A task-oriented Chatbot

Team members: Aashika Padmanabhan, Nikita Pardeshi, Mitali Shah, Shazia Sulaiman

INTRODUCTION

Chatbots are increasingly seen as a valuable complement to customer service. About 23% of service companies have started using chatbots and about 80% of us have interacted with it one way or another. They can handle around 70% of full conversations, making it a perfect candidate for an NLP project.

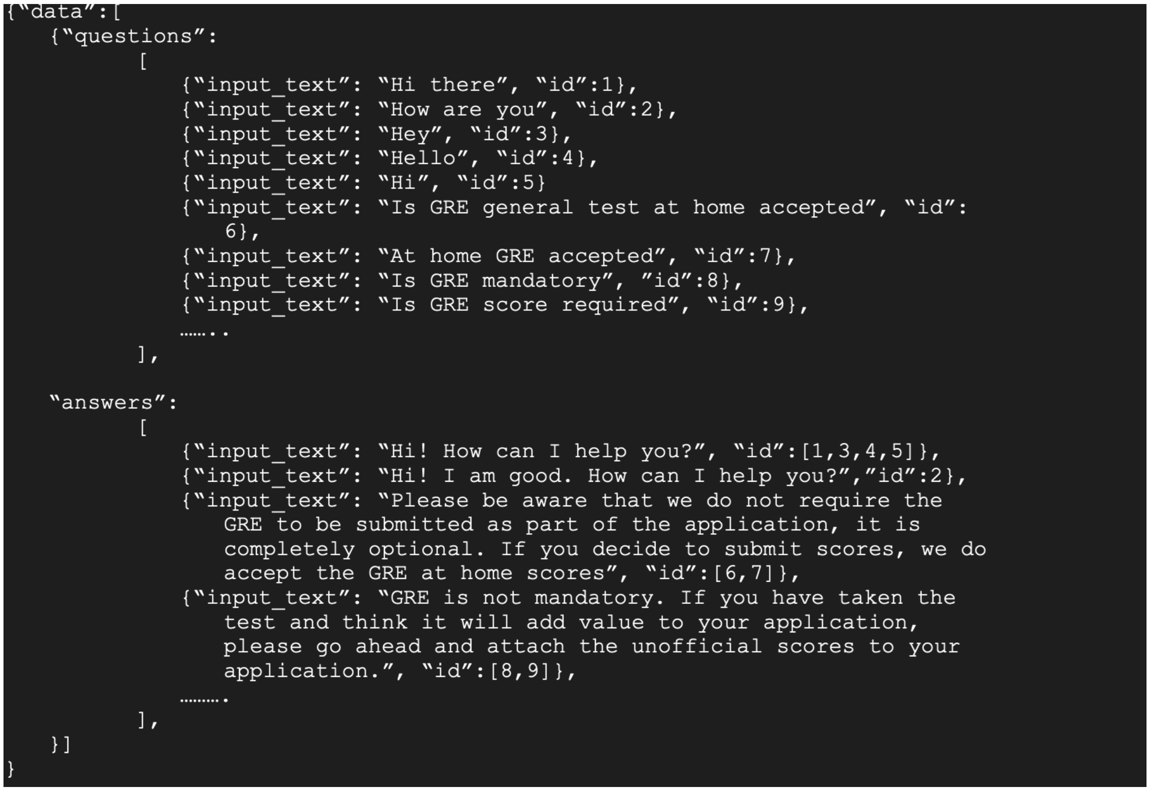
Problem statement: The current system at Georgetown DSAN program has to answer any question that prospective students and new admits have is a manual one. Students have the option to either find answers from the FAQ section or if their question is not covered in the FAQ , they can send questions via email and wait for the student ambassadors to answer them. This process is time consuming and on average takes 2-3 days. Time to get clarity impacts students' decision making especially when deadlines are close by. Our program could end up losing prospective students and also cause anxiety to new admits especially if they have questions that are time sensitive.

Solution: We believe an automated chatbot can help solve the above problem and help provide answers to most of their questions in a much quicker way.

Our idea is to leverage NLP principles to study the most popular questions and choose the best answer from a predefined set of answers. This reduces the workload on students and helps provide instant responses to 80% of the questions.

DATA PREPARATION

We intend to use a predefined set of common questions and answers taken from the gradanalytics inbox after approval from concerned authorities. Each data point under both lists of questions and answers is a dictionary consisting of a question identified by a unique identifier which is used to identify the most appropriate answer.



```
{“data”:[
   {“questions”:
         [
            {“input_text”: “Hi there”, “id”:1},
            {“input_text”: “How are you”, “id”:2},
            {“input_text”: “Hey”, “id”:3},
            {“input_text”: “Hello”, “id”:4},
            {“input_text”: “Hi”, “id”:5}
            {“input_text”: “Is GRE general test at home accepted”, “id”:
               6},
            {“input_text”: “At home GRE accepted”, “id”:7},
            {“input_text”: “Is GRE mandatory”, ”id”:8},
            {“input_text”: “Is GRE score required”, “id”:9},
            ……..
         ],

   “answers”:
         [
            {“input_text”: “Hi! How can I help you?”, “id”:[1,3,4,5]},
            {“input_text”: “Hi! I am good. How can I help you?”,”id”:2},
            {“input_text”: “Please be aware that we do not require the
               GRE to be submitted as part of the application, it is
               completely optional. If you decide to submit scores, we do
               accept the GRE at home scores”, “id”:[6,7]},
            {“input_text”: “GRE is not mandatory. If you have taken the
               test and think it will add value to your application,
               please go ahead and attach the unofficial scores to your
               application.”, “id”:[8,9]},
            ……….
         ],
   }]
}
```

MODELS

This is a task-oriented/ retrieval chatbot which is trained to provide the best possible response from a database of predefined responses. Retrieval-based chatbots use techniques like keywords matching, machine learning or deep learning to identify the most appropriate response. We intend to look at two different models to build the chatbot and select the one with higher accuracy.

Model 1: The Sequential Model - Keras

To implement this model we first need to tokenize the data and then use lemmatization to create a bag of words. We then pass the bag of words to a deep neural network built using Keras sequential. The neural network predicts the ‘ID’ of the input which can be then used to locate the response. The benefit of using this method is that the neural network can be customized to change the number of layers, number of neurons and activation functions to get better accuracy.

Model 2: BERT (BertForQuestionAnswering)

One of the important reasons to use BERT is its capability to understand the full context of a word by looking at the words that come before and after it, which helps to understand the intent behind the query asked. Here, we intend to use the BertForQuestionAnswering model by HuggingFace.

MODEL EVALUATION

The quality of chatbot conversations is key to good chatbot user experience. In a production environment, CTR, chatbot activity volume, bounce rate will be a good evaluator. As it is not applicable for this project, we will attempt at the following :-

1. Repeated expected baseline behavior - Verify if the chatbot provides the same answer for different worded but similar questions.
2. Model performance on validation data

BIGGEST CHALLENGES

Computational/ Hardware considerations

At this point, we do not anticipate additional computational/hardware resources. However, our contingency is to leverage google colab notebook with GPU accelerator if it were to run into compute inadequacy.

Unknowns that dictate the success or failure

1. Handling of different informal slang/contractions, typos
2. Handling of questions the bot has not seen before and decides to not answer the question and route to the email group.

RESULT

We intend to use Slide deck + Oral presentation of the project.