Mini-Project 3: Sentence Reading

David Adamashvili
dadamashvili3gatech.edu

Abstract— In this paper I will outline my solution for the Sentence Reading problem. We are given a sentence containing some knowledge and a question. We want to return an answer to the question using the information contained in the sentence.

1 AGENT DESCRIPTION

My agent uses a case-based reasoning approach. First, I looked through the types of questions that were being asked and came up with various heuristics and rules to see which type of question is being asked and then answer accordingly. These classes are the following:

1. Question contains the word "**color**" – look for colors in the sentence and return the first one
2. Question contains the word "**animal**" – look for animals in the sentence and return the first one
3. Question contains either the word "**when**" or "**time**" – look for a date-time substring (**08:00AM**) in the sentence and return it. If there's no such substring, look for time words (such as "morning", "night"…)
4. Question contains the word "**what**" – look for the first **noun** in the sentence and return it. Give priority to the nouns that don't appear in the question.
5. Question contains the word "**where**" – look for location words ("school", "room", "farm"…) in the sentence and return the first one
6. Question contains the word "**who**": - look for proper names ("Jim", "boy", "children"…) in the sentence that are not in the question and return the first one
7. Question contains the word "**how**" and the word that follows it is:
    a. **Far**: look for a distance word in the sentence ("inch", "mile"…)
    b. **Long**: if there is a time word in the sentence ("hour", "second"…) return it, otherwise return a distance word.
    c. **Many**: look for a number word in the sentence ("hundred", "three"…) and return it

d. **Big**: look for a size word in the sentence ("large", "small", "big"…) and return it
e. Otherwise return the first **verb** in the sentence

Everything except for verbs and nouns were classified and put into lists by me. However, there were too many nouns and verbs to filter by hand, so I used the Python library Spacy to preprocess the list of words and put them in a dictionary along with their labels. I then copied this dictionary into my agent file.

2 PERFORMANCE AND EFFICIENCY

2.1 Performance

My agent received a 38/40 score on the question set. I was getting 34-s and 36-s but after some reruns and tweaking of word lists I managed to get it to a 38. Subsequent runs often give a lower score, so I guess I got lucky in a sense (but I do have more than a couple of dozen submission attempts left, so I could probably get a 40 if I got lucky).

One major flaw of my solution is that it always returns the first instance of the search word that it comes across. This is a problem in questions like the following:

Sentence: "The white dog and the blue horse play together"

Question: "What color is the horse?"

My agent will see the word "color" in the question, so it will start looking for colors in the sentence. It will find the words "white" and "blue", but it will return the first one, "white", even though the correct answer would be "blue".

To possibly solve this, we could start looking for nouns in the question, to find out what is the object whose color we are looking for. Then we could look for pairs of words next to each other such as "white dog" and "blue horse", and get to the answer from there.

2.2 Efficiency (Big O)

The asymptotic complexity of my solution is linear, **O(n)**, where n is the number of characters in each sentence, if we disregard the preprocessing. Since there are only a finite amount of checks being done on each question, and a linear search of substrings is done on the sentence in each check, the total time complexity comes out to be linear.

The space complexity is whatever space is occupied by the preprocessed dictionary and the hand-written lists (locations, sizes, times etc.), which is constant **O(1)** as they never change during the run of the algorithm.

3 HUMAN COMPARISON

I cannot exactly explain how a human would answer these questions. To me, for example, the answer comes naturally without much thought. However, when contemplating the answer to a question, I need to first look at the question word such as when, how, what etc. and based on the context find out exactly what it's asking. Then I would go back to the sentence and look for this answer. In theory, this approach is similar to the one outlined in this paper, but more refined. For example, if I saw the word "animal" in the question, I would not be immediately looking for a type of animal in the sentence. I would probably read the whole question and then look at the context. Maybe the question is asking "What types of animals eat grass?", and the answer might be "herbivores".

The way humans approach these types of questions is no doubt much more optimal than how my algorithm does. A human would always outperform my agent on more general question sets, as my agent is **over-fitted** on the given dataset. This is because the way I solved this problem is to look for what types of questions my agent was failing on and then used case based reasoning to create checks to see if a given question fit one of the previously seen patterns.

As for the efficiency of the human algorithm, I think it would be linear, as I only read each word once. However, I don't know what kind of complex brain activity is going on behind the scenes, so I can't be sure.