv
        |--- users_proficiency.csv
        |--- users_raw.csv
        |--- users.csv
```

The following is a brief explanation for each folder/file located inside the repository
- core (directory): Contains all the python files used for the analysis
- main.py: The main python file to invoke commands from the files in core directory
- README.md: Contains a brief explanation on how to setup the environment and use the available commands
- requirements.txt: Contains a list of python libraries needed to run the commands
- user_study_data (directory): Contains all the files required to do analysis

The users.sql file in the other_files/database/ path contains the data recorded throughout the experiment and used for the analysis.

The following files/directories can be found in the user_study_data directory:
- All-Relevances.csv: csv file containing all the timestamps when the user submitted a relevance judgment (Obtained from the Database – Relevances table)
- results.csv: csv file containing all the search results presented to the users during the user study (Obtained from the Database – Results table)
- users_median_split.csv: csv file containing the distribution of users based on language proficiency split into Low/High based on median split on four different categories (ILR, Test, CEFR, and Paper). Each column is coded using 3 letters.

Character #	Explanation	Possible Char
1	Language Prof Level	l/h: Low/High
2	Language	c/s/e: Chinese/Spanish/English
3	Criteria for the split	s/t/c/p: ILR/Test/CEFR/Paper

- users_proficiency.csv: csv file containing the distribution of users based on their L1/L2

Column Name	Explanation
c2e1	Chinese Users: Chinese L2, English L1
c1e2	Chinese Users: Chinese L1, English L2
s2e1	Spanish Users: Spanish L2, English L1
s1e2	Spanish Users: Spanish L1, English L2
s1e1	Spanish Users: Spanish L1, English L1

- users_raw.csv: csv file containing the raw test score from the questionnaires before converted into Low/Medium/High
- users.csv: csv file containing userID, L1, and Proficiencies of the participants
- A list of User folders

Each user folder initially contains only the following 3 files:
- initial_file.csv: individual user csv file containing all eye gaze data (obtained from Gazepoint)
- initial_fixation.csv: individual user csv file containing eye gaze data per fixation (obtained from Gazepoint)
- task_file.csv: individual user csv file containing timestamps of when the user moves from a page to the next (obtained from the Database – TaskPage table)

The following is a list of commands available for data analysis:
- *process_all*
- *process_one_user*
- *combine_csv*
- *compare_order_click*
- *switch_penalty*
- *compare_by_relevance*
- *median_split*

process_all
➤ *python3 main.py process_all user_study_data/*

This command is used to process all the users located at user_study_data/ at once. Each user folder must contain the word "User" to be processed, and any user folder containing the phrase "NAH" in it are skipped (due to them being deemed outliers/unusable for analysis).

This command produced several files in each user folder. However, the ones that are of importance are:
- combined_aoi.csv
- combined_aoipage.csv
- combined_page.csv

Each of the file contains the processed features for the specific user per-AOI, per-AOI & Page, and per-Page, respectively.

process_one_user
➤ *python3 main.py process_one_user user_study_data/<User# Folder>/*

This command is used to process **one** user in a specific folder. This command produces the same files as the *process_all* command. It can be used when a few new participants were added instead of *process_all* to avoid longer processing time.

combine_csv

> ➤ *python3 main.py combine_csv user_study_data/*

This command is used to combine the aoi/aoipage/page csv files from **all** users that were produced by the previous commands. Only user folders containing the name "User" are processed, and the ones containing "NAH" are skipped. The results are stored in user_study_data/Combined/ folder

This command produces various csv files combining the results from each user based on specific criteria. Mostly, it combines each of the aoi/aoipage/page files for all users while separating them based on the 3 different languages. Thus, you can see csv files like aoi_english, aoi_spanish, aoipage_english, page_chinese, etc. in the Combined folder.

compare_order_click

> ➤ *python3 main.py compare_order_click user_study_data/combined_relevances.csv*

This command is used to produce several csv files used to compare user's speed in performing relevance judgment based on the layout of the search result page as well as the language of the result. Since there are 4 types of layout and 3 languages, 12 csv files are produced in analysis_results/comparison_results/by_order/ folder.

switch_penalty

> ➤ *python3 main.py switch_penalty user_study_data/combined_duration_click.csv*

This command is used to produce several csv files used to compare user's speed in performing relevance judgment based on the occurrence of language switch as well as the language proficiency of the users, particularly their L1 vs L2. The csv files are produced in analysis_results/comparison_results/by_order/ folder.

compare_by_relevance

> ➤ *python3 main.py compare_by_relevance user_study_data/combined_relevances.csv*

This command is used to produce several csv files used to compare user's speed in performing relevance judgment based on whether or not they think that a result is relevant. Since there are 2 possible answers (yes/no) and 3 languages, 6 csv files are produced in analysis_results/comparison_results/by_switch/ folder.

median_split

> ➤ *python3 main.py median_split user_study_data/combined_median_split/*

This command is used to convert the combined csv files produced with *combined_csv* command. Please note that the *combined_csv* command produced the result based on

Low/Medium/High classification while this command convert it into Low/High classification based on 4 different criteria (Self/Likert/CEFR/Paper). It takes the files in user_study_data/combined_median_split/original folder as an input and produce the output in the other 4 folders in user_study_data/combined_median_split/ folder.