National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Sachin Muttappanavar
Student ID: 20144253

School of Computing
National College of Ireland

Supervisor: Dr. Rejwanul Haque

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Sachin Muttappanavar
Student ID:	20144253
Programme:	Data Analytics
Year:	2022
Module:	MSc Research Project
Supervisor:	Dr. Rejwanul Haque
Submission Due Date:	31/01/2022
Project Title:	Configuration Manual
Word Count:	481
Page Count:	6

I hereby certify that the information contained in this (my submission) is information pertaining to 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:	Sachin Muttappanavar
Date:	30th January 2022

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

Sachin Muttappanavar
20144253

1 Introduction

The goal of this documentation is to outline the configuration set up to be followed in the implementation process of research project. We have described the software and hardware prerequisites for re-creating project. It also outlines the coding process and procedure to be followed to execute the code.

2 Scraper Setup

We have used Eclipse integrated development environment to code the automation script using Java Selenium.Following tools, programing language and libraries are used for implementation of scraper:

- JavaSE 1.8v is used for the implementation of the code
- Selenium library - v3.141.59 is used for automating the web page
- Jsoup library - v1.14.3 is used for parsing the html files
- Eclipse IDE - Photon version tool is used for development activities.

Snapshot of the scraper implementation is shown in the Figure 1 To scrape the match commentary, we must execute the project's Scraper.java class, and for reports (news articles), we must execute the Reports.java method.

3 Anaconda Setup

For data processing and model development following tools, programming language and libraries are used.

- Anaconda navigator of version 2.0.3 for web based interactive computing Jupyter Notebook - v6.4.0(Figure 2) to implement.
- Python version 3.8.5 was installed on Anaconda Navigator.
- Libraries required for data processing code are pandas, numpy, nltk, os, re
- Libraries required for executing the BART model are pandas, numpy, simpletransformers, rouge, matplotlib.

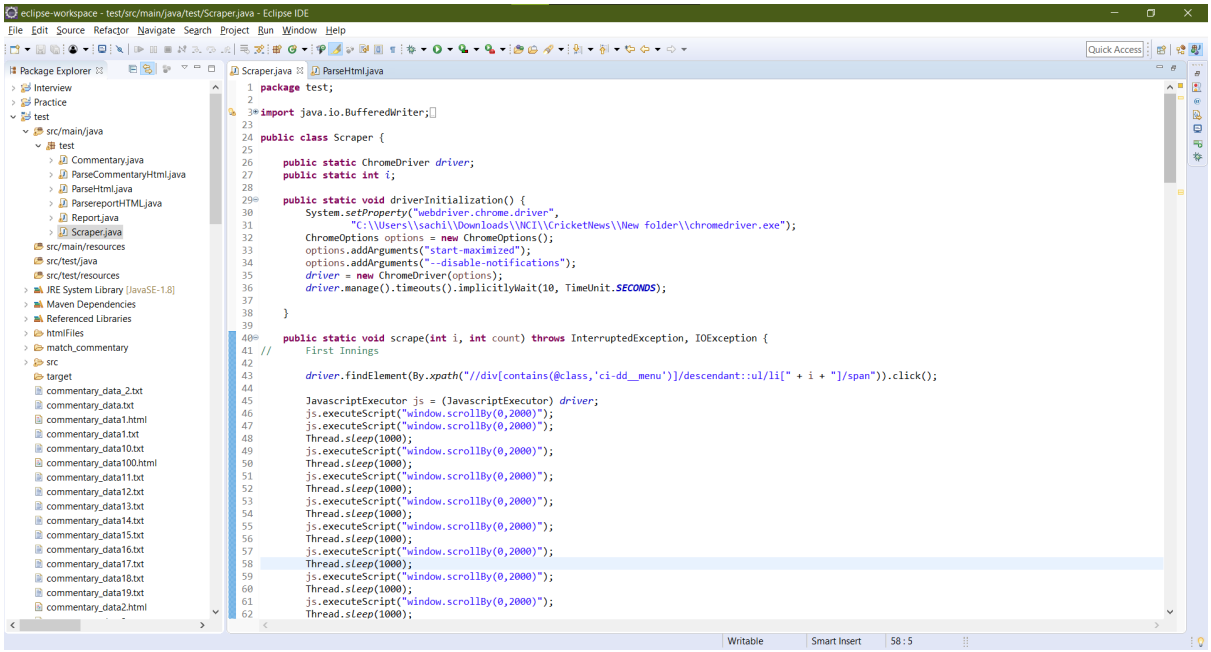

Figure 1: Scraper Implementation

- Libraries required for executing the T5base model are pandas, numpy,rouge, pytorch, sentencepiece, transformers, CUDA, tqdm, torch.
- For Named Entity Recognizeer, satndford NER jars 4.2.0v is used. To run NamedEntity_MaskingPlayers, NER jars needs to installed path of the jars needs to be updated.

4 System Configuration

For implementing the pre-trained models, we leveraged 'Google Colab'(Figure 9) of Google cloud platform. GPU runtime environment was set to execute and train the models as GPU can process numerous computations parallelly. We have used google colab GPU computational power for running executing computational intensive activities. Another advantage of utilising Google Colab is that it gives a simple method to connect to a drive where data may be saved. Local system configuration is shown in the Figure 4.

Steps to follow to connect to GPU: Runtime - Change runtime type - select GPU under hardware accelerator

5 Libraries Used

Liraries and Modules used of the data Processing shown in the Figure 6. Commentary data processing code can be found with name commentary_processing.ipynb and Reports cleaning code is in Report_Processing.ipynb. We can find the Named recognization code in NamedEntity_MaskingPlayers.ipynb. Also, libraries used in the building BART and T5base are shown in Figure 7 and Figure 5 respectively.

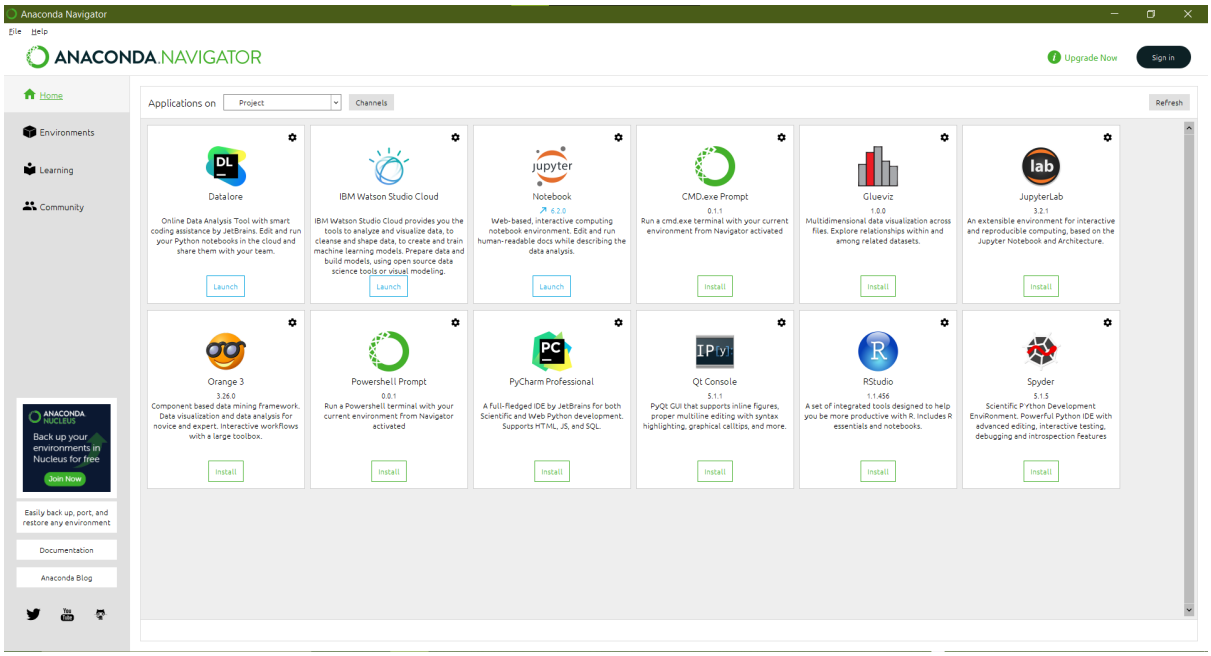

Figure 2: Anaconda Navigator

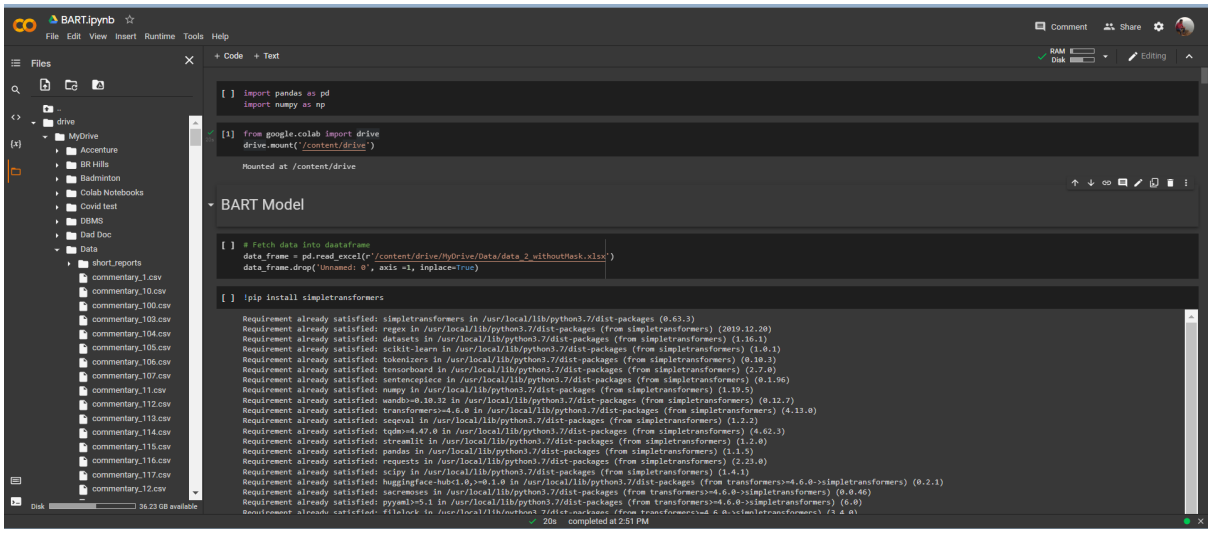

Figure 3: Google Colab

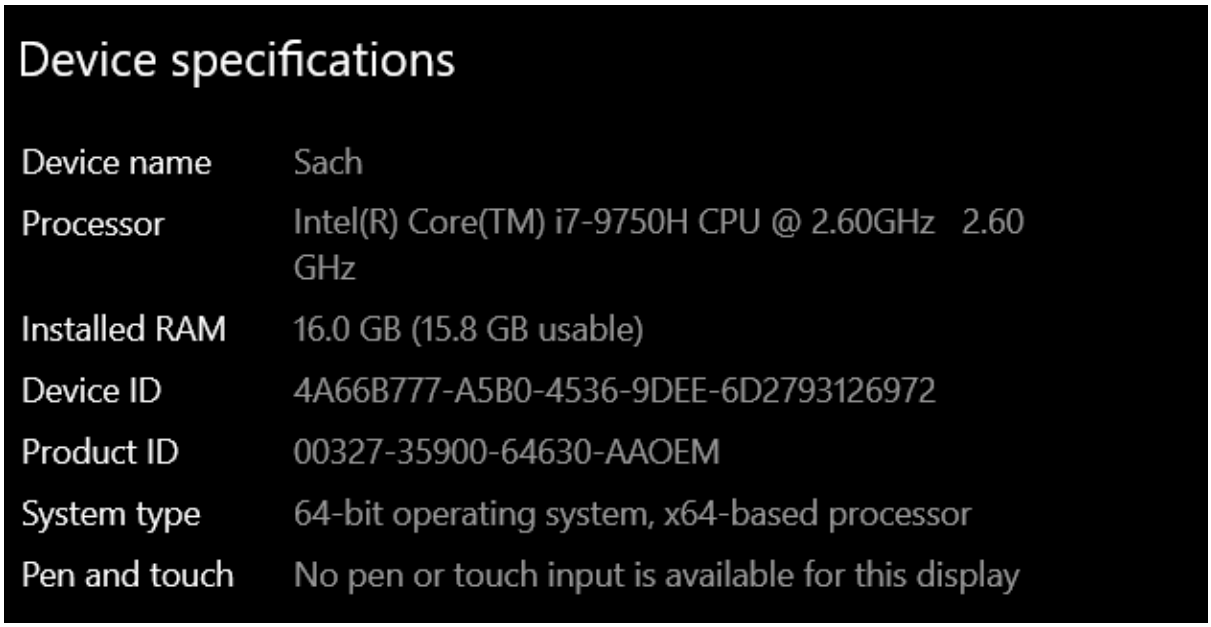

Figure 4: Local Machine Configuration

```
import pandas as pd
import numpy as np
import nltk
import re
nltk.download('stopwords')
nltk.download('punkt')
import os.path
from nltk.tag import StanfordNERTagger
from nltk.tokenize import word_tokenize
```

executed in 473ms, finished 19:06:06 2021-12-14

Figure 5: Libraries and modules used for Data Processing

```
import torch
import numpy as np
import pandas as pd
from tqdm.auto import tqdm
from transformers import (
    AdamW,
    T5ForConditionalGeneration,
    MT5ForConditionalGeneration,
    ByT5Tokenizer,
    PreTrainedTokenizer,
    T5TokenizerFast as T5Tokenizer,
    MT5TokenizerFast as MT5Tokenizer,
)
from transformers import AutoTokenizer

# from fastT5 import export_and_get_onnx_model
from torch.utils.data import Dataset, DataLoader
from transformers import AutoModelWithLMHead, AutoTokenizer
import pytorch_lightning as pl
from pytorch_lightning.loggers import TensorBoardLogger
from pytorch_lightning.callbacks import ModelCheckpoint
from pytorch_lightning.callbacks.early_stopping import EarlyStopping

torch.cuda.empty_cache()
pl.seed_everything(42)
```

Figure 6: Libraries and modules used for T5Base

```
import pandas as pd
import numpy as np
!pip install simpletransformers
from google.colab import drive
import rouge
from simpletransformers.seq2seq import Seq2SeqModel,Seq2SeqArgs
```

Figure 7: Libraries and modules used for BART

6 Models Parameters

This section provides information about the parameters defined and values assigned for them in building model. Figure 7 illustrates the BART model parameters and Figure 7 illustrates T5 base model parameters. BART model implementation is given in BART.ipynb and T5 model implementation is in T5.ipynb.

```
# Model parameters setting
model_args = Seq2SeqArgs()
model_args.num_train_epochs = 5
model_args.no_save = True
model_args.evaluate_generated_text = True
model_args.evaluate_during_training = True
model_args.evaluate_during_training_verbose = True
model_args.max_length = 1024
model_args.optimizer = 'AdamW'
model_args.use_early_stopping = True
model_args.learning_rate = 0.0001

# Initialize model
model = Seq2SeqModel(
    encoder_decoder_type="bart",
    encoder_decoder_name="facebook/bart-large",
    args=model_args,
    use_cuda=True,
)
```

Figure 8: BART model

7 Dataset

Dataset used for this research study are provided under the Dataset folder. Dataset without masking is in file data2_withoutMask.xlsx and masked data in Data_2.xlsx.

```
[ ]  # load (supports t5, mt5, byT5 models)
     model.from_pretrained("t5","t5-base")

     # train
     model.train(train_df=train_df, # pandas dataframe with 2 columns: source_text & target_text
                 eval_df=eval_df, # pandas dataframe with 2 columns: source_text & target_text
                 source_max_token_len = 512,
                 target_max_token_len = 128,
                 batch_size = 4,
                 max_epochs = 5,
                 use_gpu = True,
                 outputdir = "outputs",
                 early_stopping_patience_epochs = 0,
                 precision = 32
                 )
```

Figure 9: T5 model