

User Testing Guide v1.0

Welcome to yoGERT user testing guide! Thank you for taking the time to provide your valuable feedback to keep improving our open source Python toolbox.

This document includes background information on the toolbox, user testing goals, user testing guidelines, and user testing instructions. *Let's get started!*

Table of Contents

Background Information	1
Relevant Terminology	1
User Testing Goals	2
User Testing Guidelines	2
User Testing Instructions	2
1. Getting yoGERT	2
2. Test yoGERT	3
Suggested Use Case Scenarios:	3
3. Answer Questionnaire	4
References	4

Background Information

Relevant Terminology

Term	Definition (Paez, 2023)
Entity	A uniquely identified GPS recording device for a person, animal, or object. It is uniquely identified with a device ID number.
GPS Ping	A collection of attributes that at a minimum contain an identifier for position in space, GPS coordinate with latitude and longitude, and the time stamp.
Trace	A collection of GPS pings for one entity.
Segment	A line connecting 2 GPS pings.
Transportation Mode	Movement behavior that is identified from segments. It can be a walk, drive, or stop mode.
Episode	A collection of segments for one type of transportation mode.

Network Graph	A graph composed of vertices and directed edges. Each vertex is identified with a unique ID. Each edge may have additional attributes, such as labels to describe their use, speed limits, etc.
Route	A travel path for a set of GPS pings that is constrained by the network graph layout.

User Testing Goals

You are here to help the yoGERT team in assessing the usability of the system. Here are the goals of this user testing version:

- How effectively does the system meet the appearance requirements?
- How effectively does the system meet the ease of use requirements?
- How effectively does the system meet the learning requirements?
- How effectively does the system meet the cultural requirements?

User Testing Guidelines

As the primary tester for usability here are helpful points to follow:

- Prerequisite for testing is being familiar with Python3.
- Having a unix command line environment with python 3 and pip installed.
 - For more help please refer to [How to Install Python](#).
 - For more help please refer to [How to Install pip](#) or [How to Install pip via Python](#).
- Read provided documentation including this document before using the system.
- Read the provided questionnaire document.
- Keep a diary to record any thoughts, feelings, or questions when using the system.
- If faced with any issues try to refer to the “User Guide” document before asking yoGERT’s testing team.

User Testing Instructions

1. Getting yoGERT

Before using the system you must get the system first. Here are the instructions to get yoGERT:

1. Clone [PyERT-BLUE](#) repository and work on the [userTestingV1.0](#) branch.
 - a. For more help refer to [How to Clone Repo](#).
 - b. For more help refer to [How to Use Branches](#).
2. Move to the src folder to access the toolbox modules.
 - a. For more help refer to [How to use cd Command](#).
3. Run the setup script, setup.sh.
 - a. For more help refer to [How to Run Script](#).
4. Create your own python file(s) where you will be using the toolbox modules
5. Inside your python file(s) import the module name(s) you would like to test.

2. Test yoGERT

Test the system by following the [User Testing Guidelines](#) and performing a use case scenario. A use case scenario can be your own idea of how the system should be used or choosing one of the [below use case scenarios](#). Either way please document which option(s) you chose to execute.

Suggested Use Case Scenarios:

- A. Using yoGERT toolbox functionality to analyze geo-data for one entity.
 - a. Preprocess the entity's geo-data stored in a CSV file.
 - i. Open the newly created file.
 - b. Generate episodes for the preprocessed CSV output called trace.
 - i. Open the newly created file(s).
 - c. Create a summary mode file for the episode generation output.
 - i. Open the newly created file.
 - d. Create a statistics file for the trace.
 - i. Open the newly created file.
 - e. Create an activity location file for the trace.
 - i. Open the newly created file.
 - f. Create a shortest route for the trace.
 - g. Create a shortest route for an episode.
 - h. Create an alternative route for the trace.
 - i. Map the shortest route for the trace.
 - i. Open the newly created file.
 - j. Map the shortest route for an episode.
 - i. Open the newly created file.
 - k. Map the alternative route for the trace.
 - i. Open the newly created file.
 - l. Map activity locations of the trace.
 - i. Open the newly created file.
 - m. Map an episode.
 - i. Open the newly created file.
 - n. Repeat steps g and j for as many episodes as you want limited by the number of generated episode CSV files.
- B. Using yoGERT toolbox functionality to analyze geo-data for n entities.
 - a. Preprocess the entities' geo-data stored in a CSV file.
 - i. Open the newly created files.
 - b. Generate episodes for one of the preprocessed CSV outputs called trace.
 - i. Open the newly created file(s).
 - c. Create a summary mode file for the recent episode generation output.
 - i. Open the newly created file.
 - d. Create a statistics file for the recent trace.
 - i. Open the newly created file.
 - e. Create an activity location file for the recent trace.
 - i. Open the newly created file.

- f. Create a shortest route for the recent trace.
- g. Create a shortest route for an episode from the recent trace.
- h. Create an alternative route for the recent trace.
- i. Map the shortest route for the recent trace.
 - i. Open the newly created file.
- j. Map the shortest route for an episode from the recent trace.
 - i. Open the newly created file.
- k. Map the alternative route for the recent trace.
 - i. Open the newly created file.
- l. Map activity locations of the recent trace.
 - i. Open the newly created file.
- m. Map an episode from the recent trace.
 - i. Open the newly created file.
- n. Repeat steps g and j for as many episodes as you want limited by the number of generated episode CSV files.
- o. Repeat steps b-n for as many traces as you want limited by the number of preprocessing CSV file outputs.

3. Answer Questionnaire

Answer the questions on the provided questionnaire document.

References

1. A. Paez, "Notes Concerning Path Imputation: Terminology and Concepts." McMaster University, Hamilton, 2023.