COSC 310 - Team 15

Explanation of Features Implemented in Assignment 3

For our simple GUI we decided to integrate our agent with the Telegram API GUI. We decided to use the Telegram API as it allows anyone in the world to have access to it so long as they have a device connected to the internet. All the user needs to do is install Telegram on their device and in the search bar type @COSC310_Physiobot and they can immediately start chatting with our agent. The GUI shows the texts, and shows read texts. The GUI also shows our agent's name. The GUI includes a keyboard which is used by the user to communicate with the agent. The GUI allows the user to do speech to text translation so the user can input their phrases using their voice rather than microphone. A timestamp of when the message was sent is shown by the GUI.

The second topic that has been added is an exercise suggestion function, which helps the user to work out specific parts of their body. This improves our agent's conversation for our overall system because now our agent is able to respond to a user in another context but still related to our original topic.

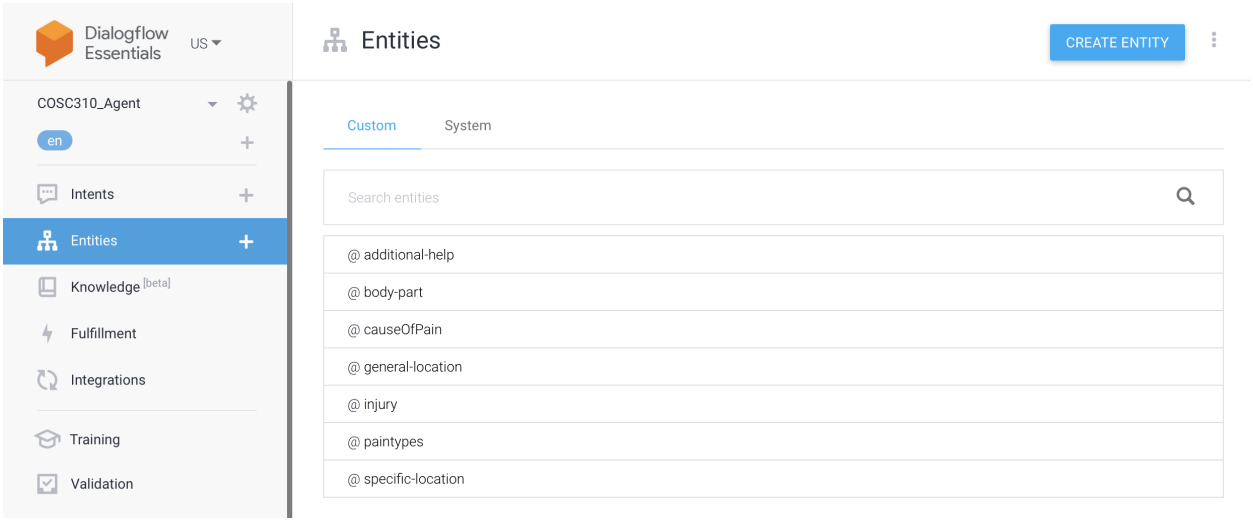


Figure 1: Defined Entities

Through Google's DialogFlow API we have our defined entities (Figure 1), within these entities we have selected the synonyms feature. This feature allows us to enter synonyms to train our agent that would be recognized by our agent and improve the overall flow of conversation. With this feature our agent is now better able to understand the user's input when phrased differently. Figure 2 shows a snippet of what is exactly activated by selecting this feature and some of the synonyms that we have included.

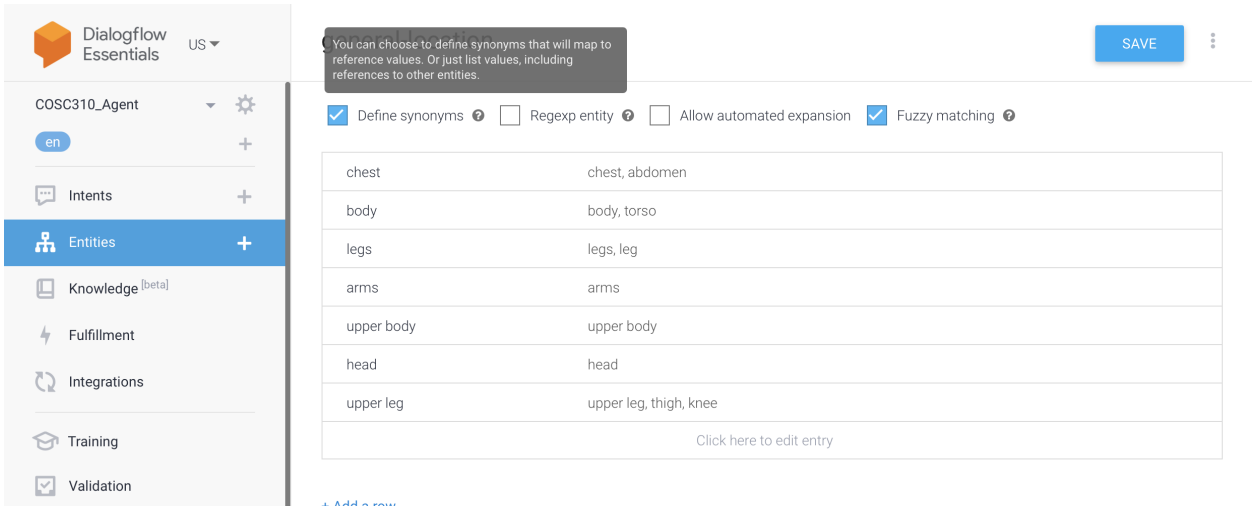


Figure 2: Feature - Synonym recognition

The entities are primarily used in the QueryHandler which acts as a diverter for the flow of conversation. As shown in Figure 3 there are specific entities in the training phrases for our agent which allow the agent to recognize which entities are required to be matched. When the agent receives user input that does not have all the required entities in place it will store the entities already collected and will prompt the user to enter the remaining entities that are required for the conversation to proceed.

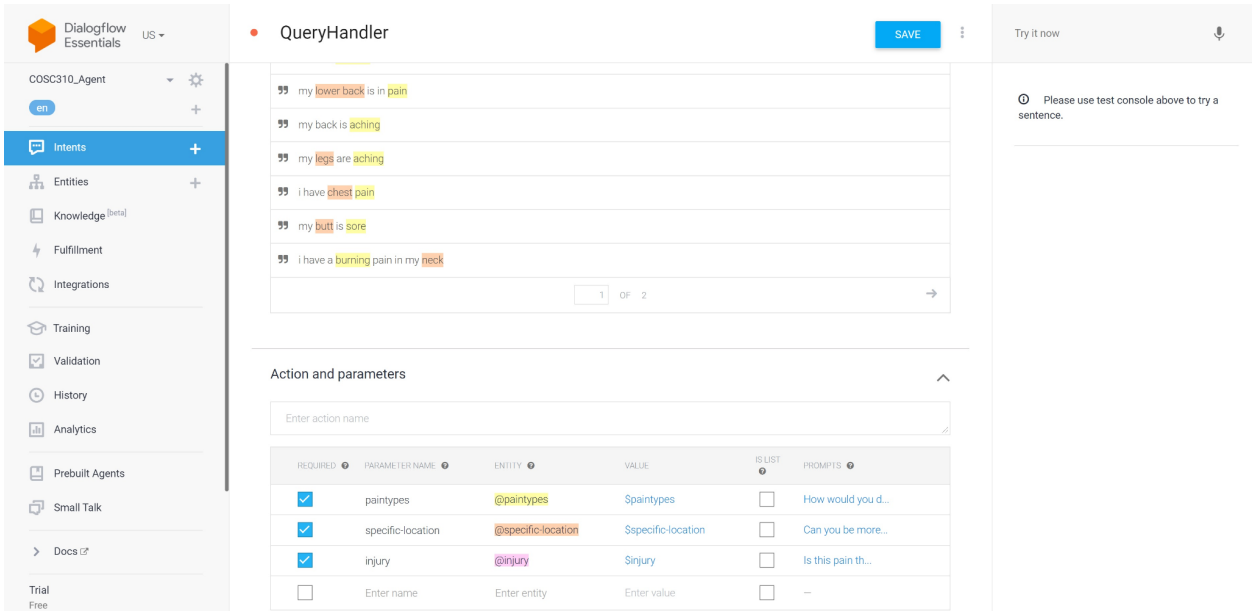


Figure 3: Feature - Entity Matching

Another feature that our team has selected is fuzzy matching. By adding this feature our agent is enabled to handle spelling mistakes of words the agent is trained to recognize. Through the use of this feature the agent does not get stuck or misunderstand the user therefore, we are able to

improve the overall flow of conversation. Figure 4 shows a snippet of what is exactly enabled by selecting this feature

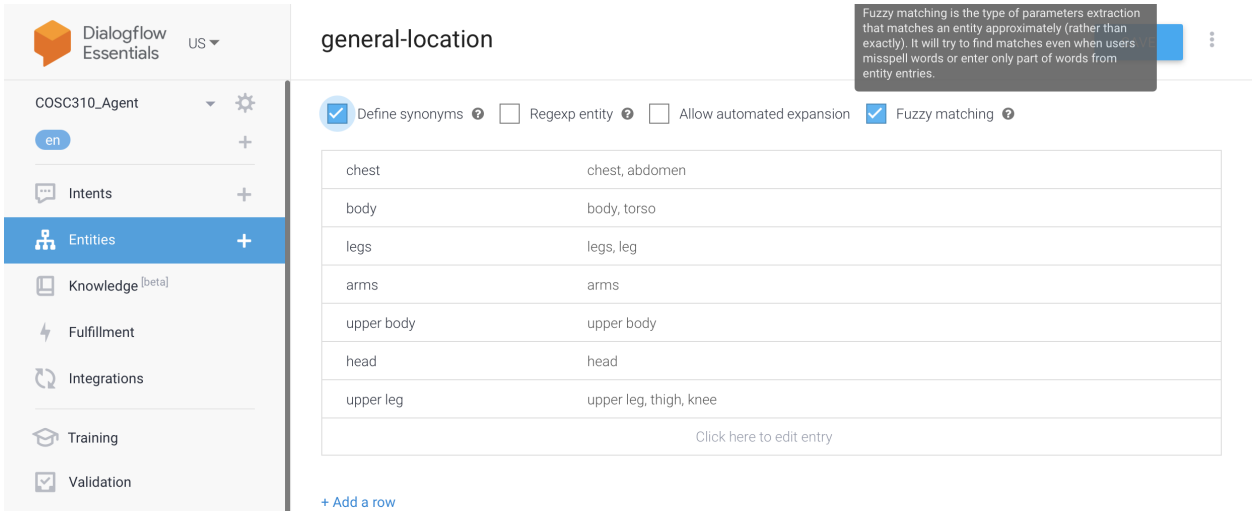


Figure 4: Feature - Spelling Mistakes (Fuzzy Matching)

In the instance that the user enters something outside of the two topics in which the agent is trained to respond to the agent will not be able to recognize a pain type or specific location. Rather than going to the QueryHandler in this case would trigger the default fallback intent.

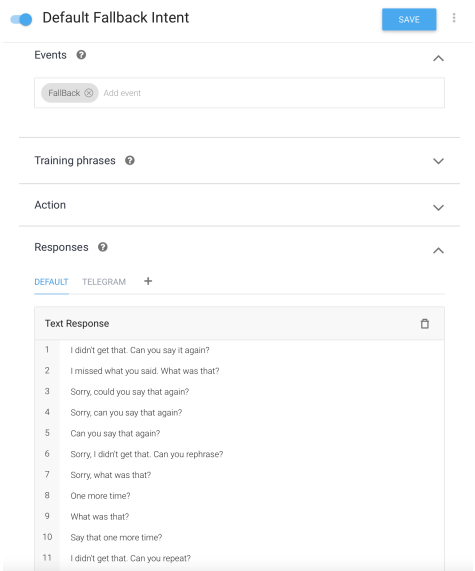


Figure 5: Feature - Reasonable Responses

The default fallback intent is a major intent with training phrases to help the agent ask the user for more information to better understand how to respond. The agent would respond with a default response which is likely to be asking the user to rephrase or restate what they are trying to say. The current intent and parameters are stored for one turn in most cases which allows the bot to resume conversation where it left off.

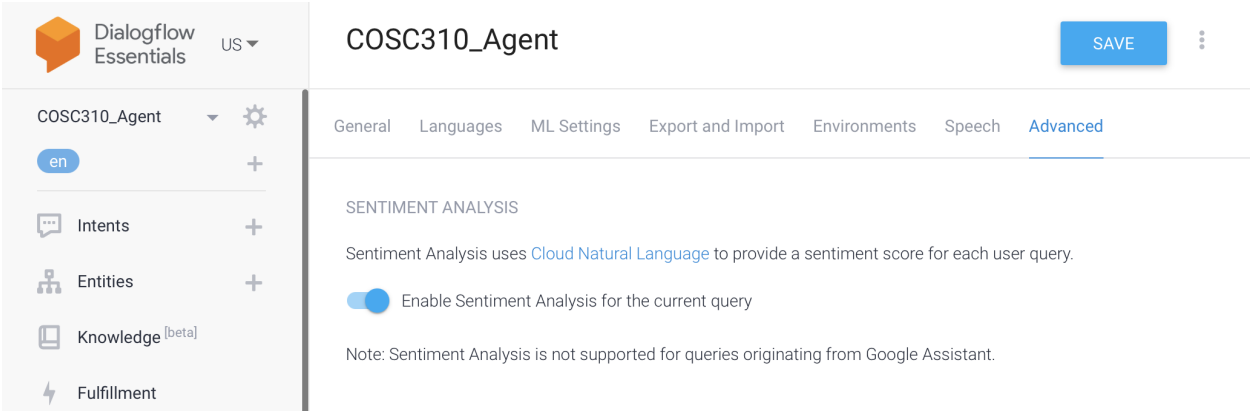
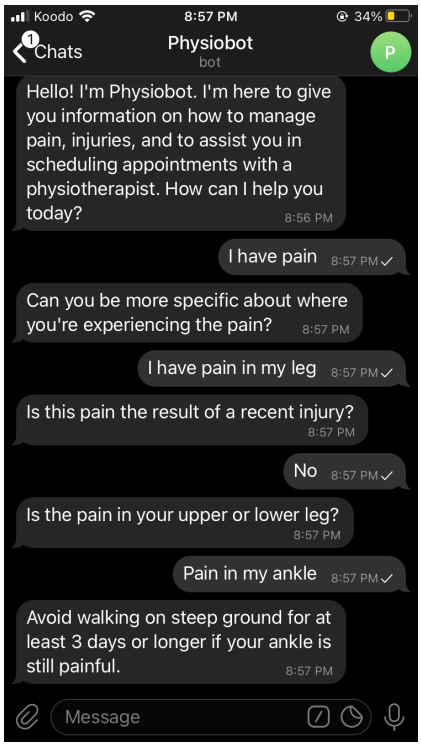


Figure 6: Feature - Sentiment Analysis

Within Google's DialogFlow API we are able to enable the sentiment analysis feature. As presented in Figure 5, the sentiment analysis feature uses cloud natural language to provide a sentiment score for each input received from the user. Our sentiment analysis feature allows the bot to analyze the sentiment of the user text input in real time and allows the bot to respond accordingly.

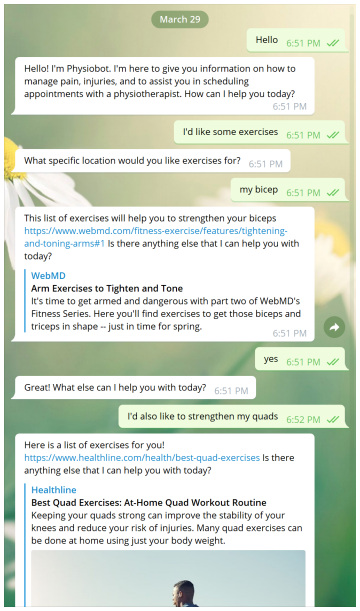
Snippets of Conversation Demonstrating Each Feature

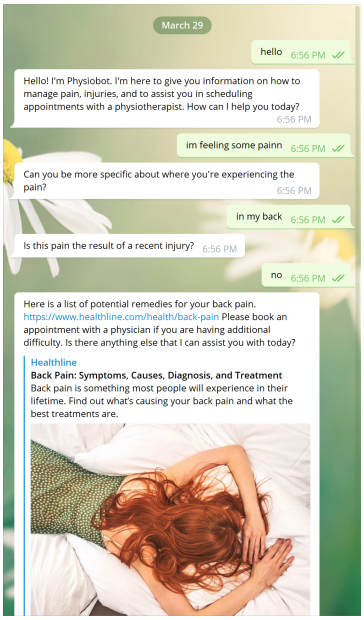
- **Simple GUI**



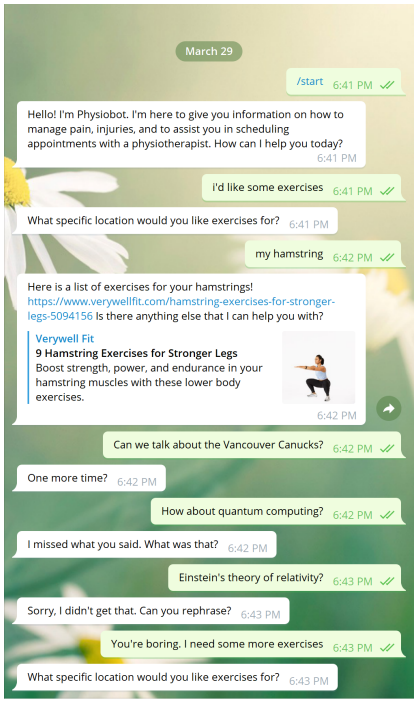
- **An extra topic.**

  These two conversation snippets are showing the two different topics that are handled by the bot. The first topic has to do with diagnosing injuries and pain types and is the bot's main function. The conversation on the left is the exercise function and the conversation on the right is the pain diagnosis function.



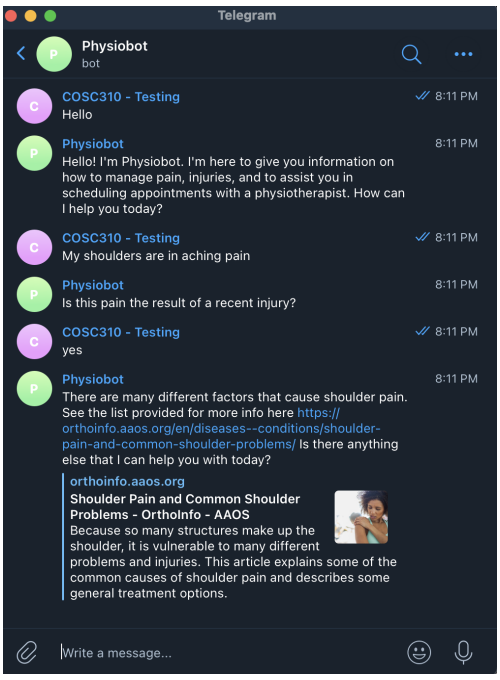


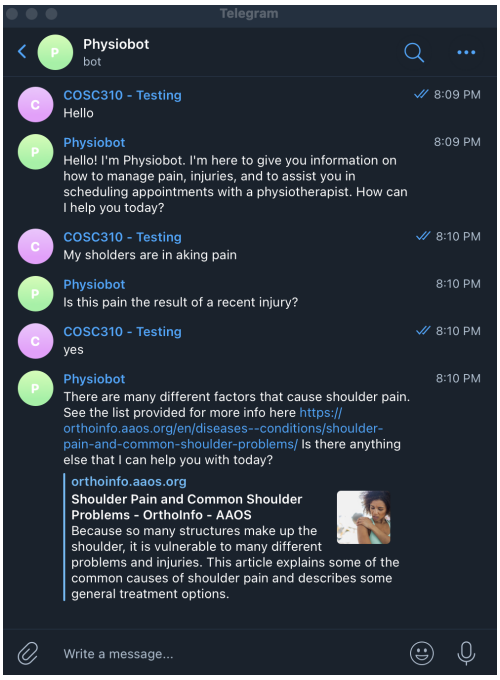
- **5 reasonable responses (when the user enters something outside the two topics).**



A simple demonstration of some of the reasonable responses that the bot will give when asked questions outside of its scope. The bot will only handle a case where it knows which entity to match a particular query to. In this instance it provides reasonable responses until the correct intent (exerciseHandler) is matched from the user's last query.

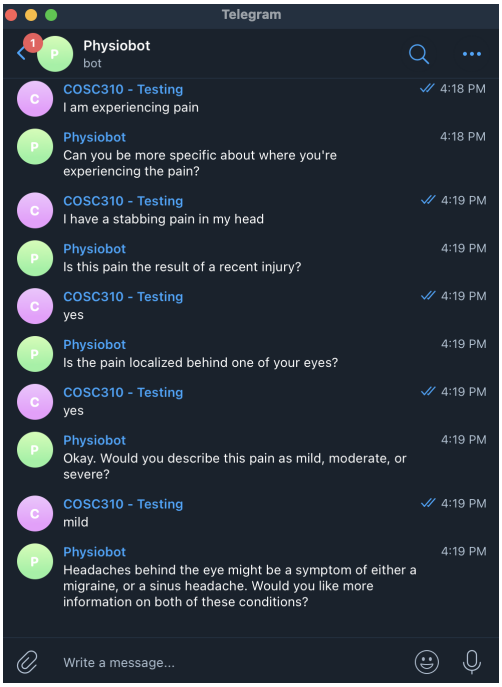
- **Spelling mistakes (Fuzzy matching).**



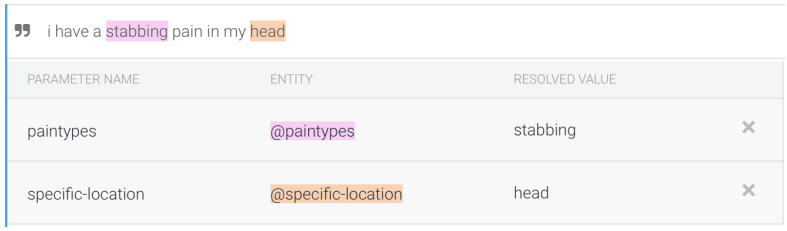


The snippet of the conversation on the left presents a regular conversation of the user prompting the agent with a pain type and a specific location of the pain. In the snippet on the right there is the same thing but with spelling mistakes within the pain type and a specific location of the pain. Since our agent is enabled with the feature to handle spelling mistakes (this is not hardcoded) the agent's response is with approximate entity matching rather than exact. In a general enquiry from the user with correct spelling there is exact entity matching.

- **Entity Matching.**



From this snippet of conversation we see that the agent gathers specific information about pain type and specific location of the pain in order to match the associated entities. This is more clearly presented in the screenshot below that is taken from our training completed within our QueryHandler intent. The resolved value is what is matched to an entity.



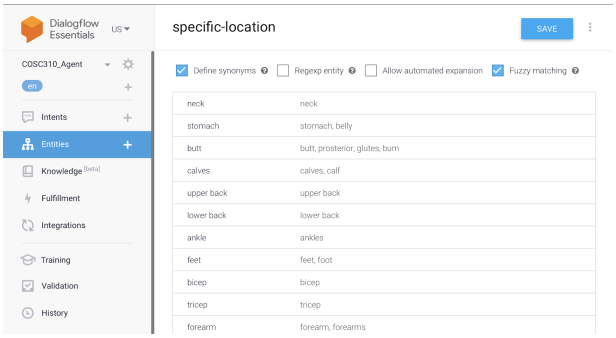


Figure: @specific-location entity

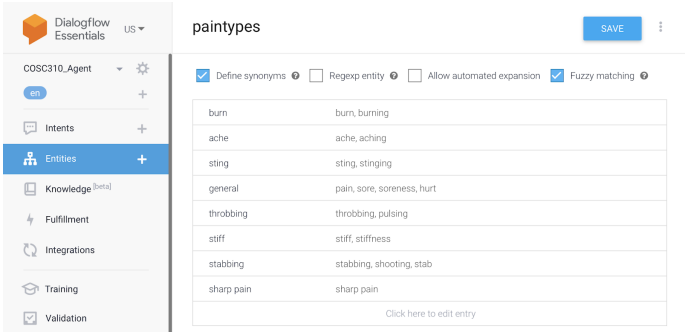
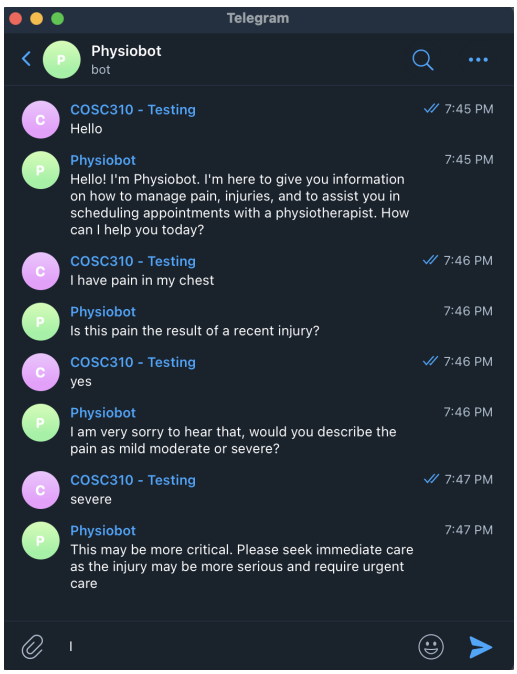
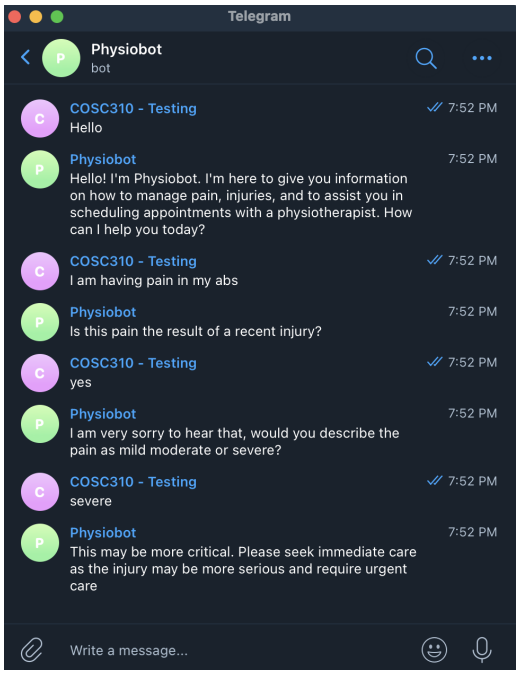


Figure: @pain-types entity

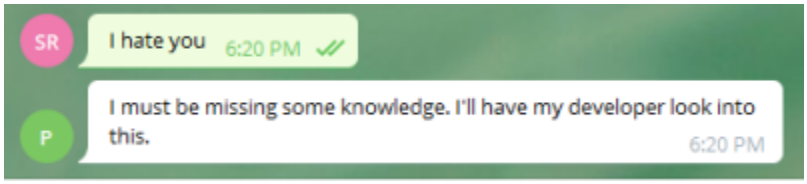
- **Synonym Recognition.**





The agent recognizes “chest” and “abs” as synonyms (implemented in training) therefore presents the same response.

- **Sentiment Analysis.**



In this example the user is frustrated and tells the bot that they hate it. The agent reads the negative sentiment of the user and gives a response in an attempt to calm the user down. The bot lets the user know it is missing information and the user can feel better as they the issue will be looked into.