Configuration Manual

MSc Research Project
Data Analytics

Kirubakaran Balaraman
Student ID: x19241658

School of Computing
National College of Ireland

Supervisor: Hicham Rifai

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Kirubakaran Balaraman

Student ID: X19241658

Programme: Data Analytics **Year:** 2020/2021

Module: Msc Research Project

Lecturer: Hicham Rifai

Submission Due Date: 16/08/2021

Project Title: A Robust Text-to-SQL Parser with Optimized Pretraining Approach

Word Count: 1151 **Page Count:** 5

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: 16/08/2021

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

Kirubakaran Balarman
x19241658

1. Introduction

This configuration manual provides detailed documentation and instruction on solution implemented as part of the research thesis, A Robust Text-to-SQL Parser with Optimized Pretraining Approach. The document covers all neseccasry instructions needed to reproduce the results.

2. System Configuration

2.1 Hardware Requirements

The solution is implemented in the Google Collaboratory environment as model training needed huge computational resource and GPU power. The specifications for the host machine are:

- Operating System: Windows 10 x64
- Processor: Intel Core i5-9300H @ 2.4 GHz
- RAM: 8 GB
- Hard drive: 500 GB SSD

The Google collaboratory environment provides GPU and RAM for free and 12 hours of continuous GPU time which is more than enough for running the whole training process.

2.2 Software Requirements

The project is implemented using Python 3.6. The project related code is written in the Jupyter notebook hosted in Google Colab. The code for implementing the pretrained model RoBERTa and the Bi-LSTMs has been adopted from previous research by importing the libraries. Changes have been made to use RoBERTa as a tokenizer and augment data using paraphrasing. These files need to be uploaded to google drive which will be discussed in detail in the following section. We used Pytorch to implement the code and SQLite database to store the sample tables and query them. Most of the libraries are preinstalled in Google Colab environment.

The following softwares are used for this research.

- Python 3.6
- Suitable text editor for python scripts (Used Notepad++)
- Jupyter Notebook 6.0.4

Initial steps can be done in local system, model training needs CUDA enabled GPUs and due to resource constraints, training is performed in Google Colab only. Local system is used to write new code and make changes to the reused code.

The python packages required for reproducing the results are listed below:

- torch
- transformers = 3.4.0
- tecords
- sqlachemy = 1.3.23
- nltk
- pandas
- matplotlib
- cuda
- time
- os
- torchsummary
- json
- argparse
- corenlp_local
- tqdm

3. Dataset Description

The WikiSQL dataset is released as part of the research done by Zhong et al. (2017). It contains natural language question and SQL pairs along with the table schema files. The datasets are present in the artefacts file in the path TextToSQL/data. The train_knowledge.jsonl is the final file passed to the model. It contains question tokens, header tokens, question and header knowledge vectors and the ground truth SQL.

4. Environment Setup

The google colab environment needs to be setup to carry out the research. To run and test the model successfully the code and data files need to be set in place. Follow the below steps carefully.

i. The datasets and libraries needed along with the python scripts are provided in the 19241658_Artefacts.zip file. Unzip the file to find the TextToSQL folder and Notebooks folder

ii. Upload only the TextToSQL folder to a location in your google drive.

iii. The Notebooks folder contains two .ipynb files. In order to train the model from scratch, test and see the evaluation results use the notebook **nl2sql_full_training.ipynb**. To skip the training process and directly predict SQL query by using the trained checkpoints use the notebook **nl2sql_infer.ipynb.** The steps (iv) to (ix) are similar for both the cases.

iv. Open Google colab and login through the same gmail account corresponding to the drive containing TextToSQL folder.

v. Click File -> Upload Notebook -> Browse and upload the notebook you want from the local system.

vi. As the code needs a GPU to run on, change the runtime type to GPU in colab notebook as shown in Fig 1. Then change the hardware accelerator to GPU.

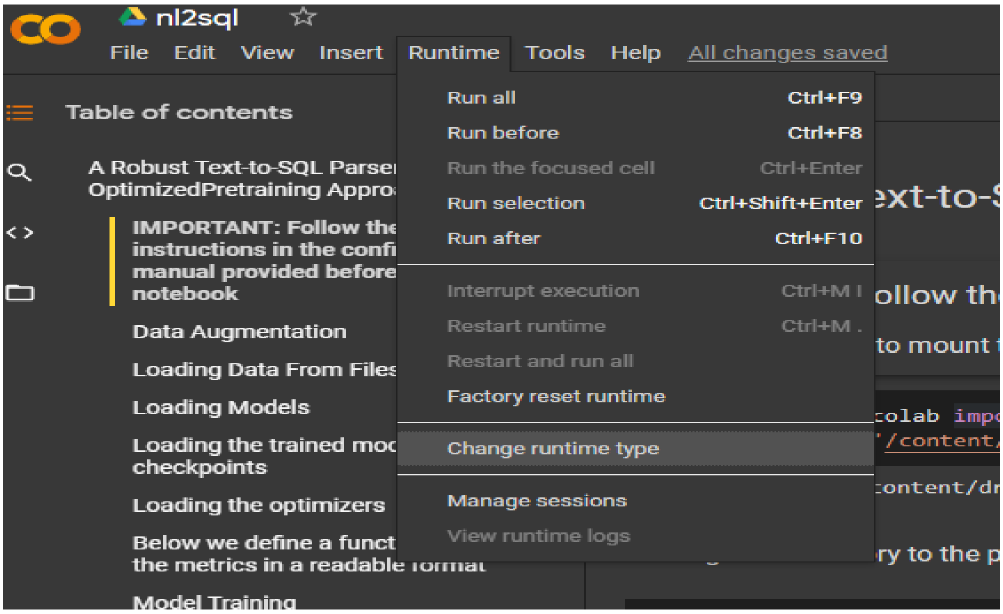
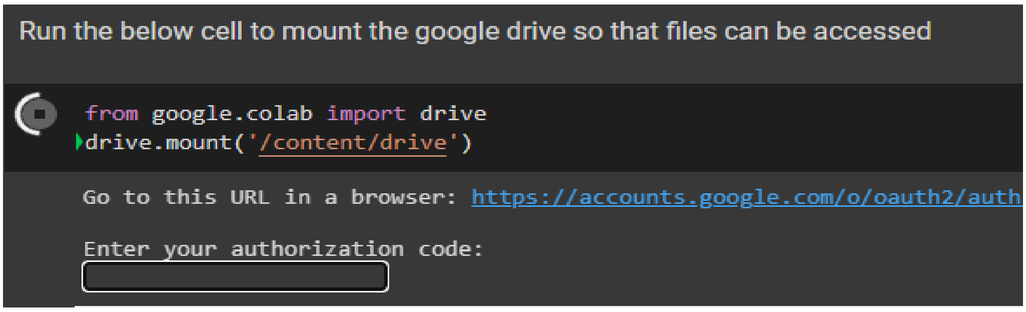


Figure 1 Changing runtime to GPU in Google Colab

vii. Now mount your google drive to the notebook session by running the cell shown in Fig 2. and paste the authorization code by following the URL. This will successfully mount your drive to the colab notebook.



Run the below cell to mount the google drive so that files can be accessed

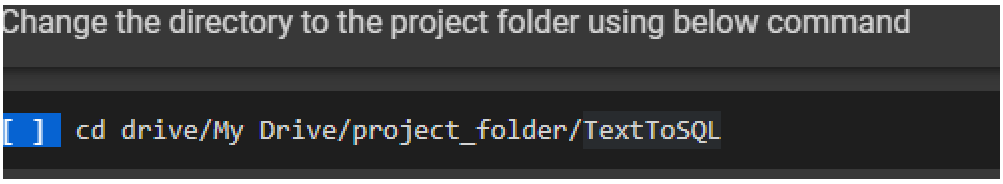
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth

Enter your authorization code:
```

Figure 2 Mounting the drive to the colab notebook session

viii. Change the path to the TextToSQL folder by using the command in the cell in Fig 3. The path should be changed according to the path of TextToSQL folder in your drive.

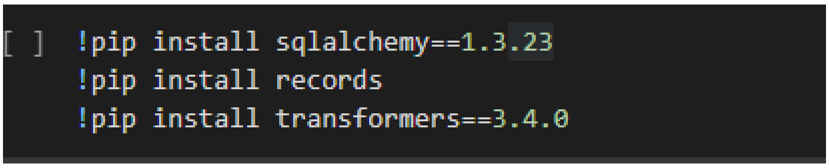


Change the directory to the project folder using below command

```
[ ] cd drive/My Drive/project_folder/TextToSQL
```

Figure 3 Changing directory using cd command

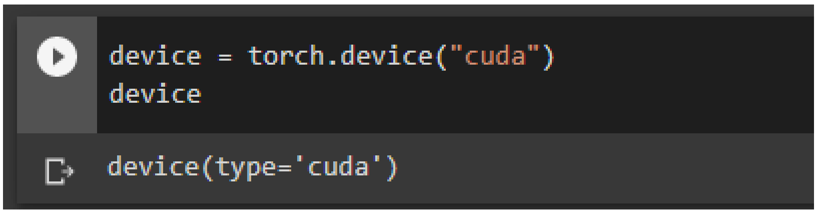
ix. The google colab environment comes with all the basic packages installed. But some packages are not available which can be installed by running the cell in Fig 4.



```
!pip install sqlalchemy==1.3.23
!pip install records
!pip install transformers==3.4.0
```

Figure 4 Installing necessary libraries in the colab session

x. Import the necessary libraries by running the corresponding cell in the notebook
xi. As the last step run the cell shown in Fig 5 to set the device to GPU instead of CPU



```
device = torch.device("cuda")
device
```

```
device(type='cuda')
```

Figure 5 Setting the Pytorch device to cuda to make it run on GPU

The environment will be setup successfully if you followed the previous steps correctly. In order to train the model from scratch continue with Section 5. To skip the training process and directly run the model go to Section 6.

5. Training and Evaluating the model

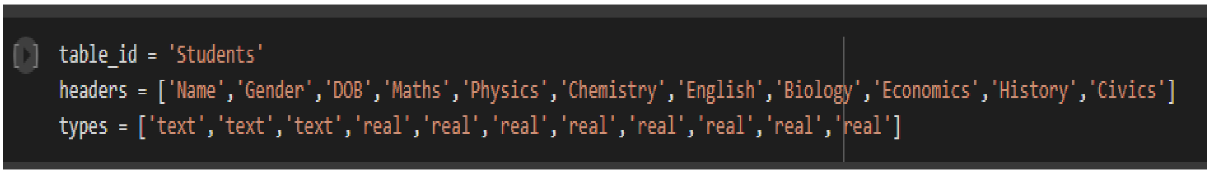
The nl2sql_full_training.ipynb notebook is used to train the model and evaluate it. Run every cell in the notebook to successfully complete the training process. This notebook contains code to trigger the script doing the paraphrasing to augment the data. After augmentation two files will be generated in the TextToSQL/data folder namely aug_train_knowledge.jsonl and aug_dev_knowledge.jsonl. Then the data and the models will be loaded along with the optimizer. Then the model will be trained for 5 epochs and the best model checkpoints are saved/replaced in the model folder as model_roberta_best.pt and model_best.pt for the RoBERTa and the Sequence-to-sequence model respectively. Then the model is evaluated using the test set and the results will be printed in graphs. The final section 'TEST NOW' contains the code to predict SQL by passing input question. This part is separately given in the nl2sql_infer.ipynb notebook to directly use the model to predict.

6. Making Predictions

The nl2sql_infer.ipynb notebook contains the code for making predictions. If you skip the training process, then it is important to download the model checkpoints and upload it to google drive before running the notebook. Follow the below steps to successfully make predictions:

i. The trained model checkpoints can be downloaded from the following link : https://drive.google.com/drive/folders/16KVe_QmLvA2KSsXIdn8GS-LPd6Sn__ut?usp=sharing
ii. Unzip the models.zip file to find two pickle files namely model_roberta_best.pt and model_best.pt.
iii. Upload both the files to the TextToSQL/model folder of your google drive

iv. Then continue running the nl2sql_infer notebook with the instructions provided in it to make predictions. The steps to mount the drive and to set the folder path are provided in the environment section.



```
table_id = 'Students'
headers = ['Name','Gender','DOB','Maths','Physics','Chemistry','English','Biology','Economics','History','Civics']
types = ['text','text','text','real','real','real','real','real','real','real','real']
```

Figure 6 Inputting sample table schema

Two example schema for Students and band related table is already tested. In order to test with your own database, provide the table name, header list and their types to the variables and lists shown in Figure 6. Then submit a meaningful question with respect to the table schema provided to predict the SQL query.

References

Zhong, V., Xiong, C. and Socher, R. (2017). Seq2sql: Generating structured queries from natural language using reinforcement learning,arXiv preprint arXiv:1709.00103.