

I Chatbot Query Generation

Algorithm 1 shows a slightly simplified version of how the chatbot chooses its queries to short user responses. The user response is stored together with a \textit{checkpoint list}, indicating the progress of the chat. Each time the user gives an argument or a counterargument, an integer is added to the checkpoint list: positive if the argument is long enough or a negative if the the argument is too short. The number is odd if it is an argument and even if it is a counterargument. If the user gives a response that is over 12 words in length the chatbot asks either for an argument or a counterargument, depending on the number in the checkpoint list. If the user response is below 12 words in length, the chatbot queries the user to expand on her answer.

- One-word answer: Chatbot asks how \textit{this} hinders the user doing more exercise.
- Answer contained words 'time' or 'busy': Chatbot asks why the user has no time.
- Answer contained negation: Chatbot asks why not.
- Answer contained transitional word: Chatbot "understands" but asks to give more details.
- Answer below 5 words: Chatbot asks why.
- Answer above 5 words: Chatbot asks to elaborate on given answer.

After one query, the according checkpoint is added to the checkpoint list, indicating that the chatbot already queried once. If the chatbot queried once it will not query again and move on to the next question, asking for an argument or a counterargument.

Algorithm 1 Chatbot question generation

```

1: procedure GENERATEQUESTION(user_response, stage)
2:   if length(user_response) < 13 and stage[-1] ≥ 0 then generateQuery(user_response)
3:   else analyse_stage(stage) ▷ returns appropriate chatbot answer according to stage in chat
     return chatbot_answer
4:
5: procedure GENERATEQUERY(user_response)
6:   time_words = ['time', 'busy']
7:   negations = neg = ['not', 'dont', 'arent', 'cant', 'wouldnt', 'isnt', 'don't', 'aren't', 'can't',
     'wouldn't', 'isn't']
8:   explanations = ['that is', 'because', 'due to', 'since', 'that's', 'that's', 'hence', 'therefore']
9:
10:  ▷ The questions are examples. They were randomly chosen from a list with several similar
     possibilities
11:  if length(user_response) is 1 then
12:    chatbot_answer = "How does this hinder you doing exercise?"
13:  else if user_response contains word in time_words then
14:    chatbot_answer = "I understand you have no time, but can you expand on that please. Why
     not?"
15:  else if user_response contains word in negations then
16:    chatbot_answer = "Why not?"
17:  else if user_response contains word in explanations then
18:    chatbot_answer = "Could you go into more detail? Why do you say that?"
19:  else if length(user_response) ≤ 5 then
20:    chatbot_answer = "Why?"
21:  else if length(user_response) > 5 then
22:    chatbot_answer = "Elaborate please...how does this hinder you from doing more exercise?"
     return chatbot_answer

```
