NANYANG POLYTECHNIC

Software Design Specification

AI Project for Smart Assignment Phase 2(draft)

Revision 1.0

Table of Contents

Table of Contents ..... **ii**
Revision History ..... **iii**
Approved By ..... **iii**
1. Introduction **1**
1.1 Purpose 1
1.2 System Overview 1
2. Design Considerations **1**
2.1 Assumptions 1
2.2 Constraints 1
2.3 System Environment 1
2.4 Design Methodology 1
2.5 Risks and Volatile Areas 1
3. Architecture **2**
3.1 NLP Model Overview 2
3.2 BERT model ..... **Error! Bookmark not defined.**
4. Implementation **5**
4.1 General Software Architecture 5
4.2 Dataset 5
4.3 BERT Model 6
4.4 Predication API 7

Appendix A: Project Timeline **9**

Revision History

Version	Name	Reason For Changes	Date
1.0	*Wee Chee Hong*	*Initial Revision*	*5/14/2021*

Approved By

Name	Signature	Department	Date
Kim Foong		SIT	

1. Introduction

1.1 Purpose

This design will detail the implementation of the requirements as defined in the User Software Requirements Specification – AI Project for Smart Assignment (proof of concept) – Phase 1.

1.2 System Overview

This project will design and implement the functionality to predict the attention level needed for the job based on the user case description in the text input. It will be integrated as part of the main HDB maintenance request system for the AI for smart assignment.

2. Design Considerations

All design considerations were based on the User Requirement in Phase 1. (Using the Natural Language Processing-NLP with Deep Learning approach)

2.1 Assumptions

The design is based on the assumption of the sample data given (case description).

2.2 Constraints

We do not have the full knowledge regarding how HDB decides the importance of the case base on the case description.

2.3 System Environment

The Python case description prediction application will reside in the server computer on the customer's machine that has a Windows 10 operating system.

2.4 Design Methodology

Since the problem domain is a text classification problem, we will be using the NLP deep learning approach to design the application.

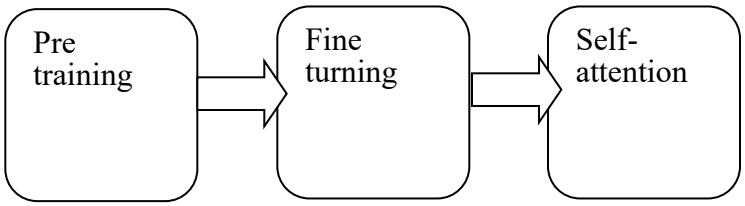
2.5 Risks and Volatile Areas

The degree of classifying a text case description with important intent is very diverse. We have tried to cast a range we think is appropriate. However, due to the diversity in the English language expression, there will be a potential risk of miss classification.

3. Design Architecture

3.1 Natural Language Processing(NLP) Model Overview

One of the biggest challenges in NLP is the lack of enough training data. Overall, there is an enormous amount of text data available, but if we want to create task-specific datasets, we need to split that pile into the very many diverse fields. And when we do this, we end up with only a few thousand or a few hundred thousand human-labeled training examples. Unfortunately, to perform well, deep learning-based NLP models require much larger amounts of data — they see major improvements when trained on millions, or billions, of annotated training examples. To help bridge this gap in data, researchers have developed various techniques for training general-purpose language representation models using the enormous piles of unannotated text on the web (this is known as **pre-training**). These general-purpose pre-trained models can then be **fine-tuned** on smaller task-specific datasets, e.g., when working with problems like question answering and sentiment analysis. This approach results in great accuracy improvements compared to training on the smaller task-specific datasets from scratch. BERT is a recent addition to these techniques for NLP pre-training; it caused a stir in the deep learning community because it presented state-of-the-art results in a wide variety of NLP tasks, like question answering.

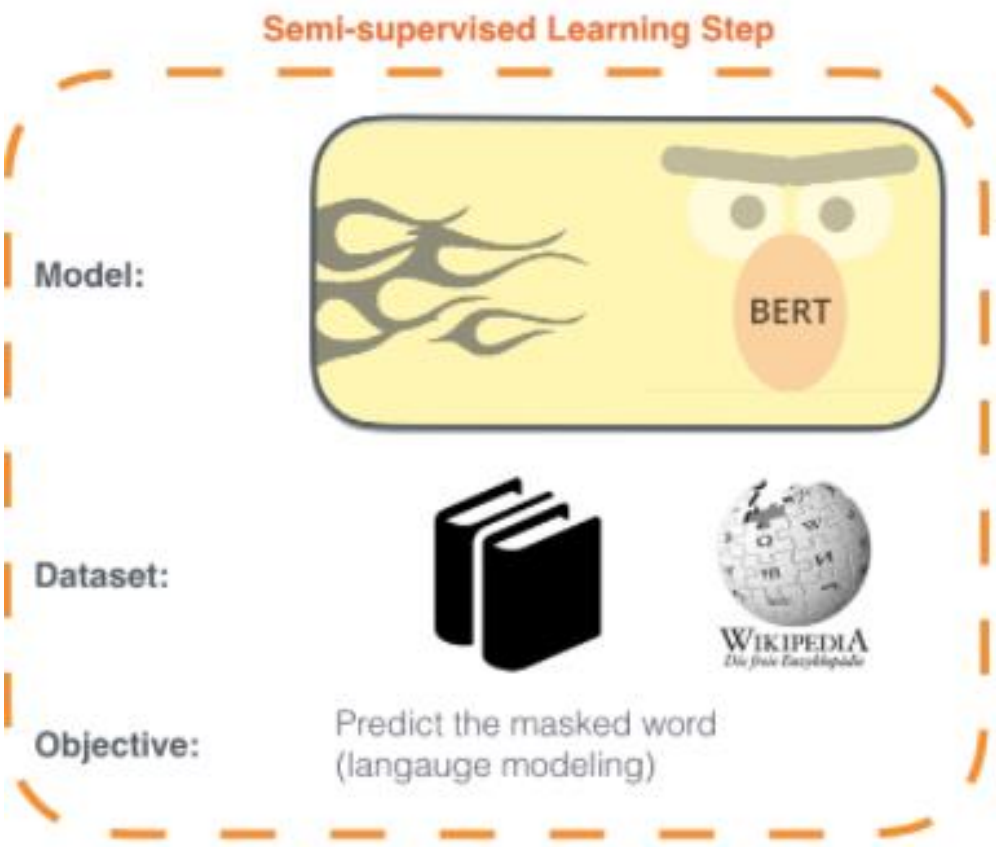


3.2 BERT model

BERT builds on top of several clever ideas that have been bubbling up in the NLP community recently – including but not limited to Semi-supervised Sequence Learning and the Transformer.
The following shows the two steps of how BERT is developed. First, it can download the model pre-trained in step(trained in un-annotated data), and only worry about fine-tuning it for step 2.

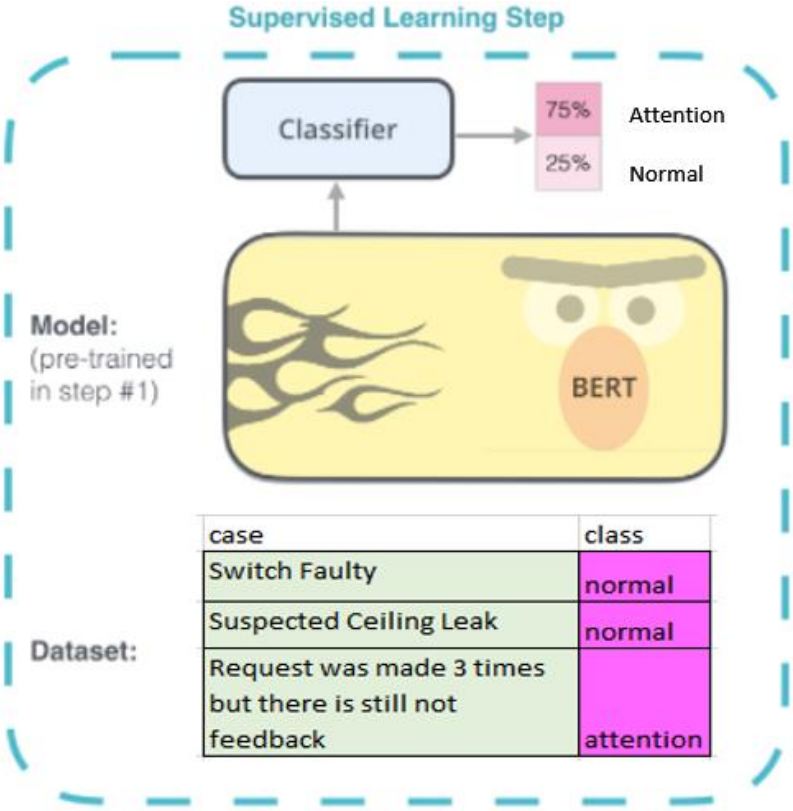
1. Semi-supervised training on large amounts of text(books, Wikipedia..etc)

The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build in a supervised way.



2. Supervised training on a specific task with labeled data.

In part of the model, we will use the domain-specific data to do the training.
In this case, we using the case description and label to do the supervised training.



4. Implementation

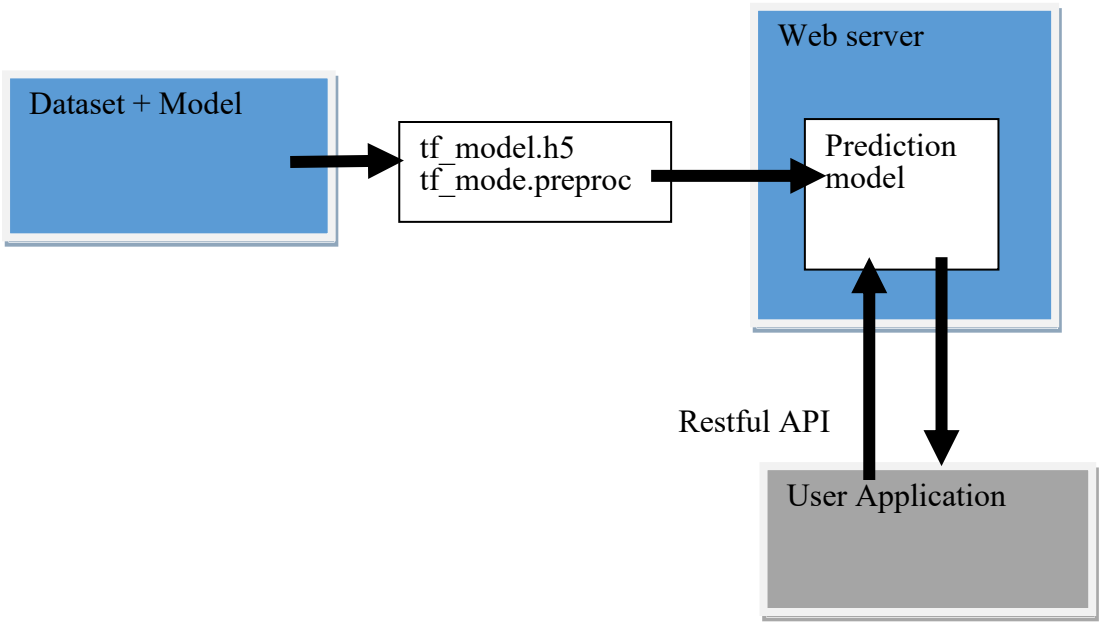
Here provide an implementation description of the general software architecture, dataset, model and prediction application for the design described in section 3.

4.1 General Software Architecture

The software can be divided into to development segments.

The first segment is the Dataset and the Model. We will preprocess the data to be used for the training of the model. The trained model result will store in two binary files. One is the model description and the other is the model learned parameters.

The second segment is the Prediction application. The prediction application will load the 2 binary files from the trained model to do the prediction for the input data. A restful API is implemented in order for the user to request for the prediction services.



4.2 Dataset

From the user requirements phase 1 we had identified the data field required to decide the case is of some level of importance.

Data field is a text sentence that describes the building maintenance case reported to the HDB.

Data Sample

Following is some example of the data provided.

Case Number	Case Description
1	Wrong key given for main gate.
2	window latch broke at master and common bedroom
3	Window Grille request
4	WC seat defective.
5	WC seat defective
6	Water stains outside CT
7	Water stain marks on master bedroom toilet ceiling.

Case Description Intention

No information was provided to decide on the importance of the case based on the case description.

We use the intent approach to classify the importance of the case description.

Based on the 363 case descriptions we had to identify the following intents which are potential to classify the case description as important.

- repeat/recurrent/again/resend
- Unhappy/upset/disappoint/angry/impatient
- Request many times/numerous time/
- Not safe/dangerous/hazards
- report x weeks ago/days ago/year ago
- persistence/continuous
- urgent/

Format the Data into the training dataset

There is no automatic way to associate each case description with the expected result (or Label/Class).

We need to manually read each of the cases and associate it to the defined intent and label the output.

We will use either the normal or attention to represent the expected result of a case description and label it as the output.

Example of the case description with label

Case description	Label/Class
Incident Location: 740 XXX CIRCLE #03-419 XXX0740) Incident Location Description: Dear Team, I would like to seek assistance for the installation of 2 toilet exhaust fans. Kindly provide me with the correct channel to seek the answer for this. Thank you and appreciate your kind reply.	normal
Incident Location: 739 XXX CIRCLE #05-397 XXX0739) Incident Location Description: Mr Lim request return call from SEE Mr Alvin Yeo, claim that EEIC is aware of his case. He would like EEIC to provide contractor number for his MBR ceiling repair. Please assist to return call, thank you.	normal
Incident Location: 738 XXX CIRCLE #10-375 XXX0738) Incident Location Description: Please assist with these two defect at my common toilet and and master bedroom toilet. The common toilet is ceiling leaking and water seepage is very bad and I unable to switch on the light. Please assist with this as soon as possible urgently.	attention
Incident Location: 738 XXX CIRCLE #10-369 XXX0738) Incident Location Description: Hi, the problem still exist, any update?	attention

4.3 Model

As described in section 3 Design, we will be using BERT model to train the dataset that we had prepared.

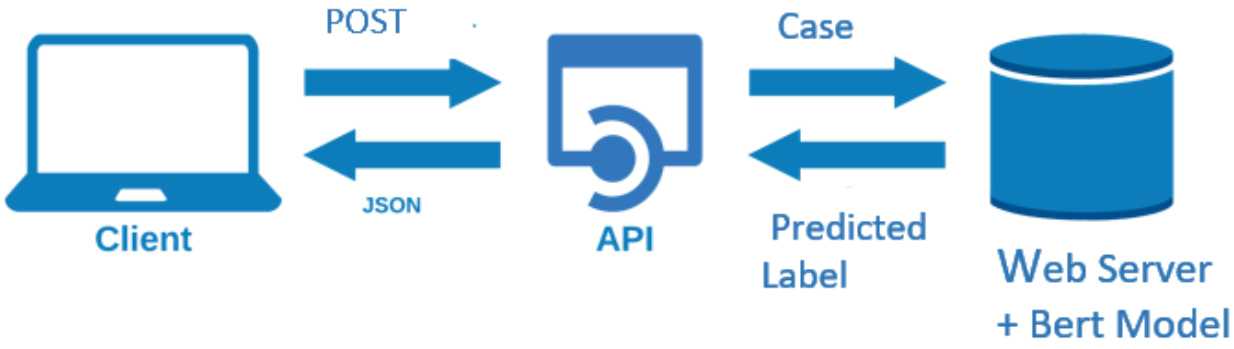
The Bert model is 12-layer, 768-hidden, 12-heads, 109M parameters. Trained on uncased English text. Apply the transfer learning to train the case description/label dataset to do a case classification.

```
begin training using onecycle policy with max lr of 2e-05...
Epoch 1/12
57/57 [==============================] - 54s 600ms/step - loss: 0.6155 - accuracy: 0.7245 - val_loss: 0.5186
- val_accuracy: 0.7326
Epoch 2/12
57/57 [==============================] - 30s 526ms/step - loss: 0.3313 - accuracy: 0.8709 - val_loss: 0.4701
- val_accuracy: 0.7907
Epoch 3/12
57/57 [==============================] - 30s 526ms/step - loss: 0.2678 - accuracy: 0.8819 - val_loss: 0.4586
- val_accuracy: 0.8372
Epoch 4/12
57/57 [==============================] - 30s 525ms/step - loss: 0.1433 - accuracy: 0.9582 - val_loss: 0.6300
- val_accuracy: 0.8023
Epoch 5/12
57/57 [==============================] - 30s 525ms/step - loss: 0.1223 - accuracy: 0.9484 - val_loss: 0.5706
- val_accuracy: 0.7558
Epoch 6/12
57/57 [==============================] - 30s 526ms/step - loss: 0.1413 - accuracy: 0.9711 - val_loss: 0.6496
- val_accuracy: 0.8372
Epoch 7/12
```

```
57/57 [==============================] - 30s 527ms/step - loss: 0.0551 - accuracy: 0.9792 - val_loss: 0.3661
- val_accuracy: 0.8256
Epoch 8/12
57/57 [==============================] - 30s 526ms/step - loss: 0.0407 - accuracy: 0.9934 - val_loss: 0.5495
- val_accuracy: 0.8372
Epoch 9/12
57/57 [==============================] - 30s 525ms/step - loss: 0.0064 - accuracy: 1.0000 - val_loss: 0.6546
- val_accuracy: 0.8372
Epoch 10/12
57/57 [==============================] - 30s 524ms/step - loss: 0.0037 - accuracy: 1.0000 - val_loss: 0.7009
- val_accuracy: 0.8372
Epoch 11/12
57/57 [==============================] - 30s 525ms/step - loss: 0.0029 - accuracy: 1.0000 - val_loss: 0.7171
- val_accuracy: 0.8372
Epoch 12/12
57/57 [==============================] - 30s 526ms/step - loss: 0.0027 - accuracy: 1.0000 - val_loss: 0.7240
- val_accuracy: 0.8372
<tensorflow.python.keras.callbacks.History at 0x7f4d70d81c90>
```

4.4 Prediction API

The trained model will be stored in a compressed file. We will implement a web api for the other application to integrate into the predictive model.



A python webserver with POST API integrates with the trained BERT model will be implemented.

A client program can access the service from the BERT model through the web API.

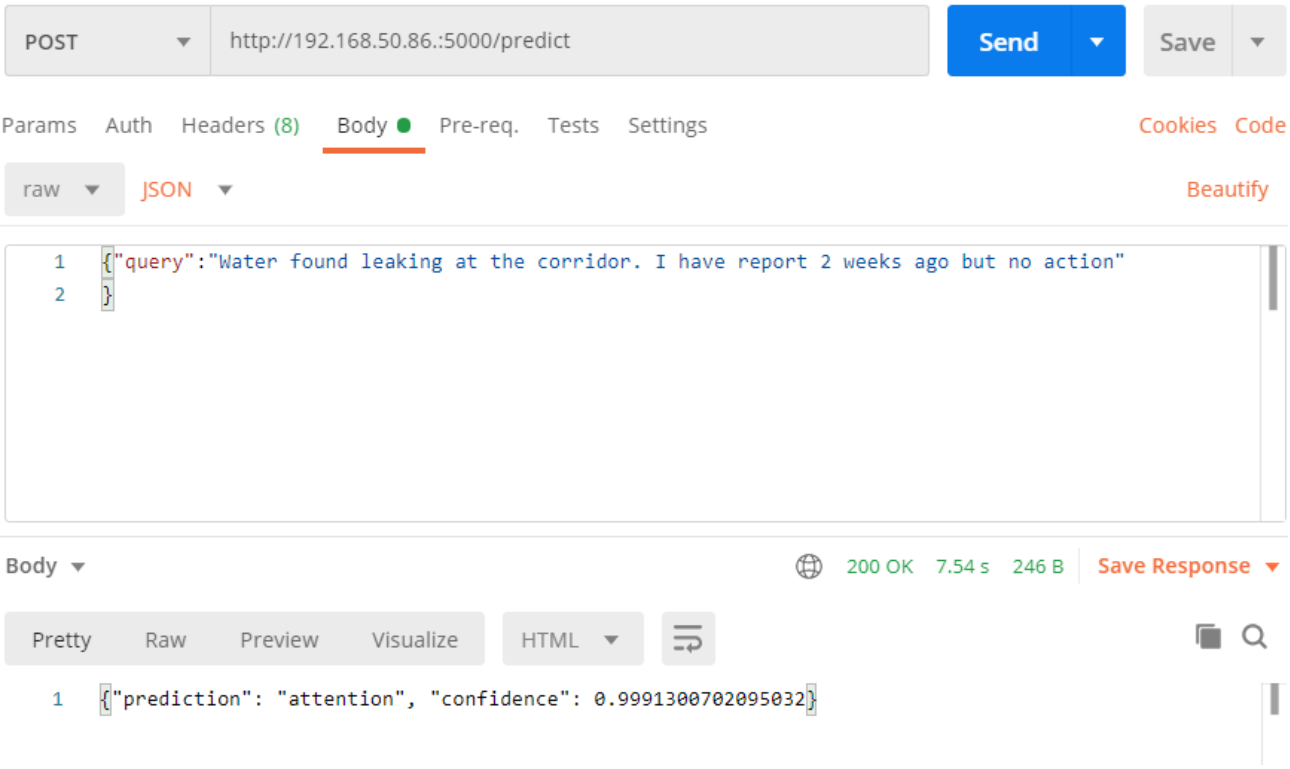
POST API

Request
http://server ip address:5000/predict

Body of API
Raw: Json
Format:
{ “query”:”case text”}

Response
Json: {“predication”: “attention”, ‘confidence”: 0.999}

Prediction: can be normal or attention
Confidence: between 0 to 1



5. Appendix A: Project Timeline

Reference the Microsoft project Binder Request Release 2 – Development..

	Start Week	Apr 5, 2021

Week	1	2	3	4	5	6	7	8	9	10	11	12	13
Starting	Apr 5	Apr 12	Apr 19	Apr 26	May 3	May 10	May 17	May 24	May 31	Jun 7	Jun 14	Jun 21	Jun 28
1. Data Preprocess													
Collection													
Testing label													
Check label result													
2. Model Training and Testing													
NLP Model update													
NLP Model training (fine tuning)													
3. NLP Model API													
Model API development													
Model API testing													