

# Configuration Manual

MSc Research Project  
MSc in Cloud Computing

Mohammad Saif  
Student ID: 22248218

School of Computing  
National College of Ireland

Supervisor: Aqeel Kazmi

**National College of Ireland**  
**MSc Project Submission Sheet**  
**School of Computing**



**Student Name:** ..... Mohammad Saif.....

**Student ID:** .....22248218.....

**Programme:** .....MSc Cloud Computing..... **Year:** .....2024.....

**Module:** .....MSc Research Project.....

**Lecturer:** .....Aqeel Kazmi.....

**Submission Due Date:** .....16<sup>th</sup> September 2024.....

**Project Title:** .....Configuration manual.....

**Word Count:** .....733..... **Page Count:** .....4.....

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:** .....  .....

**Date:** .....16<sup>th</sup> September 2024.....

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

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

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

<b>Office Use Only</b>	
Signature:	
Date:	
Penalty Applied (if applicable):	

# Configuration Manual

Mohammad Saif  
x22248218

## 1 Introduction

This configuration manual contains the steps required to setup the configuration and environment for running and developing the NLP and RL models for the automated test case generation tool. The manual covers the installation of necessary software, environment setup and configuration of the project files. This guide is intended for academic students and researchers interested in reproducing the results or extending the work presented in this project.

## 2 Prerequisites

Before setting up the environment, ensure that the following prerequisite are met:

- **Operating system:** Ubuntu 20.04 LTS or Later, Windows 10, macOS Catalina or Later
- **Python Version:** Python 3.8 or later
- **IDE:** PyCharm/jupyter notebook/Anaconda installed

## 3 Libraries/Packages Used

The following Python libraries and packages are required for the NLP and RL models:

- NLP Model:
  - numpy
  - nltk
  - spacy
  - PyMuPDF
  - pandas
  - python-docx
- RL Model:
  - stable-baselines3
  - gym
  - tensorflow

- matplotlib
- seaborn

These packages are also included in requirement.txt file in the source code folder.

## 4 Configuration Steps

**For running NLP model for generating test cases:**

### Step 1: Install Python packages

First, install the required Python packages using the following command:

```
pip install -r requirements.txt
```

### Step 2: Install LibreOffice

This project relies on LibreOffice for converting .doc files to .docx format. Please follow the instructions below to install LibreOffice:

Windows:

1. Download the LibreOffice installer from the official website.
2. Run the installer and follow the on-screen instructions to install LibreOffice.
3. By default, LibreOffice will be installed in C:\Program Files\LibreOffice\program\soffice.exe. If you install it in a different location, update the libreoffice\_path in the config.txt file.

macOS:

1. Download LibreOffice from the official website.
2. Open the downloaded .dmg file and drag the LibreOffice icon into the Applications folder.
3. The default installation path is /Applications/LibreOffice.app/Contents/MacOS/soffice. If it's different, update the libreoffice\_path in the config.txt file.

### Step 3: Configure LibreOffice Path

After installing LibreOffice, ensure the correct path to the LibreOffice executable is specified in the src/config.txt file:

**Example config.txt:** libreoffice\_path=C:\Program Files\LibreOffice\program\soffice.exe

**For macOS or Linux, it might look like this:**

libreoffice\_path=/Applications/LibreOffice.app/Contents/MacOS/soffice

**Step 4: Download and prepare data**

Follow these steps to download the necessary data files from PURE dataset by Ferrari, A., Spagnolo, G. O., & Gnesi, S. (2017, September) and prepare them for processing:

1. Download the zip file

```
wget -O requirements.zip
```

```
https://zenodo.org/records/1414117/files/requirements.zip?download=1
```

2. Unzip the downloaded file

```
unzip requirements.zip
```

3. Transfer the contents of the 'req' folder to the 'data' folder

```
mv req/* data/
```

4. Delete the 'req' folder

```
rm -r req
```

(Optional) Delete the downloaded zip file to clean up

```
rm requirements.zip
```

**Step 5: Run the Project**

To process the specification files and generate a CSV output, simply run from the module's main directory:

```
python main.py
```

NOTE: If you encounter issues with LibreOffice not being found, double-check that the path in src/config.txt is correct and points to the actual location of the office executable.

**For Running RL Agent Simulations**

**Step 1: Run the experiments**

To run the experiments, you first need to install Python 3.7 and Stable Baselines2 ("2.10"). You can then go to the RL module main directory and rerun the experiment using python testCase\_prioritization/TPDRL.py by passing the following options:

-m MODE --> ranking model that can be either pairwise or listwise  
-a ALGO --> The RL algorithm to train the agent that can be either A2C, DQN  
-t TRAIN\_DATA --> Location of the training dataset  
-e EPISODES. --> the number of episodes used to train the agent  
[-d DATASET\_TYPE] --> type of the dataset that can be either simple or enriched  
[-w WIN\_SIZE] --> history window  
[-f FIRST\_CYCLE] --> a cycle from which the training will be started. Each dataset contains many cycles of regression testing, and this parameter specifies the number of cycles whose logs are to be used to train the initial agent.  
[-c CYCLE\_COUNT] --> How frequently the agent will be retrained, e.g., one means that the agent will be retrained after each CI cycle.  
[-o OUTPUT\_PATH] --> The path where the results and the trained agents will be saved

### **Example:**

```
python testCase_prioritization/TPDRL.py -m pairwise -a A2C -t ../data/iofrol-additional-features.csv -e 200 -w 10 -d simple
```

The above command starts training of an RL agent using A2C algorithm and pairwise ranking based on dataset iofrol

## **References**

Ferrari, A., Spagnolo, G. O., & Gnesi, S. (2017, September). PURE: A dataset of public requirements documents. In 2017 IEEE 25th International Requirements Engineering Conference (RE) (pp. 502-505). IEEE.