

Purpose

To build a chatbot in telegram according to emergency evaluation methods used by doctors.

Code structure

The chatbot set consist of a set of libraries that supports the main telegram bot code in the file chatbot_telegram.

Function of supporting libraries:

NewConditionQuestions: To produce a sequence of questions provided by doctor in accordance to emergency evaluation methods. The recording functions that access stored data and then record them onto PDF and TXT files are also found here. It is supported by the PDF libraries found in reingart-pyfpdf-cfe3d03 to help document the conversation into PDF file. The main libraries used for the chatbot's recording function are moved out to the Final Chatbot set file.

These files consist of:

`_init_`

Fonts

Fpdf

Html

Php

Py3k

Template

ttfonts

RasaNLU: The NLP capabilities of the chatbot comes from this library. It is used to train the code and also to load the learnt model into the chatbot. Any new models trained will appear in the projects folder under default. The other 2 rasafiles are rasa libraries called into RasaNLU file to train and load the models. The training file must be in json format which can be easily made using <https://yuukanoo.github.io/tracy/#/agents>. The chatbot current training file is demo-rasa1 file found in Final Chatbot set.

Workings of Code

Chatbot_telegram:

Recording

Telegram API stores information in a conversation through python dictionary user_data. Information of conversation are stored using user_data["relevant" name]= text. At the end of a conversation, the recording function is called in to produce the content in a pdf file.

Keeping track of conversation conditions

The user_data dictionary is also used to keep track of the state of the conversation, to bring over relevant information over to the next state of the conversation.

E.g user_data[helpfulcount] under helpful is used to keep track on whether the user is going through the conversation the first or second time, asking relevant questions again the first time or ending the conversation the second time.

Redflag questions

Redflag questions are questions where if answered correctly, will indicate that a redflag situation is present. These situations might have age and previous conditions present for the redflag situation to arise. Conditions are remembered in user_data and conditioned using a list where these conditions are stored.

E.g under function ProcessConditions1, for gastroenteritis question, for number of times a patient vomit questions if the patient age is under 6months, and vomit/diarrhea more than 3 times in a day, it is an emergency situation. These conditions are captured in the list redflagage and redflaganswers Where they are conditioned on.

Producing relevant sequence of question:

Initially, when prompted with open text, the chatbot will process it with ProcessConditionStart and subsequently, for all the remaining questions, ProcessCondition1 will loop back to itself until all the relevant questions are answered.

NewConditionQuestions

These are where the processing functions for relevant question generation is stored. Each question is numbered with an index and when running through the function, user_data[SequenceCount] is updated and question will be generated at the same time. Depending on SequenceCount number, the relevant question will be output. Note that because the updated SequenceCount and questions are output by the same function, when conditioning on question with index 1 in NewConditionQuestion file, the index for the same question in telegram_chatbot is index 2. Note that if the last index is 6, then the one alluded in the telegram_chatbot file is 0.

Training of Chatbot

For details on the training of chatbot visit <https://nlu.rasa.com/python.html>

Recording of conversation

The recoding of conversation is under the NewConditionQuestion code. For each type of ending in the conversation to be recorded there is a different code. However, the differences are mainly in the ending sentences to be output to user. The main workings of the code are just conditioning on if the key exist in user_data and if it does, to record it into the pdf file.