Modelling, Measurement and Control D
Vol. 45, No. 1-4, December, 2024, pp. 79-86
Journal homepage: http://iieta.org/journals/mmc_d

Chatbots: Transformative Agents in Modern Industrial Practices

Fernando Aragon-Rogelio*, Iván Juan Carlos Pérez-Olguín, Israel Soto, Israel U. Ponce

Institute of Engineering and Technology, Universidad Autónoma de Ciudad Juárez, Ciudad Juárez 32310, Mexico

Corresponding Author Email: angel.soto@uacj.mx

https://doi.org/10.18280/mmc_d.451-412

Received: 8 September 2024
Revised: 28 October 2024
Accepted: 17 November 2024
Available online: 31 December 2024

Keywords:
chatbot, industry, artificial intelligence, programming, Dialogflow CX, machine learning, Python

ABSTRACT

We present the development of a chatbot designed to assist industries in analyzing their databases and providing relevant information to support decision-making. The chatbot is built using the Dialogflow CX platform, which offers tools for its creation, incorporating machine learning and artificial intelligence functionalities. Additionally, specialized Python code facilitates the connection between Dialogflow CX and Python, enabling the upload and analysis of user databases through an external web page. Despite the challenges posed by the creation of the Python code, which were the correct selection of libraries, since for the processing, analysis, and response of the selected information, we should op t for options that can capture all the information, in the case of some libraries only took segments of the file provided, and for the use of this case study, we opted for the Pandas package for its simple and effective use for this chatbot, the result is a responsive chatbot capable of effectively managing database loads and providing insightful analysis to users.

1. INTRODUCTION

This research explores the use of chatbots as a tool for the analysis of industrial databases. Within industries, there is a wide diversity of databases, each managed by specialized areas familiar with their structure, special characters, and specific language. Handling large volumes of information requires specialized software for storing, distributing, and categorizing data. Several widely used software solutions in industries include:

- STEL Order, a program supporting sales billing within companies, is accessible via smartphones, laptops, and tablets. It reduces invoice creation time by storing customer and supplier information [1].
- SAP, one of the most comprehensive programs for companies. Offering modules that support various areas for decision-making. It requires specific expertise and is used in small and medium-sized enterprises [2].
- Odoo, an "all-in-one" management software that automates commercial, operational, and administrative processes with specialized solutions for different types of companies. It is a comparable alternative to SAP [3].
- HubSpot, focused equally on sales management and identifying new market niches, is functional for small businesses. It is user-friendly, and its license is affordable and easily accessible [4].
- Notion: A productivity tool that helps companies set objectives for improving critical areas within their processes. It can be related to Google Drive but is aimed at businesses [5].

These software solutions facilitate data management in various ways. However, when it comes to extracting relevant information from these databases through specific queries or everyday language, a chatbot might be the optimal solution. Chatbots, primarily used in customer service due to their human-like communication capabilities and 24/7 availability, can be implemented in industrial settings to analyze databases and provide relevant information. This approach allows users to submit less structured queries, with the chatbot identifying relevant information quickly and easily, through natural language processing (NLP).

To develop the chatbot, Dialogflow CX will be proposed. This platform enables the creation of chatbots with artificial intelligence (AI) and machine learning capabilities, allowing for the programming of different inputs for comprehension. Additionally, Python will be used to create the code that uploads data files and delivers the desired answers. Therefore, the scope of the research is as follows:

- Design the graphical interface.
- Development of filters for Chatbot responses.
- Creation of the code.
- Chatbot learning.
- Chatbot implementation.
- Monitoring and Adjustments.

With the following research questions:

- How does the graphical interface design influence the user experience when interacting with a Chatbot?
- What is the impact of implementing filters in the Chatbot responses to improve the accuracy and relevance of the responses given?
- How does the development of filters in Chatbot influence the optimization of everyday language processing?
- What methods and strategies are most effective for

learning Chatbot to adapt to different contexts and communication styles?
- How can the effectiveness of the Chatbot implementation in specific environments be measured and evaluated, considering its usability, efficiency, and user satisfaction?

2. THEORETICAL FRAMEWORK

2.1 Artificial intelligence

AI involves machines that emulate human thinking and behavior, enabling them to think, learn, and perform tasks typically requiring human intelligence. AI can perform tasks such as playing chess, generating images from descriptions, and comprehending multiple languages [6]. It represents a scientific field focused on revolutionizing technology and its applications.

AI leverages algorithms and learns from extensive datasets to make decisions like human reasoning. It can handle multiple tasks with a minimal margin of error compared to humans and operates more efficiently without requiring downtime [7]. According to Ponce [7], AI comprises three main branches: fuzzy logic, artificial neural networks, and genetic algorithms.

2.1.1 Fuzzy logic

Fuzzy logic is used to create mathematical estimates for solving problems where the data is imprecise, making it widely applicable in electronic and computational contexts [8]. This is due to the presence of variables that are often difficult to control. For instance, the concept of "warm" can vary significantly: For someone from Alaska, it might mean temperatures above ten degrees Celsius, while for someone from Mexico, it could mean temperatures above thirty degrees Celsius. These types of data are known and not considered fuzzy. Fuzzy data or logic involves data with unclear boundaries or that are not well-defined [9].

2.1.2 Artificial neural networks

Artificial neural networks (ANNs) attempt to emulate the behavior of the human brain, characterized by learning from experience, acquiring knowledge, and converting it into data for study [10]. These networks are inspired by biological neural networks. According to Veintimilla and Cisneros [11], an artificial neural network is structured into three layers:
- Input layer, receives the information.
- Hidden layer, interacts internally without any external contact. It consists of levels that are interconnected in various ways, depending on the input received.
- Output layer, which provides the result that the neural network has determined.

Artificial neural networks can solve problems either individually or in combination with other methods, yielding a variety of solutions. This system mimics the human nervous system [12]. The key properties of Artificial Neural Networks include their ability to learn from the training data provided [13]. During training, large amounts of data are introduced so that the neural network can automatically discover patterns and relevant hidden information to achieve the objective model [14].

One of the most extensively studied types of neural networks is the Feedforward Neural Network. It is characterized by a set of neural networks that receive, classify, process information, and provide multiple similar responses [15]. This type of neural network consists of multiple layers: an input layer, one or more hidden layers, and an output layer. The prediction process in feedforward neural networks involves data moving through the different input layers and hidden layers before producing an output. Each piece of data passes through every hidden layer of the neural network to generate the most accurate output or the best set of options for the output.

2.1.3 Genetic algorithms

When it comes to genetic algorithms, it refers to obtaining a group of individuals that possess the best qualities or capabilities to reproduce them so that the next generation is better than the previous one [16].

It is a programming technique that seeks to solve problems through biological evolution, in which a fitness function is used to quantitatively evaluate each proposed candidate. These candidates are known to already have solutions for previous problems that are similar, but the purpose is that the genetic algorithm improves them in the next programming or generation [17].

2.2 Machine learning

The machine learning concept was introduced by Samuel, in 1959, describing it as the ability of computers to perform tasks without being explicitly programmed for them [18]. Machine learning, a branch of AI, aims to develop programs or applications that can produce fast and accurate results. By training on a set of data, machine learning models can identify trends, patterns, and relevant information to make predictions on previously unknown data [19]. In contrast to traditional programming, where results are obtained through predefined rules, machine learning relies on data to learn and evolve. Machine learning encompasses a wide range of concepts and techniques designed to achieve various objectives.

2.2.1 Recurrent neural networks

This type of network has the characteristic of processing and obtaining information from sequential data; a clear example is video analysis, image captioning, and NLP [20].

In traditional neural networks, data input produces a single output, which can sometimes complicate the model. Recurrent neural networks, however, can handle sequences of data, making them suitable for tasks such as speech recognition and voice identification [21]. They work by continuously improving their performance through the use of feedback loops, which allow them to retain and use information from previous inputs to influence future outputs [22].

2.2.2 Generative adversarial networks

This type of neural network consists of two main types of neural networks: a generative network and a discriminative network. The generative network creates data samples, while the discriminative network classifies these samples as real or fake. The generative adjusts its data until the discriminative can no longer distinguish between real and generated samples, aiming to produce data that closely mimics the true distribution [23].

The generative network starts with random variables as input and tries to generate data that resembles the target distribution. Conversely, the discriminative network is trained to differentiate between real data (true samples) and fake data (generated samples). The discriminative network is trained with labeled images that are part of the target value and other images that are not so that the network can be identified, while the generative network is released in the same way as the discriminative one, but with opposite purposes, so both networks seek to identify which network is failing or does not correctly detect the target value [24].

2.2.3 Natural language processing (NLP)

It is a field of AI and linguistics that focuses on the interactions between computers and human language [25]. NLP involves using natural language to communicate with computers, making the development and execution of tasks more intuitive and efficient [26]. There are various methods within NLP to transform everyday language into a format that can be used in programming. These methods and models enable the application of human language in computational processes, facilitating improved communication and functionality. Some of them will be presented in the next sections.

2.3 Dialogflow CX integration complementation with Python

Based on the study [27] tells us how to start with a database for the implementation of the chatbot with the preparation of intents that seeks a chatbot that understands the most simplified artistic language, the musical, for which it requires answers related to artists and musical notes, and with the support of Python makes a better interaction for obtaining results.

On the other hand, in the study [28] through the use of a fighting video game, a data analyzer or chatbot to ask questions about the video game through the tools of interest and mentions the facilities of the implementation of Dialoglfow CX and Python so that both interact within the video game with some difficulties that were presented in this project and are of support to have a clearer path to the development of the chatbot.

3. METHODOLOGY

As previously mentioned, the development of the chatbot in this research will be carried out using the Dialogflow CX platform along with the Python programming language.

3.1 Creation of the Chatbot, skeleton in Dialogflow CX

To use the Dialogflow CX platform, a Gmail account is required, which is free to create. However, using Dialogflow CX also necessitates an additional registration through the Gmail account, and you may be required to enter credit card information.

Figure 1 shows all the functions provided to create the Chatbot for the project. In the Manage tab, you will find different features that enable the Chatbot work effectively.

3.1.1 Intents

Intents represent the user's intention during a conversation. Each intent is labeled to categorize and distinguish it from others, preventing confusion. In this section, the chatbot is trained with specific phrases so it can recognize user inputs and respond appropriately, leading to the next flow or page.

Dialogflow CX provides default welcome and closing intents to manage the start and end of interactions with users. These can be customized to deliver messages that best suit the programmer's needs.

The first step in creating an intent is to give it a name and provide various training phrases. When the user inputs a phrase that matches or closely resembles one of the training phrases, the intent is triggered. Figure 2 shows an example of common phrases used to identify the user's language and intent.

3.1.2 Flows

Complex dialogs often include several topics of conversation. For example, a pizza delivery agent may have separate flows for food ordering, customer information, and order confirmation. Each topic requires multiple turns of conversation for the agent to gather the relevant information from the user. For the purposes of this chatbot, only one flow is used to optimize the interaction.

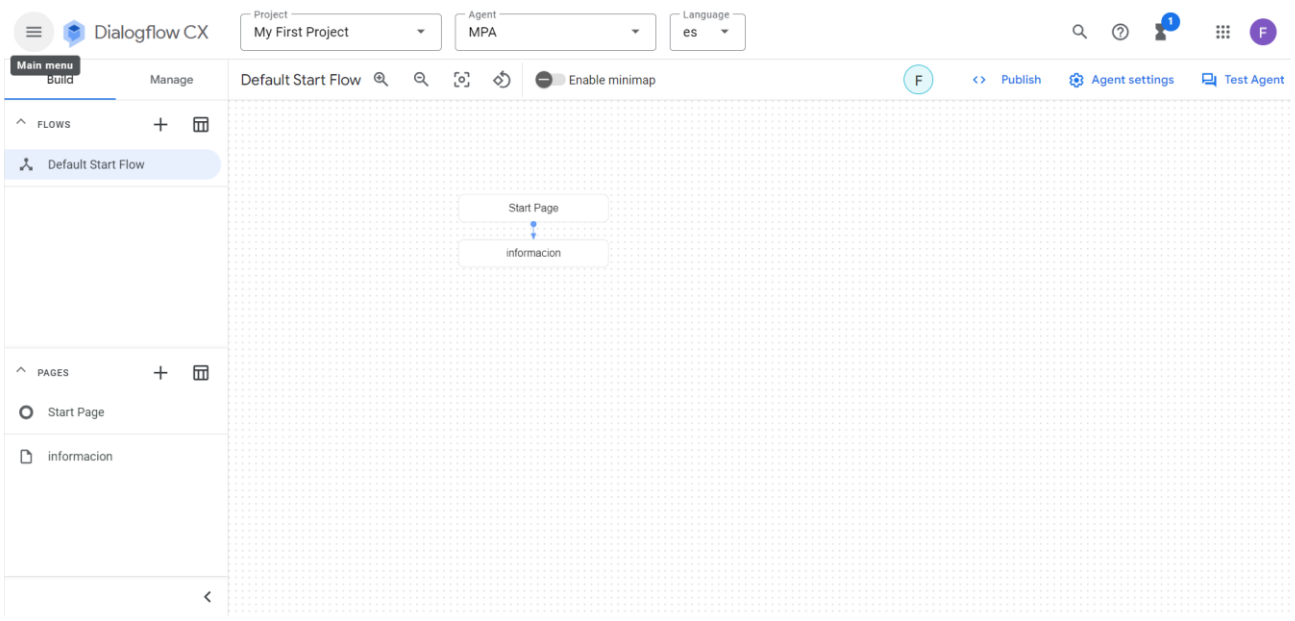
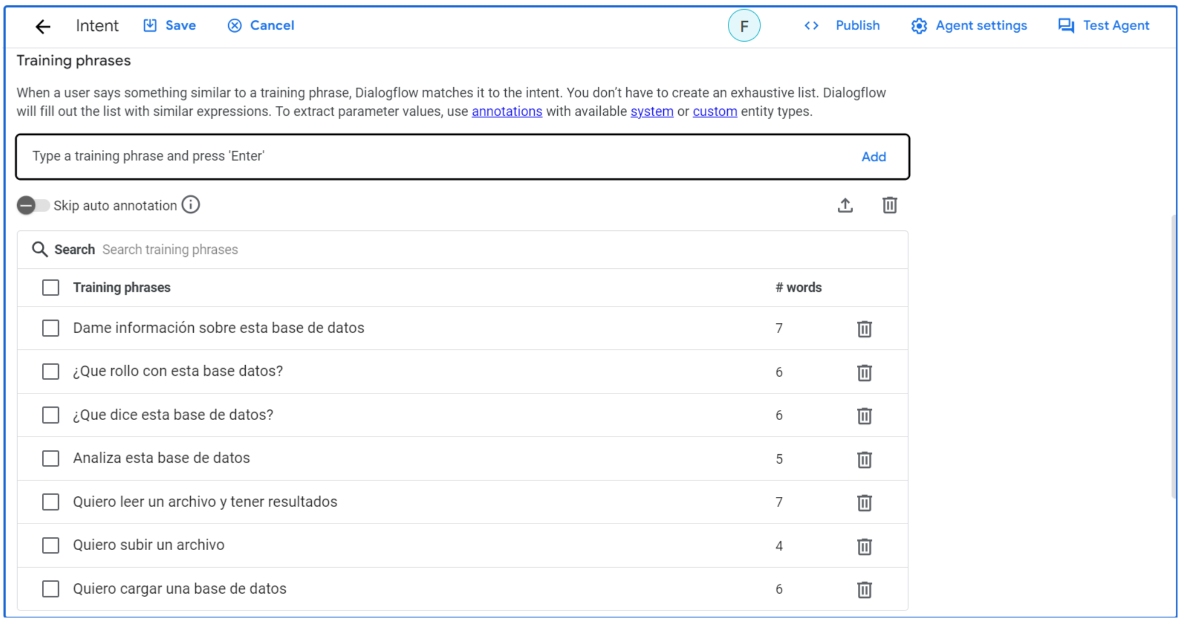


Figure 1. Dialogflow CX interface


Intent Save Cancel Publish Agent settings Test Agent

Training phrases

When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use annotations with available system or custom entity types.

Type a training phrase and press 'Enter' Add

Skip auto annotation

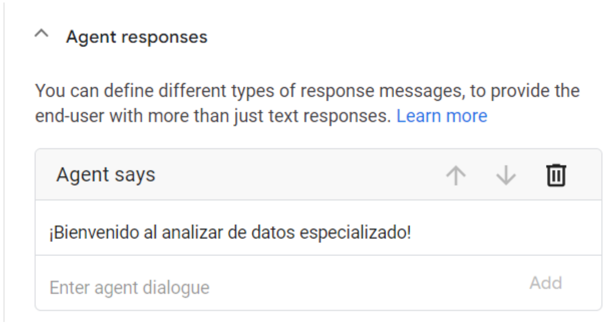
Search Search training phrases

Training phrases	# words
Dame información sobre esta base de datos	7
¿Que rollo con esta base datos?	6
¿Que dice esta base de datos?	6
Analiza esta base de datos	5
Quiero leer un archivo y tener resultados	7
Quiero subir un archivo	4
Quiero cargar una base de datos	6

Figure 2. Creation of the Intent LoadDatabase

3.1.3 Pages

At this point, two pages are used; the first page contains the welcome and closing intents, while the second page handles the intents related to loading the database.


Agent responses

You can define different types of response messages, to provide the end-user with more than just text responses. Learn more

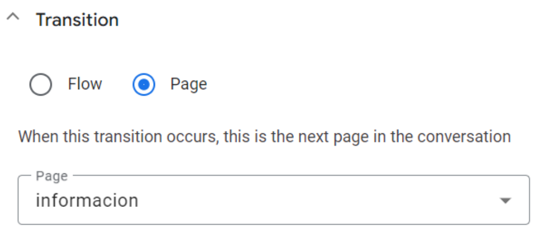
Agent says

¡Bienvenido al analizar de datos especializado!

Enter agent dialogue Add

Figure 3. Creation of welcome message

The first page, called the Start Page, is customized with a welcome message for the user. Figure 3 illustrates the features contained in this page.


Transition

Flow [✓ Page]

When this transition occurs, this is the next page in the conversation

Page: informacion

Figure 4. Transition method between pages

Figure 4 shows the page jump when the user uses a word that matches the training words previously programmed. In the second page, named information, which contains the Intents that were created previously, you have the feature that contains webhooks.

3.1.4 Webhook

A webhook is a service that hosts various specialized programming functions. In this case, a webhook named Linkparadatos has been created to serve as a bridge between Dialogflow CX and Python. This integration allows the chatbot to communicate with Python scripts, facilitating more complex operations and interactions beyond the capabilities of Dialogflow CX alone.

3.2 Installation of Python package

Before coding in Python, it's crucial to install the necessary packages. The essential packages for this project include:

- Flask, for creating web applications.
- Pandas, for CSV and Excel files processing.
- Openpyxl, for handling Excel (XLSX) files.
- Xlrd, for handling Excel (XLS) files.

To install these packages, open the command prompt and follow the instructions shown in Figure 5. Ensure that all packages are installed correctly.

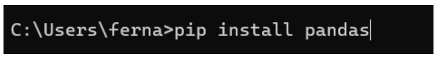

```
C:\Users\ferna>pip install pandas
```

Figure 5. How to install packages in the operating system

3.3 Creation of the code

The code development is divided into three sections:

<u>*Path to handle Dialogflow CX Webhook requests*</u>

```
@app.route('/webhook', methods=['POST'])
def webhook():
  req = request.get_json()
  print(f"Request received: {req}")

  # Procesa los datos recibidos del webhook
  intent_info = req.get('intentInfo')
  if intent_info:
     intent_name = intent_info.get('displayName')

  if intent_name == 'AbrirPaginaCarga':
response = {"fulfillmentText": "Puedes subir tu archivo en el siguiente enlace: /upload"}
     elif intent_name == 'Default Welcome Intent':
```

```
response = {"fulfillmentText": "¡Bienvenido al analizar de datos especializado!"}
    else:
response = {"fulfillmentText": "No he entendido tu solicitud."}
    else:
response = {"fulfillmentText": "No se encontró información del intent."}
    return jsonify(response)
```

The previous code demonstrates the configuration process, establishing a connection through a provided URL. This URL is generated by another component, which will be reviewed in the next step of the methodology. By leveraging the pre-defined Intents in Dialogflow CX, this setup facilitates the delivery of messages and information from the database.

Path to handle file upload

```
@app.route('/upload', methods=['GET', 'POST'])
def upload_file():
  if request.method == 'POST':
try:

# Verifica que el archivo esté en la solicitud
   if 'file' not in request.files:
   return "No se ha seleccionado ningún archivo"

   file = request.files['file']
   if file.filename == '':
   return "No se ha seleccionado ningún archivo"

   # Guarda el archivo en el directorio 'uploaded_files'
   upload_folder = 'uploaded_files'
   if not os.path.exists(upload_folder):
   os.makedirs(upload_folder)

   file_path = os.path.join(upload_folder, file.filename)
   file.save(file_path)

   # Detecta la extensión del archivo y lo lee adecuadamente
   file_extension = os.path.splitext(file.filename)[1].lower()
   if file_extension == '.csv':
   try:
     df = pd.read_csv(file_path, encoding='utf-8', on_bad_lines='warn')
     except UnicodeDecodeError:
     try:
       df = pd.read_csv(file_path, encoding='latin1', on_bad_lines='warn')
       except Exception as e:
       return f"Error al leer el archivo: {str(e)}"
       elif file_extension in ['.xls', '.xlsx']:
         try:
           df = pd.read_excel(file_path)
           except Exception as e:
           return f"Error al leer el archivo: {str(e)}"
           else:
             return "Formato de archivo no soportado. Por favor, sube un archivo
             CSV o Excel."
           num_rows = df.shape[0]
           num_columns = df.shape[1]

       return f"Archivo subido exitosamente. El archivo tiene {num_rows} filas
         y {num_columns} columnas."

         except Exception as e:
         return f"Error durante el procesamiento: {str(e)}"
```

This section of the code analyzes the incoming database and displays a confirmation message if the upload is successful. If the database is invalid, an error message is generated for further review.

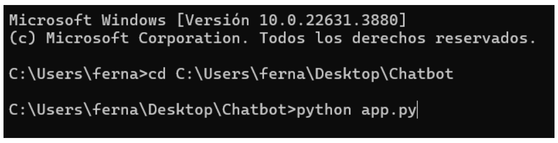
HTML form to upload files

Finally, a web page was created for database uploads, as the Chatbot interface was too complex to handle the large number of columns and rows. We opted for a separate page, provided by the Chatbot, to facilitate database uploads. The following code was used to develop this page.

```
html_form = '''
  <!doctype html>
  <html lang="es">
    <head>
      <meta charset="utf-8">
      <title>Subir Archivo</title>
    </head>
    <body>
      <h1>Subir Archivo</h1>
      <form method="post" enctype="multipart/form-data">
        <input type="file" name="file">
        <input type="submit" value="Subir">
      </form>
    </body>
  </html>
  '''
  return render_template_string(html_form)

if __name__ == '__main__':
  app.run(port=5000)
```

3.4 Obtaining the URL for the Dialogflow CX Webhook

Once the code is running, the next step is to obtain a URL for the Dialogflow CX Webhook. Follow the instructions in Figure 6 to run the Python code.

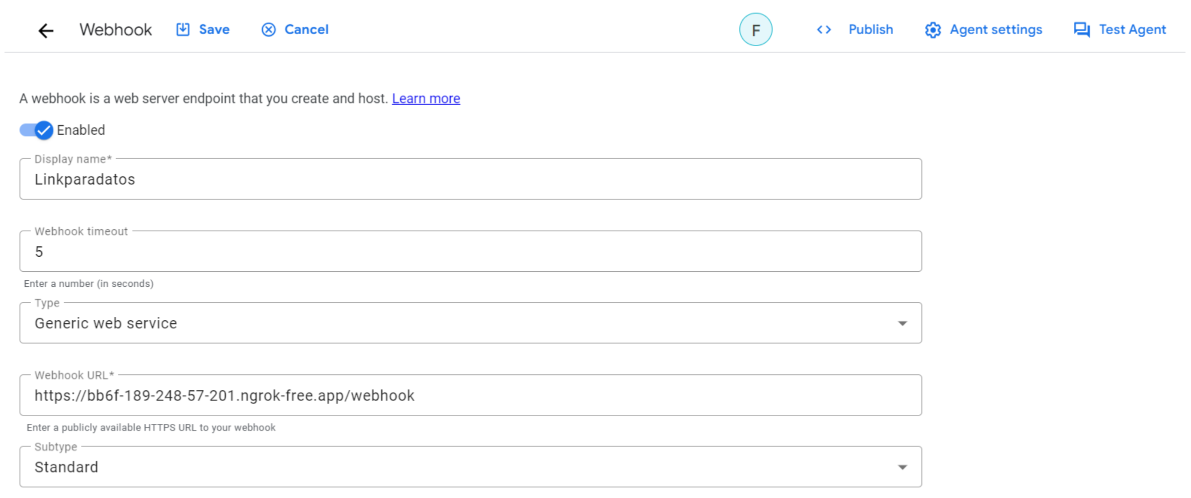


```
Microsoft Windows [Versión 10.0.22631.3880]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\ferna>cd C:\Users\ferna\Desktop\Chatbot

C:\Users\ferna\Desktop\Chatbot>python app.py
```

Figure 6. How to run the realized Python code

The following sequence should be displayed to identify that it works correctly and is ready for the next step.

```
C:\Users\ferna>cd C:\Users\ferna\Desktop\Chatbot
C:\Users\ferna\Desktop\Chatbot>python app.py
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```


Webhook Save Cancel F Publish Agent settings Test Agent

A webhook is a web server endpoint that you create and host. Learn more

Enabled

Display name*: Linkparadatos

Webhook timeout: 5

Enter a number (in seconds)

Type: Generic web service

Webhook URL*: https://bb6f-189-248-57-201.ngrok-free.app/webhook

Enter a publicly available HTTPS URL to your webhook

Subtype: Standard

Figure 7. Webhook configuration

Install the ngrok program, which provides a URL for the webhook. The port used in the Python code is necessary for the connection between Dialogflow CX and Python. It is shown in the yellow shaded part which URL is needed to use it in webhook.

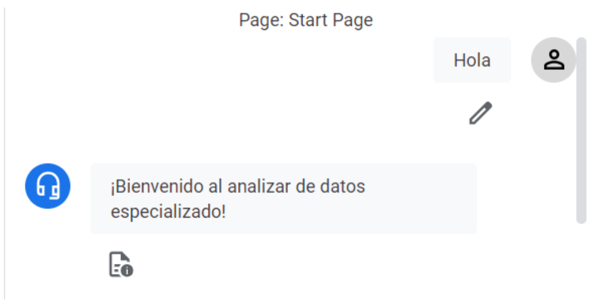
```
ngrok (Ctrl+C to quit)
Try our new Traffic Inspector: https://ngrok.com/r/ti
Session Status online
Account fernandoaragon063@gmail.com (Plan: Free)
Update update available (version 3.13.0, Ctrl-U to update)
Version 3.10.0
Region United States (California) (us-cal-1)
Web Interface http://127.0.0.1:4040
Forwarding https://61ee-189-248-4-89.ngrok-free.app ->
http://localhost:5000
Connections   ttl   opn   rt1    rt5    p50    p90
              0     0     0.00   0.00   0.00   0.00
```

In the Linkparadata Webhook configuration, enter the URL as shown in Figure 7.

With these steps completed, you can now use the agent test feature to run and test the chatbot.

4. RESULTS

When interacting with the Chatbot within the Dialogflow CX platform, users are presented with a user-friendly interface that is simple and intuitive. Figure 8 illustrates the interaction with the Chatbot and the responses it provides, which are configured and programmed using Python.


Page: Start Page

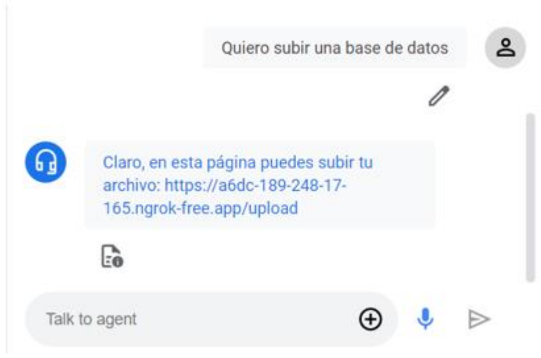
Hola

¡Bienvenido al analizar de datos especializado!

Figure 8. First message of interaction

The default message that was configured is functioning correctly and provides a welcoming greeting.

Figure 9 demonstrates how the connection between pages is established to ensure the Chatbot is functioning correctly. It provides a URL to upload the database. Upon entering the URL, two options for uploading the database are presented. Once the database is successfully uploaded, a confirmation message is displayed.


Quiero subir una base de datos

Claro, en esta página puedes subir tu archivo: https://a6dc-189-248-17-165.ngrok-free.app/upload

Figure 9. Keywords for change of page

File uploaded successfully. The file has 25276 rows and 20 columns.

In which identifies the size of rows and columns of the file to give the user to understand if it is the corresponding one. In the display of the ngrok command prompt, there are movements that refer to the connection between Dialogflow CX and Python, in addition to the functionality of the page to upload the databases.

```
GET /robots.txt      404 NOT FOUND
POST /upload         200 OK
GET /upload          200 OK
GET /favicon.ico     404 NOT FOUND
POST /webhook        200 OK
POST /webhook        200 OK
```

In the above programming, it indicates the functionality of the code and the configuration of the Chatbot. To test the performance of the Chatbot, testing was performed on 10 different documents that ranged from 56 to 90 pages of content, with the following performance metrics.

Table 1. Results 1

Characteristic	Test 1	Test 2	Test 3	Test 4	Test 5
Pages number	54	58	70	84	62
Loading speed	180s	185s	240s	250s	202s
Response time (questions < 80 characters)	120s	120s	180s	225s	225s
Response time (questions > 80 characters)	180s	170s	190s	245s	180s
Response quality	7	6	9	8	8
Error rate	Low	Half	Low	Low	Half
Characteristic	**Test 6**	**Test 7**	**Test 8**	**Test 9**	**Test 10**
Pages number	78	88	86	66	87
Loading speed	225s	270s	255s	220s	265s
Response time (questions < 80 characters)	200s	240s	225s	180s	225s
Response time (questions > 80 characters)	240s	296s	273s	195s	263s
Response quality	7	7	8	9	7

The time it took to load the files inside the Chatbot system was taken, as well as its time in rating the answer delivered by the Chatbot by two criteria about whether the question and answer contained less than 80 characters and more than 80 characters for the quality of the answer a Likert scale was used where 1 is an unfavorable answer and 10 is a concise and helpful answer, lastly in the error rate an ordinal scale of low, medium and high was used which presents whether the Chatbot can work with the file and the questions that the user can ask. In Table 1 you can see in detail the metrics used and the results that were obtained within the analysis and what performance the Chatbot obtained.

5. CONCLUSIONS

The use of chatbots in the industry has traditionally been limited to customer service or addressing minor issues that don't require complex analysis. Chatbots are designed to simulate human interaction, making it easier for users to address their concerns. The Dialogflow CX platform offers a range of tools that significantly simplify chatbot creation, enabling companies and individuals to adapt to emerging trends more easily. In the past, creating a chatbot required extensive programming knowledge, but modern platforms have streamlined this process, making it quick and accessible.

Companies should embrace these new tools and technologies, as they offer opportunities to achieve better results without substantial financial investment.

The integration of a chatbot in industry, as developed in this project with Dialogflow CX and Python, demonstrates the ability of these tools to simplify the interaction and processing of complex data. The technical connection was achieved through a custom webhook that enables efficient communication between the Dialogflow CX interface and the Python backend, using Flask to handle requests and ngrok to create an accessible test environment. This setup enables the chatbot to receive data, analyze it and respond in an intuitive and personalized way. With this infrastructure, companies can take full advantage of the benefits of AI in their processes, optimizing resources and improving the user experience in an agile and cost-effective way.

REFERENCES

[1] Sol Iglesias, M. (2024). ¿Qué es Stel Order? El software ideal para tu negocio. https://www.crehana.com/blog/negocios/stel-order-que-es/, accessed on Jan. 15, 2024.

[2] Galiana, P. (2022). ¿Qué es SAP y qué soluciones ofrece? Thinking for Innovation. https://www.iebschool.com/blog/que-es-para-que-sirve-sap-management/, accessed on Jan. 15, 2024.

[3] MIT. (2024). Todo lo que necesitas saber sobre Odoo. https://www.mit-mut.com/odoo, accessed on Jan. 15, 2024.

[4] Echeverria, P. (2024). ¿Qué es y cómo funciona HubSpot CRM? https://blog.growxco.com/articulos/que-es-y-como-funciona-hubspot-crm, accessed on Jan. 15, 2024.

[5] Sánchez, M. (2024). Organiza toda tu vida en una sola app: Notion. https://www.anahuac.mx/mexico/noticias/Organiza-toda-tu-vida-en-una-sola-app-Notion, accessed on Jan. 15, 2024.

[6] Tuomi, I. (2018). El impacto de la inteligencia artificial en el aprendizaje, la enseñanza y la educación. Oficina de Publicaciones de la Unión Europea. Repositorio de Publicaciones del CCI, 47. https://doi.org/10.2760/12297

[7] Ponce, P. (2010). Inteligencia artificial: con aplicaciones a la ingeniería. Retrieved April 14, 2024, from https://books.google.es/books?hl=es&lr=&id=9ENzEAAAQBAJ&oi=fnd&pg=PR1&dq=inteligencia+artificial&ots=3Ir6Tn-hbo&sig=-7qEx6n3DvIZlHj0HxE8aX-FIZY#v=onepage&q=inteligencia%20artificial&f=false, accessed on Apr. 14, 2024.

[8] Eduardo, C., De Vito, E.L. (2006). Introducción al razonamiento aproximado: Lógica difusa. Revista Americana de Medicina Respiratoria, 6(3): 126-136.

[9] Guzmán, D., Castaño, V.M. (2006). La lógica difusa en ingeniería: principios, aplicaciones y futuro. Revista de Ciencia y Tecnología, 24(2). https://revistas.ucr.ac.cr/index.php/cienciaytecnologia/article/view/2640/2591.

[10] Flores, R., Fernandez, J. (2008). Las Redes Neuronales Artificiales. https://books.google.es/books?hl=es&lr=&id=X0uLwi1Ap4QC&oi=fnd&pg=PA11&dq=redes+neuronales+artificiales&ots=gPHDmskr_l&sig=SiAjPI8Qt33BHmgPw5ncBHmQ894#v=onepage&q=redes%20neuronales%20artificiales&f=false, accessed on Apr. 14, 2024.

[11] Veintimilla, R.J., Cisneros, E.F. (2014). Vista de Redes Neuronales Artificiales (RNA) aplicadas en la prediccion de caudales para intervalos de tiempo horarios. UCUENCA. https://rte.espol.edu.ec/index.php/tecnologica/article/view/304/217.

[12] Lara, F. (2019). Fundamentos de redes neuronales artificiales. https://conceptos.sociales.unam.mx/conceptos_final/598trabajo.pdf, accessed on Apr. 14, 2024.

[13] Asanza, W.R., Olivo, B.M. (2018). Redes neuronales artificiales aplicadas al reconocimiento de patrones. Editorial UTMACH, 1(4): 5. www.utmachala.edu.ec.

[14] Acevedo, E., Serna, A., Serna, E. (2017). Desarrollo e innovación en ingeniería. http://fundacioniai.org, accessed on Apr. 14, 2024.

[15] Vasquez Lopez, J.P. (2014). Red neuronal feedforward como estimador de patrones de corrientes en el interior del puerto de manzanillo sujeto a la acción de tsunamis. PUBLICACION TECNICA, (406): 63. https://www.imt.mx/archivos/Publicaciones/PublicacionTecnica/pt406.pdf.

[16] Marczyk, A. (2004). Algoritmos genéticos y computación evolutiva. Computer Based Learning Unit, University of Leeds. https://the-geek.org/docs/algen/algen.html.

[17] Gestal, M. (2010). Introducción a los Algoritmos Genéticos. http://sabia.tic.udc.es/, accessed on Apr. 14, 2024.

[18] Zaballa, M. (2016). Técnicas de machine learning para la detección de la negación en textos clínicos en español. Archivo Digital UPM Universidad Politecnica de Madrid. https://oa.upm.es/39927/.

[19] Lepe, M. (2021). Modelos híbridos basados en lexicones y machine learning para la detección de agresividad sobre textos en idioma español.

[20] Arana, C. (2021). Redes neuronales recurrentes: Análisis de los modelos especializados en datos secuenciales. Serie Documentos de Trabajo. https://www.econstor.eu/bitstream/10419/238422/1/797.pdf.

[21] Aragon, F., Perez, I., Soto, A., Ponce, I. (2024). Chatbots: La nueva frontera de la industria moderna, 97-101.

[22] Alvarez, P. (2020). Redes recurrentes profundas para el diagnostico del TDAH en la infancia. Valladolid: Trabajo de fin de grado. Grado en ingenieria de tecnologías de telecomunicación.

[23] De la Torre, J. (2023). Redes generativas adversarias (GAN) fundamentos teóricos y aplicaciones SURVEY.

[24] Calcagni, L.R. (2020). Redes Generativas Antagónicas y sus aplicaciones. Doctoral Dissertation, Universidad Nacional de La Plata. https://cathi.uacj.mx/20.500.11961/30371.

[25] Hernández, M., Gómez, J. (2013). Aplicaciones de procesamiento de lenguaje natural. Revista Politécnica, 32: 87-96.

[26] Echeverri Torres, M.M., Manjarrés-Betancur, R. (2019). Asistente virtual académico utilizando tecnologías cognitivas de procesamiento de lenguaje natural. Revista Politécnica, 16(31): 85-96. https://doi.org/10.33571/rpolitec.v16n31a7

[27] Ye, C. (2023). Chatbot de recomendación musical basado en Dialogflow.

[28] de Gracia, P.F. (2024). Desarrollo de una Plataforma para el Análisis y Visualización de Datos de un Videojuego de Combate.