*Computational Models of Argument*
*S. Modgil et al. (Eds.)*
*© 2018 The authors and IOS Press.*
*This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).*
*doi:10.3233/978-1-61499-906-5-457*

# Chatbot Design for Argument Harvesting

Lisa A. CHALAGUINE [a] Anthony HUNTER [a]
[a] *Department of Computer Science, University College London, London, UK*

**Abstract.** In this paper we present design concepts for a chatbot than can be used for argument acquisition. We call the acquisition of arguments by means of a chatbot *argument harvesting*. The chatbot asks the user for his or her arguments on a topic of interest. It can also be used to harvest counterargument, values, preferences, as well as information about the user like his/her personal circumstances. The harvested arguments and other attributes have various applications from the instantiation of argument graphs to the development of computational persuasion systems.

**Keywords.** argument harvesting, argument acquisition, chatbots

## 1. Introduction

A common approach to argument acquisition assumes a static resource available on the internet where the topic of interest is/was already discussed. This raises several problems: firstly, what if no discussion platform for a particular topic exists? Secondly, even if it exists, assuming the topic is extensively discussed, it provides us with (all) arguments that exist on a certain topic but does not provide any information on the individuals who posited the arguments. This is a drawback of other systems such as D-BAS [1] that are more suitable for public argumentation or collective decision making. We, on the other hand, focus on behaviour change which requires a more individual approach. Sending out questionnaires or interviewing people directly, however, may be a labour-intensive and expensive undertaking. We have automated the process of acquiring arguments with a chatbot, terming this way of argument acquisition *argument harvesting*.

## 2. Chatbot Design

We present a design concept for argument harvesting with no or little prior domain-knowledge. We assume no programming knowledge regarding the infrastructure of the chatbot. We use an API (e.g. Facebook's *Send API*) that handles the sending and receiving of messages and the ability to chat with many users simultaneously.

The chatbot code consists of two parts. The server code that communicates with the API and the text-processing code that analyses the incoming messages and generates the chatbot responses. The chatbot asks the user to provide arguments on a topic of interest. Depending on the topic, there will be certain domain-specific user responses that have to be taken into consideration. This behaviour, if not known in advance, will be identified after a short trial period during which the chatbot code only accommodates the general user behaviour, outlined below. Users generally give

- a very short, not very informative response (statement)
- a single argument without going into specific detail (general argument)
- a single argument with specific detail (specific argument)
- several arguments at once (1+ arguments)

Most often, people give one-worded or very short responses. The chatbot replies with the following queries, if the reply is less than 12 words in length[1]:

- *One-word answer:* Chatbot repeats the word and asks the user why he or she gave *this* as a reason/argument.
- *Answer contains negation:* Chatbot asks why not.
- *Answer contains transition word*[2]*:* Chatbot says it *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.

The chatbot can either query once or query until a word threshold is met. If the user-reply is very long (e.g. more than 30 words) the chatbot can ask the user to break down her answer into several arguments, but in our experience this almost never happened.

Depending on what other attributes are of interest, the chatbot can also ask for those. For example: ask users to assign values to the arguments; ask the user to rank her arguments (preference relationships); ask to provide counterargument for each of her arguments; or ask domain or user specific questions in order to obtain a user profile. Those attributes could then potentially be used to instantiate argument graphs or generate predictive models of the user.

## 3. Example

In our case study [2], we asked women who do not (regularly) engage in physical exercise to provide arguments for their behaviour. After a short trial period, we observed that participants often replied with *"I am busy"* and *"I have no time"*. We adjusted the chatbot by asking the participant why she had no time or what she was busy with, if her reply contained the words *time* or *busy* and was under 12 words long. We also asked the participants to provide a counterargument to each argument they gave. We generated a database of arguments with their corresponding counterarguments, see Appendix A [3]. We intend to use the harvested arguments to develop a chatbot that can give counterarguments. We want to create and instantiate argument graphs with the resulting dialogues in order to analyse argumentation dynamics in the field of behaviour change.

## References

[1] T. Krauthoff, M. Baurmann, G. Betz, and M. Mauve. Dialog-based online argumentation. *In Proceedings of the 2016 Conference on Computational Models of Argument (COMMA'16)*, pages 33–40, 2016.

[2] L. A. Chalaguine, F. L. Hamilton, A. Hunter, and H. W. W. Potts. Argument harvesting using chatbots. *In Proceedings of the 2018 Conference on Computational Models of Argument (COMMA'18)*, 2018.

[3] Appendices. https://tinyurl.com/y8fjsab8.

[1]Length can be adjusted as seen fit.

[2]Transitions are phrases or words used to connect one idea to the next, e.g. *because, since, hence* etc.