Gregory Knapp

Improvement Assignment Report

As mentioned in class I wanted to focus on improving the way I was extracting values from my knowledge graph. Firstly, I wanted to see if there was a way to create sub-trees from the whole knowledge graph, since it is unreasonable to assume that every character in the game knows everything about the world. After that, I wanted to try out different ways to extracting values from these subtrees that result in more sensical sentences compared to the random selection I was doing before.

The sub-tree work came out better than I could have hoped. I spent a couple hours just playing in the database and doing random queries to get more familiar with Cypher. I knew I was going to have to do some sort of repeated series of queries on relationships and nodes in order to get a sub-tree made so I wanted some practice before coding. I created a Node class that would store node name, node type and child nodes in an array. The child nodes were stored in tuples with the relationship to the parent node as the second value. From there I wrote a series of functions that would take an initial character name and query the database for relationships and connected nodes to that name, parse the values from the result item that Neo4J returns, and append those values to the starting node. Once the starting node had values appended, these nodes were added to a Queue and pushed out one at a time. The query function was recursively called on each node, which built the subtree beyond the starting node. I used either reaching another Person type node or reaching a dead end to be the base cases for the recursion. At the end of the sequence, the starting node for the chosen character is returned. I wrote a print subtree helper function which recursively prints the child nodes. (I’ll add examples below the report) Overall, the function runs fairly quickly, especially if a connection to the database has already been opened before, and having a subtree stored in nodes locally is nice because it could be saved to a character class and accessed anytime from there.

With a subtree made, I adapted my code from before to now extract values using the sub-tree and not querying the database for all nodes. Again, I wrote a series of functions to streamline the templating process. A process sentence function takes the sub-tree head node created using the previous functions, a templated sentence and a parameter of which search method to use. From there, the code calls other helpers to extract tags from the sentences, search the subtree for matching values, and place those matching values back into the templated sentence, returning the now complete quest sentence. I started with just a greedy recursive search where I would search the subtree for nodes of a matching type as the tag that I am trying to replace and return the first match found. This usually returned a closely connected node, but occasionally that was not the case since the first path traversed could sometimes be a bit long to a match. To fix this, I wrote a non-greedy variation of the algorithm that takes in the same parameters but also a recursion level counter and a results array. Instead of stopping at the first match, any matching type nodes are added to the result array in a tuple of the node name and the depth of the recursion. Once the entire tree is searched, the nodes are sorted from smallest to largest in terms of recursion depth and then the lowest depth option is selected. I chose to use this metric as a starting point with the thought that the closer a node is to the starting node, the more “connected” or “closely related” it is to the speaker.

I think there is still a lot more that can be done with this field of knowledge graph extraction but a lot of it will rely on making some decisions as to the way things are stored in the graph and what sort of connections and relationships should exist in it. For example, I am currently stopping at the next person node over, but if we standardize the types of connections in the graph and characters can all have like “strengths” or “weakness” nodes, it would be very easy to change my code to continue building the sub-tree if the connection is of a subset of “generally known” relationships. Or if we wanted to add some randomness to the values being chose and not just defaulting to the closest direct relationship to the speaker, there is room for that too. But I think the code I added has plenty of room and flexibility to modify and add more functionality down the line.

Time spent on tasks: 11 hours.

Link to colab file: <https://colab.research.google.com/drive/1m-DPEKZjLIbjJRfKyL7ujycpYHP0OY-C#scrollTo=dCcDjZMc4A54>

Examples:

Printing a subtree – After making the node this is what the print subtree helper function outputs. I just defaulted to a file tree structure, where the horizontal dashes indicate the depth of the tree (every level of recursion, 2 more – are added).

Each node is printed as [Current Node type]:[Current node name] – [relationship to child node] -> [Child node type]:[Child node name]

|

|

Person:Q'zali - works\_as -> Job:White Mage]

|

|

Person:Q'zali - friend\_of -> Person:Alphinaud]

|

|

Person:Q'zali - friend\_of -> Person:Y'shtola]

|

|

Person:Q'zali - friend\_of -> Person:Alisae]

|

|

Person:Q'zali - type\_of -> Label:Person]

|

|

Person:Q'zali - wants -> Object:Gil]

|

|

--Object:Gil - used\_to\_buy -> Object:Staff]

|

|

----Object:Staff - used\_to\_fight -> Group:Ascians]

|

|

------Group:Ascians - type\_of -> Label:Group]

|

|

----Object:Staff - used\_by -> Job:White Mage]

|

|

----Object:Staff - type\_of -> Label:Object]

|

|

--Object:Gil - used\_to\_buy -> Object:House]

|

|

----Object:House - located\_in -> Location:The Lavendar Beds]

|

|

------Location:The Lavendar Beds - type\_of -> Label:Location]

|

|

----Object:House - located\_in -> Object:Fireplace]

|

|

------Object:Fireplace - type\_of -> Label:Object]

|

|

----Object:House - type\_of -> Label:Object]

|

|

--Object:Gil - type\_of -> Label:Object]

|

|

Person:Q'zali - belongs\_to -> Group:Scions of the Dawn]

|

|

--Group:Scions of the Dawn - fight\_against -> Group:Ascians]

|

|

----Group:Ascians - type\_of -> Label:Group]

|

|

--Group:Scions of the Dawn - type\_of -> Label:Group]

|

|

Person:Q'zali - belongs\_to -> Group:Carpenter's Guild]

|

|

--Group:Carpenter's Guild - located\_in -> Location:The Black Shroud]

|

|

----Location:The Black Shroud - type\_of -> Label:Location]

|

|

--Group:Carpenter's Guild - type\_of -> Label:Group]

|

|

--Group:Carpenter's Guild - oversees -> Job:Carpenter]

|

|

----Job:Carpenter - does\_business\_in -> Location:Limsa Lominsa]

|

|

------Location:Limsa Lominsa - type\_of -> Label:Location]

|

|

----Job:Carpenter - does\_business\_in -> Location:Ul'dah]

|

|

------Location:Ul'dah - type\_of -> Label:Location]

|

|

----Job:Carpenter - makes -> Object:Furniture]

|

|

------Object:Furniture - sells\_for -> Object:Gil]

|

|

--------Object:Gil - used\_to\_buy -> Object:Staff]

|

|

----------Object:Staff - used\_to\_fight -> Group:Ascians]

|

|

------------Group:Ascians - type\_of -> Label:Group]

|

|

----------Object:Staff - used\_by -> Job:White Mage]

|

|

----------Object:Staff - type\_of -> Label:Object]

|

|

--------Object:Gil - used\_to\_buy -> Object:House]

|

|

----------Object:House - located\_in -> Location:The Lavendar Beds]

|

|

------------Location:The Lavendar Beds - type\_of -> Label:Location]

|

|

----------Object:House - located\_in -> Object:Fireplace]

|

|

------------Object:Fireplace - type\_of -> Label:Object]

|

|

----------Object:House - type\_of -> Label:Object]

|

|

--------Object:Gil - type\_of -> Label:Object]

|

|

------Object:Furniture - type\_of -> Label:Object]

|

|

----Job:Carpenter - makes -> Object:Bows]

|

|

------Object:Bows - type\_of -> Label:Object]

|

|

----Job:Carpenter - makes -> Object:Swords]

|

|

------Object:Swords - type\_of -> Label:Object]

|

|

Person:Q'zali - works\_as -> Job:Carpenter]

|

|

--Job:Carpenter - does\_business\_in -> Location:Limsa Lominsa]

|

|

----Location:Limsa Lominsa - type\_of -> Label:Location]

|

|

--Job:Carpenter - does\_business\_in -> Location:Ul'dah]

|

|

----Location:Ul'dah - type\_of -> Label:Location]

|

|

--Job:Carpenter - makes -> Object:Furniture]

|

|

----Object:Furniture - sells\_for -> Object:Gil]

|

|

------Object:Gil - used\_to\_buy -> Object:Staff]

|

|

--------Object:Staff - used\_to\_fight -> Group:Ascians]

|

|

----------Group:Ascians - type\_of -> Label:Group]

|

|

--------Object:Staff - used\_by -> Job:White Mage]

|

|

--------Object:Staff - type\_of -> Label:Object]

|

|

------Object:Gil - used\_to\_buy -> Object:House]

|

|

--------Object:House - located\_in -> Location:The Lavendar Beds]

|

|

----------Location:The Lavendar Beds - type\_of -> Label:Location]

|

|

--------Object:House - located\_in -> Object:Fireplace]

|

|

----------Object:Fireplace - type\_of -> Label:Object]

|

|

--------Object:House - type\_of -> Label:Object]

|

|

------Object:Gil - type\_of -> Label:Object]

|

|

----Object:Furniture - type\_of -> Label:Object]

|

|

--Job:Carpenter - makes -> Object:Bows]

|

|

----Object:Bows - type\_of -> Label:Object]

|

|

--Job:Carpenter - makes -> Object:Swords]

|

|

----Object:Swords - type\_of -> Label:Object]

Greedy vs. Non-greedy filling in:

I just used the same challenge template as before but here’s just an example of the output between the two searching variations.

Greedy:

Keepers of the Hammer : The Ascians had intended to pledge our allegiance to your forces. We know war has come with the Ascians.\n\nHowever, we have been betrayed from within and must now ask for your aid.\n\nA great relic, the Gil of Alphinaud, was stolen from our people by the vile Ascians\n\nWe need your help in recovering it

Non-Greedy:

Keepers of the Hammer : The Scions of the Dawn had intended to pledge our allegiance to your forces. We know war has come with the Scions of the Dawn.\n\nHowever, we have been betrayed from within and must now ask for your aid.\n\nA great relic, the Gil of Alphinaud, was stolen from our people by the vile Scions of the Dawn\n\nWe need your help in recovering it.

Only one of the variables was different in this case, but the fact that it produced a different output made it seem worth it to have the option to choose the closest node.