

ChatterBot

La biblioteca ChatterBot de Python está diseñada para responder de manera automatizada a las entradas que de él usuario, con la ayuda de varios algoritmos de aprendizaje automático que generan una variedad de respuestas, lo que permite que al desarrollar chatbots se puedan dar respuestas apropiadas en diferentes contextos. El chatbot también va a poder mejorar su rendimiento conforme aprende con el paso del tiempo, presenta independencia de lenguaje, de esta manera se puede entrenar para hablar en cualquier idioma.

Funcionamiento:

Un chatbot no entrenado de ChatterBot inicia sin conocimiento de cómo debe comunicarse con el usuario, se pueden introducir frases y respuestas para que el chatbot aprenda qué debe responder en cada ocasión, cada vez que el usuario ingresa una entrada el chatbot guarda las entradas y respuestas con las que va tratando y con esos mismos datos genera respuestas automatizadas relevantes, cuando recibe una nueva entrada, compara la nueva entrada con datos anteriores, así el chatbot puede seleccionar una respuesta que esté vinculada a la entrada conocida más cercana posible.

Capacidades:

Como se mencionó anteriormente, algunas de sus capacidades son el **aprendizaje automático** así como la **independencia de lenguaje**. Entre otras de sus capacidades están:

Almacenamiento y consulta de datos: ChatterBot cuenta con módulos llamados *storage adapters* los cuales permiten interactuar con el sistema de almacenamiento y proporcionan una abstracción, en su versión 1.0.8 tiene los siguientes *storage adapters*; *mongodb*, *sql_storage*, *django_storage*. De esta manera permite mantener un registro de las interacciones y facilita la gestión de datos.

Procesamiento de lenguaje natural(NLP siglas en ingles): ChatterBot utiliza técnicas de procesamiento de lenguaje natural para poder entender y responder a los mensajes de los usuarios, de esta manera analiza las preguntas y comentarios de los usuarios para generar respuestas relevantes y coherentes.

Por último está la **personalización** que permite al desarrollador personalizar las respuestas y entrenar al chatbot con datos específicos para que responda de manera más precisa y personalizada.

Limitaciones:

Falta de inteligencia emocional: Los chatbots presentan dificultad para comprender las emociones humanas, sarcasmo... Lo cual puede representar un problema a la hora de interpretar el objetivo de las preguntas.

Dependencia de los datos de entrenamiento: La calidad y variedad de los datos de entrenamiento de ChatterBot influyen en gran medida con la calidad de las respuestas que genere. Los datos sesgados o incompletos pueden dar lugar a respuestas incorrectas o sesgos involuntarios en el comportamiento del chatbot. Por lo cual depende mucho de un entrenamiento adecuado para que brinde respuestas más precisas.

Incapacidad para afrontar escenarios inesperados:

Al igual que otros chatbots, ChatterBot funciona siguiendo reglas y patrones predefinidos, cuando se enfrenta a situaciones que no se espera, lo más probable es que no sepa cómo responder de manera

adecuada, de respuestas fuera de contexto e incluso que responda con información poco precisa, por lo cual la base de datos del ChatterBot requiere actualizarse de manera frecuente

Limitaciones en el contexto: Dado que ChatterBot en su versión 1.0.8, solo trabaja con el contexto de la entrada anterior esto puede ser un problema para seguir el hilo de una conversación por ejemplo.

Fuentes:

<https://www.codemotion.com/magazine/ai-ml/develop-chatbot-with-python-and-chatterbot/>

<https://chatterbot.readthedocs.io/en/stable/>

<https://openaccess.uoc.edu/bitstream/10609/132727/9/mafonsoTFG0621memoria.pdf>

<https://www.datacamp.com/tutorial/building-a-chatbot-using-chatterbot>

<https://medium.com/@kiran2002sk/benefits-and-limitations-of-using-ai-chatbots-7eed242b7a5f>

<https://github.com/gunthercox/ChatterBot/issues/812>