Sam Jijina

Project One Reflection

For project one, I followed a strict and thoughtful algorithm in order to design an agent that could answer questions as accurately and intelligently as possible. I made use of several concepts learned in lecture in efforts to develop the agent in such a way as to mimic human interpretation of language. In my algorithm, I first began by associating the questions words ‘when’, ‘where’, ‘what’, and ‘how’ with a thematic role respective to each word. I recognized that a user questions beginning with ‘where’ always related to an inquiry regarding a course process, the question word ‘when’ is only associated with a time frame connotation, and the remaining words ‘what’ and ‘how’ can be very open-ended. I then directly related the provided project data types to the question words as follows: ‘what’ and ‘how’ can be related to all data types, ‘where’ is only related to ‘PROCESS’, and ‘when’ is only related to ‘DUEDATE’ and ‘RELEASEDATE’. I then considered the human mind and the common verbiage that could be included in user questions, and made grammatical rules for the agent such that common adverbs, verbs, pronouns, etc would be filtered out from user questions in order to assist the agent in correctly formulating a reply to the user inquiry. Afterwards, I wrote out a few examples of user questions, each beginning with various question words, and hard-coded several common keywords associated with questions concerning all project-specified data types.

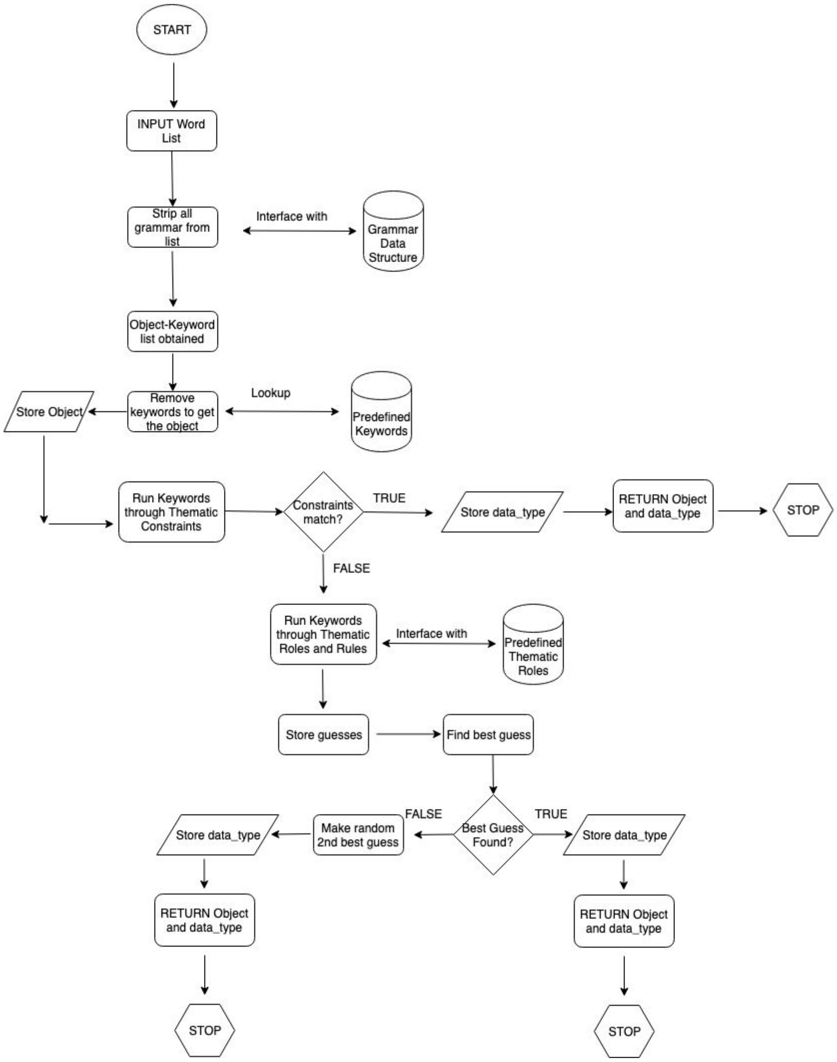
I used the aforementioned logic to ‘teach’ the agent how to make associations with the English language in ways that humans are naturally skilled, and then applied to the agent a much more detailed level of intuition. This algorithm began by accepting a user question and examining the question by removing common words unnecessary for determining the question’s answer. Afterwards, only the first ‘question word’, ‘object’ or primary subject of the sentence, and the ‘keywords’ needed for understanding the inquiry are remaining. Next, the agent reviews the thematic constraints to see if the keyword and question word pair match any exact constraint such that it could intuitively and efficiently provide the correct reply to the user question. If this is not possible, the agent examines the keywords and makes the best possible guess given the predetermined keyword-data type associations. More specifically, my thought process on how to facilitate this procedure was to associate a counter with each data type’s predefined keywords that were determined by thematic roles, and then each time such a known keyword is encountered in a question, increment the respective counter. The agent would then make highly educated guesses on how to reply based on which data type’s counter has the highest value. For example, if the agent was posed with the following user question:

“How long is it until the project deadline?”

the agent would filter out the common English words ‘is’, ‘it’, ‘until’, and ‘the’, with only the question word ‘How’, the keywords ‘long’ and ‘deadline’, and the object word ‘project’ remaining in an array called ‘sentence_list’. The agent would have pre-defined thematic roles that associate the keyword ‘long’ with both data type ‘DUEDATE’ and data type ‘DURATION’ associations, and the keyword ‘deadline’ with only the data type ‘DUEDATE’ association. To prevent the agent from not knowing how to make the best educated guess as to whether the question warrants a ‘DUEDATE’-associated reply or ‘DEADLINE’-associated reply, it would simply recognize that the ‘DUEDATE’ counter is 2 and the ‘DEADLINE’ counter is 1. Thus, the agent would make an educated guess that it should reply with the following: object=project, data type=DUEDATE, and in this case, the agent would be correct and the inquiring student would be satisfied as they now know what day their project is due.

The agent is also able to handle cases such that it does not understand how to accurately reply to the user's question. It does so by randomly returning the last two values of the sentence_list as the object and then randomly selects a data type as a reply to the user's question. The idea behind this method is that I recognized that the object is at the end of a user's question more often than it is at the beginning, so by selecting the last two items of the sentence_list to be the object and regarding the other remaining items as keywords, the agent is making the most educated guess possible.

It is apparent that the overarching purpose of this agent is to return an object and data type from a user's question, and this constitutes an appropriate reply. It can be inferred that these returned values are used by the project code provided to us to create frames presumably for use in future assignments. So, while my implementation has not used frames explicitly, it is evident that the agent itself is creating a frame using the data returned from the provided input_output function of this project. Below is a block diagram of my overall implementation of the agent for this project.


START
INPUT Word List
Strip all grammar from list
Interface with
Grammar Data Structure
Object-Keyword list obtained
Store Object
Remove keywords to get the object
Lookup
Predefined Keywords
Run Keywords through Thematic Constraints
Constraints match?
TRUE
Store data_type
RETURN Object and data_type
STOP
FALSE
Run Keywords through Thematic Roles and Rules
Interface with
Predefined Thematic Roles
Store guesses
Find best guess
Store data_type
Make random 2nd best guess
FALSE
Best Guess Found?
TRUE
Store data_type
RETURN Object and data_type
RETURN Object and data_type
STOP
STOP

I took note of 10 total questions that stood out to me, 5 of which the agent failed to answer and 5 of which the agent was able to accurately formulate a reply.

The following questions that passed are listed below:

"How can I submit space shuttle"
"What is the location of space shuttle"
"What fraction of my average is project 1"
"What is the weight of project 1"
"When is space shuttle due"

The following questions that failed are listed below:

"How long until space shuttle is out"
"Where is project 1 due"
"When can I expect space shuttle"
"How long can we work on project 1"
"Where can I get space shuttle"

In all of the above questions, the agent was able to extract the question's object accurately. However, for the questions it failed to answer, it was not able to accurately return the data type. In the cases the agent failed, it had to randomly identify the data type from the five provided data types, giving it a one in five chance of providing the correct reply. This shows that questions that used less common phraseology were more likely to result in an inaccurate reply by the agent, whereas questions that used common words and were more clearly defined likely led to a correct response.

In conclusion, this project taught me a lot about the use of thematic roles in designing AI as well as how to decipher the English language in a technical way. Given more time, I would have liked to come up with a better way for the agent to have been able to more intelligently handle questions that contained unrecognizable keywords. I would have liked to see how that would have affected the agent's responses, but I am looking forward to implementing such logic in the next project.