**Semantic Web 2024/205**

**Course Project**

The Knowledge Graphs (KGs) are used to store data as triples. DBPedia is a large KG which primarly stores encyclopaedic knowledge. It contains Entities and Relations forming a large interconnected Graph. Exploring such a graph is very interesting procedure and can be done in various ways. In this project we will try to explore DBpedia by finding paths between two entities using the python language and LLMs.

Main goal:

We will start from a given (or user selected) entity of DBpedia which will be the source. We will also select a target entity. Our system (program) will try to find a path inside DBpedia connecting the two entities.

Steps:

1. The source entity will have many nodes connected with it. We will select them and ask LLM to produce a score out of them which will depict the affiliation with the target. The biggest the score the highest affiliation with the target.
2. These nodes (according to the score) will be put in a priority queue with the highest scores entities at the higher levels.
3. The top score entity will be popped from the queue and the procedure 1-3 will be repeated
4. The above loop will break a) if the target node is found or b) if a node is found in the path which is semantically quite close to the target based again on the logic of LLM.
5. We can keep the decision of the LLMs about the nodes selected in the path, to describe the “logic” in the connecting path between the source and the target.

Caution: Smart prompt engineering is needed in producing the proximity scores and producing the “logic” of the found path

**Group 1)**

Write a paper collecting the report from the various teams and assemble a final Presentation.

Deadline 1: Related work in the area of interest (1 page with related work) (15 January)

Deadline 2: A presentation assembled with the different presentation of of the other groups

(2 February)

Deadline 3: A paper collecting various reports from other groups (07 February)

**Group 2)**

Produce the python program using chatGPT to actually find the proper path between the two entities.

Deadline 1: Make an initial script connecting to chatGPT produce an initial working demo(15 January)

Deadline 2: Final version of the program deliver it to group 3(20 January)

Deadline 3: A report and a presentation part to be provided to group 1 (25 January)

Group 3)

Produce a python program to find a path between two entities, with iterative broading of the sparql query

For example if you want the path between Einstein and Italy, start with select \* (Einstein ?x Italy) if failed try select \* (Einstein ?x ?y . ?y ?z Italy.} then {Einstein ?x ?y . ?y ?r ?e. ?e ?u Italy and so on)

Deadline 1: Make an initial script initial working demo(15 January)

Deadline 2: Final version of the program deliver it to group 4(20 January)

Deadline 3: A report and a presentation part to be provided to group 1 (25 January)

**Group 4)**

Make a web site based on the code given by group 2 in an environment that supports python (e.g. pythonanywhere) so that a user can test it lively

Deadline 1: Make an initial site on a server ready to accept the code from group 2(15 January)

Deadline 2: Final version of the web site (23 January)

Deadline 3: A report and a presentation part to be provided to group 1 (25 January)