

RECONOCIMIENTO AUTOMÁTICO DEL HABLA

Desarrollo de un chatbot para asistir en la venta de quesos

Yuri Vladimir Huallpa Vargas

1 CONTENIDO

2	Introducción	2
3	Intents	2
4	Entities	3
5	Slots	3
6	Acciones	3
6.1	action_product_search	3
6.2	action_survey_submit	3
6.3	action_order_status	3
6.4	action_gpt3	4
7	Integración	4
7.1	Telegram	4
7.2	Integración con chatGPT3	5
8	Demo	5
9	Conclusiones	7

2 INTRODUCCIÓN

Los chatbots son una de las soluciones tecnológicas más innovadoras de la última década. Estos programas de inteligencia artificial permiten a los usuarios interactuar con un sistema de conversación en lugar de tener que navegar por una interfaz de usuario complicada. Los chatbots pueden ser utilizados en una amplia gama de industrias, desde el servicio al cliente hasta la banca y las finanzas.

Una de las plataformas más conocidas para el desarrollo de chatbot es RASA, con rasa se puede construir asistentes de texto y voz de manera muy sencilla, rasa ofrece una amplia gama de herramientas y características que pueden ser usadas de manera gratuita, una documentación amplia y organizada que lo hace adecuado para ser considerado como primera opción al momento de desarrollar asistentes personales.

En el trabajo final de asignatura se ha visto conveniente usar rasa por las características ya mencionadas. Además, de ya tener un conocimiento por los laboratorios desarrollados en esta asignatura.

Se ha desarrollado un chatbot para asistir en la venta de quesos cuyas funciones son: mostrar al usuario el inventario de la tienda, informar sobre el estado de un pedido y vender. Además, se ha integrado con chatGPT3 para responder preguntas relacionadas a este sector.

Esta memoria está estructurada de la siguiente forma: en la sección 3 y 4 se describirán las entidades y las intenciones, en la sección 5 y 6 se describirán los slots y las acciones que interactúan con la base de datos y chatGPT3, en la sección 7 se mostrará la integración con Telegram, en la sección 8 se mostrará las pruebas y finalmente se darán las conclusiones del trabajo realizado.

3 INTENTS

Las intenciones son la forma en que el chatbot identifica y comprende las intenciones detrás de las preguntas y solicitudes de los usuarios. Una intención representa una acción o tarea específica que el usuario quiere realizar, como hacer una pregunta sobre el tiempo o solicitar una reserva en un restaurante.

En el desarrollo de este trabajo se han creado las siguientes intenciones:

- Greet: Se activa cuando el usuario realiza un saludo.
- goodbye: Se activa cuando el usuario realiza la acción de despedirse.
- Order_status: Se activa cuando el usuario pide conocer el estado de su orden.
- Producto_stock: Intent que se activa cuando el usuario realiza una consulta sobre un producto determinado o cuando quiere comprobar la existencia de un producto en el inventario.
- Order_cheese: Intent para capturar la intención de compra.
- Help: Se activa cuando el usuario pide ayuda para saber sobre algo en específico en el contexto de los quesos.
- Deny: Intent para capturar una negación.
- Affirm: Se activa cuando el usuario afirma algo.
- user_feedback: Captura la puntuación que asigna el usuario al servicio.

4 ENTITIES

Una entidad es un concepto específico que se extrae de una conversación con un usuario. Las entidades pueden incluir información como fechas, horas, lugares, nombres de personas y productos, entre otros.

Las entidades creadas fueron los siguientes:

1. Cheese
2. Brand
3. Number
4. Email

5 SLOTS

Los slots es un mecanismo de RASA para almacenar entidades u otros valores en memoria y a medida que va transcurriendo la conversación pueden ser utilizados.

Se ha definido los siguientes slots:

1. Cheese
2. Brand
3. Email
4. Feedback
5. survey_complete
6. is_ok

6 ACCIONES

6.1 ACTION_PRODUCT_SEARCH

Esta acción realiza una búsqueda en la base de datos en función de la entidad “cheese” y “brand”, devuelve los resultados que coincidan con alguna de estas entidades.

6.2 ACTION_SURVEY_SUBMIT

Esta acción se activa solo una vez cuando el bot realiza un flujo de conversación de manera satisfactoria solicitando al usuario que se le puntue.

6.3 ACTION_ORDER_STATUS

Acción para consultar el estado de una orden en función del correo del usuario.

6.4 ACTION_GPT3

Cuando un usuario pide ayuda o consulta otras intenciones que no están contempladas en las intenciones se llama a la api de ChatGPT3 para buscar una respuesta. Esta solo devuelve la respuesta si la pregunta que se ha realizado esta dentro del contexto de quesos.

7 INTEGRACIÓN

En este apartado se mostrará las configuraciones realizadas para integrar Telegram y chatGPT3 con nuestro asistente.

7.1 TELEGRAM

Para realizar una integración con Telegram primero se necesita crear un bot mediante BotFather de telegram el cual nos devolverá las credenciales de nuestro bot. Ver Ilustración 1. Posteriormente se inicia ngrok, con el comando `ngrok http 5005` el cual nos devolverá una url.

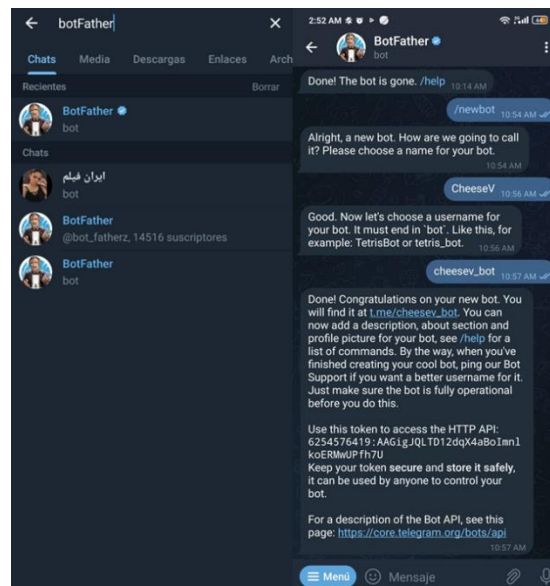


Ilustración 1. Nuevo bot en Telegram

Con los dos pasos realizados se configura el archivo `credentials.yml` de rasa.

```
telegram:
  access_token: "6254576419:AAGigJQLTD12dqX4aBoImnlkoERMw"
  verify: "cheesev_bot"
  webhook_url: "https://ede9-84-125-125-184.eu.ngrok.io/webhooks/telegram/webhook"
  # webhook_url: "https://rahyuribot.herokuapp.com/webhooks/rest/webhook/telegram/webhook"
# This entry is needed if you are using Rasa Enterprise. The entry represents credentials
# for the Rasa Enterprise "channel", i.e. Talk to your bot and Share with guest testers.
rasa:
  url: "http://localhost:5002/api"
```

Ilustración 2. Configuración para telegram

Finalmente, se inicializa el bot mediante el comando **rasa run** y **rasa run actions**.

7.2 INTEGRACIÓN CON CHATGPT3

La integración de chatGPT3 se encuentra bien detallado en la página oficial de OpenAI el cual puede consultarse mediante el siguiente enlace: <https://platform.openai.com/docs/api-reference?lang=python>

Se muestra la configuración realizada para realizar una solicitud hacia chatGPT3.

```
class ActionGPT3(Action):
    def name(self) -> str:
        return "action_gpt3"

    def run(self, dispatcher, tracker, domain):
        request = " ".join(tracker.latest_message['text'].split())
        print(request)
        response = openai.Completion.create(
            engine="text-davinci-003",
            prompt="{}. Responde la pregunta, todo en español".format(request),
            max_tokens=1024,
            n=1,
            stop=None,
            temperature=0.5,
        ).choices[0].text
        dispatcher.utter_message(text=response)
        return []
```

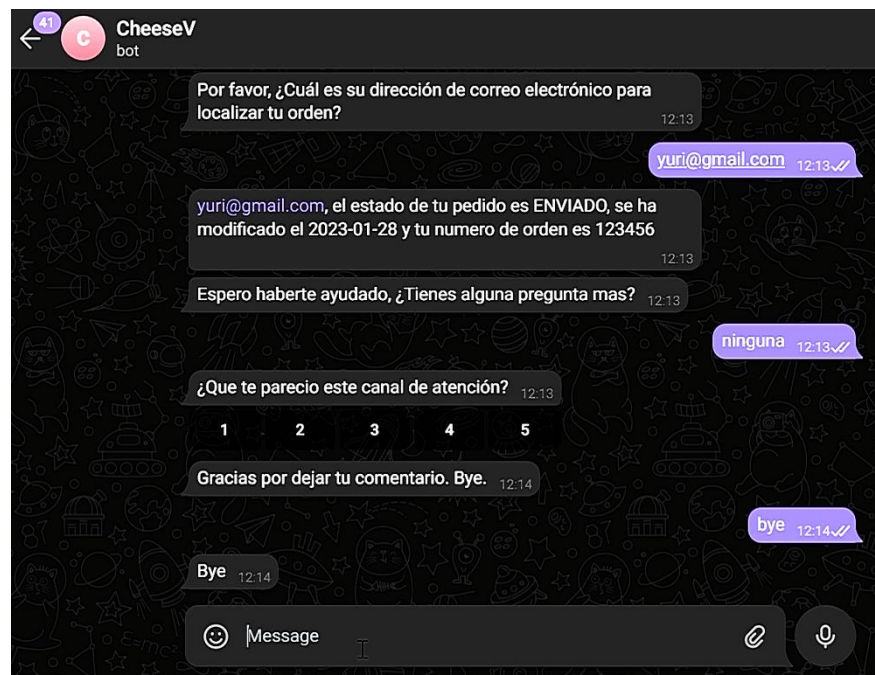
Ilustración 3. ChatGPT3 action.

8 DEMO

Con las configuraciones realizadas es momento de realizar las pruebas, se ha realizado una demo y el video se puede ver en el siguiente enlace:

https://drive.google.com/file/d/1bD1H1O2_jlhq97TnAqvXWVhRLFZwt5A5/view?usp=sharing

Se muestra algunas capturas de las conversaciones realizadas.



En este ejemplo se ha activado el flujo para la intención estado de una orden. Primero, el sistema pide al usuario un correo electrónico para realizar la búsqueda. Segundo, el usuario ingresa un correo que no existe en la base de datos y el sistema indica que no existe este correo. Tercero, el sistema pide nuevamente un correo y se ingresa uno correcto. Cuarto, el sistema indica cual es el estado de su orden y pregunta si el servicio que presto el bot le a ayudado. Finalmente, el usuario califica la atención y el bot se despide.

9 CONCLUSIONES

La realización de este trabajo no ha ayudado a reforzar los conceptos aprendidos en clases respecto a interfaces conversacionales y se ha aprendido una nueva herramienta que nos será útil en el corto plazo. Además, nos ha permitido conocer cuales son los requisitos necesarios antes de comenzar a construir un asistente.