User Documentation (22-1-R-2)

Dr. Dan Lemberg
Mrs. Elena Kramer

Rafael Elkoby - rafaelelkoby2@gmail.com
Dan Gutchin – dando400@gmail.com

June 2022

Contents

1 Folder Organization

The project folder contains folders and files. The arraignment will be the same for those using Google drive and those who run it locally. The organization will be elaborated when each subsection will be a folder contained in the BioKey folder.

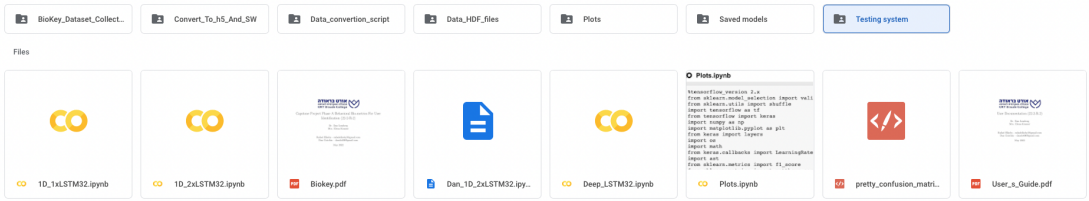


Figure 1: The BioKey Folder content

1.1 BioKey

- **pretty_confusion_matrix.py** - This file is used by the Plot.ipynb file. All the information regarding the way the file is used can be documented and elaborated inside the Plot.ipynb Jupiter Notebook.
- **1D_1xLSTM32.ipynb** - This file contains the code for the 1D_1xLSTM32 model.
- **1D_2xLSTM32.ipynb** - This file contains the code for the 1D_2xLSTM32.ipynb model.
- **Deep_LSTM32.ipynb** - This file contains the code for the 2xLSTM32 model.
- **Dan_1D_2xLSTM32.ipynb** - This file contains the code for the 2xLSTM32 model That we trained using the data we collected according to the Buffalo dataset.
- **Biokey_Project_Book.pdf** - This the project book in which we describe our work.

1.2 Data_HDF_files

This folder contains the HDF files that contain the data That is ready to be fed into the NN. the files are labeled as Rand_WSXX.h5 where the XX value marks the window size for each data file. For example, Rand_WS50.h5 is a file that can be found inside the folder, and the window size used on the data is 50. Here you can find the file that contains the data we collected according to the Buffalo Dataset "Rand_Dan_WS30.h5". This file is used to train the model used later on in the authentication program.

1.3 Plots

This folder contains the plots generated by the Plot script, which can be found inside the main folder as Plots.ipynb.

1.4 Saved models

This folder contains the HDF files that contain the models that were already trained using the corresponding model code in the Biokey folder. The naming system is meaningful. For example, 1D_2XLSTM32_rand_YY.h5, the last two characters marked by YY are the size of the window used in the model. Note that any data file loaded that doesn't have the same window size mentioned at the end of the trained model name won't work with it.

1.5 Convert_To_h5_And_SW

This folder contains the scripts that applies the sliding window on data collected from text files that were converted to our format by another script. the **DataToHd5_SlidingWindow.py** file is for local use as it doesn't require mounting google drive. The **DataToHd5_SlidingWindow.ipynb** file is the one to use over Google drive. The **"Dan_hdf5.py"** file is for local use, and it is the same script as **DataToHd5_SlidingWindow.py** where the only difference is that it uses the data collected by us instead of only the data from the Buffalo dataset. The resulted file from this script is used to train the **Dan_1D_2xLSTM32** model.

1.6 Data_convertion_script

This folder contains the script "convert.py" that coverts the buffalo dataset format into the format we applied in our research.
Note the script cant work unless you posses the Buffalo dataset. To get the dataset check the instructions in this link.
The folder contains 3 other folder which contains the converted data resulted from the conversion script.
The folders are -

- **Converted_S0**
- **Converted_S1**
- **Converted_S2**

The **Dan Data** folder contains the data collected by us according to the collection method presented in the Buffalo dataset. Because we collected it, no conversion was needed as we collected it using the dataset collector script.

1.7 BioKey_Dataset_Collector_ID

This folder contains the files that compose the data collector program we built.

- **database.py -** This file contains the database class where the database connection details are contained, and the connection is established.
- **helperMethods.py -** This file contains helper methods to validate text input like id and emails and a UI helper method for a popup window.
- **questions.py -** This file contains the questions given to the user.

- **welcomeUI.py -** This file contains the main method as well as the code for the UI welcome window.
- **q1.py -** This file contains the UI code for the questions window, as seen in figure 8, and the code that starts the keystroke listener thread.
- **keyboardFeatureExtraction.py -** This file contains the listener code where the keystrokes are collected. And the functions that are activated when a press or a key release takes place. The file also includes the code that does the post processing before sending it to the data set.
- **welcomeUi.ui & Q1.ui -** these files are the files outputted from the qt designer application.

1.8 Testing system

This folder contains the **demoWs30.py** file, this is the code file for the system we created to test our model for continues authentication.

2 User's Guide

In this section, we will explain how to run and use the models on the cloud (Google Drive) and on your own computer.

2.1 Google Drive (Recommended)

2.1.1 Upload to drive

To open the project on your own Google Drive you will need to upload the project. To do this follow the steps below -

- Go to this link.
- Click the download option as shown in figure 2

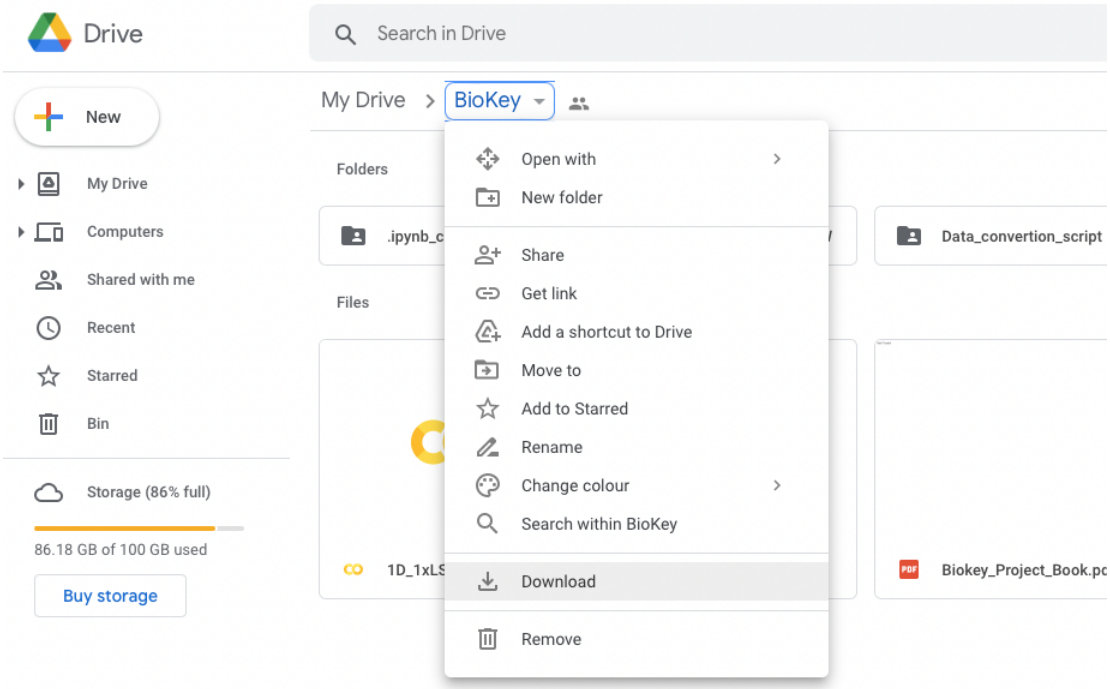
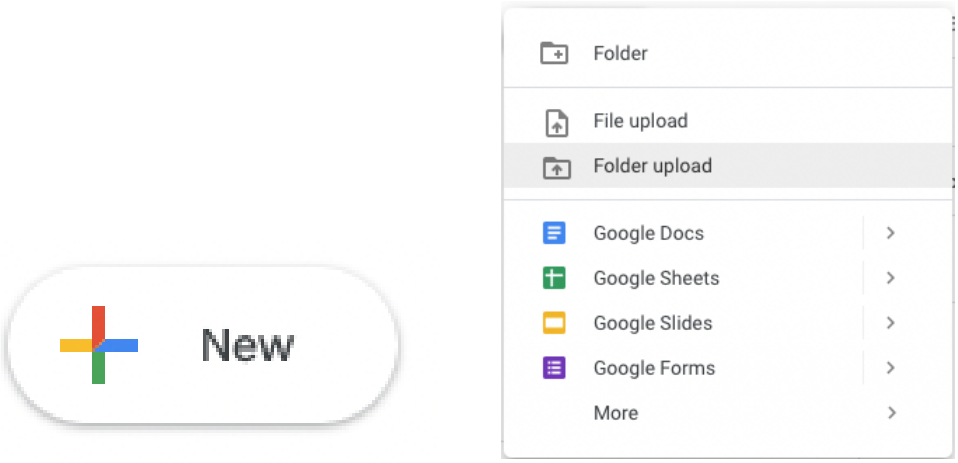


Figure 2: The download option

- Once the file is downloaded navigate to the downloaded zip file.

- Extract the files into a new folder.
- Go to your google drive and press the new button, where you will be presented with a list.
- Choose the Upload folder option look at Figure 3.



(a) New button (b) List

Figure 3

- Then navigate to the folder in which you extracted the zip into and select it to be uploaded.

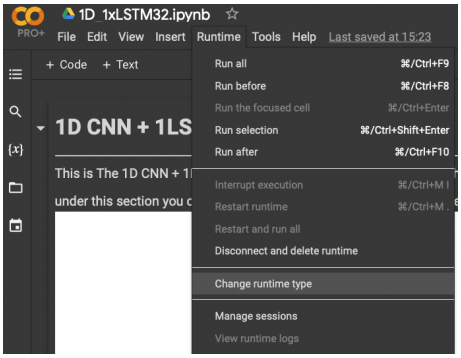
2.2 Run model files

2.2.1 Google drive

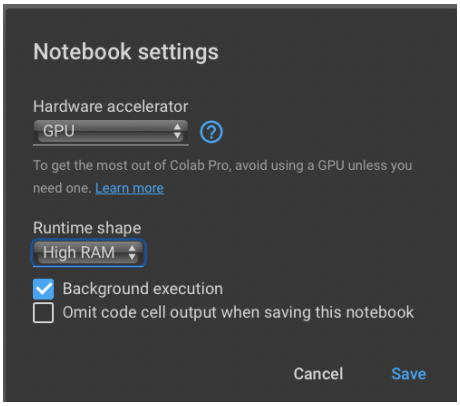
To run every file in the drive, there are some minor adjustments you will need to make in order for things to run properly.

For each model file do as following -

- Open a model file
- In the toolbar select "Runtime" and then select "Change Runtime type" as shown in figure 4
- Change the Hardware accelerator to GPU (The other settings are for Colab Pro+ users)



(a) Select The highlighted option



(b) Set to GPU

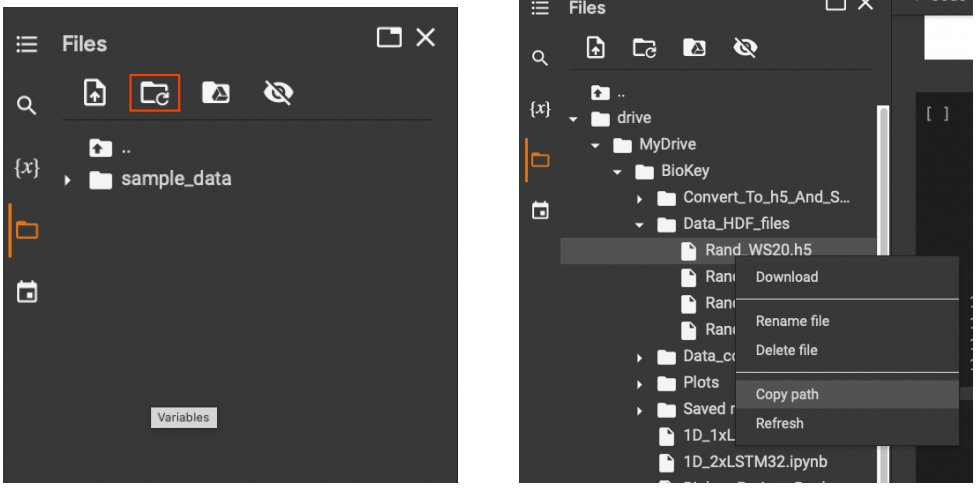
Figure 4

- Press the "Save" button.
- Run the 1D CNN + 1LSTM section that includes the following code

```
from google.colab import drive
drive.mount('/content/drive')
```

  and allow access to drive when prompted.

- When the cell finishes execution and you are allowed access to the drive, you will need to open the side menu by clicking the folder icon. If a folder by the name drive doesn't appear, refresh the folder menu by clicking the folder icon marked by a red square as seen in Figure 5(a)
- inside the first section of the notebook (1D CNN + 1LSTM) in the second block where the variable **input_shape** is set the desired window size. For example, if the desired window size is 40, set the input_shape = (40, 6), where the input_shape[0] is the window size.
- Navigate to the folder you uploaded to your drive that contains the project folder. as seen in figure 5(b)
- Inside the folder, select the **"Data_HDF_files"** folder and copy the path to the relevant window size data file by right-clicking the file and selecting the copy path from the dropout menu. **Note if the input shape wasn't changed, the model won't run**.



(a) Colab side menu (b) The path to the data files

Figure 5

- At the Data loading section, copy and paste the new path at the first code section. As seen below in the first row. The path should be a String type, so be sure you wrap it with ".

```
with h5py.File('/content/drive/MyDrive/BioKey/Data_HDF_files/Rand_WS50.h5','r') as hdf:
  data = hdf.get('train_data')
  train_data = np.array(data)
  data = hdf.get('train_labels')
  train_labels = np.array(data)
```

- Go to the "Runtime" section at the menu bar and select Run all. The menu bar just as in 4(a) and select the "Run all" option that will start running and training the model.

2.2.2 Run models Locally

This section is for users who run the project on a computer. **Note if you don't have CUDnn drivers installed on your PC, the model won't run as well if you don't have an Nvidia GPU.**.
Make sure you have the TensorFlow library installed. If not, check the installation guide at this link.

- Go to this link
- Click the download option as shown in figure 2
- Once the file is downloaded, navigate to the downloaded zip file.

- Extract the files into a new folder.
- Open a model file located in the extracted folder in an IDE that supports Jupiter notebooks, for example, VScode.
- Inside the first section of the notebook (1D CNN + 1LSTM) in the second block where the variable **input_shape** is set the desired window size. For example, if the desired window size is 40, set the input_shape = (40, 6), where the input_shape[0] is the window size.
- Navigate to the folder you extracted to your computer that contains the project.
- Inside the folder, select the **"Data_HDF_files"** folder and copy the path to the relevant window size data file. **Note if the input shape wasn't changed, the model won't run**.
- At the Data loading section, copy and paste the new path at the first code section. As seen below in the first row. The path should be a String type, so be sure you wrap it with ".

```
with h5py.File('/content/drive/MyDrive/BioKey/Data_HDF_files/Rand_WS50.h5','r') as hdf:
    data = hdf.get('train_data')
    train_data = np.array(data)
    data = hdf.get('train_labels')
    train_labels = np.array(data)
```

- Then Run all The notebook. That will start running and training the model.

2.3 Run the DataToHd5_SlidingWindow script

2.3.1 Google drive

- Go to the **DataToHd5_SlidingWindow.ipynb** file that can be found in the **Convert_To_h5_And_SW** folder.
- Once it is open, run the second cell containing the code that mounts the drive folder and allow access to the drive when prompted.
- When the cell will finish execution, and you are allowed access to the drive, you will need to open the side menu by clicking the folder icon. If a folder by the name drive doesn't appear, refresh the folder menu by clicking the folder icon marked by a red square as seen in Figure 5(a)
- Navigate to the **Data_convertion_script** folder and copy the path to the converted data folders by right-clicking the file and selecting copy path from the dropout menu. as presented in Figure 6

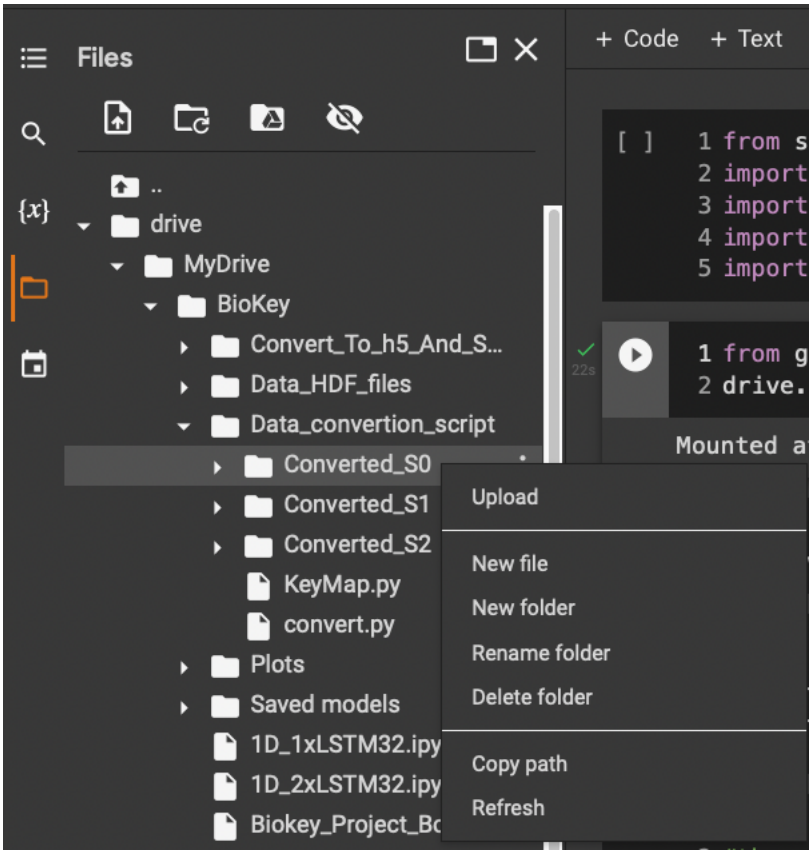


Figure 6: Copying the path to the converted data folder

- paste the paths of all 3 converted folders to the first cell of the "Main" block replacing the existing path (in bold below).

  dataTrans("**/content/drive/MyDrive/BioKey/Data_convertion_script /Converted_S0/Converted_baseline**")

- Go to the "Runtime" section at the menu bar and select Run all. The menu bar just as in 4(a) and select the "Run all" option that will start running and training the model.

2.3.2 Local

- Go to the **DataToHd5_SlidingWindow.py** file that can be found in the **Convert_To_h5_And_SW** folder.
- Navigate to the **Data_convertion_script** folder and copy the path to the converted data folders.

- paste the paths of all 3 converted folders to the code lines 71-73 replacing the existing path (in bold below).

  dataTrans("**/content/drive/MyDrive/BioKey/Data_convertion_script /Converted_S0/Converted_baseline**")

- Then start running the script in your IDE.

2.3.3 Dan_hdf5.py

- Go to the **Dan_hdf5.py** file that can be found in the **Convert_To_h5_And_SW** folder.
- Navigate to the **Data_convertion_script** folder and copy the path to the converted data folders.
- paste the paths of all 3 converted folders to the code lines 67-69 replacing the existing path (in bold below).

  dataTrans("**/content/drive/MyDrive/BioKey/Data_convertion_script /Converted_S0/Converted_baseline**")

- Navigate to the **Data_convertion_script** folder and copy the path to the Dan data folder.
- paste the path of the "Dan data" folder to the code line 72 replacing the existing path (in bold below).

  dp = **'/Volumes/GoogleDrive-112371676829911023923/ My Drive/BioKey/Data_convertion_script/Dan Data'**

- Then start running the script in your IDE.

2.4 Run Plots script

2.4.1 Google drive

- Go to the **Plots.ipynb** file that can be found in the **BioKey** folder.
- Once it is open, run the last cell under the first section of the notebook, containing the code that mounts the drive folder and allow access to the drive when prompted.
- When the cell will finish execution, and you allowed access to drive, you will need to open the side menu by clicking the folder icon. If a folder by the name drive doesn't appear, refresh the folder menu by clicking the folder icon marked by a red square as seen in Figure 5(a)
- Navigate to the **"Data_HDF_files"** folder and copy the path to it by right-clicking the file and selecting copy path from the dropout menu. as presented in Figure 6 where instead of copying the file path, you will copy the folder path.

- Replace the path assigned to the dirhdf variable that can be found under the "Main" section to the path you copied.
  dirhdf = **'/content/drive/MyDrive/BioKey/Data_HDF_files'**
- Navigate to the **"Saved models"** folder and copy the path to it by right-clicking the file and selecting copy path from the dropout menu. as presented in Figure 6 where instead of copying the file path, you will copy the folder path.
- Replace the path assigned to the dir variable that can be found under the "Main" section to the path you copied.

- Go to the "Runtime" section at the menu bar and select Run all. The menu bar just as in 4(a) and select the "Run all" option that will start running the script.

2.4.2 Local use

- Go to the **Plots.ipynb** file that can be found in the **BioKey** folder.
- Navigate to the **"Data_HDF_files"** folder and copy the path to it.
- Replace the path assigned to the dirhdf variable that can be found under "Main" section to the path you copied(in bold).
  dirhdf = **'/content/drive/MyDrive/BioKey/Data_HDF_files'**
- Navigate to the **"Saved models"** folder and copy the path to it.
- Replace the path assigned to the dir variable that can be found under "Main" section to the path you copied (in bold).
  dir = **'/content/drive/MyDrive/BioKey/Saved models'**
- Start running the notebook in your IDE that will start running the script.

2.5 Data conversion script

This folder contains the script that converts the buffalo dataset format into the format we applied in our research.
Note the script can't work unless you possess the Buffalo dataset. To get the dataset to check the instructions in the link. The folder contains 3 other folder which contains the converted data resulting from the conversion script.

If you do posses the dataset do the following steps -

- Go to the **convert.py** file that can be found in the **Data_convertion_script** folder.
- Replace the path assigned to the a_directory variable to the path of the buffalo dataset session you want to convert(in bold).

  a_directory = **"/Users/rafaelelkoby/Desktop/UB_keystroke_dataset/s2/rotation"**

- Replace the path assigned to the out_directory variable to the path of the folder you want the output of the script to go to(in bold).

  out_directory = **'/content/drive/MyDrive/BioKey/Saved models'**

- Start running the Python file in your IDE that will start running the script.

2.6 BioKey_Dataset_Collector_ID

The instructions given here are for the database instance created by us. This means that the EXE file may not work when you open this program because the dataset may already be closed. In that case, run it using an IDE and add your database instance to the database.py file. We use MySQL as the database query engine. and the scheme looks like in table 1. The input column was of type "longtext".

email	input
123456	[(0.23,0.1,0.032,0.3,0.23,0.03,0.09).....)]
851358	[(0.12,0.1,0.0612,0.26,0.34,0.02,0.019)...)]

Table 1: The scheme of the database

2.6.1 Runing EXE file

This is the file that executes the program. Just open it and start using it. More instructions about the program can be found below.

2.6.2 Runing inside IDE

This folder contains the files related to the program we used to collect the dataset. **Note** that this file can only run **localy** and wont run on google Colab.

To run this program, some addons are required to install the make sure you have Python installed and then run the following pip installs -

- For UI we use - pip install PyQt6
- For keyboard monitoring we use - pip install pynput==1.7.6

After installing both libraries, you should be ready to run this program. The only step is to open the welcomeUI.py file and run it in your IDE.

2.6.3 Using the program

- Open the program using one of the methods mentiond above.
- Your ID number needs to be entered in the text line marked in green in figure 7.

- Click the Next button marked in blue in figure7.

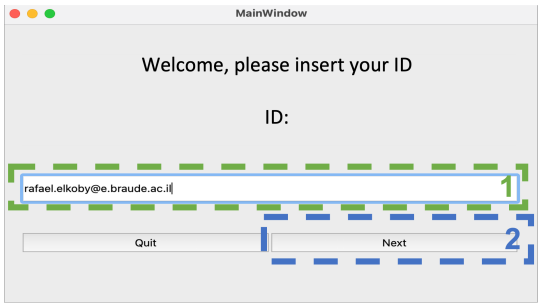


Figure 7: login window

- In the area marked in red, write your answer in English in figure 8.
- When you have finished writing the answer, click on the Next button, which is marked in blue in figure 8.
- After the fifth question, the software will close on its own.

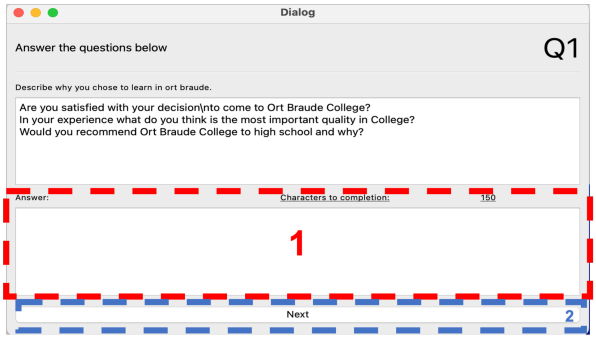


Figure 8: question window

2.7 Testing system

2.7.1 demoWs30 - UI

The **demoWs30.py** file can be found inside the **Testing system** folder. This file can run only locally. That is because of the lack of support for the pynput library that we use to monitor keystrokes.

To run this on your computer please make sure you have the pynput and PyQT5 libraries installed. If you don't have the libraries do the following pip installs -

- For UI we use - pip install PyQt5

- For keyboard monitoring we use - pip install pynput==1.7.6

After installing the libraries

- Locate the BioKey project folder that you downloaded
- open the **demoWs30.py** in your IDE and change the path in line 310

  ```
  310 model = keras.models.load_model(r'C:\Users\Dan Gutchin

  \Desktop\final_produi\1D_2XLSTM32_1x1_Dan_150epochs_WS30.h5'})
  ```

  To the path to the **1D_2XLSTM32_1x1_Dan_150epochs_WS30.h5** file which is located in the **Testing system** folder inside the main **BioKey** folder you located on your PC.
- Run the file inside your IDE

After ruining the Testing UI will appear as in figure 9.

- To start a typing session, press the green "Start" button
- Start typing in the left text box, after 30 characters the system will start outputting predictions on the right text box.
- To end a session, press the "esc" key and close the program.

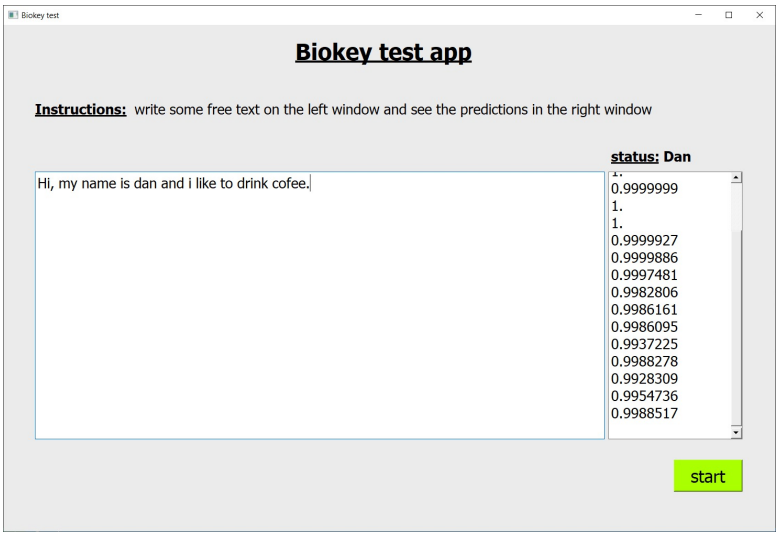


Figure 9: Model testing system UI

2.7.2 FAR_FRR.py

This file is meant to run only locally. This file calculates and plots the FRR, RAF, and EER values from a given directory containing the predictions in a .txt file.
To run the file, change the value of "a_directory" the code (line 7) to the path to the folder of the **Experiment results** which can be found inside the **BioKey/Testing system** folder.