

# Development of a Front-End with Dynamic Searches for Chatbot Retraining Using ReactJS

Marco A. Bosquez-González<sup>1</sup>, Luis E. Bojórquez-Almazán<sup>1</sup>,  
Kevin G. Zazueta-Sánchez<sup>2</sup>, Yoel Ledo-Mezquita<sup>1</sup>,  
Elioth Macias-Frotto<sup>1</sup>

<sup>1</sup> Tecnológico de Monterrey,  
School of Engineering and Sciences,  
Mexico

<sup>2</sup> NDS Cognitive Labs.,  
Mexico

{a01653247, a01336625, yledo, a00837025}@tec.mx,  
kzazueta@ndscognitivelabs.com

**Abstract.** This article will show the development of a web platform using ReactJS for dynamic document searches, and the collaboration of a front-end team in the project "Virtual Assistant Management Platform" for Chatbot Retraining. Copyright © 2022 Tec de Monterrey and NDS Cognitive Labs.

**Keywords:** Chatbots, dynamic search, front-end, reactJS, computing.

## 1 Introduction

In order to give a real work experience to Tecnológico de Monterrey students, calls were opened for stays or professional internships that would begin for the semester of the period August – December 2022. One of the companies that would be part of this experience is NDS Cognitive Labs, which is located in Mexico and the United States.

NDS Cognitive Labs is responsible for developing cognitive computing and innovating technology in Mexico. To this end, it was decided to enrich the profile of the students by participating in professional experiences and developing real skills [1].

In conjunction with Tecnológico de Monterrey and NDS Cognitive Labs, these calls were opened to Tecnológico de Monterrey students. At the opening of the interviews, the professor and person in charge of communication between NDS and Tecnológico de Monterrey, Dr. Yoel Ledo, together with the coordinator of the professional stay on behalf of NDS Cognitive Labs, Dr. Raúl Morales, presented the different experiences.

These experiences were assigned based on the student's profile, in this way the most suitable one for the members could be defined. For this article, experience 3, "Virtual Assistant Management Platform", will be highlighted. This experience is described as the development of a Natural Language Processing (NLP) platform, which would use tools such as ReactJS, Javascript, Python, MongoDB, Google Cloud Platform, Flask and IBM Watson.

As part of this experience, a chatbot was built using IBM Watson, which handles artificial intelligence (AI) and machine learning (ML) technologies [2]. The foundation

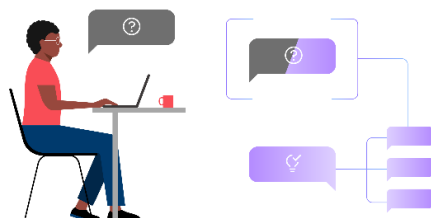


Fig. 1. IBM. (October, 2022). Leading conversation.

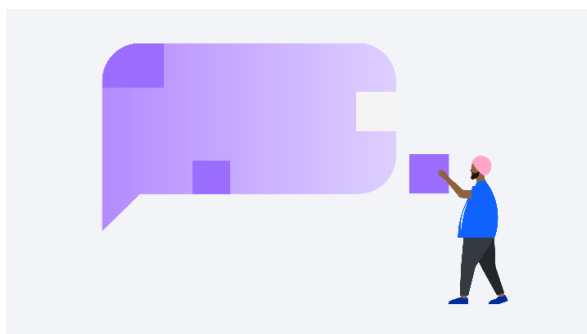


Fig. 2. IBM. (October, 2022). Simple construction.

of the IBM Cloud platform made it possible to create a Bot based on an advanced and publicly available machine learning system.

Chatbots are a useful way for customers to ask questions and get answers right away. The Bot can answer questions or direct the customer to other resources. These chatbots can run within a wide range of devices, applications, channels, and mobiles [3]. Several webpages were also used as the basis for the construction of our own model [4-6].

## 2 Materials and Methods

### 2.1 Second Level Title

Through IBM Watson that was mentioned above, the communication tool between the chatbot and the customer was created. The input of information in this case would be the client's request that would then be processed through natural language to then choose the best possible response.

For the creation of the dialogue with the chatbot, it is important to use methods that represent the user's information that is relevant to the user's purpose. These tools are known as intentions and entities [7].



Fig. 3. IBM. (November, 2021). Comprehensive.

Intentions are defined as verbs and actions that the user wants to perform, while entities are nouns such as objects, places, or things related to a business. The chatbot gathers and analyzes them to get a possible answer.

## 2.2 Project Context

By using the technologies and tools mentioned above. The project of this work experience sought to develop a platform in which the creation, editing, visualization, deletion and search of intentions and entities can be managed, in order to have all the possible answers to questions that are asked.

By correctly defining the intentions and entities, the aim is to gather the relevant information to determine the purpose of the request requested by a potential client. It has the ability to assign names, assign different types of values, and bind intents and entities. With this data, the chatbot algorithm developed by IBM Watson was trained.

For this project, the MongoDB NoSQL database was used, in which documents are stored, which are data records in BSON-formatted objects. These documents are separated into collections, in which intentions and entities are separated. In this way, this database is used to perform CRUD functions on these data types.

In conjunction with MongoDB and IBM Watson, we developed a platform that allows us to handle CRUD functions of intents and entities in a better way. This platform is accessible to users in a way that can be accessed from the computer or a mobile device.

All this set of technologies, methods, natural language processes and data help us to be able to retrain the chatbot with all the possible answers to all the possible questions that the user has.

## 2.3 Work Sharing

Taking into account the technologies and tools to be used, as well as the profile of the members of the experience, the work to be carried out was distributed among the first



Fig. 4. MongoDB. (2022). Homepage Hero. MongoDB, Inc.

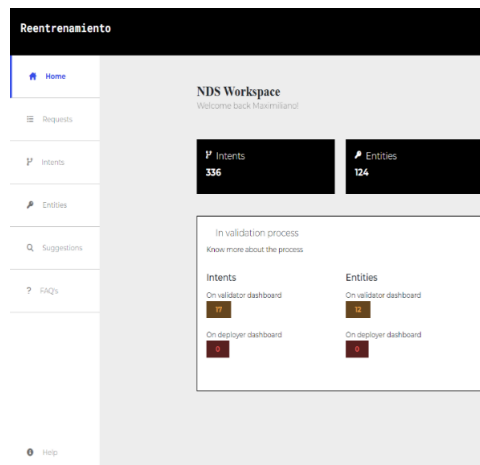


Fig. 5. (2022). Chatbot Retraining Platform.

weeks of working in the team. The work teams within this project were split between the back-end team that would work with Python, Flask, and MongoDB, and the front-end team that would work primarily with ReactJS.

Seven students from Tecnológico de Monterrey took part in this experience, so, without taking into account the leaders of the experiences, initially 5 people would be distributed in the back-end team and 2 people in the front-end team.

The front-end team to which the authors of this article mainly belong, were selected due to their previous experiences with the open-source front-end JavaScript library, ReactJS. Likewise, due to previous experiences in Python and MongoDB, you would have the opportunity to collaborate and contribute to the back-end team.

The experience leaders would give the instruction to take MongoDB courses on their official page to relate to the database and its relationship with the development of the platform.



Fig. 6. (2022). Search Bar.

Intent	Description	Entities	Examples	Conflicts	Modified
<a href="#">no_intent</a>	No definition	7	7	0	Date
<a href="#">no_intent</a>	No definition	2	2	0	Date
<a href="#">no_intent</a>	No definition	2	2	0	Date
<a href="#">no_intent</a>	No definition	1	1	0	Date
<a href="#">no_intent</a>	No definition	2	2	0	Date

Fig. 7. (2022). Table of Intentions.

Entity	Intents linked	Synonyms	Modified
<a href="#">Diseases</a>	1	1	2 weeks ago
<a href="#">Diseases</a>	0	1	2 weeks ago

Fig. 8. (2022). Table of Entities and Search Bar.

As for the way the team works, the members of the front-end team, Marco A. Bósquez González and Luis E. Bojórquez Almazán, were presented with the base platform with which they were going to work. The base of the platform has the defined components to work with and that would need to be operationalized. Likewise, it was tested to observe the flow of the platform simulating its use for end users.

The components to be worked on include working the creation, editing, viewing and deletion screens, using ReactJS. At the back of the page, endpoints developed by the back-end team would be used to extract the information and data necessary to give the platform and components a proper function.

## 2.4 Communication & Development

As for the communication and development of the project by the back-end team, every week meetings are scheduled through the Google Meets platform in which the weekly progress is shown and the functionalities to be developed for the next reviews are discussed. Additionally, in order to have a record of the progress of the project, weekly logs were written about the main points of each week and the constant progress of the application. This was demonstrated by evidence that is included in the logs.

The project leader, engineer Kevin G. Zazueta Sanchez, reviewed the progress of the team consisting of Marco A. Bosquez Gonzalez and Luis E. Bojorquez Almazan. In the same way, in case of needing support from the leader, video calls are scheduled in order to resolve doubts and develop the aforementioned components. As for the way the front-end team works, the members worked in different perspectives in which the

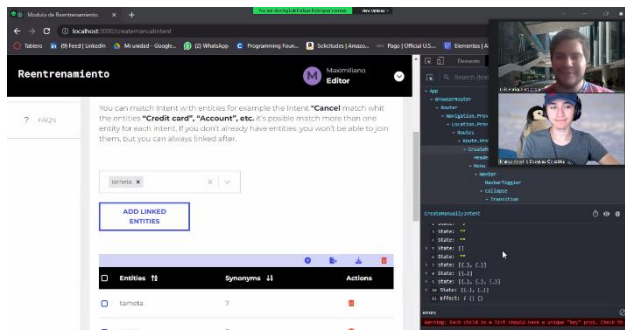


Fig. 9. Marco A. Bosquez González, Luis E. Bojórquez. (2022). Zoom meeting.

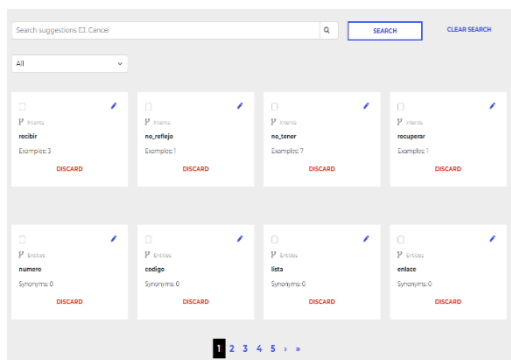


Fig. 10. (2022). Suggestions View with Search Bar and Selection of Options to Display.

defined components were developed. This is in order to get the components up and running as quickly as possible and move on to the next components. Sometimes the support of both members is required to solve a problem, so there is constant communication in the front-end team.

The front-end project works with a repository on GitLabs in which versions are maintained that different team members can work with. The front-end team, which is composed of Marco A. Bósquez González and Luis E. Bojórquez Almazán, worked mainly on all the views that include intentions, entities, and suggestions, so they worked on a specific branch.

### 3 Results

By using ReactJS and the endpoints created by the back-end team, the search bar component could be worked on and added in most of the platform's views, including intents, entities, suggestions, and requests.

The search bar makes use of inputs, useState useEffect hook functions, component rendering, Axios, and components from React's library, MaterialUI (MUI). All of this makes up the search bar and its buttons that update every time you type or perform a search.

Intents	Description	Entities	Examples	Conflicts	Modified
<a href="#">no_borrar</a>	No definition	7	7	0	Date
<a href="#">no_borrar</a>	No definition	2	2	0	Date
<a href="#">no_terminar</a>	No definition	2	2	0	Date
<a href="#">salir</a>	No definition	11	11	0	Date
<a href="#">salir</a>	No definition	2	2	0	Date

Fig. 11. (2022). Table of Intentions.

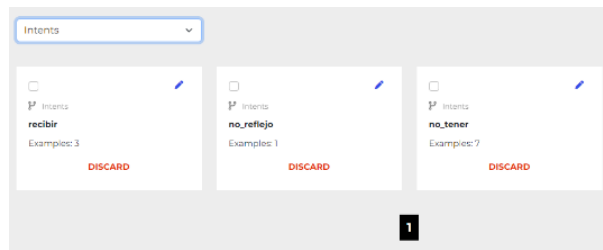


Fig. 12. (2022). Table of Suggestions with the option to only show Intents.

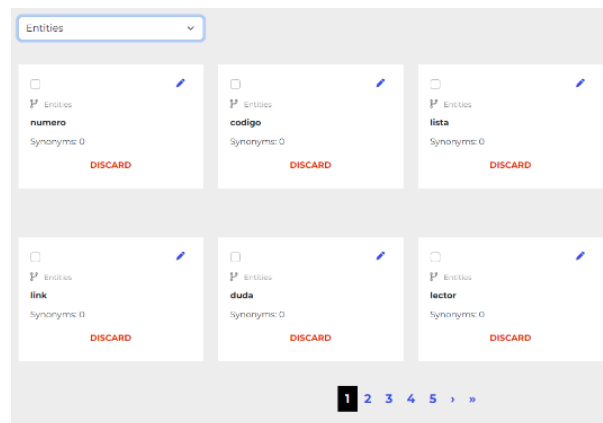
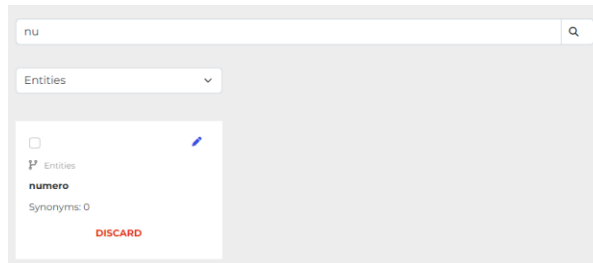


Fig. 13. (2022). Table of Suggestions with the option to only show Entities.

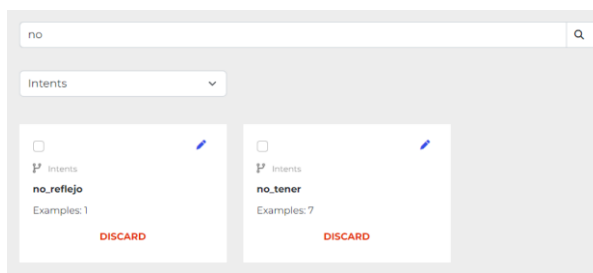
You can make use of Axios and React Hooks to store lists of objects that are fetched from the MongoDB database in state variables. State variables are Hook functions that render and update in the lifecycle of the page. These Hook functions are preferred over normal variables, as the normals disappear instead of the Hooks that are preserved by React.

In this case, by calling an endpoint with the GET method in Flask on the back-end side, we get a list of objects that we can then identify and list on the page. By making use of and manipulating the lifecycle and state cycle of a Hook use State, variables can be updated and search filters can be performed on a list.

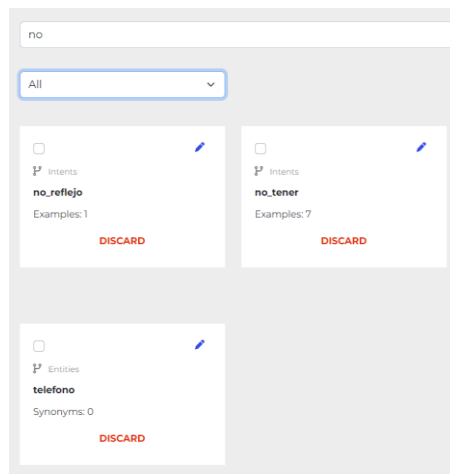
So, depending on whether the list is, whether it's a list of intents, a list of entities, or both, you can do a search function on whatever is listed on the page and then update that list that you render.



**Fig. 14.** (2022). Searching for "a" among the list of entities.



**Fig. 15.** (2022). Searching for "no" among the list of intentions.



**Fig. 16.** (2022). Searching for "no" among the list of intentions and entities.

Likewise, when you search or select an option to display, a pagination is updated that limits the number of objects that render on the page.

The dynamic search bar allows you to search for intents or entities by name, which helps to filter the intentions or entities that need to be viewed and then make a change to the database.

By using dynamic document search, you can access all the functions that an intent or entity has. Whether it's editing, viewing information, creating or deleting from the database.



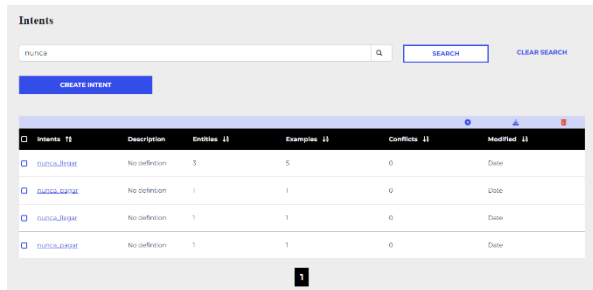


Fig. 17. (2022). Searching for "never" in the intentions section.

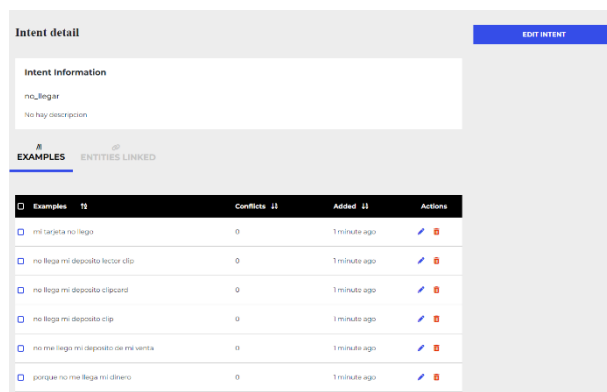


Fig. 18. (2022). Visualizing the details of the intent named "no\_llegar".

## 4 Conclusions

In conclusion, with everything that was mentioned in this article, the progress of the project has allowed us to generate solutions, front-end and methods of performing dynamic functions on the chatbot that help us fulfill the purpose of the project in developing a Natural Language Processing (NLP) platform.

Making use of various technologies such as ReactJS, Javascript, Python, MongoDB, Flask, and IBM Watson. In conjunction with React components and libraries such as Material UI (MUI), Hooks, and Axios, a platform is developed that is still in process for generating, creating, editing, visualizing, linking, and deleting intents, entities, and suggestions.

## References

1. NDS CONGNITIVE LABS: Rapid, cost-effective implementation of digital technologies. <https://ndscognitivelabs.com/company-page/> (2022)
2. Burns, S.: Contextual entities with IBM Watson assistant: IBM Watson assistant. IBM watsonx Assistant. Accessed: <https://medium.com/ibm-watson/contextual-entities-with-ibm-watson-assistant-f41b2e0ca82e> (2022)

*Marco A. Bosquez-González, Luis E. Bojórquez-Almazán, Kevin G. Zazueta-Sánchez, et al.*

3. IBM: Assistant: Virtual agent. IBM. <https://www.ibm.com/products/watson-assistant> (2022)
4. MongoDB: The developer data platform. MongoDB. <https://www.mongodb.com/> (2022)
5. React: Using the state Hook. <https://reactjs.org/docs/hooks-state.html> (2022)
6. Axios: Getting started. <https://axios-http.com/docs/intro> (2022)
7. Duraj, M.: How to build a chatbot using IBM Watson. <https://www.pluralsight.com/guides/how-to-build-a-chatbot-using-ibm-watson> (2022)