

41 JAIIO EST 2012 15° Concurso de Trabajos Estudiantiles

aiStudent: Agente inteligente para la interacción en Lenguaje Natural

Autores:

María Julia Blas – mariajuliablas@gmail.com

Nicolás Emilio Díaz Ferreyra – nicoediaz@gmail.com

Juan Leonardo Sarli – juanleonardosarli@gmail.com

Institución:

UTN Regional Santa Fe

Carrera:

Ingeniería en Sistemas de Información

Cátedra:

Inteligencia Artificial

Docentes:

Dra. María de los Milagros Gutierrez

Ing. Jorge Marcelo Roa

aiStudent: Agente Inteligente para la Interacción en Lenguaje Natural

Abstract. Se presenta la implementación de un agente conversacional desarrollado en el marco de la asignatura Inteligencia Artificial; el cual es capaz de mantener un diálogo con un estudiante sobre temas inherentes a la organización de la materia. El vocabulario, los mecanismos de búsqueda y el sistema de reglas de producción se detallan teniendo en cuenta la complejidad existente para el procesamiento del lenguaje natural. Se especifica un algoritmo de preprocesamiento especialmente diseñado con el objetivo de simplificar el universo de discusión del agente. Los resultados obtenidos son validados en función de la habilidad y rapidez con la que el agente responde durante la interacción.

Keywords. Agente inteligente, inteligencia artificial, lenguaje natural, chat-bot.

1 Introducción

La Inteligencia Artificial (IA) es una rama de las ciencias de la computación que intenta encontrar esquemas generales de representación del conocimiento, y formalizar procesos de razonamiento coherentes, que permitan resolver problemas difíciles en dominios de aplicación concretos [1]. Para esto, define cuatro categorías [2]: sistemas que piensan como humanos, sistemas que actúan como humanos, sistemas que piensan racionalmente y sistemas que actúan racionalmente. Estos últimos, intentan emular en forma racional el comportamiento humano; siendo este, el caso de los agentes inteligentes (AI).

Un AI, es una entidad capaz de percibir su entorno, procesar tales percepciones, y actuar en consecuencia de manera racional sobre el mismo [3]. En los últimos años, con el objetivo de resolver problemas de alta complejidad, se ha incrementado la investigación y el desarrollo de este tipo de herramientas. Una temática recurrente ha sido la implementación de sistemas que permitan a una computadora mantener una conversación en lenguaje natural [4,5].

Se denomina agente conversacional, o chat-bot (CB); a un AI que simula un diálogo inteligente con humanos a través de métodos auditivos o textuales. Todo CB debe contener un mecanismo de procesamiento de lenguaje natural que le permita comprender, mediante el estudio de la sintaxis, la conversación en la que se encuentra involucrado.

En este contexto, como trabajo práctico, la cátedra Inteligencia Artificial de la Facultad Regional Santa Fe – Universidad Tecnológica Nacional ha solicitado a los alumnos que desarrollen un agente CB llamado aiStudent, que tenga la capacidad de mantener una conversación con un estudiante sobre la organización de la materia. El presente trabajo detalla la implementación realizada indicando, no sólo cuales han sido las herramientas utilizadas, sino también los factores que el equipo de desarrollo ha considerado relevantes durante la construcción del agente.

En las siguientes secciones se describe el vocabulario del agente, la implementación del mecanismo de búsqueda [2] que utilizará para encontrar el conjunto de reglas aplicables a una entrada, y el sistema de reglas de producción [2] con el que decidirá la respuesta mas adecuada.

2 Desarrollo e Implementación de aiStudent

2.1 Definición del Vocabulario

Con el objetivo de establecer la base de conocimientos del CB, se definió el conjunto de palabras que el agente reconocerá. Para esto, se realizaron una serie de actividades previas con la intención de identificar patrones de conversación en lenguaje natural. El resultado de esta etapa, fue la definición del grafo gramatical que muestra las relaciones entre las palabras claves; y el algoritmo de preprocesamiento propuesto para ajustar cualquier frase ingresada a un conjunto de palabras claves válidas.

Estudio del Lenguaje Natural.

Se analizó la estructura general de una conversación, identificando el intercambio de frases como área problemática debido a que puede contener diferentes estilos de expresiones (Fig. 1).



Fig. 1. Modelado de una conversación.

Ámbito Elegido.

Debido a la amplitud de la clasificación previamente esquematizada, se limitó el alcance del CB. Una conversación de un usuario con aiStudent abarcará saludos y despedidas; pero no se reconocerá la totalidad de las frases, sino que sólo se interpretarán algunas preguntas. En función de los elementos gramaticales encontrados dentro una expresión interrogativa, se decidió incorporar preguntas del tipo elaboradas-básicas al

ámbito del agente. Esto quiere decir que solo se reconocerán aquellas expresiones que inicien con: cómo, cuál, dónde, qué, cuándo, cuáles, cuántas y quién. Posteriormente se definieron 200 preguntas a las cuales el agente deberá responder.

Definición de la Gramática.

Teniendo en cuenta que el lenguaje español es extremadamente extenso y posee una gran cantidad de elementos gramaticales; se planteó la necesidad de establecer un mecanismo que garantice el uso de la menor cantidad de términos posibles durante una conversación. No es deseable tener una gramática extensa, debido a que aumenta proporcionalmente la base de conocimientos del CB, generando tiempos de respuesta altos. Por esto, se estableció un conjunto mínimo de palabras claves sobre las cuales aiStudent decidirá una respuesta. Esto obligó a generar dos tipos de listas.

Por un lado, se generaron listas de sinónimos; con el propósito de identificar con una única palabra, términos que dentro del contexto de trabajo tienen significados iguales o parecidos. En la Tabla 1 se presentan algunas de las listas definidas.

SINÓNIMOS	PALABRA CLAVE	PALABRAS INCLUIDAS
LISTA N°1	MATERIA	INTELIGENCIA ARTIFICIAL - CÁTEDRA - ARTIFICIAL - IA
LISTA N°2	FRAMEWORK	SOFTWARE - SOFTWARES - SW
LISTA N°3	PARCIAL	EXÁMEN - EXÁMENES - PARCIALES

Tabla 1. Ejemplos de listas de sinónimos.

Por otro lado, se generaron listas de verbos; con el objetivo de utilizar la conjugación en infinitivo para representar todas las posibles conjugaciones de un verbo. En la Tabla 2 se presentan algunos ejemplos de este tipo de listas.

VERBOS	PALABRA CLAVE	PALABRAS INCLUIDAS
LISTA N°1	APROBAR	APRUEBO - APROBAMOS - APRUEBA - APRUEBAN - APROBADOS - APROBADAS - APROBARSE
LISTA N°2	DESARROLLAR	DESARROLLO - DESARROLLAMOS - DESARROLLA - DESARROLLAN - DESARROLLADAS - DESARROLLADOS - DESARROLLARSE
LISTA N°3	EVALUAR	EVALUACIÓN - EVALÚA - EVALÚAN - EVALUADAS - EVALUADOS - EVALUARSE

Tabla 2. Ejemplos de listas de verbos en infinitivo.

Algoritmo de Pre-Procesamiento.

El mecanismo de trabajo basado en el uso de listas, y la necesidad de reemplazar palabras de la frase original por sus equivalentes o infinitivos, llevó a la necesidad de definir un algoritmo que realice estas conversiones en forma automática. Este algoritmo de preprocesamiento, se encuentra esquematizado en la Fig. 2.

```

String oracionUsuario= getOracionIngresada();
IF (esPregunta(oracionUsuario)==TRUE)
    resultadoParcial1= getVerbosInfitivo(oracionUsuario);
ELSE
{
    resultadoParcial1= getFrase(oracionUsuario);
    IF (resultadoParcial1==Cadena Vacía)
        return No Buscar;
}
resultadoParcial2 = getPalabrasClave(resultadoParcial1);
IF (resultadoParcial2==Cadena Vacía)
    return No Buscar;
IF (esFrase(resultadoParcial2)==TRUE)
    return Buscar;
ELSE IF (esPreguntaBienFormulada(resultadoParcial2)==TRUE)
    return Buscar;
return No Buscar;

```

Fig. 2. Pseudocódigo del algoritmo de preprocesamiento.

Grafo Gramatical.

La definición de una gramática y un algoritmo de preprocesamiento llevaron a que el grafo dirigido que representa las relaciones existentes entre palabras claves tenga una dimensión reducida. Con 55 nodos y 121 arcos, esta estructura permite dar respuesta a las 200 preguntas planteadas como posibles entradas de conversación para el agente.

2.2 Desarrollo del Agente que Utiliza Búsqueda

El agente aiStudent utiliza dos estrategias de IA:

- (a) Búsqueda, con el objetivo de encontrar el conjunto de reglas aplicables a una oración dada;
- (b) Sistemas de producción, con el fin de seleccionar una única respuesta dentro de la colección que incluye todas las reglas posibles.

Se detalla en esta sección la implementación del mecanismo de búsqueda que utiliza el CB, junto con los mecanismos de prueba ideados para validar lo desarrollado.

En la sección 2.3, se presenta la solución del sistema de producción planteado.

Modelado del Agente.

Se modeló el problema de encontrar el conjunto de reglas a aplicar ante el ingreso de una determinada frase, planteando un problema de búsqueda. Se definieron en consecuencia, conceptos tales como el estado del agente, la prueba de meta, el conjunto de operadores y el formato de cada percepción enviada desde el ambiente.

Diseño Conceptual del Agente.

Para modelar al agente CB, se utilizó un ambiente de desarrollo integrado basado en la plataforma Eclipse llamado IDEMIA [6]. Este entorno brinda un conjunto de herramientas y funcionalidades gráficas que dan soporte al modelado de AI.

Se construyó el diseño conceptual de aiStudent (Fig. 3) en base a una perspectiva de alto nivel en la que se identificaron, entre otras cosas, percepciones, estados y pruebas de meta.

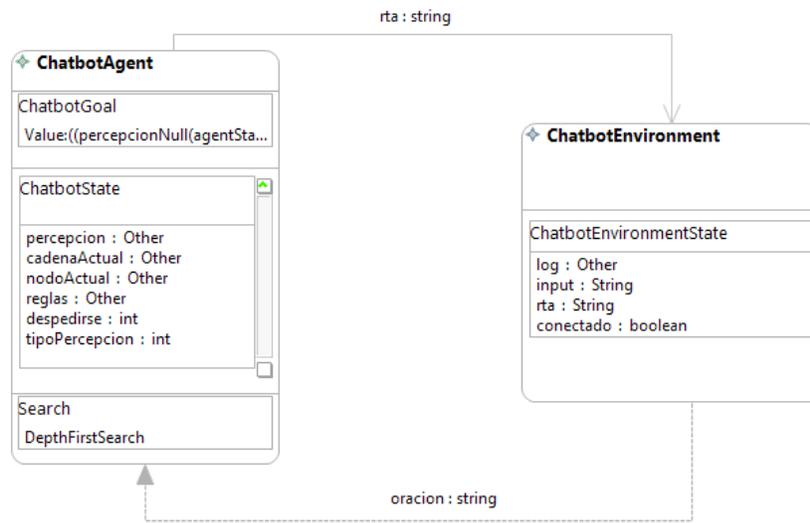


Fig. 3. Diseño conceptual de aiStudent.

Estado del Agente.

Quedó definido por un conjunto de siete atributos que determinan la situación del agente en un momento dado. Estos atributos son:

- (a) Grafo: Representa el grafo definido como resultado de la gramática planteada para el agente. Cada nodo se compone de una palabra clave que el CB reconocerá, y del conjunto de reglas que dan respuesta a ese término. Navegando esta estructura mediante el uso de operadores de búsqueda el agente encuentra las posibles respuestas a una determinada frase.
- (b) Despedirse: Contador de preguntas no entendidas; que ayuda al agente a tomar la decisión de abandonar la conversación.
- (c) Tipo de percepción: Indica de que tipo es la oración ingresada por el usuario (frase o pregunta).
- (d) Percepción: Oración que ingresó el usuario.
- (e) Cadena actual: Cadena de palabras que el agente ha generado durante el proceso de búsqueda. Al navegar el grafo por medio de la aplicación de operadores, el agente CB crea una oración compuesta de palabras claves donde cada término es un nodo previamente visitado. Este atributo almacena dicha frase.
- (f) Nodo actual: Nodo del grafo en el que se encuentra posicionado el agente.
- (g) Reglas: Conjunto de reglas que han sido recolectadas durante el proceso de búsqueda.

Percepción.

Cada percepción que el ambiente le envía al agente, será simplemente la cadena de texto ingresada por el usuario.

Prueba de Meta.

La meta del agente de búsqueda es encontrar el conjunto de reglas aplicables a una oración ingresada por el usuario (percepción). De esta manera, se estableció como estado objetivo del agente que la cadena actual sea igual a la percepción recibida; lo que significa que durante el proceso de búsqueda, el agente ha recorrido todas las palabras claves que el usuario ingresó.

Operadores.

Se definió un operador para cada destino posible, es decir, para cada nodo del grafo. El comportamiento de cada uno de ellos es similar, por lo que en la Fig. 4 se visualiza a modo de ejemplo el pseudocódigo del operador irHola.

```

irHola()
{
    Node actualNode= getMyNode();
    IF (graphConnectsWhith(actualNode, "hola")==TRUE)
    {
        setMyNode("hola");
        Sentence cadenaActual= getMysentence();
        Rules listaReglas= getMyrules();
        setMySentence(cadenaActual + "hola");
        setMyRules(listaReglas + getRulesFor(actualNode));
    }
}

```

Fig. 4. Pseudocódigo del operador irHola.

Implementación del Agente.

IDEMIA además de las funcionalidades de modelado, brinda la facilidad de generación semiautomática de código Java. Para esto utiliza FAIA [7], que es un framework que permite resolver problemas de IA en base a agentes basados en objetivos. Este entorno además, contiene predefinidos los algoritmos de búsqueda requeridos.

Se resolvió entonces, partiendo del diseño conceptual previamente realizado, realizar la implementación del agente aiStudent utilizando dicho framework (provisto por la cátedra).

En la Fig. 5, se presenta el pseudocódigo del mecanismo que el agente CB utiliza al momento de decidir la acción que ejecutará. En el caso de que la percepción no sea válida, aiStudent deberá decidir entre abandonar la conversación (despidiéndose), o indicarle al usuario que no ha comprendido lo que le ha dicho.

```

AgentState estadoAgente= getMiEstado();
IF (tienePercepcionValida(estadoAgente)==TRUE)
{
    Action selectedAction= null;
    selectedAction= ejecutarBusqueda(estadoAgente);
    return selectedAction;
}
ELSE
{
    IF (deboDespedirme(estadoAgente))==TRUE)
        agregarReglasDespedida(estadoAgente);
    ELSE
        agregarReglasNoComprendo(estadoAgente);
}
return null;

```

Fig. 5. Pseudocódigo de la toma de decisión.

Selección de la Estrategia de Búsqueda.

Como estrategia de búsqueda se decidió utilizar métodos no informados debido a que funcionan en mundos desconocidos. Éste es el caso del CB, puesto que no existe ningún tipo de conocimiento sobre el mundo con el cual se está interactuando debido a que las percepciones son aleatorias. Se optó por utilizar Búsqueda en Profundidad (MBP) ya que es una estrategia completa (dado un espacio finito de búsqueda, si existe una solución seguro la encontrará), y relativamente rápida en cuanto a tiempos. Sin embargo no es una estrategia óptima (salvo que la primera solución que encuentre sea la mejor); pero en este caso, esta particularidad no afecta la solución del problema, debido a que existe una única solución para cada situación.

Como el grafo gramatical es conexo, existe la posibilidad de que se generen bucles durante el proceso de búsqueda. Para evitarlos, se modificó el algoritmo original haciendo que una misma rama del árbol no pueda pasar dos veces por un mismo nodo del grafo.

Pruebas.

Se realizaron dos tipos de pruebas semiautomáticas, una sobre el preprocesamiento y otra sobre la búsqueda. En ambos casos, las frases ingresadas son leídas desde un archivo de texto que contiene todas las oraciones que reconoce el CB.

El preprocesamiento fue testeado por medio de un método que ejecuta dicho algoritmo por cada una de las entradas leídas, y graba en un archivo el resultado obtenido. De esta manera se puede evaluar si el conjunto de palabras claves generado para una pregunta o frase dada es el esperado.

La búsqueda fue probada utilizando otro método, el cual ejecuta el simulador por cada una de las entradas, y graba en un archivo de salida el resultado obtenido. Esto permite evaluar si el conjunto de reglas obtenido para una oración dada es el esperado.

2.3 Modelado del Sistema de Producción

Cada nodo del grafo gramatical contiene un conjunto de reglas que corresponden a posibles respuestas que el agente CB puede dar al usuario. Estas reglas han sido definidas como se visualiza en la Tabla 3; en donde la condición representa el conjunto de palabras claves que determinan la aplicación de la acción, y la acción representa la respuesta al usuario.

REGLA	CONDICIÓN	ACCIÓN
14	QUE - HACER	ESTUDIANDO IA...A VER SI LOGRO ENTENDER LO QUE ESTAMOS DANDO
36	COMO - SER - INFORMAR - GRUPO	MANDANDO UN MAIL CON LA FORMACIÓN DEL MISMO INDICANDO NOMBRE Y APELLIDO + DIRECCIÓN DE E-MAIL DE TODOS LOS INTEGRANTES

Tabla 3. Ejemplos de reglas.

3 Resultados

En esta sección, se presentan los resultados obtenidos hasta el momento a lo largo de la realización del trabajo práctico.

En la Fig. 6 se observa un ejemplo de una conversación con aiStudent. Esta imagen además, presenta la interfaz de usuario desarrollada con el objetivo de tener un mecanismo de interacción usuario-agente simple e intuitivo. Se optó diseñar la pantalla con un formato similar al de las interfaces de la mensajería instantánea existentes, distinguiéndose con diferentes colores y formatos los textos que ingresa cada uno de los participantes de la conversación.



Fig. 6. Ejemplo de ejecución.

Esta conversación, genera tres procesos de búsqueda internos al CB (uno para cada una de las oraciones ingresadas). En la Tabla 4, se visualiza el preprocesamiento resultante de estas frases; mientras que en la Fig. 7, se observa el árbol de búsqueda generado para una de ellas resaltándose el orden de los operadores ejecutados y el camino objetivo.

ID	FRASE	RESULTADO PREPROCESAMIENTO
1	HOLA, COMO VA?	HOLA – COMO - ESTAR
2	TUDO BÁRBARO... TENES IDEA CUALES SON LOS PROFESORES DE IA?	CUAL – SER - PROFESOR
3	BUENO MUCHAS GRACIAS, NOS ESTAMOS VIENDO ENTONCES!	GRACIAS

Tabla 4. Preprocesamiento de una conversación.

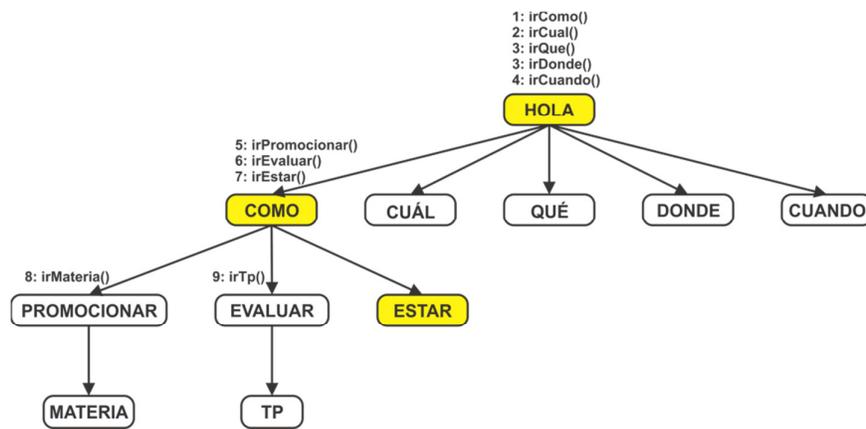


Fig. 7. Árbol de búsqueda de la frase 1.

Actualmente el agente CB, elige una respuesta al azar entre el conjunto de reglas resultantes de la búsqueda. Esto hace que el comportamiento no siempre sea el esperado, puesto que puede elegir para una pregunta dada, alguna respuesta no esperada. En etapas de desarrollo futuras, se incorporará el sistema de producción; que le dará al agente la capacidad de decidir sobre el conjunto de reglas obtenido en el proceso de búsqueda. Esta decisión se tomará en base al camino recorrido en el grafo de búsqueda, y al historial de chat de la conversación.

4 Conclusiones

El desarrollo de un agente conversacional puede llevarse a cabo simplemente mediante un proceso de comparación con el listado de frases interpretadas. Sin embargo, esto no es una solución eficiente. La interpretación y el procesamiento del lenguaje natural empleando herramientas computacionales presentaron un gran desafío, como así también la optimización del tiempo de respuesta del agente.

Con el objetivo de no perjudicar la dinámica de la interacción usuario-agente las estrategias de IA ayudaron a superar los inconvenientes referidos a la reducción de tiempos. El estudio realizado sobre el lenguaje natural, junto con la definición de una gramática; han simplificado la base de conocimiento del agente. Esto ha dado como resultado un vocabulario compuesto de un conjunto reducido de palabras claves, lo que en consecuencia ha disminuido el espacio de búsqueda haciendo que este proceso sea más rápido y eficiente.

El uso de IDEMIA y FAIA, tanto para el modelado como para la implementación, ha facilitado el desarrollo permitiendo al equipo de trabajo centrarse en los aspectos relevantes del diseño del comportamiento del agente.

De esta manera, con la ayuda de técnicas de Inteligencia Artificial, se ha avanzado en el desarrollo de un agente inteligente capaz de mantener una conversación en lenguaje natural sobre temas referentes a la organización de una cátedra universitaria. Partiendo del uso de un algoritmo de preprocesamiento especialmente diseñado, es capaz de determinar si podrá o no dar una respuesta a una frase ingresada por el usuario.

Mediante el uso de una interfaz el funcionamiento del chat-bot queda oculto, impidiéndole al usuario diferenciar la naturaleza del otro participante de la conversación.

Referencias

1. Moret, B.: Fundamentos de Inteligencia Artificial. 2° Edición. (2004)
2. Russell, S.; Norvig, P.: Inteligencia Artificial: Un Enfoque Moderno. Pearson Prentice Hall, 3° Edición. (2010)
3. Corchado, J.; Perez, J.: Advances on Practical Applications of Agents and Multiagent Systems. Springer. (2011)
4. Zakos, J.; Capper, L.: CLIVE – An Artificially Intelligent Chat Robot for Conversational Language Practice. (2008)
5. Shawar, B.; Atwell, E.: Chatbots: are they really useful? (2007)
6. Santana, W.; Roa, J.; Gutiérrez, M.; Stegmayer, G.: IDEMIA: Un entorno de desarrollo integrado para el modelado de agentes inteligentes. Revista Iberoamericana de Informática Educativa. (2012)
7. Roa, J.; Gutiérrez, M.; Stegmayer, G.: FAIA: Framework para la enseñanza de agentes en IA. Revista Iberoamericana de Informática Educativa. (2008)
8. Walece, R.: The anatomy of ALICE. ALICE Artificial Intelligence Foundation Inc. (2002)