

Recuperación Practicas Laboratorio

RESUMEN

Este proyecto trata sobre la realización de un bot con capacidad de conversación sobre preguntas planteadas a cerca de la Escuela Superior de Ingeniería Informática (ESEI) y la Universidad de Vigo. También puede realizar diversos servicios relacionados con dichas instituciones, con variaciones de preguntas, siendo capaz de aprender preguntas que no estén su base de conocimiento.

Para esto se hará uso de AIML 2.0 y Jason, así como archivos .java suplementarios (envío de correos...).

INTRODUCCIÓN

Los **chatbots** (o agentes conversacionales) son programas que pueden relacionarse con el usuario respondiendo preguntas que este realiza y utilizando el lenguaje natural. Pueden así mantener una conversación por medio de entradas de texto.

Su funcionamiento sigue el siguiente patrón:

1. El usuario pregunta por medio de la entrada de texto
2. La pregunta es analizada por el agente y por su inteligencia artificial
3. El agente responde en tiempo real

Estos agentes no son algo novedosos, ya que se implementan desde hace años, con varios ejemplos notables:

- **ELIZA** [1] (1964): el primer bot, desarrollado por Joshep Weizenbaum para IBM, era capaz de mantener una conversación en inglés.
- **CHATTERBOT** [2] [3] (1989): en el juego TinyMud se podía mantener una conversación multiusuario y escenarios simulados.
- **ALICE** (1995): es un bot inspirado en ELIZA. Utilizaba patrones para la conversación y lenguaje AIML.
- **SIRI** [4] (2011): primer asistente virtual para un teléfono Apple. Este bot es capaz de responder a preguntas sobre el clima, cálculos, música, y también utiliza Google para preguntas sobre servicios de web. Siri aumenta su conocimiento cuantas más personas la utilizan gracias al llamado Machine Learning.
- **Alexa** [5] (2014): asistente creado por Amazon, permite conocer información sobre productos, climas, curiosidades, videos o llamadas.

Para nuestro bot utilizamos el lenguaje **AIML** (Artificial Intelligence Markup Language), que es un lenguaje de programación basado en XML para responder preguntas. Fue diseñado para ALICE, y se basa en la sintaxis de **patrones** y **plantillas**; siendo los patrones lo que el usuario pregunta y la plantilla es lo que define las respuestas del bot.

APLICACION

De una forma genérica, podemos dividir nuestra aplicación en dos partes bien diferenciadas: AIML y Jason.

La primera es la que emplea el ya mencionado AIML 2.0, encargado de facilitar las respuestas al usuario en base a una serie de preguntas que este realiza. La segunda son los dos agentes de Jason (student y master) cuya labor será hacer simulaciones de conversaciones, así como diversas funciones, ya sea envío de correos, creación de ficheros (sets), filtro de respuestas...

AIML [6]

Dentro del proyecto creado, encontramos varios archivos con extensión. aiml, que serán los encargados de responder a las preguntas realizadas por el usuario.

Entre ellos encontramos:

- **anexo1.aiml**: en estos archivos encontramos las preguntas referentes a las cuestiones tq y ntq. Las primeras son las que figuran en el Anexo I tal cual están, mientras que las ntq son variantes de las anteriores, donde hemos identificado las palabras clave para contestar, ante similar pregunta, igual respuesta. Si alguna pregunta no se identifica correctamente, el bot preguntará y confirmará para asignar una respuesta a esa pregunta no identificada.
- **personales.aiml**: contiene preguntas del tipo aq. Estas son preguntas personales que el usuario hace a nuestro bot. En este archivo tenemos almacenadas un par de preguntas, pero lo ideal es que el bot identifique las que no estén. Si alguna pregunta no se identifica correctamente, el bot preguntará y confirmará para asignar una respuesta a esa pregunta no identificada.
- **saludos.aiml**: aquí es donde el bot interacciona con el usuario, respondiendo preguntas de saludos o del estado del bot.
- **servicios.aiml**: este archivo trata preguntas proporcionadas por el Anexo II, referentes a los servicios que puede solicitar el alumno al bot (cambio de grupo, alta en una signatura...). Para solicitar dichos servicios, el alumno debe identificarse mediante su nombre.

ASL [7]

Estos archivos son los referentes a los agentes: student y master. Estos dos archivos usan el archivo Chatter.java como un entorno de comunicación con nuestro bot:

- **student.asl**: tendrá como objetivo principal realizar preguntas al bot. El agente realiza cinco preguntas aleatorias referentes al Anexo I (tq y ntq) y otras cinco preguntas aleatorias referentes a las preguntas tipo aq. Además, debe solicitar los servicios al bot. Para realizar estos servicios el agente debe identificarse antes, si no se le pedirá identificación.
- **master.asl**: este agente realiza las acciones filtrando y parseando las etiquetas que proporciona el bot.
Dicho es el encargado de enviar los correos de servicio del bot (mailing.java) esto se hace mediante el envío de un correo electrónico a la cuenta asociada al bot (la cuenta debe tener el acceso de aplicaciones poco seguras activado). Puede tardar un cierto tiempo en mandar estos correos. También puede crear archivos con información a las carpetas set mediante la etiqueta <addset> y maps a la carpeta maps, mediante la etiqueta <addmap>. Esta información se suministra con el comando .send.

CONCLUSIÓN

Tanto el lenguaje de AIML como las APIs Java que se pueden implementar junto a los agentes ASL, hacen el trabajo más eficiente, pues son de gran ayuda y su conocimiento, aunque pueda parecer costoso al principio, luego se va asimilando gracias a su sintaxis y uno se puede familiarizar con estos dos lenguajes de manera fácil. También ha ayudado mucho a realizar las pruebas que nos dieran una plantilla de los agentes ASL.

Respecto a las conclusiones personales, para conocer cómo funciona Jason nos ha llevado más tiempo de lo esperado, pero una vez conocidos se hace muy fácil; aunque no tanto como AIML, ya que con el uso de etiquetas se ve muy sencillo. Son dos lenguajes interesantes, nuevos y que son de gran utilidad si quieres escapar de los lenguajes de programación con estructuras definidas que se estudian durante tantos años en la carrera (C, C++, Java...). Aunque sin duda la mayor dificultad ha sido compaginar la realización de esta práctica con la situación de pandemia, ya que si había alguna duda debía responderse por correo y esto ralentizaba el avance.

Con todo esto, acabamos esta práctica con un dominio claro de AIML sobre Jason, pero conociendo ambos lenguajes.

REFERENCIAS

[1] Así era ELIZA, el primer bot conversacional de la historia. Recuperado de <https://www.xataka.com/historia-tecnologica/asi-era-eliza-el-primer-bot-conversacional-de-la-historia>

[2] ChatterBot. Recuperado de <https://chatterbot.readthedocs.io/en/stable/>

[3] Daniel Cerdas Méndez, (2017). Evolución de los Chatbots. Recuperado de <https://planetachatbot.com/evoluci%C3%B3n-de-los-chatbots-48ff7d670201>

[4] ¿Qué es Alexa, Google Home y Siri? Recuperado de <https://revistaderobots.com/inteligencia-artificial/que-es-alexa-google-home-y-siri/>

[5] El lenguaje AIML, apuntes sobre AIML2.X. Recuperado de <https://fatic.uvigo.es/index.php/gl/>

[6] ¡Documentación AIML, BOTlibre! botlibre.com. Recuperado de <http://es.botlibre.com/manual-aiml.jsp>

[7] Jason: a Java-based interpreter for an extended versión of AgentSpeal. Recuperado de <http://jason.sourceforge.net/wp/>



Yago Pérez García

ypgarcia@esei.uvigo.es

Estudiante de Ingeniería Informática en la Uvigo 2014-2020

(Campus de Ourense)

Título B2 de inglés



Oscar Lista Rivera

olrivera@esei.uvigo.es, Rúa de Rairo Nº 87 1º D

Estudiante de Ingeniería Informática en la Universidad de Vigo
2014-2020 (ESEI)

Operaciones de mantenimiento de vehículos