KBAI

Mini-Project 3: Sentence Reading (Spring 2021)

In this project, you'll implement an agent that can answer simple questions about simple sentences made from the 500 most common words in the English language, as well as a set of 20 possible names and properly-formatted times. Your agent will be given a sentence and a question, and required to return an answer to the question; the answer will always be a word from the question. You will submit the code for answering these questions to the Mini-Project 3 assignment in Gradescope. You will also submit a report describing your agent to Canvas. Your grade will be based on a combination of your report (50%) and your agent's performance (50%).

About the Project

In this project, you'll be given pairs of sentences and questions. Your agent should read the sentence, read the question, and return an answer to the question baesd on the knowledge contained in the sentences. Importantly, while this is a natural language processing-themed project, you won't be using any existing libraries; our goal here is for you to understand the low-level reasoning of NLP, not merely put existing libraries to work.

To keep things relatively reasonable, your agent will only be required to answer questions about the 500 most common words in the English language, as well as a list of 20 possible names. Your agent should also be able to interpret clock times: you may assume these will always be HH:MM(AM/PM) or simply HH:MM. For example, 9:00AM, 11:00, or 12:34PM.

Because there are disagreements on what the most common words are, we've given you our own list of the 500 most common words for our purposes, along with the 20 names your agent should recognize: these are contained in the file mostcommon.txt.

Your Agent

To write your agent, download the starter code below. Complete the `solve()` method, then upload it to Gradescope to test it against the autograder. Before the deadline, make sure to select your best performance in Gradescope as your submission to be graded.

Starter Code

Here is your starter code (and the mostcommon.txt file): SentenceReadingAgent.zip. You may also access the code from the course Github repository.

The starter code contains two files: SentenceReadingAgent.py and main.py. You will write your agent in SentenceReadingAgent.py. You may test your agent by running main.py. You will only submit SentenceReadingAgent.py; you may modify main.py to test your agent with different inputs.

Your `solve()` method will have two parameters: a string representing a sentence to read, and a string representing a question to answer. Both will contain only the 500 most common words, the names listed in that file, and/or clock times. The only punctuation will be the last character in the string, either a period for the sentence or a question mark for the question.

For example, an input sentence could be:

- “Ada brought a short note to Irene.”

Questions about that sentence might include:

- “Who brought the note?” (“Ada”)
- “What did Ada bring?” (“note” or “a note”)
- “Who did Ada bring the note to?” (“Irene”)
- “How long was the note?” (“short”)

Another input sentence could be:

- “David and Lucy walk one mile to go to school every day at 8:00AM when there is no snow.”

Questions about that sentence might include:

- “Who does Lucy go to school with?” (“David”)
- “Where do David and Lucy go?” (“school”)
- “How far do David and Lucy walk?” (“mile” or “one mile”)
- “How do David and Lucy get to school?” (“walk”)
- “At what time do David and Lucy walk to school?” (“8:00AM”)

You may assume that this second example will be the upper limit of complexity you may see in our sentences.

Returning Your Solution

Your `solve()` method should return an answer to the question as a string. You may assume every question will be answerable by a single word from the original sentence, although we may accept multi-word answers as well (such as accepting "mile" and "one mile" above).

Submitting Your Solution

To submit your agent, go to the course in Canvas and click Gradescope on the left side. Then, select CS7637 if need be.

You will see an assignment named Mini-Project 3. Select this project, then drag your SentenceReadingAgent.py file into the autograder. If you have multiple files, add them to a zip file and drag that zip file into the autograder.

When your submission is done running, you'll see your results.

How You Will Be Graded

Your agent will be run against 20 question-answer pairs. The first eight will always be the same; these are the eight contained within the main.py file provided above. The remaining 12 will be randomly selected from a large library of sentence-question pairs.

You can earn up to 40 points. You will earn 2 points for each of the 20 questions you answer correctly.

You may submit as many times as you want prior to the deadline. You **must** select which of your submissions you want to count for a grade prior to the deadline. Note that by default, Gradescope marks your last submission as your submission to be graded. We cannot automatically select your best submission. Your agent score is worth 50% of your overall mini-project grade.

Your Report

In addition to submitting your agent to Gradescope, you should also write up a short report describing your agent's design and performance. Your report may be up to 4 pages, and should answer the following questions:

- How does your agent work? Does it use some concepts covered in our course? Or some other approach?
- How well does your agent perform? Does it struggle on any particular cases?
- How efficient is your agent? How does its performance change as the sentence complexity grows?
- Does your agent do anything particularly clever to try to arrive at an answer more efficiently?
- How does your agent compare to a human? Do you feel people interpret the questions similarly?

You are encouraged but not required to include visuals and diagrams in your four page report. The primary goal of the report is to share with your classmates your approach, and to let you see your classmates' approaches. You may include code snippits if you think they are particularly novel, but please do not include the entirety of your code.

Submission Instructions

Complete your assignment using JDF, then save your submission as a PDF. Assignments should be submitted to the corresponding assignment submission page in Canvas. You should submit a **single** PDF for this assignment. This PDF will be ported over to Peer Feedback for peer review by your classmates. If your assignment involves things (like videos, working prototypes, etc.) that cannot be provided in PDF, you should provide them separately (through OneDrive, Google Drive, Dropbox, etc.) and submit a PDF that links to or otherwise describes how to access that material.

This is an individual assignment. All work you submit should be your own. Make sure to cite any sources you reference, and use quotes and in-line citations to mark any direct quotes.

Late work is not accepted without advanced agreement except in cases of medical or family emergencies. In the case of such an emergency, please contact the Dean of Students.

Grading Information

Your report is worth 50% of your mini-project grade. As such, your report will be graded on a 40-point scale coinciding with a rubric designed to mirror the questions above. Make sure to answer those questions; if any of the questions are irrelevant to the design of your agent, explain why.

Peer Review

After submission, your assignment will be ported to Peer Feedback for review by your classmates. Grading is *not* the primary function of this peer review process; the primary function is simply to give you the opportunity to read and comment on your classmates' ideas, and receive additional feedback on your own. All grades will come from the graders alone.

You receive 1.5 participation points for completing a peer review by the end of the day Thursday; 1.0 for completing a peer review by the end of the day Sunday; and 0.5 for completing it after Sunday but before the end of the semester. For more details, see the participation policy.