



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA

CURSO ACADÉMICO 2018/2019

TRABAJO FIN DE GRADO

METACHATBOT: CHATBOT PARA CREAR CHATBOTS

Autor: Antonny Gerald Chumpitaz Rios

Director: Francisco Domínguez Mateos

AGRADECIMIENTOS

Después de unos intensos meses de trabajo por fin ha llegado el día en el que veo los frutos de mi curiosidad por la programación avanzada con metodologías que no había visto antes, y la investigación realizada a lo largo de este trabajo.

He podido aprender nuevas tecnologías, dar soluciones a los problemas que surgían a medida que realizaba el proyecto, aunque no siempre de la forma que se esperaba ni la más óptima.

Primero, me gustaría agradecer a mi tutor por los consejos que me ha dado y la paciencia que tuvo durante estos meses ya que no siempre he podido cumplir los hitos que nos planteamos por falta de tiempo al estar trabajando, por buscar tiempo para poder resolver mis dudas y plantearme nuevos desafíos para que pueda conseguir realizarlos y así mejorar el proyecto que teníamos en mano.

Agradecer a mi familia por el apoyo que me dieron siempre, aconsejándome a no desistir cuando me encontraba bloqueado, por las conversaciones ajenas al trabajo que me ayudaron a entender más cosas sobre mi futuro.

Muchas gracias...

RESUMEN

El desarrollo de este proyecto está centrado en los *Chatbots*. Estos son sistemas inteligentes que permiten interaccionar, mediante voz o texto, con usuarios, simulando el comportamiento humano.

Los *Chatbots* tienen muchas aplicaciones ya que permiten una interacción persona-máquina más sencilla y fluida. Por ejemplo, se encuentran en sistemas de atención al usuario, en webs, sistemas de ayuda telefónica, etc.

En este caso, se ha definido un formato de representación de *Chatbots* y se ha desarrollado un *Motor de Chatbots*, de tal manera que se puede procesar cualquier *Chatbot* que se desee, entrenando y generando su modelo. La técnica utilizada está basada en aprendizaje supervisado con *redes neuronales*, intenciones, sacos de palabras y acciones (estos conceptos se detallarán en la memoria). Para probar dicho motor, se creó un sencillo *Chatbot* de gestión de Lista de la Compra.

Con lo expuesto anteriormente, se ha podido crear un tipo especial de *Chatbot*, un *Metachatbot*, que permite crear *Chatbots* interactuando con él mediante texto o voz y utilizar el *Motor de ChatBots* que entrena y genera modelos de los *Chatbots*.

Finalmente se ha añadido la posibilidad de ir mejorando de forma continua e interactiva el comportamiento de los *Chatbots*, modificando el modelo *de los Chatbots* utilizando el *Motor de Chatbots* y añadiendo un nuevo *Chatbot de entrenamiento y mejora* llamado *Solve Error* que permite detectar y solucionar problemas, mejorando así la interacción.

A lo largo de este proyecto, se ha podido adquirir conocimientos del uso y aplicación de nuevas tecnologías como la programación avanzada en Python, paquetes de código para *Inteligencia Artificial* tales como Keras, TensorFlow, etc., nuevas nociones de programación modular, el poder gestionar mejor el tiempo para desarrollar y mejorar el proyecto, hacer llamadas a servicios para realizar la interacción por voz, entre otros.

Palabras claves : Chatbot, sistemas inteligentes, Motor de Chatbots, redes neuronales, Metachatbot, sistema de mejora y entrenamiento.

ÍNDICE

ÍNDICE DE TABLAS	III
ÍNDICE DE ILUSTRACIONES	IV
CAPÍTULO 1: INTRODUCCIÓN.....	1
1.1 INTRODUCCIÓN.....	1
1.2 ESTRUCTURA.....	7
1.3 OBJETIVOS	8
CAPÍTULO 2: ESTADO DEL ARTE	9
2.1 PANORÁMICA	9
2.2 COMPARATIVA	13
2.3 CRÍTICA DEL ESTADO DEL ARTE.....	16
2.4 PROPUESTA.....	17
2.5 RESUMEN DEL CAPÍTULO.....	17
CAPÍTULO 3: OBJETIVOS	19
CAPÍTULO 4: IMPLEMENTACIÓN DEL SISTEMA	21
4.1 ARQUITECTURA	21
4.1.1 JSON	21
4.1.2 Diagrama de clases del sistema.....	23
4.1.3 Flujo de ejecución	24
4.2 MOTOR DE CHATBOTS.....	25
4.2.1 Análisis.....	25
4.2.2 Diseño	26
4.2.3 Pruebas	28
4.3 METACHATBOT	28
4.3.1 Análisis.....	28
4.3.2 Diseño	30
4.3.3 Pruebas	31
4.4 LISTA DE LA COMPRA	34
4.4.1 Análisis.....	34
4.4.2 Diseño	35
4.4.3 Pruebas	35
4.5 SISTEMA DE MEJORA Y ENTRENAMIENTO	37

4.5.1 Análisis.....	37
4.5.2 Diseño	38
4.5.3 Pruebas	39
4.6 RESUMEN DEL CAPÍTULO.....	41
CAPÍTULO 5: EVALUACIÓN CUALITATIVA	43
CAPÍTULO 6: RECURSOS EMPLEADOS.....	45
6.1 MATERIALES	45
6.2 TABLAS DE COSTES.....	46
CAPÍTULO 7: TRABAJO FUTURO	49
CAPÍTULO 8: CONCLUSIONES	51
GLOSARIO	53
BIBLIOGRAFÍA	57

ÍNDICE DE TABLAS

Tabla 1: Comparación de entornos de creación de Chatbots y Chatbots.....	15
Tabla 2: Comparación entre Metachatbot, otros entornos y Chatbots.....	43
Tabla 3: Tabla de gastos monetarios.....	46
Tabla 4: Tabla de costes temporales por mes.	47
Tabla 5: Costes finales del proyecto.	47

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Estructura del Chatbot "Lista Compra"	22
Ilustración 2: Interacción de clases de un Chatbot.....	23
Ilustración 3: Flujo de estados general de un Chatbot.	24
Ilustración 4: Entrenamiento y creación del modelo de un Chatbot.	27
Ilustración 5: Modelo generado por el motor de Chatbots.	28
Ilustración 6: Acciones del Metachatbot.....	31
Ilustración 7: Prueba Metachatbot - Crear Chatbot.	32
Ilustración 8: Prueba Metachatbot - Crear elementos del Chatbot.	32
Ilustración 9: Prueba Metachatbot - Generar Chatbot.	33
Ilustración 10: Prueba Metachatbot - Ficheros creados automáticamente.....	33
Ilustración 11: Prueba Lista Compra - Insertar y Mostrar la cesta.	36
Ilustración 12: Prueba Lista Compra - Cancelar acción, guardar error y terminar ejecución.	36
Ilustración 13: Acciones del Sistema de mejora y entrenamiento.	39
Ilustración 14: Prueba Sist. Mejora y Entrenamiento - Selección de Chatbot a mejorar.	39
Ilustración 15: Prueba Sist. Mejora y Entrenamiento - Procesar el error.	40
Ilustración 16: Prueba Sist. Mejora y Entrenamiento – Resultado de procesar el error (1)	40
Ilustración 17: Prueba Sist. Mejora y Entrenamiento – Resultado de procesar el error (2)	40

CAPÍTULO 1: INTRODUCCIÓN

Este capítulo se dividirá en distintas secciones en las cuales se hablará de conceptos necesarios para entender el tema, estructura y objetivos de este proyecto.

En la primera sección de este capítulo se abordarán distintos conceptos que ayudarán a comprender el tema principal que tiene este trabajo, mencionando los tipos de sistemas inteligentes, características, la finalidad que tienen en distintos sectores de interés, en dónde se encuentran, tecnologías, puntos fuertes y débiles.

A su vez, en otra sección se hablará de la estructura que presenta toda la memoria, dando una visión general de los temas a tratar en los distintos capítulos, y secciones dentro de los mismos.

Por último, en la sección final de este capítulo se mencionan, sin entrar al detalle, los distintos objetivos planteados en este trabajo dando una visión de lo que se pretende conseguir con el desarrollo del mismo.

1.1 INTRODUCCIÓN

Con el paso del tiempo han llegado a existir numerosos sistemas inteligentes implementados de distintas formas y con diversas funcionalidades.

Con ayuda de tecnologías basadas en Inteligencia Artificial, estos sistemas son capaces de realizar numerosas actividades logrando una sustitución parcial, y en algunos casos completa, de las personas.

Estos sistemas se encuentran en aeropuertos ayudando a los viajeros dando servicios de atención al cliente, en hospitales atendiendo a llamadas de reservas de citas o ayuda, en sistemas del hogar como es el caso de Alexa de Amazon, en los coches inteligentes, en las tomas de decisiones, en aplicaciones móviles como asistentes personales, para realizar pedidos, consultar información, reserva, etc.

Como se puede apreciar de lo explicado anteriormente, un sistema inteligente tiene un amplio campo de desarrollo.

Según la definición de (Vasant 2015) en su libro “Artificial Intelligence Techniques and Algorithms” un sistema inteligente es:

“Intelligent System (IS) can be defined as the system that incorporates intelligence into applications being handled by machines. Intelligent systems perform search and

optimization along with learning capabilities. Different types of machine learning such as supervised, unsupervised and reinforcement learning can be modeled in designing intelligent systems. Intelligent systems also perform complex automated tasks which are not possible by traditional computing paradigm. Various diagnostic, robotics and engineering systems are results of intelligent procedures implemented in Intelligent System Design."

Dentro de estos sistemas inteligentes, se encuentran los sistemas conversacionales o agentes virtuales y se hablará en específico de los Chatbots, que es el tema principal de este trabajo.

La definición de Chatbot por parte de (Rouse 2018) menciona que:

"Un chatbot (a veces referido como un chatterbot) es un programa de computadora que intenta simular la conversación o "parloteo" de un ser humano a través de texto o interacciones de voz. Un usuario puede hacer una pregunta a un robot de chat o introducir un comando, y el robot de chat responde o realiza la acción solicitada."

Por lo cual, se entiende que un Chatbot es un sistema inteligente que interacciona con los humanos simulando una conversación.

En este trabajo, el Chatbot que se construirá no estará enfocado a ser un Chatbot convencional para hacer un pedido de una compra o para consultar información, sino que será un Chatbot que se comporte como un entorno de creación de otros Chatbots, un Metachatbot. Es decir, se comportará como un Chatbot pero enfocado a realizar tareas de un entorno de programación de Chatbots. En los siguientes capítulos se detalla cómo se ha desarrollado el Metachatbot.

A continuación se hablará de distintos temas sobre los Chatbots que son de interés para este trabajo, por ejemplo sus características, la finalidad que tienen en distintos sectores de interés, en dónde se encuentran, tecnologías, puntos fuertes y débiles.

CARACTERÍSTICAS DE UN CHATBOT

A continuación se presentan algunas características generales que pueden tener los Chatbots apoyándose en lo mencionado por Contieri en el documento que publicó en **planetachatbot.com** (Contieri 2017).

- Atención híbrida: La atención a los usuarios puede apoyarse en estos Chatbots y aquellas peticiones de usuarios complicadas son atendidas por humanos.
- Registro y revisión: Capacidad para guardar y aprender de las conversaciones pasadas.

- Reinicio de conversaciones: Permite a los Chatbots a poder reiniciar su flujo tras un tiempo de espera, restableciendo el contexto de la conversación.
- Aprendizaje supervisado: Las personas indican mejorar los Chatbots resolviendo los errores que hayan tenido.
- Analítica: Tener la opción de analizar su rendimiento para futuras mejoras.
- Etiquetas: El Chatbot tendrá que permitir el uso de indicadores de los diferentes temas que puede tratar los usuarios, estas etiquetas será puestas por humanos.
- Comunicación: Simulan el comportamiento cognitivo de los humanos.

Los Chatbots pueden presentar otras características dependiendo de tipo que sean y los objetivos que necesitan realizar. Por ejemplo la posibilidad de interactuar por voz, comunicarse con otros Chatbots, puede utilizar Inteligencia Artificial o prescindir de ella, etc.

FINALIDAD DE UN CHATBOT

Se entiende como finalidad del Chatbot a los objetivos que pretende conseguir utilizando distintos medios, herramientas y tecnologías.

A continuación se presentarán algunos objetivos que pueden tener los Chatbots:

- Mejorar el marketing: Los Chatbots tienen el objetivo de informar a los clientes o los futuros clientes de los productos de la empresa, ofertas, descuentos, etc.
- Captación de leads: En este sector, los Chatbots son capaces de contactar con personas por distintos medios para ofrecerles los servicios de la empresa con el fin de convencerlos de adquirir dichos servicios.
- Social: Tienen como función el mantener el contacto con distintas personas, dar información, etc. Se suelen usar para hacer tareas similares y de gran volumen, tales como responder a preguntas repetitivas, realizar comentarios diarios a la misma hora, etc.
- Mejorar el comercio electrónico: Se asemeja a un asistente de compras que acompaña a los clientes durante el proceso de compra. Un ejemplo es el Chatbot de H&M, que ayuda al usuario a buscar lo que quiere con sólo pocas preguntas ahorrando así, tiempo para el usuario.

- Soporte: Estos Chatbots son una buena opción para las empresas que busquen dar un servicio de ayuda más rápido, mejorando la experiencia del usuario, reduciendo la cola de peticiones, tiempo de espera, etc.
- Atención al usuario: Muchos de estos Chatbots ya están implementados y funcionando en el sector médico, comercio online, hostelería, investigación, etc. Ayudando al usuario a conseguir la información necesario, realizar pedidos, reservas, etc.
- Asistentes: Estos Chatbots se pueden encontrar tanto en dispositivos electrónicos como móviles, ordenadores, neveras, sistemas de hogares, etc. Algunos ejemplos más destacados son de los asistentes que se encuentran en los móviles y ordenadores como SIRI, CORTANA, siendo estos asistentes personales; o los que están en los sistemas del hogar como ALEXA.
- Apoyo en el trabajo: En este sector, los Chatbots son más específicos para realizar tareas concretas que ayuden a los trabajadores interaccionando con otros sistemas informáticos.

SISTEMAS DE EJECUCIÓN DE CHATBOTS

Existen muchos sistemas donde se encuentran los Chatbots, pueden ser aplicaciones enfocadas a una interacción por texto, gráficos, imágenes, voz, etc.

Según Maximiliano Latorre en una publicación en la web **elobservador.com**, los Chatbots más usados suelen estar en sistemas de texto, por ejemplo aplicaciones de mensajería (Latorre 2016).

Pero también se pueden encontrar en sistemas de audio, usando tecnologías de reconocimiento de voz y *text to speech*, que son capaces de transformar el lenguaje humano a cadenas de texto comprensibles, en este caso, para el Chatbot. Posteriormente las respuestas del Chatbot son producidas por estos entornos de voz para que lleguen en formato de audio a las personas (botanalytics 2018).

Otros sistemas en los que se pueden encontrar son los sistemas avanzados donde los Chatbots se comunican con otros sistemas inteligentes. Por ejemplo, puede haber un Chatbot cuya funcionalidad es dar algún recurso y muchos otros Chatbots en vez de realizar el trabajo de buscar y dar un formato a ese recurso, directamente utilizan al Chatbot que les proporciona lo que buscan.

TIPOS DE TECNOLOGÍAS DE LOS CHATBOT

Aunque muchos Chatbots se comuniquen con las personas, no todos presentan tecnologías de Inteligencia Artificial para realizar esta tarea.

A continuación se presentan los tipos de Chatbots en base a su tecnología:

- Con reglas o comandos: Utilizan palabras claves que son reconocidos por el Chatbot y su heurísticas. Estas reglas tienen que ser programadas a mano.

Su funcionamiento se basa en buscar entre su banco de palabras coincidencias con la petición del usuario y el contexto, y dar una respuesta que tiene definida. Cuando el usuario inicia este Chatbot, la contestación de éste es mostrando una serie de opciones para que el usuario elija. Las peticiones que puede realizar el usuario son limitadas al “conocimiento” del Chatbot.

No pueden generar nuevo texto, sólo pueden dar respuesta dentro de un contexto y limitadas a las respuestas previamente definidas, sin poder realizar funciones que no estén programadas.

- Con Inteligencia Artificial: Son capaces de reconocer el contexto de la conversación, recordar conversaciones antiguas, responder a preguntas ambiguas, pueden crear textos desde cero con el uso del procesamiento del lenguaje natural, aprenden de todos los casos que se ha dado en el pasado para mejorar sus respuestas.

Para poder realizar estas tareas, el Chatbot debe utilizar alguna técnica que ofrece el Machine Learning que es una rama de la Inteligencia Artificial que proporciona distintas técnicas para que las máquinas puedan aprender.

Según la definición de la información ofrecida por Wikipedia con el tema de “Aprendizaje automático” (Wikipedia 2018), se define Machine Learning como:

“Un proceso de inducción del conocimiento. En muchas ocasiones el campo de actuación del aprendizaje automático se solapa con el de la estadística computacional, ya que las dos disciplinas se basan en el análisis de datos. Sin embargo, el aprendizaje automático también se centra en el estudio de la complejidad computacional de los problemas. Muchos problemas son de clase NP-hard, por lo que gran parte de la investigación realizada en aprendizaje automático está enfocada al diseño de soluciones factibles a esos problemas. El aprendizaje automático puede ser visto como un intento de automatizar algunas partes del método científico mediante métodos matemáticos.”

Dentro de los algoritmos de aprendizaje automático que ofrece el Machine Learning se pueden encontrar distintos tipos.

La siguiente información está apoyada en el artículo “Types of Machine Learning Algorithms You Should Know” de *David Fumo* (Fumo 2017) :

- Supervisada: esta técnica utiliza datos etiquetados. Es decir, cuando se está entrenando el modelo, una persona etiqueta los datos para reconocer al conjunto de datos al que pertenecen. Después de un periodo de entrenamiento, genera el modelo que es capaz de clasificar nuevas entradas sin etiquetarse que no han sido introducidas en la etapa de entrenamiento llegando a generalizar nuevos datos.
- No supervisada: cuando se utiliza esta técnica no se etiquetan los datos. A la hora de realizar el entrenamiento este algoritmo genera una serie de listas con valores que identifican cada dato, agrupa los que más se parezcan y genera el modelo para reconocer datos semejantes a los grupos del entrenamiento.

Cuando se introducen nuevos datos se transforma ese dato en una lista de bits característicos, para que luego se busque la semejanza con los grupos establecidos del entrenamiento. Tiene objetivo aprender e inducir los patrones de los datos usando técnicas como el algoritmo de agrupamiento (clustering).

- Semisupervisado: este algoritmo emplea tanto datos etiquetados como datos sin etiquetar. Utiliza en mayor medida datos sin etiquetar para realizar el entrenamiento y generar un modelo que se adecue a los datos etiquetados.
- Aprendizaje por refuerzo: esta técnica es una de las más interesante ya que no necesita numerosos datos de entrenamiento. Aplica una técnica de prueba y error. El algoritmo da al sistema un “premio” si acierta y en caso contrario se le “castiga”, el sistema siempre buscará recibir el “premio” por lo cual después de numerosos intentos fallidos aprende a conseguir su objetivo, el premio. Este premio debe de estar relacionado con los resultados correctos.

Por lo explicado anteriormente, se entiende que el uso de ambas tecnologías presentan desventajas y ventajas. Los Chatbots por comandos dan una respuesta más confiable y comprensibles, pero no responden a preguntas fuera de su conocimiento de respuestas.

Por otro lado, los Chatbots basados en las técnicas de Machine Learning necesitan una gran cantidad de datos o numerosos entrenamientos para su aprendizaje y tienen muchas dificultades para formar respuestas, pero son capaces de dar una respuesta a preguntas que no estén en su base de conocimiento.

PUNTOS FUERTES

Los Chatbots pueden utilizarse en numerosas aplicaciones o entornos teniendo ciertas ventajas:

- Pueden entender lo que dicen las personas tanto por texto escrito o hablado e interpretar su significado, aunque no siempre es preciso.
- Pueden buscar información basándose en los conocimientos obtenidos previamente por experiencias y ofrecerlos a las personas.
- Son un apoyo importante para las aplicaciones modernas de móviles inteligentes.
- Facilitan las tareas monótonas y repetitivas.

PUNTOS DÉBILES

Aunque los Chatbots estén en constante desarrollo presentan ciertos puntos débiles:

- Aún no son precisos.
- Necesitan muchos datos para poder ser un poco estables.
- Necesitan numerosas actualizaciones.
- Programación complicada.

Este proyecto se basará en un Chatbot que utiliza la tecnología de Inteligencia Artificial apoyándose en la técnica de aprendizaje supervisado para crear otros Chatbots. Cuya estructura, hecha en JSON, estará basada en intenciones, que son las ideas que el usuario quiere transmitir en cada petición que realice al Chatbot. También la estructura tendrá unos patrones, que son las sentencias que el usuario escribe para transmitir sus intenciones. El Chatbot al reconocer estos patrones devolverá, unas respuestas que es otro elemento de su estructura. Por último, si a esa intención en la que se encuentre el Chatbot está asociada una acción, otro elemento de la estructura, procederá a ejecutarla.

Se irán detallando los conceptos mencionados previamente así como las herramientas usadas.

1.2 ESTRUCTURA

El contenido se desarrollará en distintos apartados, empezando por el estado del arte donde se explicará la situación actual sobre los Chatbots presentando algunas críticas y algunas propuestas para su mejora. Seguida de las hipótesis del proyecto, es decir, los objetivos que el proyecto pretende conseguir para mejorar algunos puntos respecto del estado actual de los Chatbots.

Después de estos apartados de detallará las soluciones que se han desarrollado para lograr los objetivos, detallando por separado cada solución con sus respectivas imágenes, herramientas, diseños, etc.

Además, se planteará las conclusiones del proyecto resumiendo lo que se ha logrado con el desarrollo del mismo, para luego terminar mencionando las posibles mejoras en el apartado de trabajo futuro.

1.3 OBJETIVOS

Este proyecto tiene como objetivo conseguir un sistema inteligente que cree Chatbots, un Metachatbot.

El Metachatbot será capaz de reconocer una estructura, creada para este trabajo, para guardar los elementos del Chatbot, y tendrá medios de interacción por voz o por texto, lo que hará más versátil al sistema. Este Metachatbot podrá generar Chatbots con las mismas posibilidades de interacciones.

Se creará un motor de Chatbots para procesar las estructuras de los Chatbots y servirá para realizar los trabajos de entrenamiento y generar el modelo del Chatbot que son necesarios para el reconocimiento de peticiones. A su vez, el usuario podrá interrumpir el flujo del motor para terminar ejecuciones o indicar algún fallo.

Para ofrecer al usuario una herramienta de mejora y entrenamiento automático, se creará otro Chatbot que permita interactuar de forma escrita o por voz para editar la estructura interna del Chatbot creado.

CAPÍTULO 2: ESTADO DEL ARTE

En este capítulo se hablará más en detalle sobre las aplicaciones disponibles para la creación de Chatbots.

La primera sección habla del panorama actual de los Chatbots y de algunos programas que los crean mostrando las similitudes entre ellas. Luego se concretará, en otra sección, las características que presentan estos entornos de programación de Chatbots.

Seguida de esta sección, se presentará una serie de críticas con respecto al estado en el que se encuentran estos entornos de programación de Chatbots y unas propuestas generales de las críticas.

2.1 PANORÁMICA

En la actualidad los Chatbots se encuentran en numerosos sectores prestando servicios de diferentes formas.

Muchos de ellos implementados de una forma específica, teniendo como objetivo realizar tareas que no implican la evolución o mejora a grandes rasgos de su arquitectura, presentando una lógica cerrada. Por ejemplo los Chatbots que dan opciones de respuestas para que el usuario seleccione alguna.

Otros, en cambio, realizan tareas que reclaman la flexibilidad del Chatbot para generalizar los resultados. Esta flexibilidad se logra gracias a tecnologías basadas en Inteligencia Artificial.

El uso de Inteligencia Artificial permite al Chatbot mantener una conversación natural con una persona, Chatbot o sistema. Esto es gracias a que ha sido diseñado, programado y entrenado para interpretar las intenciones de los usuarios, entender sus respuestas y en base a ello decidir qué responder o qué acción ejecutar. El poder entender y procesar un diálogo es debido a técnicas NLP (Natural Language Processing, o en español Procesamiento del Lenguaje Natural) y redes neuronales.

Muchas aplicaciones que permiten crear Chatbots están dirigidos a que éstos se utilicen en aplicaciones de mensajería, lo que incrementa la posibilidad de interacción con más usuarios.

Otros Chatbots están dirigidos a entornos en el que utilizan interfaces conversacionales por voz, algunos de los más conocidos son APPLE HOMEPOD de **Apple**, GOOGLE HOME de **Google**, ALEXA de **Amazon** que liberó su código en el 2015 para que se pueda añadir nuevas características a su sistema, pudiendo así, mejorarlo a la necesidad

del usuario. Estos tipos de Chatbots utilizan redes neuronales complejas para poder generalizar el contexto de las peticiones de los usuarios.

SISTEMAS DE EJECUCIÓN DE CHATBOTS

La mayor parte de empresas han mejorado sus plataformas de mensajerías para ofrecer entornos de trabajo sobre los cuales se pueden ejecutar los Chatbots. Algunas de estas aplicaciones, además, han creado Chatbots Stores y sistemas avanzados de búsqueda y valoración de Chatbot, con el objetivo de que los desarrolladores sean quienes generen las utilidades que potencien el uso de dichas aplicaciones.

Según distintas fuentes como (pulsosocial 2017) o (Planeta Chatbot 2017), las principales aplicaciones de mensajería a nivel mundial en las que se encuentran los Chatbots son:

- Messenger
- Telegram
- Slack
- Skype
- Kik
- Wechat

Aunque la plataforma de mensajería es la más conocida, no es la única en donde se pueden encontrar los Chatbots. Como se han explicado en anteriores apartados y capítulos, los Chatbots pueden estar dirigidos a diferentes áreas.

Existen otras plataformas en las que los Chatbots pueden cumplir con su rol asignado. Por ejemplo, se encuentran en la propia web de una tienda online, dando soporte para solicitar alguna cita médica, en sistemas inteligentes en nuestros móviles y dispositivos del hogar, dando apoyo en sistemas implantados en centros de salud, investigación, etc.

ENTORNOS DE PROGRAMACIÓN DE CHATBOTS

Los entornos de programación de Chatbots tienen una serie de características en común. Entre las cuales podemos encontrar una interfaz web como medio de interacción con el usuario. En esta interfaz, el usuario es capaz de ver el estado de su Chatbot, algunos de estos entornos ofrecen un espacio en la misma web para probar el funcionamiento del Chatbot, otros en cambio carecen de esta herramienta o necesitan de otros programas para realizar las pruebas.

Por otro lado, tras realizar una búsqueda, no se ha encontrado algún entorno, de fácil accesibilidad para los usuarios, que permita crear los Chatbots con una interacción distinta a la del teclado o ratón. Otra característica es que algunos permiten programar directamente las acciones que el Chatbot tiene que ejecutar.

Estos Chatbots resultantes de estos entornos pueden, en su mayoría dependiendo de la plataforma en que se ejecuten, reconocer y responder por audio en distintas plataformas, ya sea en aplicaciones de mensajería, sistemas del hogar o trabajo, asistentes inteligentes de móviles, etc.

Algunos ejemplos más destacados de empresas que proporcionan entornos para crear Chatbots son:

- IBM: su tecnología de computación cognitiva ofrece un conjunto de servicios en modo SaaS que se centran en el procesamiento de lenguaje, como es Watson Conversation Service, permitiendo definir intenciones y entidades.

Algunas de las características que ofrece IBM Watson Conversation al crear Chatbots es la interfaz web para el diseño del Chatbot, una zona de pruebas, modificar el Chatbot, su interacción es con el uso del teclado y ratón, no ofrece la capacidad de programación, puede entrenar automáticamente el modelo del Chatbot.

Por otro lado, los Chatbots que se creen con este sistema poseen una interacción por voz, se pueden usar en aplicaciones de mensajería, puede comunicarse con otros sistemas para obtener información.

- Microsoft: dentro de su plataforma de cloud Azure ofrece una tecnología pensada para el desarrollo de Bots y la tecnología Language Understanding Intelligence Service a la que han llamado LUIS, como una serie de servicios alrededor del procesamiento de lenguaje natural como es el análisis lingüístico y ofrece un conjunto de herramientas que permite entrenar a la plataforma en modelos de conversación.

Una aplicación de la tecnología de Chatbots desarrollada por la propia empresa Microsoft la encontramos en el proyecto CALENDAR.HELP que ofrece una utilidad para la gestión de agenda y reuniones.

Como plataforma para crear Chatbots, Microsoft Azure Bot Services ofrece servicios similares a la plataforma y a los Chatbots resultantes de IBM Watson Conversation, diferenciándose en que permite una programación compleja para desarrolladores.

- Google: ofrece la API Natural Language sobre su plataforma en la nube y es utilizada para descubrir la estructura y el significado del texto usando modelos de aprendizaje automático. Puede usarse para conocer las opiniones sobre un producto en las redes sociales o analizar las intenciones de los clientes a partir de las conversaciones de un centro de llamadas o una aplicación de mensajería.

Por otro lado, Google cuenta con API.ai que compró en 2016 convirtiéndose en una de las principales tecnologías para el desarrollo de Chatbots por parte de

empresas, gracias a su capacidad de procesamiento del lenguaje natural; también ha creado CHATBASE desde la cual se ofrece un servicio de analítica con el cual los desarrolladores pueden analizar los resultados y aprender de las incidencias o errores que puedan aparecer en sus Chatbots.

En cuando a la creación de Chatbots con API.ai, no se diferencia mucho de los anteriores entornos, pero no está enfocado a poder realizar una programación compleja sino a una más sencilla, utilizan intenciones para crear el modelo del Chatbot. Los Chatbots resultantes también poseen interacción por voz y se pueden ejecutar en diversas plataformas.

- Amazon: ha desarrollado la tecnología Lex como un servicio para crear interfaces de conversación en cualquier aplicación con voz y texto, con funcionalidades de aprendizaje profundo avanzadas del reconocimiento automático del habla.

Gracias a esto se pueden crear aplicaciones con interacciones de conversaciones realistas. A través de este servicio, se encuentra ALEXA a disposición de cualquier desarrollador, para que pueda crear Chatbots conversacionales con un lenguaje natural sofisticado de forma rápida y fácil.

Amazon con la plataforma Lex permite ir un poco más lejos, permitiendo la libertad total de la programación de las acciones de los Chatbots resultantes, lo que permite a los desarrolladores crear cualquier tipo de acciones, aunque enfocados a los productos de Amazon y a sus arquitecturas.

- Facebook: además tener las plataformas de Messenger y Whatsapp en las cuales se encuentran el mayor número de Chatbots, también compró en 2015 Wit.ai, un servicio por medio del cual los desarrolladores podían crear Chatbots autónomos y que aprende con el tiempo. Para ello se ofrecía una interfaz especialmente diseñada para simplificar el proceso de creación del bots, y varios SDK y API. Pero se ha anunciado que el servicio únicamente se ofrecerá para el procesamiento del lenguaje natural.

Este entorno presenta las mismas características que el anterior, y en cuando a sus Chatbots están más enfocados a la propia plataforma de mensajería de Facebook, Facebook Messenger.

Otras aplicaciones alternativas son:

- Chatfuel: trabaja sobre tecnología NLP. Se necesita establecer las reglas a través de su interfaz para que el Chatbot vaya reconociendo las frases y pueda reconocer las variaciones de los usuarios al utilizarlas.

En cuando a la plataforma no permite a los desarrolladores realizar programar las acciones de los Chatbots, por lo demás es muy similar a los anteriores ejemplos.

Los Chatbots creados están enfocados a interacción por texto ya que la plataforma de uso es en aplicaciones de mensajería, y pueden realizar múltiples mensajes.

- Pandorabots: es una empresa de Inteligencia Artificial que presta servicios web para construir y desarrollar Chatbots, siendo una de las más antiguas y grandes del mundo. Es compatible con el nuevo AIML 2.0 como lenguaje de marcado de contenido de conocimiento, y muchas de sus fortalezas provienen de la potencia y la sencillez de AIML y un motor de análisis y traducción basado en Lisp.

Es una plataforma que presenta las mismas características al anterior. Pero los Chatbots resultantes pueden responder por medio de audio aunque las peticiones de los usuarios son por texto, pero no pueden realizar múltiples mensajes como Chatfuel.

- MobileMonkey: en el mes de junio del 2018, la empresa de Chattertype fue adquirida por MobileMonkey que pretende crecer dentro de la plataforma de Facebook Messenger ofreciendo su herramienta para crear Chatbots. Estos tienen una función llamada "Chat Blast", que es la capacidad de enviar mensajes a múltiples usuarios, podría ser a todos sus usuarios o un segmento en particular.

Esta característica es, sin duda, valiosa para los vendedores aunque no es la única herramienta con esta característica. Por ejemplo, Chatfuel tiene una característica similar que llaman "transmisión".

Tanto el entorno como los Chatbots resultantes son similares a Chatfuel.

2.2 COMPARATIVA

A continuación se llevará a cabo una comparativa entre los entornos anteriormente explicados, y también se comparará algunas características de los Chatbots resultantes de usar dichos entornos.

Hay que tener en cuenta que estos entornos con los que se ha establecido la comparación son en su versión gratuita, los complementos existentes en versiones de pagos no afectan a los resultados, y que los Chatbots generados por los entornos tendrán las características básicas, es decir, no tendrán alguna característica que un usuario haya creado antes de crear el Chatbot.

Las características a comparar entre los entornos serán:

1. Permite programar las acciones del Chatbots (Prog. acciones Cbots.): Se podrá programar las acciones del Chatbot para que pueda realizar cualquier tarea que el usuario quiera, pudiendo modificar el código para establecer nuevas funcionalidades.
2. Interacción por voz (Inter. voz): El usuario podrá usar el entorno con la voz, sin necesidad de utilizar el teclado, ratón o una pantalla táctil.
3. Sistema de mejora y entrenamiento por voz. (Sist. mejora y entr. por voz): El entorno proporcionará una herramienta para poder mejorar el Chatbot sin necesidad de tocar la estructura JSON del Chatbot, añadiendo las peticiones no reconocidas a dicha estructura.

Entre las características a comparar de los Chatbots resultantes se encuentran:

1. Ejecución en distintas aplicaciones o sistemas (Ejec. en distintas app./sist.): Se entiende como aplicación o sistema, a los programas donde los Chatbots funcionarán. No estarán enfocados sólo a una aplicación o sistema, pudiendo ser aplicaciones de mensajería, sistemas en el trabajo, hogar, webs, etc.
2. Tipo de entrada/salida de datos por voz. (E/S voz): Los Chatbots podrán reconocer entradas tanto por texto o por audio y las respuestas las darán en estos formatos.
3. Posibilidad de guardar los fallos para su mejora (Guardar fallos): Si el Chatbot no llega a reconocer una petición del usuario, podrá guardar dicha petición para que el entorno de creación de Chatbots lo mejore y vuelva a generar el modelo.
4. Posibilidad de enviar mensaje múltiple (Mensaje múltip.): El Chatbot tiene la posibilidad de realizar múltiples mensajes a distintos usuarios.

	Prog. acciones CB.	Inter. voz	Sist. mejora y entr. por voz	Ejec. en distintas app./sist.	E/S voz	Guardar fallos	Mensaje múltip.
IBM Watson Conversation	O			O	O		
Microsoft Azure Bot Service	O			O	O		
Google API.ai	O			O	O		
Amazon Lex	O			O	O		
Facebook Messenger API	O			O	O		
Chatfuel							O
Pandorabots				O	O		O
APPSUMO Mobile- Monkey				O			O

Tabla 1: Comparación de entornos de creación de Chatbots y Chatbots

Como se puede apreciar en la comparativa de los entornos (en color azul), carecen de la interacción por voz, centrándose más en la interfaz web para el desarrollo de Chatbots.

Estos entornos para poder mejorar y entrenar los Chatbots, necesitan que el usuario introduzca los cambios que desea realizar con los medios de interacción que los entornos les facilitan (teclado, ratón, etc.) pero sin la posibilidad de hacerlo con la voz.

La mayoría de los entornos comparados, al presentar API para su desarrollo, permiten la creación de nuevas acción que los Chatbots pueden realizar o modificación de algunas existentes, aunque esta última es más compleja o puede que no esté totalmente permitido.

En cuanto a la ejecución de los Chatbots creados, casi todos se pueden ejecutar en diversos sistemas, ya sea en aplicaciones de mensajería (Telegram, Facebook Messenger,

etc.), webs, sistemas del hogar, etc. Pero no es el caso de Chatfuel porque se centra más en las aplicaciones de mensajería.

Por otro lado, la comparativa de los Chatbots (en color verde) resultantes de los entornos, la mayoría pueden ofrecer una interacción por voz ya que se les puede programar acciones que realicen un reconocimiento de voz o porque ya esté a disposición del usuario en la API. En este caso, los Chatbots de Chatfuel y MobileMonkey carecen de esta ventaja. Pero Chatfuel tiene, al igual que Pandorabots y MobileMonkey, de proporcionar a sus Chatbots la posibilidad de enviar un mensaje a distintos usuarios.

En cambio, la singularidad de proporcionar al usuario la opción de guardar una petición no reconocida para luego mejorar el Chatbot no la presenta ningún Chatbot con su configuración básica.

2.3 CRÍTICA DEL ESTADO DEL ARTE

De lo explicado en la sección del estado del arte, en relación a los entornos de creación de Chatbots, se concluye que las grandes empresas (Facebook, Amazon, Google, IBM y Microsoft) son las que proporcionan tecnologías potentes para poder ofrecer entornos de programación con características avanzadas para ser utilizadas por desarrolladores, aunque usuarios sin muchos conocimientos pueden utilizar estos entornos sin adentrarse a programar en el código. Estas empresas teniendo el capital suficiente, no han aportado alguna herramienta o mejora de las ya existentes para poder realizar la tarea de crear Chatbots con la voz a través de la interfaz web de sus entornos.

Esta característica podría llegar a ser una revolución al igual que fue la llegada del ratón a los ordenadores, cambiaría la lógica que tienen los entornos para comportarse como un Chatbot. Esto supondría que cualquier usuario acceda a tareas específicas con mayor facilidad y rapidez, en vez de estar navegando en la interfaz de la web, aplicación, ordenador, etc., para encontrar una opción. Ésta es una de las ventajas que ofrece la lógica de los Chatbots.

Por otro lado, las pequeñas empresas (Pandorabots, Chatfuel y MobileMonkey) en comparación a las otras empresas, aparte de ofrecer un entorno amigable para su uso presentan funcionalidades parecidas a los entornos de las otras empresas como probar el Chatbot en el entorno, mejorarlo introduciendo cambios al Chatbot manualmente, entre otras.

En cuanto a los Chatbots creados por estos entornos de las grandes empresas, tienen la posibilidad de interaccionar con el usuario mediante la voz haciendo más interesante el uso de los mismo. También permiten la ejecución de los Chatbots en diferentes plataformas o sistemas, logrando que diferentes usuarios puedan utilizarlos.

En cambio, los Chatbots creados por algunos entornos de las otras empresas no dan la posibilidad de que se ejecuten en distintas plataformas, centrándose en plataformas de mensajería como es el caso de Chatfuel.

Pero todos estos Chatbots no presentan la característica de que el usuario guarde la petición no reconocida para que lo pueda mejorar y entrenar con estas peticiones más adelante. Algo que haría más cómodo el entrenar y generar el modelo es si tuvieran una herramienta que se use con la voz para leer las peticiones no reconocidas y que genere el modelo tras las indicaciones del usuario para introducir las mejoras.

2.4 PROPUESTA

Con lo visto en anteriores apartados, se deduce que los Chatbots están siendo un medio que muchos usuarios utilizan como medio de interacción para buscar información o adquirir recursos. Aunque las interacciones con los Chatbots aún no son muy precisos y necesitan de numerosas mejoras, estas interacciones persona-máquina irán evolucionando con la tecnología haciendo que uso de los sistemas sean más cómodo y preciso.

Entonces, un buen entorno de creación de Chatbots tiene que ofrecer la flexibilidad de aprendizaje, diversos tipos de interacción con los usuarios, dar la posibilidad de reprogramar los Chatbots creados, aportar un soporte,

Este trabajo tiene como objetivos proporcionar distintas formas de interacción, para que el usuario se sienta más cómodo y sea más rápido el poder crear Chatbots, dar la libertad de uso del entorno al usuario a la hora de mantener su Chatbot, afianzar la idea de que un Chatbot es un sistema el cual necesita una interacción humana para poder realizar sus tareas.

2.5 RESUMEN DEL CAPÍTULO

En este capítulo se ha dado una visión más profunda de los que son los Chatbots y de los entornos que los crean.

Los Chatbots aunque en su mayoría se encuentran en aplicaciones de mensajería, también se encuentra en gran medida en distintos sistemas (trabajo, hogar, webs, toma de decisiones, ordenadores a bordo, etc.) ayudando a los usuarios a realizar sus tareas. Estos sistemas pueden tener tecnologías de Machine Learning para entender el lenguaje humano o presentar una lógica sencilla y cerrada.

Finalmente, se ha podido observar con las comparaciones que a los entornos de creación de Chatbots les falta ofrecer herramientas para que la interacción se pueda realizar con la

voz, y que sus Chatbots resultantes sean capas de guardar las peticiones no reconocidas para que el usuario, mediante la voz, pueda modificar la estructura del Chatbot introduciendo aquellas peticiones no reconocidas.

CAPÍTULO 3: OBJETIVOS

En este capítulo se detallarán los objetivos a conseguir en este proyecto, que han sido mencionados en anteriores capítulos. También se dará una visión global del funcionamiento del entorno construido en este proyecto.

El proyecto se enfocará en conseguir un entorno y Chatbots resultantes con las siguientes características:

ENTORNO DE CREACIÓN DE CHATBOTS

1. Capaz de comunicarse con los usuario por texto y voz.
2. Tiene que reconocer una estructura de Chatbots para su procesamiento.
3. Debe proporcionar un servicio de mejora y entrenamiento.
4. El usuario podrá realizar cambios de la estructura de su Chatbot.
5. El entorno tiene que generar los ficheros necesarios automáticamente.
6. Cada vez que se mejore el Chatbot, se volverá a entrenar y generar su modelo.
7. Se podrá testear el Chatbot que se esté creando.
8. Permitirá ejecutar el Chatbot en distintos sistemas, programando las acciones del Chatbot.

CHATBOTS RESULTANTES

1. Podrá interaccionar por voz o texto.
2. Permitirá al usuario guardar los errores.
3. Indicará en todo momento el estado en que se encuentra el usuario.

A continuación se especificará el funcionamiento general del entorno y una visión global de su arquitectura.

El sistema o entorno completo está compuesto por tres distintos sub-sistemas que se complementan para ofrecer un mejor servicio al usuario. Estos sub-sistemas son el motor de Chatbots, Metachatbot y el sistema de entrenamiento y mejora.

Cuando el usuario ejecute el programa, entra en funcionamiento el Metachatbot. Éste a su vez usará el motor de Chatbots para reconocer la estructura de los Chatbots.

El usuario al indicar por texto o voz sus intenciones, el Metachatbot lo procesará con ayuda del motor de Chatbot para dar una respuesta o ejecutar una acción.

Al finalizar todas las tareas que el usuario haya indicado, éste podrá generar el Chatbot que está construyendo haciendo que el Metachatbot genere todos los archivos necesarios, modelo, código, etc. El usuario también podrá probar el Chatbot creado desde el Metachatbot.

Por otro lado, si al probar su Chatbot no reconocer alguna intención podrá mejorarlo usando el sistema de entrenamiento y mejora, que tras mejorar el Chatbot actualiza el modelo volviéndolo a generar automáticamente o si lo prefiere el usuario, puede programar directamente la estructura del Chatbot para que pueda reconocer las intenciones fallidas y generar el modelo a mano.

CAPÍTULO 4: IMPLEMENTACIÓN DEL SISTEMA

En este capítulo se detallará, en el primer apartado, la arquitectura que tienen los Chatbots resultantes y todos los Chatbots del entorno: Metachatbot y el sistema de entrenamiento y mejora.

Posteriormente, en distintos apartados se entrará más en detalle sobre las herramientas, análisis, diseño y pruebas de cada subsistema: motor de Chatbots, Metachatbot y el sistema de entrenamiento y mejora.

De igual forma se hablará del Chatbot “Lista de la Compra” que ha servido de ayuda para entender el funcionamiento de los Chatbots. Con este apartado se planteará el uso y comportamiento de todo el entorno o sistema.

4.1 ARQUITECTURA

En este apartado se mostrará la estructura que tiene el sistema software implementado, al igual que el fichero JSON que tienen los Chatbots, un diagrama de clases de los Chatbots que indica cómo están relacionadas y por último se presentará el flujo general que tiene la ejecución de un Chatbot.

4.1.1 JSON

Según la definición de (Enesimus 2017):

“JSON es un formato de datos basado en texto que sigue la sintaxis de objeto de JavaScript, popularizado por Douglas Crockford. Aunque es muy parecido a la sintaxis de objeto literal de JavaScript, puede ser utilizado independientemente de JavaScript, y muchos ambientes de programación poseen la capacidad de leer (analizar; parse) y generar JSON.”

Con esta definición, se puede entender que JSON no es más que la interpretación de datos en texto plano con una sintaxis de JavaScript, presentando una estructura formada por claves y valores.

A continuación se muestra una imagen en la que presenta la estructura JSON que tienen todos los Chatbots, en este caso se ha tomado como ejemplo el Chatbot “Lista Compra”.



Ilustración 1: Estructura del Chatbot "Lista Compra"

Tal y como se muestra en la imagen, la estructura está formada por una clave a la que se le asocia un valor. El valor puede ser un elemento simple (número, literal, etc.) o un elemento con estructura clave-valor, o también se puede presentar una lista que contenga elementos simples o estructurados.

En la imagen se presenta una clave que indica el nombre del Chatbot, en este caso "Lista_Compra". El valor de esta clave está formado por una lista de elementos estructurados. Estos elementos o subestructuras tienen a su vez otras claves (tag, patterns, responses y action) que tendrán asociados otros elementos.

Como se puede apreciar, el valor de las claves "tags" son elementos simples que contienen las intenciones de los usuarios, es decir, la idea de lo que el usuario pretende realizar.

Las claves "patterns" tienen asociada una lista de elementos simples que son las peticiones que el usuario utilizará para indicar su intención al Chatbot.

Las claves "responses", al igual que los "patterns", tienen una lista de elementos simples que son las respuestas asociadas a las peticiones que el Chatbot mostrará al usuario.

La clave "action" tiene como valor un elemento simple que será aquella acción que ejecutará el Chatbot al reconocer la intención del usuario.

4.1.2 Diagrama de clases del sistema

En la siguiente imagen se mostrará la interacción interna del entorno que será común a todos los Chatbots del entorno o creados por él. Se tomará como ejemplo la interacción de un Chatbot. El motor de Chatbots en el diagrama de clases está representado por “**CProcessor**” y el sistema de mejora y entrenamiento es representado por “**SolveError**”.

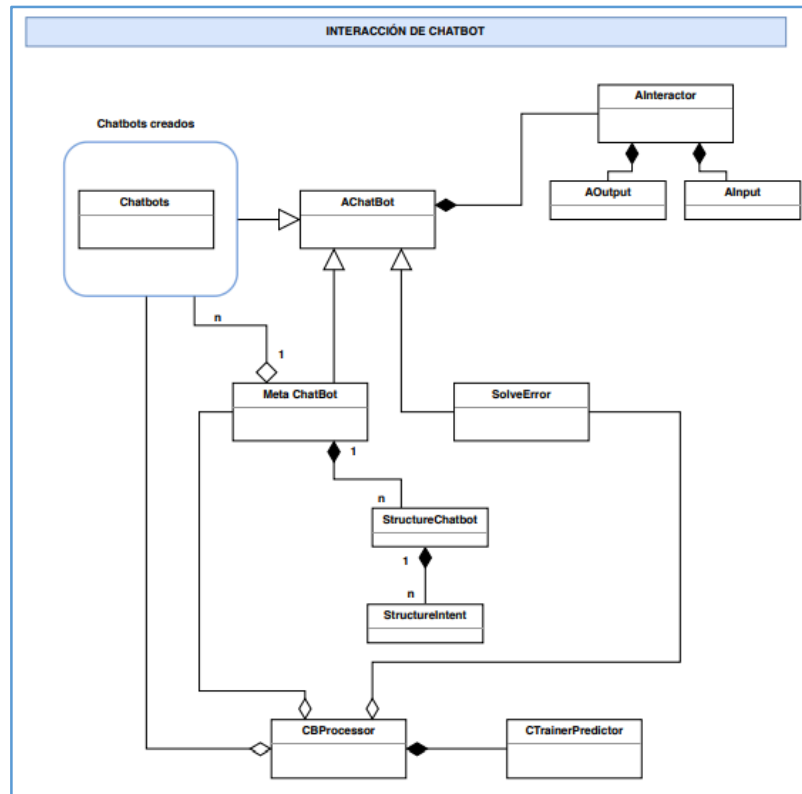


Ilustración 2: Interacción de clases de un Chatbot.

Como se puede observar en la imagen, los Chatbots de este entorno y los creados por él están relacionados con el motor de Chatbots (CBProcessor) y con la clase “AChatBot” que tiene todas las variables básicas de un Chatbot para que funcione correctamente. Esta clase a su vez es quien se relaciona con la clase de “AInteractor” que posee los distintos módulos de interacción que los Chatbots pueden usar.

Por otro lado, el motor de Chatbots interacciona con la clase de “CTrainerPredictor” que se encarga de utilizar el modelo generado para reconocer las peticiones del usuario.

Por último, destaca que el Metachatbot y el sistema de mejora y entrenamiento (Solve Error) son a su vez Chatbots. Característica que proporciona la posibilidad de tener la interacción por voz. El Metachatbot también tiene acceso a clases que representan la

estructura del JSON de los Chatbots para facilitar la creación, modificación y eliminación de los mismos.

4.1.3 Flujo de ejecución

En esta sección se explicará como es el flujo de ejecución e interacción entre las clases del entorno. Para ello, se mostrará en una imagen de flujo de estados general que presentan los Chatbots del entorno o los creados por él.

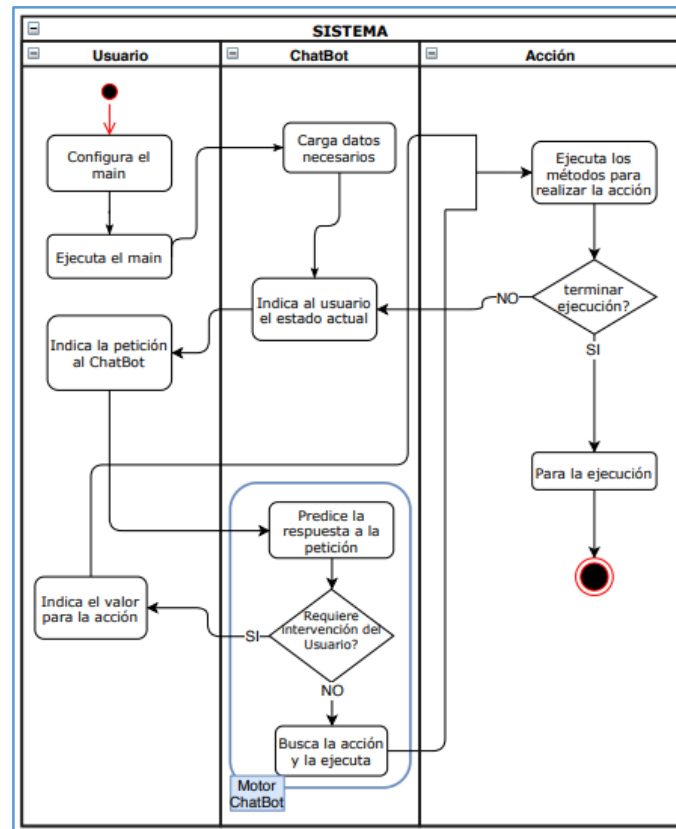


Ilustración 3: Flujo de estados general de un Chatbot.

Para explicar la imagen se tomará como ejemplo un caso de uso del Metachatbot, en concreto será el de crear un Chatbot.

Como se puede apreciar, el flujo se inicia con el usuario cuando realiza las modificaciones necesarias para ejecutar el Metachatbot. Una vez hecha esta parte y ejecutar el programa principal, el sistema o entrono llama al Metachatbot que se encarga de cargar todos los datos que necesita. Posteriormente mostrará al usuario el estado en el que se encuentra para que el usuario le indique la petición de crear un nuevo Chatbot.

Es aquí cuando el Metachatbot hace uso del motor de Chatbots para clasificar la petición del usuario y dar una respuesta óptima, esto se hará con cada petición que realice el usuario. Si dicha petición requiere de una segunda intervención, el Metachatbot le

indicará al usuario que le indique el dato que necesita, en caso contrario pasará a ejecutar la acción relacionada a la petición. En este ejemplo, el Metachatbot necesita que el usuario le indique el valor que necesita para ejecutar la acción.

Luego de que el usuario introduzca el valor necesario, que en este ejemplo es el nombre del nuevo Chatbot, el Metachatbot pasará a ejecutar la acción previamente guardada. Como la acción no es para salir del Metachatbot, indica al usuario el estado en el que está con un mensaje de éxito si se ha creado el Chatbot o un mensaje de error si no se ha podido crear.

Si el usuario indica que quiere ver el Chatbot actual, el Metachatbot nuevamente utilizará el motor de Chatbots, pero en este caso al no ser una acción que requiera de una segunda intervención del usuario, el Metachatbot pasa a ejecutar la acción directamente.

Nuevamente se pasa por el estado de comprobación de que no es una acción de finalización, para después mostrar el resultado de la acción.

Este camino se seguirá ejecutando para crear, editar y eliminar los componentes del Chatbot: tags (intenciones), patterns (peticiones), responses (respuestas) y action (acciones).

Cuando el usuario no quiera continuar creando el Chatbot, indicará al Metachatbot que quiere salir y éste enviará la acción de finalización al estado de parar ejecución para salir del Metachatbot.

Este es el flujo que siguen todos los Chatbots y se puede realizar por voz o introduciendo por teclado las peticiones.

4.2 MOTOR DE CHATBOTS

Este subsistema no es un Chatbot, es un programa implementado para poder realizar las tareas de reconocimiento de intenciones y de la estructura de los Chatbots. Siendo el encargado de utilizar técnicas de Machine Learning que se detallarán más adelante.

4.2.1 Análisis

El motor de Chatbots implementado para este proyecto es capaz de reconocer las ambigüedades de las intenciones de los usuarios gracias al Procesamiento Natural del Lenguaje (PNL, o en inglés NLP).

El PNL estudia la interacción entre los ordenadores y el lenguaje humano, cuyos modelos se enfocan en el comportamiento cognitivo en general y a la organización de la memoria. Al principio se basaba en un conjunto de reglas hechas a mano, pero en 1980 se introdujo el aprendizaje automático.

Además de reconocer el lenguaje de los humanos, reconoce la estructura que usan los Chatbots del sistema y los creados por dicho sistema.

4.2.1.1 Requisitos Funcionales

Los requisitos funcionales son aquellas funciones que realiza el sistema, ya sea entradas, acciones y salidas.

Entre los requisitos funcionales para este programa se encuentran:

- RF01: Tiene que reconocer las ambigüedades de las peticiones.
- RF02: Procesará las intenciones para dar una respuesta.
- RF03: En caso de no encontrar coincidencias no dará ninguna respuesta.
- RF04: Escogerá la respuesta con mayor porcentaje de acierto.
- RF05: Generará el modelo de los Chatbots.
- RF06: Guardará el modelo generado del Chatbot.
- RF07: En el caso de que el Chatbot no tenga un modelo, lo generará usando la estructura de dicho Chatbot.
- RF08: Se encargará de entrenar los Chatbots.

4.2.1.2 Requisitos No Funcionales

Los requisitos no funcionales son las características de funcionamiento, es decir, atributos de calidad del sistema.

Los requisitos no funcionales del motor de Chatbots son:

- RNF01: Mostrará mensajes del estado en el que se encuentra la ejecución del entrenamiento.
- RNF02: Mostrará un mensaje al finalizar de generar el modelo.

4.2.2 Diseño

El lenguaje con el que se implementó es Python. Este programa realiza el procesamiento del lenguaje natural (PLN) gracias a la librería NLTK.

Esta librería se utiliza para analizar las intenciones y generar así una lista de posibles entradas y salidas. Estas se consiguen gracias al saco de palabras, que contiene las raíces de las intenciones que están relacionadas a unas respuestas, ambas se encuentran la estructura JSON.

Tras conseguir estas listas de entradas y sus salidas, se hace uso de las redes neuronales con ayuda de KERAS.

Es aquí donde se entrena las neuronas usando las entradas y salidas, la finalizar el entrenamiento se guarda el modelo resultante para su posterior uso.

Cuando se ejecuta el programa, el motor de Chatbots carga el modelo generado del Chatbot en ejecución. Al introducir una petición, realiza una clasificación para hallar la mejor coincidencia con los datos de entrenamiento y poder dar la respuesta correcta.

Si encuentra alguna respuesta, el motor de Chatbots devuelve los datos que necesita el Chatbot para llevar a cabo la acción correspondiente. Esta acción puede ser devolver la respuesta asociada o desencadenar otras acciones.

El motor de Chatbot se encarga de utilizar el modelo del Chatbot para encontrar la mejor respuesta. Una vez encontrado, comunica al Chatbot qué acción ejecutar y éste lo realiza a través de su estructura. Otra característica de este programa, es que puede entrenar y realizar el modelo del Chatbot y utilizarlo para reconocer nuevas entradas de intenciones por parte del usuario. La siguiente esquema muestra este funcionamiento.

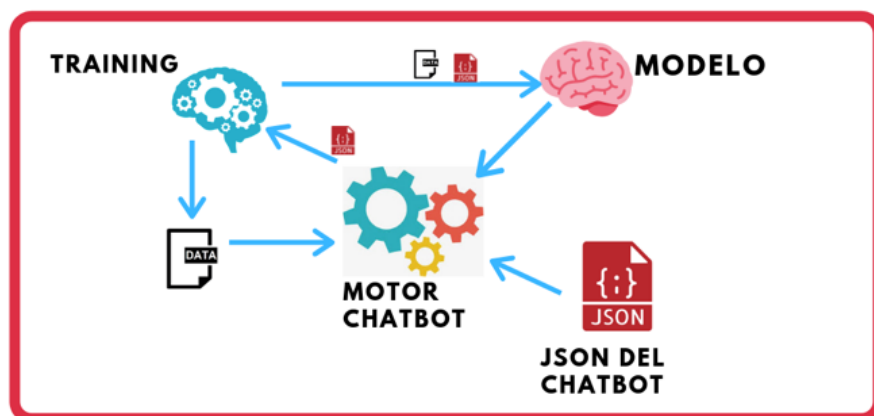


Ilustración 4: Entrenamiento y creación del modelo de un Chatbot.

Para el entrenamiento, el motor de Chatbot lee el JSON del Chatbot. Con esto se produce unos datos necesarios para generar el modelo del Chatbot que luego estará a disposición del motor del Chatbot.

4.2.3 Pruebas

En las siguientes imágenes se mostrarán el cumplimiento de algunos requisitos funcionales y no funcionales.

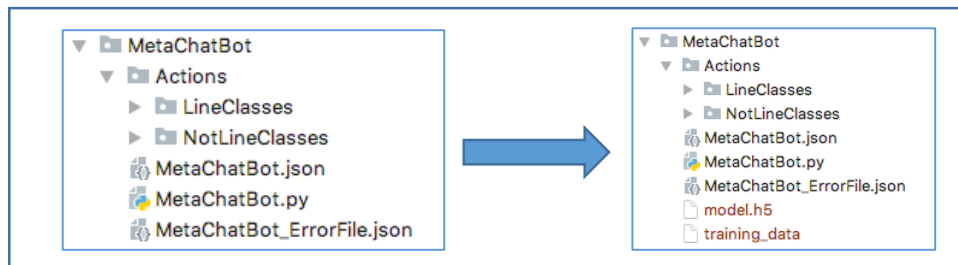


Ilustración 5: Modelo generado por el motor de Chatbots.

Como se puede observar, el motor de Chatbots ha sido capaz de realizar el entrenamiento y crear el modelo con los datos de la estructura JSON del Chatbot. En el ejemplo se muestra un estado previo, sin modelo ni datos de entrenamiento, y el estado después de ejecutar el motor de Chatbots cuyo resultado son los ficheros que se encuentran en color rojo.

4.3 METACHATBOT

Este sub-sistema es el encargado de la interacción con el usuario y el de generar la estructura de los Chatbots.

El Metachatbot tiene como objetivo crear Chatbots por voz utilizando la estructura definida en JSON, haciendo uso del motor de Chatbots.

4.3.1 Análisis

Desde la ejecución del Metachatbot, éste se encargará de cargar todos los Chatbots creados hasta el momento a su lista de Chatbots. Es decir, tiene la capacidad de pasar la estructura JSON a código Python.

Y a medida que se vaya creando un Chatbot, internamente el Metachatbot está generando la estructura en JSON para que después se generen los ficheros de código, carpetas, etc., del Chatbot que se esté creando. Además, proporcionará la opción de ejecutar el Chatbot creado para testarlo y ver si falla.

También es el encargado de llamar a los métodos del motor de Chatbots para que se genere el entrenamiento y modelo del Chatbot que se esté creando.

Ofrecerá la posibilidad de realizar estas tareas con la interacción por voz.

4.3.1.1 Requisitos Funcionales

Los requisitos funcionales son aquellas funciones que realiza el sistema, ya sea entradas, acciones y salidas.

Dentro de los requisitos funcionales para este sub-sistema están:

- RF01: Tendrá interacción por voz o texto.
- RF02: Convertirá la estructura JSON a código Python.
- RF03: Convertirá código Python a estructura JSON.
- RF04: Permitirá crear distintos Chatbots.
- RF05: Permitirá mostrar el Chatbot actual, es decir, el Chatbot que se está editando.
- RF06: Permitirá mostrar todos los Chatbots que se están creando y/o se han creado.
- RF07: Cargará automáticamente todos los Chatbots que se hayan creado a la lista de Chatbots.
- RF08: Permitirá cambiar de Chatbot.
- RF09: Guardará los cambios de un Chatbot en todo momento.
- RF10: Permitirá crear Intenciones para cada Chatbot.
- RF11: Mostrará la Intención actual, es decir, la Intención que se está editando.
- RF12: Mostrará la lista de todas las intenciones de un Chatbot.
- RF13: Podrá mostrar la Acción creada por cada Intención.
- RF14: Podrá añadir, eliminar, cambiar un Chatbot.
- RF15: Podrá añadir, eliminar, cambiar una Intención.
- RF16: Podrá añadir, eliminar una acción.
- RF17: Podrá generar el Chatbot actual creando sus ficheros Python y JSON.
- RF18: Mostrará mensajes indicativos del estado de la creación del Chatbot.
- RF19: Permitirá elegir el Chatbot que desee ejecutar.
- RF20: Permitirá parar la ejecución del Chatbot que ha ejecutado.
- RF21: Podrá guardar las entradas no reconocidas, usando una de las palabras clave “Guardar sentencia” o “Salvar sentencia”, en un fichero JSON.
- RF22: Podrá parar su ejecución.
- RF23: Permitirá guardar las sentencias no reconocidas.

4.3.1.2 Requisitos No Funcionales

Los requisitos no funcionales son las características de funcionamiento, es decir, atributos de calidad del sistema.

Entre los requisitos no funcionales se encuentran:

- RNF01: Los mensajes mostrados deben ser claros.
- RNF02: El tiempo de espera no debe ser mayor a cinco segundos.
- RNF03: El lenguaje usado para las interacciones debe ser español.
- RNF04: Debe indicar en qué estado se encuentra el usuario.

4.3.2 Diseño

Para que este sub-sistema pudiese generar todos los ficheros necesarios se utilizó código Python que facilita esta tarea. Se implementó métodos que convierten la estructura JSON de los Chatbots en una cadena de texto, con la sintaxis de Python para después incrustar dicha cadena en un fichero Python.

Otra característica es que puede leer y añadir caracteres especiales a la estructura JSON, lo que se implementó con el método `"json.dump ()"` de la librería JSON de Python, añadiendo ciertos parámetros para que guardase los caracteres especiales.

El Metachatbot guarda la estructura de los Chatbots en objetos que mantienen la información, gracias a estos objetos el Metachatbot es capaz de añadir, editar y eliminar nuevos Chatbots, intenciones, respuestas y acciones, según lo vaya solicitando el usuario.

Este sub-sistema, al igual que los otros Chatbots, puede comunicarse por voz. Esto lo realiza gracias a un servicio XML-RPC.

El SimpleXMLRPCServer hace uso del threading para comunicar el host, previamente configurado con el código, con el cliente. Puede hacerse con un móvil desde el cual el usuario podrá ejecutar un script que enviará la petición al puerto del thread para ejecutar el código del módulo de entrada, que responderá ejecutando un código en el módulo de salida para que se produzca una respuesta con audio.

En la siguiente imagen se muestra un diagrama con las acciones implementadas.

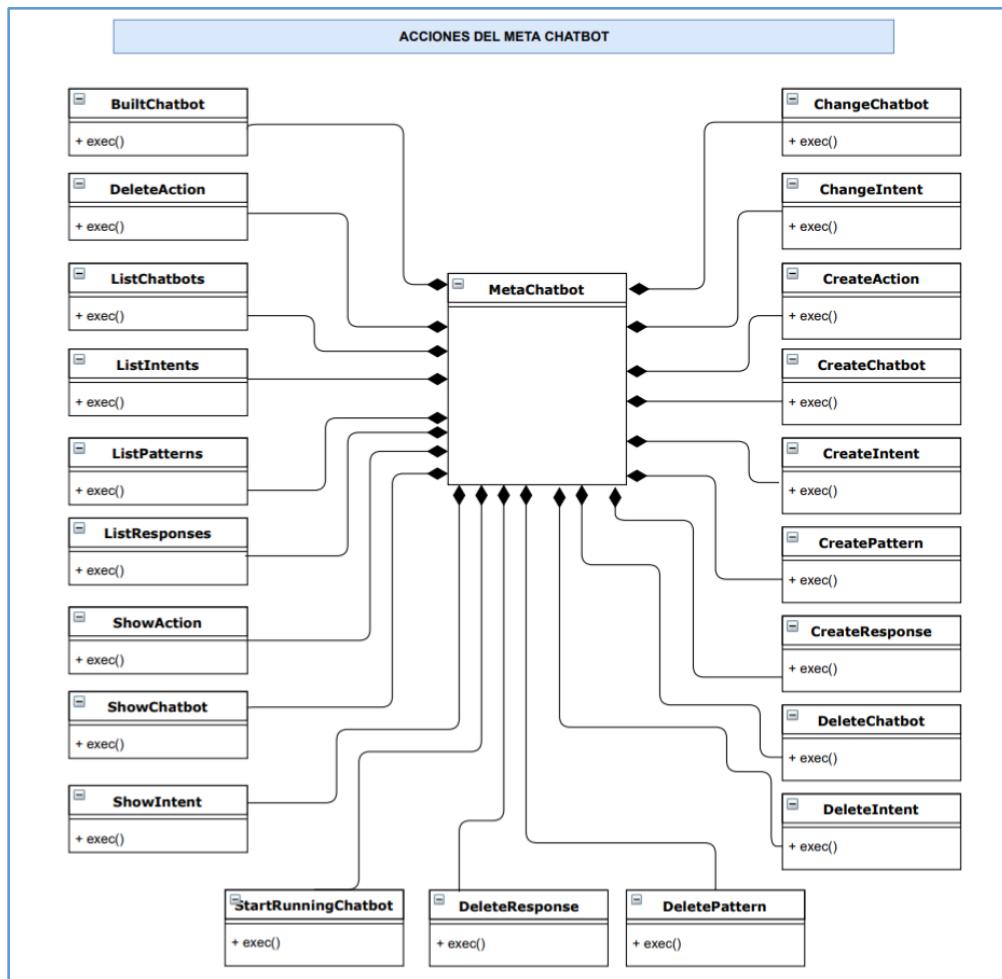


Ilustración 6: Acciones del Metachatbot.

Todas estas acciones permiten el cumplimiento de los requisitos especificados anteriormente.

4.3.3 Pruebas

En las siguientes imágenes se mostrarán algunos ejemplos de la funcionalidad lograda que cumplen con los requisitos funcionales y no funcionales.

Las peticiones del usuario estarán en color verde y las respuestas del sistema en color negro.

- Crear el Chatbot “compra online”: para la ejecución de este ejemplo, en el sistema ya existe otros Chatbots. Se mostrará la interacción teclado/pantalla.

```

Se está ejecutando el ChatBot "Meta ChatBot".
=> listar chatbots
Los chatbot creados son: [ memoria online, compra online ]
=> chatbot actual
El ChatBot actual es "memoria online"
=> listar intenciones
Las Intenciones del ChatBot "memoria online" son: finishRunningChatbot, saveSentence, dontSaveSentence, buscar opcion, mostrar memoria
=> crear chatbot
Qué chatbot quiere añadir?
=> Lista Compra
El ChatBot "Lista Compra" se ha añadido correctamente.
=> chatbot actual
El ChatBot actual es "Lista Compra"
|>

```

Ilustración 7: Prueba Metachatbot - Crear Chatbot.

Como se puede ver, las acciones implementadas han sido ejecutadas gracias a que se ha podido reconocer la sintáxis del lenguaje.

- Crear una Intención, Patrones, Respuestas y Acción: a continuación se muestra una imagen donde el usuario ha creado una serie de componentes (intención, patrones, respuestas y acción) para su Chatbot.

```

=> crear una intencion
Qué intención quiere añadir?
=> nuevo producto
La Intención "nuevo producto" se ha añadido correctamente.
=> mostrar intencion actual
La Intención actual es "nuevo producto".
=> crear un patron
Qué patrón quiere añadir?
=> quiero introducir un producto
Se ha añadido el Patrón "quiero introducir un producto" correctamente.
=> crear patrón
Dígame el nombre del patrón
=> añadir producto
Se ha añadido el Patrón "añadir producto" correctamente.
=> nueva respuesta
Qué respuesta quiere crear?
=> qué producto quiere añadir?
Se ha añadido la Respuesta "qué producto quiere añadir?" correctamente.
=> crear respuesta
Qué respuesta quiere añadir?
=> Diga el nombre del producto
Se ha añadido la Respuesta "Diga el nombre del producto" correctamente.
=> crear accion
Qué acción quiere crear ?
=> newProduct
Se ha guardado la acción "newProduct".
=>

```

Ilustración 8: Prueba Metachatbot - Crear elementos del Chatbot.

Como se puede apreciar, el Metachatbot reconoce distintas formas de patrones y puede dar una respuesta al azar entre la lista de Respuestas. A su vez, puede mostrar y reconocer caracteres especiales tales como las tildes, eñes, signos, etc.

- Generar el Chatbot: Para realizar este ejemplo, se han creado más intenciones del Chatbot que se está construyendo.

En la siguiente imagen se puede ver que el Metachatbot utiliza el motor de Chatbots para entrenar y generar el modelo del Chatbot “compra online” reconociendo las raíces de los patrones introducidos por el usuario.

```
=> generar chatbot
16 documents
6classes [dontSaveSentence, finishRunningChatbot, forma pago, nuevo producto, pagar, saveSentence]
20unique stemmed words [0, a ad, chatbot, con, de, del, ejecut, guard, introduc, no, pag, par, product, quier, realiz, sal, salv, tarjet, termin, un]
[[0 1 0 0 0 0]
 [0 1 0 0 0 0]
 [0 1 0 0 0 0]
 [0 1 0 0 0 0]
 [0 1 0 0 0 0]
 [0 0 0 0 0 1]
 [0 0 0 0 0 1]
 [1 0 0 0 0 0]
 [1 0 0 0 0 0]
 [0 0 0 1 0 0]
 [0 0 0 1 0 0]
 [0 0 0 1 0 0]
 [0 0 0 1 0 0]
 [0 0 0 1 0 0]
 [0 0 1 0 0 0]
 [0 0 1 0 0 0]]
Epoch 1/1000

8/16 [=====>.....] - ETA: 0s - loss: 2.4227 - acc: 0.1250
16/16 [=====] - 1s 63ms/step - loss: 2.4501 - acc: 0.1250
Epoch 2/1000
```

Ilustración 9: Prueba Metachatbot - Generar Chatbot.

También genera los ficheros necesarios tales como la estructura JSON, las acciones definidas, el modelo, datos del entrenamiento y código de la clase principal del Chatbot, tal y como se puede apreciar en la siguiente imagen.

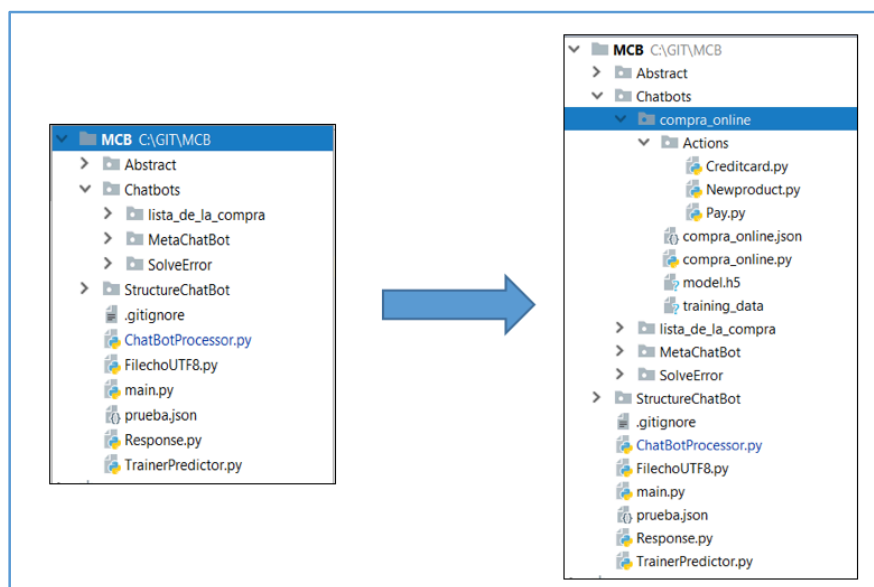


Ilustración 10: Prueba Metachatbot - Ficheros creados automáticamente.

Como se puede observar, el Metachatbot ha generado todos los ficheros de código y los ficheros de las acciones están a disposición del usuario para que los pueda editar.

4.4 LISTA DE LA COMPRA

Este Chatbot fue el primero en construirse con el fin de entender cómo funcionan los Chatbots. Presenta una estructura sencilla y en un principio no contaba con la posibilidad de interaccionar mediante la voz, pero al desarrollar más el entorno se pudo crear una nueva “Lista de la Compra” que tiene esta característica.

Al igual que todos los Chatbots utiliza el motor de Chatbots para poder generar su modelo y entrenamiento.

4.4.1 Análisis

El Chatbot “Lista de la Compra” presenta una estructura sencilla, ya que desde un inicio fue implementado para responder a las intenciones de los usuarios y mostrar los resultados de los mismos con una interacción sólo por texto.

La funcionalidad de este Chatbot es poder guardar el listado de productos que el usuario quiere comprar pudiendo editarla según su conveniencia, para luego realizar la compra o cancelarla.

4.4.1.1 Requisitos Funcionales

Los requisitos funcionales son aquellas funciones que realiza el sistema, ya sea entradas, acciones y salidas.

Entre los requisitos funcionales, se encuentran:

- RF01: Debe permitir añadir nuevos productos, mostrando un mensaje de éxito.
- RF02: Se podrá eliminar los productos ya introducidos, mostrando un mensaje de éxito.
- RF03: Dará un mensaje de error en caso de que no exista el producto a eliminar.
- RF04: Permitirá mostrar todos los productos de la cesta.
- RF05: En caso de realizar la compra, dará opciones de pago.
- RF06: Tras seleccionar el modo de pago se mostrará un mensaje de éxito.
- RF07: En caso de cancelar la compra, eliminará todos los productos guardados mostrando un mensaje de aviso.
- RF08: La interacción será por texto.

- RF09: Todos los mensajes de aviso se visualizarán en pantalla.
- RF10: Podrá guardar las entradas no reconocidas, usando una de las palabras clave “Guardar sentencia” o “Salvar sentencia”, en un fichero JSON.
- RF11: Se podrá parar la ejecución del Chatbot.

4.4.1.2 Requisitos No Funcionales

Los requisitos no funcionales son las características de funcionamiento, es decir, atributos de calidad del sistema.

Entre los requisitos no funcionales para este Chatbot se encuentran:

- RNF01: El usuario debe saber en cada momento en qué estado se encuentra el sistema.
- RNF02: El lenguaje que se usará para la interacción será el español.
- RNF03: Los mensajes de respuesta deberán ser claros.
- RNF04: El Chatbot no debe tardar más de cinco segundos en dar una respuesta.
- RNF05: La interacción debe ser fácil de realizar.

4.4.2 Diseño

El lenguaje con el que se implementó es Python y su estructura está hecha con JSON, y el diseño se basa en la comunicación con el usuario para recibir las intenciones y procesarlas con el motor de Chatbots para dar una respuesta.

Por ejemplo, cuando un usuario introduce una petición el Chatbot hace uso del motor de Chatbots para clasificar la petición y asignarle una intención que tiene asociada una acción. Esta acción se ejecutará con el fin de realizar lo que el usuario haya pedido.

Hay que tener en cuenta que las acciones de los Chatbots creados vendrán vacías, pudiendo ser editadas libremente por los usuarios para que realicen cualquier acción.

4.4.3 Pruebas

En las siguientes imágenes se verá el funcionamiento básico de este Chatbot, mostrando el cumplimiento de requisitos funcionales y no funcionales de algunas intenciones.

```

El modelo ya existe
Ejecutando el Predictor...
Predictor ejecutado correctamente.
Se está ejecutando el ChatBot "Lista_Compra".
=> nuevo producto
¿cuál es el producto que desea añadir?
=> pera
Se ha introducido el producto.
=> insertar producto
¿qué producto quiere añadir?
=> manzana
Se ha introducido el producto.
=> ver la compra
La cesta tiene: ['pera', 'manzana']
=> quiero añadir otro producto
Dígame el producto que desea guardar
=> jamón
Se ha introducido el producto.
=> quiero ver la compra
La cesta tiene: ['pera', 'manzana', 'jamón']
=>

```

Ilustración 11: Prueba Lista Compra - Insertar y Mostrar la cesta.

Como se puede observar, este Chatbot tiene el mismo funcionamiento que el Metachatbot. Tras una petición por parte del usuario, el Chatbot ejecuta una acción o solicita el dato necesario para poder llevar a cabo la acción.

En la siguiente imagen se muestra cómo se puede salir de una acción cuando el Chatbot se confunde, y en caso de que no reconozca una petición tiene la posibilidad de guardar esa petición para su mejora y también se observa que se puede terminar la ejecución del Chatbot. Todas estas funcionalidades son básicas para todos los Chatbots creados por entorno y para el Metachatbot y el sistema de entrenamiento y mejora.

```

=> prefiero cambiar el producto por otro
¿qué producto quiere añadir?
=> cancelar
Se ha cancelado la operación.
=> guardar
No se ha reconocido la sentencia "guardar".
=> guardar sentencia
Se ha guardado la sentencia "guardar" que se le asoció con la intención "saveSentence".
=> salir
Se ha parado de ejecutar el Chatbot "Lista_Compra".

```

Ilustración 12: Prueba Lista Compra - Cancelar acción, guardar error y terminar ejecución.

4.5 SISTEMA DE MEJORA Y ENTRENAMIENTO

Este subsistema proporciona al usuario la capacidad de resolver las entradas no reconocidas. El sistema de mejora y entrenamiento, al igual que el Metachatbot, es un Chatbot por lo que también utiliza la estructura definida para un Chatbot, el motor de Chatbots, está implementado con Python y ofrece la interacción por texto o voz, haciendo más cómo el resolver los errores. Dentro del entorno se llama “Solve Error”.

4.5.1 Análisis

Este Chatbot tiene la funcionalidad de editar la estructura JSON del Chatbot a mejorar sin que el usuario tenga que tocar directamente la estructura.

Tras seleccionar todos los fallos, indicar los valores correctos y editar el JSON, el sistema de mejora y entrenamiento vuelve a entrenar y generar el modelo automáticamente del Chatbot que se esté mejorando.

4.5.1.1 Requisitos Funcionales

Los requisitos funcionales son aquellas funciones que realiza el sistema, ya sea entradas, acciones y salidas.

Entre sus requisitos funcionales se encuentran:

- RF01: Permitirá mostrar todos los Chatbots, incluso el Metachatbot y él mismo.
- RF02: Permitirá seleccionar la petición que no se ha reconocido y una intención a la que se le quiera asociar para resolverlo.
- RF03: Al seleccionar el Chatbot a resolver, cargará sus intenciones y sus errores automáticamente.
- RF04: Mostrará los errores e intenciones del Chatbot a resolver.
- RF05: Permitirá guardar las soluciones.
- RF06: Permitirá mostrar las soluciones guardadas.
- RF07: Permitirá mostrar el último error e intención seleccionado.
- RF08: Resolverá los errores que se han resuelto y guardado en la lista.
- RF09: Editará el JSON del Chatbot que se está resolviendo para introducir los cambios.
- RF10: Volverá a entrenar el Chatbot editado.
- RF11: Volverá a generará el modelo del Chatbot editado.
- RF12: Mostrará mensajes indicativos del estado en el que se encuentra el usuario.

- RF13: Podrá guardar las entradas no reconocidas, usando una de las palabras clave “Guardar sentencia” o “Salvar sentencia”, en un fichero JSON.
- RF14: Podrá parar su ejecución.

4.5.1.2 Requisitos No Funcionales

Los requisitos no funcionales son las características de funcionamiento, es decir, atributos de calidad del sistema.

Presenta los siguientes requisitos no funcionales:

- RNF01: El lenguaje, tanto escrito como hablado, debe ser en español.
- RNF02: Las respuestas deben ser claras en cada situación.
- RNF03: Los mensajes de error deberán ser claros.
- RNF04: El tiempo de respuesta no deberá ser muy larga.

4.5.2 Diseño

Este sistema de mejora y entrenamiento tiene como objetivo poder realizar mejoras del modelo de los Chatbots con los cambios que el usuario le indique, y lo puede realizar mediante la interacción por voz.

Al iniciarse, lista los nombres de los Chatbots creados hasta el momento que tengan el fichero de errores, para que el usuario pueda escoger el que necesite mejorar.

Cuando el usuario selecciona un Chatbot, el sistema de mejora y entrenamiento carga todas las intenciones y la lista de errores, guardándolas en las estructuras correspondientes para facilitar su manejo.

Por último, cuando el usuario decida que ya ha terminado, y haya al menos un fallo arreglado, el sistema de mejora y entrenamiento procederá a actualizar el JSON del Chatbot que se está resolviendo, y volverá a entrenar y generar el modelo para que los cambios se carguen. Esta edición del JSON se facilita gracias a la librería JSON de Python.

A continuación se mostrará una imagen donde se muestra todas las clases necesarias que el sistema de mejora y entrenamiento, Solve Error, necesita para poder llevar a cabo las acciones.

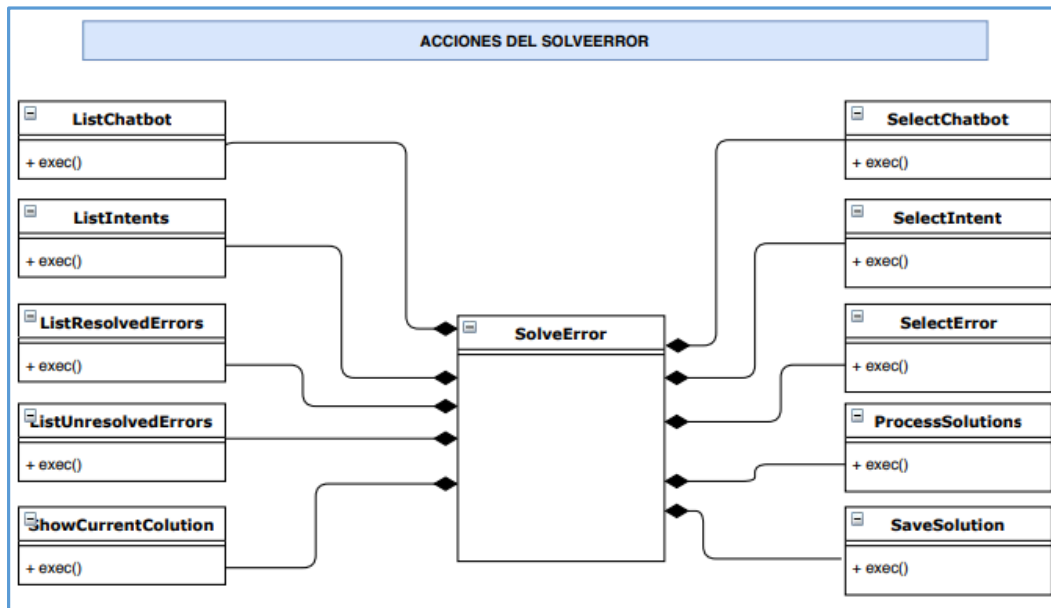


Ilustración 13: Acciones del Sistema de mejora y entrenamiento.

4.5.3 Pruebas

En este apartado, se mostrarán imágenes de la ejecución del sistema de mejora y entrenamiento.

Siguiendo el flujo de los anteriores ejemplos, se mostrará el funcionamiento con el error que presentó el Chatbot “Lista Compra”.

```

Using TensorFlow backend.
El modelo ya existe
Ejecutando el Predictor...
Predictor ejecutado correctamente.
Se está ejecutando el ChatBot "SolveError".
=> listar chatbots
['SolveError', 'compra online', 'Lista_Compra', 'Meta ChatBot']
=> seleccionar un chatbot
¿ Qué chatbot quiere resolver ?
=> Lista_Compra
Se ha seleccionado el ChatBot "Lista_Compra".
=> mostrar intenciones
['finishRunningChatbot', 'saveSentence', 'dontSaveSentence', 'añadirProducto', 'eliminarProducto']
=> mostrar errores
"guardar" que se asoció con el intent "saveSentence"
=>

```

Ilustración 14: Prueba Sist. Mejora y Entrenamiento - Selección de Chatbot a mejorar.

Como se puede ver, la petición “guardar” se le asoció correctamente a la intención “saveSentence” pero al no superar el umbral de clasificación se obtuvo un falso negativo.

A continuación se mostrarán los resultados de solucionar este fallo con el Solve Error.

```

=> seleccionar intención
¿ A qué intención lo quiere añadir ?
=> saveSentence
Se ha seleccionado la Intención "saveSentence".
=> seleccionar error
¿ Qué error quiere procesar ?
=> guardar
Se ha seleccionado el error "guardar".
=> guardar solución
La sentencia "guardar" se ha asociado a la Intención "saveSentence".
=> procesar solución
Generando el modelo para el ChatBot "Lista_Compra".
44 documents
9classes [añadirProducto, cancelarCompra, dontSaveSentence, eliminarProducto, finishRunningChatbot, listarProductos, modoDePago, realizarCompra, saveSentence]

```

Ilustración 15: Prueba Sist. Mejora y Entrenamiento - Procesar el error.

```

El modelo se ha generado correctamente
Se han resuelto los errores: guardar
=> mostrar errores
El ChatBot "Lista_Compra" no tiene errores.
=>

```

Ilustración 16: Prueba Sist. Mejora y Entrenamiento – Resultado de procesar el error (1)

Como muestran las imágenes, el Chatbot Solve Error tiene la capacidad de guardar todas las soluciones y permitir que el usuario siga asociando el fallo a una intención. Tras guardar todas las soluciones, el usuario indica que quiere procesar las soluciones, y es aquí cuando el Solve Error edita la estructura JSON del Chatbot volviendo a entrenar y generar el modelo. Tras procesar los errores, como sólo había uno por procesar ya no hay mas errores en el Chatbot de “Lista_Compra”.

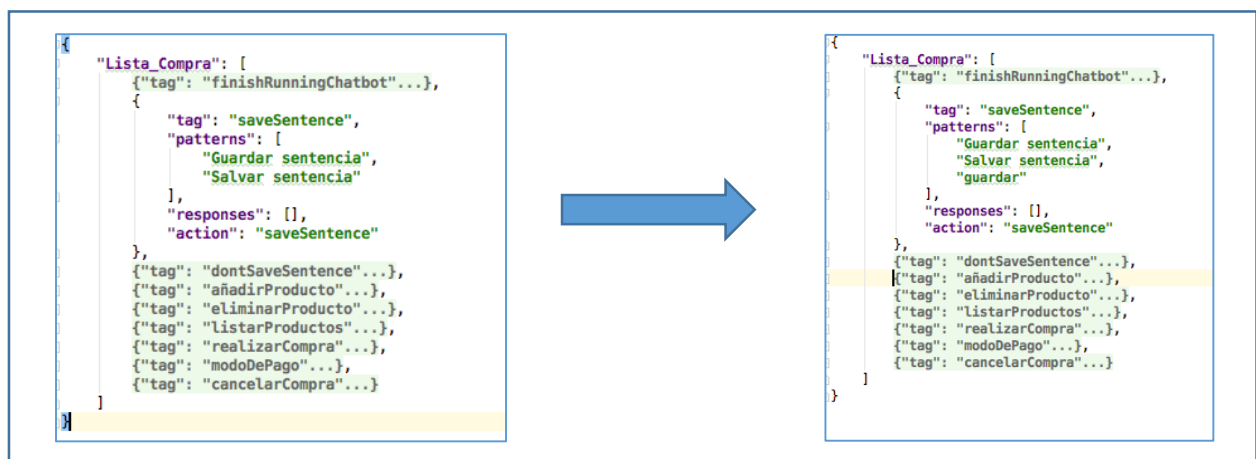


Ilustración 17: Prueba Sist. Mejora y Entrenamiento – Resultado de procesar el error (2)

El resultado final es que se ha modificado la estructura JSON del Chatbot que se ha mejorado a través de una herramienta que también permite realizar esta acción con la voz, haciéndola más interactiva.

4.6 RESUMEN DEL CAPÍTULO

De lo explicado a lo largo de este capítulo se concluye que el desarrollo del entorno se divide en 3 subsistemas. Un motor de Chatbots que se encarga del procesamiento del lenguaje natural usando técnicas de Machine Learning, aprendizaje supervisado, para entrenar y generar el modelo. El Metachatbot que hace uso de este motor para reconocer las peticiones de los usuarios, pero también controla la estructura de los Chatbots para su creación. Y por último, el sistema de mejora y entrenamiento que es un Chatbot de ayuda para mejorar aquellas peticiones que no hayan sido reconocidas.

Con estos tres subsistemas se obtiene un entorno capaz de realizar tareas que otros entornos no ofrecen, como es la capacidad que tienen todos los Chatbots de guardar los errores para su posterior mejora o el poder realizar una interacción con la voz a la hora de crear Chatbots. Otra característica importante es permitir programar las acciones que los Chatbots realizarán, dando libertad a los desarrolladores.

CAPÍTULO 5: EVALUACIÓN CUALITATIVA

En este capítulo se evaluará la forma, medios y los resultados de todo el trabajo que se ha realizado para conseguir los objetivos.

Para ello se mostrará la tabla de comparaciones añadiendo el entorno diseñado en este trabajo para ver lo que ofrece frente a los otros entornos y Chatbots. En las columnas con el color verde se indican las características para los Chatbots y con el color azul las características de los entornos de creación de Chatbots.

	Prog. acciones Cbot.	Inter. voz	Sist. mejora y entr. por voz	Ejec. en distintas app./sist.	E/S voz	Guardar fallos	Mensaje múltip.
IBM Watson Conversation	O			O	O		
Microsoft Azure Bot Service	O			O	O		
Google API.ai	O			O	O		
Amazon Lex	O			O	O		
Facebook Messenger API	O			O	O		
Chatfuel							O
Pandorabots				O	O		O
APPSUMO Mobile- Monkey				O			O
Metachatbot	O	O	O		O	O	

Tabla 2: Comparación entre Metachatbot, otros entornos y Chatbots

Con lo explicado en anteriores capítulos, se concluye que el resultado del desarrollo del entorno de creación de Chatbots, Metachatbot, es un entorno versátil y proporciona características que otros entornos no proporcionan tal y como lo refleja la anterior tabla.

En comparación con los otros entornos, el Metachatbot es capaz de tener una conversación por voz con el usuario, ofrecer un medio de mejora y entrenamiento por voz; y en cuando a los Chatbots resultantes del entorno implementado en este trabajo, lo que lo diferencia de los demás es que permite guardar los fallos para luego utilizarse con el sistema de mejora y entrenamiento.

La forma en la que se realizó este proyecto, fue siguiendo un esquema desde el inicio en el que se planteaba tareas con dificultades incrementales. Empezando por averiguar todo lo posible sobre los Chatbots, seguido de las herramientas necesarias para empezar a programar a programarlos.

Una vez se obtuvo lo necesario para realizar un Chatbot se construyó uno básico, “Lista Compra”, con el cual se puede experimentar y ver las necesidades que pueda llegar a tener un entorno a la hora de crearlos. Con esto aprendido, el siguiente paso fue programar el motor de Chatbots para que pueda ser usado por cualquier Chatbot.

Tras conseguir el motor de Chatbots se empezó a desarrollar el Metachatbot. Este punto fue uno de los más complicados ya que tenía que poder ofrecer diversas características como la interacción por voz, ejecutar los Chatbots, generar los ficheros, cargar todos los Chatbots en las clases, etc. Cuando se consiguió este punto, el desarrollo del sistema de mejora y entrenamiento fue más sencillo, ya que era similar al Metachatbot.

Con todo esto desarrollado, se pasó mejorar la interacción que tienen los Chatbots. Esta interacción se realizó modularmente para poder introducir cualquier tipo de interacción, comunicación, entrada o salida y que los Chatbots puedan ejecutarlos sin problemas.

Siguiendo todas estas etapas de trabajo, se pudo conseguir un entorno completo con características competentes frente a otros entornos.

CAPÍTULO 6: RECURSOS EMPLEADOS

En este capítulo se detallará los materiales usados, ya sea a nivel de software o hardware, explicando el por qué de su uso frente a otras opciones.

También se planteará unas tablas de costes de implementación del sistema.

6.1 MATERIALES

Dentro de los materiales usados para el desarrollo de este sistema, se puede diferenciar entre materiales software y hardware.

MATERIALES SOFTWARE

Los recursos software usados son:

1. Python: se ha utilizado este lenguaje de programación, ya que es donde se encuentran la mayor parte de librerías para la programación con Inteligencia Artificial. Por otro lado, según numerosos post en *es.quora.com*, resulta más fácil el programar con este lenguaje, ya que es un lenguaje no tipificado, por la legibilidad de código, etc.
2. PyCharm: este entorno de programación se escogió porque presenta una interfaz amigable y porque ya se tenía conocimiento del mismo.
3. Anaconda 3: Instalador de componentes para Python.
4. GitHub (web): Se ha utilizado este repositorio de versiones porque ofrece herramientas para facilitar su uso. Además de aportar una gran comunidad para cualquier duda.
5. GitHub Desktop: Aplicación de escritorio para la actualización del repositorio.
6. OS HighSierra: Sistema operativo con el que se realizó el proyecto.
7. Keras: Esta es una librería de Python para utilizar las redes neuronales. Se optó por ella en vez de TensorFlow por la incompatibilidad de esta última con la versión de Python.
8. NLTK: Esta herramienta sirve para reconocer el lenguaje natural, consiguiendo los conjuntos de entradas y salidas necesarias para el entrenamiento de las neuronas. Aplicando la técnica de aprendizaje supervisado.

9. JSON: Este formato de texto se ha utilizado para la estructura de los Chatbots, ya que es más ligera que XML.

MATERIALES HARDWARE

Durante todo el proceso se ha contado con los siguientes componentes hardware:

1. Ordenador portátil: Para la programación del sistema. Preferiblemente con más de 8GB de RAM.
2. Teléfono móvil: Para la prueba de interacción por voz.
3. Memoria RAM (8GB): Se aumentó la memoria del ordenador porque no tenía la suficiente para trabajar con muchas aplicaciones a la vez y soportar las características de la versión actualizada del sistema operativo.
4. Disco de estado sólido (500GB): Se tuvo que cambiar el disco duro por uno de estado sólido porque estaba dañado.

6.2 TABLAS DE COSTES

A continuación se detallarán los costes económicos y temporales que han sido invertidos en el desarrollo del proyecto.

En la siguiente tabla se aprecia los gastos monetarios realizados para la obtención de los materiales usados para el sistema.

Material	Precio en Euros
Memoria RAM de 8GB x 1	55,45 €
Disco de estado sólido 500GB x 1	92,99 €
Ordenador portátil x1	0 €
Teléfono móvil x1	0 €
Adquisición de todo el material software	0 €
TOTAL	148,44 €

Tabla 3: Tabla de gastos monetarios.

En la siguiente tabla se mostrará el coste temporal que ha requerido el proyecto. Mostrando las horas totales mínimas y máximas por cada mes.

Mes	Horas mínimas (total)	Horas máximas (total)
Octubre 2017	36	38
Noviembre 2017	35	36
Marzo 2018	32	33
Abril 2018	30	32
Mayo 2018	27	29
Agosto 2018	38	40
Septiembre 2018	41	43
Octubre 2018	46	48
TOTAL	285	299

Tabla 4: Tabla de costes temporales por mes.

Con el objetivo de proporcionar el coste total del proyecto en la unidad monetaria, se convertirán las horas trabajadas a euros multiplicándolos por un valor de 38 € la hora.

La siguiente tabla mostrará el coste total del proyecto, sumando los gastos monetarios y los costes temporales.

		Coste temporal mínimo	Coste temporal máximo
		10.830,00 €	11.362,00 €
Gastos monetarios	148,44 €	10.978,44 €	11.510,44 €

Tabla 5: Costes finales del proyecto.

Las celdas amarillas de esta tabla muestran el resultado de sumar los costes temporales acumulados (mínimos y máximo) con los gastos monetarios.

CAPÍTULO 7: TRABAJO FUTURO

En este capítulo se dará una visión de las mejoras que se pueden dar en este trabajo para mejorar sus funcionalidades. Pudiendo así, reforzar los puntos débiles que presenta para obtener un sistemas robusto.

Como se ha podido observar, este sistema aún no está completo. Presenta puntos que necesitan ser implementados y otros por mejorar. Algunas ideas para su mejora son:

1. Programar una aplicación móvil que use este sistema para aumentar la facilidad de uso.
2. Si se quiere enfocar a una aplicación web, el desarrollo de la parte FrontEnd de la misma ayudaría a que la aplicación sea más fácil de entender.
3. Además, la programación de un nuevo Chatbot de ayuda sería ideal para aquellas personas que no entendiesen mucho del tema, lo que haría llegar este sistema a muchas más personas sin que éstas tuviesen mucho conocimiento del tema.
4. También se podría incluir un Chatbot “escritorio”, que permitiese al usuario poder seleccionar el Chatbot que quiera ejecutar.

Siguiendo el símil del escritorio, el nuevo Chatbot sería el escritorio de un ordenador cualquiera y los demás Chatbots (generales o creados por el Metachatbot) sería las aplicaciones que el usuario podrá ejecutar seleccionándolas.

5. Tocando la parte funcional interna, se podría cambiar las acciones por otros Chatbots para conseguir encapsular aún más la funcionalidad. Por lo que se conseguiría simular la contextualización del lenguaje.
6. Mejorar las interacciones con el sistema.
7. Permitir al usuario seleccionar dinámicamente el modo de interacción para acercar el sistema a la usabilidad universal.
8. Permitir al usuario indicar la ruta donde desea guardar la estructura del Chatbots que está creando y, además darle la opción de introducir una ruta para que el Metachatbot cargue todos los Chatbots que se encuentren en esta ruta.

9. Dar la facilidad al usuario para que pueda introducir el código de las acciones de sus Chatbots.

Este proyecto se puede enfocar en muchas áreas, si se llega a desarrollar más puede llegar a ser una herramienta competitiva entre los demás entornos para crear Chatbots.

CAPÍTULO 8: CONCLUSIONES

Este proyecto se planteó para desarrollar un sistema capaz de crear Chatbots.

El sistema creado permite que el usuario configure interacción por texto o voz. La estructura diseñada en la que se basan todos los Chatbots es usado en este sistema. El Metachatbot se encarga de procesar esta estructura para generar todos los ficheros necesarios automáticamente y de testear, es decir, ejecutar un Chatbot que se esté creando o ya creado previamente.

Además, se ha implementado un Chatbot, sistema de mejora y entrenamiento, con el que se puede mejorar cualquier Chatbot del sistema. Tras esta mejora, el sistema de mejora y entrenamiento vuelve a entrenar y generar el modelo del Chatbot que se esté mejorando.

Para aquellos usuarios que tengan más conocimiento de programación, podrán programar directamente sus acciones para añadirlos al sistema para su uso.

Con esta implementación se ha podido satisfacer la finalidad del proyecto, cumpliendo con todos los objetivos planteados.

Este trabajo puede ayudar a crear distintos Chatbots que puede comunicarse con otros sistemas, aunque limitados, para aumentar su funcionalidad.

Los beneficios frente a otros entornos de desarrollo de Chatbots es que presenta una interacción por voz, lo que lo hace más cómodo a la hora de usarlo. También se puede programar las acción a ejecutar. Esta característica es lo que beneficia a los desarrolladores para poder usar los Chatbots creados para comunicarse con otros sistemas. Aunque el uso no es para todos los usuarios, con futuras mejoras se puede conseguir este objetivo.

Como valoración final, aunque ya existan programas propios de empresas de renombre que permiten hacer lo que se planteó con este proyecto, no se ha podido encontrar uno que permita crear cualquier tipo de Chatbot utilizando la voz.

Sin duda, de lo aprendido con este proyecto, lo destacado es el desarrollo del sistema de forma modular para poder realizar cambios/mejoras de forma más fácil como la interacción por voz que se logró a mitad del proyecto.

Otro de los puntos a tener en cuenta, es que este proyecto está planteando cambiar el modo de uso de software como muchas otras aplicaciones que usan sus sistemas por voz, asistentes personales.

Primero fue la interacción por consola, luego se implementó interfaces de usuario y se usó el ratón, ahora se está intentando implantar la idea de que el software se puede usar con la voz.

GLOSARIO

A

Acciones

Son las tareas a realizar por el Chatbot tras identificar la(s) intención(es) del usuario.

C

Chatbots

Un chatbot es un programa informático con el que es posible mantener una conversación, tanto si queremos pedirle algún tipo de información o que lleve a cabo una acción.

Computación cognitiva

Sistemas de autoaprendizaje que utilizan la minería de datos, el reconocimiento de patrones y el procesamiento del lenguaje natural para imitar el funcionamiento del cerebro humano.

H

Hitos

Los hitos son una serie de etapas dentro de un mismo proyecto. Se determinan desde la planificación previa del mismo, se van revisando a medida que avanza nuestro trabajo y se pueden ir modificando según las necesidades del proyecto o cliente.

I

Inteligencia Artificial

Es la inteligencia exhibida por máquinas. En ciencias de la computación, una máquina "inteligente" ideal es un agente racional flexible que percibe su entorno y lleva a cabo acciones que maximicen sus posibilidades de éxito en algún objetivo o tarea.

Intenciones

Es la idea que el usuario le comunica al Chatbot para que lleve a cabo la acción correspondiente.

K

KERAS

Es una biblioteca de red neuronal de código abierto escrita en Python. Es capaz de ejecutarse sobre TensorFlow , Microsoft Cognitive Toolkit o Theano.

M

Machine Learning

Es una rama de la inteligencia artificial, cuyo objetivo es desarrollar técnicas que permitan que los ordenadores aprendan.

Motor de Chatbots

Es el programa capaz de procesar la arquitectura definida para un chatbot y ejecutar las diferentes tareas definidas para que el chatbot funcione correctamente.

P

Procesamiento Natural del Lenguaje

Es un campo de las ciencias de la computación, inteligencia artificial y lingüística que estudia las interacciones entre las computadoras y el lenguaje humano.....

Python

Es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

R

Redes neuronales

Una red neuronal artificial es un grupo interconectado de nodos similar a la vasta red de neuronas en un cerebro biológico.....

S

SaaS

Es un modelo de distribución de software donde el soporte lógico y los datos que maneja se alojan en servidores de una compañía de tecnologías de información y comunicación (TIC), a los que se accede vía Internet desde un cliente.

Sacos de palabras

El modelo de bolsa de palabras o saco de palabras es una representación simplificada utilizada en el procesamiento del lenguaje natural y la recuperación de información.

T

Thread

Es una secuencia de tareas encadenadas muy pequeña que puede ser ejecutada por un sistema operativo.

X

XML-RPC

Es un protocolo de llamada a procedimiento remoto que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes.

BIBLIOGRAFÍA

botanalytics. *Voice Enabled Chatbots Vs. Messenger Bots: Everything You Need To Know*. 21 de 02 de 2018. <https://botanalytics.co/blog/2018/02/21/voice-chatbots-vs-messenger-bots/>.

Contieri, Maximiliano. *planetachatbot*. 30 de julio de 2017. <https://planetachatbot.com/las-diez-caracter%C3%ADsticas-que-todo-chatbot-decente-debe-tener-4f991a09c491>.

Enesimus. *Trabajando con JSON - Aprende sobre desarrollo web | MDN*. 11 de 11 de 2017. <https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON>.

Fumo, David. *Types of Machine Learning Algorithms You Should Know*. 15 de 06 de 2017. <https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861>.

Latorre, Maximiliano. *La revolución de los bots*. 08 de 10 de 2016. <https://www.elobservador.com.uy/nota/la-revolucion-de-los-bots-2016108500>.

Planeta Chatbot. *Introducción al mundo chatbot*. 19 de 05 de 2017. <https://planetachatbot.com/introducci%C3%B3n-al-mundo-chatbot-bddd0167da41>.

pulsosocial. *Las cinco mejores 'chat apps' para crear un chatbot*. 26 de 07 de 2017. <https://pulsosocial.com/2017/07/26/las-cinco-mejores-chat-apps-crear-chatbot/>.

Rouse, Margaret. *¿Qué es Chatbot?* 15 de Mayo de 2018. <https://searchdatacenter.techtarget.com/es/definicion/Chatbot>.

Vasant, Pandian. *Handbook of Research on Artificial Intelligence Techniques and Algorithms*. Perak: Idea Group,U.S., United States, 2015.

Wikipedia. *Aprendizaje automático*. 7 de 08 de 2018. https://es.wikipedia.org/wiki/Aprendizaje_autom%C3%A1tico.

Bird, Steven, Edward Loper and Ewan Klein (2009), *Natural Language Processing with Python*. O'Reilly Media Inc.

elEconomista. «¿Qué son exactamente los chatbots y para qué sirven?» elEconomista, 13 de Abril de 2016: 1.

gk_. «Contextual Chatbots with Tensorflow.» chatbotmagazine, 2017: 1. <https://chatbotmagazine.com/contextual-chat-bots-with-tensorflow-4391749d0077>

HAI-SALEH, ALBERTO. «Qué son exactamente los 'bots' y cómo funcionan.» GQ, 2017. <https://www.revistagq.com/noticias/tecnologia/articulos/que-son-exactamente-los-bots-y-como-funcionan/25633>

Wikipedia. «Artificial Linguistic Internet Computer Entity.» 2018. https://es.wikipedia.org/wiki/Artificial_Linguistic_Internet_Computer_Entity

Wikipedia. «Bot Conversacional.» 1. 2018. https://es.wikipedia.org/wiki/Bot_conversacional

Dimitry Kagan. «tutorial como construir un chatbot con Facebook messenger.» 8 de junio de 2017. <https://planetachatbot.com/tutorial-como-construir-un-chatbot-con-facebook-messenger-de474ee93f92>

