# Results

The total messages sent through the application were 429 out of which the users chose to opt for the suggestions for only 82 times. Figure 1 describes the times each of the two chatbots were chosen to be clicked. It can be noted that Blenderbot was clicked on 49 times and DialoGPT was clicked on 33 times.

Chart, bar chart

Description automatically generated

One of the reasons for the AdvisorBot to not be clicked frequently was caused due to the lack of the computational resources required for generating the suggestive feed. As described by the users that the suggestions took time to load. The average time taken for a message to be received and the suggestions to be provided by the users was 19.59 seconds with a maximum time taken of 69.62 seconds. Figure 2 describes the box plot and the histogram of the delay in time for the message to be received and the suggestions to be computed and provided to the user.

A picture containing logo

Description automatically generated

The users were asked to vote the conversation status to provide insights to how the conversation is flowing. Since the user can vote any time to provide insights on how the conversation is flowing. There were a total of 11 cases where users felt that the conversation quality was good and three cases where the users felt that the conversation quality was poor. When the user votes that the conversation is flowing well, it can be determined by the number of messages the user has sent and if the user has clicked on any suggestions that was provided by the AdvisorBot. From this, it was recorded that there was a total of 97 instances of messages sent when the users have selected the suggestion provided to them and voted that the conversation was going well, and 0 instances of messages when the users have not selected that the suggestions provided to them and voted that the conversation was not going well. This can determine that the conversation quality can be improved when the user selects the advice from the AdvisorBot. This is visualized in Figure 3

Shape, square

Description automatically generated

Similarly, the users were asked to rate the suggestions from the AdvisorBot. There were 17 instances where the users have voted the suggestions to as good and 6 instances where the users have voted the suggestions as poor. This can help to determine the amount of messages that the user has sent prior to the vote while including the number of times the user has clicked for the suggestion provided by the AdvisorBot. As shown in Figure 4, there were a total of 137 instances of messages sent prior to when the user votes that the AdvisorBot suggestions provided were good and 36 instances of messages sent prior to when the user votes that the AdvisorBot suggestions provided were poor.

Chart

Description automatically generated

Applying the Bert Base Model for the sentence embeddings, the cosine similarity can be determined between the suggestion that was provided to the user by the AdvisorBot and the message that the user actually sent. There was total 86 instances where the message sent by the user from the suggestion provided by the AdvisorBot had atleast cosine similarity of 0.8 and 130 instances when the message sent by the user and the suggestion provided by the AdvisorBot had a cosine similarity of less than 0.8. However, it can be noted that there was no instance of cosine similarity being negative which can help to understand that the suggestions provided to the user and the messages sent were not strongly opposite vectors.

The popular input message or the message received to the user for the user to reply and get a suggestion was common greeting messages such as hi, hello, or when sender has expressed their likes and interest to the user such as “I like anime”, or when the sender questions the user about their interests.

The popular input messages when the message sent by the user was completely different from the suggestion was when the sender talks about specific items such as city weather, or in details about a specific interest. Another important feature here is that sometimes the user did not choose the suggestion from the AdvisorBot for common greeting messages such as hi or hello. This can be due to the delay in the suggestions being received by the user and the user to send their reply. Or due to the fact that the user wants to switch to a different topic.

The data from the cosine similarity can help to determine when the user wants to switch to a different topic and the AdvisorBot can be trained to identify when the user wants to switch to a different topic and provide suggestions about a different topic. This can be simply achieved by getting the message from the sender and the message that was sent by the user and train the AdvisorBot through a transfer learning approach.

Some other notable observations from the data are word cloud representations of messages when the users have voted that the AdvisorBot was performing good or bad as shown in Figure 5.

Text

Description automatically generated

AdvisorBot vote Good

Text

Description automatically generated

AdvisorBot vote Bad