Project 1

Background

Continuing Assignment 1, Project 1 is mainly focused on solving the internal problem of designing a chatbot that recognizes the object and data requested of different prompts. Some of the sample prompts include ‘What is the project 2 release date?’ or ‘When must order 5 be turned in?’

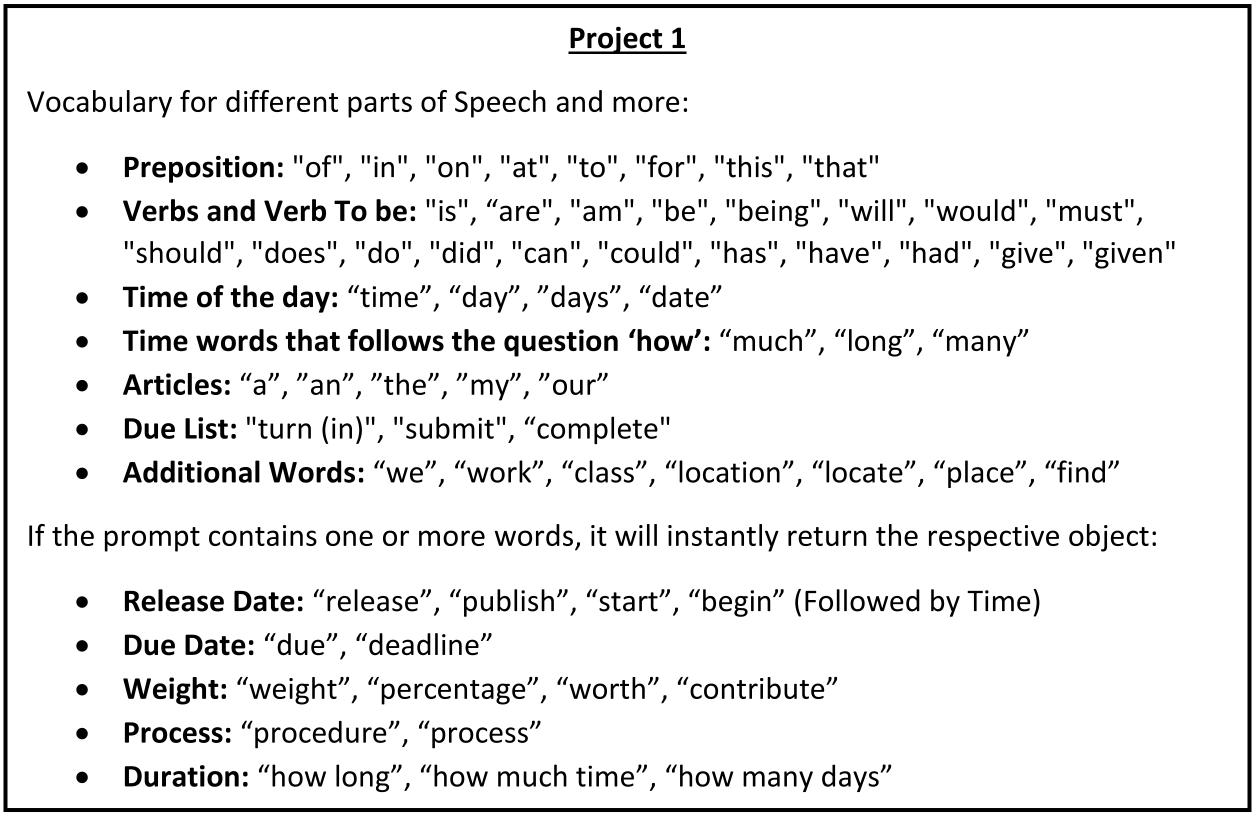
First, it is assumed that the chatbot handles the problem space within the class (CS 4635 Fall ’18) and the vocabulary of the bot is accustomed to words commonly used in a class setting. More details on the Strengths and Weaknesses section.

Second, as per the requirements the bot only needs to return the ‘data requested’ and ‘object requested’. The data requested can be of 5 different prompts of ‘Release Date’, ‘Due Date’, ‘Duration’, ‘Weight’ and ‘Process’. The object requested can range to anything from ‘Project 2’ to ‘Order 5’ in the above example.

It is also important to notice that all prompts must start with one of ‘What’, ’When’, ’Where’ or ‘How’ as per the project requirements.

Algorithms Breakdown

Going over the program on a very high level, this section will outline the different cases and vocabulary that are hardcoded into the program. This gives the bot a better understanding of different parts of speech and utilize frames and thematic roles better. More details on Fig. 1 and Fig. 2.



Project 1

Vocabulary for different parts of Speech and more:

- **Preposition:** "of", "in", "on", "at", "to", "for", "this", "that"
- **Verbs and Verb To be:** "is", “are", "am", "be", "being", "will", "would", "must", "should", "does", "do", "did", "can", "could", "has", "have", "had", "give", "given"
- **Time of the day:** “time”, “day”, ”days”, “date”
- **Time words that follows the question ‘how’:** “much”, “long”, “many”
- **Articles:** “a”, ”an”, ”the”, ”my”, ”our”
- **Due List:** "turn (in)", "submit", “complete"
- **Additional Words:** “we”, “work”, “class”, “location”, “locate”, “place”, “find”

If the prompt contains one or more words, it will instantly return the respective object:

- **Release Date:** “release”, “publish”, “start”, “begin” (Followed by Time)
- **Due Date:** “due”, “deadline”
- **Weight:** “weight”, “percentage”, “worth”, “contribute”
- **Process:** “procedure”, “process”
- **Duration:** “how long”, “how much time”, “how many days”

Figure 1 Hard-coded vocabulary with respective labels for the bot

<u>Question Words</u>

- **What**:
  - **Default Return**: Data = Weight
  - **If the next word is part of *Time:*** Data = Release Date
- **Where:**
  - **Default Return:** Data = Process
- **When:**
  - **Default Return:** Data = Release Date
  - **If the sentence contains word(s) that is part of the *DueList***: Data = Due Date
- **How:**
  - **Default Return:** Data = Process
  - **If followed by part of *Time***: Data = Duration

Figure 2 Different Question Words with Default Returns and various cases

As mentioned previously, the hard-coded vocabulary in the bot is used to apply thematic roles to every prompt and classify different words to different parts of speech. The bot knows for sure that the object or data requested cannot be a preposition, article or verb. Furthermore, the bot understands that based on different question words, it defaults the data return to the respective solutions and handle different cases based on the words in the sentence as outlined in Figure 1 and 2.

Case 1: Articles next to Object

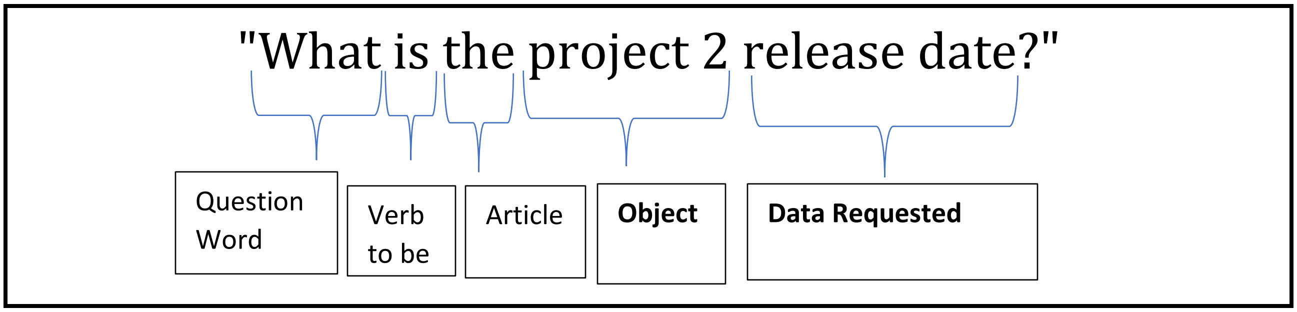


Figure 3 Sample Prompt of Case 1 asking the release date of Project 2

To give further context, let's break down this example. This is one of the simplest case of prompts and on a very high level, the answer to this should be 'Release Date' as the data requested and 'Project 2' as the object requested. The bot firstly understands that the question word in this case is 'What' and the word is not a part of time, hence it defaults the **data** return to be **weight**. By using frames and thematic roles, we can classify the different parts of sentence into groups of 'Question', 'Verb To Be', 'Article', 'Object' and 'Data Requested'. For instance, 'is' would be part of the 'Verb To Be' group and 'the' would be part of 'Article'. We also know that 'release date' is part of the escape words that would instantly return **data** as **release date** as per Fig 1. Generally, the object would come after an article and hence we check the next words which are 'Project 2' and they are not part of any parts of speech or escape words and hence the bot returns **Project 2** as the **Object**.

This resembles the way humans think in a way by firstly classifying the different words of the sentence into different groups and parts of speech. We know for sure ‘What’ is a question word and so on, but Project 2 does not belong in any group of the part of speech, but it is after an article and it is the likeliest to be the object. And as soon as we see the phrase ‘release date’, we know for sure that the question is asking about a timeframe of a release date. Project 2 is naturally the object since it is ‘unknown’ by process of elimination and classifying the other words into different groups. Obviously, this is very prone to mistakes into misleading a word next to an article as the object in sentences like ‘Where is the location of project 1 specified?’ More on this case will be on the Strengths and Weaknesses section.

Case 2: No Articles

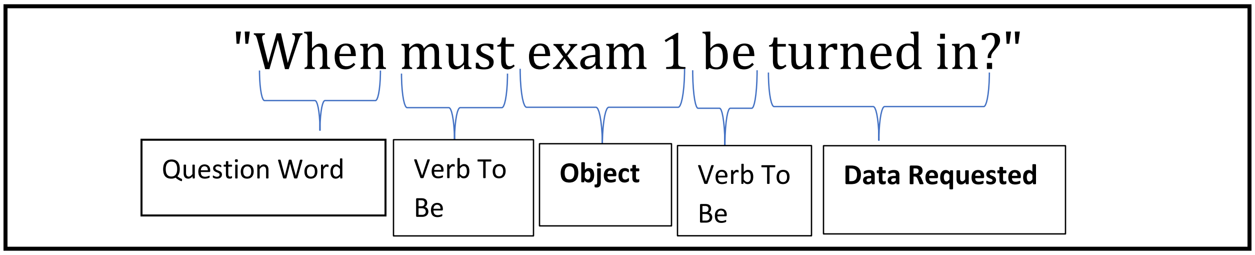


Figure 4 Second example for Case 2 for prompt with no articles

Now examine this slightly more complicated because there are 2 words in ‘Verb To be’ and there is no known article in the sentence. Using a similar approach, we will break down the sentence into different frames such as putting ‘When’ as the question. The bot immediately knows that it is asking for time and it is defaulted to **Release Date**. The bot then classifies ‘must’ and ‘be’ to the same group of ‘Verb to be’ via thematic roles in the sentence and we are left with ‘exam 1 turned in’. Notice that ‘turn in’ is part of the escape word for due when the question word is ‘when’ since we know for sure that it is asking for a timeframe and so we override the data requested to **Due Date**. We are left with **exam 1** and since it is again ‘unknown’, we classify it as the **object.**

With a human’s eyes, we immediately recognize the word ‘when’ asks for a time frame and similarly classify ‘must’ and ‘be’ as ‘To be’. Humans also know that turned in means it is asking for the time it is due from the meaning of the words and because the question begins with ‘when’. Again, exam 1 being left naturally becomes the object.

Test Questions

Sample prompts include the Piazza thread about the valid inputs that meets the scope of the bot and some self-made questions if I were to ask something to a class bot. This includes changing the objects, data, sentence structure, words and extreme cases to handle various inputs. It is very important to note the scope of the bot and what questions the bot can handle.

Sample Test Questions

Basic Questions

- When is the project due?
- When is exam 1?
- When must order 3 be turned in?

Advanced Questions

- What day should we turn the project in?
- How much time do we have to submit the order?
- When is the project1 for this class due?
- How do we turn in our project2?
- Where is the location of project 1?

Figure 5 Sample Basic and Advanced Test Questions

Strengths

Some of the strengths of this bot includes being very sensitive to words that trigger various data returns, such as release (date) signifying the data as release date or 'percentage' signifying weight of something.

Another strength that the bot has is the speed that the bot operates to different queries due to its very limited vocabulary and scope.

In addition, the bot also does a good job of returning the right object by eliminating all the words in its vocabulary/ part of speech especially when there is no article. For example, a prompt such as 'Where is the location of project 1?' can be handled properly by recognizing the question word, the known words and parts of speech and eliminating all of them and hence left with project 1 as the object.

Weaknesses

The bot has very limited and static scope and vocabulary as mentioned in the first paragraph and they are all hard coded. Anything left behind or outside the scope can get very messy very easily.

The other main weakness is handling cases with two articles or an article that is not next to the object as mentioned previously in cases like 'Where is the location of project 1 specified?' This is again due to the limited vocabulary of the bot and it has a difficult time differentiating whether a word is an appropriate object or not if there are more than 1 choices. In an ideal world, the bot must understand every part of the sentence and understand what role they play for the sentence significantly improving the bot's accuracy.