

Configuration Manual

MSc Research Project
Cloud Computing

Meerath Nida Aman
Student ID: x21228841

School of Computing
National College of Ireland

Supervisor: Rejwanul Haque

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: Meerath Nida Aman.....

Student ID: X21228841.....

Programme: MSc in Cloud Computing..... **Year:** 2023.....

Module: MSc Research Project.....

Lecturer: 14/12/2023.....

Submission Due Date:

Project Title: Integrating BERT-Based Feature Extraction with Traditional Algorithms for Low-Latency DNS Mappings in Osmotic Computing.

Word Count: 167..... **Page Count:** 7.....

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: Meerath Nida Aman.....

Date: 14/12/2023.....

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

Meerath Nida Aman
Student ID: x21228841

1 Introduction

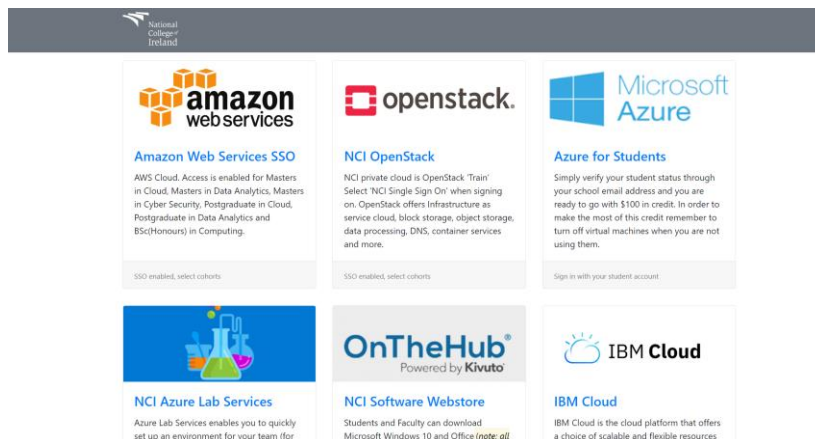
The setup process of the research work, utilizing BERT(Bidirectional Encoder Representations from Transformers)-based semantic modeling and traditional clustering algorithms like KMeans, DBSCAN and Agglomerative Clustering to enhance MicroElement identification within DNS datasets for enhanced discovery and reduced latency in Osmotic Computing is discussed in this Configuration Manual.

2 Prerequisites

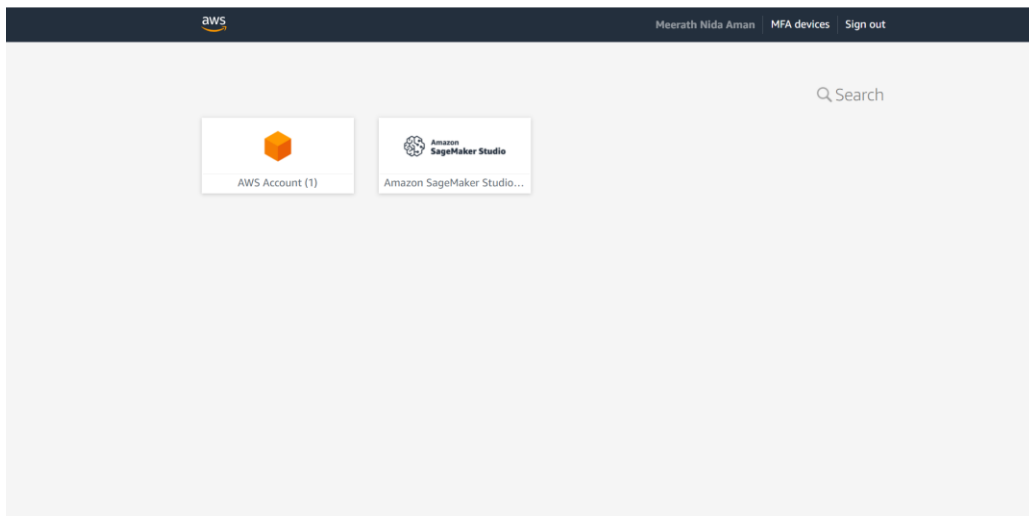
1. Amazon Sagemaker

3 Prerequisite Installation

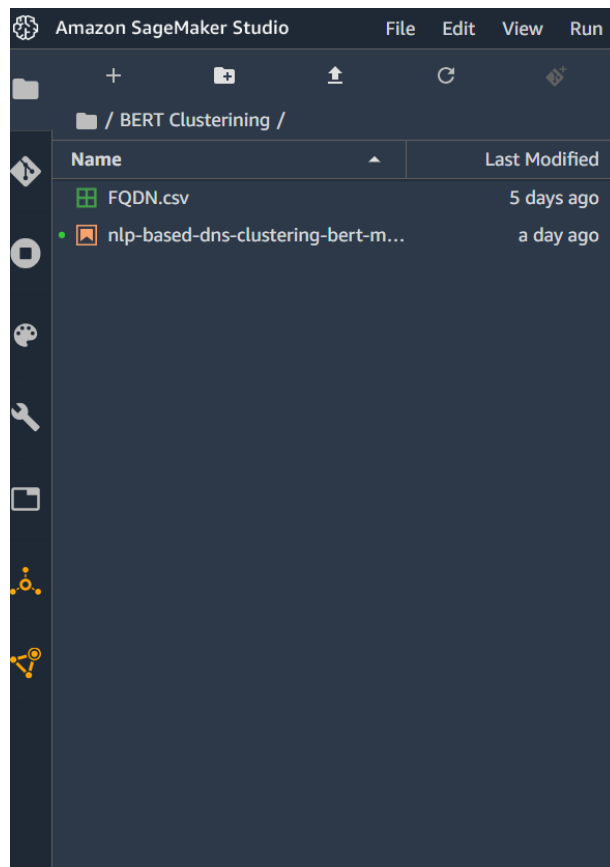
1. Go to the link <https://cloud.ncirl.ie/> with appropriate access.



2. Click on Amazon Web Services
3. Select Amazon Sagemaker Studio



4. Create a Folder Called BERT and add the dataset with the code.



- The folder FQDN.csv contains the Dataset containing 2000 DNS addresses and their features.

Name		Last Modified	FQDN					IP		Location	Hops	Latency	ResponseTime
FQDN.csv		10 hours ago	1	gazetteandherald.co.uk				36.108.22.65		Los Angeles	8	29.51	49.78
nlp-based-dns-cluste...		10 hours ago	2	mlssooccer.com				151.203.156.167		Sydney	1	78.86	15.55
			3	reddit.com				4.21.169.144		Los Angeles	9	88.19	30.66
			4	redbullracing.com				56.240.90.237		Sydney	5	33.66	30.57
			5	stackexchange.com				200.76.224.26		Dublin	4	48.58	21.03
			6	fifa.com				37.133.159.54		Hyderabad	5	70.28	35.08
			7	sufo.co.uk				139.115.241.174		Berlin	2	29.38	16.79
			8	economist.com				136.121.238.48		Los Angeles	6	83.22	46.21
			9	vulture.com				7.0.146.14		Dublin	1	22.13	39.1
			10	refinery29.com				104.117.134.233		Dublin	10	93.68	15.69
			11	musicweek.com				87.211.17.227		Sydney	5	90.95	18.74
			12	stackoverflow.com				67.195.204.205		Sydney	6	72.7	42.37
			13	pinterest.com				206.85.21.204		Berlin	7	82.93	40.93
			14	chesterfield-fc.co.uk				24.20.52.106		Sydney	1	56.51	9.98
			15	slack.com				210.69.20.195		Hyderabad	8	45.89	39.06
			16	boltonwanderers.co.uk				112.14.250.252		Dublin	3	63.53	9.48
			17	cafc.co.uk				106.155.205.122		Dublin	2	98.41	47.61
			18	techcrunch.com				172.107.75.166		Sydney	1	56.36	10.61
			19	nationalgeographic.com				240.202.98.99		Sydney	6	49.43	19.92
			20	bfc.co.uk				88.154.105.103		Dublin	6	93.42	46.37
			21	waitfordc.com				139.156.21.75		Los Angeles	1	1.2	17.29
			22	espn.com/racing				64.221.212.211		Hyderabad	10	97.56	9.56

4 Running the proposed Code

- Install all the prerequired packages by running the below part of the code.

```

!pip install tensorflow pandas scikit-learn nltk transformers
import tensorflow as tf
import pandas as pd
from sklearn.model_selection import train_test_split
import numpy as np
import re
import unicodedata
import nltk
from nltk.corpus import stopwords
from tensorflow import keras
from tensorflow.keras.layers import Dense,Dropout, Input, Reshape
from tqdm import tqdm
import pickle
from sklearn.metrics import confusion_matrix,f1_score,classification_report
import matplotlib.pyplot as plt
import itertools
from sklearn.utils import shuffle
from tensorflow.keras import regularizers
from transformers import BertTokenizer, TFBertModel, BertConfig,TFDistilBertModel,DistilBertTokenizer,DistilBertConfig
import numpy as np
import pandas as pd
# https://swatimeena989.medium.com/distilbert-text-classification-using-keras-c1201d3a3d9d
from sklearn import preprocessing
import warnings
warnings.filterwarnings("ignore")

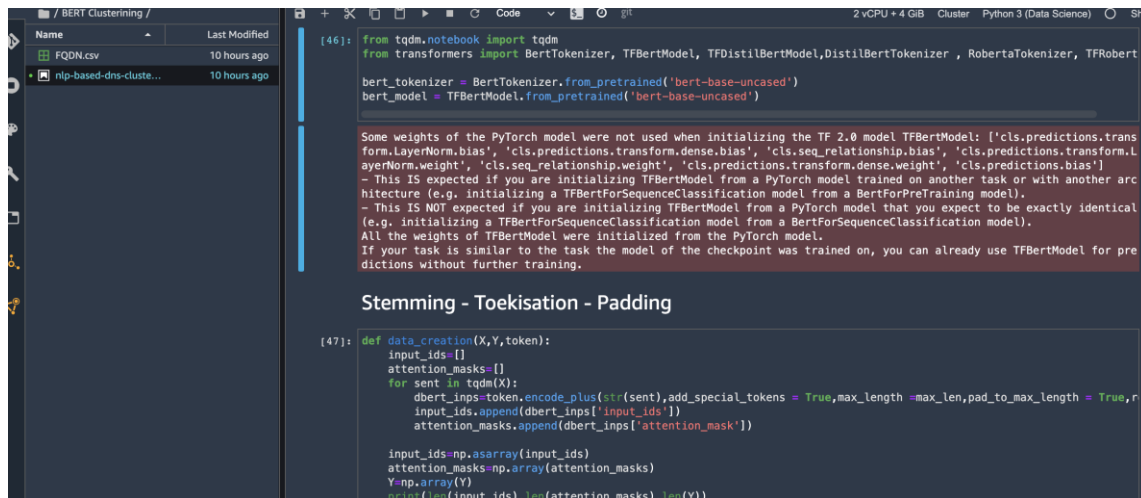
```

```

Requirement already satisfied: tensorflow in /opt/conda/lib/python3.7/site-packages (2.11.0)
Requirement already satisfied: pandas in /opt/conda/lib/python3.7/site-packages (1.3.5)
Requirement already satisfied: scikit-learn in /opt/conda/lib/python3.7/site-packages (0.22.1)
Requirement already satisfied: nltk in /opt/conda/lib/python3.7/site-packages (3.8.1)
Requirement already satisfied: transformers in /opt/conda/lib/python3.7/site-packages (4.30.2)
Requirement already satisfied: absl-py>=1.0.0 in /opt/conda/lib/python3.7/site-packages (from tensorflow) (2.0.0)
Requirement already satisfied: astunparse>=1.6.0 in /opt/conda/lib/python3.7/site-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=2.0 in /opt/conda/lib/python3.7/site-packages (from tensorflow) (23.5.26)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in /opt/conda/lib/python3.7/site-packages (from tensorflow) (0.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in /opt/conda/lib/python3.7/site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /opt/conda/lib/python3.7/site-packages (from tensorflow) (1.60.0)
Requirement already satisfied: h5py>=2.9.0 in /opt/conda/lib/python3.7/site-packages (from tensorflow) (2.10.0)
Requirement already satisfied: keras<2.12,>=2.11.0 in /opt/conda/lib/python3.7/site-packages (from tensorflow) (2.11.0)
Requirement already satisfied: libclang>=13.0.0 in /opt/conda/lib/python3.7/site-packages (from tensorflow) (16.0.6)

```

2. Preprocess the data in dataset and extract features by running the below part of the code to implement BERT.



The screenshot shows a Jupyter Notebook with a file explorer on the left containing 'FQDN.csv' and 'nlp-based-dns-cluste...'. The main area displays code cell [46] which imports 'tqdm.notebook.tqdm' and 'transformers' components like 'BertTokenizer', 'TFBertModel', 'TFDistilBertModel', 'DistilBertTokenizer', 'RobertaTokenizer', and 'TFRobertaModel'. It then initializes 'bert_tokenizer' and 'bert_model' from pre-trained 'bert-base-uncased' models. A warning message follows, stating that some weights of the PyTorch model were not used when initializing the TF 2.0 model. Below the warning, the text 'Stemming - Toekisation - Padding' is shown. Cell [47] defines a 'data_creation' function that takes 'X', 'Y', and 'token' as inputs, processes the tokens into 'input_ids' and 'attention_masks', and returns them as NumPy arrays.

```
[46]: from tqdm.notebook import tqdm
from transformers import BertTokenizer, TFBertModel, TFDistilBertModel, DistilBertTokenizer, RobertaTokenizer, TFRobertaModel

bert_tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
bert_model = TFBertModel.from_pretrained('bert-base-uncased')
```

Some weights of the PyTorch model were not used when initializing the TF 2.0 model TFBertModel: ['cls.predictions.transform.LayerNorm.bias', 'cls.predictions.transform.dense.bias', 'cls.seq_relationship.bias', 'cls.predictions.transform.LayerNorm.weight', 'cls.seq_relationship.weight', 'cls.predictions.transform.dense.weight', 'cls.predictions.bias']
- This IS expected if you are initializing TFBertModel from a PyTorch model trained on another task or with another architecture (e.g. initializing a TFBertForSequenceClassification model from a BertForPreTraining model).
- This IS NOT expected if you are initializing TFBertModel from a PyTorch model that you expect to be exactly identical (e.g. initializing a TFBertForSequenceClassification model from a BertForSequenceClassification model).
All the weights of TFBertModel were initialized from the PyTorch model.
If your task is similar to the task the model of the checkpoint was trained on, you can already use TFBertModel for predictions without further training.

Stemming - Toekisation - Padding

```
[47]: def data_creation(X,Y,token):
    input_ids=[]
    attention_masks=[]
    for sent in tqdm(X):
        dbert_inps=token.encode_plus(str(sent),add_special_tokens = True,max_length =max_len,pad_to_max_length = True,return_tensors='tf')
        input_ids.append(dbert_inps['input_ids'])
        attention_masks.append(dbert_inps['attention_mask'])

    input_ids=np.asarray(input_ids)
    attention_masks=np.array(attention_masks)
    Y=np.array(Y)
    print(len(input_ids),len(attention_masks),len(Y))
```

3. Implement KMeans Clustering by running below code

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from mpl_toolkits.mplot3d import Axes3D
# Create a figure with a specific size
plt.figure(figsize=(20, 10))
plt.grid(True)
# Generate synthetic data
X = dfop
# Standardize the features
X = StandardScaler().fit_transform(X)
# Standardize the features
# K-Means clustering
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans_labels = kmeans.fit_predict(X)
plot_clusters(X, kmeans_labels, "K-Means Clustering")
evaluate_clusters(X, kmeans_labels)
```

4. Implement DBSCAN Clustering by running below code

```
# DBSCAN clustering
plt.figure(figsize=(20, 10))
plt.grid(True)
dbscan = DBSCAN(eps=0.03, min_samples=5)
dbscan_labels = dbscan.fit_predict(X)
plot_clusters(X, dbscan_labels, "DBSCAN Clustering")
# DBSCAN doesn't require specifying the number of clusters, so there's no need for evaluation
```

5. Implement Agglomerative Clustering by running below code

```
plt.figure(figsize=(20, 10))
# Agglomerative clustering
plt.figure(figsize=(20, 10))
plt.grid(True)
agglomerative = AgglomerativeClustering(n_clusters=3)
agglomerative_labels = agglomerative.fit_predict(X)
plot_clusters(X, agglomerative_labels, "Agglomerative Clustering")
evaluate_clusters(X, agglomerative_labels)
```

Video Presentation Link:

https://studentncirl-my.sharepoint.com/:v/g/personal/x21228841_student_ncirl_ie/Ea7J3FiycLtGqCgD6Vg-FHYBfG2JZu9E8SRNV-iVMC3gSg?e=ybXLEv

References

1. Zhang, K., Tan, L., & Wang, H. (2010). Feature selection in clustering high-dimensional data: A new solution. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 10).