A Chatterbot Based on Genetic Algorithm: Preliminary Results

Cristian Orellana[1], Martín Tobar[1], Jeremy Yazán[1], D. Peluffo-Ordóñez[2,3,4], and Lorena Guachi-Guachi[1,3](✉)

[1] Department of Mechatronics, Universidad Internacional del Ecuador, Av. Simon Bolivar, Quito 170411, Ecuador
loguachigu@uide.edu.ec

[2] Mohammed VI Polytechnic University, Ben Guerir, Morocco

[3] SDAS Research Group, Ibarra, Ecuador

[4] Faculty of Engineering, Department of Electronics, Corporación Universitaria Autónoma de Nariño, Pasto 520001, Colombia
http://www.sdas-group.com

Abstract. Chatterbots are programs that simulate an intelligent conversation with people. They are commonly used in customer service, product suggestions, e-commerce, travel and vacations, queries, and complaints. Although some works have presented valuable studies by using several technologies including evolutionary computing, artificial intelligence, machine learning, and natural language processing, creating chatterbots with a low rate of grammatical errors and good user satisfaction is still a challenging task. Therefore, this work introduces a preliminary study for the development of a GA-based chatterbot that generates intelligent dialogues with a low rate of grammatical errors and a strong sense of responsiveness, so boosting the personals satisfaction of individuals who interact with it. Preliminary results show that the proposed GA-based chatterbot yields 69% of "Good" responses for typical conversations regarding orders and receipts in a cafeteria.

Keywords: Chatbot · Evolutionary programming · Natural language processing

1 Introduction

Chatterbots are computer programs that simulate intelligent human conversations. The usual chatterbot execution involves a user providing natural language input, to which the chatterbot responds with a reasonable and presumably intelligent response to the input sentence. Chatterbots are commonly used in customer service, product suggestions, e-commerce, travel and vacations, queries and complaints [1,2]. Since the 1990s, several technologies have been developed for the development of chatterbot-based solutions, including evolutionary computing, artificial intelligence, machine learning, natural language processing [2].

H. Florez and M. F. Pollo-Cattaneo (Eds.): ICAI 2021, CCIS 1455, pp. 3–12, 2021.
https://doi.org/10.1007/978-3-030-89654-6_1

In recent research, Genetic Algorithms (GA), a technique from evolutionary computing, have been used to create chatterbots that retrieve dialog responses to user input [3–6]. GA-based chatterbots reflects the process of natural selection and attempts to improve the fitness function by fitting patterns and rules to human inputs. This technique has been used primarily for trivial conversations, and it, like other techniques, such as deep learning, is prone to producing meaningless or inappropriate responses [7].

In this work, we present a GA-based chatterbot aiming at providing satisfactory and grammatically correct responses. For purposes of this work, we create a dataset of 120 phrases appropriate for conversations about food orders or recipes in a cafeteria. To reduce the risk of grammatical errors and to maintain structure, the population is created with randomly selected words from the database (adverbs, nouns, verbs, etc.). The mutation operator determines the population's diversity in each generation replacing a random word from a phrase with a new word of the same type (adverbs, nouns, verbs, etc.) from our dataset.

This paper is structured as follows: Sect. 2 presents a brief review of state-of-the-art related works. The database is described in Sect. 3. The use of GA for building a chatterbot is described in Sect. 4. Section 5 describes the experiments carried out over the database suitable for ordering food or recipes in a cafeteria. Section 6 presents the preliminary results and discussion across the experiments. Finally, Sect. 7 gathers the concluding remarks.

2 Related Works

On one hand, GAs have proven to be effective in the development of chatterbots and the identification of appropriate responses. For instance, [8], proposes a chatterbot model built on a natural language-adapted algorithm that combines indexing and pattern matching to generate new phrases from preloaded words. A GA-based chatterbot to learn through samples of conversations between the user and the system is proposed in [9]. It generates the response evaluating the type of expressions used by the user based on rules built from the last samples of conversation. In [10], an inductive chatterbot model based on a GA with sexual selection (SeGa-ILSD) is developed in order to handle different languages: English and Japanese.

On the other hand, some hybrid techniques based on GA have been also introduced. For instance, the work carried out in [11] addresses a user-adaptive communication robot using interactive evolutionary computation (IEC) and machine-learning approaches for robot's movements. In particular, IEC genes are initialized at random. Then, one of them is translated into the parameters of the robot's value system, and the robot interacts with the user. In this case, fitness is provided by users at the end of each interaction. This sequence continues until all genes are evaluated. Finally, more appropriate genes are generated.

The study in [12] determines that others relate better to people who have similar personality characteristics to them. In this sense, this research introduces a GA capable of changing their responses behavior in real-time. The change in GA behavior depends on the fitness function, which is calculated based on how comfortable/satisfactory the response has been from the user's perception. The comfortable scale ranges from 1 to 10. The best responses (comfortable/satisfactory) are passed on to the next generation, while the uncomfortable ones are combined with a different random generation.

Furthermore, more specialized research in [3] address the creation of a chatterbot with AIML, which is similar to an XML database that employs specific tags for each instance. The new chatterbot is being used for an interactive university chatterbot that will answer related FAQs. Users submit their questions, which are subsequently processed to match the specified answer and returned. In [4] a worth-method for creating new sentences in Spanish by combining prior sentences is presented. Each phrase is treated as an individual, with the genes representing the words that make up the phrase. The mutation coefficient is inversely proportional to the number of words, and there is no crossover. Mutation, which is carried out depending on the feature of the word, provides population diversity. The amount of results found for a specific phrase on the WWW is used to assess the fitness function.

Although there are valuable research on GA-based chatterbots in the literature, the usage of GA with a low rate of grammatical errors and good user satisfaction is a challenging task yet. Therefore, we introduce a chatterbot based on GA that generates intelligent dialogues with a low rate of grammatical errors and a high sense of responsiveness, increasing the personal pleasure of those who interact with it.

3 Database

The database consists of 120 statements suitable for ordering food or recipes in a cafeteria. All sentences are written in a .JSON file and more sentences can be added depending on the target application. Samples of the entries in the database are as follows:

- **"Greeting"**
  - *Patterns:* "Hi", "How are you", "Is anyone there?", "Hello", "Good day", "Whats up"
  - *Responses:* "I can help you", "You have a good day", "I am happy to see you"
- **"Goodbye"**
  - *Patterns:* "See you later", "Goodbye", "I am Leaving", "Have a Good day", "Bye"
  - *Responses:* "I am sad", "I see you later", "It was a pleasure"

- **“Age”**
  - *Patterns:* “how old”, “what is your age”, “how old are you”, “age?”
  - *Responses:* “I am 18”, “I have no idea”, “It does not matter”, “It is a secret”
- **“Name”**
  - *Patterns:* “what is your name”, “what should I call you”, “whats your name?”
  - *Responses:* “My name is Tim”, “I am Tim”, “It is Tim”
- **”Shop”**
  - *Patterns:* “Id like to buy something”, “whats on the menu”, “what do you recommend?”, “could i get something to eat”
  - *Responses:* “It costs 20”, ”We sell cookies”
- **“Hours”**
  - *Patterns:* “when are you guys open”, “what are your hours”, “hours of operation”
  - *Responses:* “We are open at 8”, “We are closed at 20”

4 The Use of Genetic Algorithm

This work uses a GA to generate a suitable sentence based on the combination of a set of words. The general workflow of the proposed approach is schematized in Fig. 1. The population of s sentences is made up of w words randomly selected from dataset of words to reduce the risk of grammatical errors. Every sentence is mutated by substituting a random word with a new one from word dataset. The “fitness” of each sentence is determined by its frequency on the World Wide Web (WWW).

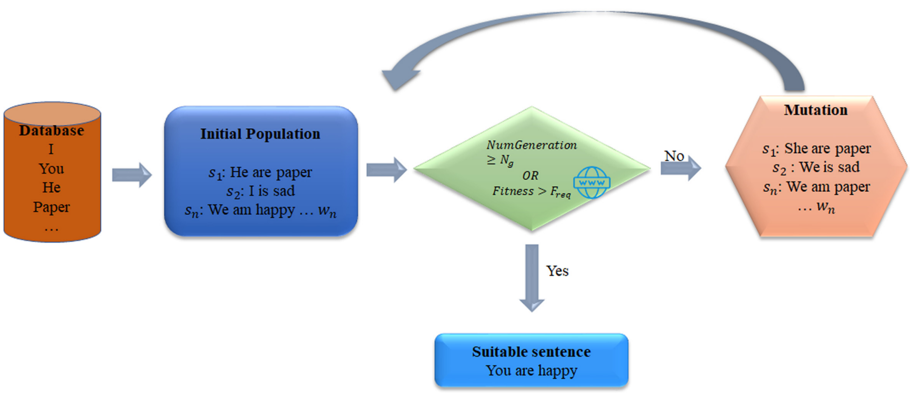


Fig. 1. Workflow of genetic algorithm to generate the most appropriate sentence.

4.1 Initial Population Generation

This step selects random phrases from database pool which contains sentences that frequently used in human conversations. Like "How was your day?", "How are you?", "I'm doing great and you"?, etc. Then, initial population of sentences is created extracting words from selected phrases and keeping the structure (adverbs, nouns, verbs, etc.).

4.2 Mutation

For purposes of this work, the mutation probability (P) is the inverse of the total amount of words in the phrase.

The mutation probability (P) is given by:

$$P(N) = \frac{1}{N} \tag{1}$$

where N is the number of words.

The algorithm identifies the word to be replaced (adverbs, nouns, verbs, etc.) and substitutes it with the same kind of word in order to keep the structure as good as possible.

4.3 Evaluation

The evaluation is carried out by searching the WWW and finding out how frequently the phrase is found. To accomplish this, each phrase is typed into the Google search engine, and the frequency is determined by the number of times the engine finds the same phrase. The greater the frequency of phrase, the better the individual's fitness. When a complete generated phrase can not be found literally on the searching engine, a phrase partition strategy is proposed, since the shorter the strings entered into the engine, the greater the number of results are obtained. Furthermore, x threshold values will be assigned to classify whether a generated sentence is good or bad.

4.4 Pseudo-code

Genes (individuals) are composed of three essential parts: subject, verb, and predicate. Each one is represented by a char string with a distinct meaning. The initial population is created with phrases chosen randomly. Then, the fitness is calculated by comparing each phrase to a corresponding sentence in the database. If the sentence is consistently logical, it is Google-searched to determine the search frequency. This implementation is extended in Algorithm 1.

Algorithm 1. Used Algorithm's Pseudo Code

```
Select random phrases from database                        ▷ Create Initial Population
Generations = 0
function EVALUATE                                          ▷ Evaluate new population
    Compare generated phrase similarity with data base phrases.
    Fitness = Amount of Google results [4]    ▷ Check the amount of times the
  phrase is found on google in a literal way.
end function
while Generations < 5 or Fitness < 2 000 000 do
    Mutate Population
    Probability = 1/AmountOfWords
    for N do                                               ▷ N is the amount of words
        R = Random Number between 0 and 1
        if Probability > R then
            Substitute word for a new one
        end if
        function EVALUATE                                  ▷ Call Evaluate function
        end function
    end for
    Generations = Generations + 1
end while
```

5 Experimental Setup

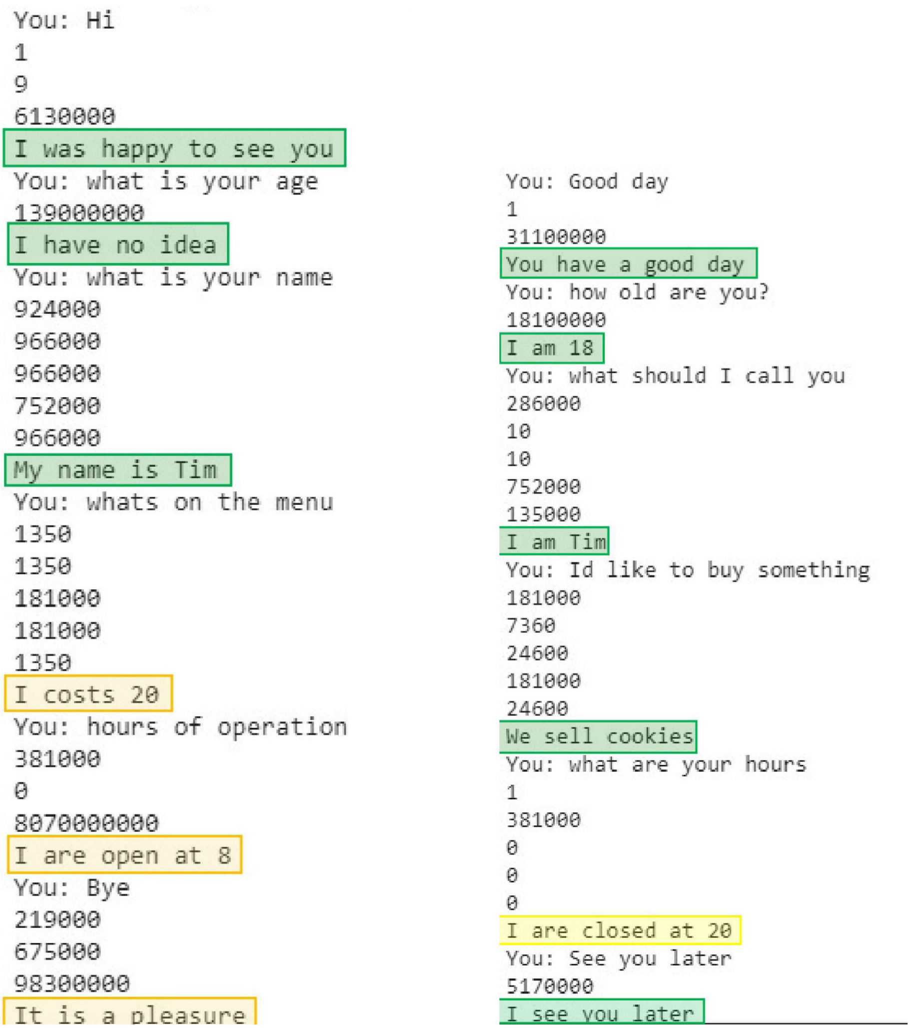
For experimental purposes, Python routines have been implemented using Google Colab[1]. This environment has 12 GB of RAM and 100 GB of storage space. To work with human language data, the Natural Language Toolkit (NLTK) was also used. Arrays were also manipulated using Numpy and Scikit-learn to obtain the linear regression function. Python reads json files using JSON. BS4 was used to connect to the Internet. Difflib was used to compare two char strings, and the Requests library was used to search a website for a specific query.

The quality of the responses generated by the proposed chatterbot (fitness function) are evaluated using the database presented in this work together with an Internet search algorithm; if these answers do not meet a 90 % similarity percentage with the dictionary along with a minimum number of 2000000 internet searches, the algorithm generates a new response until these conditions are satisfied, otherwise, the program ends at the end of four iterations. It is worth noting that the minimum search value was obtained as the most optimal from experimental tests. The program was run on a total of 20 times and subsequently, the resulting conversations are evaluated by the users according to the degree of satisfaction they had with the responses made by the chatterbot

[1] Source: https://colab.research.google.com/drive/1Y7Cd5xTVZ0TtGIXDGNmBBEQe7u_Lic3b?usp=sharing.

6 Experimental Results

Samples of answer obtained are depicted in Fig. 2 and Fig. 3.

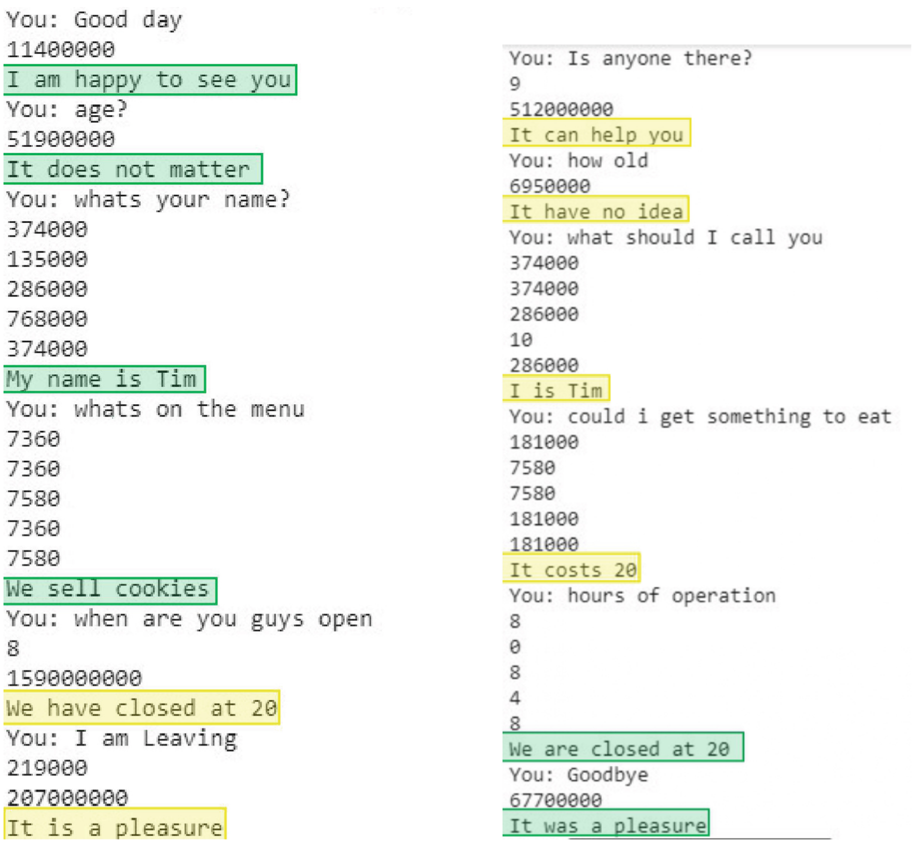


(a) First example of a conversation test user-Chatbot

(b) Second example of a conversation test user-Chatbot

Fig. 2. Conversation sample with answers obtained by using the proposed genetic algorithm. The numbers under the user's sentence correspond to the number of searches found on the internet of the answers generated by the Chatbot. (Color figure online)

The fifty-eight responses obtained by the Chatbot to experimental questions have been evaluated based on an application metric in which users assigned a



(a) Third example of a conversation test user-Chatbot

(b) Fourth example of a conversation test user-Chatbot

Fig. 3. Conversation sample with answers obtained by using the proposed genetic algorithm. The numbers under the user's sentence correspond to the number of searches found on the internet of the answers generated by the Chatbot. (Color figure online)

rating based on the level of personal satisfaction, which is structured in three categories:

- Good: Green square.
- Regular: Yellow square.
- Bad: Red square

The users' evaluation is based on the orthography and coherence in the structure of the ChatBot's answers, as well as their similarity to the words in the database. In this way, if the answer meets the aforementioned criteria, it will be rated "good". If it contains spelling errors that do not significantly alter the meaning of the answer, it will be rated "regular." Finally, if the response is grammatically incorrect, it will receive a "bad" grade. The number of answers grouped by category is illustrated in Figure 4.

From Fig. 4, it can be seen that the proposed chatterbot by using genetic algorithm generates 69% Good phrases, 31% regular, and 0% Bad phrases. Therefore, the proposed chatterbot meets to a great extent the need for a cafeteria application since the answers classified as regular did not present errors that mean a loss of direction in the user-chattbot conversation.

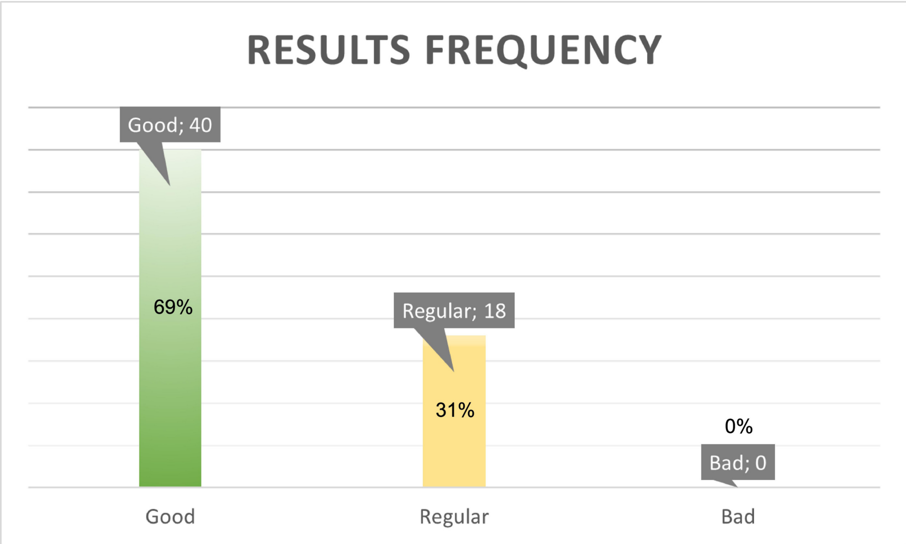


Fig. 4. Personal satisfaction

The processing time of the proposed approach has been also evaluated. Table 1 shows the average of the time consumed by the operations included in the genetic algorithm, besides, we can see the significant consumption of the RAM memory. Since the execution of the algorithm involves a reduced time, the proposed chatterbot can be an appropriated tool for providing an immediate support.

Table 1. Average of the time and RAM consumed for each execution of the algorithm.

Metric	Average value
Time (sec)	0.112
RAM used (MB)	840

7 Conclusions

This work presents a GA-based chatterbot that can be used to improve customer experience in situations related to ordering in a cafeteria. It generates responses with a low rate of grammatical errors and a strong sense of responsiveness, so boosting the personals satisfaction of individuals who interact with it. The algorithm is still in a preliminary stage. Genetic evolution is able to generate new responses and improve the diversity of the chatterbot while maintaining the overall style of ordering conversation. Preliminary results show that the proposed GA-based chatterbot produces 69% good responses for typical conversations regarding ordering and receipts in a cafeteria. On the downside, responses are frequently supplied after multiple iterations of the algorithm, which should be improved in order to allow for faster client communication.

References

1. Satu, M.S., Parvez, M.H., et al.: Review of integrated applications with aiml based chatbot. In: 2015 International Conference on Computer and Information Engineering (ICCIE). IEEE, pp. 87–90 (2015)
2. Ahirwar, Gajendra Kumar: Chatterbot: technologies, tools and applications. In: Nanda, Aparajita, Chaurasia, Nisha (eds.) High Performance Vision Intelligence. SCI, vol. 913, pp. 203–213. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-6844-2_14
3. Ranoliya, B.R., Raghuwanshi, N., Singh, S.: Chatbot for university related faqs. In: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, pp. 1525–1530 (2017)
4. Montero, C.S., Araki, K.: Genetic Algorithm-like Approach to Natural Language Phrase Generation-Case of Study Spanish. Technical Report. http://www.um.es/lacell/proyectos/dfe/
5. Pham, X.L., Pham, T., Nguyen, Q.M., Nguyen, T.H., Cao, T.T.H.: Chatbot as an intelligent personal assistant for mobile language learning. In: Proceedings of the 2018 2nd International Conference on Education and E-Learning, pp. 16–21 (2018)
6. Daniel, G., Cabot, J., Deruelle, L., Derras, M.: Xatkit: a multimodal low-code chatbot development framework. IEEE Access **8**, 15 332-15 346 (2020)
7. Hadjiev, A.M., Araki, K.: Chatbot system for open ended topics based on multiple response generation methods
8. Vrajitoru, D.: Evolutionary sentence building for chatterbots. In: Genetic and Evolutionary Computation Conference (GECCO). Citeseer (2003)
9. Kimura, Y., Araki, K., Momouchi, Y., Tochinai, K.: Spoken dialogue processing method using inductive learning with genetic algorithm. Syst. Comput. Japan **35**(12), 67–82 (2004)
10. Araki, K., Kuroda, M.: Generality of spoken dialogue system using sega-il for different languages. In: Computational Intelligence, pp. 72–77 (2006)
11. Suga, Y., Ikuma, Y., Ogata, T., Sugano, S.: Development of user-adaptive value system of learning function using interactive ec. IFAC ProC. Vol. **41**(2), 9156–9161 (2008). 17th IFAC World Congress. https://www.sciencedirect.com/science/article/pii/S1474667016404258
12. Yorita, A., Egerton, S., Oakman, J., Chan, C., Kubota, N.: Self-adapting chatbot personalities for better peer support. In: 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC). IEEE, pp. 4094–4100 (2019)