Programming Assignment: Siri – Software Engineer
Author: Ryan Lacerna

Time spent:

- Started on Saturday(27th May) due to M.Sc exams and full-time work
- 2.5 coding
- 1.5 hour thinking about the problem and solution
- 1 hour refactoring and unit-tests

Technology:

Language : C#
Framework: .Net Core, version 2.1.200
IDE: Visual Studio Code
OS Platform: MacOS High Sierra 10.13.4

Project Setup:
Installing .NET SDK & IDE:
<https://www.microsoft.com/net/learn/get-started/macos>

Solution Structure:

- SpeechText
    - ProgrammingAssignment
        - Interface
            - **IAnalyserService.cs**
            - **IDataService.cs**
        - Services
            - **AnalyzerService.cs**
            - **DataService.cs**
        - **User.cs**
    - ProgrammingAssignment.Test
        - **Test.cs**
    - ProgrammingAssignment.sln

Running the solution and unit-tests:

Solution:

1. Using your terminal navigate to the *ProgrammingAssignment* directory
2. Type and execute: *dotnet run*
3. You should be presented with the results on your terminal

Unit-Tests:

1. Using your terminal navigate to the *ProgrammingAssignment**.Test*** directory
2. Type and execute: *dotnet test*
3. You should be presented with the test results

Assumptions and If to be taken into live production:

One assumption, is that utterances are coming from the one user sequentially, so I've used a list and iterate through it to simulate one user that inputs utterances.

There are lots of ambiguous words (e.g BBQ, thai) that may come up from user utterances. I believe that these will need to be given specific properties using an ontology for the Concepts, so that we can retrieve a more semantic result. In our current solution, they are mainly in the context of food, and we have no ontology.

For example 1: the word 'BBQ', could also mean 'Barbeque'

For example 2: the word 'Thai', could mean more than just food, it can be a thai person, thai airlines.

For our solution we have just implemented one assumption, the word 'East' may have another word associated with it e.g 'East Asia', so I've created a set to check if any of the context contains these word and find associations.

For future work I would create an ontology to define concepts that shows properties and relationships, this may generate inferences that can give a more accurate and semantic result. E.g use of Protégé, OWL, RDFS

We currently have not implemented a data-store (e.g database) to get the Concepts from, so I have used collections to represent our test data. This mean that we need to iterate over the data which increases the computational complexity.
If to be put to live production I'll use an in-memory DB for quick data retrieval and lookups for O(1).

At the moment our data retrieval is synchronous and iterate over a collection, this is not a valid solution where you may have have millions of entries.
Make the data retrieval services and data processing services asynchronous to handle more requests.

At the moment, given the time constraint, we have written unit tests but it does not cover a wide range of scenarios. To test for more reliability, I would write more scenarios based cases for the unit –tests to cover as many braking possibilities that may be.