ISSN 1870-4069

Front-End Responsive Web Application for Chatbot Retraining Using ReactJS with MongoDB

Luis A. Fernández-Castro[1], Javier A. Valdivia-del-Ángel[1], Kevin G. Zazueta-Sánchez[2], Ricardo Cantú-Martínez[2], Yoel Ledo-Mezquita[1], Elioth Macias-Frotto[1]

[1] Tecnologico de Monterrey, School of Engineering and Sciences, Mexico

[2] NDS Cognitive Labs. Prolongación, Mexico

{a01023675, a01658296, yledo, a00837025}@tec.mx, {kzazueta, rcantu}@ndscognitivelabs.com

Abstract. This article shows the front-end implementation of a responsive web application for chatbot retraining, and the design justification for it.

Keywords: Front-end, responsive web application, chatbot retraining, reactJS, MongoDB.

1 Introduction

It was during the 1960s that computer science professor Joseph Wiezenbaum, at the artificial intelligence laboratory of the Massachusetts Institute of Technology, created the first chatterbot, later shortened to a chatbot, called "Eliza" [1]. The program sought to emulate a psychotherapist by recognizing patterns and interacting with the user through the use of a pre-set script.

Currently, chatbots are software or computer programs that simulate human conversation or "chatter" through text messages or voice interactions [2]. In addition, these programs are much more robust than their predecessors, and increasingly resemble communicating with another human being.

The importance of chatbots has been increasing because they increase efficiency and productivity, since these automated programs are delegated with tasks mainly of sales, customer service or virtual assistance. But above all, while consumers are moving away from traditional forms of communication, more and more chat-based communication methods are expected [3].

As chatbots have become more complex in their comprehension and communication capabilities, the way these programs are created, trained, and retrained has also evolved in order to improve their functionalities in accordance with their repeated interaction with users. In this sense, chatbots have gone from using predefined scripts to using machine learning algorithms to adapt as they interact more with the user.

One of the most important phases in the life cycle of a chatbot is its retraining, since it has been able to go through previous stages of design and construction where its programmer or programmers have made a judgment of its optimal behavior, but by using real user data, it is possible to evaluate whether these prejudices have been correct or not and the chatbot can be adapted to the true needs of the chatbot. user.

The intention of this article, beyond restricting itself to treating chatbots and their algorithms as its main thesis, seeks to recognize the chatbot retraining phase as a central point of study, more specifically the semi-automation of this from another computer program, which serves as an interface.

This framework includes the delimitation of the retraining phase itself and the technologies used to carry out this program. Therefore, we will delve into the technologies used and their justification for this, and we will underline the importance of these with examples of use of the product that is being developed in conjunction with NDS Cognitive Labs for this same task.

2 Chatbot Retraining

To highlight the issue of chatbot retraining, it is necessary to think about the reason for the success of these computer programs. This software depends on their adaptability and problem solving, and this is why it is possible that when they cannot adapt to the needs of the user, they must at least have the ability to retrain [4].

2.1 Manual Chatbot Retraining

The manual retraining of chatbots is simply known as conventional training, and it is very similar to the training that is done before the publication of this system, with the difference that now there is additional information, hundreds of data about their interaction with users, which helps the programmer or programmers to review the weak points of the program.

Depending on the complexity of the computer program being discussed, whether it is a chatbot based on keyword recognition, one based on menus of options, or a hybrid that combines the two mentioned above, is how the training is going to be done, either including more options that allude to the needs of the consumer or expanding the list of keywords and the answers needed for them.

2.2 Semi-Automatic Chatbot Retraining

It is called semi-automatic retraining because despite the advances that have been made in technology in recent years, it has not yet been possible to create a program that learns completely on its own through its interaction with users and does not depend on at least one programmer in the evaluation and retraining phase. However, the goal of this automation is to rely on a minimum of one person for the program to adapt, and not to require fundamental changes to its base algorithm.

Retraining is simplified as much as possible and programming structures are used at its base so that the information remains consistent throughout this retraining phase.

Table 1. Comparison of manual and semi-automatic retraining.

Manual	**Semi-automatic**
Difficult to maintain	Easy to maintain
Tighter monitoring of chatbot interactions	No code modification required
Requires code modifications	Reduced retraining times
High retraining times	Use a graphical interface
Changes are made to the data or code	

2.3 Is Manual or Semi-Automatic Retraining Better?

While both retraining's require one or more programmers to be evaluated and readapted, it certainly requires less effort to maintain semi-automatic retraining. See Table 1. In addition, because manual retraining is usually very similar to training in design and construction stages, it will often require modifying the code or modifying the data from its containers.

One of the biggest advantages of semi-automatic retraining is the use of a graphical interface, in this way you can visualize the data and perform different actions on it, such as creating, reading, editing, and deleting it. In addition, a better data flow can be ensured, ensuring that the information is complete regardless of the data entered by the person in charge of the retraining.

3 Responsive Web App

Web applications "are programs that work on the Internet. In other words, the data or files you work on are processed and stored within the website. These applications usually don't need to be installed on your computer." GCFGlobal [5]. This framework includes the front-end and back-end parts, which comprise the application as a whole, but take care of different processes.

3.1 Back-End in a Web Application

The back-end comprises the data and logic part in a web application, and is hidden from the user [5].

Different frameworks or technologies can be used to handle this part, but typically two technologies are used together, one for data handling, and one for logic and functional structure.

3.2 Front-End in a Web Application

The front-end comprises the user-interactive parts, which are displayed in the browser, and is visible to all users [5].

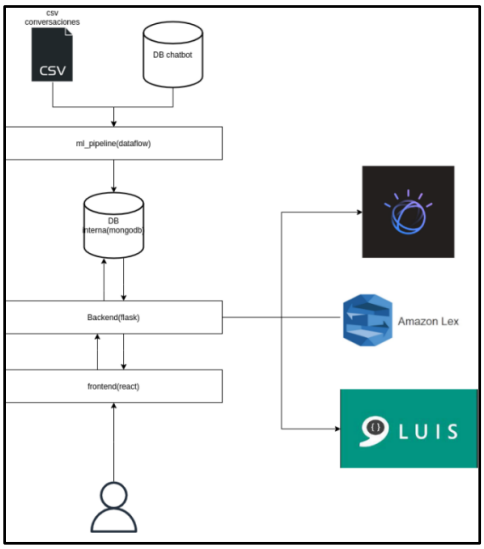

Fig. 1. Web Application Proposal Design Scheme (2022).

The front-end is one of the parts with the largest number of frameworks and options to work on it. This is due to the fact that the visual or 'interface' part is mainly worked on.

3.3 Web Application Proposal

Guided by Kevin G. Zazueta and Ricardo Cantú, engineers at NDS Cognitive Labs, it has been possible to develop a proposal for a web application for the retraining of chatbots, see Fig. 1.

For this implementation, the use of the Flask framework is proposed to handle the back-end, and React for the front-end, while the data will be managed with MongoDB.

The main objective of this development is to be a robust system that can retrain several different chatbots and is easy to use for the user.

3.4 Why a responsive Web App?

As the internet has evolved, it is increasingly used on mobile devices than on the computers where it was originally used. This has been a motivation to adapt as many programs as possible to versions that can be seen on screens of different resolutions.

While it is true that it is more effective to take advantage of a larger interface for a program with the complexity proposed, limitations must be considered with larger screens, for example, that a user has to travel and during the journey it is easier to continue with their work from the mobile device. This is why the responsive part must be considered, that is, depending on the size of the screen resolution, it adapts to it.

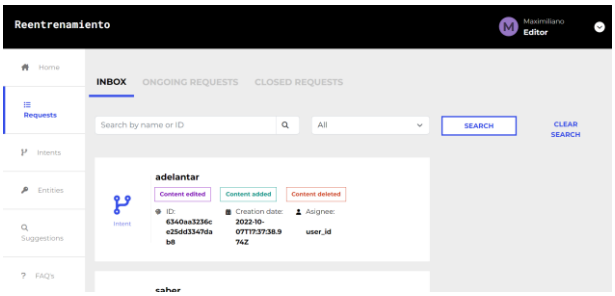

Fig. 2. Requests page view in chatbot retraining web app (2022).

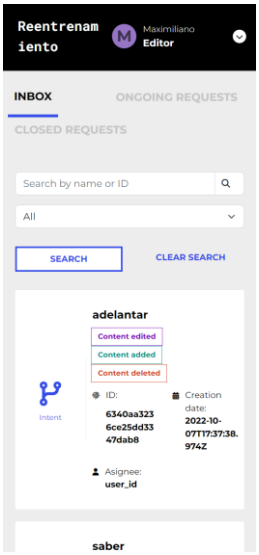

Fig. 3. Requests page view in chatbot retraining responsive web app (2022).

4 Front-End Implementation

For the implementation of the front-end in conjunction with NDS Cognitive Labs, we work mainly with Javier A. Valdivia on the functionalities required in the system.

Under the framework discussed in this article, we seek to make an implementation of the front-end part, the visual part of the program, that is adaptable to different screen resolutions. See Fig. 2. and 3. And from which you can access different editable options of the chatbot, and even change chatbots to be retrained.

The design proposal is created by NDS Cognitive Labs, and we are entrusted with the technical functionalities of the visual elements, such as buttons, menus, browsers and user interaction elements.

As functional components are built, especially those that are repeated throughout the application, individual pages are then built, which use some of these same components and others to execute their required functionalities.

5 Conclusions

In summary, we underline the importance of the chatbot retraining phase in the life cycle of these computer programs, and in this way the focus on the program developed for this same function is justified.

Although many different technologies have been selected and are even used in part for the construction of this system, ReactJS takes a leading role due to its ease of implementation and ability to interact with the Back-End implemented with MongoDB.

Together with NDS Cognitive Labs, specially led by Kevin Zazueta and Ricardo Cantú, the implementation team has been able to bring to reality an interface capable of semi-automating the chatbot retraining process. In addition to including parameterized functionalities to be able to adapt to different systems as needed.

References

1. Artyco: Chatbots: the customer service revolution. https://artyco.com/los-chatbots-la-revolucion-de-la-atencion-al-cliente (2022)
2. Brush, K., Scardina, J.: Chatbot. TechTarget. https://www.techtarget.com/searchcustomer experience/definition/chatbot (2022)
3. Bautista, I.:Backend and Frontend, What is it and how do they work in programming?https://www.servnet.mx/blog/backend-y-frontend-partes-fundamentales-de-la-programaci on-de-una-aplicacion-web (2022)
4. Enzyme: Chatbot retraining: essential to help the user. Available: https://enzyme.biz/blog/reentrenode-chatbots-imprescindible-para-ayudar-al-usuario (2022)
5. GFCGlobal: What are web applications? https://edu.gcfglobal.org/es/informatica-basica/que-son-las-aplicaciones-web/1/ (2022)