

SAMI: Un Sofbot de Charla Desarrollado con Técnicas de Razonamiento basado en Casos

A Chatbot Developer using the Technique Case Based Reasoning

Shanir Majayra Camacho Jara¹

Resumen El presente artículo se encuentra dentro del área de la Inteligencia Artificial y explica la manera en que se construyó el sofbot de charla llamado SAMI (Sistema Asistente Modelo Inteligente), para lo cual se utilizó la técnica de Razonamiento Basado en Casos (RBC). El contexto que se ha elegido para las conversaciones de SAMI, son los temas relacionados al Aula Virtual de la Universidad Nacional Mayor de San Marcos (UNMSM), de esta forma el sofbot actúa como un "asistente virtual" brindando ayuda a los usuarios. Esta interacción se realiza en lenguaje natural, mediante un diálogo escrito entre el usuario y el sistema, de tal forma que la persona piensa que está hablando con otra persona y no con un sofbot. Internamente el sofbot analiza el diálogo del humano y busca dentro de su base de conocimientos la respuesta más adecuada posible utilizando la técnica RBC. La importancia de este trabajo también radica en los diferentes usos que se le pueden dar a los sofbots de charla, ya que podrían ser usados en áreas como educación (e - learning), entretenimiento, interfaces hombre-máquina, soporte en las áreas CRM, entrenamiento, ambientes de aprendizaje, etc. Por lo tanto vemos que muchas organizaciones pueden obtener grandes ventajas implementando este tipo de aplicaciones, es por ello que su estudio y difusión es de gran importancia.

Palabras clave: inteligencia artificial, razonamiento basado en casos (RBC), sofbot de charla, aula virtual, base de conocimientos.

Abstract The present article is inside the area of the Artificial Intelligence and explains the way in that the chatbot called SAMI (Intelligent Model Assisting System) was built, for this purpose the technique Cases Based Reasoning (CBR) was used. The context that has been chosen for SAMI's conversations were the topics related to the Virtual Classroom of the UNMSM, thus the chatbot acts as a virtual assistant offering help to the users. This interaction is carried out in natural language, by means of a dialogue written between the user and the system, in such a way that the person thinks that he is speaking another person and not a chatbot. Internally the chatbot analyzes the human's dialogue and search inside its knowledge base the possible more appropriate answer using the technical CBR. The importance of this work also resides in the different uses that can be given to the chatbots, they could be used in areas like education (e-learning), entertainment, interfaces man-machine, CRM, training, etc. Therefore, many organizations can obtain great advantages implementing this type of applications, so the investigation and diffusion is of great importance.

Key words: artificial intelligence, cases based reasoning (CBR), chatbot, virtual classroom, knowledge base.

¹ SAMI: un sofbot de charla desarrollado con la técnica RBC.

1. Introducción

Un área particularmente prometedora de la computación lo constituye la Inteligencia Artificial (IA) [8] que ha desarrollado técnicas de aplicación en ramas específicas que la han transformado de ciencia puramente académica en ciencia experimental. Dentro de las técnicas de IA, el Razonamiento Basado en Casos (RBC) ocupa un importante lugar.

El Razonamiento Basado en Casos (RBC) se preocupa por el estudio de los mecanismos mentales necesarios para repetir lo que se ha hecho o vivido con anterioridad, ya sea por uno mismo, o ya sea por casos concretos recopilados en la bibliografía o en la sabiduría popular [6].

Cabe mencionar que en el presente trabajo no se intentaron solucionar todos los problemas relacionados al lenguaje natural, que a lo largo de la historia se han intentado solucionar y todavía con resultados no muy satisfactorios. Por lo tanto se puede resumir que lo que se buscó en este trabajo fue construir un softbot de charla que converse en el idioma español, y lo haga por lo menos en un dominio concreto.

2. Razonamiento Basado en Casos (RBC)

El pensamiento del ser humano es generalmente una mezcla de varios tipos de procesos como inducciones, deducciones, etc. Este tipo de razonamiento humano está basado en las analogías, ya que por ellas se hacen inferencias que permiten adquirir nuevos conocimientos. Un ejemplo de este razonamiento se presenta a continuación: *“Un médico posee conocimientos como tipos de enfermedades, síntomas y tratamientos, gracias a sus estudios y sobretodo a la experiencia acumulada durante sus años de ejercicio. Cuando un paciente le comenta acerca de su enfermedad, el médico observa los síntomas y basándose en las enfermedades y síntomas que conoce, encuentra el tipo de enfermedad que el paciente está sufriendo y le ofrece una solución. Si los síntomas que sufre el paciente no corresponden a ninguna enfermedad que el médico conozca, este adapta los casos más similares para crear una solución. La nueva información sobre la enfermedad, síntomas, solución y resultado, quedarán en la memoria del médico para ser aplicadas en un futuro.”*

Los sistemas de Razonamiento Basado en Casos (RBC) realizan el mismo proceso tratado en el ejemplo anterior, con la ventaja de que pueden procesar más información de distintas fuentes y a mayor velocidad. La resolución de problemas en

RBC se realiza de la siguiente forma: ante la descripción de un problema, se recupera un caso previo parecido al problema actual, se adapta la solución del caso resuelto para obtener la solución del caso actual, *se revisa la solución y se aprende* el nuevo caso junto con la solución revisada. Son cuatro los procesos básicos en los que se apoya este modelo que se muestra en la Figura 1 [1]:

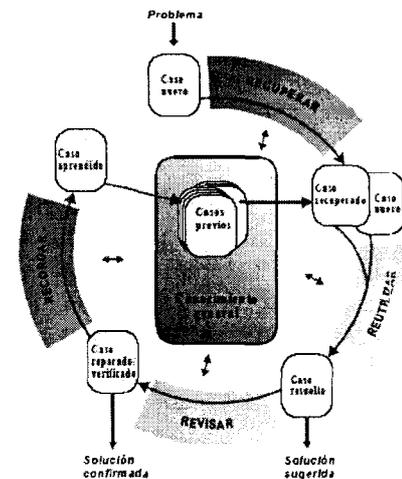


Figura 1. Ciclo del RBC

Entre las ventajas de los sistemas RBC, se pueden mencionar:

- La adquisición del conocimiento. Ya que la unidad básica del conocimiento es el caso y no la regla. Los seres humanos por lo general articulan su conocimiento mediante ejemplos de problemas y soluciones anteriores, más que por medio de reglas específicas y abstractas.
- Forma de aprendizaje. A diferencia de los sistemas basados en reglas convencionales, si la misma situación se presenta repetidamente, no se tiene que construir o generar la misma solución a partir de cero cada vez que se presente el mismo problema.
- El aprendizaje es simple ya que no requiere una comprensión profunda del dominio. Mediante el RBC se pueden proponer soluciones en dominios que no se comprenden del todo.
- Es posible evaluar las soluciones cuando no existe ningún método algorítmico para hacerlo.
- Resulta más sencillo adquirir nuevos casos que descubrir reglas y generalizaciones nuevas.
- Proporciona mejoras de rendimiento: razonamiento más rápido que desde cero, capacidad de anticipar y evitar errores pasados, capacidad de centrarse primero en las partes más importantes de un problema.

Pero también esta técnica presenta algunas dificultades en su implementación como:

- El problema de la representación de los casos.
- Estructuración de relaciones entre casos.
- Bases de casos muy grandes son difíciles de administrar.
- Por otro lado no tiene un resolvidor de problemas, es decir, sin casos no se puede resolver problemas.

3. Sofbots de charla

El principio de todo esto está en el Test de Turing, lo que Turing llamó “el juego de la imitación” (The Imitation Game) y con lo que se pretendía responder a la pregunta de si “las máquinas pueden pensar”.

Otro de los pasos importantes lo dio Hugh Loebner en 1991, con la creación del premio de su mismo nombre, el Dr. Hugh Loebner patrocina cada año un concurso basado en la “Prueba de Turing”. El concurso otorga medallas y premios en efectivo para “la computadora más humana” y ha impulsado la creación de este tipo de sofbots.

Pero antes de esto nos encontramos con el primer sistema y pionero en el campo de los sofbots de charla, “Eliza”, que fue el primer programa realizado en esta área. Era un sofbot funcional para el ámbito de la psicología, el cual fue creado en 1980 por Joseph Weizenbaum [7] y consiguió que las personas le cuenten sus problemas igual que si estuvieran hablando con un psicólogo. Incluso las secretarías y personal no-técnico del MIT (Massachusetts Institute of Technology) pensaron que la máquina era una terapeuta real, y se gastaron horas revelando sus problemas personales al programa. ELIZA es aún el programa más ampliamente distribuido en la historia de la inteligencia artificial.

Con todo, Eliza, no ha sido sólo el primero de los sofbots de charla, sino que es el pilar fundamental para casi todos ellos, ya que se basan principalmente en la creación de patrones que simulen el comportamiento humano.

Hace unos años, en 1995, el Dr. Richard Wallace escribió ALICE. El desarrollo de este sofbot de charla empezó en su primera versión, usando SETL (lenguaje basado en lógica matemática y un conjunto de teorías). A partir de este primer intento se desarrolló lo que se denominó “Program A” que fue la primera versión de ALICE, usando AIML (Artificial Inteligente Markup Language) y Java.

Un tiempo después se desarrolló la siguiente versión, “Program B”, en el que intervinieron alrededor de 300 desarrolladores y donde el AIML evolucionó a una gramática procedente de XML, esto ocasionó el desarrollo de editores y herramientas para la gestión

de AIML, y fue con esta versión con la que ganó el Loebner Prize en 2000. Posteriormente se crearon otras dos versiones de ALICE, “Program C” (que se desarrolló usando C/C++) y “Program D” (basado en la tecnología Java2). A partir de aquí se creó, en 2001 “The A.L.I.C.E. AI Foundation” cuyos objetivos son entre otros la distribución, promoción, desarrollo y mantenimiento de Alice y de la tecnología AIML.

Otros sofbots que se han desarrollado son: “Theresa” (cuyos principales temas de conversación son la música y la mitología griega), “Mimic” (el cual aprende mientras se mantienen conversaciones con él), “Brian” (que ganó el tercer puesto del Loebner Prize, en 1998), “Sofbot Rock Critic” (confecciona críticas de discos y artistas que no existen), “E.L.V.I.S.” (Sofbot de charla que simula ser Elvis Presley), “Jlaip” (intenta replicar la personalidad del ex Beatle John Lennon).

En la siguiente tabla se puede ver algunos de estos trabajos con sus respectivos autores:

NOMBRE DEL SOFBOT	AUTOR	IDIOMA
Alice	Richard S Wallace y su equipo	Inglés
Shampage	Rich Waugh	Inglés
Eliza	Joseph Weizenbaum	Inglés
Parry	Kenneth Colby	Inglés
Fred	Robby Garner en Robitron Software Research	Inglés
Jabberwock	Juergen Pirner	Inglés
Yeti	Webolutions New Media	Inglés
AI Alex	John Precedo	Inglés
Elbot	Fred Roberts	Alemán

Tabla 1. Trabajos sobre Sofbots

Como se puede notar, la mayoría de los sofbots de charla están preparados para realizar una conversación en inglés. A pesar de la supremacía de los sofbots en este idioma, también cabe destacar algunos sofbots desarrollados con el objetivo de establecer una conversación en lengua castellana. Por

ejemplo: Dr. Abuse, el cual está basado en Eliza y es uno de los mejores softbots de charla escritos en español actualmente.

Muchos de estos softbots son, en principio, de carácter general, es decir no están especializados en ningún tema en particular, mientras que otros sí lo están en temas diversos como Elvis Presley, música, rock and roll, John Lenon, etc.

Los softbots de charla pueden ser creados como medios de diversión o como parte interactiva en juegos, servicios de información de Internet, guías de sitios web, agentes de e-commerce y más. A continuación se presentan algunas aplicaciones que se pueden construir usando softbots de charla:

a) Asistente de Formularios: Un softbot de charla como asistente interactivo de formularios puede actuar como complemento del formulario digital, invitando al visitante a su llenado. Puede responder cuestiones, ayudar a resolver errores y asegurarse de que se contesten todas las preguntas. El softbot también puede sustituir totalmente al formulario, "entrevistando" al visitante en un proceso del todo natural. La información se va recogiendo mediante una agradable conversación.

b) Agente de Ventas: Un softbot interactivo de ventas es un buen vendedor que ayuda al cliente a realizar sus compras en Internet. Se puede comportar como cualquier vendedor real, averiguar qué es lo que busca el cliente, sugerir diferentes productos, encauzar la venta hacia otros productos interesantes o de gama superior, contestar preguntas, mostrar los productos y ayudar al cliente a rellenar el formulario de pedido. También puede realizar un seguimiento de la venta y descubrir si el cliente ha quedado satisfecho, efectuando después sugerencias de otros productos.

c) Asistente de Navegación: Un asistente interactivo de navegación complementa las típicas acciones del ratón. En lugar de moverse por el sitio Web, señalando los lugares de su interés, el visitante le pide al softbot que le ayude, y en sus propias palabras. Tras un breve diálogo, el softbot sabrá exactamente lo que necesita el visitante y le ofrecerá una respuesta directa, mostrándole al mismo tiempo la página Web indicada.

d) Asistente de Chat: Los asistentes de chat mantienen conversaciones divertidas y entretenidas con los usuarios. Sus respuestas suelen estar cargadas de humor y sorpresas, para mantener el interés del usuario.

E) Contrincante de Juegos: Los softbots de charla también pueden ser muy útiles en programas de juegos, haciendo la parte del contrincante. Para el usuario será como si estuviera compitiendo con una persona real, mientras va jugando puede ir conversando con su contrincante.

f) Soporte en la enseñanza: También podemos construir el softbot, dándole el aspecto de tutor, almacenando en su base de conocimientos toda la información que debe proporcionar a sus alumnos cuando éstos le hagan preguntas, haciendo así la educación más interactiva y atractiva a los usuarios.

4. Estructura de la base de conocimiento de SAMI

En este apartado se explica la estructura de la base de conocimientos del softbot SAMI [2], construida de manera que permita ingresar, modificar, actualizar y recuperar el conocimiento de la forma más sencilla posible. Ya que, como en todo sistema RBC, la estructura de los casos y su almacenamiento en la base de conocimiento, es una de las partes más importantes.

El AIML (Artificial Intelligence Mark-up Language), es un lenguaje que está estructurado para permitir ingresar conocimiento al softbot de una manera más fácil. Fue desarrollado por el Dr. Richard S. Wallace y la comunidad del software libre para el softbot ALICE durante los años 1995-2000.

Este lenguaje está compuesto por una serie de etiquetas tipo XML que ayudan a organizar el conocimiento del softbot. El funcionamiento del AIML se basa en un modelo de impulso-respuesta (stimulus-response model, en inglés).

El conocimiento de SAMI se organiza en archivos AIML que contienen etiquetas con una estructura determinada: en la primera línea se encuentra un prólogo estándar XML y el contenido del archivo debe estar cerrado dentro de las etiquetas <aiml></aiml>. En AIML son principalmente las etiquetas <category> las que contienen el conocimiento, en las etiquetas <pattern> se encuentran los patrones de preguntas que el usuario ingresa y las respuestas del softbot están en las etiquetas <template>.

En resumen el conocimiento del softbot SAMI consta de unidades con la siguiente estructura:

```
<category>
  <pattern>
    <!-- modelo de pregunta del usuario-->
  </pattern>
  <template>
    <!-- reacción del sofbot al modelo -->
  </template>
</category>
```

Figura 2. Ejemplo de una categoría

En la Figura 3 se muestra un archivo AIML sencillo de ejemplo.

```
<?xml version="1.0">
<aiml version="1.0">
  <category>
    <pattern>HOLA</pattern>
    <template>Buenos días</template>
  </category>
</aiml>
```

Figura 3. Ejemplo de un archivo AIML sencillo

Para el sofbot SAMI, se ha construido un Módulo de Ingreso de Conocimiento que hace muy sencillo el ingreso de conocimiento a SAMI, desde la edición de los archivos AIML, con un entorno amigable que nos evita el engorroso trabajo de aprendernos todas las etiquetas del lenguaje AIML (<category>, <pattern>, <template>, <random>, <get>, <set>, <think>, <srail>, <sr>, etc), hasta el ingreso de estos archivos a la base de datos.

En un análisis más profundo se puede decir que las categorías AIML son lo que en RBC, llamaríamos "casos", y es la unidad básica de conocimiento en AIML. Cada categoría consiste de una pregunta de entrada, una respuesta de salida, y un contexto opcional.

La pregunta o pattern es simple, consiste sólo de palabras, espacios, y símbolos comodines, como el * (asterisco). Las palabras pueden consistir de letras y números, pero no de otros caracteres. Las palabras están separadas por un solo espacio en blanco, y los caracteres comodines funcionan como si fueran una palabra o un grupo de palabras.

La respuesta o template representa la respuesta que da el sofbot al usuario. En su forma más simple, el template consiste de texto plano y desmarcado (es decir sin etiquetas).

Más generalmente, el template puede estar compuesto de etiquetas AIML, como random, get, set, think, srail, etc las cuales transforman la respuesta

del sofbot en un programa de computadora el cual puede salvar datos en el disco, activar otros programas, dar respuestas condicionales, e incluso llamar recursivamente a otros pattern(patrones), etc.

La porción del contexto opcional de la categoría consiste de dos variantes llamadas mediante las etiquetas <that> y <topic>. El uso del contexto puede resultar muy útil para el buen desenvolvimiento del sofbot en las conversaciones.

La etiqueta <that> que aparece dentro de las categorías indica la última respuesta que dió el sofbot en la conversación. Recordar la última repuesta del sofbot es importante si el sofbot va hacer una pregunta o repregunta.

La etiqueta <topic> aparece fuera de la categoría, y sirve para establecer un grupo de categorías en base a un tema. Esta etiqueta hace referencia al tópico o tema de la conversación actual.

5. Recuperación del conocimiento

La recuperación del conocimiento almacenado en SAMI es un proceso que abarca desde la entrada del usuario, la búsqueda del patrón correspondiente con la entrada mediante el algoritmo de emparejamiento, hasta la generación de la respuesta.

Las entradas son la(s) palabra(s) ingresadas por el usuario al sofbot, y ya que estas son escritas en lenguaje natural deben ser normalizadas, es decir, pasan por un proceso de transformación antes de ser enviadas al "Módulo de Emparejamiento de Patrones."

Este proceso de normalización se realiza en el siguiente orden:

1. **Sustituciones.** Son reemplazos que se aplican a la entrada del usuario para ajustar las palabras a los patrones almacenados en la base de conocimientos antes que se realice el emparejamiento. Estos reemplazos son muy útiles para solucionar algunos problemas de corrección de ortografía, acortamiento de palabras, uso de jergas, siglas, abreviaturas, excepciones, etc.
2. **División de entradas complejas.** Esto consiste en una serie de heurísticas aplicadas a la entrada para dividir las en dos o más oraciones simples.
3. **Ajuste.** El proceso de ajuste consiste en eliminar de la entrada caracteres no permitidos en los patrones.

Todo el conocimiento que el Maestro ingresa en el sofbot, se guarda en una base de datos, llamada Base de conocimientos, cuya estructura se explicó en el

apartado anterior.

Cuando el usuario ingresa una pregunta al softbot, primero se normaliza, y luego se procede a realizar la búsqueda en la base de conocimientos para encontrar una categoría que contenga el patrón que mejor empareje la pregunta.

El emparejamiento es un algoritmo de IA, que consiste en una búsqueda en profundidad (depth-first search en idioma inglés) que recorre la entrada palabra por palabra, y baja por la estructura del árbol desde el nodo raíz hasta llegar al nodo hoja el cual contiene la respuesta. Esto está representado gráficamente en la figura 4:

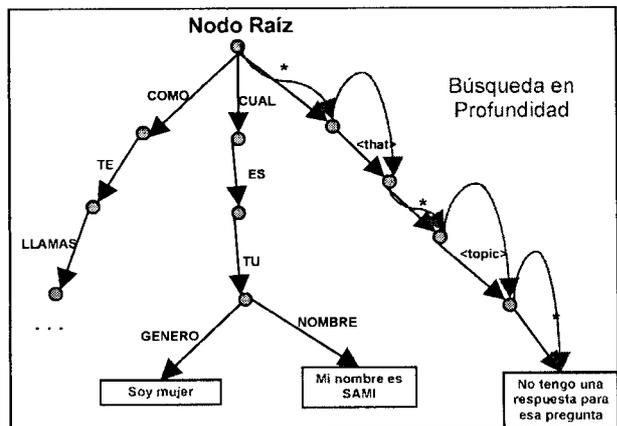


Figura 4. Emparejamiento

Para el softbot SAMI este algoritmo ha sido implementado en el componente llamado “Módulo de Emparejamiento de Patrones.”

6. Construcción de SAMI

Aquí se explica la manera en que se ha construido el softbot SAMI, llamado así por sus siglas “Sistema Asistente Modelo Inteligente”; se verá su arquitectura y se explican cada uno de los componentes creados. La elaboración de SAMI siguió una serie de fases de desarrollo, desde el análisis, diseño, hasta la implementación.

El proyecto SAMI, tuvo como principal objetivo la construcción de un softbot de charla que sea capaz de conversar en lenguaje natural y escrito, con los usuarios del Aula Virtual de la UNMSM. Los diálogos con SAMI tienen como propósito absolver las dudas de los usuarios respecto al funcionamiento del Aula Virtual, es decir, SAMI les sirve de asistente.

En la siguiente figura se muestra la arquitectura del softbot SAMI, la cual ha sido construida “a la medida”, y siguiendo el modelo 3 capas, en la figura 6.1 se muestran los principales componentes, sus relaciones entre sí y también los actores que interactúan con el sistema [2].

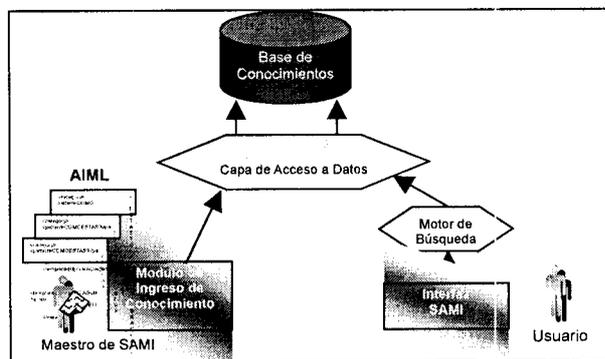


Figura 5. Arquitectura del softbot SAMI

a) Base de Conocimientos

La base de conocimientos es uno de los componentes más importantes del softbot, ya que centraliza toda la información necesaria para que el softbot SAMI pueda contestar a las preguntas de los usuarios. El Maestro es el encargado de almacenar, actualizar y modificar toda esta información.

b) Capa de Acceso a Datos

Esta capa sirve de interfase a los demás componentes que desean realizar accesos hacia la base de datos (Base de conocimientos) para el ingreso y/o actualización de la información, así como para las consultas de la misma.

c) Motor de Búsqueda (emparejamiento)

Este componente encapsula la funcionalidad de tomar las preguntas de los usuarios, realizar la recuperación del conocimiento mediante el algoritmo de emparejamiento y devolver la respuesta.

d) Módulo de ingreso de conocimiento: Maestro de SAMI

Este módulo es utilizado por el Maestro, y le permite administrar el conocimiento de SAMI, dándole las opciones de agregar, eliminar o modificar las categorías de la base de conocimiento, así como los parámetros iniciales del sistema; también cuenta con reportes estadísticos que le permiten ver el avance del llenado de la base. El módulo de ingreso de conocimiento ha sido desarrollado como una aplicación de escritorio.

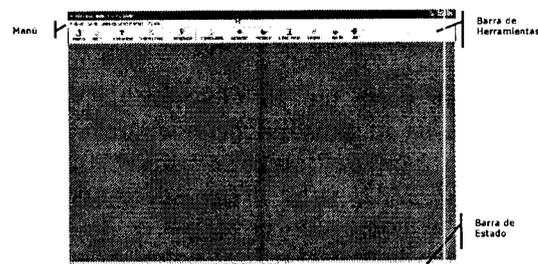


Figura 6. Maestro de SAMI

e) Interfaz: SAMI Usuario Aula Virtual

La interfaz Usuario SAMI es una aplicación Web, y dado que este es el único componente que se muestra al usuario final, se ha puesto especial interés en que sea lo más usable posible. Su función es la de pasar las preguntas de los usuarios al sistema, luego, una vez que el sistema tiene la respuesta, esta aplicación se encarga de realizar las acciones correspondientes y mostrar la respuesta al usuario.

Para su desarrollo se ha utilizado la tecnología ASP.NET que nos permite crear páginas Web independientes de la plataforma del usuario final, también se ha incluido AJAX que evita el refresco de la página de SAMI cuando el usuario envía las preguntas; asimismo se han implementado un conjunto de librerías javascript que la hacen realmente amigable al usuario final.

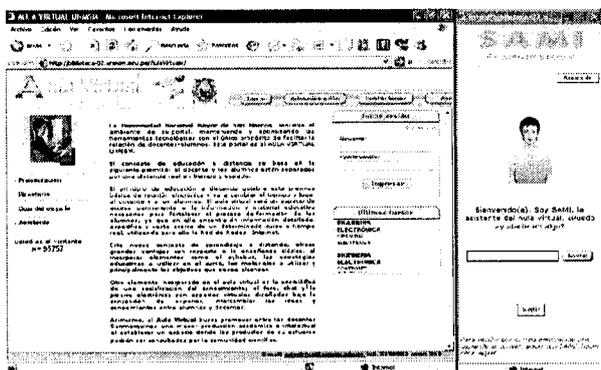


Figura 7. Asistente SAMI

Incorporación del Asistente SAMI en el Aula Virtual

La incorporación de SAMI en el Aula fue bastante sutil, es decir, el usuario que ingresa al portal del Aula, podrá desde allí mismo, invocar al asistente SAMI para realizarle preguntas en lenguaje natural y conseguir su ayuda. El asistente SAMI, se despliega como un menú en la parte derecha de la página del aula, para no interrumpir la visibilidad del usuario.

7. Análisis y evaluación de los resultados

Después de la construcción de SAMI, esta es la tarea que requirió más tiempo y esfuerzo, ya que de ella dependía el buen funcionamiento del sofbot.

El primer paso fue la elección de la persona encargada de administrar el conocimiento de SAMI, es decir el "Maestro", quien se valió de la herramienta *Maestro de SAMI* para este fin. Además del Maestro, también se contó con el apoyo de la Unidad de Informática de la Biblioteca Central de la UNMSM, quienes administran el proyecto "Aula Virtual de la

UNMSM" y cumplieron un rol importante proporcionando mucha información del funcionamiento del Aula Virtual.

Luego, se establecieron una serie de Etapas para el llenado de la base, cada una de ellas requirió de un proceso de preparación de la base de conocimientos, pruebas con usuarios, y finalmente el análisis de los resultados. Todo esto con el objetivo de mejorar y afinar gradualmente la base de conocimientos de SAMI.

Cabe mencionar que, en sistemas RBC, el proceso de llenado de la base de conocimientos es gradual e interactivo, con retroalimentación en cada interacción. No se tiene una especificación exacta del número de interacciones que se deben realizar, ya que esto depende de si las respuestas del sofbot son las esperadas o si el conocimiento va cambiando en el tiempo, con lo cual la base de conocimientos puede estar en constante actualización. Para el caso de SAMI, se determinaron las interacciones o etapas tomando en cuenta la cantidad de respuestas acertadas que SAMI proporcionó a los usuarios en el contexto del Aula Virtual.

En la tabla 7.1 se muestran los objetivos que se establecieron para cada una de las etapas.

Etapas	Resumen
Etapa 1	Configurar las variables iniciales del sistema SAMI e ingresarle "conocimiento" genérico.
Etapa 2	Retroalimentación de la Base de Conocimientos con la información de la primera etapa.
Etapa 3	Incluir en la base de conocimiento temas relacionados al Aula Virtual y proporcionarle a SAMI la capacidad de "reacción al usuario" mediante su interfaz.
Etapa 4	Afinar la base de conocimiento y proporcionarle mayor carga de usuarios al sistema SAMI.

Tabla 2. Reseña de las Etapas

En la **Tabla 3** se muestra un cuadro del resumen de todas las etapas con sus respectivas pruebas:

Etapas	Cantidad de Usuarios	Cantidad de Diálogos	Cantidad de Archivos AIMI	Cantidad de Categorías	Nodos en el Árbol de Búsqueda	Preguntas Sin Respuesta
Etapa 1	2	10	5	69	141	45
Etapa 2	4	35	15	472	992	77
Etapa 3	15	190	20	544	1168	81
Etapa 4	213	368	29	701	1553	90

Tabla 3. Resumen de las etapas y los resultados

De este cuadro se desprenden los siguientes gráficos para el análisis de resultados:

Carga de Usuarios que dialogaron con SAMI:

El gráfico de la Figura 8 muestra como se fue incrementando el número de usuarios que probaron la aplicación en todas las etapas, así como la cantidad de veces que visitaron la página para dialogar con SAMI.

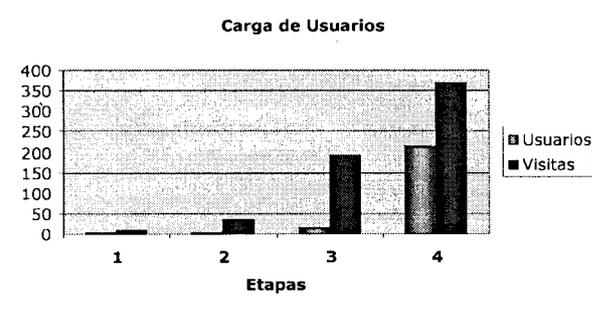


Figura 8. Uso del sofbot SAMI

Generación de archivos AIML:

En cada etapa, producto de las retroalimentaciones, se fueron descubriendo más categorías que fueron separadas en archivos AIML, el siguiente gráfico muestra que esto se produjo en forma incremental.

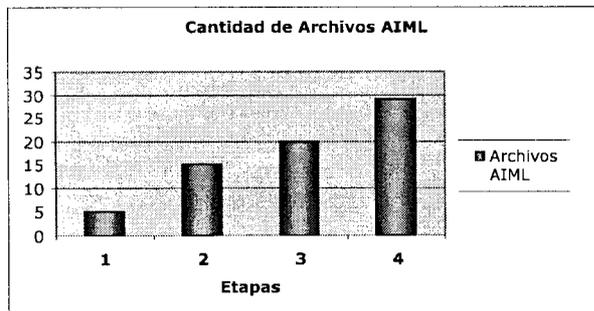


Figura 9. Variación de la cantidad de archivos AIML

Evolución de la Base de conocimientos:

Como se mencionó anteriormente, las categorías son las unidades mínimas de la base de conocimientos de SAMI, en la Figura 10 se muestra el gráfico que indica como se fueron incrementando las categorías y por lo tanto la base de conocimientos a lo largo de las etapas.

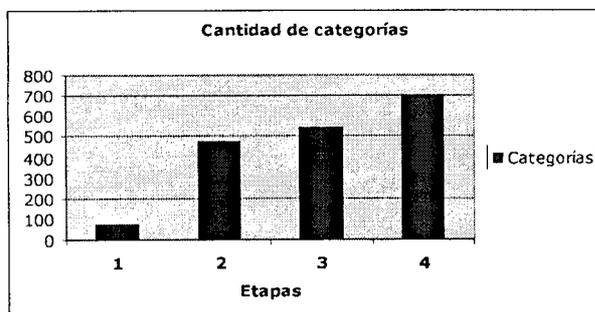


Figura 10. Variación de la cantidad de categorías

Incremento de la complejidad del Árbol de Búsqueda:

El siguiente gráfico muestra el incremento de la cantidad de nodos en el árbol de búsqueda de SAMI.

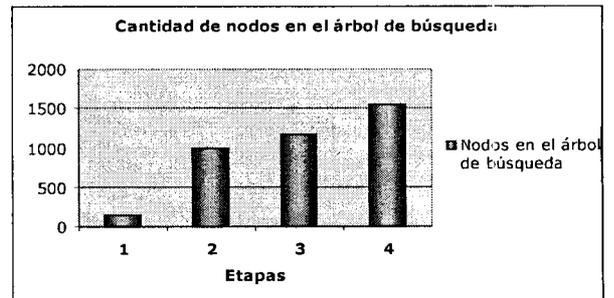


Figura 11. Variación de la cantidad de nodos en el Árbol de búsqueda

Decremento en la cantidad de preguntas sin respuesta:

Para medir la efectividad de SAMI respondiendo a las preguntas de los usuarios, se utilizó la siguiente fórmula aplicada a los resultados de cada etapa:

$$\frac{\# \text{ de Preguntas sin respuesta}}{\# \text{ de Usuarios}} \times 100\%$$

La Figura 12 se muestra el resultado (en porcentaje) de esta fórmula para cada etapa. Con lo cual se observa que mientras más conocimiento era ingresado en SAMI, ella respondía mejor y a mayor cantidad de preguntas.

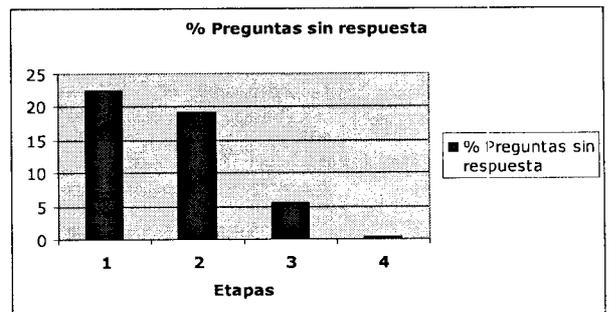


Figura 12. Disminución de preguntas sin respuesta

Luego, SAMI no sólo se comportó como un sofbot de charla, sino que también cumplió su función de asistente, brindando información de ayuda inmediata al usuario final.

8. Conclusiones y trabajo futuro

A lo largo de este artículo hemos visto la forma en que se ha implementado el sofbot de charla SAMI. Como puntos importantes cabe resaltar:

- El estudio de la técnica de Inteligencia Artificial,

Razonamiento Basado en Casos, la cual se enfoca en imitar el modelo de razonamiento humano, que es capaz de resolver problemas utilizando el conocimiento adquirido en situaciones previas. Este mismo comportamiento es el que presenta SAMI cuando responde a las preguntas de los usuarios, busca en su base de conocimientos la solución que más se aproxime al caso y luego genera la respuesta.

- La definición de la estructura de los casos (llamados categorías en lenguaje AIML) y su almacenamiento en la base de conocimientos, que permitieron una mejor administración de la información.
- Un eficiente mecanismo para la recuperación de los casos almacenados, desde que el usuario realiza la pregunta, la búsqueda del caso correspondiente mediante un algoritmo de emparejamiento, hasta la generación de la respuesta. Mediante las pruebas demostró ser bastante óptimo y eficiente al generar las respuestas en forma rápida.
- El diseño de una arquitectura de varias capas para el sistema SAMI, que permitió la reutilización de sus componentes.
- El desarrollo del aplicativo "Maestro de SAMI", construido ad hoc con el fin de ayudar a administrar el conocimiento de SAMI en forma sencilla.
- La construcción de la interfaz del sofbot de charla SAMI, utilizando tecnología ASP.NET y AJAX, que la hicieron realmente eficiente y amigable al usuario final.
- Las pruebas con usuarios finales, que ayudaron (mediante algunas métricas) a medir la eficiencia de SAMI respondiendo a la mayoría de las preguntas satisfactoriamente.
- Finalmente, se comprobó la aplicación del sofbot de charla SAMI en un caso real, es decir, su uso como asistente virtual en el Aula Virtual de la UNMSM.

Con todo, se puede decir que el desarrollo de los sofbots de charla en Perú está aún en sus inicios. Con la construcción de SAMI se pretende dar un gran paso para su difusión y aplicación, también utilizando de guía los trabajos que se han desarrollado en otros países donde incluso se usan estos sofbots a nivel corporativo. Por otro lado, este trabajo sirve para demostrar que una máquina es capaz de expresarse de forma parecida al ser humano sin llegar a serlo.

Ciertamente a partir de aquí son muchas las posibilidades que se abren, ya que este trabajo no acaba aquí, sino que está sujeto a distintas mejoras y ampliaciones, como son la ampliación de la base de conocimientos, o se le puede insertar conocimiento referente a otros temas en los cuales también sea

aplicable, como por ejemplo a un portal de ventas, e-learning, entretenimiento, etc. Incluso se le puede unir las tecnologías de reconocimiento de voz, video y animación para mejorar su interacción con los usuarios.

Referencias

- [1] AAMODT, A., PLAZA E. "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches". Artificial Intelligence Communications. 1994
- [2] CAMACHO, S. SAMI: Un Robot de Charla desarrollado con la técnica de Razonamiento Basado en Casos. UNMSM. Perú. 2006.
- [3] DE LA OSSA, A. "Manuscrito del Aprendizaje Mecánico". Programa de Posgrado en Ciencias Cognoscitivas, Universidad de Costa Rica, 1993
- [4] LIDDY, E.D. Natural Language Processing. In Encyclopedia of Library and Information Science, 2nd Ed. NY. 2001.
- [5] PORTER, B.W. Similarity Assessment: computation vs. representation. Procs of the Workshop on case-based reasoning (DARPA), Morgan Kaufmann, Florida. 1989.
- [6] SCHANK R. Dynamic memory; a theory of reminding and learning in computers and people. Cambridge University Press. 1982.
- [7] WEIZENBAUM, J. "ELIZA - A Computer Program for the Study of Natural Language Communication between Man and Machine," Communications of the Association for Computing Machinery. 1966.
- [8] WINSTON P. Inteligencia Artificial. Addison-Wesley. 1994
- [9] WATSON, I. Applying Case-Based Reasoning: Techniques for Enterprise Systems. Morgan Kaufman Publishers. 1997.