# Assignment–2

**Group Member–**

**Deepanshi Jindal (M20MA202)**

**Sushil Kumar Lakhiwal (M20MA205)**

Answer(1)

(a) As mentioned in the question1 of the assignment to build the classifier that can detect a predefined class of relations, as specified in the dataset.

We first of all specify the predefined classes in the dataset.The predefined classes are as follows–

CEO,SUBSIDIARY_OF,DATE_OF_BIRTH,RESIDENCE,BIRTH_PLACE,NATIONALITY, EDUCATED_AT, POLITIVCAL_AFF,SPOUSE,DATE_FOUNDED,EMPLOYER_OR_MEMBER_OF,CHILD_OF, HEADQUARTERS_FOUNDED_BY.

When we load the train.json data Then we define a relation classifier that predicts the relation in a paragraph. We define a preprocess text function to preprocess the data and a defined preprocess data function to extract the passages from the document text which have any one relation define in the predefined set of classes and we defined y list as a list of relationships present in the paragraphs.

Now we splitted the data in training and testing data with 80% for training and 20% for testing. Then we extracted bag of words, features from the text data with and defined a classifier model with dropout. We use ADAM optimizer with learning rate 0.001 and binary cross entropy as loss function.we then converted the relations to classify in numerical labels within then we train the model and find the history of that model on 10 epochs and the output of this model is as follows.

```
Epoch 1/10
98/98 [==============================] - 13s 121ms/step - loss: 0.3325 - accuracy:
      0.1356 - val_loss: 0.2239 - val_accuracy: 0.5019
Epoch 2/10
98/98 [==============================] - 12s 118ms/step - loss: 0.1983 - accuracy:
      0.4493 - val_loss: 0.1369 - val_accuracy: 0.7091
Epoch 3/10
98/98 [==============================] - 11s 114ms/step - loss: 0.1131 - accuracy:
      0.7481 - val_loss: 0.1073 - val_accuracy: 0.7413
Epoch 4/10
98/98 [==============================] - 11s 112ms/step - loss: 0.0641 - accuracy:
      0.8857 - val_loss: 0.1066 - val_accuracy: 0.7310
Epoch 5/10
```

```
98/98 [==============================] - 10s 106ms/step - loss: 0.0448 - accuracy:
        0.9224 - val_loss: 0.1082 - val_accuracy: 0.7375
Epoch 6/10
98/98 [==============================] - 11s 113ms/step - loss: 0.0338 - accuracy:
        0.9372 - val_loss: 0.1124 - val_accuracy: 0.7310
Epoch 7/10
98/98 [==============================] - 11s 115ms/step - loss: 0.0270 - accuracy:
        0.9378 - val_loss: 0.1163 - val_accuracy: 0.7439
Epoch 8/10
98/98 [==============================] - 11s 113ms/step - loss: 0.0241 - accuracy:
        0.9472 - val_loss: 0.1227 - val_accuracy: 0.7400
Epoch 9/10
98/98 [==============================] - 11s 113ms/step - loss: 0.0231 - accuracy:
        0.9427 - val_loss: 0.1213 - val_accuracy: 0.7477
Epoch 10/10
98/98 [==============================] - 11s 115ms/step - loss: 0.0196 - accuracy:
        0.9427 - val_loss: 0.1216 - val_accuracy: 0.7516
```

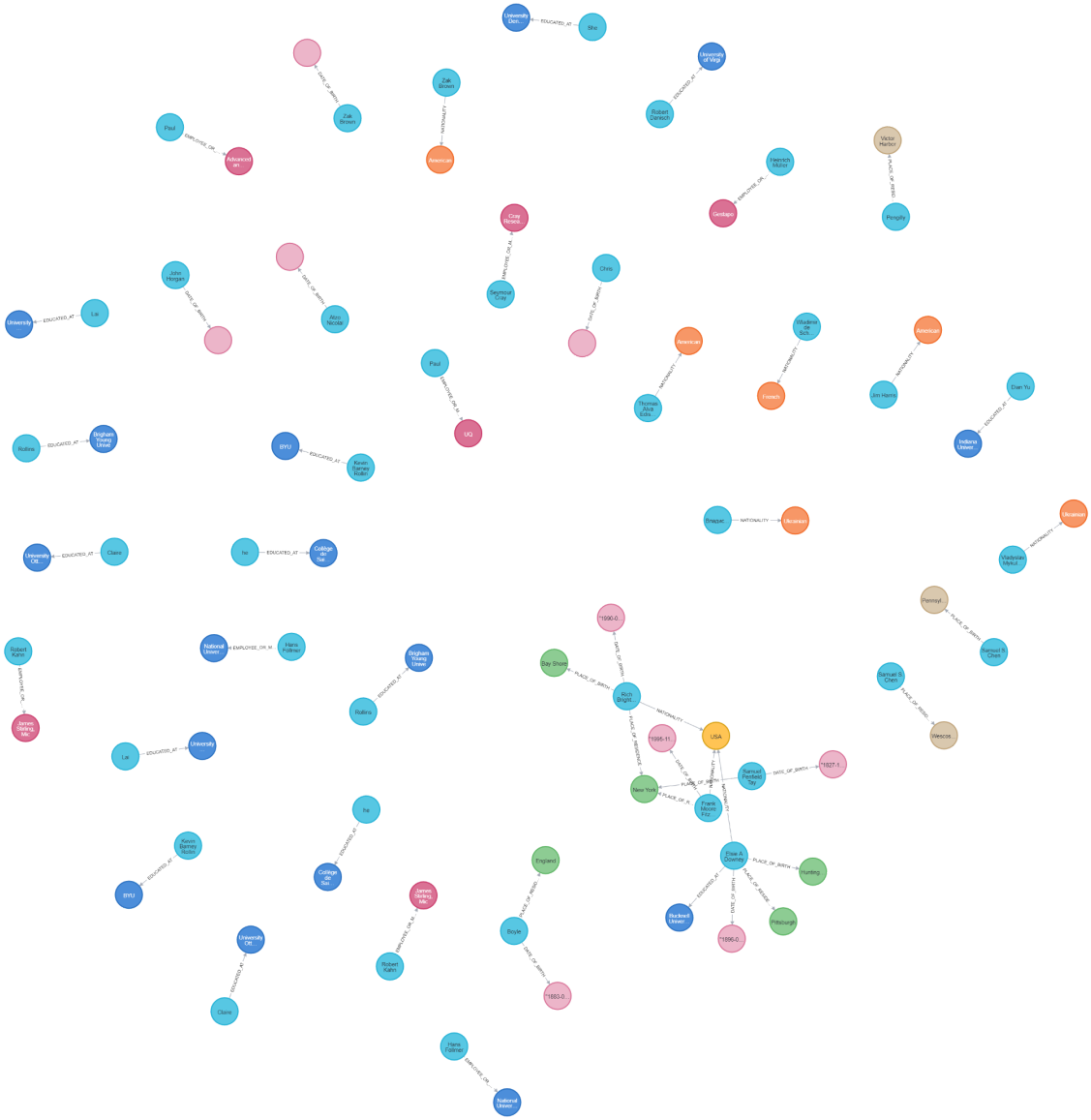
(b) Now we define the function that creates a subset of the knowledge data with using sentence which contain any of the following relations.

a.DATE_OF_BIRTH (PER–DATE)
b.RESIDENCE (PER–LOC)
c.BIRTHPLACE (PER–LOC)
d.NATIONALITY (PER–LOC)
e.EMPLOYEE_OF (PER–ORG)
f.EDUCATED_AT (PER–ORG)

We firstly check for the documents which contains any of the above relations in the data and then we append that documents to a list in this way we create the subset of the data in a list in colab Notebook which can be see with the following colab link–

**Colab link:-**
https://colab.research.google.com/drive/1BuMUkE0iPEufPD5XC35H4blqweGmwk7Q?usp=sharing

(c) now for building knowledge graph databases we coded to extract human readable text which has any of the relations defined in the (b) part. In this way we created human readable list which consists of triples with first subject second relation and third object entities. We use this list to find information about the nodes and relationships information which is required for creating nodes and relationships in Neo4j. We then used neo4j open source to create knowledge graph, the knowledge graph obtained is given in the next page of this document. We submitted the dump database of neo4j in zip file for evaluation and also provided a downloaded sample cypher query used to create nodes and relationships.



(d) now we created chatbot using a chatbot api,and streamlit package of python. we gave query to that chatbot and its output is the cypher query then we gave that cypher query to neo4j KG and found graphs the example snapshots
Given in the next page. We submitted the .py file of the code and also attached the snapshots of the chatbot and its given queries in the zip file.

# NLU Assignment--2

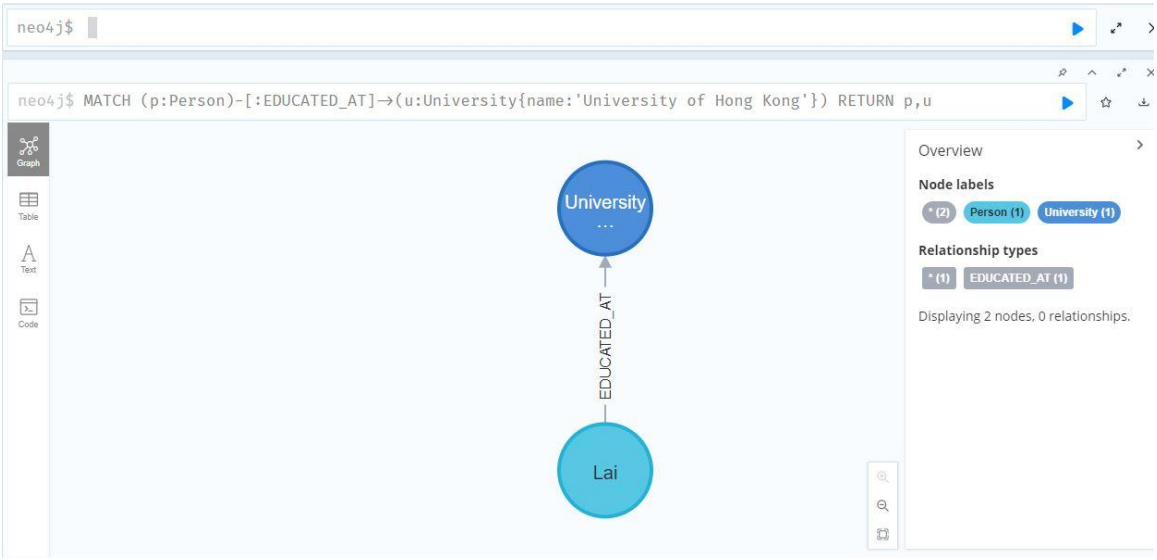
You:

give cypher query for all person that EDUCATED_AT from University of Hong Kong

The cypher query to find all persons that EDUCATED_AT from University of Hong Kong can be as follows:

```
MATCH (p:Person)-[:EDUCATED_AT]->
(u:University{name:'University of Hong Kong'})
RETURN p,u
```

In this query, we first match all nodes with the label "Person" that are connected via the "EDUCATED_AT" relationship to a node with label "University" and the property "name" equal to "University of Hong Kong". Finally, we return the matched nodes "p" and "u" representing the person and university nodes respectively.



# NLU Assignment--2

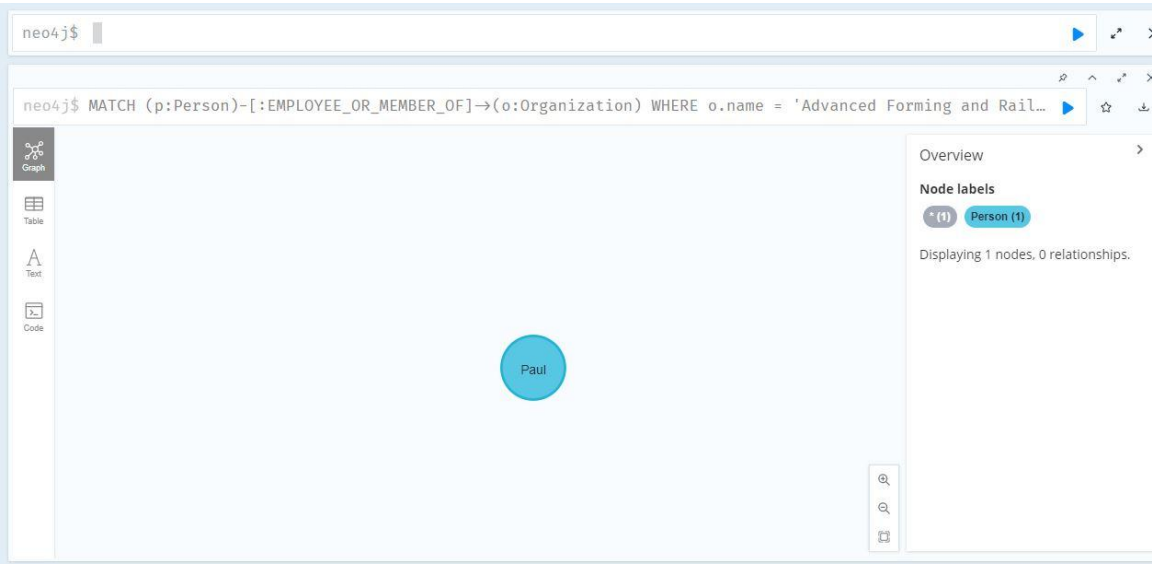
You:

give cypher query for all person that EMPLOYEE_OR_MEMBER_OF Advanced Forming and Railway Mech

Assuming all the nodes are in the default database of Neo4j, the following Cypher query can retrieve all the Persons that have the relationship EMPLOYEE_OR_MEMBER_OF with the Organization named Advanced Forming and Railway Mechanics.

```
MATCH (p:Person)-[:EMPLOYEE_OR_MEMBER_OF]->
(o:Organization)
WHERE o.name = 'Advanced Forming and Railway Mechanics'
RETURN p
```

This query uses the MATCH clause to find all the nodes with the labels Person and Organization that are connected through the EMPLOYEE_OR_MEMBER_OF relationship. The WHERE clause filters the results to only include the Organizations with the given name. Finally, the RETURN clause retrieves only the Person nodes.




**Colab link:-**
https://colab.research.google.com/drive/1BuMUkE0iPEufPD5XC35H4blqweGmwk7Q?usp=sharing

Now the difficulties faced by us are understanding the relation classifier and then creating a chatbot that connects Neo4j KG to chatbot.
Each member of  the group contributed equally in the assignment.