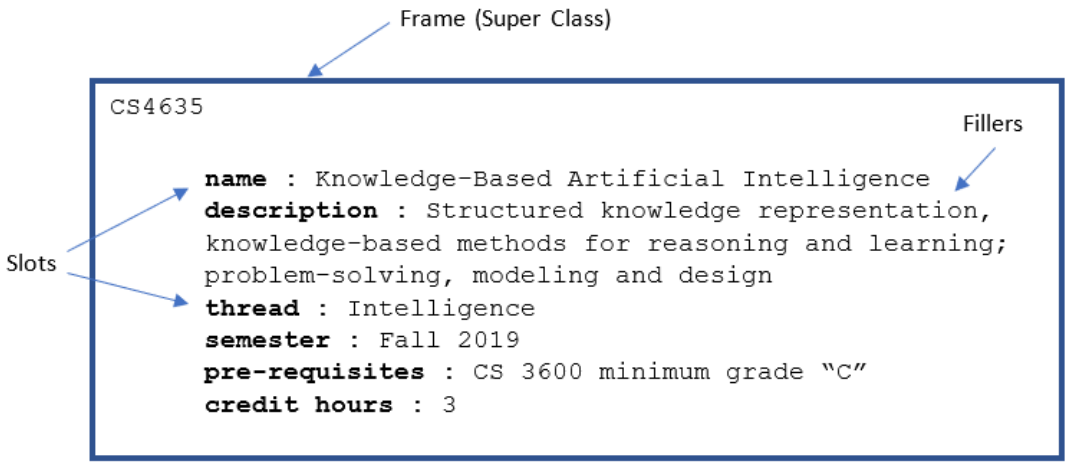
Zachary Waters
Ashok Goel
CS 4635 Knowledge-Based AI
September 2, 2019

Assignment 1

The objective of this assignment is for me to explain how I would use frames to design a chatbot. This program should be able to respond and answer questions about this class. Before I continue, I believe a disclaimer is needed: I am unfamiliar with the details of implementing a chatbot, and as such my attempt at explaining how I would go about designing such a program will be rather overgeneralized. However, I believe this paper will succeed in offering a launching-off point for how I would approach the future programming project. To return to the original topic, I believe that there are two main problems when creating a chatbot. The first of these is acquiring and efficiently storing all the relevant domain information in a way that allows for optimized retrieval. The second problem is creating a system that can parse user questions into a format that is capable of being handled by the software. As expanded upon further in this paper, frames provide a step in solving these problems.

To tackle the first problem, we need to determine the domain of information that our program will be expected to operate within. As previously established, our chatbot will be limited to answering questions about our class, so "CS 4635" is our domain. Now that the boundaries of the domain have been established, the next step is fleshing out the information that populates our domain space. This is the information that our users will be requesting through their questions. Some examples of the anticipated information the users will be asking for include dates of homework and exams, contact information for teaching assistants (TAs), and the

class grading criteria. This gets us to the crux of the problem: we need to find a way to store all relevant information for later retrieval to answer the user's questions. Frames help provide a solution to this by acting as a knowledge structure that can orderly represent the domain objects and their corresponding qualities in a commonsense manor. Frames are a data structure that divide knowledge into substructures that represent stereotyped situations. These substructures are referred to as slots, with each taking a value called a slot filler. To get a better understanding of frames and slots, an example frame is shown below for our CS 4635 domain:



From the above image you can see that CS 4635 is the super class, which is the subject of the frame. For each of its slots there is an attribute type, with the corresponding filler information. The actual code-implementation of such a frame could be accomplished by using a graph-based structure. A node like object can represent our super class, inside the node an array list could be contained. Where each of the indices point to a node that contains the attribute that the object possesses. Frames also have the additional property of being able to be used hierarchically. This means a frame could have "child" sub-frames which, depending on the rule system is implemented, can inherit values from the parent or be automatically filled with default

values. For example, the TA frame could be a child frame that inherits some of its slot information from the Professor frame, such as office hours, contact information, and meeting location. Using inheritance and default values would allow me to quickly and efficiently skip slots that would be repeated otherwise. Therefore, to summarize an answer to the original question, I would manually create a frame for each object in the domain that could be the subject to a user's question. I would likely draw out this information from the syllabus, and I would use inheritance to streamline the creation of certain repetitive object frames.

Now that we know how the information in the system will be stored, I can address the next problem: how to determine what information to find and return in response to a question. Typically, a chatbot must extract three things from a user's question in order to provide the appropriate answer: domain classification, intent determination and slot filling (Daniel Jurafsky & James H. Martin, 2008, Chapter 24, p. 13). Domain classification is determining what domain the subject's question falls under. In our follow-on project we only have one domain to worry about, CS 4635. Intent determination is trying to figure out what general task or goal the user is trying to accomplish with their question. For the questions the project covers, intent determination refers to the frame the user is seeking to draw information from. Slot filling is figuring out what slots and fillers the user intents to draw information from. So, considering we don't have to worry about domain classification, we are left with the task of trying to parse out the intent determination and slot filling for each user's question. Normally this would be an incredibly daunting task, having to anticipate a nearly infinite number of possible questions that could be asked. However, the project's scope makes this more manageable as the number of slot fillers is limited to only five: "DUEDATE", "RELEASEDATE", "WEIGHT", "PROCESS" and "DURATION". This means we then need to figure out which of the five we are dealing with and

find the object frame the question is referring to. We can use a combination of parsing tricks and semantic analysis to figure out the answer to these final questions. For example, we can manually look for keywords such as “due” to determine that we are dealing with a “DUEDATE” type. We would then rely on thematic roles to determine the object and frame we are looking for. Now that we know both the frame and slot, we can simply look though the collection of frames until we find the corresponding information, and then return it in a textual form back to the user.

Bibliography:

Daniel Jurafsky & James H. Martin. (May 16, 2008). Speech and Language Processing. Prentice Hall; 2nd edition