# NLU Coding Assignment - 2 [31/03/2023]
# CSL7340

## Instructions:

1. **Evaluation will be done through viva, where you have to demonstrate your model live, along with reviews/code.**
2. **Total marks - 20 marks, Deadline:- 25-04-2023 05:00 PM,**
3. **ANY ONE MEMBER OF THE GROUP CAN SUBMIT AND TURNIN. No need for other people to submit/turn in.**
4. **All libraries and tools are allowed.**
5. Submit a **.zip** file containing all the working codes (.py files and .pdf files). The zip file should be named in the format <RollNo1_RollNo2_RollNo3_NLU_A1>.zip.
   **[Do not include model weights and data in the zip file]**
6. Submit a **.pdf** report containing:
   a. A detailed description of what all you have done,
   b. A description of what else could be done to improve results - challenges faced
   c. Links to the Google-Colab files (if any),
   d. Clearly mention the contribution of each group member.
7. **Copying from the Internet and/or your classmates is strictly prohibited. Any team found guilty will be awarded a suitable penalty as per IIT rules.**
8. **Models have to be saved and run during the demonstration - model training should be complete, and no training should be initiated during the demonstration.**

# 1. Relation Classifier:

**Dataset:-** KnowledgeNet (data description and dataset can be found here:- link), use train.json inside the dataset as your source of data.

### Tasks:-

1. Building a relation classifier that can detect a predefined class of relations, as specified in the dataset.

2. Create a subset of the KnowledgeNet data using sentences which contain any of the following relations: (make a subset of train.json with these relations only)
    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)
3. Create a Knowledge Graph that can store the information contained in these sentences. You can use any open-source graph database for this purpose.
4. Connect the Knowledge Graph to a front end that can take in Natural Language Queries and give the answers back. You can use any open-source chatbot for this purpose. That way, the system will also have the power to continue a conversation rather than only Question-Answering.

NOTE

1. Save your relation classification models. We will test with arbitrary inputs while evaluating the system.
2. Your KG will be tested with random questions during the evaluation. You have to justify the answers generated by the system.
3. During evaluation - a conversation with the system will be tried.
4. Detailed report about the implementation and evaluation of each component should be submitted using train-test splits.
5. Evaluation will be in-person - new inputs will be used for testing.