**Diagnostic Tool Manual**

*Last Revised: 27 Oct 2022*

[**01 Introduction**](#_dytmzdqsx9hz) **2**

[**02 Installation**](#_nnr922ysx8p0) **2**

[**03 Getting Started**](#_fdm8ep6w0ajf) **2**

[3.1 Input File Requirements](#_1oxbn6d0rttm) 2

[3.2 Customizable Files](#_cmz76t8x05a9) 2

[3.3 Procedures: Code Execution](#_vlbxefru69n6) 2

[3.4 Output Interpretation](#_glt4kzhtec4j) 6

[3.4.1 Procedures](#_8bif6hbbeedv) 6

[3.4.2 Understanding Output File](#_fu4xuyg6x0dh) 6

# 

# **01 Introduction**

This tool is meant to improve accuracy and consistency of the Machine Learning classifier used internally by giving a second opinion of labels on the same text data.

# **02 Installation**

*DiagnosticTool.ipynb*

No software is required, login into Google Drive to use it. This tool is a Google Colaboratory file run on Google Drive service. The advantage of doing so is the drastic increase in speed that the free GPU provided by Google services as compared to reserving computing power on local machines for the task.

# **03 Getting Started**

## 3.1 Input File Requirements

The input files for this tool originates from the text data collected internally and should be uploaded into the *assets* folder.

* File #1: labeled data set - requires the “text” and “label” columns.
* File #2: output of BERT classifier - requires all columns output from the internal BERT tool, critical columns include the “text”, “comment\_pred\_prob” and the “label”.

## 3.2 Customizable Files

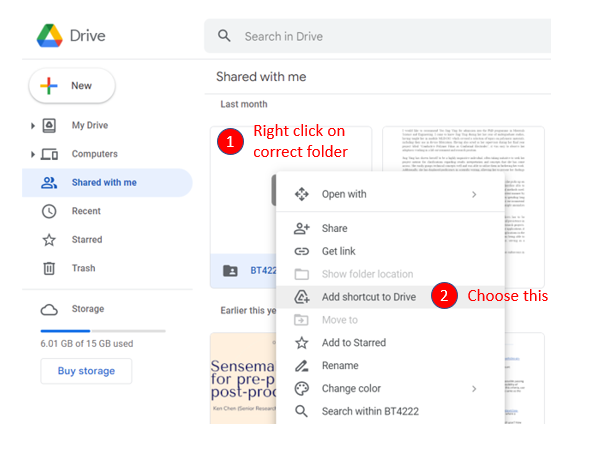
* File #3: *emoji.xlsx*
* File #4: *emoticons.xlsx*
* File #5: *nonenglish.xlsx*

Files #3-5 would be given files that already reside in the *assets* folder in the root directory. Users can further populate those rule sets as laid out in the *Preprocessing Manual.* These files provide context for emojis, emoticons and non-english words used in text collected so that the Machine Learning tool can better understand the text and label texts more accurately.

## 3.3 Procedures: Code Execution

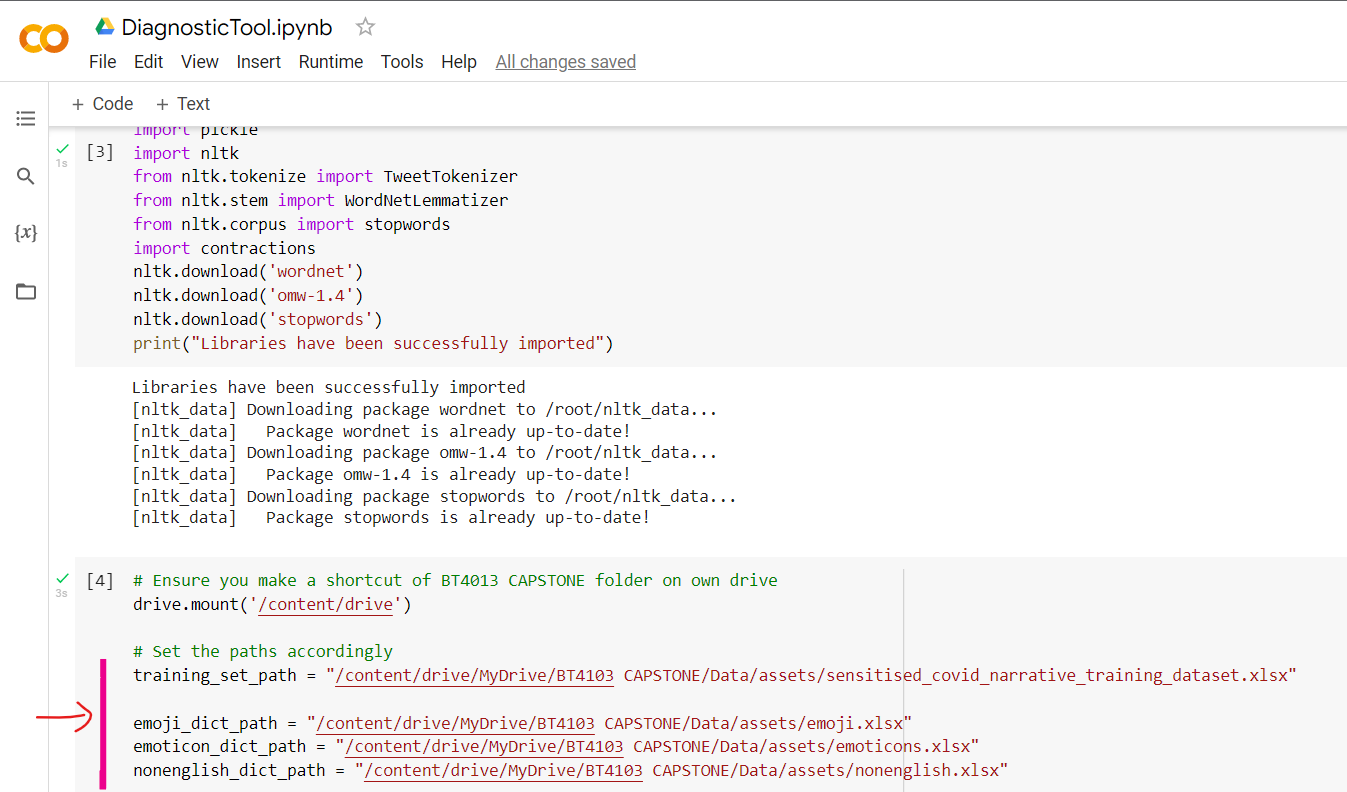
Step 1. Upload files #1-2 into assets folder

Step 2. Make a copy of entire folder to own drive



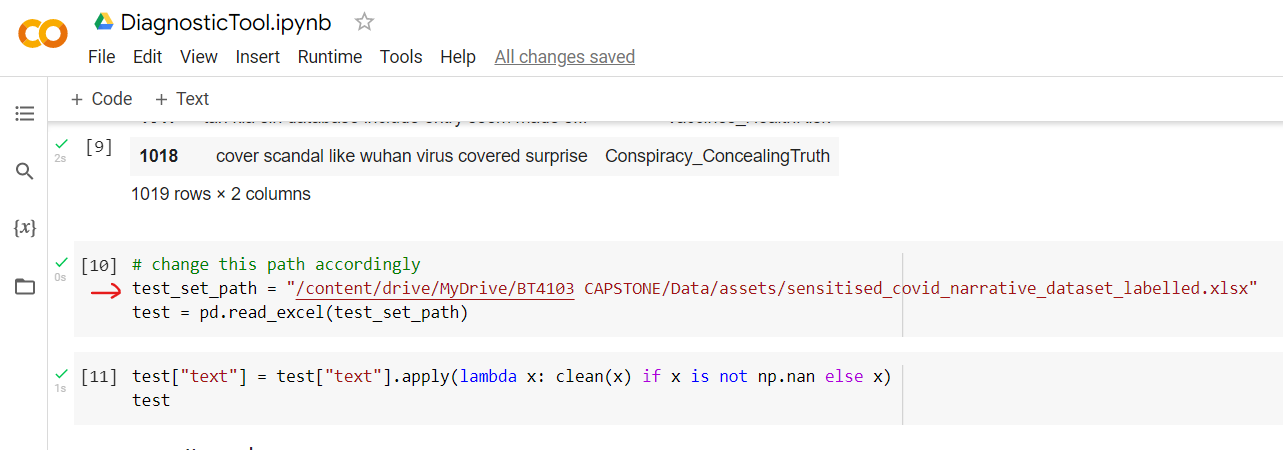
*Creating a shortcut*

Step 3. In code block #4, change the paths accordingly to the input files of #1, #3, #4, #5



*Paths to be changed in code block #4*

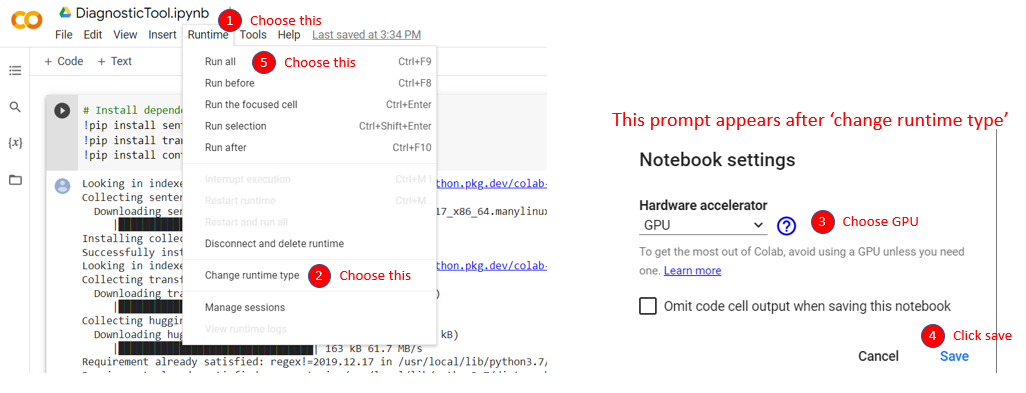
Step 4. In code block #10, change the file path accordingly to the input file #2



*Path to be changed in code block #10*

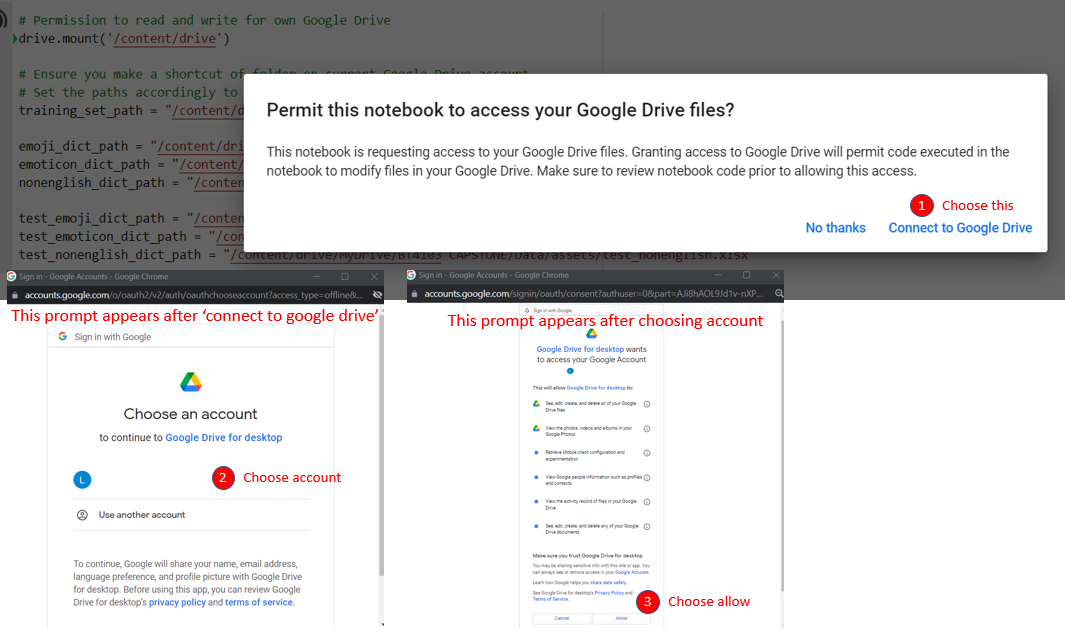
Step 5. Change to a GPU runtime in the ribbon under the *Runtime > Change runtime type* button

Step 6. After successfully connecting to a GPU, click on *Runtime > Run All*



*Connecting to GPU and running code*

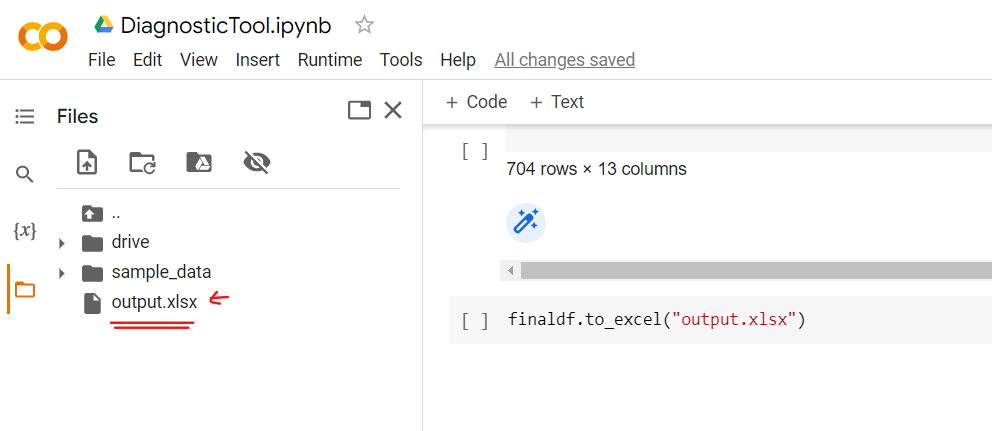
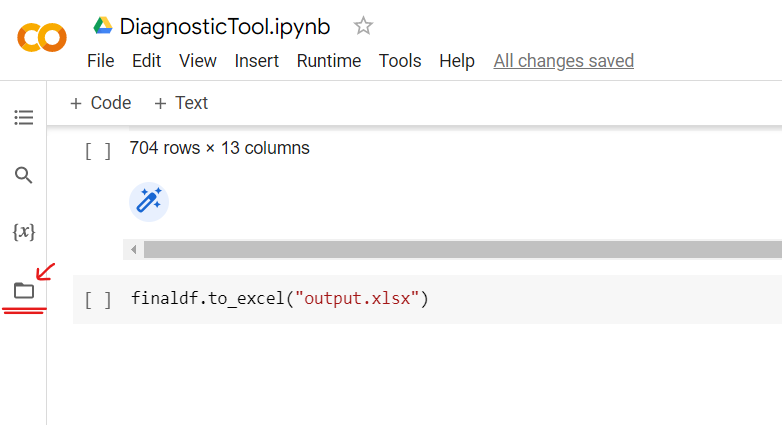
Step 7. When code executes code block #4, there would be a window pop-up asking for permission to allow Colab to read and write files to your Google Drive, choose your account and allow the necessary permissions

****

*Connecting to Drive and allowing permissions*

*FYI:* All code blocks will execute sequentially and the result file will be located in the runtime files after the final code block is executed. Code block #24 is the slowest step of all the blocks, there will be a progress bar of each epoch being trained in the model if user requires a gauge of time taken

Step 8. To access the output file, go to *Files* under the left pane the icon is a folder, *output.xlsx* would reside here. Double click the file to download.



*Location of output.xlsx*

## 3.4 Output Interpretation

*Disclaimer: The usefulness of the output is highly dependent on the quality of labels within the training data provided to both models, and would also be affected by class imbalances*

### 3.4.1 Procedures

1. Find the file in the downloads folder after following Step 8 of Section 3.3.

2. Users are advised to pick the **first 200, 500 or 5% of rows of data** depending on the size of the test dataset.

3. Read the text provided in column ‘text’, manually label these rows, labels provided by both models (columns ‘label’ and ‘xlnet\_label’) only serve as a suggestion.

4. The row number corresponding to input file #2 is in column B.

5. Cut these text from input file #2 and paste it in the labeled data file, input file #1, along with the manually labeled label.

5a. Remember to remove the entire row from input file #2.

6. Re-run the internal model to obtain labels for further deliberation on whether the f1 score is satisfactory.

6a. If so, labels generated by the internal model can be directly used for analysis.

6b. Otherwise, iterate through the diagnostic tool process again to manually label more rows.

### 3.4.2 Understanding Output File

As compared to File #2, there are 2 additional columns, xlnet\_prob and xlnet\_label. These are the probability and the corresponding most confident label as predicted by the diagnostic tool. The other key columns to look at would be the text column which contains the original post and the comment\_pred\_prob, label columns which are the probability and corresponding most confident label as predicted by the internal model.

The data is ordered as such, lowest 10 percentile of probability by internal model at the top, followed by difference in probability found by the diagnostic tool, and ending off with rows with the same label predicted by both models.

The first category of rows up to 10th percentile suggests the BERT model is highly unsure of its prediction which requires more training for these types of texts. The second category of rows with high difference in probability suggests deviation between the models which also requires more training to improve the internal machine learning model.