Introduction

If you are new to working with verbal reports and are keen to use the framework described in our paper but don't know how to start, this guide is for you. It will walk you through using the framework for both offline (i.e., lab-based) and online experiments.

Before we begin:

As JsPsych is one of the most popular frameworks for experimental psychologists, excelling in running both online and offline experiments, we developed this framework to be part of the JsPsych environment. Consequently, this tutorial assumes that you are already familiar with the JsPsych framework. It also assumes that you have either already coded your experiment or are planning to code one within the JsPsych framework. In other words, at this point, **all you need to know is simply how to apply this software to your experiment**.

For a smooth run through this guide, you will need a few key things. Lets go through them:

1. **Know about a computer's Terminal and how to use it** In some steps, you will be requested to use your computer's Terminal. The Terminal is a powerful tool that allows users to interact with their computers and perform tasks efficiently and automatically. While many users are accustomed to using a graphical user interface (GUI) with a mouse and keyboard, the Terminal allows users to execute commands and perform tasks directly by typing text commands.

   If you're a beginner using the Terminal (or Shell on a Mac), there's no need to panic. Here are some useful links for you: a tutorial for mac users, and a tutorial for windows users.

2. **Know about the computing power needed to run this framework** - Understanding the Computing Power Required to Run This Framework - To transcribe recordings (the recorded verbal reports) and perform text analysis, substantial computing power is necessary. You can check how much your

computer currently has here. What constitutes sufficient computing power? For the large Whisper model, used exclusively for the transcription of recordings, you will need approximately 10 GB of VRAM.

(a) For **Mac** users: In the top-left corner of your desktop screen, click the Apple icon. From the Apple menu that appears, select "About This Mac." The "Graphics" section will display details about your video hardware, including the available memory.

Checking GPU Capacity: Click the Apple icon in the top-left corner of the screen. Choose "About This Mac" from the Apple menu. Next, click on the "System Report" button. In the System Information window, select "Graphics/Displays" from the sidebar to view detailed information about the GPUs in your Mac.

(b) For **Windows** users: To check your basic specs on a Windows 10 PC, click on the Windows start button, then click on the gear icon for Settings. In the Windows Settings menu, select System. Scroll down and select About. From here, you will see specs for your processor, RAM, and other system information. For checking for **GPU** capacity: Right-click on the taskbar and select "Task Manager" or press "Ctrl" + "Shift" + "Esc". In Task Manager, click on the "Performance" tab. On the left side, you will see "GPU" listed along with CPU, Memory, and Disk. Click on "GPU" to see details about your graphics card, including the number of GPUs.

However, most computers may not have sufficient GPU resources. If this is the case with your computer, please consult with your IT team about how to meet these requirements. When discussing your needs with the IT team, be sure to communicate some important details about your project, including:

- Mention that you are planning to have recordings transcribed using the Whisper model from OpenAI. You can provide them with this link for more information whisper model card).

- Explain that you will later perform extensive text analysis on the transcriptions.
- Inform the team about the number of participants involved in your project.

Once you know how to open your Terminal (or Shell) and have confirmed that you have sufficient computing power for the transcription task and subsequent analysis, you can begin with the following steps:

Offline Experiments

Note: Even if you plan to run your experiment online, you must still follow steps 1-5 listed below.

1. To begin, you will need to download our folders containing the code scripts for running your text analysis. These scripts are all stored in one folder, which can be downloaded using one of the following methods:
   - Download the Zip file from this Github Page
   - Clone it from this Github Page
2. Save the downloaded files to your preferred location. You will now have a folder that contains multiple sub-folders, all of which are essential for the recording and analysis processes to function properly.
3. Copy your study code (the index.html file) into the `webserver/public` folder.
4. We have prepared a template file named `index_template.html` in this folder. This template includes the necessary code to add recording functionality and demonstrates the basic structure. Please refer to the needed basic code structure in our GitHub Repository.
5. If you used the template file please rename the `index_template.html` to `index.html`

6. **Very Important Note: Unfinished trials will be deleted after a set period** (in our study, this was about 120 minutes). The duration might vary in your case. Therefore, if you require more time before data is deleted, please navigate to `LLM/webserver/config.json file` and adjust the `MINUTES_TO_DELETE` setting. Set this duration to a number greater than the length of your experiment to ensure that participants' data is not deleted before they have completed their sessions.

7. Great! Now you are all set to run-test the experiment locally on your computer. **But how?**

   (a) Install Docker by following the installation guidelines for your operating system. If you are using a Mac, you can find the instructions here, if you are using windows, this link) (see Figure 1)

   (b) Open the Docker app that you just installed. If you cannot find it, check your Applications folder.

   (c) Wait until the Docker Engine is started (see Figure 2 and 4).

   (d) Open your Terminal (/Shell)

   (e) Navigate to the folder, in which you saved Experiment's code is (learn here how to navigate to folders using the terminal). In our case, this is called `$ cd your path to the folder/LLM/webserver` .

   (f) Now we need to start the Docker-container (learn more about what a Docker container is here). To do that, run the following commend in your Terminal (Shell): `docker-compose -f docker-compose.yaml up --build -d` , see Figure 3.

   (g) Wait until the Docker container is started. This may take a while because it downloads all necessary dependencies (see Figure 5)

   (h) Enter this line into your browser (any browser should work) to test your experiment locally: `http://localhost:8000` .

(i) The first time you open this browser page, there will be a request to allow microphone access. Once you allow the microphone to be activated, you will see your study displayed in your browser.

8. Run through your experiment. Remember that at this point, you are running your study on your own device, so anything you do will be accessible and saved on your computer and **not on a server**).

9. When you are finished with your experiment, it will ask you the following "Your voice recording will be saved and sent to our server. Do you want to continue?"

10. Open the folder where you stored the experiment, inside navigate to the `resources` folder (`LLM/resources`) that you saved earlier. If everything worked as expected, you will see a sub-folder within the "resources" folder. This sub-folder will be named with the ID allocated to the participant (because you tested the experiment locally, this new ID is yours, as you were the participant at this point, right?). Inside this folder, you will find many .wav files, which are your recorded trials. You will also find (usually somewhere around the bottom of the folder) a file named `trial_data.txt` . If you can see this file, you can rest assured that everything was recorded properly (e.g., all the trials were recorded and are inside the folder).

11. Because we are done at this point, we should make sure to stop the container. To do that, go back to (or re-open) the Terminal/Shell and enter this line: `docker-compose stop` . **Very Important Note**: If you made any changes to your experiment, Docker needs to be informed. This step (of stopping the container, making some changes and restarting the container) must be applied every time you make a change to your experiment's code. For example, if you changed your instructions, you will need to stop the container and restart it after you have applied the changes (to start the container see step 6.5).

12. Run the transcription of your recording:

(a) Since each project is different, it's time to get familiar with the features the framework offers for the analysis of verbal reports. Here is the link to the of the different features we offer under this documentation. For example, if you are not interested in summarizing the verbal reports (e.g., perhaps because they are too short to be summarized), you can omit this feature. To make specific changes, navigate to `LLM/data_pipeline` and open the `config.json` file. Adjust the variables according to your personal needs and save those changes.

(b) Once you are happy with those changes, you can start running the transcription and analysis. The first question to ask is: Do you have enough computing power to run the transcription? If you don't know, go back to the top of this blog and see how to find out more about this topic. In essence, there are two ways to get enough computing power:

- **University Resources:** Most universities should have GPUs available for your use. Your university IT team should be able to provide help.
- **Renting GPUs from Big Tech Companies:** If university resources are not sufficient, you can rent computing power from big tech companies. See sub point below.

i. For those who do not have access to GPUs but can chip in a bit of money to rent some from big tech companies, we have prepared a Google Colab template for you (note: this is still in progress). You can run it online and pay Google as you go. However, **it is important to note that the privacy regulations of these companies might be compromised**. Since you are dealing with sensitive data (you participants are being recorded), you should make an informed decision and consult your ethics committee to determine whether using Google Colab for this part of the analysis is appropriate.

ii. Another possibility, if you have limited access to GPUs, is to use a smaller model. This option requires only a single GPU (a small amount

of computing power). We have prepared a template using a distilled version of the large transcription model. While this model is slightly less powerful (approximately <1%), it will still get the job done. You can find it in the configuration file (note: this is still in progress).

(c) Once you have resolved the computing power issue, it's time to open your terminal/shell.

(d) In your terminal, navigate to `LLM/data_pipeline`

(e) Execute `docker-compose up` in your Terminal to start the analysis script. Please be patient, as this process may take some time.

(f) Once it's finished, you will find a .csv file with the entire data in the `LLM/Output` folder on your computer.

Congratulations, you are done now! You have successfully completed the setup and analysis of your verbal reports. Enjoy the unlimited possibilities that come with analyzing these reports, and make the most of the insights you can gain from your data. Happy analyzing!

Online Experiments

1. To start, follow points 1-5 from the guide to offline experiments.

2. Test your experiment locally, (check number 6-7 to do so) and happy with this version), send them the folder you just created in points 1-5. It is now time to ensure that you have a server to host your experiment. Contact your university IT department to set up a server for your experiment. **Tip**: you could also send them the link to the Github so they can prepare the requirements for this project.

3. When you are done collecting the data on the server, download the Resource folder from the server. Then, proceed as described above starting from point No. 9.

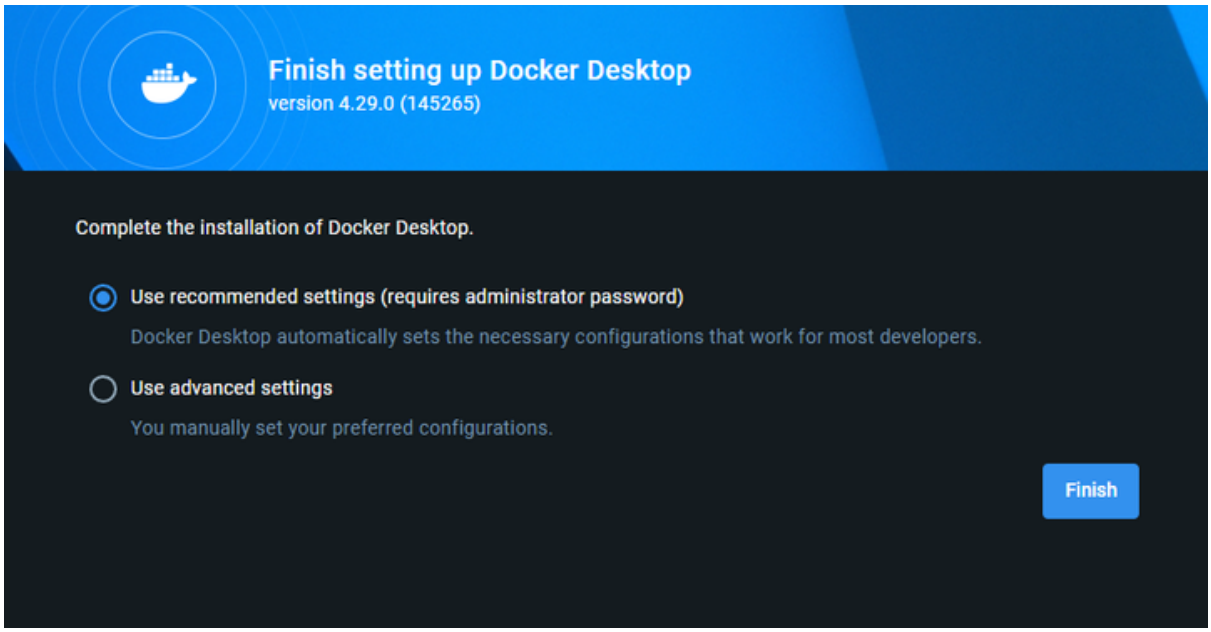

Figure 1. Installing Docker

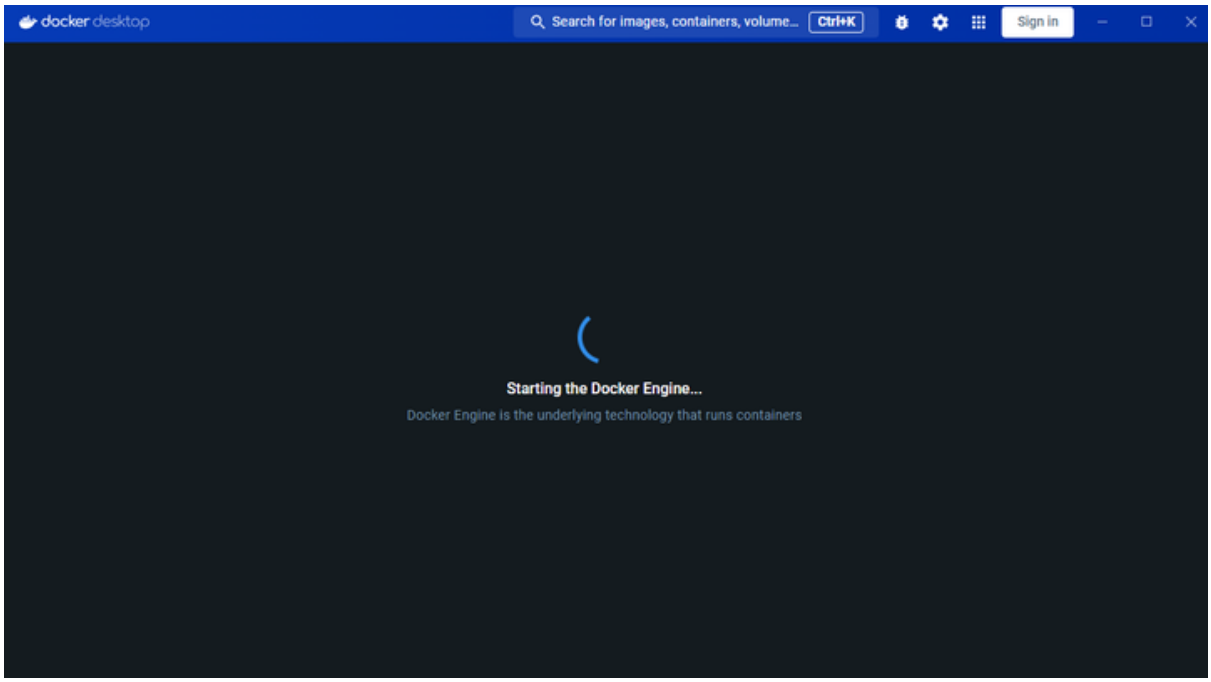

Figure 2. Docker starting

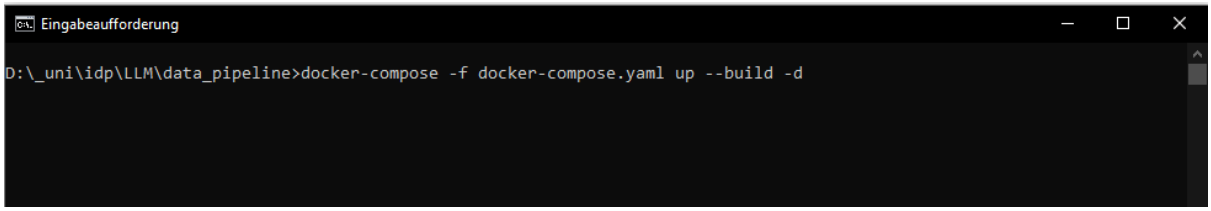

Figure 3. Docker ready

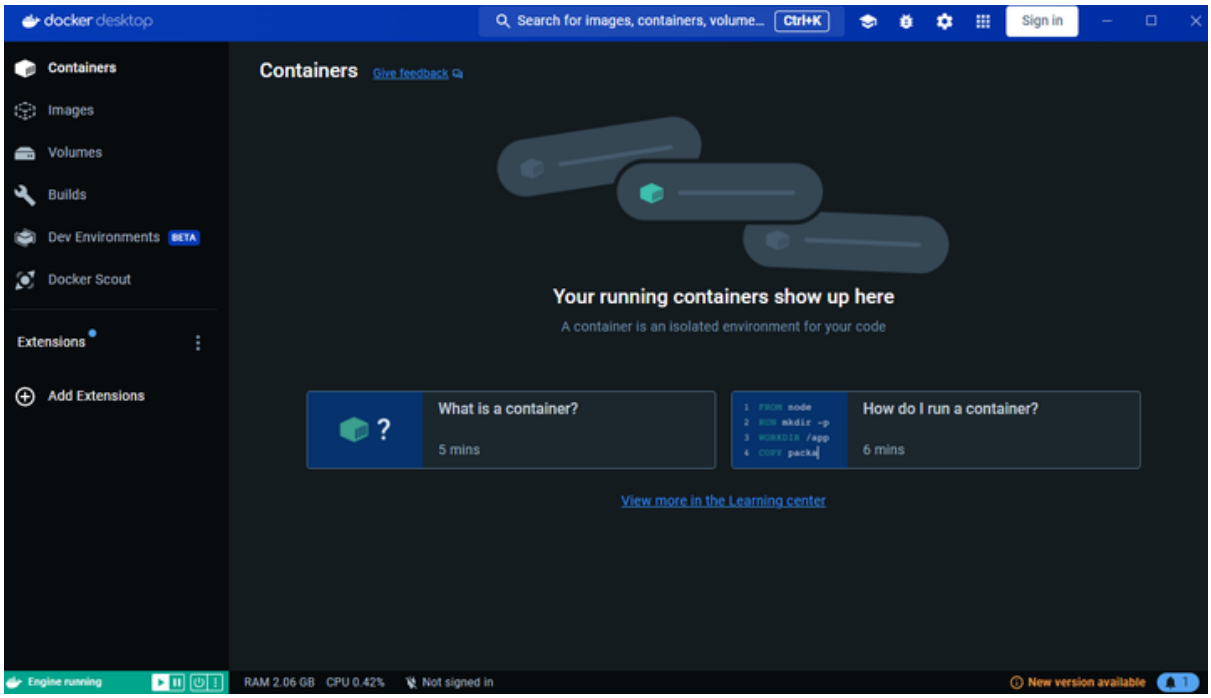

Figure 4. Start Docker container

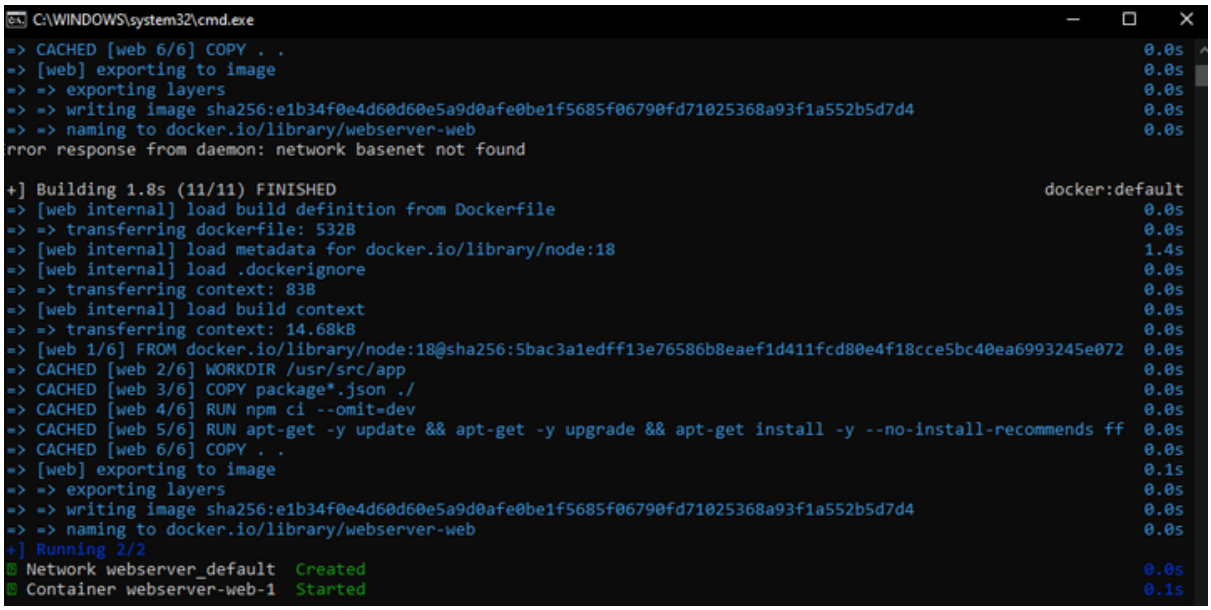

Figure 5. Docker container started