Configuration Manual

MSc Research Project
MSc Data Analytics

Karthikeya Anusury
Student ID: 22217096

School of Computing
National College of Ireland

Supervisor: Abid Yaqoob

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……. Karthikeya Anusury……………………………………………………………………….

Student ID: ………22217096……………………………………………………………………………….……..

Programme: ……… MSc Data Analytics …………………………… **Year:** ……2023-2024…

Module: ……… MSc Research Project ……………………………………………….………………

Lecturer: ………… Abid Yaqoob …………………………………………………………………….…………

Submission Due Date: ……16-09-24……………………………………………………………………………….………

Project Title: RoBERTa-Based NLP System for Enhanced Disease Prediction from Symptom Descriptions

Word Count: …………877………………… **Page Count:** ……………………6……………..……..………

I hereby certify that the information contained in this (my submission) is information about research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: …………Karthikeya Anusury…………………………………………………………

Date: …………16-09-24…………………………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Karthikeya Anusury
Student ID: 22217096

1. Manual for running RoBERTa and BERT models in Google colab

1.1 Google colab

- Navigate to Google Colab and select the option to upload a new notebook. Once selected, upload the “MSc_Research_Project_Disease_Prediction_Demo.ipynb” notebook file and connect to a runtime session to start working

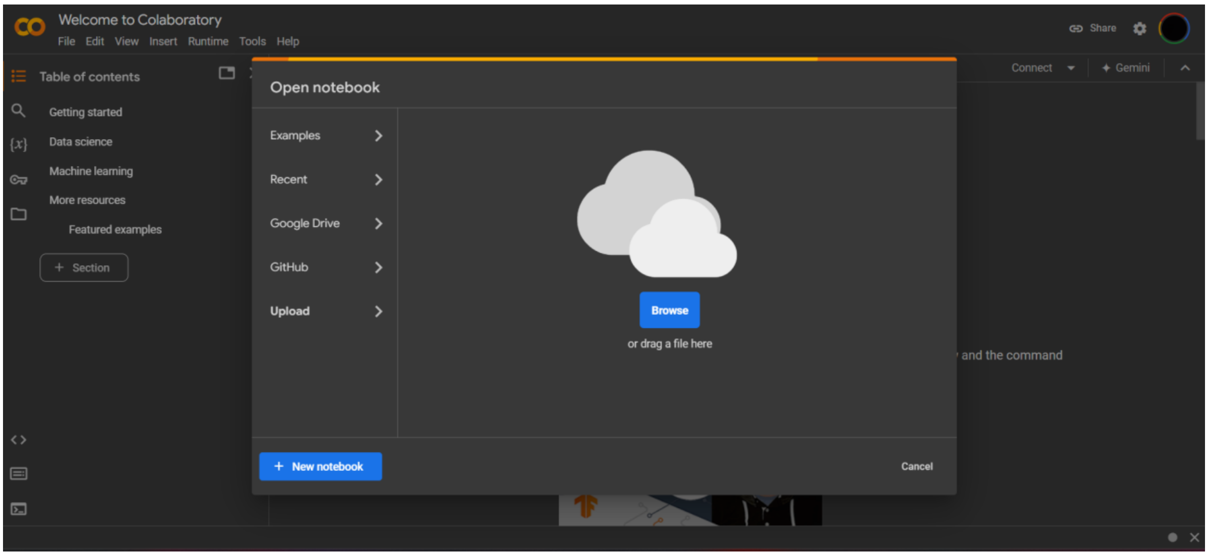
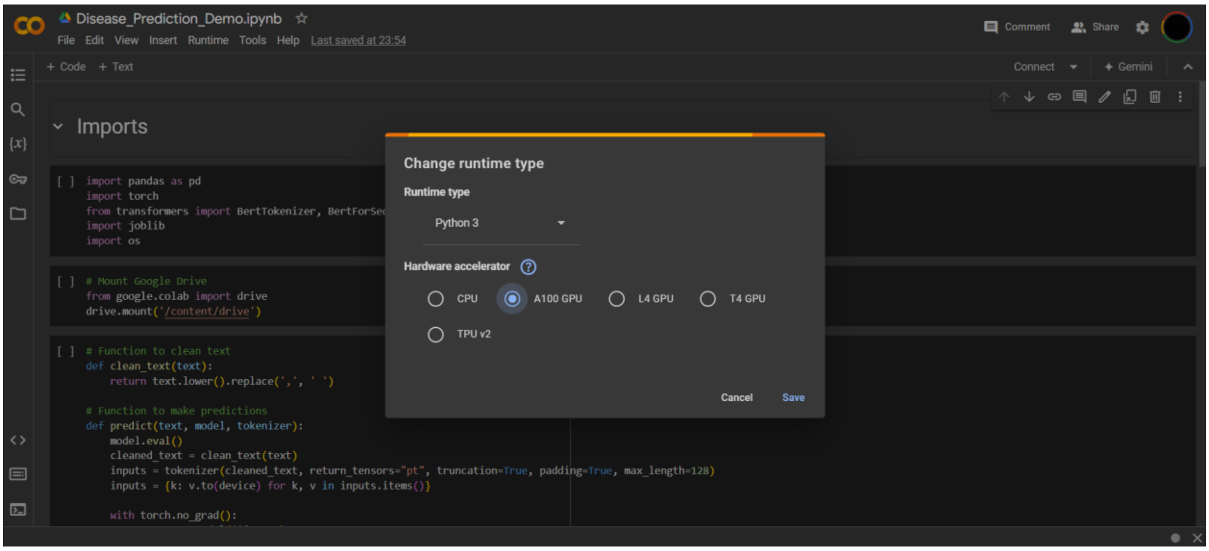


Figure 1 Uploading the notebook


Disease_Prediction_Demo.ipynb
File Edit View Insert Runtime Tools Help Last saved at 23:54
Comment
Share
+ Code + Text
Connect
Gemini
Imports
Cancel
Save

Change runtime type

Runtime type: Python 3

Hardware accelerator

○ CPU ● A100 GPU ○ L4 GPU ○ T4 GPU
○ TPU v2

```
[ ] import pandas as pd
    import torch
    from transformers import BertTokenizer, BertForSe
    import joblib
    import os
```

```
[ ] # Mount Google Drive
    from google.colab import drive
    drive.mount('/content/drive')
```

```
[ ] # Function to clean text
    def clean_text(text):
        return text.lower().replace(',', ' ')

    # Function to make predictions
    def predict(text, model, tokenizer):
        model.eval()
        cleaned_text = clean_text(text)
        inputs = tokenizer(cleaned_text, return_tensors="pt", truncation=True, padding=True, max_length=128)
        inputs = {k: v.to(device) for k, v in inputs.items()}

        with torch.no_grad():
```

Figure 2 Select run time

1.2 Set up Google Drive folder structure and upload files

- Go to drive.google.com and sign in to your Google account.
- Create a folder structure as follows:
  - Create a main folder named NLP_Models (Base Path)
  - Inside NLP_Models, create two subfolders: RoBERTa and BERT
  - Inside each of these (RoBERTa and BERT), create two more subfolders: Preprocessing and Model
- Directory should look like below

```
NLP_Models/
├── RoBERTa/
│   ├── Preprocessing/
│   └── Model/
└── BERT/
    ├── Preprocessing/
    └── Model/
```

1.3 Upload the files as follows:

The required model and preprocessing files for both models are in the Google Drive path below. The path can be accessed without any restrictions.

Path:
https://drive.google.com/drive/folders/1yDVD3UulC2PJrvwVmIlLDTWcs2RSVg7a?usp=sharing

1.3.1 For RoBERTa:

- In NLP_Models/RoBERTa/Preprocessing/, upload the “RoBERTa_preprocessing” folder from the above shared link.

- In NLP_Models/RoBERTa/Model/, upload the “RoBERTa_models” folder from the above shared link

1.3.2 For BERT:

- In NLP_Models/BERT/Preprocessing/, upload the “BERT_preprocessing” folder from the above shared link.

- In NLP_Models/BERT/Model/, upload the “BERT_models” folder from the above shared link

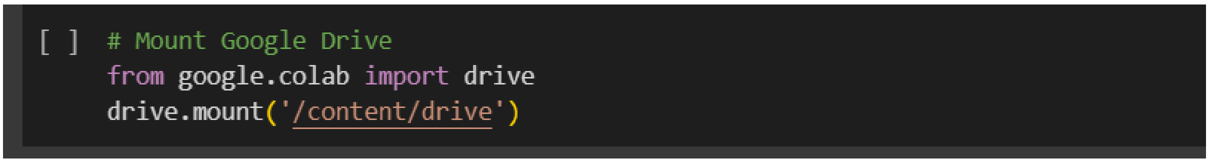
1.3.3 Direct Google Drive link to the above NLP_Models

- If facing any issues by following the above procedure, use the below Google Drive link to directly copy the NLP_Models folder for testing.

  Link to NLP_Models folder for testing:
  https://drive.google.com/drive/folders/1RVzGHyZoWIc7lxcztIkjQ22xxlxwNSQ_?usp=sharing

1.4 Mount Drive

Run the following cell to mount the Google Drive.

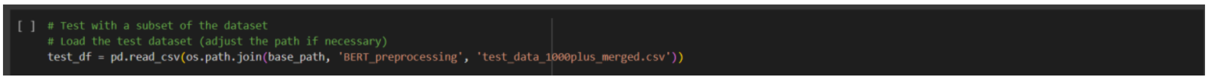


```
[ ] # Mount Google Drive
    from google.colab import drive
    drive.mount('/content/drive')
```

Figure 3 Mount Google Drive

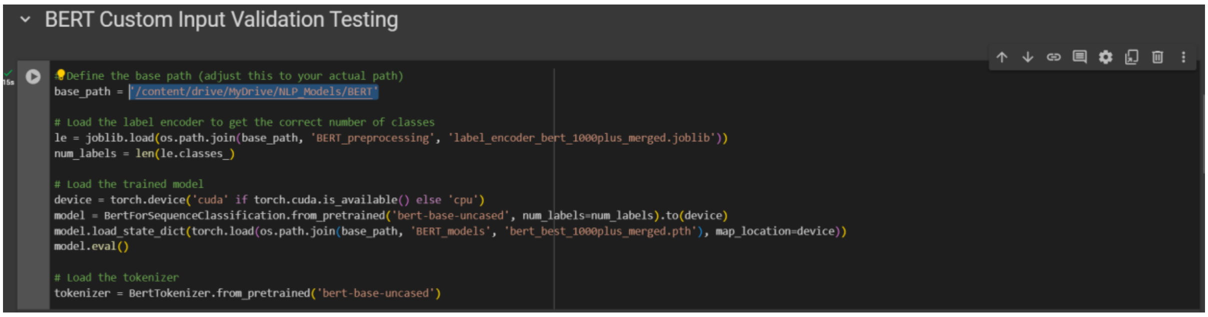
2. Adjust the file paths in the code

- The base path and file paths will be updated to point to the uploaded files in Section 1.3 for both models.



```
[ ] # Test with a subset of the dataset
    # Load the test dataset (adjust the path if necessary)
    test_df = pd.read_csv(os.path.join(base_path, 'BERT_preprocessing', 'test_data_1000plus_merged.csv'))
```

Figure 4 Dataset path



BERT Custom Input Validation Testing

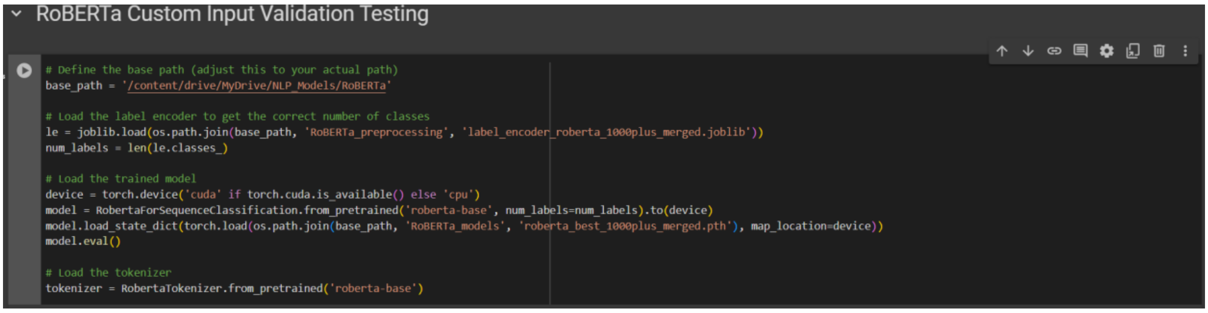
```
#Define the base path (adjust this to your actual path)
base_path = '/content/drive/MyDrive/NLP_Models/BERT'

# Load the label encoder to get the correct number of classes
le = joblib.load(os.path.join(base_path, 'BERT_preprocessing', 'label_encoder_bert_1000plus_merged.joblib'))
num_labels = len(le.classes_)

# Load the trained model
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model = BertForSequenceClassification.from_pretrained('bert-base-uncased', num_labels=num_labels).to(device)
model.load_state_dict(torch.load(os.path.join(base_path, 'BERT_models', 'bert_best_1000plus_merged.pth'), map_location=device))
model.eval()

# Load the tokenizer
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
```

Figure 5 BERT Model paths



RoBERTa Custom Input Validation Testing

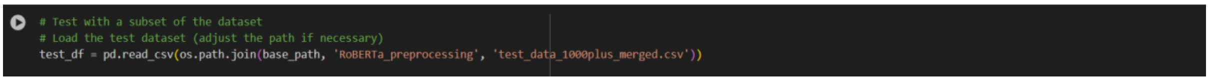
```
# Define the base path (adjust this to your actual path)
base_path = '/content/drive/MyDrive/NLP_Models/RoBERTa'

# Load the label encoder to get the correct number of classes
le = joblib.load(os.path.join(base_path, 'RoBERTa_preprocessing', 'label_encoder_roberta_1000plus_merged.joblib'))
num_labels = len(le.classes_)

# Load the trained model
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model = RobertaForSequenceClassification.from_pretrained('roberta-base', num_labels=num_labels).to(device)
model.load_state_dict(torch.load(os.path.join(base_path, 'RoBERTa_models', 'roberta_best_1000plus_merged.pth'), map_location=device))
model.eval()

# Load the tokenizer
tokenizer = RobertaTokenizer.from_pretrained('roberta-base')
```

Figure 6 for RoBERTa Model Paths



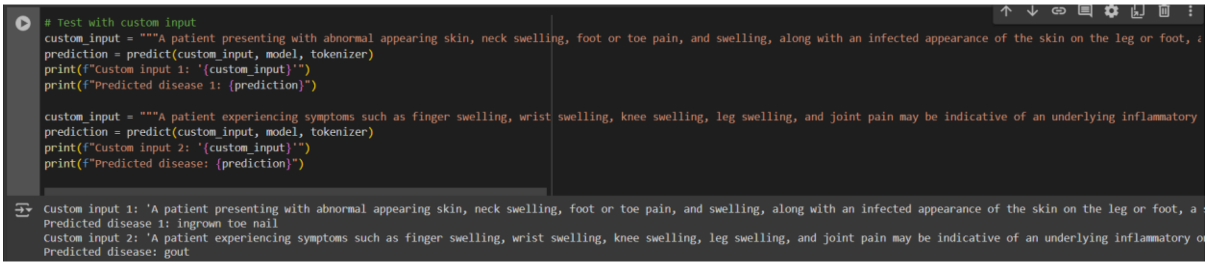
```
# Test with a subset of the dataset
# Load the test dataset (adjust the path if necessary)
test_df = pd.read_csv(os.path.join(base_path, 'RoBERTa_preprocessing', 'test_data_1000plus_merged.csv'))
```

Figure 7 Dataset path for the RoBERTa

3. Inference

After setting up your Google Drive and adjusting the file paths, follow these steps to run the inference code for both the BERT and RoBERTa models. The Models are tested using a subset of the test dataset and by providing a custom input where symptoms are described in a paragraph. The custom inputs can be generated by using symptoms of any of the diseases from the dataset. For example, custom input 1 in the demo code is created by using symptoms of one of the diseases “ingrown toe nail”. Once the setup is completed, start running the demo, and additional inputs can be generated to test these models. Below are the responses and predictions after running the code:

Run the Demo block of BERT: It will ask for custom input and as well as run the inference on 5 rows on the test data.

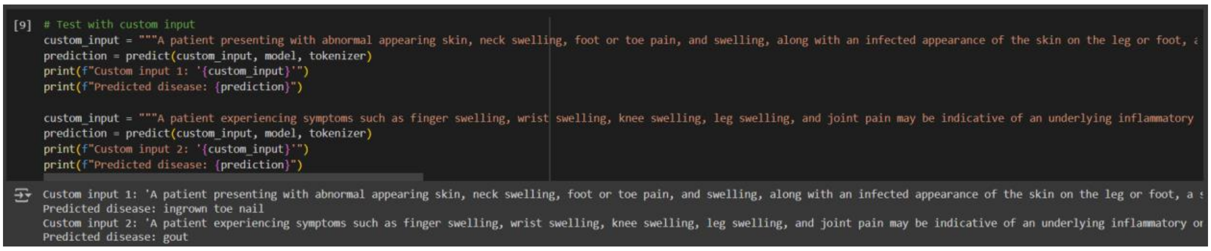


```
# Test with custom input
custom_input = """A patient presenting with abnormal appearing skin, neck swelling, foot or toe pain, and swelling, along with an infected appearance of the skin on the leg or foot, a
prediction = predict(custom_input, model, tokenizer)
print(f"Custom input 1: '{custom_input}'")
print(f"Predicted disease 1: {prediction}")

custom_input = """A patient experiencing symptoms such as finger swelling, wrist swelling, knee swelling, leg swelling, and joint pain may be indicative of an underlying inflammatory
prediction = predict(custom_input, model, tokenizer)
print(f"Custom input 2: '{custom_input}'")
print(f"Predicted disease: {prediction}")
```

```
Custom input 1: 'A patient presenting with abnormal appearing skin, neck swelling, foot or toe pain, and swelling, along with an infected appearance of the skin on the leg or foot, a
Predicted disease 1: ingrown toe nail
Custom input 2: 'A patient experiencing symptoms such as finger swelling, wrist swelling, knee swelling, leg swelling, and joint pain may be indicative of an underlying inflammatory o
Predicted disease: gout
```

Figure 8 Sample Demo for BERT using custom inputs



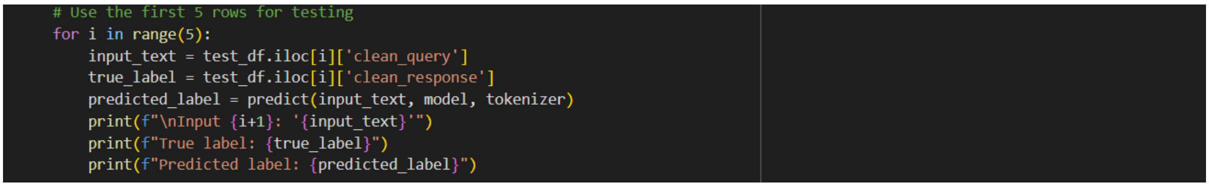
```
[9] # Test with custom input
custom_input = """A patient presenting with abnormal appearing skin, neck swelling, foot or toe pain, and swelling, along with an infected appearance of the skin on the leg or foot, a
prediction = predict(custom_input, model, tokenizer)
print(f"Custom input 1: '{custom_input}'")
print(f"Predicted disease: {prediction}")

custom_input = """A patient experiencing symptoms such as finger swelling, wrist swelling, knee swelling, leg swelling, and joint pain may be indicative of an underlying inflammatory
prediction = predict(custom_input, model, tokenizer)
print(f"Custom input 2: '{custom_input}'")
print(f"Predicted disease: {prediction}")
```

```
Custom input 1: 'A patient presenting with abnormal appearing skin, neck swelling, foot or toe pain, and swelling, along with an infected appearance of the skin on the leg or foot, a
Predicted disease: ingrown toe nail
Custom input 2: 'A patient experiencing symptoms such as finger swelling, wrist swelling, knee swelling, leg swelling, and joint pain may be indicative of an underlying inflammatory or
Predicted disease: gout
```

Figure 9 Sample Demo for RoBERTa using custom inputs

Additionally, a subset of the test dataset can be used to predict the diseases. The below loop can be used for both models. The number of test samples can be increased or decreased by updating the iteration value of the loop.



```
# Use the first 5 rows for testing
for i in range(5):
    input_text = test_df.iloc[i]['clean_query']
    true_label = test_df.iloc[i]['clean_response']
    predicted_label = predict(input_text, model, tokenizer)
    print(f"\nInput {i+1}: '{input_text}'")
    print(f"True label: {true_label}")
    print(f"Predicted label: {predicted_label}")
```

Figure 10 Increasing the test data size

References

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1907.11692

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1810.04805

colab.google. (n.d.). colab.google. https://colab.google/